Proceedings of the 36th Chinese Control Conference
July 26-28, 2017, Dalian, China

Distributed Encirclement Control with Arbitrary Spacing for Multiple Anonymous Mobile Robots

Weijia Yao, Zhiwen Zeng, Xiangke Wang, Huimin Lu*, Zhiqiang Zheng
College of Mechatronics and Automation, National University of Defense Technology, Changsha 410073, P. R. China
E-mail: lhmnew@nudt.edu.cn

Abstract: Encirclement control enables a multi-robot system to rotate around a target while they still preserve a circle formation, which is useful in real world applications such as entrapping a hostile target. In this paper, a distributed control law is proposed for any number of anonymous and oblivious robots in random three dimensional positions to form a specified circular formation with any desired inter-robot angular distances (i.e. spacing) and encircle around the target. Arbitrary spacing is useful for a system composed of heterogeneous robots which, for example, possess different kinematics capabilities, since the spacing can be designed manually for any specific purpose. The robots are modelled by single-integrator models, and they can only sense the angular positions of their two neighbouring robots, so the control law is distributed. Theoretical analysis and simulation results are provided to prove the stability and effectiveness of the proposed control strategy.

Key Words: encirclement, distributed control, multi-robot system, arbitrary spacing, heterogeneous robots

1 Introduction

More and more research focuses on multi-robot systems since there are many advantages over single-robot systems [1]. In a multi-robot system, for example, there are distributed sensors and end-effectors, leading to parallel sensing and control. It has the desirable property of more redundancy, higher robustness and greater fault tolerance. The ways of controlling a multi-robot system can be categorized into two classes: centralized control and distributed control. As for centralized control, a central robot is responsible for almost all the work of sensing, computation and decision-making, and then a centralized control signal is transmitted to other robots in the system, so an effective coordination of the robots can be achieved. However, the prerequisite for this system is that there is at least one robot (the central robot) that is able to communicate with all the other robots [2], which is not always possible in many real scenarios. In contrast, distributed control employs local information among robots to realize collective behaviour and achieve a global objective [1]. Considering the narrow communication bandwidth, limited computation/memory resources and the mass scale of the group to control, distributed control possesses greater potential for applications. Currently, it has a wide range of applications in military, aerospace, industry and educational activities.

One of the most prominent research topics on multi-robot system is the formation control problem. The objective is for a group of robots to form a desired formation by local interactions among them, so a particular global aim can be achieved. Generally, formation control problems fall into two categories: formation producing and formation tracking, depending on whether a formation reference exists or not [3]. An early literature on formation control [4] was published in 1996. In this study, it was shown that simple interactions between robots could lead to line or circle formations of robots with particular considerations on their physical constraints, such as sensing ranges and physical dimensions. However, convergence and stability analysis are not provided. In this line of research, significant efforts have been made on the circle formation control and encirclement control problem. The terms, circle formation control and encirclement control, are similar because of the tight connection to the geometric shapes of the formation. In circle formation control [5], robots remain in their positions after the formation is generated, while in encirclement control [6, 7], they still encircle around the target. In this sense, circle formation control could be regarded as a special case of encirclement control when the encirclement speed equals to zero. The control laws and methods, however, are different for these two kinds of problems. Recently, encirclement control has attracted more and more attention since it has many promising applications albeit being more complex.

There are already many studies on encirclement control. The study presented in [8], inspired by the problem of n-bug cyclic pursuit, considers the second order dynamics model and realizes the encirclement control for multiple robots. It also presents detailed analysis of the equilibrium and stability properties. Based on the previous study, [9] presents a decentralized strategy to achieve symmetric formations using an extended cyclic pursuit method, i.e. each agent pursuing its leading neighbour along the line of sight rotated by a common offset angle, the different value of which leads to different formations. The authors of [7] model the foraging behaviour of dolphins as a formation control problem with multiple constraints. Then a control law is effectively designed with the introduction of an orthogonal projection operator [10]. Its convergence and stability analysis is carried out based on the bearing rigidity graph theory [10] and the cyclic-small-gain theorem [11]. However, the problem of obstacle avoidance is not taken into account. The study presented in [6] provides a relatively comprehensive solution for distributed circumnavigation control of robots around a 2D or 3D target. It not only considers three cases of the encirclement speed, but also addresses the problem of collision avoidance. Furthermore, consensus theory is applied to obtain some global variables of interest. Ref. [5] proposes a distributed control law to form circle formations with any desired angular distances among robots. However, it assumes that the robots are placed initially on a prescribed circle, which is not always desirable in practice.

