

A Method for Obstacle Avoidance in Role Reassignment of Robot Formation Control

YU-CHENG CHEN and YIN-TIEN WANG

Department of Mechanical and Electro-Mechanical Engineering

Tamkang University

151 Ying-Chuan Rd., Tamsui, Taipei Hsien, 25137

TAIWAN

{131544; ytwang}@mail.tku.edu.tw; <http://mail.tku.edu.tw/ytwang>

Abstract: - In the paper, an algorithm for the obstacle avoidance and a procedure for the role reassignment of robot formation control based on this algorithm are proposed. This algorithm contains an edge detector that is used to determine the position of the vertical edges of the obstacle in the environment. Meanwhile, a motion controller is also designed in this research based on the dynamic equation for robots. The developed algorithm is applied to the formation control of omni-directional driven robots. Simulation and experiment are performed to verify the proposed algorithm and the results show that the performance of the proposed algorithm for obstacle avoidance and procedure for role reassignment are efficient for robot formation control.

Key-Words: - Obstacle avoidance, Formation control, Role assignment, Robot control

1 Introduction

Robot team formation means that a group of robots form a specific geometric pattern and maintain it while moving [1,2]. There are many advantages for a robot team to keep a formation while moving. For example, only the leader of the robot team needs to perform the task of self-localization, and the other robots obtain their own location by referring to the leader. Furthermore, it is a purpose of safety for a robot team formation to avoid collision with other robots while moving. The control of team formation is an important research topic within the research fields of multi-robot systems, problems in formation control that have been investigated include formation shape maintenance [3], vision-based formation control [4], formation switching [5], and multi-team formation control [6]. Team formation is applied widely in multi-robot navigation [7,8], manipulation [9], soccer robot [10,11], and automated highway systems [12].

In this paper, we address the methods for obstacle avoidance of robot formation control. When a robot formation is forming or changing, the moving robot may collide with each other. Therefore, the robot formation must have the mechanism providing the robots with the capability of obstacle avoidance. Obstacle avoidance includes three important issues which are “If the obstacle avoidance is needed?”, “When should the avoidance action begin?” and “Where should the robot go in order to avoid the obstacle?” The first issue

concerns with the geometry relation between the robot and obstacles. The second issue is related to the dynamics of the robot and is designed by the distance between the robot and obstacles. The third issue is related to the size of the robot and a suitable candidate point is selected. There are many popular obstacle avoidance methods in the literatures, for example, edge-detection, certainty grids, and potential field methods [13-15]. In this research, we utilize the edge-detection method to determine the position of the vertical edges of the obstacle and then steer the robot around either one of the edges.

It is also a challenging problem in formation control if the role for each robot must be reassigned when a robot formation is switching. In this situation, which robot should be assigned as the leader in new formation and which robot should be a follower? This is a critical problem for robots to keep a formation and a suitable role assignment mechanism is needed. In the field of formation control, researchers generally control the formation with fixed robot roles, and little research allows role changing in formation [10,16,3]. In [10], the robots can switch roles within formation according to trigger signals. The trigger conditions are predefined and limited to some simple cases. In [16] and [3], they show methods for role assignment performed based on a potential field, however, the nonlinearity of potential field results in computational complexity. In this paper, we propose an efficient procedure for robot role reassignment while a formation is forming or switching. The procedure is

based on the spatial relationship between robots in formation and the information of obstacles surrounding the robots.

The contributions in this paper are two-fold. First, we design an algorithm for obstacle avoidance of robot formation control. Especially, an efficient procedure based on edge detector is proposed for the robot formation control system to determine obstacles in the nearby environment. Our second contribution is the development of a role reassignment procedure for robot formation control. Meanwhile, the obligation to avoiding obstacles during formation switching is also considered in designing the role reassignment algorithm.

The paper is organized as follows. In Section 2, we design the methods for obstacle avoidance of robot formation control. The procedure to perform role reassignment of formation control is introduced in Section 3. Section 4 depicts the experimental validation through simulations and real experiments. Finally, some conclusions are provided in Section 5.

2 Obstacle Avoidance

In this section, the algorithm for obstacle avoidance is designed based on the sequential situations when the robots likely have a chance to collide with each other. First, the obstacle avoidance is enabled if the distance between robot and obstacle is smaller than a designed distance and the obstacle is located on the way where the robot is moving to. Second, the time to turn on the mechanism for obstacle avoidance is decided by a predefined distance. Finally, an alternative path is generated for the robot to avoid obstacles by using a vector operation.

The obstacle avoidance is enabled when the conditions are conformed: the distance between robot and obstacle is smaller than a designed distance and the obstacle is located on the way where the robot is moving to. The distance between the robot P and the obstacle O_r is determined as shown in Fig. 1. If the distance between P and O_r is smaller than the avoidance enable distance (aed), the obstacle is judged if it is on the direction where the robot is moving to. Define the angle α from vector d_2 to vector d_1 , where d_1 is the vector from the obstacle to the robot initial position, and d_2 is the vector from the obstacle to the robot desired position. If the condition $\pi/2 < \alpha < 3\pi/2$ holds, the robot needs to avoid the obstacle and the obstacle avoidance mechanism is enabled. On the other hand, when the condition $-\pi/2 < \alpha < \pi/2$ holds, which imply

that the robot already finishes the action of obstacle avoidance and the avoidance action is not needed.

For a moving obstacle as shown in Fig. 2, the robot will collide with the obstacle in Δt_1 seconds without avoidance,

$$\Delta t_1 = \frac{aed - 2L}{V_{\max} + V'_{\max}} \quad (1)$$

where L is the radius of a robot; V_{\max} and V'_{\max} are the maximum speeds of the robot and obstacle, respectively, in opposition direction on line segment O_rP ; aed is the avoidance enable distance. Suppose that the robot move a distance L in the perpendicular direction of O_rP to avoid obstacle, the acceleration of robot in the perpendicular direction of O_rP is

$$A_{\max} = \frac{f_a}{M}$$

where M is the mass of robot; f_a is the force applied on the mass center of the robot. Assume that the initial speed of robot on the perpendicular direction of O_rP is zero, therefore, the time for the robot move a distance L in perpendicular direction of O_rP is

$$\Delta t_2 = \sqrt{\frac{2L}{A_{\max}}} = \sqrt{\frac{2ML}{f_a}} \quad (2)$$

The condition for the robot to avoid the obstacle is

$$\Delta t_2 \leq \Delta t_1 \quad (3)$$

Substitute Eqs. (1) and (2) into Eq. (3), and the minimum value of aed must be

$$aed \geq 2L + \sqrt{\frac{2ML}{f_a}}(V_{\max} + V'_{\max})$$

aed can be multiplied by a safety factor for successfully avoiding the obstacle.

