

# Leader-Follower Formation Control of Perturbed Nonholonomic Agents along Parametric Curves with Directed Communication

Bin Zhang, Xiaodong Shao, Hui Zhi, Liuming Qiu, Jose Guadalupe Romero, and David Navarro-Alarcon

**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**—Parametric curves; Multi-agent systems; Formation control; Directed graphs; Disturbance rejection.

## 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.

In classical formation control methods, the desired patterns that agents need to form are usually specified by some 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

This work is supported by the Research Grants Council under grant 15212721 and 15231023. (*Corresponding author: D. Navarro-Alarcon*)

B. Zhang, H. Zhi, L. Qiu and D. Navarro-Alarcon are with Department of Mechanical Engineering, The Hong Kong Polytechnic University (PolyU), KLN, Hong Kong.

X. Shao is with the School of Automation Science and Electrical Engineering, Beihang University (BUAA), Beijing, China.

J. G. Romero is with the Department of Electrical and Electronic Engineering, Instituto Tecnológico Autónomo de México (ITAM), CDMX, Mexico.

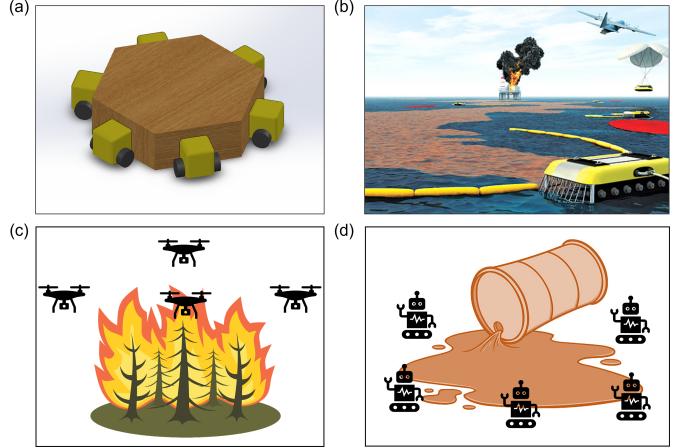


Fig. 1. Conceptual applications of multi-agent systems: (a) robots enclosing an object; (b) robots monitoring spilled oil using AEROS robots [13]; (c) drones monitoring wildfires; (d) robots cleaning leaked chemicals.

distribution rather than the specific positions of agents, such as when caging an object during transportation tasks or when 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 high 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 the 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 performance of the proposed method. In the experimental validations, we not only examine 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 mathematical 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 as bold letters, while 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 matrix of zeros.  $\mathbf{I}_n$  denotes a  $n \times n$  identity matrix. We use the symbol  $\min()$  to represent the operation of finding the minimum and  $\text{diag}(\cdot)$  to represent the diagonalization operation, where “.” 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:

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

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  $\xi \in \mathbb{R}^{2H \times 1}$  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:

$$\mathbf{C} = [\mathbf{c}_1^\top, \dots, \mathbf{c}_N^\top]^\top \quad (2)$$

where  $N$  is the total number of sampled points. Then, appropriately select a series of basis functions and compute the basis matrix corresponding to each sampled point, denoted as:

$$\mathbf{G}_h = [\mathbf{G}_1^\top, \dots, \mathbf{G}_N^\top]^\top \quad (3)$$

Finally, we can estimate the curve coefficients by:

$$\xi = (\mathbf{G}_h^\top \mathbf{G}_h)^{-1} \mathbf{G}_h \mathbf{C} \quad (4)$$

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  $\theta_i$  are the center position and the orientation of the agent, respectively. The dynamic model of agent  $i$  is given by:

$$\begin{bmatrix} \dot{x}_i \\ \dot{y}_i \\ \dot{\theta}_i \end{bmatrix} = \begin{bmatrix} \cos \theta_i & 0 \\ \sin \theta_i & 0 \\ 0 & 1 \end{bmatrix} (\mathbf{u}_i + \mathbf{d}_i) \quad (5)$$

where  $\mathbf{u}_i = [v_i, \omega_i]^\top$  denotes the control input, with  $v_i$  and  $\omega_i$  as the linear and angular velocities of the agent, respectively;  $\mathbf{d}_i = [d_{i1}, d_{i2}]^\top$  denotes the constant input disturbance.