The goal of this paper is to propose a distributed encirclement control law which is able to control a group of mobile robots from any initial positions to encircle around a target with any desired inter-robot angular distance in the three dimensional space. We suppose that these mobile robots are heterogeneous in terms of their kinematics abilities, such as moving speeds, maximum accelerations, etc. In a scenario where these mobile robots need to entrap a hostile target, their inter-robot angular distances should be different for better performance; those robots with lower mobility are supposed to gather together with smaller inter-robot distances than those with higher mobility, so the probability for the target to flee away from the circle can be lower. However, we do not consider the particular variances in mobility or the optimal spacing among robots. Instead, a desired spacing is assumed to be appropriate and specified beforehand. So this study focuses more on the distributed encirclement control laws. The main contribution of this paper is that the control law proposed here does not require any requirements on the initial positions or the number of robots, and it can lead to any desired spacing in a circular formation in the 3D space.

The remainder of this paper is organized as follows. Section II introduces the encirclement problem with arbitrary spacing and derives its corresponding mathematical formulation. Section III proposes the encirclement control law that solves the encirclement problem. Furthermore, in section IV, simulation experiments are performed and results are analyzed. Finally, section V concludes the paper and summarizes the future work.

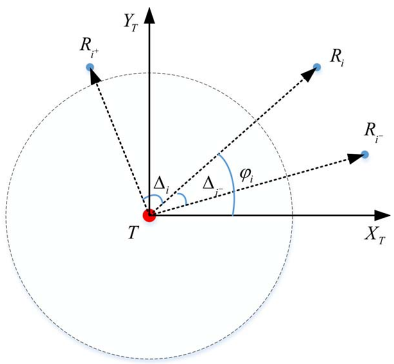

Fig. 1: The illustration of the multi-robot encirclement control problem

1.1 Problem Formulation

The research question is that a group of n $(n \geq 2)$ mobile robots, denoted by $r_i, \ i = 1, ..., n$, encircle a target in 3D space with any predefined spacing on a circle formation. Suppose each mobile robot is modelled by a 3D kinematic point

$$\dot{p}_i(t) = u_i(t), \ \ i = 1, ..., n \tag{1}$$

where $u_i(t)$ is the control input to the robot r_i and $p_i(t) \in \mathrm{R}^3$ is its position in the world (global) reference frame *W*. In this problem, robots are required to maintain on the same plane, which contains the target modelled by another kinematic point. These robots keep a specified distance to the target point when they rotate around it. Therefore, a body reference frame *B* centred at the target point T is introduced (see Fig. 1). In addition, a representation in cylindrical coordinates is preferred to the commonly used one in Cartesian coordinates since the former itself embodies three elements of interest: the distance between the projection of the robot on the $X_T - Y_T$ plane to the target (ρ), the height relative to the $X_T - Y_T$ plane (z) and the angle between theX_T-axis and the line joining the projection of the robot on the $X_T - Y_T$ plane with the target $T(\varphi)$. The cylindrical coordinate for robot r_i is denoted by $q_i = (\rho_i, \varphi_i, z_i)^T$. To relate the cylindrical coordinates with the Cartesian coordinates, a vector function is defined as:

$$q(v) = (\rho(v), \varphi(v), z(v))^T \tag{2}$$

where $v \in \mathrm{R}^3$ is a generic vector with components v_x, v_y, v_z and

$$\rho(v) = \sqrt{v_x^2 + v_y^2} \tag{3}$$

$$\varphi(v) = \tan^{-1}(v_y/v_x) \tag{4}$$

$$z(v) = v_z \tag{5}$$

Note that $\varphi \in [0, 2\pi)$. The Jacobian matrix of formula (2) will be used later, which is,

$$J = \frac{\partial q}{\partial v^T} = \begin{bmatrix} \frac{v_x}{\sqrt{v_x^2+v_y^2}} & \frac{v_y}{\sqrt{v_x^2+v_y^2}} & 0 \\ \frac{-v_y}{v_x^2+v_y^2} & \frac{v_x}{v_x^2+v_y^2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{6}$$

For better analysis, we label the robots counterclockwise according to their initial (angular) positions in the target body reference frame *B* as shown in Fig. 1 . Note that the subscript i^- and i^+ represent the robot indices before and after the index i respectively. Especially, if $i = n$, $i^+ = 1$, while if $i = 1$, $i^- = n$. Δ_i represents the difference between the angular positions of R_{i^+} and that of R_i. In particular,

$$\Delta_i = \begin{cases} \varphi_{i^+} - \varphi_i, & i = 1, ..., n-1 \\ \varphi_1 - \varphi_n + 2\pi, & i = n \end{cases} \tag{7}$$