The candidate position the robot moving to can be determined as shown in Fig. 3. Since P , P_d and O_r are known, the projection of line segment PO_r on line segment PP_d can be determined by

$$|PI| = |PO_r| \cos \beta = \frac{PO_r \cdot PP_d}{|PP_d|}$$

The length of vector PI can be found by using the inner product of PO_r and the unit vector of PP_d . Define a safe vector, SD , which has the same direction with O_rI . Therefore, $|SD|$ represents a safe vertical distance from the center of obstacle for the robot to avoid the obstacle. Define an obstacle avoidance vector, $PI+SD-O_rI$, hence, the candidate point can be determined by adding up the robot position and the obstacle avoidance vector. The safe distance $|SD|$ must larger or equal to aed . If $|SD|$ is smaller than aed , assume that $|SD|=aed$.

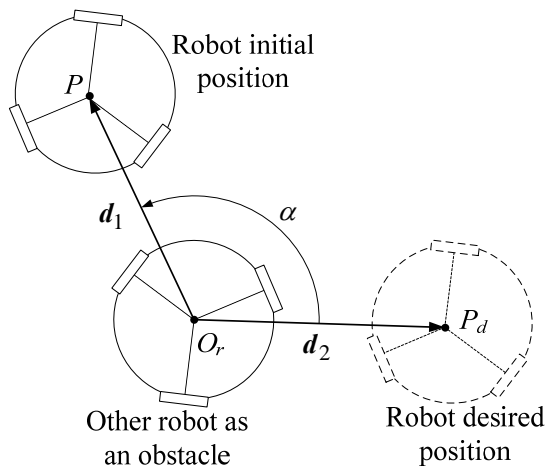


Fig. 1 Obstacle avoidance conditions

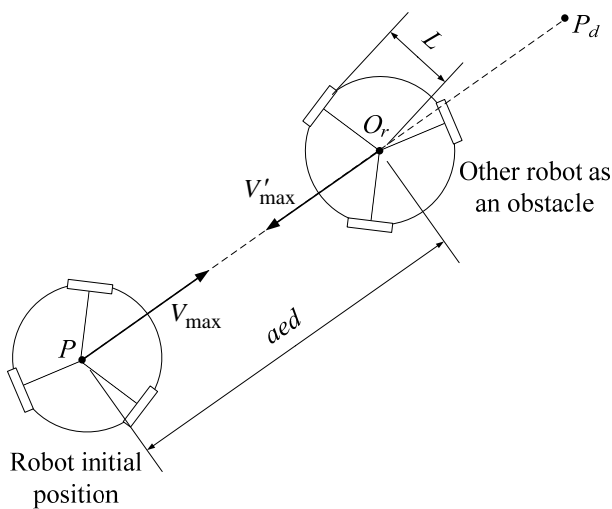


Fig. 2 Avoidance enable distance

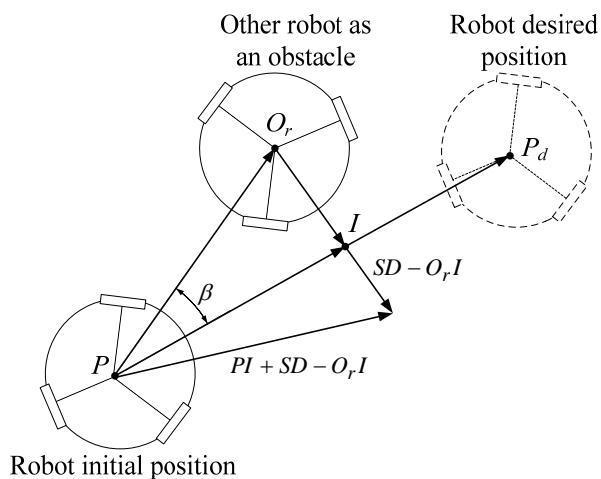


Fig. 3 Vector obstacle avoidance method

3 Formation Control

When the robot team forms the formation from a tangled initial condition or a switching formation

shape, the role of each robot must be resigned. In this section, a procedure for role reassignment is designed for robot formation control. We discuss the representation of a robot formation, and then present the role reassignment procedure.

3.1 Formation Representation

The formation presentation includes the formation shape in geometry pattern that the robot team needs to maintain and the reference type that robots keep their positions in a formation.

Formation shape can be represented as a directed acyclic graph [17]. Each robot is viewed as a vertex in the graph, and the relation between two robots as an edge. Each robot is given an identification number, called Robot ID (RID), and marked with a subscript of RID, such as R_1, R_2 , etc. The RID determines the position of a robot in a formation. Forming a formation by using the leader-following method, the RID of the leader robot is 1, and the rest of robots are called the followers. A directional edge is a line from the predecessor robot to the successor robot, as shown in Fig. 4, and the edge is described as

$$E(c, sucRID, preRID, l^d, \phi^d)$$

When the condition c is true, a successor robot with RID $sucRID$ follows its predecessor robot with RID $preRID$, and keeps a distance of l^d and an angle of ϕ^d . The distance l^d is the desired distance between two robots, and the angle ϕ^d is the desired angle between the positive x axis and the line l^d .

