Journal of Intelligent & Robotic Systems (2019) 94:883–905
https://doi.org/10.1007/s10846-018-0906-5

Distributed Static and Dynamic Circumnavigation Control with Arbitrary Spacings for a Heterogeneous Multi-robot System

Weijia Yao[1] · Huimin Lu[1] · Zhiwen Zeng[1] · Junhao Xiao[1] · Zhiqiang Zheng[1]

Received: 1 November 2017 / Accepted: 6 July 2018 / Published online: 25 July 2018

Abstract

Circumnavigation control algorithms enable multiple robots to rotate around a target while they still preserve a circular formation, which is useful in real world applications such as entrapping a hostile target. Specifically, four quantities are involved: the circumnavigation radius, the angular speed, the height and the phase differences among robots, which are termed *spacings* in this paper. Based on whether these quantities vary or not, the circumnavigation control problem is divided into two categories: the static one and the dynamic one. Corresponding to these two classes, distributed control algorithms are proposed for any number of mobile robots in random 3D positions to circumnavigate a target with arbitrarily given spacings or dynamic spacings. It should be noted that arbitrary spacings or dynamic spacings are useful for a heterogeneous multi-robot system in which robots may possess different kinematics capabilities; robots with higher movement speeds, for instance, can compensate for the insufficiency of those with lower movement speeds by decreasing the corresponding spacings. The robots can only perceive the positions of their two neighbouring robots, so the proposed control algorithms are distributed and scalable. Simulations along with real-robot experiments using soccer-playing robots are conducted to validate the theoretical results.

Keywords Static circumnavigation · Dynamic circumnavigation · Distributed control · Multi-robot system

1 Introduction

Increasing research focuses on multi-robot systems since there are many advantages over single-robot systems [18]. The ways of controlling a multi-robot system can be categorized into two classes: centralized control and distributed control. In 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. However, the prerequisite for this system is that there is at least one robot that is able to communicate with all the other robots [14]. In contrast, distributed control employs local information among robots to realize collective behavior and achieve a global objective [18]. 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 been applied in the military, aerospace, industrial 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. Significant efforts have been made on the *circular formation control* and *circumnavigation control* problems. In the circular formation control problem, robots remain in their positions after the formation is generated, while in circumnavigation control problem, they still encircle around the target. In this sense, circular formation control could be regarded as a special case of circumnavigation control when the circumnavigation speed equals to zero. The control algorithms, however, are different for these two kinds of problems.

There are already many studies on circumnavigation control (or circular formation control) problems. Most of

Electronic supplementary material The online version of this article (https://doi.org/10.1007/s10846-018-0906-5) contains supplementary material, which is available to authorized users.

✉ Huimin Lu
lhmnew@nudt.edu.cn

[1] Department of Automation, National University of Defense Technology, Changsha, Hunan, China

the existing studies only consider the case where robots are distributed evenly on the formation (i.e., equal spacings), such as [9, 13, 15, 23, 25, 26]. In addition, the control algorithms proposed in these studies are only applicable on the 2D plane. Nevertheless, [24] and [5] propose algorithms which are also effective in the 3D space. The formation spacings, however, are fixed and equal. Although this is effective for a homogeneous multi-robot system, it may not be sufficient for a heterogeneous one where robots have different properties, such as maximum movement speeds. [16] and [17] propose a distributed control law for a multi-robot system to form a circular formation with any desired spacings among robots. However, it assumes that the robots are placed initially on a prescribed circle and the control algorithm is not applicable in the 3D space. Furthermore, to the best of our knowledge, there are no studies concerning dynamic spacings for a heterogeneous multi-robot system.

In this paper, the circumnavigation problem is divided into two categories: the static one and the dynamic one. To explain these two categories, four chief quantities are considered in a circumnavigation problem, i.e., the circumnavigation radius, the phase differences among robots (i.e., *spacings*), the circumnavigation height and the circumnavigation angular speed. In the static circumnavigation problem, these four quantities are given and constant during the whole circumnavigation process. Differently, as for the dynamic circumnavigation process, one or more of these quantities will vary dynamically, which is particularly useful in practical scenarios. For example, robots can decrease their spacings to avoid a target from escaping from the circular formation. Most of the existing research only deal with the static circumnavigation problem while in this study, both the static and the dynamic circumnavigation problems are considered.

In this study, we suppose that mobile robots are heterogeneous in terms of their kinematics abilities, such as maximum locomotion speeds, etc. In a scenario where these mobile robots need to entrap a hostile target, their inter-robot spacings should be different for better performance; those robots with lower mobility are supposed to gather together with smaller spacings than those with higher mobility, so the probability for the target to flee away from the formation is lower. Therefore, the first goal of the paper, which corresponds to the static circumnavigation problem, is to propose a distributed circumnavigation control algorithm which is able to control a group of mobile robots from any initial positions to encircle around a target with any given desired spacings in the three dimensional space. However, we do not consider the particular variances in the mobility or the optimal spacings among robots. Instead, the desired spacings are assumed to be appropriate and specified beforehand. We also consider the deterioration of individual performance due to physical worn-out or damage. Therefore, their spacings should be varied in a dynamic way during the circumnavigation process. Based on this, the second goal of this paper, which corresponds to the dynamic circumnavigation problem, is to propose a new distributed circumnavigation control algorithm which is able to drive a group of heterogeneous mobile robots to circumnavigate a target with dynamic spacings in the 3D space.

The main contribution of this work is twofold. First, as for the static circumnavigation problem, the corresponding control algorithm proposed here is able to drive all robots to form a circular formation with given arbitrary spacings in the 3D space. Second, as for the dynamic circumnavigation problem, we propose two new concepts: *utility* and *formation guideline*. Based on them, a distributed dynamic circumnavigation control algorithm is designed, which enables robots to adjust their spacings dynamically according to the local variations of robots' utilities. Moreover, the control algorithm can respond effectively to the situation where robots quit from or join into the formation.

The paper draws upon work previously published in conference proceedings [20–22]. We also publish for the first time new results about the spacing convergence performance as presented in Section 4.3 and we also derive new control algorithms to alleviate the effects of external perturbations as presented in Section 6. Some parts are explained with more details compared with those published in the conference proceedings.

The remainder of this paper is organized as follows. Section 2 introduces the preliminary knowledge and notations that will be used in this paper. Then Section 3 derives the control input from the Cartesian coordinate system to the cylindrical coordinate system, which facilitates the design of control algorithms. In Sections 4 and 5, the static and the dynamic circumnavigation control problems are formulated, and the corresponding control algorithms are proposed. The circumnavigation control problem considering external perturbations is elaborated in Section 6. In Section 7, simulation and real robot experiments are performed to validate the theoretical results. Finally, Section 8 concludes the paper and summarizes the future work. The video about the simulations and experiments is attached to this manuscript.

2 Preliminary Knowledge and Notations

First we list some useful concepts and notations from the matrix analysis theory. For positive integers m and n, $M_n(\mathbb{F})$ and $M_{m\times n}(\mathbb{F})$ are sets of all n-by-n and m-by-n matrices with entries from the field $\mathbb{F}$ respectively. If $\mathbb{F}$ is neglected, then it represents $\mathbb{R}$ by default. If all the entries in a matrix are nonnegative, then this matrix is called nonnegative. We denote $I_d \in M_d$ as the $d \times d$ identity matrix. $\mathbf{1}$ and $\mathbf{0}$ are vectors of all 1's or 0's of suitable dimensions respectively.

$||v||$ represents the Euclidean norm of a vector v. For a nonnegative matrix $A \in M_{m\times n}$, $\sqrt{A} = [\sqrt{a_{ij}}], i = 1, .., m, j = 1, ..., n$. For a vector $v \in \mathbb{R}^n$, $A = diag\{v\} \in M_n$ is a diagonal matrix where $[a_{ii}] = v_i$ and $a_{ij} = 0, i \neq j$. We denote $\lambda(B)$ as the set of all eigenvalues of $B \in M_n$. In addition, if all the eigenvalues of B are real, we use the notation $\lambda_i(B)$, $i = 1, ..., n$, to represent the ith smallest eigenvalue. That is, $\lambda_i(B) \in \lambda(B) \subseteqq \mathbb{R}$ are ordered as follows:

$$\lambda_m(B) = \lambda_1(B) \leq \lambda_2(B) \leq ... \leq \lambda_n(B) = \lambda_M(B). \qquad (1)$$

In clear context, B is neglected.

Some knowledge about the graph theory is also provided here. *Graph* is essentially a concept from the set theory. A graph containing n nodes can be denoted by $\mathcal{G}_n = (\mathcal{V}_n, \mathcal{E}_n)$, where $\mathcal{V}_n = \{v_1, \ldots, v_n\}$ is the *node set* and $\mathcal{E}_n \subseteqq \mathcal{V}_n \times \mathcal{V}_n$ is the *edge set*. $\epsilon_{ij} = (v_i, v_j)$, $i, j \in \{1, \ldots, n\}$ represents an *edge* connecting nodes v_i and v_j. For simplicity, it can also be written as $\epsilon_{ij} = (i, j)$. When $i = j$, ϵ_{ij} is called a *self-loop* [4], which will not be considered in this study. When the elements in the edge set $\mathcal{E}_n$ are ordered pairs, $\mathcal{G}_n$ is a *directed graph* (or *digraph*) while it is an *undirected graph* if the elements are unordered pairs. In other words, $\epsilon_{ij} \neq \epsilon_{ji}$ for directed graph but $\epsilon_{ij} = \epsilon_{ji}$ for undirected graph. Node i of ϵ_{ij} is called the *parent node* and node j is the *child node* [14]. A *directed path* is an ordered set of edges such that the child node of the previous edge is the parent node of the next edge, such as $(v_1, v_2), (v_2, v_3), \ldots$. For undirected graph, the concept of *undirected path* is similarly defined. A directed graph is called *strongly connected* if for every pair of nodes there is a directed path between them [10]. Similarly, an undirected graph is *connected* if for every pair of nodes there is an undirected path between them. The corresponding *adjacent matrix* to graph $\mathcal{G}_n = (\mathcal{V}_n, \mathcal{E}_n)$ is denoted by $A_n = [a_{ij}] \in M_n, i, j \in 1, \ldots, n$. It represents whether there is an edge between any two nodes, that is, $a_{ji} > 0$ when the edge $(i, j) \in \mathcal{E}_n$ and $a_{ji} = 0$ when the edge $(i, j) \notin \mathcal{E}_n$. The *Laplacian matrix* is denoted by $L_n = [l_{ij}] \in M_n$, where $l_{ij} = \sum_{k=1,k\neq i}^{n} a_{ik}, i = j;\ l_{ij} = -a_{ij}, i \neq j$. The underlying directed graph of a nonnegative matrix $M \in M_n$, denoted by $\mathcal{G}(M)$, is the directed graph with the node set $\{v_i\}, i \in \{1, ..., n\}$, such that there is a directed edge in $\mathcal{G}(M)$ from v_j to v_i if and only if $m_{ij} \neq 0$ [7].

3 Cylindrical Control Input

In this part, the control input to the robot is derived. Instead of using the Cartesian coordinate system to represent the control input, the cylindrical coordinate system is adopted for better simplicity and clarity. 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 on a circular formation. Suppose each mobile robot is modeled by a 3D kinematic point:

$$\dot{p}_i(t) = u_i(t), \quad i = 1, ..., n, \qquad (2)$$

where $u_i(t)$ is the control input to the robot r_i and $p_i(t) \in \mathbb{R}^3$ is its position in the world reference frame $\mathcal{W}$. In this problem, robots are required to maintain on the same plane with the encircled target which is modeled by another kinematic point. Therefore, a (target) body reference frame $\mathcal{B}$ centered at the target S is introduced (see Fig. 1). In addition, the cylindrical coordinate system is preferred to the commonly used Cartesian coordinate system since the former itself embodies three elements of interest: the distance between the projection of the robot on the XSY plane to the target (ρ), the height relative to the XSY plane (z) and the angle between the X-axis and the line joining the projection of the robot on the XSY plane with the target (φ). The cylindrical coordinates for r_i are 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 $q(p) = (\rho(p), \varphi(p), z(p))^T$, where $p \in \mathbb{R}^3$ is a vector with components p_x, p_y, p_z and $\rho(p) = \sqrt{p_x^2 + p_y^2}$, $\varphi(p) = \tan^{-1}(p_y/p_x)$, $z(p) = p_z$. Note that $\varphi \in [0, 2\pi)$. The Jacobian matrix of the vector function will be used later, which is

$$J = \frac{\partial q}{\partial p^T} = \begin{bmatrix} \frac{p_x}{\sqrt{p_x^2+p_y^2}} & \frac{p_y}{\sqrt{p_x^2+p_y^2}} & 0 \\ \frac{-p_y}{p_x^2+p_y^2} & \frac{p_x}{p_x^2+p_y^2} & 0 \\ 0 & 0 & 1 \end{bmatrix}. \qquad (3)$$

For better analysis, we label the robots in the counterclockwise direction according to their initial (angular) positions (φ_i) in $\mathcal{B}$ as shown in Fig. 2. Note that the subscript i^- and i^+ represent the indices of the neighbouring robots of r_i in the clockwise and counter-clockwise direction respectively. Especially, if $i = n$, $i^+ = 1$, and if $i = 1$,

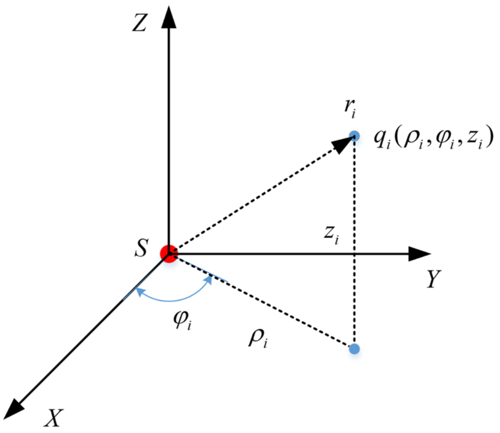


Fig. 1 The body reference frame $\mathcal{B}$ with the target S as the origin

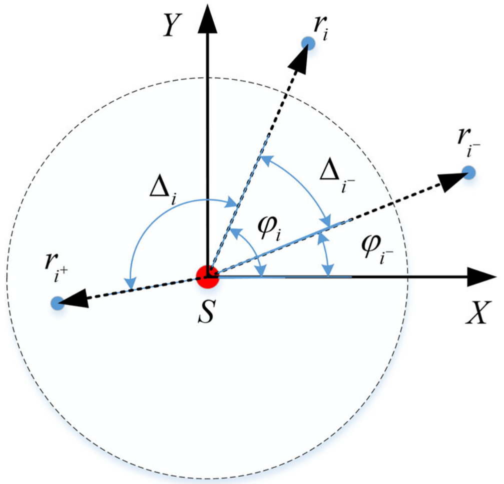


Fig. 2 Robots' projections and the target S on the XSY plane

$i^- = n$. $\Delta_i > 0$ represents the difference between the angular position of r_{i^+} and that of r_i. In particular,

$$\Delta_i = \begin{cases} \varphi_{i^+} - \varphi_i, & i = 1, \ldots, n-1, \\ \varphi_1 - \varphi_n + 2\pi, & i = n. \end{cases} \tag{4}$$

To consider the circumnavigation control problem in the body reference frame $\mathcal{B}$ so that we can take advantage of the cylindrical coordinates, first we define a rotational matrix R_b, which is the representation of $\mathcal{B}$ with respective to the world reference frame $\mathcal{W}$. Therefore, the following formula calculates the cylindrical coordinates of the robot r_i in frame $\mathcal{B}$.

$$q_i = q(R_b^T(p_i - p_b)), \tag{5}$$

where p_i and p_b are the Cartesian coordinates of the robot r_i and the target in frame $\mathcal{W}$ respectively. Then the derivative of Eq. 5 is the dynamics of robots in the cylindrical coordinates, which is

$$\dot{q}_i = J_i \left[\dot{R}_b^T(p_i - p_b) + R_b^T(\dot{p}_i - \dot{p}_b) \right], \tag{6}$$

where J_i is the Jacobian matrix as shown in Eq. 3, i.e. $J_i = \frac{\partial q}{\partial p^T}\Big|_{p = R_b^T(p_i - p_b)}$. Note that in view of Eq. 3, $\det(J) = \frac{1}{\sqrt{p_x^2 + p_y^2}}$ as long as $p_x^2 + p_y^2 \neq 0$. In the scenario of the circumnavigation 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 do not coincide, and by designing appropriate control laws, the distance can be guaranteed to be nonzero all the time. By letting

$$u_i = \dot{p}_i = \dot{p}_b + R_b(J_i^{-1}v_i - \dot{R}_b^T(p_i - p_b)), \tag{7}$$

we can switch our focus to the new control input in cylindrical coordinates $v_i = \dot{q}_i = (\dot{\rho}_i, \dot{\varphi}_i, \dot{z}_i)^T$ [5]. 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 circumnavigation control problem.