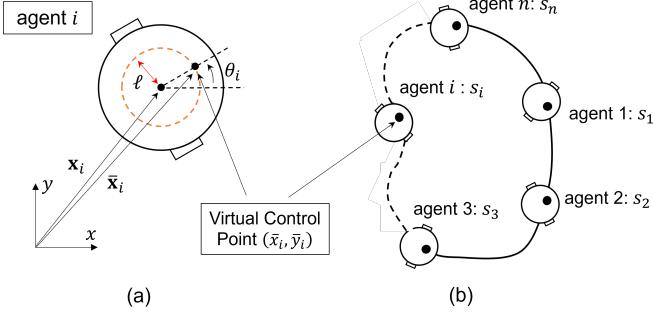


Fig. 2. The configuration of the agent: (a) the change of coordinates; (b) the assignment of length parameters for agents.

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]):

$$\begin{cases} \bar{x}_i := x_i + \ell \cos \theta_i \\ \bar{y}_i := y_i + \ell \sin \theta_i \\ \bar{\theta}_i := \theta_i \end{cases} \quad (6)$$

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  $\bar{\mathbf{x}}_i = [\bar{x}_i, \bar{y}_i]^T$ , whose time derivative yields a shifted dynamic model of the form:

$$\begin{cases} \dot{\bar{x}}_i = \mathbf{R}_i(\theta_i)(\mathbf{u}_i + \mathbf{d}_i) \\ \dot{\bar{\theta}}_i = \omega_i + d_{i2} \end{cases} \quad (7)$$

for a full-rank matrix defined as:

$$\mathbf{R}_i(\theta_i) = \begin{bmatrix} \cos \theta_i & -\ell \sin \theta_i \\ \sin \theta_i & \ell \cos \theta_i \end{bmatrix} \quad (8)$$

We can easily derive  $\|\mathbf{R}_i\| = 1$  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 (7) can be reduced to a single integrator with input disturbance:

$$\dot{\bar{\mathbf{x}}}_i = \bar{\mathbf{u}}_i + \bar{\mathbf{d}}_i \quad (9)$$

### C. Interaction Topology

The interactions topology among the agents can be represented by a graph  $\mathcal{G} = (\mathcal{G}, \mathcal{E}, a_{ij})$ , where  $\mathcal{V} = \{1, \dots, n\}$  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:

$$l_{ij} = \begin{cases} \sum_{i=1}^n a_{ij}, & \text{if } i = j \\ -a_{ij}, & \text{if } i \neq j \end{cases} \quad (10)$$

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

$$\begin{cases} \mathbf{q} = [q_i]^T = (\mathbf{L} + \mathbf{B})^{-1} \mathbf{1}_n \\ \mathbf{p} = [p_i]^T = (\mathbf{L} + \mathbf{B})^{-T} \mathbf{1}_n \\ \mathbf{P} = \text{diag}(p_i/q_i) \\ \mathbf{Q} = \mathbf{P}(\mathbf{L} + \mathbf{B}) + (\mathbf{L} + \mathbf{B})^T \mathbf{P} \end{cases} \quad (11)$$

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

## III. PROPOSED CONTROL METHOD

### A. Controller Design

By stacking the position dynamics (9) of agents, we can represent the position dynamics of the multi-agent system as:

$$\dot{\bar{\mathbf{x}}} = \bar{\mathbf{u}} + \bar{\mathbf{d}} \quad (12)$$

where  $\bar{\mathbf{x}} = [\bar{x}_1^T, \dots, \bar{x}_n^T]^T$ ,  $\bar{\mathbf{u}} = [\bar{u}_1^T, \dots, \bar{u}_n^T]^T$ , and  $\bar{\mathbf{d}} = [\bar{d}_1^T, \dots, \bar{d}_n^T]^T \in \mathbb{R}^{2n \times 1}$  are the extended position vector, input vector, and disturbance vector, respectively.