A formation shape is formed by a set of edges. Fig. 5 depicts some scalable formation shapes, including the shapes of column, line, wedge, and double-platoon. In these figures, the Arabic numerals are the RIDs of robots, and the thin arrows indicate the direction of robots. For a column formation as shown in Fig. 5(a), each robot except the leader robot uses the following edge relations to match its formation position in a column formation:

$$F = \{ E(i > 1, i, i-1, l^d, 270^\circ) \} \quad (4)$$

where i is the RID of a robot. Assume that the leader robot R_1 , where $i=1$, does not need to match the edge relation. When $i > 1$, each robot will follow its predecessor, $RID=i-1$, with a desired distance l^d and an angle of 270° . In these relations, R_2 is the predecessor of R_3 , and R_3 is the successor of R_2 . Other formation shapes in Fig. 5 can also be represented by a set of edge relations as follows:

$$F = \{ \begin{array}{l} E(i = 2, 2, 1, l^d, 180^\circ) \\ E(i > 2, i, i - 2, l^d, 180^\circ + 180^\circ(i \bmod 2)) \end{array} \} \quad (5)$$

$$F = \{ \begin{array}{l} E(i = 2, 2, 1, l^d, 225^\circ) \\ E(i > 2, i, i - 2, l^d, 225^\circ + 90^\circ(i \bmod 2)) \end{array} \} \quad (6)$$

$$F = \{ \begin{array}{l} E(i = 2, i, i - 1, l^d, 0^\circ) \\ E(i > 2, i, i - 2, l^d, 270^\circ) \end{array} \} \quad (7)$$

Equations (5), (6), and (7) represent the edge relations of formation shapes in Figs. 5(b), 5(c), and 5(d), respectively [18].

Reference type is an important method for a group of robots to keep a formation [2,6]. For a double-platoon, three reference types are described in Fig. 6, namely predecessor, leader, and neighbor. In predecessor reference type, the desired position of each robot in the formation is calculated with respect to its own predecessor. From the edge relation of the double-platoon formation in Eq. (7), each robot can directly extract the geometric relation relative to its predecessor, such as l^d and ϕ^d , and use this relation to maintain the formation with its predecessor. Similarly, in the leader reference type, each robot calculates its desired position relative to the leader robot, as shown in Fig. 6(b). However, in the formation shapes described in Eqs. (4)-(7), the edges show only the relations between the predecessor and successor robots. Therefore, in the leader reference type, the desired position for each robot relative to the leader robot must be calculated and utilized to keep the desired formation with the leader robot. Similar consideration is made for the neighbor reference type in Fig. 6(c). Furthermore, the combination of two or three different reference types will form more hybrid reference types, for examples, leader-predecessor, leader-neighbor, predecessor-neighbor and leader-predecessor-neighbor reference types [6].

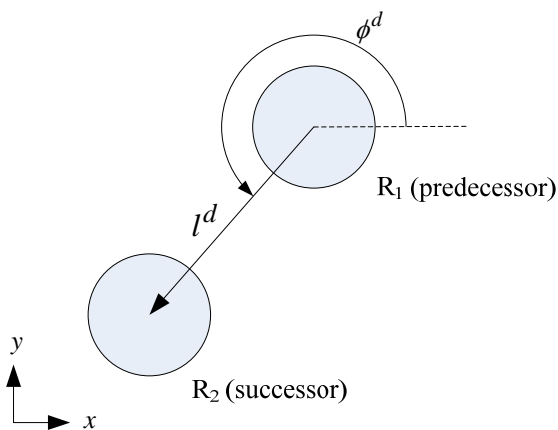


Fig. 4 Formation Description

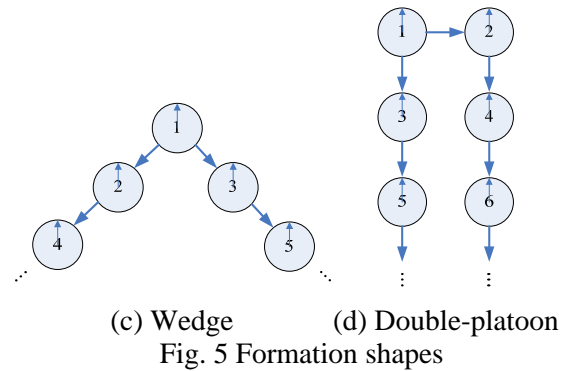
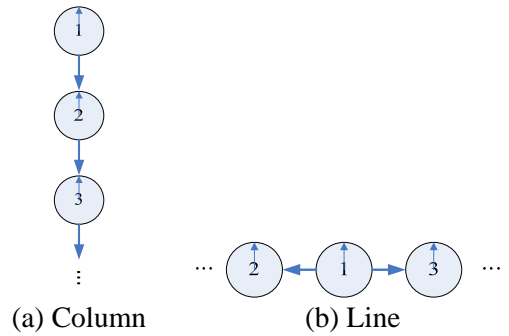


Fig. 5 Formation shapes

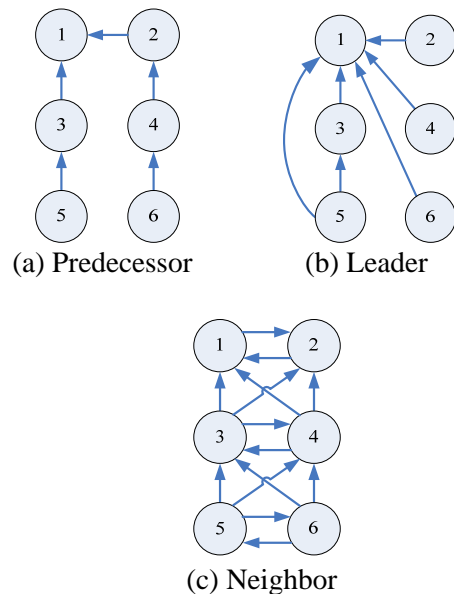


Fig. 6 Reference types of formation

3.2 Role Reassignment Procedure

The role-reassignment procedure is divided into three stages. First, the degree of difficulty for a robot been assigned a specified role in a formation is defined as the character cost. The cost function is calculated based on the spatial relationship and the information of obstacles surrounding the robots. Second, define the character set fitness of a robot formation by determining the character cost of each robot. Finally, select the best character set of a formation by calculating the character set fitness.

