Leader-Follower Formation Control of Perturbed Nonholonomic Agents Along Parametric Curves With Directed Communication

Bin Zhang, *Graduate Student Member, IEEE*, Xiaodong Shao, *Member, IEEE*, Hui Zhi, *Graduate Student Member, IEEE*, Liuming Qiu, *Member, IEEE*, Jose Guadalupe Romero, *Member, IEEE*, and David Navarro-Alarcon, *Senior Member, IEEE*

***Abstract*—In this letter, we propose a novel formation controller for nonholonomic agents to form general parametric curves. First, we derive a unified parametric representation for both open and closed curves. Then, a leader-follower formation controller is designed to drive agents to form the desired parametric curves using the curve coefficients as feedbacks. We consider directed communications and constant input disturbances rejection in the controller design. Rigorous Lyapunov-based stability analysis proves the asymptotic stability of the proposed controller. The convergence of the orientations of agents to some constant values is also guaranteed. The method has the potential to be extended to deal with various real-world applications, such as object enclosing. Detailed numerical simulations and experimental studies are conducted to verify the performance of the proposed method.**

***Index Terms*—Directed graphs, disturbance rejection, formation control, multi-agent systems, parametric curves.**

I. Introduction

The application of multi-agent systems has gained popularity over the past decades [1]. A typical multi-agent system usually has multiple layers to deal with assigned tasks, such as formation control, path planning, environmental sensing, and coordination [2]. Among those, multi-agent formation control is the fundamental problem studied by many researchers. Formation control techniques can be used in various practical problems, such as object transport [3], rescue [4], and environmental surveillance [5]. In these scenarios, agents are usually controlled to form a desired pattern and keep/change it according to the task requirements [6]. Generally, agents can achieve formation control in three ways, that is, position-based, displacement-based, and distance-based methods [7]. The platforms that carry out the formation control can be mobile robots [8], [9], aerial vehicles [10], satellites [11], manipulators [12], etc.

Manuscript received 25 March 2024; accepted 8 August 2024. Date of publication 19 August 2024; date of current version 30 August 2024. This article was recommended for publication by Associate Editor G. Notomista and Editor M. A. Hsieh upon evaluation of the reviewers' comments. This work was supported by Research Grants Council under Grant 15231023. *(Corresponding author: David Navarro-Alarcon.)*

Bin Zhang, Hui Zhi, Liuming Qiu, and David Navarro-Alarcon are with the Department of Mechanical Engineering, The Hong Kong Polytechnic University (PolyU), Kowloon, Hong Kong (e-mail: me-bin.zhang@connect.polyu.hk; hui1225.zhi@connect.polyu.hk; valen.qiu@connect.polyu.hk; dna@ieee.org).

Xiaodong Shao is with the School of Automation Science and Electrical Engineering, Beihang-University (BUAA), Beijing 100191, China (e-mail: xdshao_sasee@buaa.edu.cn).

Jose Guadalupe Romero is with the Department of Electrical and Electronic Engineering, Instituto Tecnológico Autónomo de México (ITAM), CDMX 01080, Mexico (e-mail: jose.romerovelazquez@itam.mx).

This letter has supplementary downloadable material available at https://doi.org/10.1109/LRA.2024.3445657, provided by the authors.

Digital Object Identifier 10.1109/LRA.2024.3445657

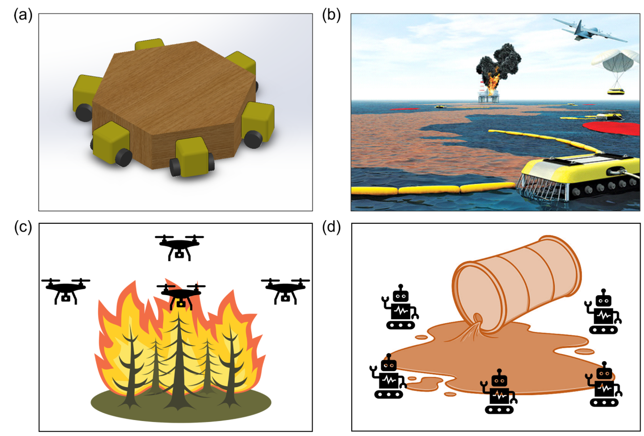

Fig. 1. Conceptual applications of multi-agent systems: (a) Robots enclosing an object; (b) Robots monitoring spilled oil using AEROS robots [13]; (c) Drones monitoring wildfires; (d) Robots cleaning leaked chemicals.

In classical formation control methods, the desired patterns that agents need to form are usually specified by absolute or relative positions, and those patterns are usually simple in terms of morphology, such as polygons and lines. This could cause problems when dealing with complicated cases where the desired pattern has a complex structure, such as curves, because it is usually difficult to fully depict the characteristics of such complex patterns simply with several positions. There are also scenarios where we only care about the overall distribution rather than the specific positions of agents, such as caging an object during transportation tasks or monitoring the boundaries of oil spills, wildfires, and chemical leakages, etc., see Fig. 1. In these cases, representing the desired pattern with several positions can limit the flexibility of the system. In this work, we consider adopting parametric curves, a general approach to represent the desired patterns, and develop novel formation controllers to drive agents toward this structure.

Related work: Compared with the abundant literature on multi-agent formation control, works on driving agents to form patterns described by parametric curves are limited. An early work can be seen in [14], where the author developed a steering control to drive particles to travel along closed invariant patterns (curves). In this work, particles are coupled in a chain structure and limited to a constant unit speed. Hsieh et al. [15] proposed a decentralized controller for robotic swarms to form closed planar curves. In this work, agents with double-integrator dynamics were considered, and they did not need to exchange information during the control process. Song et al. [16] studied the coverage control of a multi-agent system on a closed curve. They considered the single-integrator agents and optimized the coverage cost of agents in a finite time. A Fourier series-based method was developed in [17] for nonholonomic agents to track evolving curves, but a strongly connected communication topology was required. Aranda [18] also adopted the Fourier descriptors to specify and control multi-agent formations. They exploited the structure of the discrete Fourier transform (DFT) and proposed controllers for both single-integrator and double-integrator agent models. More examples can also be seen in [19], [20], where estimating and tracking environmental boundaries, such as profiles of plumes and liquids, by multi-agent systems are studied.

By reviewing these previous works, we find that they considered either only closed curves or only open curves, but none of them considered both types of curves in a unified way. Also, most previous controllers do not incorporate disturbance rejection capabilities, which significantly limits their applicability in real-world missions. Besides, most previous works assume simple agents' dynamics, such as single and double integrator dynamics. Extensions to more complex agent dynamics, such as those of nonholonomic systems, still need to be further studied to generate practically realistic solutions. Finally, most previous work has not conducted experimental validations on real-world platforms to assess the performance of their proposed methods.

Our contribution: In this work, we study the formation control problem of multi-agent systems to form complex patterns represented by parametric curves. The main contributions of our work are summarized as follows:

1) We present a unified parametric representation of open and closed curves for the formation control problem. This provides flexibility to automatically allocate the agents along a profile without requiring specific target positions.
2) We propose a novel leader-follower formation controller for nonholonomic agents to form parametric curves under directed communications and constant input disturbances. The controller is constructed based on curve coefficients which encodes the general characteristics of the curve. With different combinations of assigned parameters of agents, the proposed controller can easily drive agents to achieve various distributions on the desired curve.
3) We report a rigorous stability analysis, numerical simulations, and experimental study to investigate the properties and validate the proposed method. In the experimental validations, we not only examinate the performance of proposed method on forming virtual curves, but also extend the method to handle the object enclosing tasks.