4 Static Circumnavigation Control Problem

In the static circumnavigation control problem, robots and the target maintain on the same plane with any given constant circumnavigation radius, spacings, angular speed and height during the steady state. These four quantities are unchanged compared with those in the dynamic circumnavigation control problem which will be discussed in Section 5. In this section, the formulation of the static circumnavigation control problem is presented first. And then the corresponding control algorithm is proposed and the convergence speed of the robots' spacings is analyzed.

4.1 Problem Formulation

We define $\Delta_i^d > 0$, $i = 1, \ldots, n$, as the desired spacings between the robot r_{i^+} and r_i. Also note that

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

where $\Delta_i > 0$, $\Delta_i^d > 0$. Therefore, the static circumnavigation control problem can be formulated as follows:

Definition 1 (**Static Circumnavigation Control Problem with Arbitrary Spacings**) In a heterogeneous multi-robot system composed of n ($n \geq 2$) mobile robots, of which the dynamics are modeled by Eq. 2, the static circumnavigation control problem with arbitrary spacings is to seek control laws satisfying the following asymptotic conditions:

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

$$\lim_{t\to\infty} \Delta_i(t) = \Delta_i^d, \tag{10}$$

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

$$\lim_{t\to\infty} z_i(t) = z^*, \tag{12}$$

for all $i = 1, ..., n$. Here, $\rho^* > 0$, $\omega^* > 0$ and $\Delta_i^d > 0$ denote the circumnavigation radius, the angular speed and the desired spacings among robots respectively, and Δ_i^d satisfies Eq. 8.

In this paper, it is required that all robots and the target remain on the same plane in the end. Therefore, the default value of z^* is 0.

4.2 Control Algorithm

The following is a preliminary result related to any strongly connected digraph.

Lemma 1 (**Theorem 3 of** [12]) *Assume G is a strongly connected digraph with Laplacian L satisfying $Lw_r = \mathbf{0}$, $w_l^T L = \mathbf{0}$ and $w_l^T w_r = 1$. Then*

$$R = \lim_{t\to\infty} \exp(-Lt) = w_r w_l^T \in M_n. \tag{13}$$

Theorem 1 *Consider a multi-robot system with robot dynamics described by* Eq. 2, *by introducing the control input $v_i = \dot{q}_i = (\dot{\rho}_i, \dot{\varphi}_i, \dot{z}_i)^T$ into* Eq. 7, *where*

$$\dot{\rho}_i = k_\rho(\rho^* - \rho_i), \tag{14}$$

$$\dot{z}_i = -k_z z_i, \tag{15}$$

$$\dot{\varphi}_i = \omega^* + k_\varphi(\bar{\varphi}_i - \varphi_i). \tag{16}$$

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

$$\bar{\varphi}_i = \begin{cases} \varphi_{i^-} + \dfrac{\Delta_i + \Delta_{i^-}}{\Delta_i^d + \Delta_{i^-}^d}\Delta_{i^-}^d, & i = 2, 3, \ldots, n, \\ \varphi_n + \dfrac{\Delta_1 + \Delta_n}{\Delta_1^d + \Delta_n^d}\Delta_n^d - 2\pi, & i = 1, \end{cases} \tag{17}$$

then the static circumnavigation control problem with arbitrary spacings encoded by Eqs. 9, 10, 11 *and* 12 *can be solved.*

Equation 17 is based on the way-point control law proposed in [16] but here we add a particular condition for the robot indexed 1. In particular, substituting Eq. 4 into Eq. 17, we have

$$\begin{cases} \bar{\varphi}_1 = \frac{\Delta_1^d}{\Delta_1^d+\Delta_n^d}\varphi_n + \frac{\Delta_n^d}{\Delta_1^d+\Delta_n^d}\varphi_2 - \frac{2\pi\Delta_1^d}{\Delta_1^d+\Delta_n^d}, \\ \bar{\varphi}_i = \frac{\Delta_i^d}{\Delta_i^d+\Delta_{i-1}^d}\varphi_{i-1} + \frac{\Delta_{i-1}^d}{\Delta_i^d+\Delta_{i-1}^d}\varphi_{i+1}, \ i = 2, \ldots, n-1, \\ \bar{\varphi}_n = \frac{\Delta_n^d}{\Delta_n^d+\Delta_{n-1}^d}\varphi_{n-1} + \frac{\Delta_{n-1}^d}{\Delta_n^d+\Delta_{n-1}^d}\varphi_1 + \frac{2\pi\Delta_{n-1}^d}{\Delta_n^d+\Delta_{n-1}^d}. \end{cases} \tag{18}$$

Proof It is obvious that Eqs. 14 and 15 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 Eqs. 16 and 18 can be written into compact forms:

$$\dot{\varphi} = \omega^*\mathbf{1} + k_\varphi(\bar{\varphi} - \varphi), \tag{19}$$

$$\bar{\varphi} = A\varphi + b, \tag{20}$$

where $A \in M_n$ and $b \in R^n$ are as follows:

$$A = \begin{bmatrix} 0 & \frac{\Delta_n^d}{\Delta_1^d+\Delta_n^d} & 0 & \ldots & 0 & 0 & \frac{\Delta_1^d}{\Delta_1^d+\Delta_n^d} \\ \frac{\Delta_2^d}{\Delta_2^d+\Delta_1^d} & 0 & \frac{\Delta_1^d}{\Delta_2^d+\Delta_1^d} & \ldots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & \frac{\Delta_{n-1}^d}{\Delta_{n-1}^d+\Delta_{n-2}^d} & 0 & \frac{\Delta_{n-2}^d}{\Delta_{n-1}^d+\Delta_{n-2}^d} \\ \frac{\Delta_{n-1}^d}{\Delta_n^d+\Delta_{n-1}^d} & 0 & 0 & \ldots & 0 & \frac{\Delta_n^d}{\Delta_n^d+\Delta_{n-1}^d} & 0 \end{bmatrix}. \tag{21}$$

$$b = 2\pi \begin{bmatrix} \frac{-\Delta_1^d}{\Delta_1^d+\Delta_n^d} & 0 \cdots 0 & \frac{\Delta_{n-1}^d}{\Delta_n^d+\Delta_{n-1}^d} \end{bmatrix}^T. \tag{22}$$

Matrix A is a row stochastic matrix [7] and furthermore, it could be considered as the adjacency matrix [10] corresponding to a weighted directed ring denoted by $\mathcal{G}(A)$, which is strongly connected. Next we define the *error signal* as

$$e_\varphi = \bar{\varphi} - \varphi = (A - I_n)\varphi + b = -L_p\varphi + b, \tag{23}$$

where $L_p = I_n - A$, which is the Laplacian matrix of $\mathcal{G}(A)$. Therefore, the derivative of e_φ is $\dot{e}_\varphi = -L_p\dot{\varphi}$. By substituting Eqs. 19 and 23 into this equation, we further obtain the *error dynamics* as

$$\dot{e}_\varphi = -\omega^* L_p\mathbf{1} - k_\varphi L_p e_\varphi = -k_\varphi L_p e_\varphi. \tag{24}$$

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 Eq. 24 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 $L_p w_r = \mathbf{0}$, $w_l^T L_p = \mathbf{0}$ and $w_l^T w_r = 1$. By substituting Eq. 23 into this equation, we obtain the following:

$$\lim_{t\to\infty} e_\varphi(t) = w_r(-w_l^T L_p\varphi + w_l^T b) = w_l^T b w_r. \tag{25}$$

Let $w_r = \mathbf{1}$ and

$$w_l = \frac{w_L}{\sum_{w_L}}, \tag{26}$$

where the ith entry of w_L is $[w_{L_i} = (\Delta_i^d + \Delta_{i^-}^d)\prod_{j=1,j\neq i,i^-}^n \Delta_j^d]$ and $\sum_{w_L} = \sum_{i=1}^n w_{L_i}$. 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, Eq. 25 becomes $\lim_{t\to\infty} e_\varphi(t) = \mathbf{0}$. In other words,

$$\lim_{t\to\infty} \varphi(t) = \lim_{t\to\infty} \bar{\varphi}(t). \tag{27}$$

According to Eq. 19, the circumnavigation 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 Eq. 17 and therefore for robots with indices $i = 2, ..., n$, the equation $\varphi_i = \varphi_{i^-} + \frac{\Delta_i + \Delta_{i^-}}{\Delta_i^d + \Delta_{i^-}^d} \Delta_{i^-}^d$ further becomes

$$\frac{\Delta_i}{\Delta_{i^-}} = \frac{\Delta_i^d}{\Delta_{i^-}^d}. \tag{28}$$

Equation 28 means a sequence of equations $\frac{\Delta_n}{\Delta_{n-1}} = \frac{\Delta_n^d}{\Delta_{n-1}^d}, ..., \frac{\Delta_2}{\Delta_1} = \frac{\Delta_2^d}{\Delta_1^d}$. Assuming $\Delta_1 = k\Delta_1^d$, $k \neq 0$, we have $\Delta_i = k\Delta_i^d, i = 2, ..., n$. Since $2\pi = \sum_{i=1}^n \Delta_i^d = \sum_{i=1}^n \Delta_i = k \cdot 2\pi$, it can be concluded that $k = 1$. Therefore, $\Delta_i = \Delta_i^d$, $i = 1, ..., n$, so the desired spacings are achieved. □

Remark 1 If $\Delta_1^d = \cdots = \Delta_n^d = \frac{2\pi}{n}$, that is, the spacings are equal. Then Eq. 18 becomes

$$\begin{cases} \bar{\varphi}_1 = \frac{\varphi_n + \varphi_2 - 2\pi}{2}, \\ \bar{\varphi}_i = \frac{\varphi_{i-1} + \varphi_{i+1}}{2}, \ i = 2, ..., n-1, \\ \bar{\varphi}_n = \frac{\varphi_{n-1} + \varphi_1 + 2\pi}{2}. \end{cases} \tag{29}$$

Combined with Eqs. 14, 15 and 16, they can solve the circumnavigation control problem with equal spacings.

Remark 2 It is interesting that when there are only two robots, i.e., $n = 2$, the subscripts of robots satisfy equation $i^- = i^+$. Equation 18 is still valid but simplifies to $\bar{\varphi}_1 = \varphi_2 - \Delta_1^d$, $\bar{\varphi}_2 = \varphi_1 + \Delta_1^d$, which becomes very intuitive to understand given that $\Delta_1^d + \Delta_2^d = 2\pi$. Furthermore, A and b degrades to $A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ and $b = \begin{bmatrix} -\Delta_1^d \\ \Delta_1^d \end{bmatrix}$ respectively .

Another problem that is worth considering is whether robots preserve their initial orders during the whole circumnavigation process. This means each robot will not overtake or be overtaken by its neighbours, which guarantees that they will not collide with each other if they are regarded as mass points. Before introducing the next theorem, the definition of a Metzler matrix [11] is given. For a real matrix $M = [m_{ij}] \in M_n$, if all its off-diagonal elements are non-negative, i.e., $m_{ij} \geq 0, i \neq j$, M is a Metzler matrix.

Theorem 2 *During the circumnavigation process, robots always keep their initial orders in the formation. In other words, $\Delta_i(t) > 0, i = 1, \ldots, n$, for $t \geq 0$.*

Proof According to Eqs. 4, 16 and 17, for $i = 1, \ldots, n$, it follows that

$$\dot{\Delta}_i = k_\varphi \left[\frac{\Delta_i^d}{\Delta_{i+}^d + \Delta_i^d} \Delta_{i+} + \left(\frac{-\Delta_{i+}^d}{\Delta_{i+}^d + \Delta_i^d} + \frac{-\Delta_{i-}^d}{\Delta_i^d + \Delta_{i-}^d} \right) \Delta_i + \frac{\Delta_i^d}{\Delta_i^d + \Delta_{i-}^d} \Delta_{i-} \right]. \tag{30}$$

Let $\Delta = [\Delta_1 \ldots \Delta_n]^T$, then Eq. 30 can be rewritten as $\dot{\Delta} = k_\varphi M_\Delta \Delta$, where M_Δ is shown in Eq. 31. Therefore, the solution of $\Delta(t)$ is $\Delta(t) = \exp(k_\varphi M_\Delta t)\Delta(0)$. Since M_Δ is a Metzler matrix, it has been proved that $\exp(k_\varphi M_\Delta t)$ is a non-negative matrix [11]. In addition, due to $\Delta(0) > 0$, it follows that $\Delta(t) > 0$, $t \geq 0$, which means that robots always keep their initial orders in the formation.

$$M_\Delta = \begin{bmatrix} \frac{-\Delta_2^d}{\Delta_2^d + \Delta_1^d} + \frac{-\Delta_n^d}{\Delta_1^d + \Delta_n^d} & \frac{\Delta_1^d}{\Delta_2^d + \Delta_1^d} & \cdots & \frac{\Delta_1^d}{\Delta_1^d + \Delta_n^d} \\ \frac{\Delta_2^d}{\Delta_2^d + \Delta_1^d} & \frac{-\Delta_3^d}{\Delta_3^d + \Delta_2^d} + \frac{-\Delta_1^d}{\Delta_2^d + \Delta_1^d} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \frac{\Delta_{n-1}^d}{\Delta_n^d + \Delta_{n-1}^d} \\ \frac{\Delta_n^d}{\Delta_1^d + \Delta_n^d} & 0 & \cdots & \frac{-\Delta_1^d}{\Delta_1^d + \Delta_n^d} + \frac{-\Delta_{n-1}^d}{\Delta_n^d + \Delta_{n-1}^d} \end{bmatrix}. \tag{31}$$

□

4.3 Spacing Convergence Performance

Note that among the control inputs represented by Eqs. 14, 15 and 16, only Eq. 16 utilizes the information among robots. It is of particular interest to study the convergence performance corresponding to this specific control input, which is directly related to the desired arbitrary spacings. In particular, we aim to find out how fast the current spacings converge to the desired spacings, i.e. $e_\varphi = \bar{\varphi} - \varphi$, using a Lyapunov function. Let $P \triangleq diag\{w_l\}$, which is obviously a positive definite square matrix and $trace(P) = 1$. The valid Lyapunov equation $Q \triangleq L_p^T P + PL_p$ and the constructed matrix $D \triangleq (\sqrt{P}^{-1})^T Q \sqrt{P}^{-1}$ are important for the following proofs, so we provide a lemma demonstrating some of their important properties first.