Consider the situation in a leader reference type, for example, when the robot team forms the formation from a tangled initial condition as shown in Fig. 7(a). In the figure, the circle indicates the robot and the number is the robot ID; the solid arrow is the heading direction of the robot and the dotted arrow is the desired path of the leader. Define the character cost to represent the degree of difficulty for a robot to be assigned a specified role in a formation,

$$C_{i,j}^l = K_x d_{x,i,j} + K_y d_{y,i,j} + K_\phi d_{\phi,i,j} + A_{i,j} \quad (8)$$

where $C_{i,j}^l$ is the cost of the robot with number i to be assigned the targeted role with $RID=j$ when the leader robot has the robot number l . $d_{x,i,j}$, $d_{y,i,j}$, and $d_{\phi,i,j}$ are the position and orientation from the robot's initial position to the target position, respectively. K_x , K_y and K_ϕ are the weight constants and $A_{i,j}$ is the additional cost caused by the action of avoiding an obstacle. Furthermore, define a character set matrix to indicate the assigned role in each character set,

$$S = [N_1 \ N_2 \ N_3 \ \dots \ N_n] \quad (9)$$

where $N_1, \dots, N_n \in N^0$ represent robot numbers; n is the total number of robots. The matrix index in Eq. (9) indicates the assigned role for a robot. For example, $S = [3 \ 1 \ 2]$ indicates that robot 3 is assigned the role with $RID=1$, robot 1 is assigned with $RID=2$, and robot 2 is assigned with $RID=3$. Calculate the cost of each set according to Eq. (8), and then use the following equation to determine the fitness of a character set, f :

$$f = \left[\sum_{i=1}^n C_i + 1 \right]^{-1}$$

where C_i is the cost of a robot to become a role in the formation with $RID=i$. In the paper, we define that the formation with the largest character set fitness is the best choice for a new role assignment.

Consider the situation of three robots in a formation and the leader role assignment as shown in Fig. 7(a). Calculate the cost for each robot to move from the initial position to the desired position, t_1 , and act as the leader. Assume that the leader needs not to deliberate the cost caused by the action of avoiding an obstacle and then $A=0$ in Eq. (8). The cost functions for all robots are shown as follows:

$$C_{1,1}^1 = K_x d_{x,1,1} + K_y d_{y,1,1} + K_\phi d_{\phi,1,1}$$

$$C_{2,1}^2 = K_x d_{x,2,1} + K_y d_{y,2,1} + K_\phi d_{\phi,2,1}$$

$$C_{3,1}^3 = K_x d_{x,3,1} + K_y d_{y,3,1} + K_\phi d_{\phi,3,1}$$

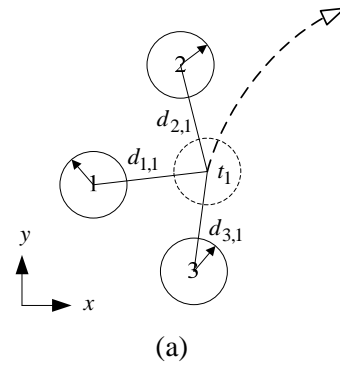
The degree of difficulty can be judged by the values of $C_{1,1}^1$, $C_{2,1}^2$ and $C_{3,1}^3$. If robot 1 is assigned to be

the leader, $RID=1$, the cost of the other robots playing other roles can be determined by moving the robots to the positions t_2 and t_3 in order to keep in a formation as shown in Fig. 7(b).

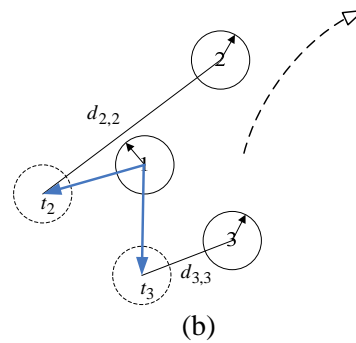
$$C_{2,2}^1 = K_x d_{x,2,2} + K_y d_{y,2,2} + K_\phi d_{\phi,2,2} + A_{2,2}$$

$$C_{3,3}^1 = K_x d_{x,3,3} + K_y d_{y,3,3} + K_\phi d_{\phi,3,3} + A_{3,3}$$

Two sets of cost are calculated, including $\{C_{1,1}^1, C_{2,2}^1, C_{3,3}^1\}$ and $\{C_{1,1}^1, C_{3,2}^1, C_{2,3}^1\}$.



(a)



(b)

Fig. 7 Calculating (a) the cost of the leader; (b) the cost of other roles.

3.3 Cost of Obstacle Avoidance

The character cost function in Eq. (8) includes the term $A_{i,j}$ caused by the behaviour of avoiding an obstacle. Define the character cost

$$A_{i,j} = CT \times G \times SD \quad (10)$$

where CT is the number of intersection of two robot paths; SD is the safe distance for a robot to avoid the obstacle; G is a weighting constant. In the paper, we utilize the method of determining whether two line segments intersect each other to estimate the number of intersection, CT . The basic concept is to investigate the direction of nearby vertices of line segments and determine whether these two line segments are intersecting each other. For two line segments in a plane as shown in Fig. 9, there are two situations involving in line-intersecting,

including 9(b) and (c). The direction of nearby vertices is defined as Fig. 8. In the figure, m_1 , m_2 and m'_2 are the slopes of line segments AB, AC and AC', respectively. Since $m_1 > m_2$, the direction from vertex A, through vertex B, to vertex C is defined as counter-clockwise and indicated as the value 1. Furthermore, $m'_2 < m_1$, the direction from vertex A, through vertex B, to vertex C' is defined as clockwise and indicated as the value -1. The procedures for determining the intersection of two line segments in Fig. 9 are marshalled as follows:

- a) Choose one line segment and one vertex of the other line segment to determine the direction of vertices. In Fig. 9(b) for example, the direction of vertices from vertex A, through vertex B, to vertex C is indicated as *Dir1*.
- b) Similarly, determine the direction of vertices from vertex A, through vertex B, to vertex D and indicate as *Dir2*.
- c) Choose the other line segment and repeat steps a) and b). Determine the direction of vertices from vertex C, through vertex D, to vertex A as *Dir3*, and the direction of vertices from vertex C, through vertex D, to vertex B as *Dir4*.
- d) Two line segments are intersecting each other if $Dir1 \cdot Dir2 < 0$ and $Dir3 \cdot Dir4 < 0$.