Organization: The rest of the paper is organized as follows: Section II presents the preliminaries; Section III derives the formation controller; Section IV validates the theory with simulations and experiments; Section V gives conclusions.

II. Mathematical Modeling

Notation: Matrices and vectors are denoted by bold letters. Scalars are denoted as plain letters. We use $\mathbf{0}$ to denote a vector of appropriate dimension with all elements as zeros, and $\mathbf{0}_m$ to denote a square zero matrix. $\mathbf{I}_n$ denotes a $n \times n$ identity matrix. We use the $\min()$ to represent the operation of finding the minimum and $\mathrm{diag}(\cdot)$ to represent the diagonalization operation, where "$\cdot$" could be square matrices or scalars.

A. Unified Curve Representation

Our goal in this paper is to study the multi-agent control method driving agents to achieve the formation along curves. Therefore, the first step is to define the curve we expect the agents to form. We only consider the planar cases in this paper and assume that the desired curve can be parameterized by the following linear regression equation:

$$\mathbf{c} = \mathbf{G}(s)\boldsymbol{\xi} \tag{1}$$

where $\mathbf{c} = [c_x, c_y]^\top$ denotes the coordinate of a random point on the curve, $s \in [0, 1]$ is the normalized length parameter, $\mathbf{G}(s) \in \mathbb{R}^{2\times 2H}$ is the matrix containing basis functions dependent on the normalized length parameter s, H is the number of basis functions, and $\boldsymbol{\xi} \in \mathbb{R}^{2H\times 1}$ is the vector containing the curve coefficients. By selecting different combinations of basis functions, we can represent the desired curve as different types, such as Fourier series, Bezier contours, and polynomial contours. Note that this means we actually can express both closed curves and open curves with the same form. Therefore, (1) is a unified curve representation of all types of curves.

In real-world missions, we can obtain the curve coefficients by estimation. First, we sample a series of points on the observed environmental curve, denoted as $\mathbf{C} = [\mathbf{c}_1^\top, \ldots, \mathbf{c}_N^\top]^\top$, where N is the total number of sampled points. Then, we compute the normalized length parameter corresponding to each point $\mathbf{c}_i$ by $s_i = l_i/l_N$ for $i = 1, \ldots, N$, where $l_1 = 0$ and $l_i = \sum_{p=2}^{i} \|\mathbf{c}_p - \mathbf{c}_{p-1}\|$ for $i \geq 2$. Next, we appropriately select a series of basis functions and compute the basis matrice corresponding to each sampled point, denoted as $\mathbf{G}_h = [\mathbf{G}_1^\top, \ldots, \mathbf{G}_N^\top]^\top$. Finally, we can estimate the curve coefficients by:

$$\boldsymbol{\xi} = (\mathbf{G}_h^\top \mathbf{G}_h)^{-1}\mathbf{G}_h\mathbf{C} \tag{2}$$

Note that the number of sample points along the curve must satisfy $N > 2H$ to guarantee the existence of $(\mathbf{G}_h^\top \mathbf{G}_h)^{-1}$, which is usually easy to fulfill in real missions by increasing the number of sampled points.

B. Dynamic Model

The agents considered in this work are assumed to conduct planar motions and have nonholonomic dynamics with constant input disturbances. We define the configuration of agent i as $[\mathbf{x}_i^\top, \theta_i]^\top$, where $\mathbf{x}_i = [x_i, y_i]^\top$ and θ_i are the center position and

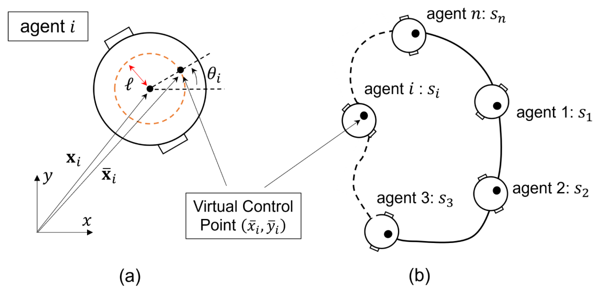

Fig. 2. The configuration of the agent: (a) The change of coordinates; (b) The assignment of length parameters for agents.

the orientation of the agent, respectively. The dynamic model of agent i is given by:

$$\begin{bmatrix} \dot{x}_i \\ \dot{y}_i \\ \dot{\theta}_i \end{bmatrix} = \begin{bmatrix} \cos\theta_i & 0 \\ \sin\theta_i & 0 \\ 0 & 1 \end{bmatrix} (\mathbf{u}_i + \mathbf{d}_i) \tag{3}$$

where $\mathbf{u}_i = [v_i, \omega_i]^\mathsf{T}$ denotes the control input, with v_i and ω_i as the linear and angular velocities of the agent, respectively; $\mathbf{d}_i = [d_{i1}, d_{i2}]^\mathsf{T}$ denotes the constant input disturbances that could rise up from motor asymmetry or imperfect wheel alignment, etc.

To simplify the controller design and stability analysis, we define a virtual control point adopting the input/output feedback linearization [21]. The virtual control point is obtained by the following change of coordinates [22] (see also [23]):

$$\begin{cases} \bar{x}_i := x_i + \ell\cos\theta_i \\ \bar{y}_i := y_i + \ell\sin\theta_i \\ \bar{\theta}_i := \theta_i \end{cases} \tag{4}$$

where $\ell \neq 0$ is an arbitrary scalar that translates the agent's position to an arbitrary close location. The process of the change of coordinates is shown in Fig. 2(a).

By using these new coordinates, we can obtain the agent's shifted position $\bar{\mathbf{x}}_i = [\bar{x}_i, \bar{y}_i]^\mathsf{T}$, whose time derivative yields a shifted dynamic model of the form:

$$\begin{cases} \dot{\bar{\mathbf{x}}}_i = \mathbf{R}_i(\theta_i)(\mathbf{u}_i + \mathbf{d}_i) \\ \dot{\bar{\theta}}_i = \omega_i + d_{i2} \end{cases} \tag{5}$$

for a full-rank matrix defined as:

$$\mathbf{R}_i(\theta_i) = \begin{bmatrix} \cos\theta_i & -\ell\sin\theta_i \\ \sin\theta_i & \ell\cos\theta_i \end{bmatrix} \tag{6}$$

We can easily derive $\|\mathbf{R}_i\| = |\ell|$ and $\|\mathbf{R}_i^{-1}\| = 1/|\ell|$ for a $|\ell| \leq 1$.

By defining a new control input $\bar{\mathbf{u}}_i = \mathbf{R}_i(\theta_i)\mathbf{u}_i$ and a new input disturbance $\bar{\mathbf{d}}_i = \mathbf{R}_i(\theta_i)\mathbf{d}_i$, the position dynamics in (5) can be reduced to a single integrator with input disturbance:

$$\dot{\bar{\mathbf{x}}}_i = \bar{\mathbf{u}}_i + \bar{\mathbf{d}}_i \tag{7}$$

C. Interaction Topology

The interactions topology among agents can be represented by a graph $\mathcal{G} = (\mathcal{G}, \mathcal{E}, a_{ij})$, where $\mathcal{V} = \{1, \ldots, n\}$ denotes the set of nodes, $\mathcal{E} = \{(i,j) \in \mathcal{V} \times \mathcal{V} : a_{ij} \neq 0\}$ denotes the set of edges, and $a_{ij} > 0$ denotes the weight of interactions with ij indicating the i-th and j-th agents. The Laplacian matrix $\mathbf{L} = [l_{ij}] \in \mathbb{R}^{n\times n}$ of the graph is defined by:

$$l_{ij} = \begin{cases} \sum_{i=1}^{n} a_{ij}, & \text{if } i = j \\ -a_{ij}, & \text{if } i \neq j \end{cases} \tag{8}$$

for l_{ij} as the entry of $\mathbf{L}$ at the i-th row and j-th column [24].