In addition, we define $d_i > 0$ as the desired difference in angular positions (i.e. spacing) between the robot R_{i^+} and R_i. Also note that

$$\sum_{i=1}^{n} \Delta_i = \sum_{i=1}^{n} d_i = 2\pi \tag{8}$$

Therefore, the research question can be formulated as follows:

Definition (Encirclement Problem with Arbitrary Spacing)**.** *The encirclement problem of n $(n \geq 2)$ mobile robots, of which the dynamics are modelled by formula (1), is to seek control laws satisfying the following asymptotic conditions:*

$$\lim_{t \to \infty} \rho_i(t) = \rho^* \tag{9}$$

$$\lim_{t \to \infty} \Delta_i(t) = d_i \tag{10}$$

$$\lim_{t \to \infty} \dot{\varphi}_i(t) = \omega^* \tag{11}$$

$$\lim_{t \to \infty} z_i(t) = 0 \tag{12}$$

for all $i = 1, ..., n$. Here, $\rho^* > 0$, $\omega^* > 0$ and $d_i > 0$ denote the encirclement radius, angular speed and desired spacing among robots respectively, and $\sum_{i=1}^{n} d_i = 2\pi$.

2 Encirclement Control

To consider the encirclement problem in the body reference frame *B* so that we could take advantage of the cylindrical coordinates, first we define a rotational matrix R_b, which is the representation of the body reference frame *B* with respective to the world reference frame *W*. Therefore, the following formula calculates the cylindrical coordinates of the robot r_i in frame *B*.

$$q_i = q(R_b^T(p_i - p_b)) \tag{13}$$

where p_i and p_b are the Cartesian coordinates of the robot r_i and the target in frame *W* respectively. Then the derivative of formula (13) is the dynamics of robots in the cylindrical coordinates, which is,

$$\dot{q}_i = J_i[\dot{R}_b^T(p_i - p_b) + R_b^T(\dot{p}_i - \dot{p}_b)] \tag{14}$$

where J_i is the Jacobian matrix as shown in formula (6), i.e. $J_i = \frac{\partial q}{\partial v^T}\Big|_{v = R_b^T(p_i - p_b)}$. Note that in view of formula (6), $\det(J) = \frac{1}{\sqrt{v_x^2+v_y^2}}$ as long as $v_x^2 + v_y^2 \neq 0$. In the scenario of the encirclement control problem, this means J_i is invertible as long as the distance between the robot r_i and the target is nonzero. This condition can always be guaranteed since the initial positions of the robots and the target does not coincide, and by designing appropriate control laws, the distance can be guaranteed to be nonzero all the time. By letting

$$u_i = \dot{p}_i = \dot{p}_b + R_b(J_i^{-1}v_i - \dot{R}_b^T(p_i - p_b)) \tag{15}$$

we can switch our focus to the new control input in the cylindrical coordinates $v_i = \dot{q}_i = (\dot{\rho}_i, \dot{\varphi}_i, \dot{z}_i)^T$ [6]. The advantage of transforming to this control input is that we can control ρ_i, φ_i and z_i separately, which are the three main variables in the encirclement problem.

Before providing the control law for the encirclement problem with arbitrary spacing, first we list some useful concepts from the matrix analysis theory and the graph theory. For a positive integer n, M_n is a set of all n-by-n real matrices. If all the entries in a matrix is nonnegative, then this matrix is called nonnegative. The directed graph (or digraph) of a nonnegative matrix $M \in M_n$, denoted by $G(M)$, is the directed graph with the vertex set $\{v_i\}, i \in \{1, ..., n\}$, such that there is a directed edge in $G(M)$ from v_j to v_i if and only if $m_{ij} \neq 0$ [12]. A directed graph is called strongly connected if for every pair of vertices there is a directed path between them [13]. We denote $I_d \in \mathrm{R}^{d\times d}$ as the identity matrix and $\mathbf{1} \triangleq [1, ..., 1]^T$, $\mathbf{0} \triangleq [0, ..., 0]^T$. The following is a preliminary result related to any strongly connected digraph.