We uniformly assign length parameters to each agent to ensure the equal arc length separation of agents along the curve. For that, we set a sequence of length parameters corresponding to the agents  $\mathbf{s} = [s_1, \dots, s_i, \dots, s_n]^T$ , where  $s_i$  is the length parameter corresponding to the  $i$ -th agent, calculated by  $s_i = (i-1)/n$ . The process is shown in Fig. 2(b). With that, we can calculate the matrix of basis functions corresponding to each agent as:

$$\mathbf{G} = [\mathbf{G}_1(s_1)^T, \dots, \mathbf{G}_n(s_n)^T]^T \quad (13)$$

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

$$\bar{\mathbf{x}}_e = \bar{\mathbf{x}} - \bar{\mathbf{G}} \boldsymbol{\xi} \quad (14)$$

and the curve coefficient errors as:

$$\boldsymbol{\xi}_e = \bar{\mathbf{G}}^+ \bar{\mathbf{x}} - \boldsymbol{\xi} \quad (15)$$

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:

$$\bar{\mathbf{u}}_i = \begin{cases} -k_1(\bar{\mathbf{x}}_i - \mathbf{G}_i \boldsymbol{\xi}) - k_2 \mathbf{R}_i \hat{\boldsymbol{\delta}}_i & \text{leader} \\ -k_1 \sum_{j \in \mathcal{N}_i} a_{ij} (\mathbf{G}_i - \mathbf{G}_j) \boldsymbol{\xi}_e - k_2 \mathbf{R}_i \hat{\boldsymbol{\delta}}_i & \text{follower} \end{cases} \quad (16)$$

where  $k_1, k_2 > 0$  are the control gains,  $\hat{\delta}_i$  denotes the estimation of 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:

$$\dot{\hat{\delta}}_i = k_2 \mathbf{R}_i^T (\bar{\mathbf{x}}_i - \mathbf{G}_i \xi) \quad (17)$$

### 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 (16) for each agent to obtain the control input  $\bar{\mathbf{u}}$  for the whole multi-agent system:

$$\bar{\mathbf{u}} = -k_1(\bar{\mathbf{L}}\bar{\mathbf{G}}\xi_e + \bar{\Lambda}\bar{\mathbf{x}}_e) - k_2\mathbf{R}\hat{\delta} \quad (18)$$

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

Replacing the controller (18) into the stacked dynamics (19), we can obtain the complete dynamics of the multi-agent system:

$$\dot{\bar{\mathbf{x}}} = -k_1(\bar{\mathbf{L}}\bar{\mathbf{G}}\xi_e + \bar{\Lambda}\bar{\mathbf{x}}_e) - k_2\mathbf{R}\hat{\delta} + \bar{\mathbf{d}} \quad (19)$$

Now, we define a disturbance estimation error measurement:

$$\tilde{\delta} = \hat{\delta} - \frac{1}{k_2}\mathbf{d} \quad (20)$$

for  $\mathbf{d} = [\mathbf{d}_i^T]^T \in \mathbb{R}^{2n \times 1}$ . Then, we can rewrite (19) as:

$$\dot{\bar{\mathbf{x}}} = -k_1(\bar{\mathbf{L}}\bar{\mathbf{G}}\xi_e + \bar{\Lambda}\bar{\mathbf{x}}_e) - k_2\mathbf{R}\tilde{\delta} \quad (21)$$

It can be derived that:

$$\dot{\tilde{\delta}} = \dot{\hat{\delta}} = k_2\mathbf{R}^T\bar{\mathbf{x}}_e \quad (22)$$

considering that  $\mathbf{d}$  is constant. Let  $\bar{\delta} = \mathbf{R}\tilde{\delta}$ . We can see that  $\bar{\delta}$  measures the disturbance estimation error for (12).

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

**Assumption 1.** The rank of matrix  $\bar{\mathbf{G}} \in \mathbb{R}^{2n \times 2H}$  always satisfies  $\text{rank}(\bar{\mathbf{G}}) = \min\{2n, 2H\}$ .

**Assumption 2.** The number of agents and basis functions satisfies  $n \leq H$ .

With Assumption 1, we can ensure the existence of the pseudoinverse  $\bar{\mathbf{G}}^+$ . Specifically, we can compute  $\bar{\mathbf{G}}^+$  by:

$$\bar{\mathbf{G}}^+ = \begin{cases} (\bar{\mathbf{G}}^T\bar{\mathbf{G}})^{-1}\bar{\mathbf{G}}^T & n \geq H \\ \bar{\mathbf{G}}^T(\bar{\mathbf{G}}\bar{\mathbf{G}}^T)^{-1} & n < H \end{cases} \quad (23)$$

With Assumption 2, we can obtain  $\bar{\mathbf{G}}\bar{\mathbf{G}}^+ = \mathbf{I}_{2n}$ . Then, we can derive that:

$$\bar{\mathbf{G}}\xi_e = \bar{\mathbf{G}}\bar{\mathbf{G}}^+\bar{\mathbf{x}}_e = \bar{\mathbf{x}}_e \quad (24)$$

**Remark 1.** The condition proposed in Assumption 2 is reasonable. We usually need more basis functions to accurately

approximate a curve with a complex shape structure. As for “simple” curves, such as straight lines and circles, we can also 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 simple straight line as:

$$\mathbf{c} = \mathbf{a} + \mathbf{b}s + \mathbf{0}(s^2 + s^3 + \dots) \quad (25)$$

for  $\mathbf{a}$  and  $\mathbf{b} \in \mathbb{R}^{2 \times 1}$  as coefficients.