In this work, we assume that the interaction topology of agents can be represented by a directed graph containing a rooted spanning tree, i.e., there is a node that can be connected to all the remaining nodes of the graph. The Laplacian has a simple eigenvalue 0 with $\mathbf{1}_n \in \mathbb{R}^{n\times 1}$ as its corresponding right eigenvector, where $\mathbf{1}_n$ denotes a vector with all elements equal to one. All the other eigenvalues of the Laplacian have positive real parts [25].

Besides, we can define an augmented graph for a leader-follower multi-agent system based on the graph of the followers. We can describe the communication between the leader and the followers by a nonnegative matrix $\mathbf{B} = \text{diag}(b_i)$, where $b_i > 0$ if and only if there is an edge between the i-th follower and the leader. Assuming that the augmented graph contains a spanning tree, we have the following theorem on the construction of Lyapunov functions for stability analysis.

Theorem 1 (Zhang et al. [26]): Let

$$\begin{cases} \mathbf{q} = [q_i]^\mathsf{T} = (\mathbf{L} + \mathbf{B})^{-1}\mathbf{1}_n \\ \mathbf{p} = [p_i]^\mathsf{T} = (\mathbf{L} + \mathbf{B})^{-\mathsf{T}}\mathbf{1}_n \\ \mathbf{P} = \text{diag}(p_i/q_i) \\ \mathbf{Q} = \mathbf{P}(\mathbf{L} + \mathbf{B}) + (\mathbf{L} + \mathbf{B})^\mathsf{T}\mathbf{P} \end{cases} \tag{9}$$

Then both $\mathbf{P}$ and $\mathbf{Q}$ are positive definite.

III. Proposed Control Method

A. Controller Design

By stacking the position dynamics (7) of agents, we can represent the position dynamics of the multi-agent system as:

$$\dot{\bar{\mathbf{x}}} = \bar{\mathbf{u}} + \bar{\mathbf{d}} \tag{10}$$

where $\bar{\mathbf{x}} = [\bar{\mathbf{x}}_i^\mathsf{T}]^\mathsf{T}$, $\bar{\mathbf{u}} = [\bar{\mathbf{u}}_i^\mathsf{T}]^\mathsf{T}$, and $\bar{\mathbf{d}} = [\bar{\mathbf{d}}_i^\mathsf{T}]^\mathsf{T} \in \mathbb{R}^{2n\times 1}$ are the extended position vector, input vector, and disturbance vector, respectively.

According to the desired distribution of agents along the curve, we can define a sequence of length parameters corresponding to the agents $\mathbf{s} = [s_1, \ldots, s_i, \ldots, s_n]^\mathsf{T}$, where s_i is the length parameter corresponding to the ith agent. For example, if we want the agents to be uniformly distributed along the curve, we can define s_i by $s_i = (i-1)/n$. Then, we assign the length parameters s_i to each agent to ensure the desired arc length separation of agents along the curve (i.e., the desired distribution of agents). The process is shown in Fig. 2(b). With that, we can calculate the matrix of basis functions corresponding to each agent as $\bar{\mathbf{G}} = [\mathbf{G}_1(s_1)^\mathsf{T}, \ldots, \mathbf{G}_n(s_n)^\mathsf{T}]^\mathsf{T}$

Now, we can define the agent's position errors as:

$$\bar{\mathbf{x}}_e = \bar{\mathbf{x}} - \bar{\mathbf{G}}\boldsymbol{\xi} \tag{11}$$

and the curve coefficient errors as:

$$\boldsymbol{\xi}_e = \bar{\mathbf{G}}^+\bar{\mathbf{x}} - \boldsymbol{\xi} \tag{12}$$

where $\bar{\mathbf{G}}^+$ is the pseudoinverse of $\bar{\mathbf{G}}$.

To achieve formation control and reject the input disturbance, we propose the following leader-follower controller:

$$\bar{\mathbf{u}}_i = \begin{cases} -k_1(\bar{\mathbf{x}}_i - \mathbf{G}_i\boldsymbol{\xi}) - k_2\mathbf{R}_i\hat{\boldsymbol{\delta}}_i & \text{leader} \\ -k_1\sum_{j\in\mathcal{N}_i} a_{ij}(\mathbf{G}_i - \mathbf{G}_j)\boldsymbol{\xi}_e - k_2\mathbf{R}_i\hat{\boldsymbol{\delta}}_i & \text{follower} \end{cases} \tag{13}$$

where $k_1, k_2 > 0$ are the control gains, $\hat{\boldsymbol{\delta}}_i$ denotes an estimation variable related to the input disturbance, and $\mathcal{N}_i = \{j \in \mathcal{V} : a_{ij} \neq 0\}$ is the set of neighbors of agent i. The first part of the controller drives agents to achieve the formation; The second part of the controller eliminates the input disturbance. The disturbance estimation is updated by:

$$\dot{\hat{\boldsymbol{\delta}}}_i = k_2\mathbf{R}_i^{\mathsf{T}}(\bar{\mathbf{x}}_i - \mathbf{G}_i\boldsymbol{\xi}) \tag{14}$$

B. Stability Analysis

Without loss of generality, we assume that agent 1 is the root of the spanning tree contained in the interaction topology and the leader of the multi-agent system. This indicates that the first row of the Laplacian $\mathbf{L}$ is a zero vector. Then, we can stack the controller (13) for each agent to obtain the control input $\bar{\mathbf{u}}$ for the whole multi-agent system:

$$\bar{\mathbf{u}} = -k_1(\bar{\mathbf{L}}\bar{\mathbf{G}}\boldsymbol{\xi}_e + \bar{\boldsymbol{\Lambda}}\bar{\mathbf{x}}_e) - k_2\mathbf{R}\hat{\boldsymbol{\delta}} \tag{15}$$

where $\bar{\mathbf{L}} = \mathbf{L} \otimes \mathbf{I}_2$ is the extended Laplacian matrix with $\otimes$ representing the Kronecker product; $\bar{\boldsymbol{\Lambda}} = \boldsymbol{\Lambda} \otimes \mathbf{I}_2 \in \mathbb{R}^{2n\times 2n}$ for $\boldsymbol{\Lambda} = \text{diag}(1, \mathbf{0}_{n-1})$; $\mathbf{R} = \text{diag}(\mathbf{R}_i) \in \mathbb{R}^{2n\times 2n}$; and $\hat{\boldsymbol{\delta}} = [\hat{\boldsymbol{\delta}}_i^{\mathsf{T}}]^{\mathsf{T}} \in \mathbb{R}^{2n\times 1}$.

Replacing controller (15) into the stacked dynamics (16), we can obtain the complete dynamics of the multi-agent system:

$$\dot{\bar{\mathbf{x}}} = -k_1(\bar{\mathbf{L}}\bar{\mathbf{G}}\boldsymbol{\xi}_e + \bar{\boldsymbol{\Lambda}}\bar{\mathbf{x}}_e) - k_2\mathbf{R}\hat{\boldsymbol{\delta}} + \bar{\mathbf{d}} \tag{16}$$

Now, we define a disturbance estimation error measurement:

$$\tilde{\boldsymbol{\delta}} = \hat{\boldsymbol{\delta}} - \frac{1}{k_2}\mathbf{d} \tag{17}$$

for $\mathbf{d} = [\mathbf{d}_i^{\mathsf{T}}]^{\mathsf{T}} \in \mathbb{R}^{2n\times 1}$. Then, we can rewrite (16) as:

$$\dot{\bar{\mathbf{x}}} = -k_1(\bar{\mathbf{L}}\bar{\mathbf{G}}\boldsymbol{\xi}_e + \bar{\boldsymbol{\Lambda}}\bar{\mathbf{x}}_e) - k_2\mathbf{R}\tilde{\boldsymbol{\delta}} \tag{18}$$

It can be derived that $\dot{\tilde{\boldsymbol{\delta}}} = \dot{\hat{\boldsymbol{\delta}}} = k_2\mathbf{R}^{\mathsf{T}}\bar{\mathbf{x}}_e$ considering that $\mathbf{d}$ is constant. Let $\bar{\boldsymbol{\delta}} = \mathbf{R}\tilde{\boldsymbol{\delta}}$. We can see that $\bar{\boldsymbol{\delta}}$ measures the disturbance estimation error for (10).