Lemma (Theorem 3 of [14])**.** *Assume G is a strongly connected digraph with Laplacian L satisfying $Lw_r = 0$, $w_l^T L = 0$ and $w_l^T w_r = 1$. Then*

$$R = \lim_{t\to\infty} \exp(-Lt) = w_r w_l^T \in M_n \tag{16}$$

Proposition. *Consider a multi-robot system with robot dynamics described by formula (1) and formula (15), by introducing the control input $v_i = \dot{q}_i = (\dot{\rho}_i, \dot{\varphi}_i, \dot{z}_i)^T$ into formula (15), where*

$$\dot{\rho}_i = k_\rho(\rho^* - \rho_i) \tag{17}$$

$$\dot{z}_i = -k_z z_i \tag{18}$$

$$\dot{\varphi}_i = \omega^* + k_\varphi(\bar{\varphi}_i - \varphi_i) \tag{19}$$

Note that k_ρ, k_z and k_φ are positive gains, and

$$\bar{\varphi}_i = \begin{cases} \varphi_{i^-} + \frac{\Delta_i+\Delta_{i^-}}{d_i+d_{i^-}} d_{i^-},\ i = 2, 3, ..., n \\ \varphi_{i^-} + \frac{\Delta_i+\Delta_{i^-}}{d_i+d_{i^-}} d_{i^-} - 2\pi,\ i = 1 \end{cases} \tag{20}$$

then the encirclement problem with arbitrary spacing encoded by formula (9), (10), (11) and (12) can be solved with exponential convergence speed.

Formula (20) is based on the way-point control law proposed in [5] but here we add a particular condition for the robot indexed 1. In particular, substituting formula (7) into formula (20), we have

$$\begin{cases} \bar{\varphi}_1 = \frac{d_1}{d_1+d_n}\varphi_n + \frac{d_n}{d_1+d_n}\varphi_2 - \frac{2\pi d_1}{d_1+d_n} \\ \bar{\varphi}_i = \frac{d_i}{d_i+d_{i-1}}\varphi_{i-1} + \frac{d_{i-1}}{d_i+d_{i-1}}\varphi_{i+1},\ i = 2, ..., n-1 \\ \bar{\varphi}_n = \frac{d_n}{d_n+d_{n-1}}\varphi_{n-1} + \frac{d_{n-1}}{d_n+d_{n-1}}\varphi_1 + \frac{2\pi d_{n-1}}{d_n+d_{n-1}} \end{cases} \tag{21}$$

Proof. It is obvious that formula (17) and formula (18) do not rely on the states of other robots, and they are basically P control laws with reference input ρ^* and 0 respectively. So according to the classical control theory, ρ_i and z_i will converge exponentially to ρ^* and 0 respectively.

We define $\bar{\varphi} = [\bar{\varphi}_1\ ...\ \bar{\varphi}_n]^T$ and $\varphi = [\varphi_1\ ...\ \varphi_n]^T$, so formula (19) and formula (21) can be written into compact forms,

$$\dot{\varphi} = \omega^*\mathbf{1} + k_\varphi(\bar{\varphi} - \varphi) \tag{22}$$

$$\bar{\varphi} = A\varphi + b \tag{23}$$

where $A \in M_n$ is as written in formula (33), and $b \in R^n$ is as follows

$$b = 2\pi \begin{bmatrix} \frac{-d_1}{d_1+d_n} \\ 0 \\ \vdots \\ 0 \\ \frac{d_{n-1}}{d_n+d_{n-1}} \end{bmatrix} \tag{24}$$

Matrix A is a row stochastic matrix [12] and furthermore, it could be considered as the adjacency matrix [13] corresponding to a weighted directed ring denoted by $G(A)$. It can be readily verified that $G(A)$ is strongly connected. Next we define:

$$e_\varphi = \bar{\varphi} - \varphi = (A - I_n)\varphi + b = -L_p\varphi + b \tag{25}$$

where $L_p = I_n - A$, which is the Laplacian matrix of $G(A)$. Therefore, the derivative of e_φ is $\dot{e}_\varphi = -L_p\dot{\varphi}$. By substituting formula (22) and formula (25) into this equation, we further obtain:

$$\dot{e}_\varphi = -\omega^* L_p\mathbf{1} - k_\varphi L_p e_\varphi = -k_\varphi L_p e_\varphi \tag{26}$$

Note that $\mathbf{1}$ is the right eigenvector associated with the zero eigenvalue of L_p, so $-\omega^* L_p\mathbf{1} = 0$. The solution to formula (26) is $e_\varphi(t) = \exp(-k_\varphi L_p t)e_\varphi(0)$. According to

Lemma 1 and also note that $k_\varphi > 0$ only affects the convergence speed but not the convergence value, we have $\lim_{t\to\infty} e_\varphi(t) = w_r w_l^T e_\varphi(0)$, where $L_p w_r = 0,\ w_l^T L_p = 0$ and $w_l^T w_r = 1$. By substituting formula (25) into this equation, we obtain the following:

$$\lim_{t\to\infty} e_\varphi(t) = w_r(-w_l^T L_p \varphi + w_l^T b) = w_l^T b w_r \tag{27}$$