**Proposition 1.** Consider a multi-agent system of  $n$  agents with dynamics (9) and a directed graph containing a rooted spanning tree. Given a *static* planar curve in the form of (1), the controller (18) drives the agents to be uniformly distributed on the curve and ensures the asymptotic stability of position errors  $\bar{\mathbf{x}}_e$ . The orientations of the agents will converge to some constant values.

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

$$V = \bar{\mathbf{x}}_e^T(\mathbf{P} \otimes \mathbf{I}_2)\bar{\mathbf{x}}_e + \tilde{\delta}^T(\mathbf{P} \otimes \mathbf{I}_2)\tilde{\delta} \quad (26)$$

Compute the time derivative of (26), then we obtain:

$$\begin{aligned} \dot{V} &= 2\bar{\mathbf{x}}_e^T(\mathbf{P} \otimes \mathbf{I}_2)\dot{\bar{\mathbf{x}}}_e + 2\tilde{\delta}^T(\mathbf{P} \otimes \mathbf{I}_2)\dot{\tilde{\delta}} \\ &= 2\bar{\mathbf{x}}_e^T(\mathbf{P} \otimes \mathbf{I}_2)[-k_1(\bar{\mathbf{L}}\bar{\mathbf{G}}\xi_e + \bar{\Lambda}\bar{\mathbf{x}}_e) - k_2\mathbf{R}\tilde{\delta}] \\ &\quad + 2\tilde{\delta}^T(\mathbf{P} \otimes \mathbf{I}_2)k_2\mathbf{R}^T\bar{\mathbf{x}}_e \\ &= -2k_1\bar{\mathbf{x}}_e^T(\mathbf{P} \otimes \mathbf{I}_2)(\mathbf{L} \otimes \mathbf{I}_2 + \Lambda \otimes \mathbf{I}_2)\bar{\mathbf{x}}_e \\ &\quad - 2k_2\bar{\mathbf{x}}_e^T[(\mathbf{P} \otimes \mathbf{I}_2)\mathbf{R} - \mathbf{R}(\mathbf{P} \otimes \mathbf{I}_2)]\tilde{\delta} \end{aligned} \quad (27)$$

where we use (24) to obtain the last equality. Checking the structure of  $\mathbf{R}$  and  $\mathbf{P} \otimes \mathbf{I}_2$ , we can show that:

$$(\mathbf{P} \otimes \mathbf{I}_2)\mathbf{R} = \text{diag}\left(\frac{p_i}{q_i}\mathbf{I}_2\mathbf{R}_i\right) = \mathbf{R}(\mathbf{P} \otimes \mathbf{I}_2) \quad (28)$$

Therefore, we have:

$$\dot{V} = -\bar{\mathbf{x}}_e^T(\mathbf{Q} \otimes \mathbf{I}_2)\bar{\mathbf{x}}_e \leq 0 \quad (29)$$

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  $\dot{\bar{\mathbf{x}}}_e = \dot{\mathbf{x}}$  from (14), we can derive that  $\bar{\delta} = \mathbf{0}$  and  $\tilde{\delta} = \mathbf{0}$  are asymptotically stable from (21). In real-world cases,  $\bar{\mathbf{x}}_e$  usually can not strictly converge to zero when the system reaches the stable stage. From (21), we can derive the following relation:

$$\begin{aligned} \|\tilde{\delta}\| &= \frac{k_1}{k_2} \|\mathbf{R}^{-1}(\bar{\mathbf{L}} + \bar{\Lambda})\bar{\mathbf{x}}_e\| \\ &\leq \frac{k_1}{k_2|\ell|} \|\bar{\mathbf{L}} + \bar{\Lambda}\| \|\bar{\mathbf{x}}_e\| \end{aligned} \quad (30)$$