Lemma 2 *Let $P \triangleq diag\{w_l\}$, where w_l is the normalized left eigenvector of L_p associated with the zero eigenvalue as presented by* Eq. 26*. Let $Q \triangleq L_p^T P + PL_p$ and*

$D \triangleq (\sqrt{P}^{-1})^T Q \sqrt{P}^{-1}$. *$Q$ and D have the following properties respectively:*

(1) *Q is a symmetric balanced Laplacian matrix. Therefore, some of its properties include $Q = Q^T, \mathbf{1}^T Q = Q\mathbf{1} = \mathbf{0}$;*
(2) *The underlying graph of Q, i.e. $\mathcal{G}(Q)$, represents a strongly connected graph;*
(3) *D is real symmetric (or Hermitian), i.e. $D = D^T$;*
(4) *$\eta = \sqrt{P}\mathbf{1}$ and $\eta^T = (\sqrt{P}\mathbf{1})^T$ are the right and left eigenvectors of D associated with the zero eigenvalue respectively, i.e. $D\eta = \eta^T D = \mathbf{0}$;*
(5) *The eigenvalues of D are all real and nonnegative, i.e. $\lambda(D) \geq 0$. In addition, 0 is a simple eigenvalue of D, i.e. $0 = \lambda_1(D) < \lambda_2(D)$. More specifically, $\lambda(D) = 2\lambda(L_p)$.*

Proof The results of (1) to (4) are quite straightforward considering that P is a diagonal matrix with positive diagonal elements. We prove (5) as follows. According to (3), the eigenvalues of D are real. It can be easily verified that $PL_pP^{-1} = L_p^T$, or

$$PL_p = L_p^T P. \tag{32}$$

Therefore, $D = (\sqrt{P}^{-1})^T Q \sqrt{P}^{-1} = 2\sqrt{P}^{-1}(PL_p)\sqrt{P}^{-1} = 2\sqrt{P}L_p\sqrt{P}^{-1} \triangleq 2E$. Since E is similar to L_p, we have $\lambda(D) = 2\lambda(E) = 2\lambda(L_p)$. Considering L_p is a Laplacian matrix with a simple zero eigenvalue, the proof is completed. □

It should be noted that the property (5) of Lemma 2 also implies that all the eigenvalues of the Laplacian matrix L_p are real. The next lemma is a famous theorem on Rayleigh quotient [7]. We modify some notations for consistency with those used in this paper and only keep the parts we need.

Lemma 3 (Theorem 4.2.2 (Rayleigh) of [7]**)** *Let $A \in M_n(\mathbb{C})$ be Hermitian, the eigenvalues of A be ordered as in* Eq. 1*, $i_1, ..., i_k$ be given integers with $1 \leq i_1 < ... < i_k \leq n$, $x_{i_1}, ..., x_{i_k}$ be orthonormal and such that $Ax_{i_p} = \lambda_{i_p}x_{i_p}$ for each $p = 1, ..., k$, and $S = span\{x_{i_1}, ..., x_{i_k}\}$. Then*

$$\lambda_{i_1} = \min_{\{x:0\neq x\in S\}} \frac{x^*Ax}{x^*x}, \tag{33}$$

where x^* denotes the conjugate transpose of x. The minimum value is achieved if and only if $Ax = \lambda_{i_1}x$.

Corollary 1 *Let $A \in M_n$ be real symmetric, the eigenvalues of A be ordered as in* Eq. 1*, $x_1, ..., x_n \in \mathbb{R}^n$ be orthonormal and such that $Ax_i = \lambda_i x_i$ for each $i = 1, ..., n$, and $S = span\{x_2, ..., x_n\}$. Then*

$$\lambda_2 = \min_{\{x\in\mathbb{R}^n:x\neq 0, x_1^T x=0\}} \frac{x^TAx}{x^Tx}. \tag{34}$$

The minimum value is achieved if and only if $Ax = \lambda_2 x$.

Proof We consider Lemma 3 in the field of $\mathbb{R}$. Moreover, it is a basic fact that there is an orthonormal basis of $\mathbb{R}^n$ consisting of eigenvectors of A, i.e. $span\{x_1, ..., x_n\} = \mathbb{R}^n$. Also note that $S = span\{x_2, ..., x_n\} = \{x \in \mathbb{R}^n : x_1^T x = 0\} \subseteqq \mathbb{R}^{n-1}$. By letting $k = n - 1$ and $i_1 = 2, i_2 = 3, ..., i_k = n$, the remaining of the proof follows from Lemma 3, i.e. $\lambda_2 = \min_{\{x\in\mathbb{R}^n:0\neq x\in S\}} \frac{x^TAx}{x^Tx} = \min_{\{x\in\mathbb{R}^n:x\neq 0,x_1^Tx=0\}} \frac{x^TAx}{x^Tx}$. □

Lemma 4 ([1]) *Let $P = P^T \in \mathbb{R}^{n\times n}$, $\lambda_M(P)$ and $\lambda_m(P)$ denote the largest and smallest eigenvalues of P, respectively, and $||\cdot||$ denote the Euclidean norm. Then*

$$\lambda_m(P)||x||^2 \leq v(x) = x^TPx \leq \lambda_M(P)||x||^2, \tag{35}$$

for all $x \in \mathbb{R}^n$.

As for the convergence performance of $e_\varphi(t)$, we obtain the following theorem.

Theorem 3 *Consider the control input* (16)*, which involves distributed inter-robot information exchange corresponding to a fixed strongly-connected digraph. The error signal $e_\varphi(t)$, as the solution of the error dynamics* (24)*, globally exponentially vanishes, which is described by the following inequality:*

$$||e_\varphi(t)|| \leq \sqrt{\frac{\lambda_M(P)}{\lambda_m(P)}}||e_\varphi(0)|| \exp(-k_\varphi\beta t), \tag{36}$$

where $P = diag\{w_l\}$, w_l is the normalized left eigenvector of L_p associated with the zero eigenvalue as presented in Eq. 26*, $\lambda_M(\cdot)$ and $\lambda_m(\cdot)$ denote the maximum and minimum eigenvalues respectively and $\beta = \lambda_2(L_p)$.*

Proof Since P is positive definite, we define a valid Lyapunov function for the error dynamics (24) as follows:

$$V(e_\varphi) = e_\varphi^T P e_\varphi. \tag{37}$$

After taking the derivative of $V(e_\varphi)$ we have

$$\begin{aligned} \dot{V}(e_\varphi) &= \dot{e_\varphi}^T P e_\varphi + e_\varphi^T P \dot{e_\varphi} \\ &= -k_\varphi e_\varphi^T (L_p^T P + PL_p) e_\varphi \\ &= -k_\varphi e_\varphi^T Q e_\varphi. \end{aligned} \tag{38}$$

Some of the properties of Q is presented in Lemma 2. Let $D = (\sqrt{P}^{-1})^T Q \sqrt{P}^{-1}$. According to the properties presented in Lemma 2, D is real symmetric and $\hat{\eta} = \eta / ||\eta||$ is the right normalized eigenvector of D associated with the simple eigenvalue $\lambda_1(D) = 0$. Since the multiplicity of $\lambda_1(D)$ is 1, $\hat{\eta}$ is automatically orthogonal to the space spanned by the rest of the eigenvectors of D [7]. According to Corollary 1, we have

$$\min_{\{y \in \mathbb{R}^n : y \neq 0, \hat{\eta}^T y = 0\}} \frac{y^T D y}{y^T y} = \lambda_2(D). \tag{39}$$

Or, for $y \in \mathbb{R}^n$, $y \neq 0$, $\hat{\eta}^T y = 0$, the following inequality is satisfied:

$$y^T D y \geq \lambda_2(D) y^T y. \tag{40}$$

According to Eq. 24, we have $w_l^T \dot{e}_\varphi = -k_\varphi w_l^T L_p e_\varphi = 0$, so $w_l^T e_\varphi(t)$ is an invariant quantity. In addition, $w_l^T e_\varphi(\infty) = 0$, so we have

$$w_l^T e_\varphi(t) = 0, \tag{41}$$

during the whole encirclement process. Since $\hat{\eta}^T(\sqrt{P} e_\varphi) = \mathbf{1}^T P e_\varphi / ||\eta|| = w_l^T e_\varphi / ||\eta|| = 0$, we substitute y by $\sqrt{P} e_\varphi$ in Eq. 40 and obtain the following inequality:

$$e_\varphi^T Q e_\varphi \geq 2\beta V(e_\varphi), \tag{42}$$

where $\beta = \lambda_2(L_p)$. Note that we have used the fact that $Q = \sqrt{P}^T D \sqrt{P}$ and the property (5) listed in Lemma 2. Therefore, Eq. 38 can be written into

$$\dot{V}(e_\varphi) = -k_\varphi e_\varphi^T Q e_\varphi \leq -2k_\varphi \beta V(e_\varphi). \tag{43}$$

Since $V(e_\varphi)$ is positive definite, we have

$$V(e_\varphi) \leq V[e_\varphi(0)] \exp(-2k_\varphi \beta t). \tag{44}$$

According to Lemma 4, the inequality (44) can be further derived as follows:

$$\lambda_m(P) ||e_\varphi||^2 \leq V(e_\varphi) \leq \lambda_M(P) ||e_\varphi(0)||^2 \exp(-2k_\varphi \beta t). \tag{45}$$

Equivalently,

$$||e_\varphi(t)|| \leq \sqrt{\frac{\lambda_M(P)}{\lambda_m(P)}} ||e_\varphi(0)|| \exp(-k_\varphi \beta t). \tag{46}$$

□

Remark 3 Since the underlying graph for the Laplacian matrix L_p is strongly connected, the multiplicity of $\lambda_1(D)$ is 1. However, considering a more general case when the underlying graph is not strongly connected, hence the multiplicity of λ_1 is greater than 1. In this case, Gram-Schmidt process can be applied to the eigenvectors associated with λ_1 while we keep $\hat{\eta}$ unchanged. Then $\hat{\eta}$ is still orthogonal to the space spanned by the new set of the remaining eigenvectors of D. Therefore, Eq. 39 is still valid. However, since $\lambda_2(D) = 0$, the following part of the proof becomes insignificant.

Remark 4 Let $0 < \Delta_{s_1}^d \leq \Delta_{s_2}^d \leq \cdots \leq \Delta_{s_n}^d < 2\pi$, where $s_i \in \{1, 2, ..., n\}$ and $s_i \neq s_j$ $(i \neq j)$. Then it is easy to verify that $\lambda_M(P) = \frac{(\Delta_{s_1}^d + \Delta_{s_2}^d)\Delta_{s_3}^d \cdots \Delta_{s_n}^d}{\sum_{w_L}}$ and $\lambda_m(P) = \frac{(\Delta_{s_{n-1}}^d + \Delta_{s_n}^d)\Delta_{s_1}^d \cdots \Delta_{s_{n-2}}^d}{\sum_{w_L}}$. Therefore, in particular,

$$\begin{aligned} ||e_\varphi(t)|| &\leq \sqrt{\frac{(\Delta_{s_1}^d + \Delta_{s_2}^d)\Delta_{s_{n-1}}^d \Delta_{s_n}^d}{(\Delta_{s_{n-1}}^d + \Delta_{s_n}^d)\Delta_{s_1}^d \Delta_{s_2}^d}} ||e_\varphi(0)|| \exp(-k_\varphi \beta t) \\ &\leq \sqrt{\frac{\Delta_{s_{n-1}}^d \Delta_{s_n}^d}{\Delta_{s_1}^d \Delta_{s_2}^d}} ||e_\varphi(0)|| \exp(-k_\varphi \beta t) \end{aligned} \tag{47}$$

Remark 5 In some literature, such as [12], β is called the Fiedler eigenvalue. In addition, in [12], the underlying graph is required to be not only strongly connected but also balanced, and $\beta = \lambda_2(G')$ where $G' = (L + L^T)/2$ and L is the balanced Laplacian matrix. However, we do not require the underlying graph to be balanced in this study (e.g., the spacings among robots are not required to be equal). With a greater gain k_φ, the circumnavigation process will converge faster.

5 Dynamic Circumnavigation Control Problem

Different from the static circumnavigation control problem, one or several of the values of the circumnavigation radius, the angular speed, the spacings and the height are time varying in the dynamic circumnavigation control problem. In this section, we only consider the time-varying spacings among robots. First, the insufficiencies of the static circumnavigation control algorithm are pointed out. Then concerning theses insufficiencies, the problem formulation and the control algorithm for the dynamic circumnavigation control problem are proposed. Specifically, the concepts of utilities and formation guidelines are proposed in this study. Finally, a simulation and a real-robot experiment are conducted.

5.1 Insufficiencies of the Static Circumnavigation Control Algorithm

For the static circumnavigation control problem, although the spacings are arbitrarily given, they should be informed to each robot before the beginning of the circumnavigation process. Therefore, a central robot, for example, needs to transmit the values to each robot. This means any inconsistency in the values received by other robots will lead to a failure in forming the correct formation. In addition, if the desired spacings do not satisfy Eq. 8 accidentally (e.g. the central robots calculate the wrong desired spacings), then robots will not form the expected spacings. Moreover, once there are robots joining into or quitting from the formation, the desired spacings should be calculated (to satisfy Eq. 8) and transmitted again, which requires the central robot to be aware of the exact number of robots in the formation. Therefore, the control algorithm relies on the stability of the central robot and the accurate information of the number of robots. To cope with these insufficiencies, the expected spacings will be dynamic as discussed later.

5.2 Problem Formulation

Before giving the definition of the dynamic circumnavigation control problem, we propose the concept of *utility*. Let $\mathbb{R}_{\geq 0}$ denotes the field of non-negative real numbers.

Definition 2 (**Utility**) In a heterogeneous multi-robot system, given different kinds of robots, a robot's utility $\mu(t) \in \mathbb{R}_{\geq 0}$ is determined by a given criterion (such as its maximum movement speed). The utility reflects the weight of the robot in the circumnavigation process at time t.

For example, suppose a robot's maximum movement speed is the criterion. Let $\mu_i(t) = \frac{v_{mi}(t)}{v_M}, i = 1, \ldots, n$, where $v_{mi}(t)$ is the maximum movement speed of r_i and v_M is the greatest movement speed in the heterogeneous multi-robot system. Then $\mu_i(t) \in [0, \; 1], i = 1, \ldots, n$. When $\mu_i(t) = 0$, the robot r_i cannot continue the circumnavigation process with other robots. In this case, its neighbouring robots will neglect its role in the circumnavigation process. $\mu_i(t)$ will increase or decrease due to the enhancement or damage of the robot's locomotion capabilities. To explain directly how utilities are used to enable dynamic spacings among robots, we simply regard the utility of a robot to be proportional to its maximum movement speed. For simplicity of writing, the symbol t is neglected from μ unless it causes confusion. The dynamic circumnavigation control problem based on utilities is defined as follows:

Definition 3 (**Dynamic Circumnavigation Control Problem**) In a heterogeneous multi-robot system composed of n $(n \geq 2)$ mobile robots, of which the dynamics are modeled by Eq. 2, when $\lim\limits_{t\to\infty} f_i(t, \mu_1, \ldots, \mu_n)$ exists, the dynamic circumnavigation control problem based on utilities is to seek control laws satisfying the following asymptotic conditions:

$$\lim_{t\to\infty} \rho_i(t) = \rho^*, \tag{48}$$

$$\lim_{t\to\infty} \Delta_i(t) = \lim_{t\to\infty} f_i, \tag{49}$$

$$\lim_{t\to\infty} \dot{\varphi}_i(t) = \omega^*, \tag{50}$$

$$\lim_{t\to\infty} z_i(t) = z^*, \tag{51}$$

for $i = 1, ..., n$. Here, $\mu_i > 0$, $\rho^* > 0$, $\omega^* > 0$ and $z^* \in \mathbb{R}$. ρ^*, ω^* and z^* denote the circumnavigation radius, the angular speed and the circumnavigation height respectively. $f_i \triangleq f_i(t, \mu_1, \ldots, \mu_n) : \mathbb{R}^{n+1} \to \mathbb{R}$ is a function of time and the utilities of robots. It maps utilities to the final formation spacings. It is bounded and piecewise continuous.