Let $w_r = [1, ..., 1]^T$ and

$$w_L = \begin{bmatrix} (d_1 + d_n) \prod_{j=2}^{n-1} d_j \\ (d_2 + d_1) \prod_{j=3}^{n} d_j \\ \vdots \\ (d_n + d_{n-1}) \prod_{j=1}^{n-2} d_j \end{bmatrix} \tag{28}$$

$$w_l = \frac{1}{||w_L||} w_L \tag{29}$$

where the i^{th} entry of w_L is $[w_{Li} = (d_i + d_{i^-}) \prod_{j=1, j\neq i, i^-}^{n} d_j]$ and $||w_L||$ is its Euclidian norm. It can be easily verified that w_l^T and w_r are the left and right eigenvector of the Laplacian matrix L_p associated with the zero eigenvalue respectively, and $w_l^T w_r = 1$. Therefore, formula (27) becomes $\lim_{t\to\infty} e_\varphi(t) = 0$, which means that the difference between the desired angular position and the actual angular position of each robot vanishes to zero exponentially.

$$\lim_{t\to\infty} \varphi(t) = \lim_{t\to\infty} \bar{\varphi}(t) \tag{30}$$

According to formula (22), the encirclement speed of each robot converges to the desired angular speed ω^*. In addition, under this condition, $\bar{\varphi}_i$ is replaced by φ_i in formula (20) and therefore for robots with indices $i = 2, ..., n$, this equation $\varphi_i = \varphi_{i^-} + \frac{\Delta_i + \Delta_{i^-}}{d_i + d_{i^-}} d_{i^-}$ further becomes

$$\frac{\Delta_i}{\Delta_{i^-}} = \frac{d_i}{d_{i^-}} \tag{31}$$

formula (31) means a sequence of equations $\frac{\Delta_n}{\Delta_{n-1}} = \frac{d_n}{d_{n-1}}, ..., \frac{\Delta_2}{\Delta_1} = \frac{d_2}{d_1}$. Let's assume $\Delta_1 = k d_1,\ k \neq 0$. So we have $\Delta_i = k d_i, i = 2, ..., n$. Since $2\pi = \sum_{i=1}^n d_i = \sum_{i=1}^n \Delta_i = k \cdot 2\pi$, it can be concluded that $k = 1$. Therefore, $\Delta_i = d_i,\ i = 1, ..., n$ so the desired spacing among robots are achieved. □

Remark: If $d_1 = ... = d_n = \frac{2\pi}{n}$, that is, the spacing among robots are equal. Then formula (21) becomes

$$\begin{cases} \bar{\varphi}_1 = \frac{\varphi_n + \varphi_2 - 2\pi}{2} \\ \bar{\varphi}_i = \frac{\varphi_{i-1} + \varphi_{i+1}}{2},\ i = 2, ..., n-1 \\ \bar{\varphi}_n = \frac{\varphi_{n-1} + \varphi_1 + 2\pi}{2} \end{cases} \tag{32}$$

Combined with formula (17), formula (18) and formula (19), they can solve encirclement problem with equal spacing.

$$A = \begin{bmatrix} 0 & \frac{d_n}{d_1+d_n} & 0 & \cdots & 0 & 0 & \frac{d_1}{d_1+d_n} \\ \frac{d_2}{d_2+d_1} & 0 & \frac{d_1}{d_2+d_1} & \cdots & 0 & 0 & 0 \\ 0 & \frac{d_3}{d_3+d_2} & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \frac{d_{n-1}}{d_{n-1}+d_{n-2}} & 0 & \frac{d_{n-2}}{d_{n-1}+d_{n-2}} \\ \frac{d_{n-1}}{d_n+d_{n-1}} & 0 & 0 & \cdots & 0 & \frac{d_n}{d_n+d_{n-1}} & 0 \end{bmatrix} \tag{33}$$

Table 1: Different and common parameters used in Simulation I and Simulation II *