The stability analysis of the proposed controller is based on the following assumptions.

Assumption 1: The number of agents and basis functions satisfies $n \leq H$.

Assumption 2: The rank of matrix $\bar{\mathbf{G}} \in \mathbb{R}^{2n\times 2H}$ always satisfies $\text{rank}(\bar{\mathbf{G}}) = 2n$.

With Assumptions 1 and 2, we can ensure the existence of the pseudoinverse $\bar{\mathbf{G}}^+$. Specifically, we can compute $\bar{\mathbf{G}}^+$ by $\bar{\mathbf{G}}^{\mathsf{T}}(\bar{\mathbf{G}}\bar{\mathbf{G}}^{\mathsf{T}})^{-1}$. It can be derived that $\bar{\mathbf{G}}\bar{\mathbf{G}}^+ = \mathbf{I}_{2n}$. Then, we can obtain that $\bar{\mathbf{G}}\boldsymbol{\xi}_e = \bar{\mathbf{G}}\bar{\mathbf{G}}^+\bar{\mathbf{x}}_e = \bar{\mathbf{x}}_e$.

Remark 1: The condition proposed in Assumption 1 is reasonable. We usually need more basis functions to accurately approximate a curve with a complex shape structure. However, A very large H should be carefully selected to avoid unwanted oscillations in the structure of the approximated curve. As for “simple” curves, such as straight lines and circles, we can regard them as a combination of many basis functions with the coefficients of high-order basis functions as zeros. For example, we can represent a straight line as $\mathbf{c} = \mathbf{a} + \mathbf{b}s + \mathbf{0}(s^2 + s^3 + \dots)$ for $\mathbf{a}$ and $\mathbf{b} \in \mathbb{R}^{2\times 1}$ as coefficients. In this case, the condition $n \leq H$ is only a mathematical consideration for the stability analysis.

Proposition 1: Consider a multi-agent system of n agents with dynamics (7) and a directed graph containing a rooted spanning tree. Given a *static* planar curve in the form of (1), controller (15) drives agents to be uniformly distributed on the curve and ensures asymptotic stability of position errors $\bar{\mathbf{x}}_e$.

Proof: It can be seen that the Laplacian $\mathbf{L}$ and the matrix $\boldsymbol{\Lambda}$ satisfies Theorem 1. Therefore, replace $\mathbf{B}$ with $\boldsymbol{\Lambda}$ in Theorem 1, then we can obtain positive definite matrice $\mathbf{P}$, $\mathbf{Q} \in \mathbb{R}^{n\times n}$. Consider the following Lyapunov function:

$$V = \bar{\mathbf{x}}_e^{\mathsf{T}}(\mathbf{P}\otimes\mathbf{I}_2)\bar{\mathbf{x}}_e + \tilde{\boldsymbol{\delta}}^{\mathsf{T}}(\mathbf{P}\otimes\mathbf{I}_2)\tilde{\boldsymbol{\delta}} \tag{19}$$

Compute the time derivative of (19), then we obtain:

$$\begin{aligned}\dot{V} &= 2\bar{\mathbf{x}}_e^{\mathsf{T}}(\mathbf{P}\otimes\mathbf{I}_2)\dot{\bar{\mathbf{x}}}_e + 2\tilde{\boldsymbol{\delta}}^{\mathsf{T}}(\mathbf{P}\otimes\mathbf{I}_2)\dot{\tilde{\boldsymbol{\delta}}} \\ &= 2\bar{\mathbf{x}}_e^{\mathsf{T}}(\mathbf{P}\otimes\mathbf{I}_2)[-k_1(\bar{\mathbf{L}}\bar{\mathbf{G}}\boldsymbol{\xi}_e + \bar{\boldsymbol{\Lambda}}\bar{\mathbf{x}}_e) - k_2\mathbf{R}\tilde{\boldsymbol{\delta}}] \\ &\quad + 2\tilde{\boldsymbol{\delta}}^{\mathsf{T}}(\mathbf{P}\otimes\mathbf{I}_2)\dot{\tilde{\boldsymbol{\delta}}} \\ &= -2k_1\bar{\mathbf{x}}_e^{\mathsf{T}}(\mathbf{P}\otimes\mathbf{I}_2)(\mathbf{L}\otimes\mathbf{I_2} + \boldsymbol{\Lambda}\otimes\mathbf{I_2})\bar{\mathbf{x}}_e \\ &\quad - 2k_2\bar{\mathbf{x}}_e^{\mathsf{T}}(\mathbf{P}\otimes\mathbf{I}_2)\mathbf{R}\tilde{\boldsymbol{\delta}} + 2\tilde{\boldsymbol{\delta}}^{\mathsf{T}}(\mathbf{P}\otimes\mathbf{I}_2)\dot{\tilde{\boldsymbol{\delta}}}\end{aligned} \tag{20}$$

where we use $\bar{\mathbf{G}}\boldsymbol{\xi}_e = \bar{\mathbf{x}}_e$ to obtain the last equality. Checking the structure of $\mathbf{R}$ and $\mathbf{P}\otimes\mathbf{I}_2$, we can show that:

$$(\mathbf{P}\otimes\mathbf{I}_2)\mathbf{R} = \text{diag}\left(\frac{p_i}{q_i}\mathbf{I_2}\mathbf{R}_i\right) = \mathbf{R}(\mathbf{P}\otimes\mathbf{I}_2) \tag{21}$$

Therefore, to ensure the positiveness of $\dot{V}$, we choose $\dot{\tilde{\boldsymbol{\delta}}} = k_2\mathbf{R}^{\mathsf{T}}\bar{\mathbf{x}}_e$, then we can obtain:

$$\dot{V} = -\bar{\mathbf{x}}_e^{\mathsf{T}}(\mathbf{Q}\otimes\mathbf{I}_2)\bar{\mathbf{x}}_e \leq 0 \tag{22}$$

where $\dot{V} = 0$ if and only if $\bar{\mathbf{x}}_e = \mathbf{0}$. Therefore, $\bar{\mathbf{x}}_e = \mathbf{0}$ is asymptotically stable. In ideal cases, we have $\bar{\mathbf{x}}_e = \mathbf{0}$ and $\dot{\bar{\mathbf{x}}}_e = \mathbf{0}$ when the system reaches a stable stage. Since we have $\dot{\bar{\mathbf{x}}}_e = \dot{\bar{\mathbf{x}}}$ from (11), we can derive that $\bar{\boldsymbol{\delta}} = \mathbf{0}$ and $\tilde{\boldsymbol{\delta}} = \mathbf{0}$ are asymptotically stable from (18). In real-world cases, $\bar{\mathbf{x}}_e$ usually can not strictly converge to zero when the system reaches the stable stage. From (18), we can derive the following relation:

$$\left\|\tilde{\boldsymbol{\delta}}\right\| = \frac{k_1}{k_2}\left\|\mathbf{R}^{-1}(\bar{\mathbf{L}} + \bar{\boldsymbol{\Lambda}})\bar{\mathbf{x}}_e\right\| \leq \frac{k_1}{k_2|\ell|}\left\|\bar{\mathbf{L}} + \bar{\boldsymbol{\Lambda}}\right\|\left\|\bar{\mathbf{x}}_e\right\| \tag{23}$$

which indicates that the disturbance estimation error $\tilde{\boldsymbol{\delta}}$ is bounded. ■

Remark 2: With Proposition 1, we can guarantee the convergence of agents' positions. However, since $\bar{\mathbf{u}} = \mathbf{R}\mathbf{u}$, we can see that the proposed controller has effects on the orientations of agents. Therefore, we still need to analyze the convergence of the orientations of agents. The analysis is given by the following corollary.