In fact, in this paper, it is required that all robots and the target remain in the same plane in the end. Therefore, the default value of z^* is 0. Equation 49 manifests that the final formation spacings are not specified manually, but instead, it is determined by the f_i function, which will be referred to as f function for simplicity. The advantage of eliciting the f function is that the spacings among robots can be dynamically adjusted corresponding to the variations of robots' utilities.

The expression of the f function is determined by a *formation guideline*. It is proposed under specific physics background representing the relationship between the utilities of robots and the expected formation spacings. In this paper, we suppose that multiple heterogeneous robots circumnavigate a target and try to prevent it from fleeing. In Fig. 3, four robots $r_1, \ldots, r_4$ rotate around a target

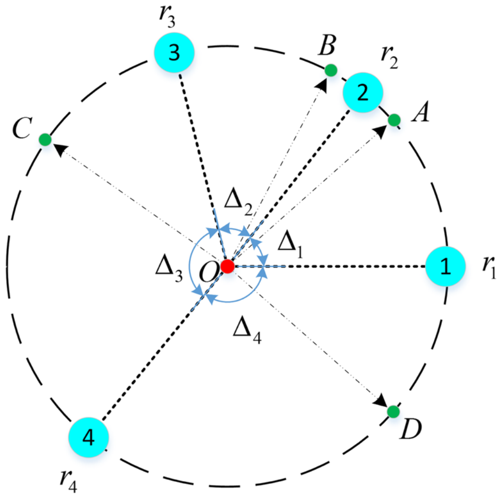


Fig. 3 The physics background of the formation guidelines

denoted by O. Suppose that the target is intelligent enough to determine the *best fleeing points* denoted by A, B, C and D in the figure. Obviously, the best fleeing points are related to the utilities (i.e., the maximum movement speeds) of robots. The position of A, for instance, is calculated by $\angle AOr_2 = \frac{\mu_2}{\mu_1+\mu_2}$. We also suppose that the probability of capturing the target by a robot is inversely proportional to the time spent on moving from its initial position along the circular trajectory at its maximum speed to the best fleeing point. Therefore, the first formation guideline can be defined as below:

Formation Guideline 1 *In the final circumnavigation formation formed by robots, when the target tries to escape via any of the best fleeing point, the two robots adjacent to the best fleeing point have the same probability of capturing the target.*

To understand the above formation guideline, taking Fig. 3 for example, it means the traveling time for r_1 and r_2 to arrive at the best fleeing point A along the circular trajectory at their maximum speeds (i.e., μ_1 and μ_2 resp.) is the same, or the traveling time for r_2 to arrive at A or B along the circular trajectory at its maximum speed (i.e., μ_2) is identical, and hence, the probability of capturing the target is equal. Following this, it can be derived that for $i = 1, \ldots, n$, $\frac{\mu_i}{\mu_i+\mu_{i^+}}\Delta_i = \frac{\mu_i}{\mu_i+\mu_{i^-}}\Delta_{i^-}$. According to this equation, the relationship between the final expected formation spacings and the utilities is

$$\Delta_1 : \Delta_2 : \cdots : \Delta_n = (\mu_1+\mu_2) : (\mu_2+\mu_3) : \cdots : (\mu_n+\mu_1). \tag{52}$$

Therefore, given $\mu_1, \ldots, \mu_n$, by Eqs. 8 and 52 the formation spacings can be determined, and the f function is expressed as follows:

$$f_i(t, \mu_1, \ldots, \mu_n) = \frac{\mu_i + \mu_{i^+}}{\sum_{i=1}^{n} \mu_i}\pi. \tag{53}$$

Similarly, another formation guideline can be defined as follows:

Formation Guideline 2 *In the final circumnavigation formation formed by robots, the spacings among robots are proportional to their utilities.*

Formation Guideline 2 can be interpreted in this way: as the utility of a robot increases, its complementary effects on its neighbouring robots increase, and therefore the spacings between them should be increased. Taking Fig. 3 for example, the formation guideline indicates that, for $i = 1, \ldots, n$,

$$\frac{\Delta_i}{\mu_i} = \frac{\Delta_{i^-}}{\mu_{i^-}} \tag{54}$$

Accordingly, given $\mu_1, \ldots, \mu_n$, the final desired formation spacings are determined by Eqs. 8 and 54 jointly. Hence the f function is

$$f_i(t, \mu_1, \ldots, \mu_n) = \frac{2\mu_i}{\sum_{i=1}^{n} \mu_i}\pi. \tag{55}$$

In fact, according to different situations, formation guidelines can be designed manually. The guidelines mentioned above are only two possible examples. For better understanding of the above concepts proposed in this paper, the relationship between the f function, the formation guidelines and the utilities is explained as follows. The formation guideline determines the explicit form of the f function, and furthermore, the f function maps the utilities to the final expected spacings. Therefore, given the utilities of all robots, under a specific formation guideline, the f function and hence the final spacings can be determined. However, it should be noted that the knowledge of the final spacings does not play any instrumental role during the circumnavigation process. Instead, the actual spacings converge to the final spacings in a distributed manner merely depending on the exchange of the utilities information among robots. In other words, in a multi-robot system, robots are not aware of what the final expected spacings are; the actual spacings among robots adapt dynamically to the variations of the local utilities of neighboring robots.

5.3 Control Algorithm

Theorem 4 *Consider a multi-robot system with robot dynamics described by* Eq. 2, *by introducing the control input* $v_i = \dot{q}_i = (\dot{\rho}_i, \dot{\varphi}_i, \dot{z}_i)^T$ *into* Eq. 7, *where* $\dot{\rho}_i$ *and* $\dot{z}_i$ *are given by* Eqs. 14 and 15 *respectively, and* $\dot{\varphi}_i$ *is as follows:*

$$\dot{\varphi}_i = \omega^* + k_\varphi(\widetilde{\varphi}_i - \varphi_i), \tag{56}$$

where

$$\widetilde{\varphi}_i = \begin{cases} \varphi_{i^-} + \dfrac{\mu_{i^-} + \mu_i}{\mu_{i^+} + 2\mu_i + \mu_{i^-}}(\Delta_i + \Delta_{i^-}),\ i = 2, 3, \ldots, n, \\ \varphi_{i^-} + \dfrac{\mu_{i^-} + \mu_i}{\mu_{i^+} + 2\mu_i + \mu_{i^-}}(\Delta_i + \Delta_{i^-}) - 2\pi,\ i = 1, \end{cases} \tag{57}$$

and μ_i *is the utility of the robot* r_i, *which is piecewise constant. If the* f *function is shown as* Eq. 53 *(Formation Guideline 1), the dynamic circumnavigation control problem based on utilities encoded by* Eqs. 48, 49, 50 *and* 51 *can be solved.*

Proof See Appendix. □

Remark 6 If $\mu_1+\mu_2 = \cdots = \mu_{n-1}+\mu_n = \mu_n+\mu_1$, Eq. 53 becomes $f_i = 2\pi/n$, that is, the spacings among robots are equal. Then Eq. 57 becomes

$$\begin{cases} \widetilde{\varphi}_1 = \frac{\varphi_n+\varphi_2-2\pi}{2}, \\ \widetilde{\varphi}_i = \frac{\varphi_{i-1}+\varphi_{i+1}}{2}, \ i = 2, ..., n-1, \\ \widetilde{\varphi}_n = \frac{\varphi_{n-1}+\varphi_1+2\pi}{2}. \end{cases} \tag{58}$$

Combined with Eqs. 14, 15 and 56, they can solve the dynamic circumnavigation control problem with equal spacings.

Remark 7 It is interesting to note that when there are only two robots, i.e., $n = 2$, the subscripts of robots satisfy $i^- = i^+$. Equation 57 is simplified to $\widetilde{\varphi}_1 = \varphi_2 - \pi$, $\widetilde{\varphi}_2 = \varphi_1 + \pi$, and Eq. 53 becomes $f_i = \pi$, $i = 1, 2$. This indicates that the formation spacings are fixed no matter how robots' utilities change (except 0); these two robots will always position on the ends of the diameter of the circular trajectory. Furthermore, A and b degrade to $A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ and $b = \begin{bmatrix} -\pi \\ \pi \end{bmatrix}$ respectively.

Remark 8 In the definition of the dynamic circumnavigation control problem based on utilities, Eq. 49 contains the utilities of all robots. However, it can be seen from Eq. 57 that each robot only needs to obtain the utilities of its two neighbouring robots. In addition, when a robot joins or leaves the formation, according to Eqs. 56 and 57, the spacings among robots will adjust dynamically through local update of the utilities of neighbouring robots. To sum up, the dynamic circumnavigation control algorithm does not rely on the number of robots, and it is able to dynamically adjust the formation spacings dependent on the change of utilities. The control algorithm is distributed, and in this way, it achieves the global aim described by Eq. 49.

Note that when $\mu_\theta = 0$, the robot r_θ has quited from the circumnavigation process, and therefore the communication topology has changed. The change of communication topology means the indices of the neighbouring robots alter accordingly. When $\mu_2 = 0$, for example, the neighbouring robots of r_3 change from r_2 and r_4 to r_1 and r_4. In this way, the circumnavigation control algorithm based on utilities can well adapt to the cases where there are local variations on utilities or where robots join or quit from the formation. The formation spacings can be adjusted dynamically based on the selected formation guideline, achieving distributed formation reconfiguration.

Similarly, under Formation Guideline 2, the dynamic circumnavigation control problem based on utilities can also be solved.

Proposition 1 *Consider a multi-robot system with robot dynamics described by* Eq. 2*, by introducing the control input* $v_i = \dot{q}_i = (\dot{\rho}_i, \dot{\varphi}_i, \dot{z}_i)^T$ *into* Eq. 7*, where* $\dot{\rho}_i$ *and* $\dot{z}_i$ *are given by* Eqs. 14 *and* 15 *respectively, and* $\dot{\varphi}_i$ *is given by*

$$\dot{\varphi}_i = \omega^* + k_\varphi(\hat{\varphi}_i - \varphi_i), \tag{59}$$

where

$$\hat{\varphi}_i = \begin{cases} \varphi_{i^-} + \dfrac{\Delta_i + \Delta_{i^-}}{\mu_i + \mu_{i^-}}\mu_{i^-}, & i = 2, 3, \ldots, n, \\ \varphi_{i^-} + \dfrac{\Delta_i + \Delta_{i^-}}{\mu_i + \mu_{i^-}}\mu_{i^-} - 2\pi, & i = 1, \end{cases} \tag{60}$$

and $\mu_i, i = 1, \ldots, n$*, is the utility of the robot* r_i *and it is piecewise constant. If the* f *function is shown as* Eq. 55 *(Formation Guideline 2), the dynamic circumnavigation control problem based on utilities encoded by* Eqs. 48, 49, 50 *and* 51 *can be solved with exponential convergence speed.*

Proof The proof is similar to that of Theorem 4, so it is omitted here. □

It should be noted that Theorem 4 also applies to the circumnavigation process under Formation Guideline 1 and 2.

6 Circumnavigation Control with External Perturbations

In practice, external perturbations are inevitable. These perturbations can come from sensor measurement noise, environmental disturbances, etc. In this section, these perturbations are considered in a general form, denoted by $d = \begin{bmatrix} d_1 & \ldots & d_n \end{bmatrix}^T$. It is reasonable to assume that the perturbations are bounded in the sense that

$$||d||_1 < d_M, \tag{61}$$

where $d_M > 0$. Regardless of the specific form of the perturbations, they are introduced into the cylindrical control input as an additional additive term, namely $v_i = \dot{q}_i - d_i$. For simplicity, the new control algorithm to reduce the effect of external perturbations is derived for the static circumnavigation control problem. However, it turns out to be very similar for the dynamic circumnavigation control problem, and thus it is omitted here. First we define the sgn function as: if $x > 0$, $\text{sgn}(x) = 1$; if $x < 0$, $\text{sgn}(x) = -1$; if $x = 0$, $\text{sgn}(x) = 0$.

Theorem 5 *Consider a multi-robot system with robot dynamics described by* Eq. 2*, by introducing the control input* $v_i = \dot{q}_i = (\dot{\rho}_i, \dot{\varphi}_i, \dot{z}_i)^T$ *into* Eq. 7*, where*

$$\dot{\rho}_i = k_\rho(\rho^* - \rho_i) - k_1 sgn(\rho^* - \rho_i) - d_i, \tag{62}$$

$$\dot{z}_i = -k_z z_i - k_2 sgn(-z_i) - d_i, \tag{63}$$

$$\dot{\varphi}_i = \omega^* + k_\varphi(\bar{\varphi}_i - \varphi_i) - k_3 sgn(\bar{\varphi}_i - \varphi_i) - d_i, \tag{64}$$

where d_i is the external perturbation satisfying Eq. 61*, and $k_i < -2d_M, i = 1, 2, 3$. k_ρ, k_z and k_φ are positive gains, and $\bar{\varphi}_i$ is shown in* Eq. 17*. Then the static circumnavigation control problem with arbitrary spacings can be almost solved in the sense that* Eqs. 9, 10 *and* 12 *are satisfied asymptotically.*

Proof Denote $\rho = \begin{bmatrix} \rho_1 & \dots & \rho_n \end{bmatrix}^T$ and $e_\rho = \rho^*\mathbf{1} - \rho$. We first prove that $\rho \rightarrow \rho^*\mathbf{1}$. Define a Lyapunov function candidate as $V(e_\rho) = \frac{1}{2}e_\rho^T e_\rho$. Take the derivative of V with respect to time we can obtain:

$$\begin{aligned} \dot{V}(e_\rho) &= -e_\rho^T \dot{\rho} \\ &= -k_\rho e_\rho^T e_\rho + k_1 e_\rho^T \text{sgn}(e_\rho) + e_\rho^T d \\ &\leq -k_\rho e_\rho^T e_\rho + (k_1 + ||d||_1)||e_\rho||_1 \\ &\leq -k_\rho e_\rho^T e_\rho - d_M ||e_\rho||_1 \leq 0. \end{aligned} \tag{$\triangle$}$$