	Simulation I	**Simulation II**
Different parameters	Target velocity: $\dot{p}_b = [0\ 20\ 30]^T$ Initial orientation in *B*: $R_b(0) = Rot(x, \pi/4)$ Angular velocity in *B*: $[0\ 0\ 0]^T$	Target velocity: $\dot{p}_b = [0\ 0\ 0]^T$ Initial orientation in *B*: $R_b(0) = I_3$ Angular velocity in *B*: $[0\ 0.5\ 0]^T$
Common parameters	$\rho^* = 200,\ w^* = 1,\ d = [\frac{\pi}{3}\ \frac{\pi}{3}\ \frac{2\pi}{3}\ \frac{2\pi}{3}]^T,\ k_\rho = 1,\ k_\varphi = 2,\ k_z = 1,$ $p_b(0) = [100\ \ 50\ \ 30]^T,\ p_1(0) = [118\ \ 59\ \text{-}100]^T,\ p_2(0) = [8\ \ 300\ \ 40]^T,$ $p_3(0) = [\text{-}87\ \ 40\ \ 80]^T,\ p_4(0) = [\text{-}200\ \ \text{-}133\ \ 100]^T$	

***** In this table, length unit, angle unit and time unit are cm, rad and s respectively. The target velocity is expressed in the world frame *W*. $Rot(x, \pi/4)$ means rotating the coordinate frame's x-axis by $\pi/4$. The components of the angular velocity in *B* represents angular velocity along x-, y- and z-axis respectively.

3 Simulation Results and Analysis

To validate the effectiveness of the control law proposed in this paper, we carry out two simulation experiments with Simulink, taking into account the possible effect of different parameters (Table 1). In view of formula (15), the orientation of the body reference frame R_b and its derivative $\dot{R}_b^T$ should be calculated or measured by robots. However, on one hand, R_b and $\dot{R}_b^T$ are very difficult to obtain and they might not be useful in practice if we only consider the entrapment application in real world scenarios. On the other hand, it is beneficial to generate complex trajectories if these variables can be specified manually. Therefore, in the simulation experiments, we manually specify them and assume that robots are informed of these two values at every time step. We also assume that the velocity of the target $\dot{p}_b$ is known by all the robots. The results of Simulation I and Simulation II are presented in Fig. 4 and Fig. 5 respectively. It can be seen that the encirclement error signals converge to zero exponentially, and the desired spacing among robots can be finally achieved. Although the two simulation experiments involve only four robots, and the desired spacings are the same, it should be noted that the control law can be extended to a system with any number of robots, and the desired spacing can be specified arbitrarily as long as it satisfies formula (8).

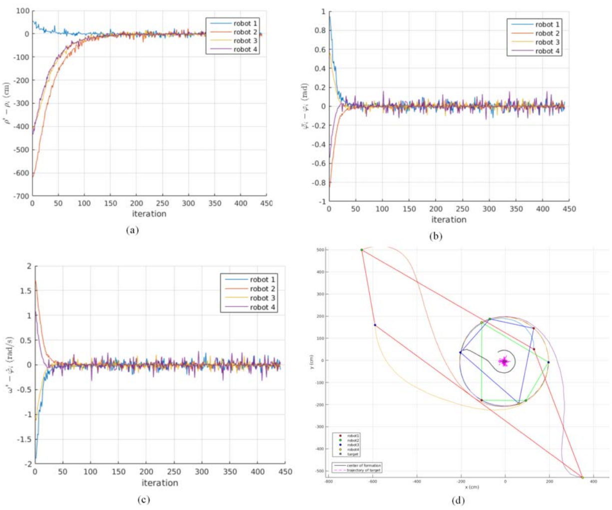

Fig. 2: The results of Gazebo Simulation (part I). (a)-(c) are the encirclement error signals. They are the curves of $\rho^* - \rho_i$, $\bar{\phi}_i - \phi_i$ and $\omega^* - \dot{\phi}_i$ respectively.It is not necessary to plot the $-z$ error signals since all robots move on the ground. (d) are the trajectories of robots on the x-y plane. Red, green and blue solid lines indicate the relative positions of four robots in the beginning, the middle and the end of the process. Broken lines starting from a robot is the trajectory of the robot. (Colour figure online)

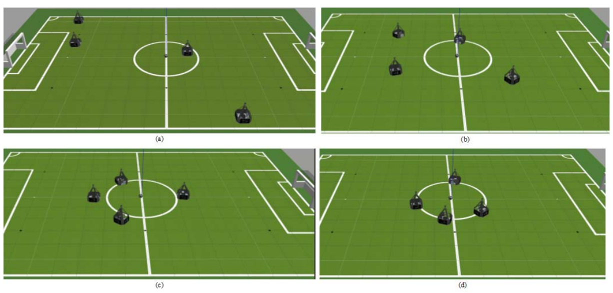

Fig. 3: The results of Gazebo Simulation (part II.) (a)-(d) are the illustrations of four robots at the time of 0s, 3s, 6s and 13s respectively (Color figure online)