In Fig. 9(b), the directions of vertices are determined as follows: $Dir1 = -1$, $Dir2 = 1$, $Dir3 = 1$ and $Dir4 = -1$. Therefore, the line segments AB and CD intersect each other, because the condition $Dir1 \cdot Dir2 < 0$ and $Dir3 \cdot Dir4 < 0$ are satisfied. By the same concept, in Fig. 9(a) the line segments AB and CD do not intersect each other, since the values of $Dir1$, $Dir2$ and $Dir3$ are 1, and the value of $Dir4$ is -1, and the condition $Dir1 \cdot Dir2 < 0$ is not satisfied.

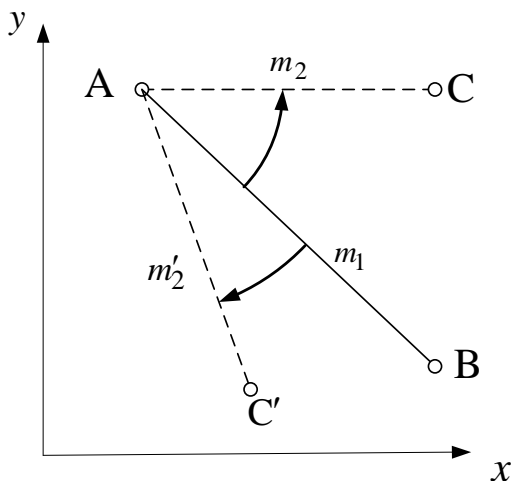


Fig. 8 Direction of three vertexes nearby

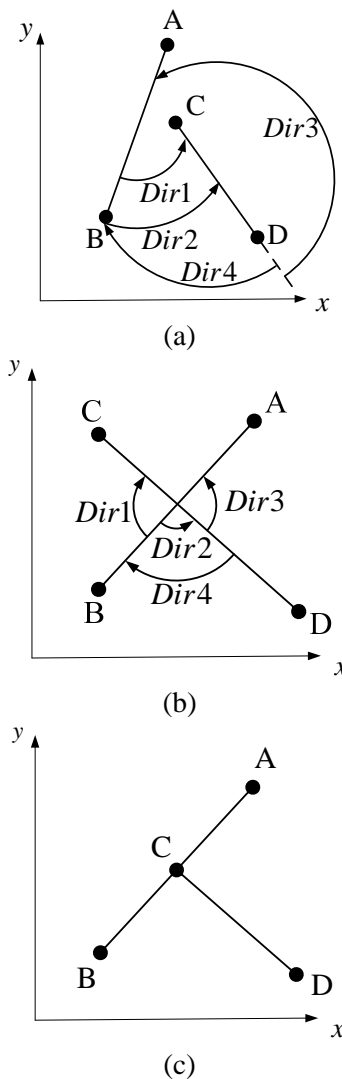


Fig. 9 Intersection of two plane segments

4 Simulation and Experimental Results

A group of omni-directional driven robots are designed and fabricated to demonstrate the proposed obstacle avoidance and dynamic role assignment algorithms for robot formation control. The robot is 24cm in height and 25cm in diameter, 2kg in weight, including two 12V/2.5AH batteries. One battery provides the power for three 12V DC motors and the other provides for the 8085 motor drive system. The control system comprises three subsystems, including a vision sensor system mounted on the ceiling, a PC-based control system for image processing and remote control, and the 8051 motor drive system on the robot. In this research, we develop the PC-based control system by utilizing the Microsoft Visual C++ with DirectX. The camera system is the only sensor to provide the robot position information in the environments. The robot drive system communicates with the PC-based controller by using a radio frequency (RF) device.

The performance of a robot motion control affects the convergence time and the trajectory error of the formation control directly. Hsu and Liu [6] treated a robot as a point mass and use a simple geometry equation to design a motion controller. The trajectory error of a robot will increase if the kinematic constraints are ignored. Many researchers considered the nonholonomic kinematic constraints of wheeled robots in formation control [4,19,20,21]. The paper deals with the dynamic control of an omni-directional driven robot. We assume that an omni-directional driven robot moves on a plane. Define the global coordinates xy and the robot body-fixed coordinates $x_m y_m$ as Fig. 10. The appearance of the robot is a cylinder, and three omni-directional wheels are fixed on the bottom of robot and separated by 120° . Define the right-front wheel as Wheel 1, the left-front wheel as Wheel 2 and the hind wheel as Wheel 3. The position vector of the robot is:

$$p = [x \quad y \quad \phi]^T$$

where x and y are the plane coordinates of the robot, ϕ is the angle between the global coordinate and the body-fixed coordinate. Define ϕ is positive when the rotation of robot is counterclockwise, otherwise, it is negative. The velocity vector of robot is:

$$\dot{p} = [\dot{x} \quad \dot{y} \quad \dot{\phi}]^T$$

Then the dynamic equation of the robot can be expressed as follows:

$$f_x = M\ddot{x} = -\sin(30^\circ + \phi)D_1 - \sin(30^\circ - \phi)D_2 + \cos\phi D_3 \quad (11)$$

$$f_y = M\ddot{y} = \cos(30^\circ + \phi)D_1 - \cos(30^\circ - \phi)D_2 + \sin\phi D_3 \quad (12)$$

$$M_I = I\ddot{\phi} = LD_1 + LD_2 + LD_3 \quad (13)$$

f_x , f_y and M_I are the forces and moment applied on the mass center of the robot, respectively; M and I are the mass and moment of inertia of the robot, respectively; D_i are the driving forces of each wheel; L is the radius of the robot. The relationship between the wheel speed and the velocities in global coordinates is expressed as:

$$\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} = \frac{1}{r} \begin{bmatrix} -\sin(30^\circ + \phi) & \cos(30^\circ + \phi) & L \\ -\sin(30^\circ - \phi) & -\cos(30^\circ - \phi) & L \\ \cos\phi & \sin\phi & L \end{bmatrix} \dot{p} \quad (14)$$

r is the radius of the wheel, ω_i , $i=1, \dots, 3$ are rotational speed of each wheel. The dynamics of the wheel is modeled by the following equation:

$$I_w \dot{\omega}_i = \tau_i - rD_i - c\omega_i \quad (15)$$

where c is the coefficient of viscous friction of the wheel shaft, I_w is the moment of inertia of the wheel