The equality is taken when $e_\rho = 0$. Therefore, it suffices to prove that $\rho \rightarrow \rho^*\mathbf{1}$. The procedure to prove that $z_i \rightarrow 0$ is the same and thus it is omitted here. Next we show that $\Delta_i \rightarrow \Delta_i^d$. We use the same notations e_φ, P, Q, L_p as in Section 4.3. $\lambda_\star = \lambda_M(Q)$ denotes the largest eigenvalue of Q, which is positive. Therefore, $\dot{e}_\varphi = -L_p\dot{\varphi} = -k_\varphi L_p e_\varphi + k_3 L_p \text{sgn}(e_\varphi) + L_p d$. Define a Lyapunov function candidate as $V(e_\varphi) = e_\varphi^T P e_\varphi$, so it follows that

$$\begin{aligned} \dot{V}(e_\varphi) &= \dot{e}_\varphi^T P e_\varphi + e_\varphi^T P \dot{e}_\varphi \\ &\overset{(*)}{=} -k_\varphi e_\varphi^T Q e_\varphi + k_3 \text{sgn}^T(e_\varphi) Q e_\varphi + d^T Q e_\varphi \\ &\overset{(**)}{\leq} -k_\varphi e_\varphi^T Q e_\varphi + k_3 n(\lambda_\star + 1)||e_\varphi||_1 \\ &\quad + (n(\lambda_\star + 1) + 1)||d||_1 ||e_\varphi||_1 \\ &\leq -k_\varphi e_\varphi^T Q e_\varphi - (n\lambda_\star + n - 1) d_M ||e_\varphi||_1 \leq 0, \end{aligned} \tag{$\triangledown$}$$

where the equality is taken when $e_\varphi = 0$. Therefore, it suffices to prove that $\varphi \rightarrow \bar{\varphi}$, and following the discussion in the previous sections, it follows that $\Delta_i \rightarrow \Delta_i^d$. Note that $(*)$ is due to the fact that $PL_p = L_p^T P$ is symmetric as shown in Eq. 32. So, for example, $\text{sgn}^T(e_\varphi) L_p^T P e_\varphi + e_\varphi^T P L_p \text{sgn}(e_\varphi) = \text{sgn}^T(e_\varphi)(L_p^T P + P L_p) e_\varphi = \text{sgn}^T(e_\varphi) Q e_\varphi$. $(**)$ is due to the following inequality:

$$\begin{aligned} \text{sgn}^T(e_\varphi) Q e_\varphi &= \text{sgn}^T(e_\varphi)(Q + I)e_\varphi - \text{sgn}^T(e_\varphi) e_\varphi \\ &\overset{(\dagger)}{\leq} \sqrt{e_\varphi^T (Q + I) e_\varphi} \\ &\quad \times \sqrt{\text{sgn}^T(e_\varphi)(Q + I)\text{sgn}(e_\varphi)} - ||e - \varphi||_1 \\ &\leq \lambda_M(Q + I)||\text{sgn}(e_\varphi)||_2 ||e_\varphi||_2 - ||e_\varphi||_1 \\ &\overset{(\ddagger)}{\leq} (\lambda_\star + 1)\sqrt{n}\sqrt{n}||e_\varphi||_1 - ||e_\varphi||_1 \\ &\leq n(\lambda_\star + 1)||e_\varphi||_1. \end{aligned}$$

Note that $(\dagger)$ is due to the Hölder's inequality (since $x^T(Q + I)y$ is the inner product) [6]. $(\ddagger)$ is justified by the equivalence between the first and the second norm and the fact that $||\text{sgn}(e_\varphi)||_2 \leq \sqrt{n}$. Similarly, $d^T Q e_\varphi = d^T(Q + I)e_\varphi - d^T e_\varphi \leq \lambda_M(Q + I)||d||_2 ||e_\varphi||_2 + ||d||_1 ||e_\varphi||_1 \leq (n(\lambda_\star + 1) + 1)||d||_1 ||e_\varphi||_1$. □

Remark 9 Using the new control algorithm, only Eqs. 9, 10 and 12 are satisfied asymptotically. In fact, $\dot{\varphi}_i \rightarrow (w^* - d_i)$, which is directly affected by the external perturbation. However, assume that $w^* \gg ||d||$, the effect of the perturbation is negligible. In addition, the less conservative choices of $k_i, i = 1, 2, 3$ are $k_i < -d_M, i = 1, 2$ and $k_3 < -(1 + \frac{1}{n(\lambda_\star + 1)})d_M$, which can be easily derived from Eqs. ($\triangle$) and ($\triangledown$).

Remark 10 In order for Eq. 11 to be satisfied as well, a disturbance observer can be designed to eliminate the disturbance from the control input [2]. Suppose the disturbance is generated by a linear exogenous system

$$\begin{aligned} \dot{\xi} &= A\xi \\ d &= C\xi, \end{aligned} \tag{65}$$

where A and C are n by n matrices and $\xi \in \mathbb{R}^n$. It is assumed that the system described by Eq. 65 is neutral stable and (A, C) is observable. The control algorithm is then designed as

$$\dot{\varphi} = w^*\mathbf{1} + k_\varphi(\bar{\varphi} - \varphi) + \hat{d} - d,$$

where $\hat{d}$ is the estimated value of the disturbance d. $\hat{d}$ is generated by the following disturbance observer:

$$\begin{aligned} \dot{z} &= (A - LC)(z + L\varphi) - L(w^*\mathbf{1} + k_\varphi(\bar{\varphi} - \varphi) - \hat{d}) \\ \hat{\xi} &= z + L\varphi \\ \hat{d} &= C\hat{\xi} \end{aligned}$$

where L is chosen such that $A - LC$ is Hurwitz.

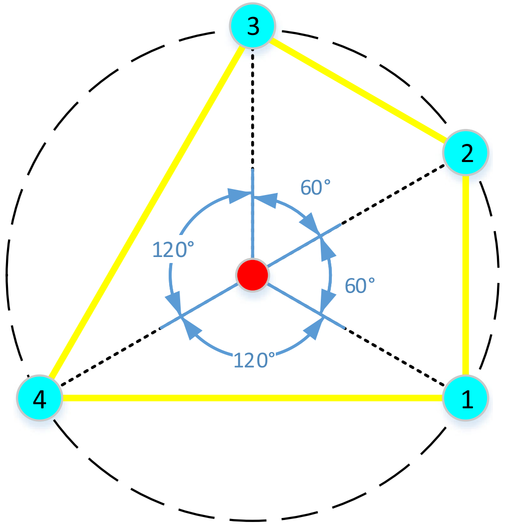


Fig. 4 The formation described by the desired spacings $\Delta^d = \left[\frac{\pi}{3}\ \frac{\pi}{3}\ \frac{2\pi}{3}\ \frac{2\pi}{3}\right]^T$

7 Simulation and Experiment Results and Analysis

This section presents the simulation and experiment results for both static and dynamic circumnavigation problems, which further validate the theory presented above.

7.1 Results for Static Circumnavigation

7.1.1 Simulation with Simulink

To validate the effectiveness of the control law proposed in this paper, we carry out two simulations with Simulink, taking into account the possible effects of different parameters (Table 1). Different from most of the previous studies, robots and the target move in the three dimensional space instead of the two dimensional space. The desired spacings of these two simulations are the same, as shown in Fig. 4. For better explanation and intuitive demonstration, we use a line to connect the centers of robots at the same time and the resultant geometry is called the *formation shape*.

In view of Eq. 7, 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 simulations, 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 Figs. 5 and 6 respectively. The red, green and blue lines connecting four robots indicate the formation shapes at the beginning, the middle and the end of the circumnavigation process. The dashed lines originating from robots represent their trajectories. It can be seen that the circumnavigation error signals converge to zero exponentially, and the desired spacings among robots can be finally achieved. Although the two simulations 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 spacings can be specified arbitrarily as long as they satisfy Eq. 8.

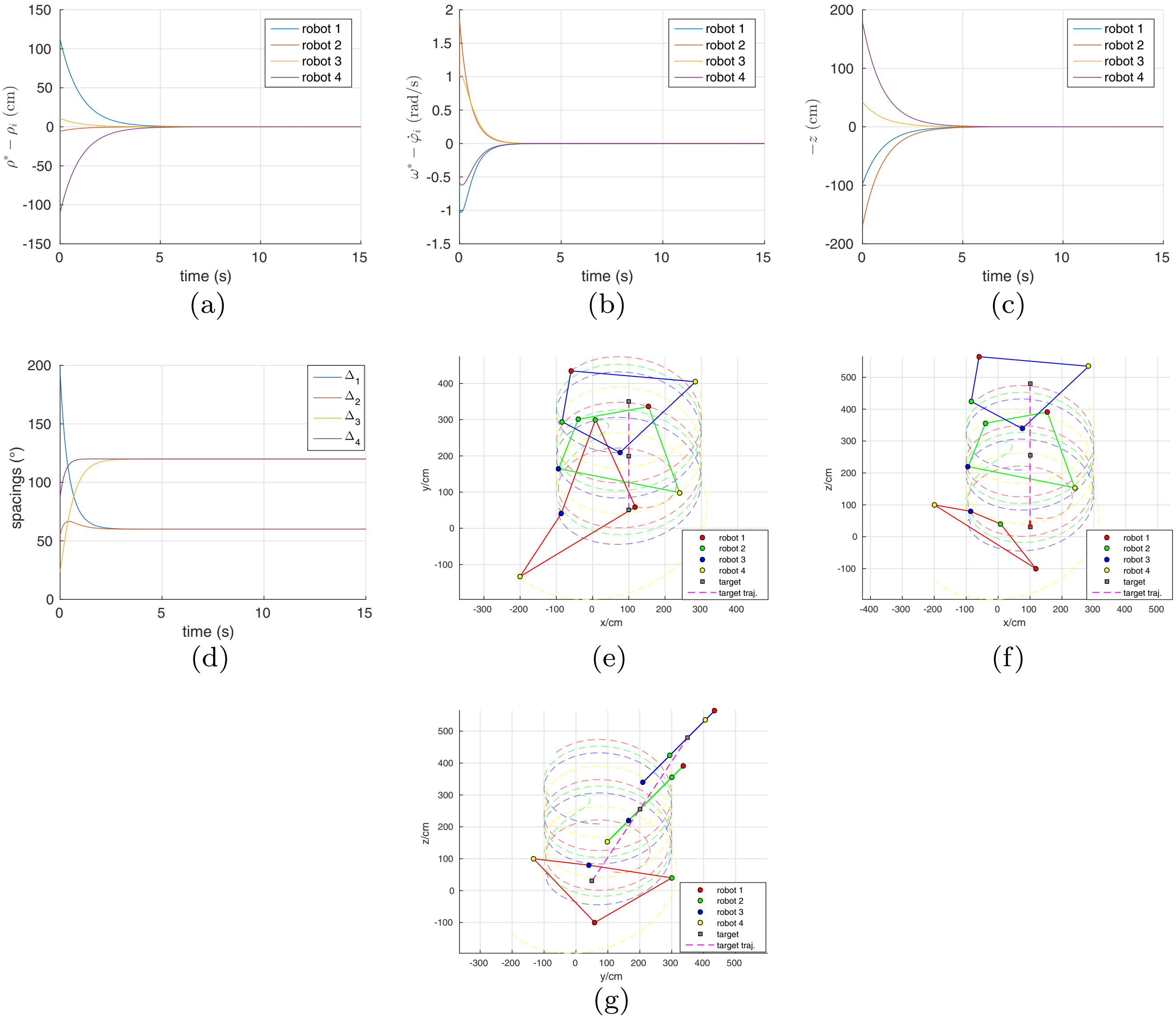
7.1.2 Experiment with Soccer-Playing Robots

The hardware platforms used in this experiment are four soccer-playing robots [3, 19]. Since they have omnidirectional movement abilities (with custom-made omnidirectional wheels), and they can reach any given valid velocity almost instantly (due to the commercial motor controller boards), their dynamics can be regarded as the first-integrator model given in Eq. 2. In addition, an omnidirectional vision system is equipped on each robot with algorithms for self-localization and the recognition of a yellow football [8]. The position and velocity of the robot itself and those of the football are obtained by their

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

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

In this table, the length unit, angle unit and time unit are *cm*, *rad* and *s* respectively. $\dot{p}_b$ is the target's velocity, which is expressed in the world reference frame $\mathcal{W}$. $R_b(0)$ is the initial orientation of the body reference frame $\mathcal{B}$. $Rot(X, \pi/4)$ means rotating the coordinate frame's X-axis by $\pi/4$. ω_b is the angular velocity vector, of which the components represent angular velocity about the X-, Y- and Z-axis of frame $\mathcal{B}$ respectively



(a) (b) (c) (d) (e) (f) (g)

Fig. 5 Simulation I for the static circumnavigation control problem. **a–c** are the error signals which represent the curves for $\rho^* - \rho_i$, $\omega^* - \dot{\varphi}_i$ and $-z$ respectively. **d** plots the spacings among robots; **e–g** are robots' trajectories projected on the plane XSY, XSZ and YSZ respectively

own on-board omnidirectional vision systems. Moreover, robots are only allowed to receive information from its neighbouring robots. The desired spacings among robots are the same as those specified in the above simulation. Specifically, the target (football) is dragged by a Turtlebot.[1] In this experiment, the circumnavigation radius is 200 cm, the angular speed is 0.5 rad/s, $k_\rho = 2$ and $k_\varphi = 2.5$.

The experiment is illustrated in Fig. 7 and the corresponding data plots are shown in Fig. 8. The pink dots in Fig. 8d are the perceived positions of the target by Robot 1 (for simplicity, perceived positions by other robots are not plotted). This shows that there is much noise existing in the perceived positions of the target, which affects the performance of the circumnavigation process. From Fig. 8 it can be seen that despite the information noise, the curves for circumnavigation radius, angular speed and spacings only fluctuate around the desired values lightly. Considering that there are measurement errors of the football position (maximum error 20 cm) and self-localization (maximum error 20 cm), and that the robot's kinematics model is only an approximation of the first-integer model, the errors of the circumnavigation radius, angular speed and spacings are

[1] http://www.turtlebot.com/

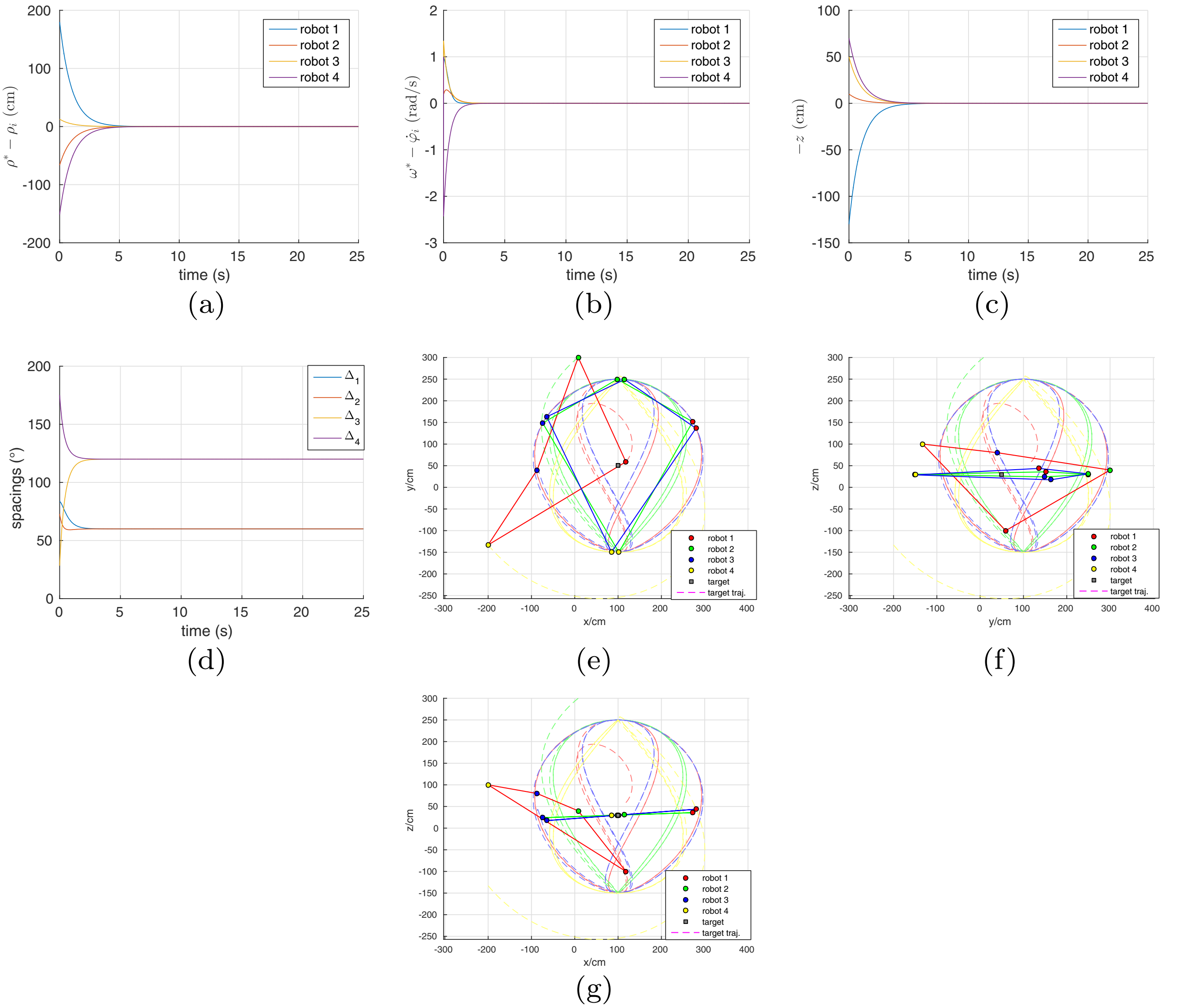


Fig. 6 Simulation II for the static circumnavigation control problem. **a–c** are the error signals which represent the curves for $\rho^* - \rho_i$, $\omega^* - \dot{\varphi}_i$ and $-z$ respectively. **d** plots the spacings among robots; **e–g** are robots' trajectories projected on the plane XSY; XSZ and YSZ respectively

acceptable. Therefore, the control algorithm for the static circumnavigation control problem is effective, which means the multi-robot system is able to realize the circumnavigation process with arbitrary spacings. Note that due to the limitation of the experiment platform, only ground vehicles are used. However, the control algorithm is also applicable for the three-dimensional case.