Therefore, we can show that the disturbance estimation error  $\tilde{\delta}$  is bounded. Now, we consider the evolution of the orientations of the agents. Subtracting (21) from (18), we can obtain  $\bar{\mathbf{u}} - \dot{\mathbf{x}} = -\mathbf{R}\mathbf{d}$ . When the system reaches the stable stage, we have  $\dot{\bar{\mathbf{x}}}_e = \dot{\mathbf{x}} = \mathbf{0}$ . Therefore, we can derive that  $\bar{\mathbf{u}} \rightarrow -\mathbf{R}\mathbf{d}$

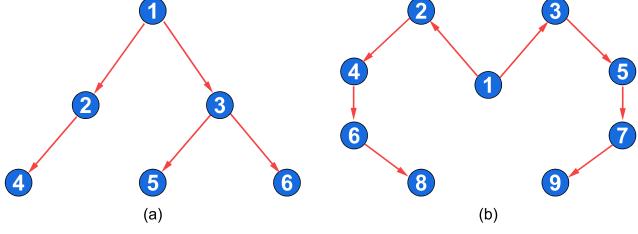


Fig. 3. Interaction graph of agents used for simulations and experiments: (a) graph for six agents; (b) graph for nine agents.

as  $t \rightarrow \infty$ . Considering  $\mathbf{u} = \mathbf{R}^{-1}\bar{\mathbf{u}}$  and  $\mathbf{u}_i = [v_i, \omega_i]^\top$ , we can obtain:

$$\dot{\theta}_i = \omega_i + d_{i2} \rightarrow 0 \quad (31)$$

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. Simulation with Single Curves

We first test our method by driving agents to form a single static curve. The simulations are conducted for a multi-agent system with six agents ( $n = 6$ ) interacting through a directed graph shown in Fig. 3(a). We set the weights of all edges as one, the scalar for the change of coordinates as  $\ell = 0.01\text{m}$ , and the control gains as  $k_1 = k_2 = 1$ . The constant input disturbances of the agents are set to  $\mathbf{d}_i = [0.1, 0.1]^\top \text{m/s}$ . 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:

$$\begin{aligned} c_x &= (8 + \sin 4\pi s) \cos 2\pi s + 4 \\ c_y &= (8 + \cos 4\pi s) \sin 2\pi s + 4 \end{aligned} \quad (32)$$

We approximate (32) to a truncated Fourier series with six harmonics (i.e.,  $H = 13$ , and Assumption 2 is satisfied) by (4). The matrix  $\mathbf{G}_i(s_i)$  has a structure of  $\mathbf{G}_i(s_i) = [\mathbf{g}_1, \dots, \mathbf{g}_6, \mathbf{I}_2] \in \mathbb{R}^{2 \times 26}$ , where:

$$\mathbf{g}_h = \begin{bmatrix} \cos 2\pi h s_i & \sin 2\pi h s_i & 0 & 0 \\ 0 & 0 & \cos 2\pi h s_i & \sin 2\pi h s_i \end{bmatrix} \quad (33)$$

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

Running the simulation, we obtain the results shown in Fig. 4(a). We can see that the agents depart from random positions and gradually converge to the desired curve. We record the norm of position errors and the evolution of agents' orientation (rounded to  $[-\pi, \pi]$ ) in Fig. 4(b) and Fig. 4(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  $\bar{\delta}$  for (12) is shown in Fig. 4(d). We can see that  $\bar{\delta}$  of all agents gradually converge to zero, which follows our analysis for the ideal cases in the proof of Proposition 1.

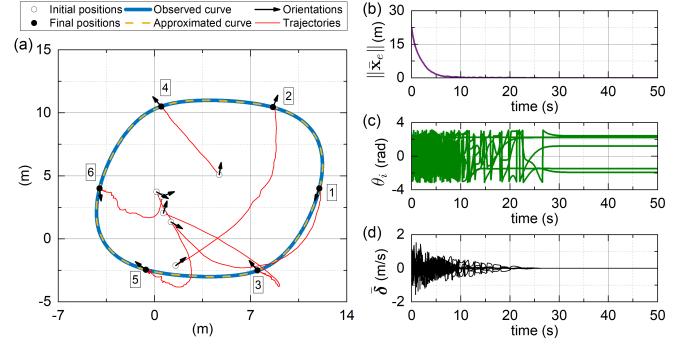


Fig. 4. Results of forming a closed curve: (a) the trajectories of the agents; (b) the evolution of the norm of position errors; (c) the evolution of agents' orientations; (d) the evolution of disturbance estimation errors for (12).