Corollary 1: Drived by the proposed controller (15), agents' orientations will converge to some constant values.

Proof: Substracting (18) from (15), we can obtain $\bar{\mathbf{u}} - \dot{\bar{\mathbf{x}}} = -\mathbf{R}\mathbf{d}$. When the system reaches the stable stage, we have

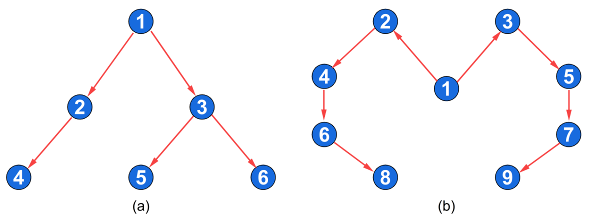

Fig. 3. Interaction graph of agents used for simulations and experiments: (a) Six agents; (b) Nine agents.

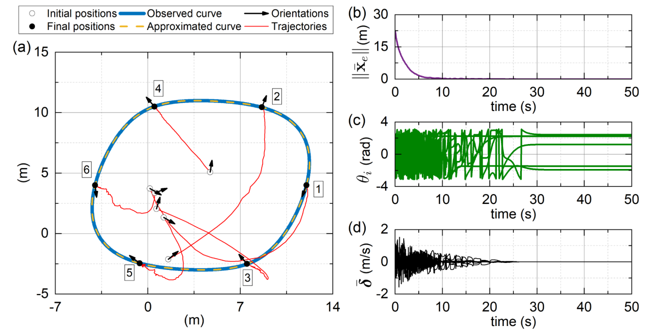

Fig. 4. Results of forming a closed curve: (a) Trajectories of the agents; (b) Norm of position errors; (c) Agents' orientations; (d) Disturbance estimation errors for (10).

$\dot{\bar{\mathbf{x}}}_e = \dot{\mathbf{x}} = \mathbf{0}$. Therefore, we can derive that $\bar{\mathbf{u}} \to -\mathbf{R}\mathbf{d}$ as $t \to \infty$. Since we have $\bar{\mathbf{u}} = \mathbf{R}\mathbf{u}$ and $\mathbf{R}$ is a full-rank matrix, we can obtain $\mathbf{u} = \mathbf{R}^{-1}\bar{\mathbf{u}} \to -\mathbf{d}$, that is, $\omega_i \to -d_{i2}$ considering $\mathbf{u}_i = [v_i, \omega_i]^\mathsf{T}$. Therefore, we can obtain $\dot{\theta}_i = \omega_i + d_{i2} \to 0$, which indicates the zero dynamics of the orientation of agents. Therefore, we can derive the result that the orientations of agents will converge to some bounded constant solutions. ■

IV. Results

A. Simulations

We first test our method by simulations. In real-world missions, the desired curves could be environmental boundaries that can be observed by appropriate sensors, such as cameras, and extracted from videos or images. However, for the convenience of simulations, we simply employ some predefined curves as the desired curves and call them observed curves. These observed curves are assumed to have an explicit form expressed in parametric equations $\mathbf{c} = f(\eta)$ for a parameter $\eta \in [0, 1]$. Note that η is different from s in that η could be parameters with any physical meanings while s is the length parameter. We sample points from these observed curves to estimate the approximated curves dependent on length parameters to be formed by agents.

Without loss of generality, we uniformly distribute the agents along the curve, i.e., $s_i = (i-1)/n$, in all tests. Without mentioning, we also adopt the same operation for the following experiments. The first two simulations are conducted for a multi-agent system with six agents ($n = 6$) and the last simulation is with nine agents ($n = 9$). During the simulations, agents are interacting through directed graphs shown in Fig. 3 and the parameters of estimated curves are broadcasted through the graph. We set the weights of all edges as one, the scalar for the change of coordinates as $\ell = 0.01$m, and the control gains as $k_1 = k_2 = 1$. The constant input disturbances of the agents are set to $\mathbf{d}_i = [0.1, 0.1]^\mathsf{T}$m/s. Agents depart from random initial positions with random initial orientations.

1) Closed Curve: The first simulation is conducted for a closed curve of the following form:

$$c_x = (8 + \sin 4\pi\eta)\cos 2\pi\eta + 4$$
$$c_y = (8 + \cos 4\pi\eta)\sin 2\pi\eta + 4 \tag{24}$$

We approximate (24) to a truncated Fourier series with six harmonics (i.e., $H = 13$, and Assumption 1 is satisfied) by (2). Matrix $\mathbf{G}_i(s_i)$ has a structure of $\mathbf{G}_i(s_i) = [\mathbf{g}_1, \ldots, \mathbf{g}_6, \mathbf{I}_2] \in \mathbb{R}^{2\times 26}$, where:

$$\mathbf{g}_h = \begin{bmatrix} \cos 2\pi h s_i & \sin 2\pi h s_i & 0 & 0 \\ 0 & 0 & \cos 2\pi h s_i & \sin 2\pi h s_i \end{bmatrix} \tag{25}$$

for $h = 1, \ldots, 6$. It can be checked that $\bar{\mathbf{G}} = [\mathbf{G}_i(s_i)^\mathsf{T}]^\mathsf{T}$ satisfies Assumption 2.

Running the simulation, we obtain the results shown in Fig. 4(a). We can see that agents depart from random positions and gradually converge to the desired curve. We record the norm of position errors and agents' orientations (rounded to $[-\pi, \pi]$) in Fig. 4(b) and (c), respectively. It can be seen that the errors asymptotically converge to zero and agents' orientations finally reach some constant values as we predicted in the stability analysis. The evolution of the disturbance estimation error $\bar{\boldsymbol{\delta}}$ for (10) is shown in Fig. 4(d). We can see that $\bar{\boldsymbol{\delta}}$ of all agents gradually converge to zero, which follows our analysis for the ideal cases in the proof of Proposition 1.

2) Open Curve: The second simulation is conducted for an open curve of the following form:

$$\mathbf{c} = \sum_{k=0}^{3} \eta^k (1-\eta)^{3-k} \mathbf{o}_{k+1} \tag{26}$$

for $\mathbf{o}_1 = [3.5, 3]^\mathsf{T}$, $\mathbf{o}_2 = [-0.5, -4]^\mathsf{T}$, $\mathbf{o}_3 = [-2, 6]^\mathsf{T}$, and $\mathbf{o}_4 = [-2, -1]^\mathsf{T}$. We approximate (26) to a sixth-order polynomial (i.e., $H = 7$, and Assumption 1 is satisfied) by (2). In this case, the matrix $\mathbf{G}_i(s_i)$ has a structure of $\mathbf{G}_i(s_i) = \mathbf{g}_i \otimes \mathbf{I}_2 \in \mathbb{R}^{2\times 14}$ for $\mathbf{g}_i = [1, s_i, \ldots, s_i^6]$. It can be checked that $\bar{\mathbf{G}} = [\mathbf{G}_i(s_i)^\mathsf{T}]^\mathsf{T}$ satisfies Assumption 2.