7.2 Results for Dynamic Circumnavigation

Although it is claimed that formation guidelines correspond to specific physics backgrounds, in the experiment, we do not try to reproduce the specific scenarios. This is because the emphasis here is the stability of the control algorithm, and how the global formation spacings react dynamically to the variation of the utilities. In the experiments, robots' utilities and the variation of the utilities are manually specified. Readers can think of an increase in the utilities as improvement of robots' locomotion capabilities, while the decrease means the deterioration of performance due to the worn-out or damage of robots. Note that formation guidelines only reflect the relationship between the utilities of robots and the final formation spacings; it does not determine the utilities of robots. The first experiment is

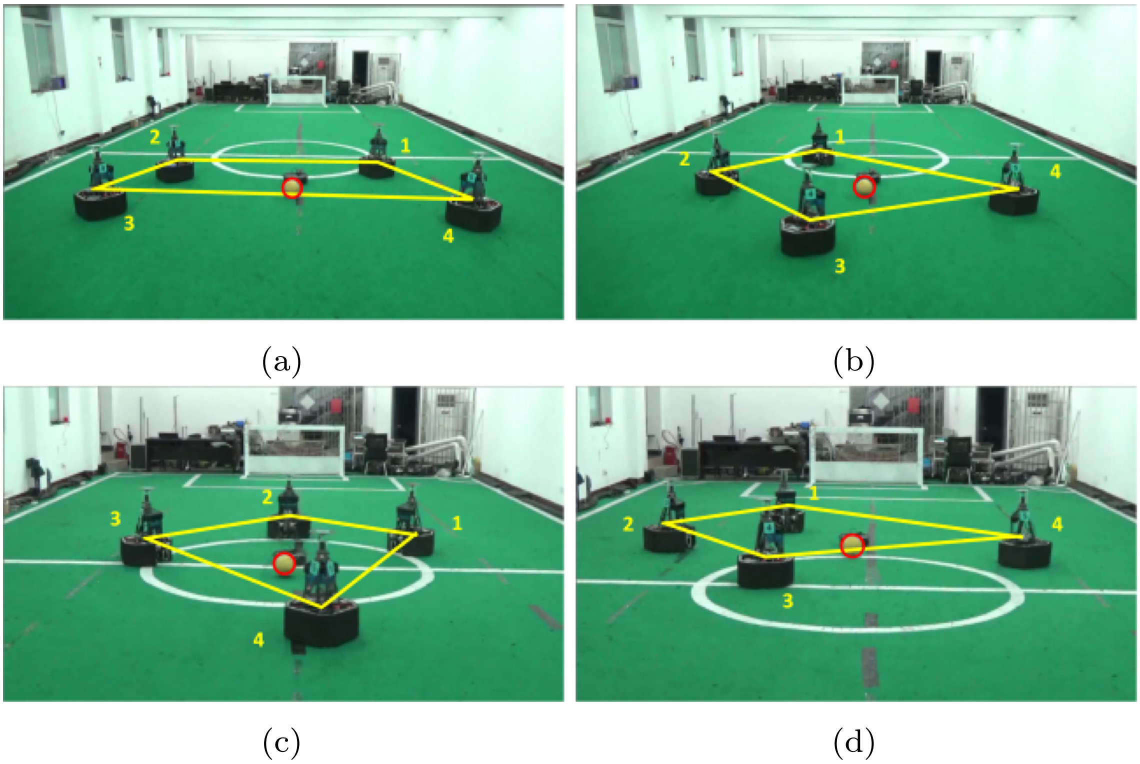
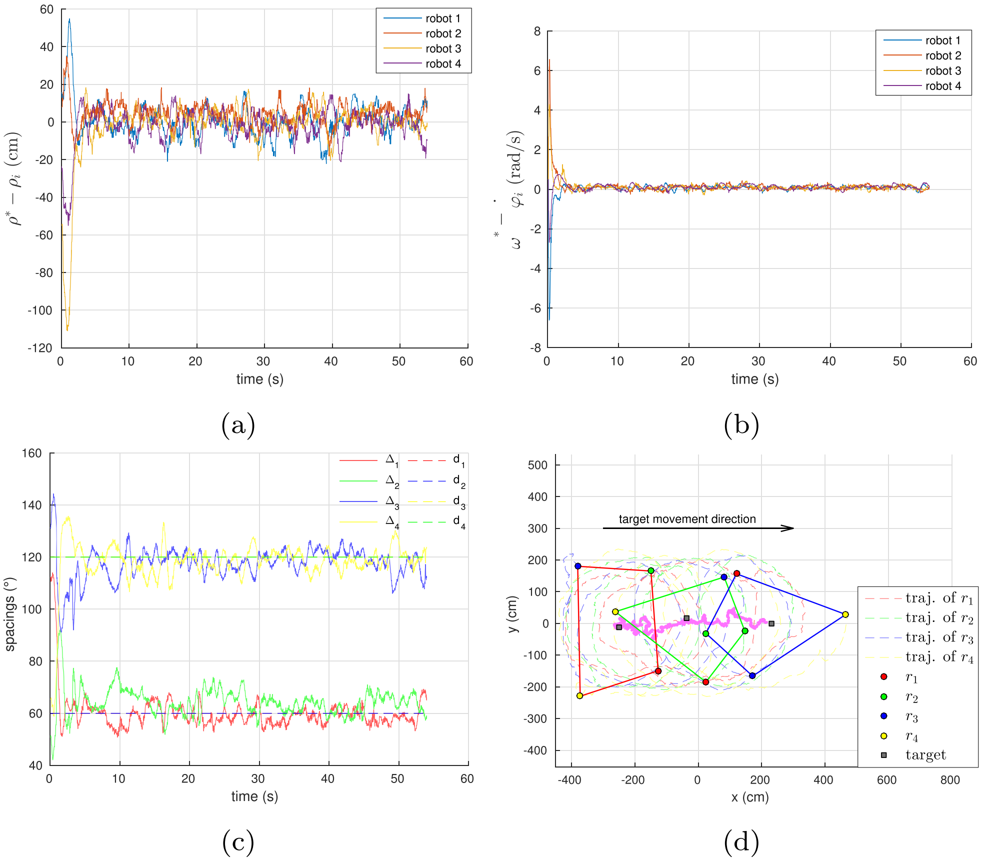


Fig. 7 Real robot experiment for the static circumnavigation control problem. The yellow lines connecting thev centers of robots indicate the formation shapes intuitively. The football in the middle of the field is the target to be encircled, which is marked by a red circle. The football is dragged by a Turtlebot, so its positions are varying during the process. **a–d** show the robots' positions at 0, 5, 32 and 51 second respectively

Fig. 8 Data plots for the real-robot experiment. **a–c** are the plots for $\rho^* - \rho_i, \omega^* - \dot{\varphi}_i$ and the spacings respectively. **d** shows the robots' trajectories on the XSY plane. The red, green and blue lines connecting four robots indicate the formation shapes at 0, 30 and 54 second respectively. The dashed lines originating from robots represent their trajectories



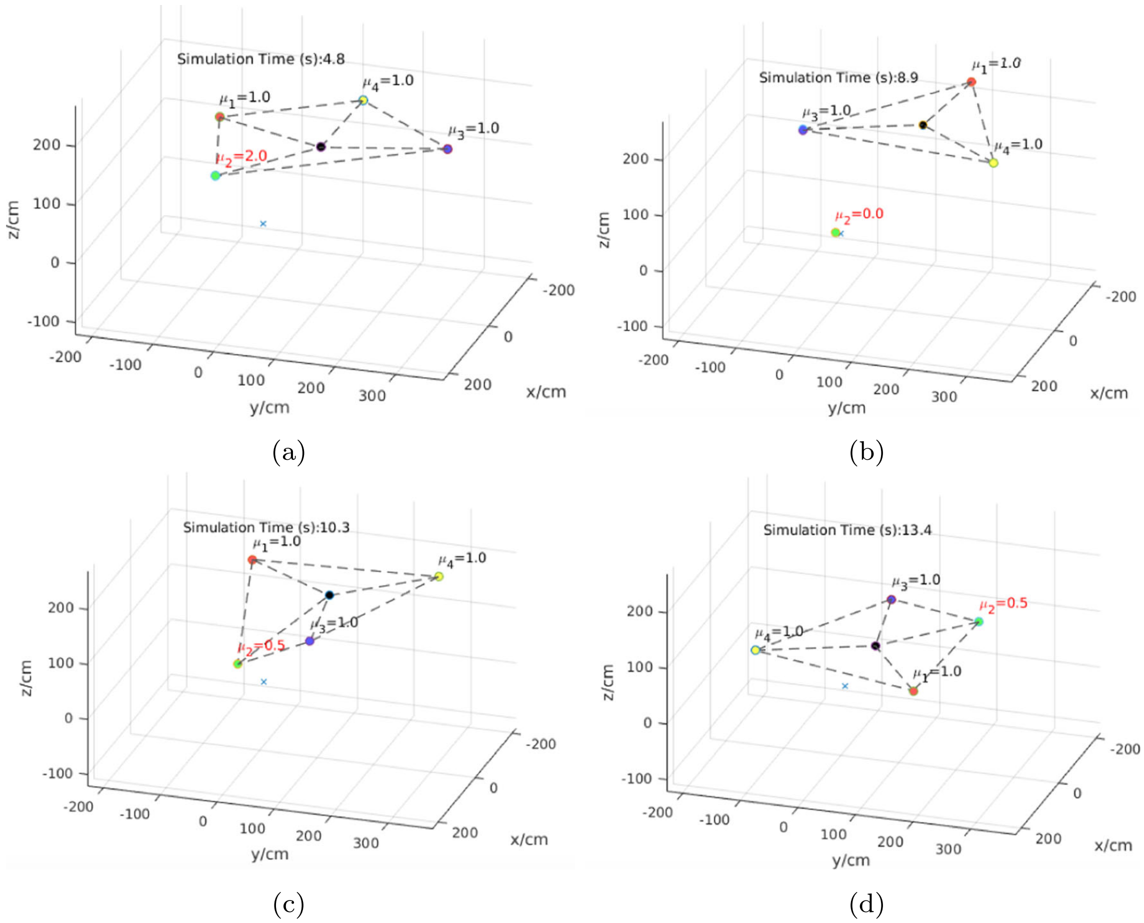
a simulation with Simulink and the other is a real-robot experiment using soccer-playing robots.

7.2.1 Simulation with Simulink

In this simulation, four robots are used and Formation Guideline 2 is adopted. For demonstrating the dynamic change in formation spacings when robots' utilities vary, the utilities of the robot r_1, r_3 and r_4 remain 1 throughout the whole circumnavigation process, while the utility of the robot r_2 varies according to a piecewise constant function. That is $\mu_2 = 2,\ 0 \leq t < 5;\ \mu_2 = 0,\ 5 \leq t < 10;\ \mu_2 = 0.5,\ t \geq 10$. For convenience, the three time ranges are denoted by Stage 1, 2 and 3 respectively. Note that at Stage 2, the utility of the robot r_2 is zero, which means it does not continue the circumnavigation process and it will retreat to a corner. In this simulation, robots' initial positions are randomly chosen. Simulation parameters are $\rho^* = 2$ m, $w^* = 1$ rad/s and $k_\varphi = k_\rho = k_z = 2$.

The circumnavigation process is illustrated in Fig. 9. The black dot in the center is the target to be encircled. Dashed lines connecting four robots represented by red, green, blue and yellow dots indicate the formation shape and the communication topologies intuitively. Dashed lines connecting the target and the robots demonstrate the circumnavigation radii. At Stage 1, four robots gradually form a stable formation and circumnavigate the target (see Fig. 9a). Then r_2 leaves the formation at Stage 2 (see Fig. 9b). At the beginning of Stage 3, r_2 joins the formation (see Fig. 9c). Note that the communication topologies have changed from Stage 1 to Stage 2 and from Stage 2 to Stage 3. Four robots form a stable formation again with different spacings in comparison with Stage 1 in the end (see Fig. 9d).

