

# Guiding Vector Fields for the Distributed Motion Coordination of Mobile Robots

Weijia Yao , Member, IEEE, Héctor García de Marina , Zhiyong Sun , and Ming Cao , Fellow, IEEE

**Abstract**—In this article, we propose coordinating guiding vector fields to achieve two tasks simultaneously with a team of robots: first, the guidance and navigation of multiple robots to possibly different paths or surfaces typically embedded in 2-D or 3-D, and second, their motion coordination while tracking their prescribed paths or surfaces. The motion coordination is defined by desired parametric displacements between robots on the path or surface. Such a desired displacement is achieved by controlling the virtual coordinates, which correspond to the path or surface’s parameters, between guiding vector fields. Rigorous mathematical guarantees underpinned by dynamical systems theory and Lyapunov theory are provided for the effective distributed motion coordination and navigation of robots on paths or surfaces from all initial positions. As an example for practical robotic applications, we derive a control algorithm from the proposed coordinating guiding vector fields for a Dubins-car-like model with actuation saturation. Our proposed algorithm is distributed and scalable to an arbitrary number of robots. Furthermore, extensive illustrative simulations and fixed-wing aircraft outdoor experiments validate the effectiveness and robustness of our algorithm.

**Index Terms**—Motion coordination, multirobot systems, navigation, path following.

## I. INTRODUCTION

THE scalable coordination of mobile robots is a key to achieve enhanced synergistic performance in a multirobot

Manuscript received 20 May 2022; revised 2 October 2022; accepted 18 October 2022. Date of publication 2 December 2022; date of current version 5 April 2023. An earlier version of this paper was presented in part at the 2021 IEEE International Conference on Robotics and Automation [DOI: 10.1109/ICRA48506.2021.9560845]. The work of W. Yao and M. Cao was supported in part by the European Research Council under Grant ERC-CoG-771687. The work of W. Yao was also supported in part by the National Natural Science Foundation of China. The work of H. G. de Marina was supported by the Spanish Ministry of Science under Grant Ramón y Cajal RYC2020-030090-I. The work of Z. Sun was supported by a starting grant from Eindhoven Artificial Intelligence Systems Institute, The Netherlands. This paper was recommended for publication by Associate Editor N. Gans and Editor P. Robuffo Giordano upon evaluation of the reviewers’ comments. (*Corresponding author: Weijia Yao.*)

Weijia Yao is with the School of Robotics, Hunan University, Hunan, China, and also with the ENTEG, University of Groningen, Groningen, The Netherlands (e-mail: weijia.yao.new@outlook.com).

Héctor García de Marina is with the Department of Computer Engineering, Automation and Robotics, and with CITIC, Universidad de Granada, 18071 Granada, Spain (e-mail: hgdemarina@gmail.com).

Zhiyong Sun is with the Eindhoven University of Technology, 5612 AZ Eindhoven, The Netherlands (e-mail: z.sun@tue.nl).

Ming Cao is with the ENTEG, University of Groningen, Groningen, The Netherlands (e-mail: ming.cao@ieee.org).

This article has supplementary material provided by the authors and color versions of one or more figures available at <https://doi.org/10.1109/TRO.2022.3224257>.

Digital Object Identifier 10.1109/TRO.2022.3224257

system. For example, the proper coordination of multiple mobile robots minimizes interference and reduces duplicate work in tasks, such as multirobot path planning and coverage, and can boost the performance of cooperative and collaborative multirobot missions [1]. However, to reliably and systematically coordinate robots in a large multirobot system without a central nexus is one of the grand challenges in robotics [2]. A fundamental question of this challenge is how to design a mechanism such that multiple robots are able to accurately follow possibly different desired paths and coordinate their motions in a distributed fashion, with the resulting formation satisfying some geometric or parametric constraints. This article proposes a coordinating guiding vector field to guide an arbitrary number of robots to achieve motion coordination on possibly different desired paths or surfaces in a distributed way through local information exchange. The proposed guiding vector field has many appealing features, such as ensuring rigorous guarantees of motion coordination and path/surface convergence.

## A. Related Work

Single-robot’s path-following capability is fundamental in mobile robotics applications, and many algorithms have been proposed and widely studied [3]. It is concluded by numerical simulations in [3] that those algorithms based on guiding vector fields achieve relatively high path-following accuracy while they require less control effort compared with several other tested algorithms, such as LQR-based ones [4] and nonlinear guidance laws [5]. This conclusion is partly experimentally supported by Caharija et al. [6]. Despite the performance advantages, these vector field based algorithms are mostly designed for a single robot, while the extension for a multirobot system is relatively less studied. In fact, to the best of our knowledge, there are only a few studies exploiting a guiding vector field for the multirobot distributed motion coordination. The guiding vector field proposed in [7] is exploited in [8] in combination with a distributed algorithm to dynamically change the radii of circular paths such that multiple fixed-wing aircraft flying at a constant speed can eventually follow the same circular path and keep predefined inter-robot distances. A different guiding vector field is derived in [9] based on various potential functions for multiple robots to move on common paths, such as a circle or a straight line. Another work [10] presents a distributed control law for a number of robots to circulate along a closed curve described in a specific form in 3-D. Recently, Wang et al. [11] employ

a guiding vector field to achieve fixed-wing UAV formation control.