Running the simulation, we obtain the results shown in Fig. 5. Similarly to the first simulation, agents depart from random initial positions and gradually converge to the desired curve (see Fig. 5(a)). The position errors asymptotically converge to zero (see Fig. 5(b)) and agents finally reach some constant orientation values (see Fig. 5(c)) as predicted in the stability analysis. We record the disturbance estimation error $\bar{\boldsymbol{\delta}}$ for (10) in Fig. 5(d). We can see that $\bar{\boldsymbol{\delta}}$ gradually converges to zero, which follows our analysis for the ideal cases in the proof of Proposition 1.

3) Simulation With Curve Shape Changes: In real-world missions, the curve that agents need to form could vary between different patterns, and our method also has the potential to handle this case. In this simulation, we simulate the situation where

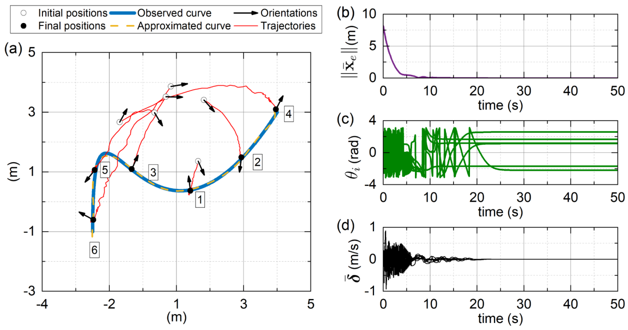

Fig. 5. Results of forming an open curve: (a) Trajectories of the agents; (b) Norm of position errors; (c) Agents' orientations; (d) Disturbance estimation errors for (10).

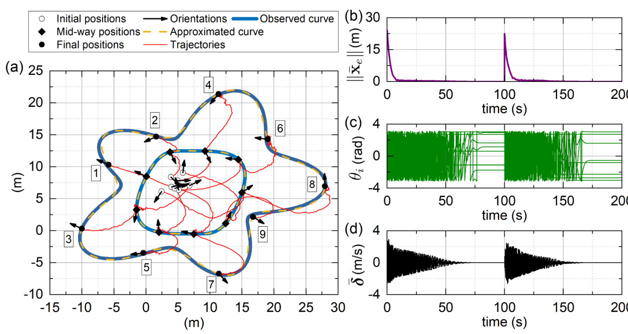

Fig. 6. Simulation with curve shape change at $t = 100$s: (a) Agents' trajectories; (b) Norm of position errors; (c) Agents' orientations; (d) Disturbance estimation errors for (10).

agents need to form one curve first and then move to another curve.

The simulation is conducted for a closed curve of the following form:

$$\begin{aligned} c_x &= (8 + 1.5 \sin 4\pi\eta) \cos 2\pi\eta + 7 \\ c_y &= (8 + 1.5 \cos 4\pi\eta) \sin 2\pi\eta + 4 \end{aligned} \tag{27}$$

for the first 100 seconds, then switched to another closed curve of the following form:

$$\begin{aligned} c_x &= (16 + 4 \cos 2\pi\eta + 2 \sin 4\pi\eta) \cos 2\pi\eta + 8 \\ c_y &= (12 + 3 \cos 2\pi\eta + 1.5 \sin 4\pi\eta) \sin 2\pi\eta + 6.75 \end{aligned} \tag{28}$$

after 100 seconds. We approximate both curves to a truncated Fourier series with eight harmonics (i.e. $H = 17$, and Assumption 1 is satisfied) by (2). Matrix $\mathbf{G}_i(s_i)$ has a structure of $\mathbf{G}_i(s_i) = [\mathbf{g}_1, \ldots, \mathbf{g}_6, \mathbf{I}_2] \in \mathbb{R}^{2\times 34}$, where $\mathbf{g}_h$ is in the form of (25). We can check that Assumption 2 is satisfied.

Running the simulation, we can obtain the results shown in Fig. 6(a). We can see that agents converge to the first desired curve and then to the second desired curve. This can also be seen in Fig. 6(b). The norm of the position errors asymptotically converges to zero when agents reach the first curve; then jumps up when the second desired curve appears, and asymptotically converges to zero again when agents reach the second curve. The orientations of agents (see Fig. 6(c)) reach some constant values as predicted in the stability analysis every time the agents reach a desired curve. We record the performance of the disturbance estimation error $\bar{\boldsymbol{\delta}}$ for (10) in Fig. 6(d). We can see that $\bar{\boldsymbol{\delta}}$ of all agents gradually converge to zero when the agents reach the first desired curve and converge to zero again when the agents reach the second desired curve, which follows our analysis for the ideal cases in the proof of Proposition 1.

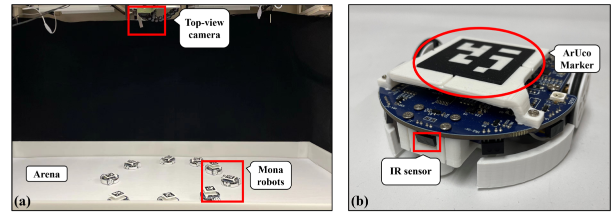

Fig. 7. The configuration of the experimental platform: (a) The platform; (b) The Mona robot.

B. Experiments

To verify the performance of our method in real-world tasks, we conduct several simple experimental tests with the platform shown in Fig. 7(a). The platform consists of nine ($n = 9$) Mona robots (see Fig. 7(b)) [27] with nonholonomic dynamics. The Mona robot is equipped with five infrared proximity sensors that can be used to perform simple obstacle avoidance tasks. In addition, the platform has a top-view camera to observe the pose of the robots, a control PC to collect data and send motion commands, and a 1.8 m×0.8 m arena for robots to move around. We build a Wi-Fi communication at a rate of 10 Hz between the control PC and robots based on ROS.

During the experiments, robots interact with each other through the topology shown in Fig. 3(b). We set the weights of all edges as one. We empirically set the gains of the controller to $k_1 = \frac{2}{3}$ and $k_2 = \frac{1}{3}$. We apply a constant input disturbance $\mathbf{d}_i = [0.5, 0.5]^\mathsf{T}$ cm/s to each robot through the control PC. Considering the hardware constraints of the robots, we limit the linear velocity applied to the robot wheels to ± 2.25 cm/s. The reported experimental results can be seen in the following web link: https://vimeo.com/872927697.

1) Closed Curves: The first experiment is conducted for a closed curve of the following form:

$$\begin{aligned} c_x &= 0.225(1 - \sin 2\pi\eta) \cos 2\pi\eta + 0.75 \\ c_y &= 0.225(1 - \sin 2\pi\eta) \sin 2\pi\eta + 0.675 \end{aligned} \tag{29}$$

for the first 60 seconds and switched to another closed curve of the following form:

$$\begin{aligned} c_x &= \frac{\cos 2\pi\eta}{100}(3 \sin 4\pi\eta + 6 \cos 10\pi\eta + 22.5) + 1.125 \\ c_y &= \frac{\sin 2\pi\eta}{100}(3 \sin 4\pi\eta + 6 \cos 10\pi\eta + 22.5) + 0.375 \end{aligned} \tag{30}$$

after 60 seconds. The approximations of both curves follow the same operations as in Section IV-A3. Therefore, both Assumptions 1 and 2 are satisfied.