The data plots of the simulation are shown in Fig. 10. Since r_2 quits from the formation during Stage 2, the corresponding data is not plotted. It can be seen from Fig. 10 that although the changes of the utility of r_2 lead to the deviation of curves from the expected values, the circumnavigation error signals converge to zero exponentially during the three stages (see Fig. 10a, b and c resp.). The spacings among robots change according to the variation of the utilities. However, the spacings converge to



(a)

(b)

(c)

(d)

Fig. 9 The simulation with Simulink. **a**–**d** show the circumnavigation process at 4.8 s (Stage 1), 8.9 s (Stage 2), 10.3 s (Stage 3) and 13.4 s (Stage 3) respectively

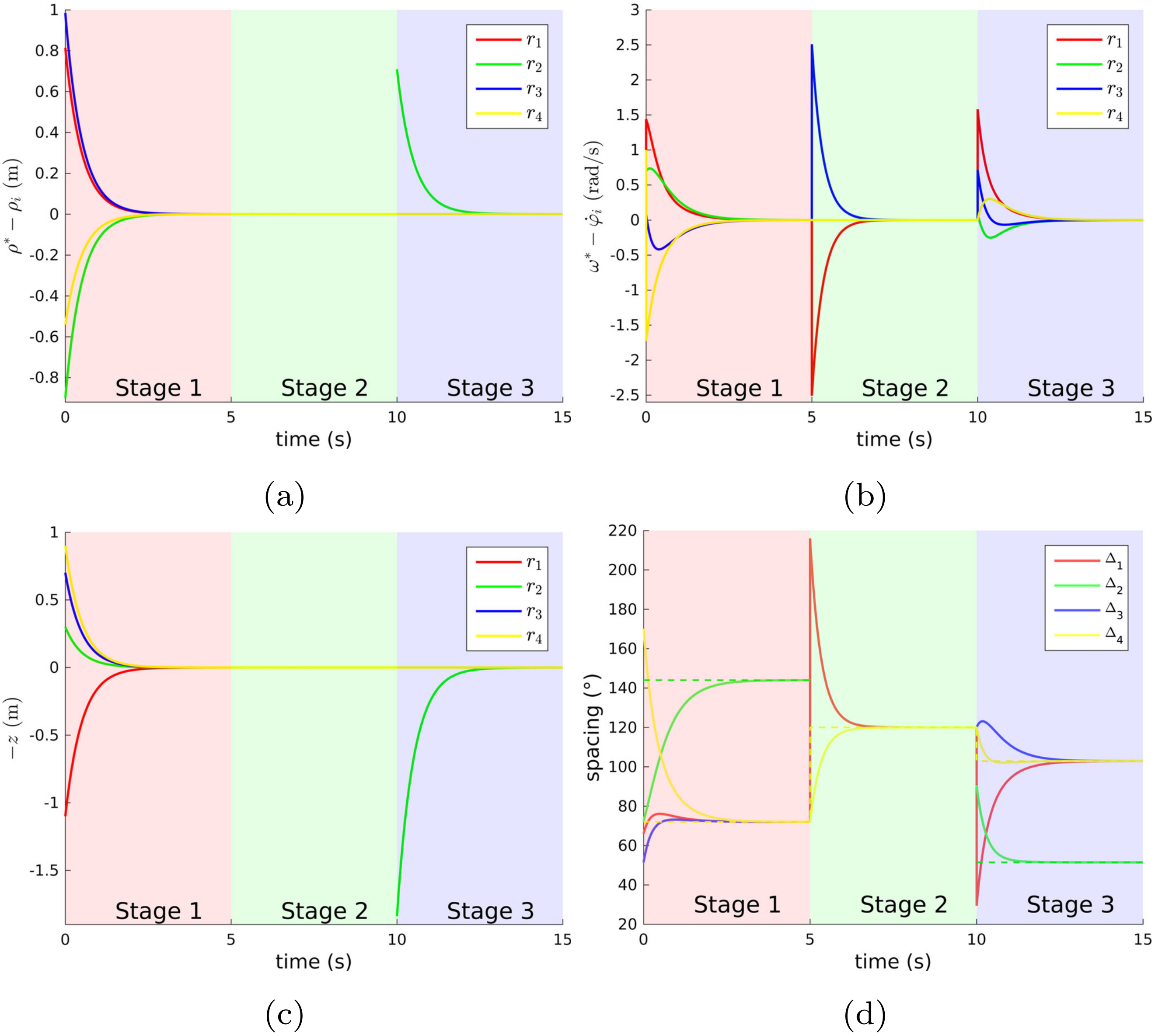


Fig. 10 The data plots of the simulation. **a**–**c** show the error signals of the circumnavigation process. They are the plots for $\rho^* - \rho_i$, $\omega^* - \dot{\varphi}_i$ and $-z$ respectively. **d** shows the curves of the real spacings and the desired spacings. The red, green, blue and yellow dashed lines are the desired spacings for robots r_1, r_2, r_3 and r_4 respectively

the desired ones at the end of each stage (see Fig. 10d). Although the simulation experiment involves only four robots, it should be noted that the control algorithm only utilizes the information of neighbouring robots, therefore it can be extended to a system with any number of robots.

7.2.2 Experiment with Soccer-Playing Robots

In this experiment, four soccer-playing robots are used and Formation Guideline 1 is adopted. The utilities of robots r_1, r_3 and r_4 remain 20 throughout the whole circumnavigation process, while the utility of the robot r_2 varies according to a piecewise constant function. That is, $\mu_2 = 1$, $(0 \leq t < 15)$; $\mu_2 = 20$, $(15 \leq t < 30)$; $\mu_2 = 50$, $(30 \leq t < 45)$; $\mu_2 = 0$, $(t \geq 45)$. For convenience, the four time ranges are denoted by Stage 1, 2, 3 and 4 respectively. Note that at Stage 4, the robot r_2 quits from the circumnavigation process as its utility becomes zero. The utilities of the four robots and the corresponding expected spacings are listed in Table 2. In this experiment, robots' initial positions are randomly chosen. The experiment parameters are $\rho^* = 2$ m, $w^* = 0.5$ rad/s, $k_\varphi = 2.5$ and $k_\rho = 2$.

The circumnavigation process is shown in Fig. 11. It demonstrates the positions of robots at different stages. The yellow lines connecting each robot's center indicate the formation shapes. The football in the middle of the field is

Table 2 The utilities and the corresponding expected spacings at four stages

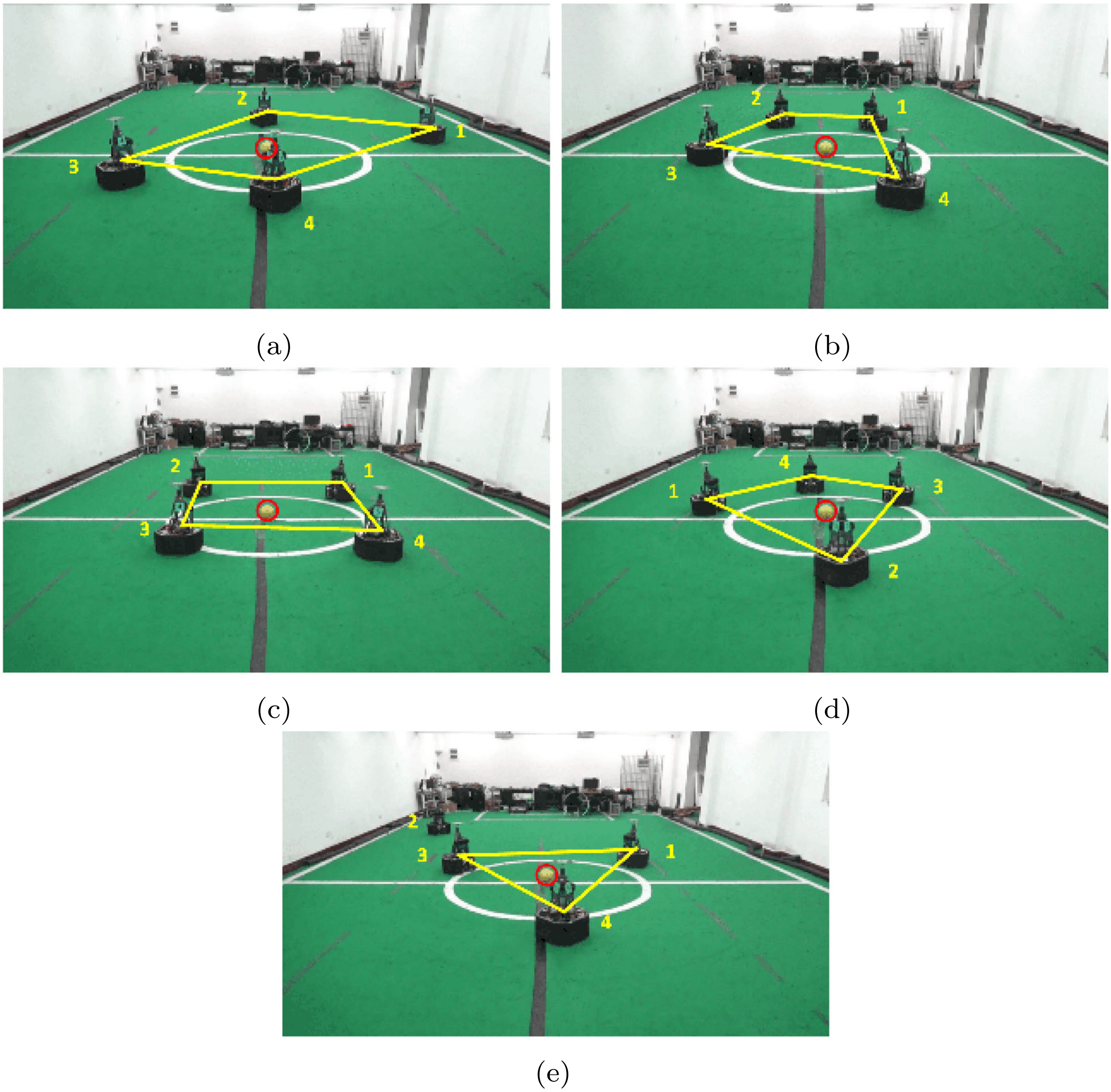
Stage	Utilities	Expected spacings
Stage 1	$[20\ \ 1\ \ 20\ \ 20]^T$	$[62\ \ 62\ \ 118\ \ 118]^T$
Stage 2	$[20\ \ 20\ \ 20\ \ 20]^T$	$[90\ \ 90\ \ 90\ \ 90]^T$
Stage 3	$[20\ \ 50\ \ 20\ \ 20]^T$	$[114\ \ 114\ \ 66\ \ 66]^T$
Stage 4	$[20\ \ 0\ \ 20\ \ 20]^T$	$[120\ \ 120\ \ 120]^T$

Under Formation Guideline 1, the expected spacings (unit: degree) are calculated according to robots' utilities. However, this information is not needed by robots during the circumnavigation process

[a] At this stage, the robot r_2 quits from the formation.

the target to be encircled, which is marked by a red circle. The corresponding data plots are shown in Fig. 12. Since robots move on the ground, the error plot of $-z$ is omitted. As for Fig. 12d, the red, green, blue and black solid lines connecting the centers of robots represent the formation shapes at Stage 1, 2, 3 and 4 respectively. The dashed lines originated from robots are their trajectories. Note that since r_2 quits from the formation at Stage 4 ($\mu_2 = 0$), the data related to r_2 is not plotted after 45 s.

It can be seen from these figures that at Stage 1, since the utility of r_2 is the least, its neighbouring robots r_1 and r_3 decrease their spacings with r_2 to compensate for this



(a)

(b)

(c)

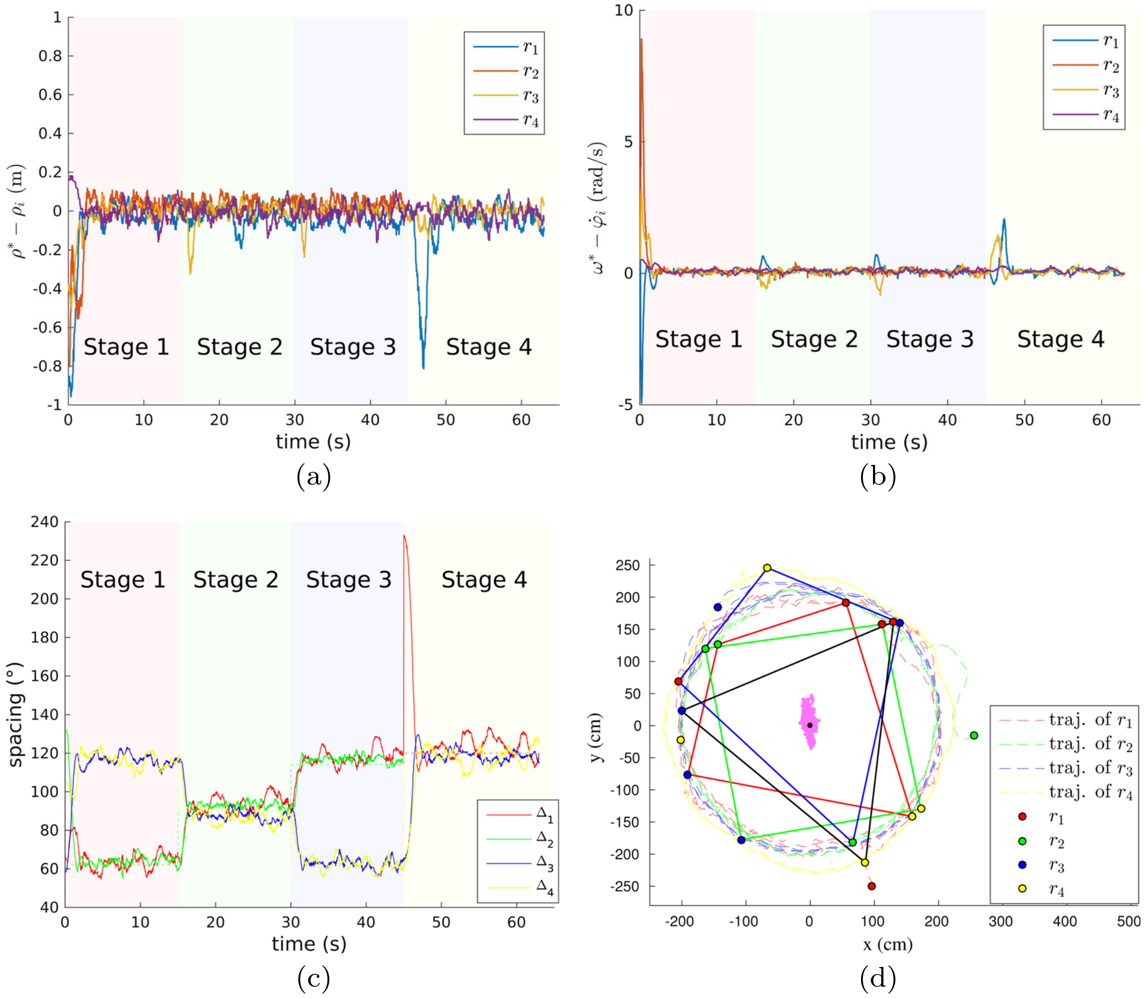
(d)

(e)

Fig. 11 The real-robot experiment. **a** shows the initial robots' positions; **b**–**e** show robots' positions at 4 s (Stage 1), 19 s (Stage 2), 39 s (Stage 3) and 61 s (Stage 4) respectively

insufficiency (see Fig. 11b or the red solid lines in Fig. 12d). When it comes to Stage 2, robots form an equal-spacings formation as their utilities are equal (see Fig. 11c or the green solid lines in Fig. 12d). Stage 3 is contrary to Stage 1, where the utility of r_2 becomes the greatest. Therefore, its two neighbouring robots increase the corresponding spacings with it (see Fig. 11d or the blue solid lines in Fig. 12d). Finally, at Stage 4, r_2 quits from the formation due to its zero utility. The other three robots form a new communication topology (i.e., excluding the robot r_2) and transform to an equilateral triangular formation as their utilities are identical (see Fig. 11e or the black solid lines in Fig. 12d).