around the driving shaft, and τ_i , $i=1, \dots, 3$, are torques induced by driving motors. Substitute Eqs. (14)-(15) into Eqs. (11)-(13), and obtain the dynamic equation of the robot in terms of applied torques,

$$M\ddot{p} + C(p, \dot{p})\dot{p} = H\tau \quad (16)$$

where the matrices are defined as

$$M = \begin{bmatrix} 2Mr^2 + 3I_w & 0 & 0 \\ 0 & 2Mr^2 + 3I_w & 0 \\ 0 & 0 & I_v r^2 + 3I_w L^2 \end{bmatrix}$$

$$C = \begin{bmatrix} 3c & 3I_w \dot{\phi} & 0 \\ -3I_w \dot{\phi} & 3c & 0 \\ 0 & 0 & 3L^2 c \end{bmatrix}$$

$$H = \begin{bmatrix} -r(\cos\phi + \sqrt{3}\sin\phi) & -r(\cos\phi - \sqrt{3}\sin\phi) & 2r\cos\phi \\ r(\sqrt{3}\cos\phi - \sin\phi) & -r(\sqrt{3}\cos\phi + \sin\phi) & 2r\sin\phi \\ Lr & Lr & Lr \end{bmatrix}$$

A motion controller can be designed for the robot [22,23]. A standard computed torque control is utilized in this paper for the motion control of omni-directional driven robots,

$$H\tau = M(\ddot{p}_d + K_v \dot{e} + K_p e) + C(p, \dot{p})\dot{p}$$

P_d is the desired position; e is the vector of position error; K_v and K_p are gain matrices of the controller. The robot controller is designed based on the kinematic constraints and dynamic equation of the robot to ensure the dynamic performance of each robot in formation control.

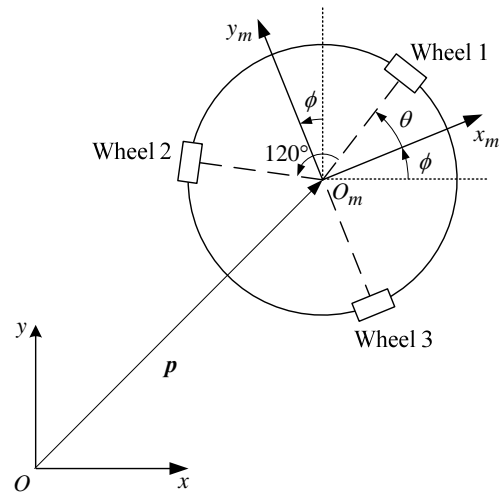


Fig. 10 Coordinates of the robot system

4.1 Simulation Results of Role Reassignment

We perform the simulation by using MATLAB. In this simulation example, the initial positions of the robots are $(151, 132, 0^\circ)$, $(267, 57, 0^\circ)$, and $(120, 224, 0^\circ)$, respectively. The leader robot is scheduled to follow the two-section path as shown in Fig. 11.

The edge relations of the desired formation shapes in section 1 and section 2 are expressed as

$$\text{Section 1: } F = \{ \begin{array}{l} E(i = 2, 2, 1, l^d, 120^\circ) \\ E(i = 3, 3, 1, l^d, 180^\circ) \end{array} \}$$

$$\text{Section 2: } F = \{ \begin{array}{l} E(i = 2, 2, 1, l^d, 0^\circ) \\ E(i = 3, 3, 1, l^d, 60^\circ) \end{array} \}$$

where $l^d = 144$. These two distinct formation shapes are shown in Fig. 12(a)-(b). In the figures, three robots have the same formation shape, but the leader (dark circle) is scheduled to switch the edge angles when path section is changed. The simulation results are plotted in Figs. 13 and 14 for methods of a fixed role assignment and the proposed role reassignment, respectively. The sampling time is 0.03 second and total simulation time is 30 seconds (999 steps). In Fig. 13, with fixed role assignment, robots 2 and 3 follow a longer distance and line-intersecting path in order to maintain a fixed character set $S=[1 \ 2 \ 3]$. In Fig. 14, by using the role-reassignment procedure, the character set in second path section is switch from $S=[1 \ 2 \ 3]$ to $S=[1 \ 3 \ 2]$ for robots 2 and 3 to prevent a longer travelling distance and line-intersecting path.