To further validate the effectiveness of the proposed control strategy, we then use a simulation system based on ROS and Gazebo for soccer-playing robots [15] to perform experiments. This simulation system is based on the ODE physics engines, and the robot models possess the physical attributes, such as the weight, inertia, etc., of their corresponding real robots [16]. The robot models, just like corresponding real omnidirectional robots, can move in any direction on a plane (i.e. they are modelled by single-integrator dynamics). Therefore, compared with the Simulink simulations, more realistic experiments can be performed on this system, and furthermore, the experimental code could be easily transferred to the real robots without major modifications. In this simulation, each robot can only sense the angular positions of its neighboring two robots. For simplicity, the desired spacings among robots are the same as the previous simulation experiments. The ball is regarded as the target to be encircled. Another important difference is that we have added the Gaussian noise $X \sim \mathcal{N}(0,\, 100)$ to the measured positions and velocities of the target and robots during the encirclement process. Fig. 2 and Fig. 3 are the illustration of the simulation results. It is obvious that although the noise has added fluctuations to the control performance, the multi-robot system can still achieve the desired encirclement behaviour (video is available at http://nubot.trustie.net/videos).

4 Concluding Remarks and Future Work

4.1 Concluding Remarks

This paper presents a distributed control law for multi-robot system to achieve encirclement control around a target with arbitrary spacing, which is useful if robots are heterogeneous in terms of kinematics capabilities. There is no assumption that the robots should be initially placed on a prescribed circle nor should they splay evenly on the circle. Since robots can only perceive the angular positions of their corresponding two neighboring robots, the control law is distributed and scalable. The theoretical analysis using graph theory along with the simulation results prove the effectiveness of the proposed encirclement control law.

4.2 Future Work

Firstly, it is proved in [5] that collisions will not happen using the way-point control law, but this claim is based on the assumption that the robots are regarded as kinematic points. So it is significant to consider the collision avoidance problem when the physical dimensions of robots are taken into account. Secondly, we assume that all the robots have already known the global quantities such as the target velocity, the orientation and the angular velocity of the body reference frame, etc. However, this is too ideal so it should be avoided. Instead, consensus algorithms or other distributed methods should be introduced for each robot to obtain unified global quantities during the encirclement process. Finally, we are interested in implementing this control strategy on real robots and addressing any practical issues that may arise.

5 Acknowledgement

Our work is supported by National Science Foundation of China (No. 61403409, No. 61503401 and No. 61403406) and the graduate school of National University of Defense Technology.

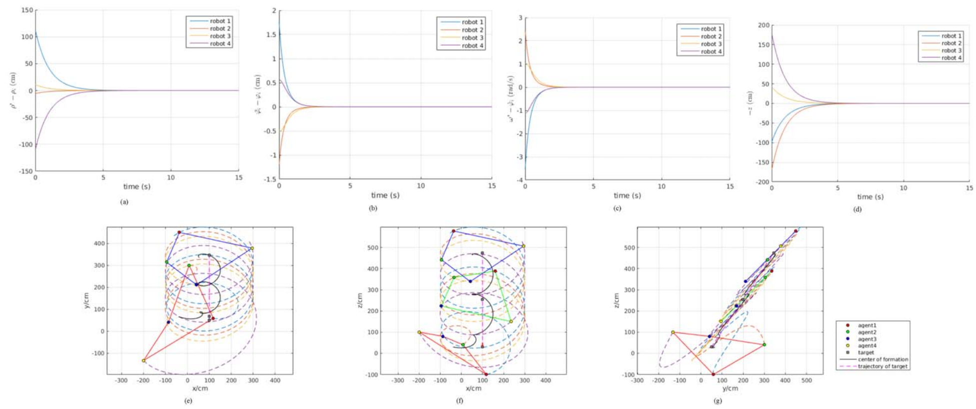

Fig. 4: The results of Simulation I. (a)-(d) are the encirclement error signals. They are the curves of $\rho^* - \rho_i$, $\bar{\phi}_i - \phi_i$, $\omega^* - \dot{\phi}_i$ and $-z$ respectively. (e)-(g) are the trajectories of robots projected to x-y, x-z and y-z plane respectively. Red, green and blue solid lines indicate the relative positions of the four robots in the beginning, the middle and the end of the process. Broken lines starting from a robot is the trajectory of the robot. (Colour figure online)