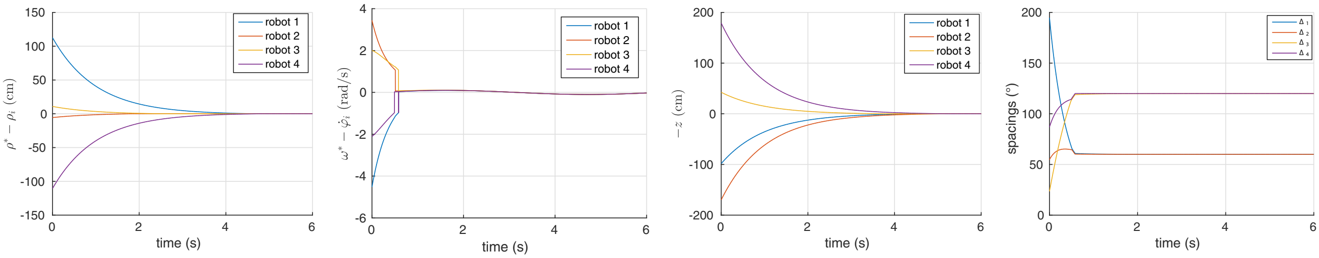
In addition, the circumnavigation radii, angular speeds and formation spacings converge to but fluctuate around the desired values at each stage (see Fig. 12a, b and c). Moreover, at the three intersections of the stages (15 s, 30 s and 45 s respectively), the circumnavigation radii and the angular speeds of robots r_1 and r_3, the two neighbouring robots of r_2, experience large deviation from the desired values. However, they converge swiftly afterwards (see Fig. 12b). Noticeably, at the last intersection (45 s), the circumnavigation radius and spacings for the robot r_1 deviate significantly from the desired values due to the absence of the robot r_2 in the formation, but the errors diminish rapidly subsequently (see Fig. 12a and c). The robot r_4 is hardly affected as it is not a neighbouring robot of r_2. In Fig. 12d, the black dot at the center is the real position of the target and the cluster of pink dots are the perceived positions of the target by r_1. This manifests that information noise increases the uncertainty of the perceived information. Although there are fluctuations due to the information noise, the real spacings converge to the expected spacings at each stage (see Fig. 12c).



(a) (b)

(c) (d)

Fig. 12 The data plots for the real robot experiment. **a** and **b** are the error signals of the circumnavigation process. They are the plots for $\rho^* - \rho_i$ and $\omega^* - \dot{\varphi}_i$ respectively. **c** is the curves of the real spacings and the desired spacings. The red, green, blue and yellow dashed lines are the desired spacings for robots r_1, r_2, r_3 and r_4 respectively. **d** is the plot of the trajectories of robots on the XSY plane



(a) (b) (c) (d)

Fig. 13 The data plots of the simulation for circumnavigation process affected by some external perturbations. **a**–**c** show the plots for $\rho^* - \rho_i$, $\omega^* - \dot{\varphi}_i$ and $-z$ respectively. **d** shows the spacing among robots

7.3 Simulation Results for Circumnavigation with External Perturbations

We simulate the circumnavigation process affected by some external perturbations as shown in Fig. 13. The control algorithms (62), (63) and (64) are employed. The parameters are: $\rho^* = 200$, $w^* = 3$, $k_\rho = 1$, $k_\varphi = 2$, $k_z = 1$ and $k_i = -1$ for $i = 1, \ldots, 4$. The external perturbations are modeled by a sinusoidal wave: $d_i = 0.1\sin(t)$ for $i = 1, \ldots, 4$. It can be seen that the results are consistent with the theoretical analysis.

8 Concluding Remarks and Future Work

This paper proposes distributed control algorithms for a heterogeneous multi-robot system to realize both the static and the dynamic circumnavigation processes in three dimensional space. There is no assumption that the robots should be initially placed on a prescribed circle nor should they splay evenly on the circle. Particularly, the spacings among robots are either arbitrarily given or dynamically changing based on robots' utilities. The concept of utilities and formation guidelines are first proposed in this study, and they jointly facilitate the design of dynamic circumnavigation control algorithm, which is able to address the issue where robots quit from or join into the circumnavigation process. The control algorithms are distributed and applicable for a heterogeneous multi-robot system of arbitrary size.

Theorem 4 cannot guarantee that robots would not collide with each other since it assumes that robots are mass points. Therefore, the collision avoidance problem taking into account the physical dimensions of robots will be studied in the future.

Acknowledgements Our work is supported by National Science Foundation of China (NO.61773393 and NO. 61503401), China Postdoctoral Science Foundation (NO. 2014M562648), and graduate school of National University of Defense Technology.

Appendix

Proof of Theorem 4

Proof The proof is similar to that of Theorem 4 except for some minor changes. First, according to the classical control theory, ρ_i and z_i will converge exponentially to ρ^* and 0 respectively. Since μ_i, $i = 1, \ldots, n$, is piecewise constant, it is obvious that $\lim\limits_{t\to\infty} f_i$ exists. As before, we define $\widetilde{\varphi} = [\widetilde{\varphi}_1 \ldots \widetilde{\varphi}_n]^T$ and $\varphi = [\varphi_1 \ldots \varphi_n]^T$, so Eqs. 56 and 57 can be written into compact forms as $\dot{\varphi} = \omega^*\mathbf{1} + k_\varphi(\widetilde{\varphi} - \varphi)$, and $\widetilde{\varphi} = \hat{A}\varphi + \hat{b}$, where $\hat{A} \in M_n$ is shown as Eq. 66, and $\hat{b} \in R^n$ is as Eq. 67:

$$\hat{A} = \begin{bmatrix} 0 & \frac{\mu_n+\mu_1}{\mu_2+2\mu_1+\mu_n} & \cdots & \frac{\mu_1+\mu_2}{\mu_2+2\mu_1+\mu_n} \\ \frac{\mu_2+\mu_3}{\mu_3+2\mu_2+\mu_1} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\mu_{n-1}+\mu_n}{\mu_1+2\mu_n+\mu_{n-1}} & 0 & \cdots & 0 \end{bmatrix}. \tag{66}$$

$$\hat{b} = 2\pi \left[\frac{-(\mu_1+\mu_2)}{\mu_2+2\mu_1+\mu_n} \; 0 \ldots 0 \; \frac{\mu_{n-1}+\mu_n}{\mu_1+2\mu_n+\mu_{n-1}} \right]^T \tag{67}$$

During each time period where μ_i is constant, $\hat{A}$ and $\hat{b}$ are constant matrix and vector respectively. Similarly, $\mathcal{G}(\hat{A})$ is strongly connected and the error signal is $e_\varphi = -\hat{L}_p\varphi + \hat{b}$, where $\hat{L}_p = I_n - \hat{A}$, which is the Laplacian matrix of $\mathcal{G}(\hat{A})$. Since $\hat{L}_p$ is constant at each time period, the derivative of e_φ is $\dot{e}_\varphi = -\hat{L}_p\dot{\varphi}$. Then $e_\varphi(t) = \exp(-k_\varphi\hat{L}_p t)e_\varphi(0)$ and $\lim\limits_{t\to\infty} e_\varphi(t) = w_r(-w_l^T\hat{L}_p\varphi + w_l^T\hat{b}) = w_l^T\hat{b}w_r$. Let $w_r = \mathbf{1}$ and $w_l = \frac{w_L}{\sum_{w_L}}$, where the ith entry of w_L is

$$\left[w_{L_i} = (\mu_{i^+} + 2\mu_i + \mu_{i^-}) \prod_{j=1, j\neq i, i^-}^{n} (\mu_j + \mu_{j^+}) \right],$$

and $\sum_{w_L} = \sum_{i=1}^n w_{L_i}$. 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, Eq. 25 becomes $\lim\limits_{t\to\infty} e_\varphi(t) = \mathbf{0}$, or

$\lim_{t\to\infty}\varphi(t) = \lim_{t\to\infty}\widetilde{\varphi}(t)$. According to $\dot{\varphi} = \omega^*\mathbf{1} + k_\varphi(\widetilde{\varphi} - \varphi)$, the circumnavigation speed of each robot converges to the desired angular speed ω^*. In addition, under this condition, $\widetilde{\varphi}_i$ is replaced by φ_i in Eq. 57 and therefore, for robots with indices $i = 2, ..., n$, the equation $\varphi_i = \varphi_{i^-} + \frac{\mu_{i^-}+\mu_i}{\mu_{i^+}+2\mu_i+\mu_{i^-}}(\Delta_i + \Delta_{i^-})$ further becomes $\frac{\Delta_i}{\Delta_{i^-}} = \frac{\mu_i+\mu_{i^+}}{\mu_i+\mu_{i^-}}$. This means a sequence of equations $\frac{\Delta_n}{\Delta_{n-1}} = \frac{\mu_n+\mu_1}{\mu_{n-1}+\mu_n}, ..., \frac{\Delta_2}{\Delta_1} = \frac{\mu_2+\mu_3}{\mu_1+\mu_2}$. Assuming $\Delta_1 = k(\mu_1 + \mu_2)$, $k \neq 0$, we have $\Delta_i = k(\mu_i + \mu_{i^+}), i = 2, ..., n$. According to Eq. 8, it follows that $2k\sum_{i=1}^n \mu_i = 2\pi$, and hence $k = \pi/\sum_{i=1}^n \mu_i$. Therefore, $\Delta_i = (\mu_i + \mu_{i^+})\pi/\sum_{i=1}^n \mu_i = f_i(t, \mu_1, \ldots, \mu_n)$, $i = 1, ..., n$. So the expected spacings expressed by Eqs. 49 and 53 can be achieved. □

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

References

1. Antsaklis, P., Michel, A.N.: Linear systems. Springer Science & Business Media (2006)
2. Chen, W.H.: Disturbance observer based control for nonlinear systems. IEEE/ASME Trans. Mechatron. **9**(4), 706–710 (2004)
3. Dai, W., Yao, W., Luo, S., Ma, J., Wang, R., Hong, S., Zhou, Z., Li, X., Han, B., Xiao, J., et al.: Nubot team description paper (2018)
4. Fax, J.A., Murray, R.M.: Information flow and cooperative control of vehicle formations. IEEE Trans. Autom. Control **35**(1), 115–120 (2002)
5. Franchi, A., Stegagno, P., Oriolo, G.: Decentralized multi-robot encirclement of a 3D target with guaranteed collision avoidance. Auton. Robot. **40**(2), 245–265 (2016). https://doi.org/10.1007/s10514-015-9450-3
6. Gowers, T., Barrow-Green, J., Leader, I.: The Princeton Companion to Mathematics. Princeton University Press, Princeton (2010)
7. Horn, R.A., Johnson, C.R.: Matrix Analysis. Cambridge University Press, Cambridge (2012)
8. Lu, H., Yang, S., Zhang, H., Zheng, Z.: A robust omnidirectional vision sensor for soccer robots. Mechatronics **21**(2), 373–389 (2011)
9. Marshall, B.J.A., Broucke, M.E., Francis, B.A.: Formation of vehicles in cyclic pursuit. IEEE Trans. Autom. Control **49**(11), 1963–1974 (2015)
10. Mesbahi, M., Egerstedt, M.: Graph Theoretic Methods in Multiagent Networks. Princeton University Press, Princeton (2010)
11. Minc, H.: Nonnegative Matrices. Wiley, New York (1988)
12. Olfati-Saber, R., Murray, R.M.: Consensus problems in networks of agents with switching topology and time-delays. IEEE Trans. Autom. Control **49**(9), 1520–1533 (2004). https://doi.org/10.1109/TAC.2004.834113
13. Pavone, M., Frazzoli, E.: Decentralized policies for geometric pattern formation and path coverage. J. Dyn. Syst. Meas. Control. **129**(5), 633–643 (2007)
14. Ren, W., Beard, R.W.: Distributed Consensus in Multi-vehicle Cooperative Control. Springer, Berlin (2008)
15. Tang, S., Shinzaki, D., Lowe, G.C., Clark, C.M.: Multi-robot control for circumnavigation of particle distributions. Springer Tracts in Advanced Robotics **104**, 149–152 (2014)
16. Wang, C., Xie, G., Cao, M.: Forming circle formations of anonymous mobile agents with order preservation. IEEE Trans. Autom. Control **58**(12), 3248–3254 (2013)
17. Wang, C., Xie, G., Cao, M.: Controlling anonymous mobile agents with unidirectional locomotion to form formations on a circle. Automatica **50**(4), 1100–1108 (2014)
18. Wang, X., Zeng, Z., Cong, Y.: Multi-agent distributed coordination control: developments and directions via graph viewpoint. Neurocomputing **199**, 204–218 (2016). https://doi.org/10.1016/j.neucom.2016.03.021
19. Xiong, D., Xiao, J., Lu, H., Zeng, Z., Yu, Q., Huang, K., Yi, X., Zheng, Z.: The design of an intelligent soccer-playing robot. Ind. Robot. **43**, 91–102 (2016). https://doi.org/10.1108/IR-05-2015-009
20. Yao, W., Dai, W., Xiao, J., Lu, H., Zheng, Z.: A simulation system based on ros and gazebo for robocup middle size league. In: 2015 IEEE International Conference on Robotics and Biomimetics (ROBIO), pp. 54–59 (2015). https://doi.org/10.1109/ROBIO.2015.7414623
21. Yao, W., Luo, S., Lu, H., Xiao, J.: Distributed circumnavigation control with dynamic spacing for a heterogeneous multi-robot system. In: 2018 RoboCup Symposium. Springer, Berlin (2018)
22. Yao, W., Zeng, Z., Wang, X., Lu, H., Zheng, Z.: Distributed encirclement control with arbitrary spacing for multiple anonymous mobile robots. In: 2017 36th Chinese Control Conference (CCC), pp. 8800–8805 (2017). https://doi.org/10.23919/ChiCC.2017.8028755
23. Yu, X., Liu, L.: Distributed circular formation control of ring-networked nonholonomic vehicles. Automatica **68**, 92–99 (2016). https://doi.org/10.1016/j.automatica.2016.01.056
24. Zeng, Z.: Multi-agent Coordination Control with Nonlinear Dynamics, Quantized Communication and Structure-constraint. Ph.D. thesis, National University of Defense Technology, Changsha (2016)
25. Zheng, R., Lin, Z., Fu, M., Sun, D.: Distributed circumnavigation by unicycles with cyclic repelling strategies. In: Control Conference (ASCC), 2013 9th Asian, pp. 1–6. IEEE (2013)
26. Zheng, R., Liu, Y., Sun, D.: Enclosing a target by nonholonomic mobile robots with bearing-only measurements. Automatica **53**, 400–407 (2015). https://doi.org/10.1016/j.automatica.2015.01.014

Weijia Yao received his Bachelors of Engineering (2015), Masters of Engineering (2017) from the National University of Defense Technology (NUDT), China. Currently he is a Ph.D. student at the research group of Discrete Technology and Production Automation, University of Groningen, The Netherlands. The focus of his research is on mobile robotics, especially on multi-robot coordination, and robot control.

Huimin Lu received his Bachelors of Engineering (2003), Masters of Engineering (2005) and Ph.D. (2010) from the National University of Defense Technology (NUDT), China. Later, he joined the Department of Automation, NUDT (2010) where he is an associate professor. The focus of his research is on mobile robotics, especially on robot vision, multi-robot coordination, robot soccer and robot rescue.

Zhiwen Zeng received his Bachelors of Engineering (2009) from the University of Electronic Science and Technology of China, Masters of Engineering (2011) and Ph.D. (2016) from the National University of Defense Technology (NUDT), China. Later, he joined the Department of Automation, NUDT (2016) where he is an assistant professor. The focus of his research is on mobile robotics, especially on multi-robot coordination, and robot control.

Junhao Xiao received his Bachelors of Engineering (2007) from the National University of Defense Technology (NUDT), China, and Ph.D. (2013) from the Institute of Technical Aspects of Multimodal Systems (TAMS), Department Informatics, University of Hamburg, Germany. Later, he joined the Department of Automation, NUDT (2013) where he is an assistant professor. The focus of his research is on mobile robotics, especially on localization, mapping, robot soccer and robot rescue.

Zhiqiang Zheng received his Ph.D (1994) from University of Liege, Belgium. Later, he joined the Department of Automation, National University of Defense Technology (NUDT), China, where he is a professor. The focus of his research is on mobile robotics and flight control, especially on multirobot coordination and robot vision.