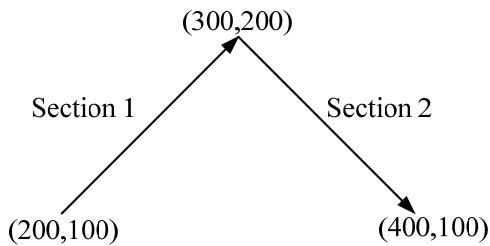


Fig. 11 Desired path of the leader robot

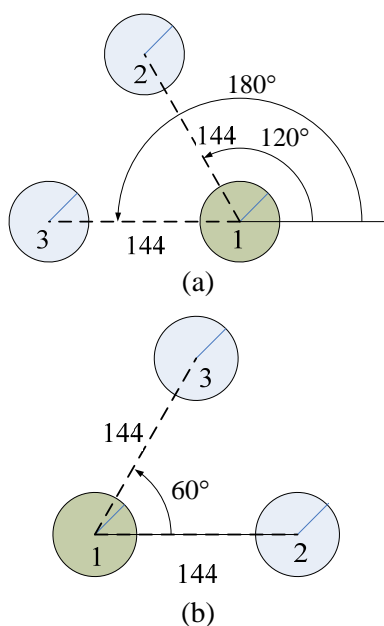


Fig. 12 Desired robot formation (a) Section 1, (b) Section 2

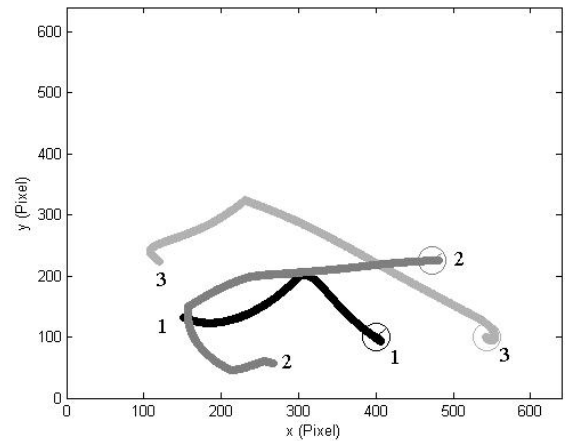


Fig. 13 Simulation results for a fixed role assignment algorithm

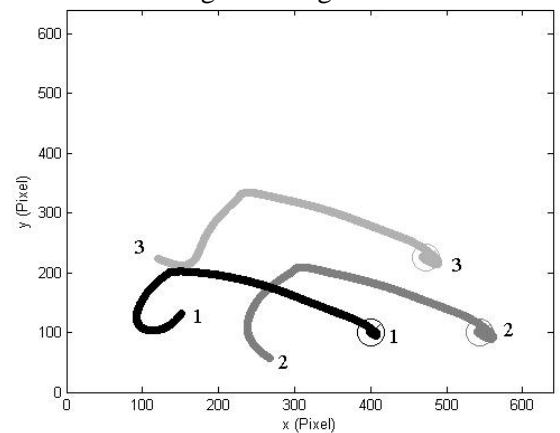


Fig. 14 Simulation results for the proposed dynamic role assignment algorithm

4.2 Experimental Results of Obstacle Avoidance

The proposed algorithm for obstacle avoidance is tested on a RoboCup small-sized robot soccer field shown in Fig. 15 and the data is plotted in Fig. 16. The initial and desired positions of robot are located at (463, 41) and (459, 423). The parameters, aed and SD , are both 111.25 and old is 133.5. The result shows that the algorithm for obstacle avoidance is feasible.



Fig. 15 Trajectory of one obstacle avoidance

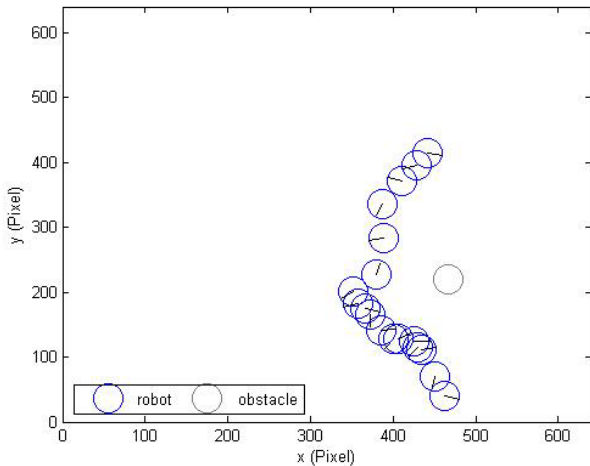


Fig. 16 Data plot of one obstacle avoidance

4.3 Experimental Results of Role Assignment

The experiment is performed on a RoboCup small-sized robot soccer field. The leader robot is scheduled to follow a two-section path as shown in Fig. 11. The edge relations of the desired formation shapes in section 1 and section 2 are planned as shown in Fig. 12. All other conditions are designed as the same as the simulation case. The results are presented in Figs. 17 and 18 for methods of a fixed role assignment and the proposed role-reassignment procedure, respectively. Time interval between every two pictures is 3 seconds, and the results are similar to those by simulation. In Figs.17(a)-(d), with fixed role assignment $S=[1\ 2\ 3]$, all robots have to switch positions when path section is changed. In Figs.18(a)-(d), by using the role-reassignment procedure, the character set is switch from $S=[1\ 2\ 3]$ to $S=[1\ 3\ 2]$ for robots 2 and 3 to prevent a longer travelling distance and line-intersecting path. Therefore, the role-reassignment procedure is more effective than the fixed one.

5 Conclusion

We developed an algorithm for obstacle avoidance and a procedure for role reassignment of multiple-robot control during a team formation is forming or switching. The proposed algorithm for obstacle avoidance is based on the edge-detection method which can determine the obstacles in the environments. The procedure for role reassignment assigns a new role for each robot by calculating the cost function for a formation, and taking account of the travel distance for all robots and the possibility for a robot to collide with obstacle. The proposed algorithm and procedure are validated through a theoretical analysis. In particular, the dynamic

model of the robot is considered which will improve the response and the trajectory error of robots in a formation. Simulation and experiment are also performed with real platform to verify the proposed algorithm.

The proposed procedure for role reassignment is suitable for a team formation with small number of robots. For robot team with large number of robots, the concept of multi-team formation control proposed by Hsu and Liu [6] can be utilized. Each robot team is equipped a role-reassignment procedure developed in this research, and multiple teams are integrated to perform formation control.