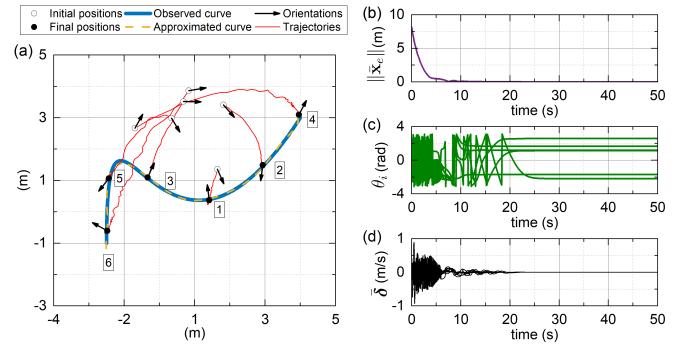


Fig. 5. Results of forming an open curve: (a) the trajectories of the agents; (b) the evolution of the norm of position errors; (c) the evolution of agents' orientations; (d) the evolution of disturbance estimation errors for (12).

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

$$\mathbf{c} = (1 - s)^3 \mathbf{o}_1 + 3s(1 - s)^2 \mathbf{o}_2 + 3s^2(1 - s) \mathbf{o}_3 + s^3 \mathbf{o}_4 \quad (34)$$

for  $\mathbf{o}_1 = [3.5, 3]^\top$ ,  $\mathbf{o}_2 = [-0.5, -4]^\top$ ,  $\mathbf{o}_3 = [-2, 6]^\top$ , and  $\mathbf{o}_4 = [-2, -1]^\top$ . We approximate (34) to a sixth-order polynomial (i.e.,  $H = 7$ , and Assumption 2 is satisfied) by (4). 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  $\mathbf{g}_i = [1, s_i, \dots, s_i^6]$ . It can be checked that  $\bar{\mathbf{G}} = [\mathbf{G}_i(s_i)^\top]^\top$  satisfies Assumption 1.

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  $\bar{\delta}$  for (12) in Fig. 5(d). We can see that  $\bar{\delta}$  gradually converges to zero, which follows our analysis for the ideal cases in the proof of Proposition 1.

##### B. 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 agents need to form one curve

first and then move to another curve. We adopt a multi-agent system of nine agents ( $n = 9$ ) interacting through a directed graph shown in Fig. 3(b). We keep the simulation settings the same as in the previous sections.

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

$$\begin{aligned} c_x &= (8 + 1.5 \sin 4\pi s) \cos 2\pi s + 7 \\ c_y &= (8 + 1.5 \cos 4\pi s) \sin 2\pi s + 4 \end{aligned} \quad (35)$$

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

$$\begin{aligned} c_x &= (16 + 4 \cos 2\pi s + 2 \sin 4\pi s) \cos 2\pi s + 8 \\ c_y &= (12 + 3 \cos 2\pi s + 1.5 \sin 4\pi s) \sin 2\pi s + 6.75 \end{aligned} \quad (36)$$

after 100 seconds. We approximate both curves to a truncated Fourier series with eight harmonics (i.e.  $H = 17$ , and Assumption 2 is satisfied) by (4). The matrix  $\mathbf{G}_i(s_i)$  has a structure of  $\mathbf{G}_i(s_i) = [\mathbf{g}_1, \dots, \mathbf{g}_6, \mathbf{I}_2] \in \mathbb{R}^{2 \times 34}$ , where  $\mathbf{g}_h$  is in the form of (33). We can check that Assumption 1 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  $\bar{\delta}$  for (12) in Fig. 6(d). We can see that  $\bar{\delta}$  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.

**Remark 2.** In real-world missions, the desired curves could be environmental boundaries or the perimeters of objects, etc., that can be observed by appropriate sensors, such as cameras, and extracted from videos or images. However, our core problem in this study is developing a control method for forming those curves. Therefore, we simply predefine some desired curves to pretend that we have already obtained the desired curves from observation for the convenience of simulations and call them observed curves.