The results of the experiment are shown in Fig. 8. We present screenshots of the experiment process in Fig. 8(a). It can be

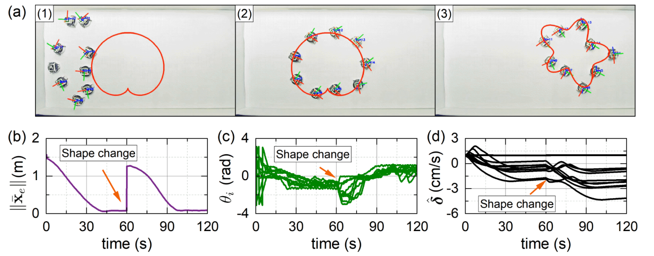

Fig. 8. Results of the experiment with closed curve. Curve's shape changes at $t = 60$s: (a) Screenshots of the experiment process; (b) Norm of position errors; (c) Agents' orientations; (d) Disturbance estimations.

seen that the robots start from random initial positions and successfully form two desired curves one after another. We record the norm of position errors and agents' orientations in Fig. 8(b) and (c), respectively. The norm of position errors converges to a stable value when agents reach the first desired curve; then jumps to a high value when the desired curve changes; and finally converges to a stable value again when agents reach the second desired curve. Agents' orientations gradually converge to some bounded values every time agents reach a desired curve. The performance of agents' position errors and orientations follows the analysis in Proposition 1. The disturbance estimations error measurement $\tilde{\boldsymbol{\delta}}$ of the agents are presented in Fig. 8(d). We can see that $\tilde{\boldsymbol{\delta}}$ gradually converges to some bounded values as agents converge to the desired curves, which is as predicted in the analysis for real-world cases in the proof of Proposition 1. This experiment proves the feasibility of our proposed method for closed curves in real-world formation tasks.

2) Open Curves: In this section, we present the experiment for an open curve with the following form:

$$\mathbf{c} = \sum_{k=0}^{3} \eta^k (1-\eta)^{3-k} \mathbf{o}_{k+1} + \mathbf{o}_5 \tag{31}$$

for $\mathbf{o}_1 = [0.525, 0.45]^\top$, $\mathbf{o}_2 = [-0.075, -0.6]^\top$, $\mathbf{o}_3 = [-0.3, 0.9]^\top$, $\mathbf{o}_4 = [-0.3, -0.15]^\top$, and $\mathbf{o}_5 = [0.9, 0.2625]^\top$. We approximate (31) to a ninth-order polynomial (i.e., $H = 10$, which satisfies Assumption 1) by (2). In this case, matrix $\mathbf{G}_i(s_i)$ is of the form of $\mathbf{G}_i(s_i) = \mathbf{g}_i \otimes \mathbf{I}_2 \in \mathbb{R}^{2\times 20}$ for $\mathbf{g}_i = [1, s_i, \ldots, s_i^9]$. Then,we can check that $\bar{\mathbf{G}} = [\mathbf{G}_i(s_i)^\top]^\top$ satisfies Assumption 2.

We present the experiment results in Fig. 9. Screenshots of the experiment process are shown in Fig. 9(a), through which we can see that the robots set off from random initial positions and successfully form the desired open curve. We record the norm of position errors and the orientations of agents in Fig. 9(b) and (c), respectively. It can be seen that agents' position errors gradually converge to zero and orientations finally reach some bounded constant values, which follows our analysis in the proof of Proposition 1. The disturbance estimations error measurement $\tilde{\boldsymbol{\delta}}$ of the agents are shown in Fig. 9(d). We can see that $\tilde{\boldsymbol{\delta}}$ are bounded as agents converge to the desired open curve, which follows the analysis for real-world cases in the proof of Proposition 1. This experiment proves the feasibility of our proposed method for open curves in real-world formation tasks.

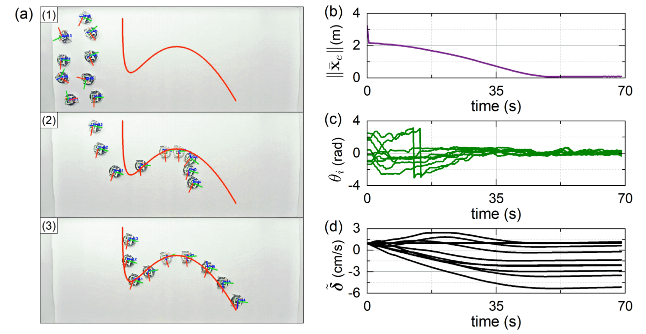

Fig. 9. Results of the experiment with open curve: (a) Trajectories of the agents; (b) Norm of position errors; (c) Agents' orientations; (d) Disturbance estimations.

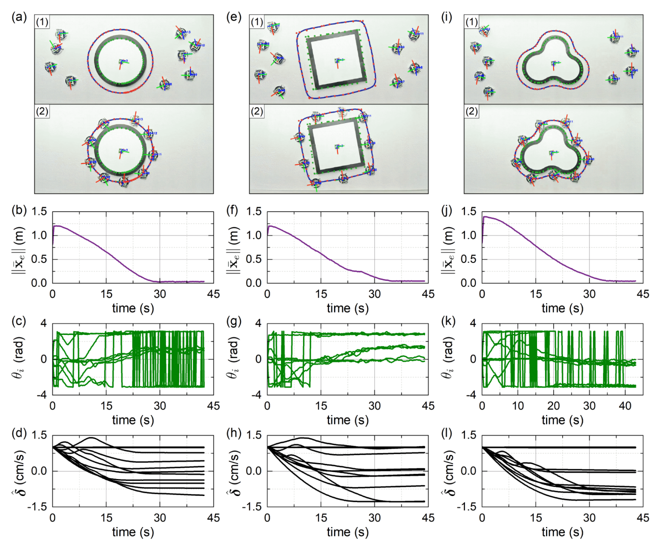

Fig. 10. Results of the experiments of enclosing objects with different shapes by the robots: (a), (e), and (i) screenshots of the experiment process; (b), (f), and (j) norm of position errors; (c), (g), and (k) agents' orientations; (d), (h), and (l) disturbance estimations.

3) Object Enclosing: Our proposed method has the potential to achieve the object enclosing in the object transport tasks with minor extensions. We conduct three simple experiments to showcase the versatility of the proposed method in dealing with enclosing objects with different shapes. Specifically, we manually make three objects using foam board, the shapes of which are circular, square, and "cloud"-like, respectively.

To obtain the desired curves for the agents to form, we first extract points of the edges of the objects (see green points in Fig. 10(a), (e), and (i)) through the picture captured by the top-view camera of the platform and represent them by polar coordinates relative to the geometric center of the objects. We achieve the point extraction by the mature algorithms from OpenCV [28]. Then we add an offset of 6 cm to the polar coordinates of the extracted points according to the configuration of the robots and obtain a bunch of new points with offset coordinates (see red points in Fig. 10(a), (e), and (i)); finally, we estimate a truncated Fourier series with eight harmonics by the points with offset coordinates and take it as the desired curve of agents. The estimation of the desired curve follows the same operations as in Section IV-A3 (see the blue curves in Fig. 10(a), (e), and (i)). Therefore, both Assumptions 2 and 1 are satisfied.

After we obtain the desired curve, we employ the proposed method to drive the agents to form it. The object enclosing is achieved after the agents reach the desired curve. To avoid collision between agents and objects, we employ the obstacle avoidance algorithm offered by the Mona robot using infrared proximity sensors. The experiment results are shown in Fig. 10. Screenshots of the experiment process are shown in Fig. 10(a), (e), and (i), respectively. We can see that agents set off from random initial positions and successfully enclose the target objects in each experiment. We also record the norm of position errors and agents' orientations for each experiment. As shown in Fig. 10(b), (f), and (j), the position errors gradually decrease to nearly zero as agents achieve the enclosing of target objects in all experiments. Agent orientations finally converge to some bounded constant values as predicted in Proposition 1 in each experiment, as shown in As shown in Fig. 10(c), (g), and (k), respectively. The evolutions of the disturbance estimations $\tilde{\boldsymbol{\delta}}$ of all agents are presented for all experiments in Fig. 10(d), (h), and (l), respectively. We can see that $\tilde{\boldsymbol{\delta}}$ converges to some bounded values, which follows the analysis for real-world cases in the proof of Proposition 1. These experiments prove the feasibility and versatility of the proposed method in dealing with object enclosing tasks.