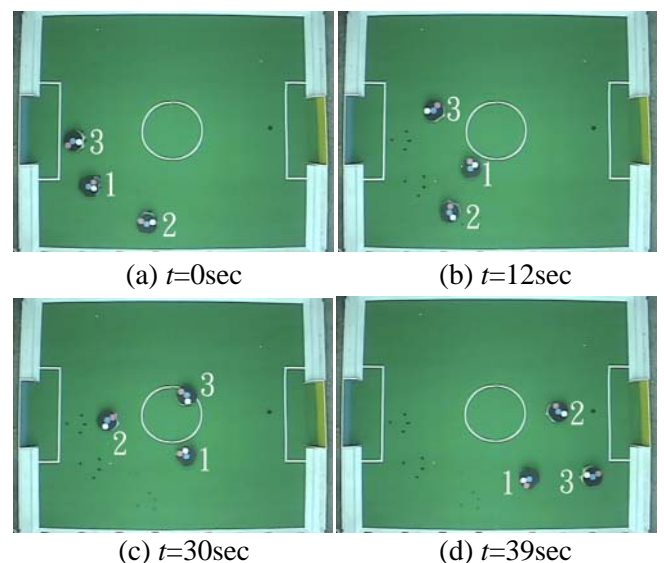


Fig. 17 Experimental results for a fixed role assignment algorithm

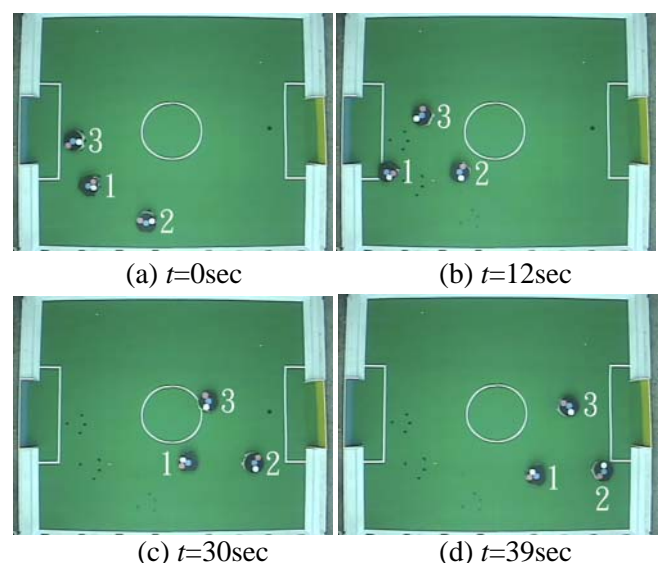


Fig. 18 Experimental results for the proposed dynamic role assignment algorithm

References:

- [1] R.C. Arkin, *Behavior-based Robotics*, MIT Press, 1998.
- [2] T. Blach and R.C. Arkin, Behavior-based formation control for multirobot teams, *IEEE Transactions on Robotics and Automation*, vol. 14, no. 6, pp. 926-939, 1998.
- [3] F.E. Schneider, D. Wildermuth, and H.-L. Wolf, Motion coordination in formations of multiple mobile robots using a potential field approach, in *Distributed Autonomous Robotic Systems*, L. E. Parker, G. Bekey, and J. Barhen, Eds., Springer-Verlag, vol.4, pp.305-314, 2000.
- [4] A.K. Das, P. Fierro, V. Kumar, J.P. Ostrowski, J. Spletzer, and C. J. Taylor, A vision-based formation control framework, *IEEE Transactions on Robotics and Automation*, vol.18, no.5, pp.813-825, 2002.
- [5] R. Fierro, P. Song, A.K. Das, and V. Kumar, Cooperative control of robot formations, in *Cooperative Control and Optimization*, R. Murphey and P. Pardalos, Eds. Dordrecht, The Netherlands: Kluwer, ch.5, pp.73-93, 2002.
- [6] H.C.-H. Hsu and A. Liu, Multiagent based multi-team formation control for mobile robots, *Journal of Intelligent and Robotic Systems*, vol.42, no.4, pp.337-360, 2005.
- [7] J.S. Jennings, G. Whelan, and W.F. Evans, Cooperative search and rescue with a team of mobile robots, in *Proceedings of the IEEE International Conference on Advanced Robotics*, pp.193-200, 1997.
- [8] D. Fox, W. Burgard, H. Kruppa, and S. Thrun, A probabilistic approach to collaborative multi-robot localization, *Autonomous Robots*, vol.8, no.3, pp.325-344, 2000.
- [9] M. Mataric, M. Nilsson, and K. Simsarian, Cooperative multi-robot box pushing, in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp.556-561, 1995.
- [10] P. Stone and M. Veloso, Task decomposition, dynamic role assignment, and low-bandwidth communication for real-time strategic teamwork, *Artificial Intelligence*, vol.110, no.2, pp.241-273, 1999.
- [11] Y.T. Wang, Z.J. You, and C.H. Chen, AIN-based Action Selection Mechanism for Soccer Robot Systems, *Journal of Control Science and Engineering*, vol. 2009, Article ID 896310, 2009.
- [12] H.C.-H. Hsu and A. Liu, Kinematic design for platoon-lane-change maneuvers, *IEEE Transactions on Intelligent Transportation Systems*, vol.9, no.1, pp.185-190, 2008.
- [13] J. Borenstein and Y. Koren, Real-time Obstacle Avoidance for Fast Mobile Robots, *IEEE Transactions on Systems, Man, and Cybernetics*, vol.19, no.5, pp.1179-1187, 1989.
- [14] J. Borenstein and Y. Koren, The Vector Field Histogram - Fast Obstacle Avoidance for Mobile Robots, *IEEE Journal of Robotics and Automation*, vol.7, no.3, pp.278-288, 1991.
- [15] T. Dierks and S. Jagannathan, Neural Network Control of Mobile Robot Formations Using RISE Feedback, *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol.39, no.2, pp.332-347, 2009.
- [16] J.H. Reif, and H. Wang, Social potential fields: A distributed behavioral control for autonomous robots, *Robotics and Autonomous Systems*, vol.27, no.3, pp.171-194, 1999.
- [17] J.P. Desai, A graph theoretic approach for modeling mobile robot team formations, *Journal of Robot Systems*, vol.19, no.11, pp.511-525, 2002.
- [18] Y.C. Chen and Y.T. Wang, Dynamic Role Assignment Algorithm for Robot Formation Control, *Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, 2007.
- [19] R.M. Murray and S. Sastry, Nonholonomic Motion Planning: Steering Using Sinusoids. *IEEE Transactions on Automatic Control*, vol.38, no.5, pp.700-716, 1993.
- [20] M.B. McMickell and B. Goodwine, Motion Planning for Nonlinear Symmetric Distributed Robotic Formations, *International Journal of Robotics Research*, Vol.26, No.10, pp.1025-1041, 2007.
- [21] L. Song, Y. Fei, and T. Lu, Study on effects of nonholonomic constraints on dynamics of a new developed quadruped leg-wheeled passive mobile robot, *WSEAS Transactions on Systems*, vol.8, no.1, pp.137-147, 2009.
- [22] S. Blazic, E. Guechi, J. Lauber, M. Dambrine, and G. Klancar, The problems of camera measurements in tracking-error fuzzy control of mobile robots, *WSEAS Transactions on Systems*, vol.8, no.4, pp.441-450, 2009.
- [23] J. Jing, Y. Ye, and Y. Fang, Optimal Sliding-mode Control Scheme for the Position Tracking Servo System, *WSEAS Transactions on Systems*, vol.7, no.5, pp.435-444, 2008.