### C. 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  $1.8m \times 0.8m$  arena for robots

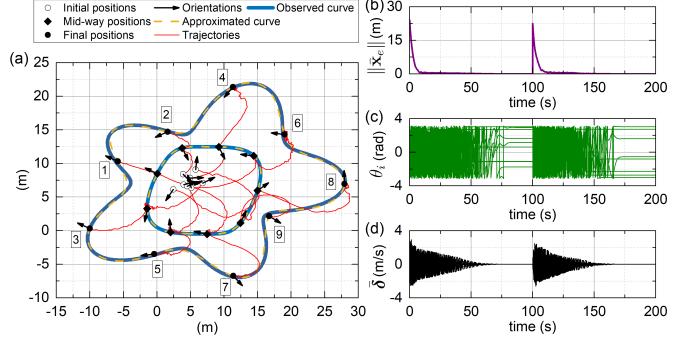


Fig. 6. Results of the simulation with curve shape changes at  $t = 100$ s: (a) the trajectories of the agents; (b) the evolution of the norm of position errors; (c) the evolution of agents' orientations; (d) the evolution of disturbance estimation errors for (12).

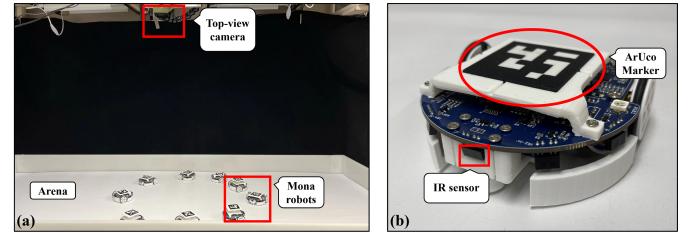


Fig. 7. The configuration of the experimental platform: (a) the platform; (b) the Mona robot.

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 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  $\mathbf{d}_i = [0.5, 0.5]^T$  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  $\pm 2.25$  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:

$$\begin{aligned} c_x &= 0.225(1 - \sin 2\pi s) \cos 2\pi s + 0.75 \\ c_y &= 0.225(1 - \sin 2\pi s) \sin 2\pi s + 0.675 \end{aligned} \quad (37)$$

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

$$\begin{aligned} c_x &= \frac{1}{100}(3 \sin 4\pi s + 6 \cos 10\pi s + 22.5) \cos 2\pi s + 1.125 \\ c_y &= \frac{1}{100}(3 \sin 4\pi s + 6 \cos 10\pi s + 22.5) \sin 2\pi s + 0.375 \end{aligned} \quad (38)$$

after 60 seconds. The approximations of both curves follow the same operations as in Section IV-B. Therefore, both Assumption 1 and Assumption 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 seen that the robots start from random initial positions and successfully form two desired curves one after another. We

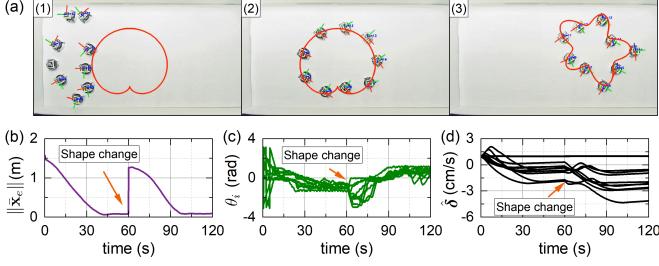


Fig. 8. Results of the experiment with closed curve. Curve's shape changes at  $t = 60$ s: (a) the screenshots of the experiment process; (b) the evolution of the norm of position errors; (c) the evolution of agents' orientations; (d) the evolution of disturbance estimations.

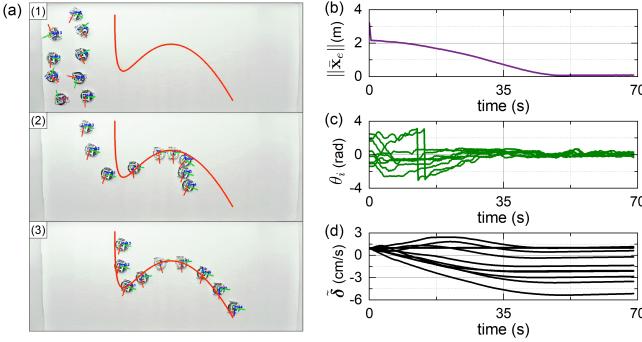


Fig. 9. Results of the experiment with open curve: (a) the trajectories of the agents; (b) the evolution of the norm of position errors; (c) the evolution of agents' orientations; (d) the evolution of disturbance estimations.