V. Conclusion

In this work, we first present a unified parametric representation for general curves, that is, open and closed curves. Then, we derive a leader-follower formation controller based on the parametric equations to form the desired curves under directed communications and constant input disturbances. We propose a Lyapunov-based stability analysis to prove the asymptotic stability of the proposed controller. Numerical simulations and experimental studies show the desirable performance of our proposed method in dealing with different cases, such as open curves, closed curves, and curve shape changes. We also explore the potential of the proposed method in dealing with object enclosing tasks by detailed experiments. The method has high flexibility and the potential to be applied to various real-world missions. However, there are still some limits to the proposed method, such as dealing with continuously evolving curves and working with time delays. In the future, we will keep extending the proposed method to solve the aforementioned problems.

References

[1] R. Olfati-Saber, J. A. Fax, and R. M. Murray, “Consensus and cooperation in networked multi-agent systems,” *Proc. IEEE*, vol. 95, no. 1, pp. 215–233, Jan. 2007.

[2] Z. H. Ismail, N. Sariff, and E. G. Hurtado, “A survey and analysis of cooperative multi-agent robot systems: Challenges and directions,” in *Applications of Mobile Robots*. London, U.K.: IntechOpen, 2018, pp. 8–14.

[3] J. Alonso-Mora, S. Baker, and D. Rus, “Multi-robot formation control and object transport in dynamic environments via constrained optimization,” *Int. J. Robot. Res.*, vol. 36, no. 9, pp. 1000–1021, Aug. 2017.

[4] Y. Q. Chen and Z. Wang, “Formation control: A review and a new consideration,” in *Proc. 2005 IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Edmonton, AB, Canada, 2005, pp. 3181–3186.

[5] Q. Lu, Q. Han, B. Zhang, D. Liu, and S. Liu, “Cooperative control of mobile sensor networks for environmental monitoring: An event-triggered finite-time control scheme,” *IEEE Trans. Cybern.*, vol. 47, no. 12, pp. 4134–4147, Dec. 2017.

[6] F. L. Iudice, F. Garofalo, and P. D. Lellis, “Formation control on a closed curve under an intermittent measurement flow,” *IEEE Control Syst. Lett.*, vol. 6, pp. 656–661, 2022.

[7] K. K. Oh, M. C. Park, and H. S. Ahn, “A survey of multi-agent formation control,” *Automatica*, vol. 53, pp. 424–440, Mar. 2015.

[8] Y. H. Choi and D. Kim, “Distance-based formation control with goal assignment for global asymptotic stability of multi-robot systems,” *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 2020–2027, Apr. 2021.

[9] S. Park and S. M. Lee, “Formation reconfiguration control with collision avoidance of nonholonomic mobile robots,” *IEEE Robot. Automat. Lett.*, vol. 8, no. 12, pp. 7905–7912, Dec. 2023.

[10] R. Tallamraju et al., “Active perception based formation control for multiple aerial vehicles,” *IEEE Robot. Automat. Lett.*, vol. 4, no. 4, pp. 4491–4498, Oct. 2019.

[11] D. P. Scharf, F. Y. Hadaegh, and S. R. Ploen, “A survey of spacecraft formation flying guidance and control. PART II: Control,” in *Proc. 2004 Amer. Control Conf.*, Boston, MA, USA, 2004, pp. 2976–2985.

[12] H. Wu, B. Jayawardhana, H. G. de Marina, and D. Xu, “Distributed formation control for manipulator end-effectors,” *IEEE Trans. Autom. Control*, vol. 68, no. 9, pp. 5413–5428, Sep. 2023.

[13] L. Calderone, “Using robots to clean oil spills,” *Robot. Tomorrow*, Accessed: Dec. 1, 2013. [Online]. Available: https://www.roboticstomorrow.com/article/2013/12/using-robots-to-clean-oil-spills/215/

[14] F. Zhang and N. E. Leonard, “Coordinated patterns of unit speed particles on a closed curve,” *Syst. Control Lett.*, vol. 56, no. 6, pp. 397–407, Jun. 2007.

[15] M. A. Hsieh, V. Kumar, and L. Chaimowicz, “Decentralized controllers for shape generation with robotic swarms,” *Robotica*, vol. 26, no. 5, pp. 691–701, Sep. 2008.

[16] C. Song, Y. Fan, and S. Xu, “Finite-time coverage control for multiagent systems with unidirectional motion on a closed curve,” *IEEE Trans. Cybern.*, vol. 51, no. 6, pp. 3071–3078, Jun. 2021.

[17] B. Zhang et al., “Fourier-based multi-agent formation control to track evolving closed boundaries,” *IEEE Trans. Circuits Syst. I., Reg. Papers*, vol. 70, no. 11, pp. 4549–4559, Nov. 2023.

[18] M. Aranda, “Formation control on planar closed curves using Fourier descriptors,” *IEEE Control Syst. Lett.*, vol. 7, pp. 3391–3396, 2023.

[19] D. Saldaña, R. Assunção, M. A. Hsieh, M. F. Campos, and V. Kumar, “Estimating boundary dynamics using robotic sensor networks with pointwise measurements,” *Auton. Robots*, vol. 45, pp. 193–208, 2021.

[20] J. W. Wang and Y. Guo, “Leaderless cooperative control of robotic sensor networks for monitoring dynamic pollutant plumes,” *IET Control Theory Appl.*, vol. 13, no. 16, pp. 2670–2680, Nov. 2019.

[21] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, “Mobile robots,” in *Robotics: Modelling, Planning and Control*, 1st ed. London, U.K.: Springer, 2009, pp. 507–508.

[22] H. Becerra, J. A. Colunga, and J. G. Romero, “Simultaneous convergence of position and orientation of wheeled mobile robots using trajectory planning and robust controllers,” *Int. J. Adv. Robot. Syst.*, vol. 15, no. 1, pp. 1–18, Jan. 2018.

[23] J. G. Romero and H. Rodríguez, “Asymptotic stability for a transformed nonlinear UAV model with a suspended load via energy shaping,” *Eur. J. Control*, vol. 52, pp. 87–96, Mar. 2020.

[24] M. Mesbahi and M. Egerstedt, “Graph theory,” in *Graph Theoretic Methods in Multiagent Networks*. Princeton, NJ, USA: Princeton Univ. Press, 2010, pp. 14–41.

[25] E. Panteley, A. Loria, and S. Sukumar, “Strict Lyapunov functions for consensus under directed connected graphs,” in *Proc. 2020 Eur. Control Conf.*, St Petersburg, Russia, 2020, pp. 935–940.

[26] H. Zhang, Z. Li, Z. Qu, and F. L. Lewis, “On constructing Lyapunov functions for multi-agent systems,” *Automatica*, vol. 58, pp. 39–42, 2015.

[27] F. Arvin, J. Espinosa, B. Bird, A. West, S. Watson, and B. Lennox, “MONA: An affordable open-source mobile robot for education and research,” *J. Intell. Robotic Syst.*, vol. 94, pp. 761–775, Jun. 2019.

[28] G. Bradski, “The OpenCV library,” *Dr Dobb’s J. Softw. Tools Professional Programmer*, vol. 25, pp. 120–123, 2000.