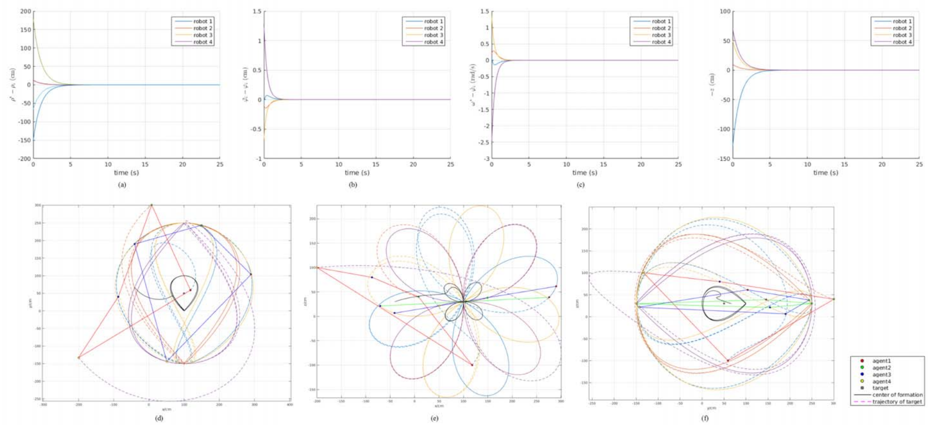

Fig. 5: The results of Simulation II. (a)-(d) are the encirclement error signals. They are the curves of $\rho^* - \rho_i$, $\bar{\phi}_i - \phi_i$, $\omega^* - \dot{\phi}_i$ and $-z$ respectively. (e)-(g) are the trajectories of robots projected to x-y, x-z and y-z plane respectively. Red, green and blue solid lines indicate the relative positions of four robots in the beginning, the middle and the end of the process. Broken lines starting from a robot is the trajectory of the robot. (Colour figure online)

References

[1] X. Wang, Z. Zeng, Y. Cong, Multi-agent distributed coordination control: developments and directions via graph viewpoint, *Neurocomputing* 199: 204-218, 2016

[2] W. Ren, R.W. Beard, *Distributed consensus in multi-vehicle cooperative control*. Springer 2008.

[3] Y. Cao, W. Yu, W. Ren, G. Chen, An overview of recent progress in the study of distributed multi-agent coordination, *IEEE Transactions on Industrial Informatics*, 9(1): 427 - 438, 2012

[4] Albayrak O. *Line and circle formation of distributed autonomous mobile robots with limited sensor range*. Naval Postgraduate School Monterey CA; 1996 Jun.

[5] C. Wang, G. Xie, M. Cao, Forming circle formations of anonymous mobile agents with order preservation, *IEEE Transactions on Automatic Control*, 58(12):3248-3254, 2013

[6] A. Franchi, P. Stegagno, G. Oriolo, Decentralized multi-robot encirclement of a 3D target with guaranteed collision avoidance, *Autonomous Robots*, 40(2):245-265, 2016

[7] Z. Zeng, *Multi-agent coordination control with nonlinear dynamics, quantized communication and structure-constraint*, College of Mechatronics and Automation, National University of Defense Technology, Changsha, Hunan, 2016.

[8] B.J.A. Marshall, M.E. Broucke, B.A. Francis, Formation of vehicles in cyclic pursuit, *IEEE Transactions on Automatic Control*, 49(11):1963-1974, 2015

[9] M. Pavone, E. Frazzoli, Decentralized policies for geometric pattern formation and path coverage, *Journal of Dynamic Systems Measurement and Control*, 129(5):633-643, 2007

[10] S. Zhao, D. Zelazo, Bearing rigidity and almost global bearing-only formation stabilization, *IEEE Transactions on Automatic Control*, 61(5):1255-1268, 2016

[11] X. Wang, T. Liu, J. Qin, Second-order consensus with unknown dynamics via cyclic-small-gain method, *IET Control Theory and Applications*, 6(18):2748-2756, 2012

[12] R.A. Horn, C.R. Johnson, *Matrix analysis*, UK: Cambridge university press, 2012.

[13] M. Mesbahi, M. Egerstedt, *Graph theoretic methods in multi-agent networks*, US: Princeton University Press, 2010.

[14] R. Olfati-Saber, R.M. Murray, Consensus problems in networks of agents with switching topology and time-delays, *IEEE Transactions on Automatic Control*, 49(9):1520-1533, 2004

[15] W. Yao, W. Dai, J. Xiao, H. Lu, A simulation system based on ROS and Gazebo for RoboCup Middle Size League, *IEEE International Conference on Robotics and Biomimetics*, 2015.

[16] D. Xiong, J. Xiao, H. Lu, Z. Zeng, Q. Yu, K. Huang, X. Yi, Z. Zheng, The design of an intelligent soccer-playing robot, *Industrial Robot*, 43(1):91-102, 2016