record the norm of position errors and agents' orientations in Fig. 8(b) and Fig. 8(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  $\tilde{\delta}$  of the agents are presented in Fig. 8(d). We can see that  $\tilde{\delta}$  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:

$$\mathbf{c} = (1-s)^3 \mathbf{o}_1 + 3s(1-s)^2 \mathbf{o}_2 + 3s^2(1-s) \mathbf{o}_3 + s^3 \mathbf{o}_4 + \mathbf{o}_5 \quad (39)$$

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 (39) to a ninth-order polynomial (i.e.,  $H = 10$ , which satisfies Assumption 2) by (4). In this case, the 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  $\mathbf{g}_i = [1, s_i, \dots, s_i^9]^\top$ . Then, we can check that  $\tilde{\mathbf{G}} = [\mathbf{G}_i(s_i)^\top]^\top$  satisfies Assumption 1.

We present the experiment results in Fig. 9. Screenshots of the experiment process are shown in Fig. 9(a), through which

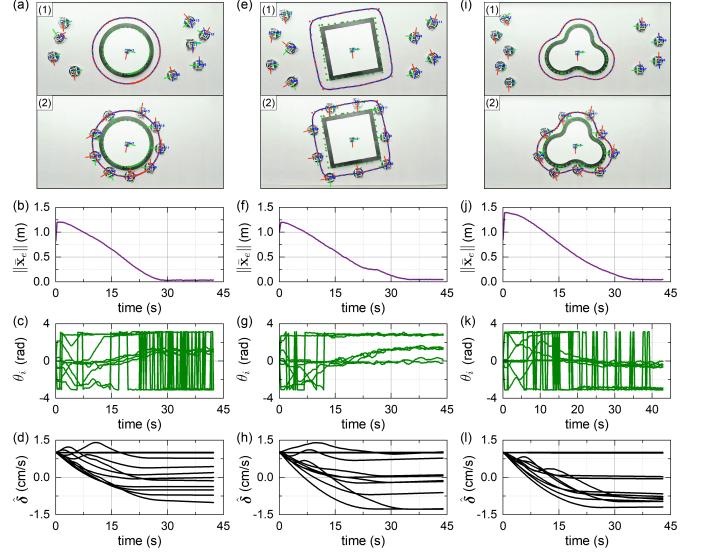


Fig. 10. Results of the experiments of enclosing objects with different shapes by the robots: (a)(e)(i) the screenshots of the experiment process; (b)(f)(j) the evolution of the norm of position errors; (c)(g)(k) the evolution of agents' orientations; (d)(h)(l) the evolution of disturbance estimations.

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 Fig. 9(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  $\tilde{\delta}$  of the agents are shown in Fig. 9(d). We can see that  $\tilde{\delta}$  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.

3) *Object Enclosing:* Our proposed method also has the potential to achieve the object enclosing tasks 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-B (see the blue curves in Fig. 10(a), (e), and (i)). Therefore, both Assumption 1 and Assumption 2 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), Fig. 10(e), and Fig. 10(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), Fig. 10(f), and Fig. 10(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 Fig. 10(c), Fig. 10(g), and Fig. 10(k), respectively. The evolutions of the disturbance estimations  $\tilde{\delta}$  of all agents are presented for all experiments in Fig. 10(d), Fig. 10(h), and Fig. 10(l), respectively. We can see that  $\tilde{\delta}$  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*, 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. Rob. 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, Aug. 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. Lo Iudice, F. Garofalo, and P. De Lellis, "Formation control on a closed curve under an intermittent measurement flow," *IEEE Contr. 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. Autom. 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. Autom. Lett.*, vol. 8, no. 12, pp. 7905–7912, Dec. 2023.
- [10] R. Tallamraju, E. Price, R. Ludwig, K. Karlapalem, H. H. Bülthoff, M. J. Black, and A. Ahmad, "Active perception based formation control for multiple aerial vehicles," *IEEE Robot. Autom. 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, Jun. 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] "Using robots to clean oil spills," *Robots Tomorrow*. [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, H. Zhi, J. G. Romero, L. Labazanova, A. Duan, X. Li, and D. Navarro-Alarcon, "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 Contr. Syst. Lett.*, vol. 7, pp. 3391–3396, Nov. 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*, 1<sup>st</sup> ed. London, UK: 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. Rob. 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 University 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, Jul. 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, Aug. 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. Rob. Syst.*, vol. 94, pp. 761–775, Jun. 2019.
- [28] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.