Without employing a guiding vector field, various algorithms are proposed in the literature for multirobot coordinated path following. A virtual structure is utilized in [12] for multiple robots to coordinate their motions while they follow predefined paths. As each robot needs to broadcast its states and reference trajectories to other robots, the communication overhead increases as the number of robots grows. A theoretical study in [13] considers planar simple closed curves and unit-speed particles. In [14], the desired motion coordination among robots are generated by a linear exosystem, and an output-regulation controller is implemented on some robots while others are controlled through local interactions. Many other works study multirobot distributed following of planar simple desired paths (e.g., a straight line), such as [15], [16], [17], [18]. Mong-ying and Kumar et al. [19], [20] design distributed controllers for robots characterized by single-integrator or double-integrator models such that they can generate the 2-D desired geometric patterns composed of simple closed curves or circulate along a 2-D closed curve.

Circular formation control and circumnavigation control, which have been extensively studied, can be regarded as specialized cases of multirobot coordinated path-following control. To be more specific, in the circular formation control problem, robots are required to distribute along a circle and maintain some desired arc distances using local interactions [21], [22], [23]. Motivated by some specific applications, such as entrapment of a hostile target, an additional requirement for all robots to move persistently along the circle is imposed in a circumnavigation control problem [24], [25], [26]. Nevertheless, as clear from their problem formulations, these studies only take into account circular paths mostly in a 2-D plane. Although it is possible to extend these studies to consider other closed paths, this might require some continuous transformation between paths, which is nontrivial in general.

### B. Contributions

There are two major contributions in this article. The first focuses on extending the guiding vector field for tracking desired paths (i.e., one-dimensional manifolds) presented in [27] to parametric surfaces. Such an extension allows a richer repertoire of coordinated collective motions. The second contribution focuses on the design of a coordinating guiding vector field for a multirobot system to achieve global convergence to desired paths and surfaces while their motions are coordinated in a distributed way via local communication.

As a practical illustration, we show how from the proposed coordinating guiding vector field, one can design a control algorithm for guiding robots using a nonholonomic Dubins-car-like model with saturated actuators. Specifically, we consider desired paths and surfaces defined as parametric equations such that the motion coordination among robots can be quantified by relative parametric differences. The parameters describing the paths or surfaces are treated as virtual coordinates to construct the singularity-free guiding vector fields. Such virtual

coordinates can be coupled between robots via a consensus term in an undirected communication graph. Consequently, these new guiding vector fields effectively control the *relative parametric separation* among robots.

We would like to highlight a nonexhaustive list of four appealing aspects of our approach.

- 1) There are rigorous mathematical guarantees on the global convergence of robots' trajectories converging to the desired paths or surfaces and achieving the motion coordination.
- 2) Our approach can deal with complex paths, such as self-intersecting, nonclosed, and nonconvex ones in an  $n$ -dimensional configuration space, and it is also applicable for arbitrary surfaces described by parametric equations. The approach can also be naturally extended for higher dimensional manifolds (e.g., a cube).
- 3) Our approach is distributed and scalable. Given a fixed communication frequency, the communication cost is low since every two neighboring robots only need to receive and transmit the virtual coordinates, of which the number is equal to that of the path or surface parameters. The approach is also computationally cheap and is suitable for real-time applications since online optimization is not required.
- 4) We also demonstrate that our algorithm is promising in applications such as area or volume coverage tasks by exploiting 2-D or 3-D Lissajous curves with *irrational coefficients*.

This article is a significant extension of our preliminary version in [28]. Hereby, we list the improvements and original contributions with respect to our conference work [28] as follows.

- 1) The conference version only considered the case of multiple robots following different desired paths (i.e., one-dimensional manifolds), while, in this article, we generalize the result for motion coordination on surfaces and higher dimensional manifolds (see Section IV).
- 2) The conference version only briefly mentioned the use of safety barrier certificate with the proposed coordinating guiding vector field for collision avoidance, but we present more technical details in this article (see Section V).
- 3) We add new simulations and experiments to verify the aforementioned new theoretical results (see Section VII).
- 4) We provide in this article all the technical proofs that were omitted in [28] due to the conference page limit. In addition, in-depth interpretation of the theoretical results, complemented with additional results (e.g., Corollary 1), is provided in this article.

The remainder of this article is organized as follows. Section II introduces the preliminaries on graph theory and guiding vector fields for path following. Then, Section III elaborates on how to extend the guiding vector field for multirobot distributed motion coordination and navigation on desired paths. The theoretical results are further extended to deal with coordinated maneuvering on parametric surfaces in Section IV. We also discuss how the coordinating guiding vector field can be seamlessly integrated with a safety barrier certificate to address the

collision issue in Section V. Next, we consider a realistic robot model for fixed-wing aircraft and derive a control law from the coordinating guiding vector field in Section VI. Moreover, extensive simulation examples and experiments with fixed-wing aircraft are carried out in Section VII. Finally, Section VIII concludes this article and indicates future work.

## II. PRELIMINARIES

### A. Notations

The set of integers  $\{m \in \mathbb{Z} : i \leq m \leq j\}$  is denoted by  $\mathbb{Z}_i^j$ . We use boldface for a vector  $\mathbf{v} \in \mathbb{R}^n$ , and its  $j$ th entry is denoted by  $v_j$  for  $j \in \mathbb{Z}_1^n$ . Consider a system consisting of  $N$  robots. Any quantity associated with the  $i$ th robot is symbolized by the superscript  $(\cdot)^{[i]}$ . For example, the notation  $\mathbf{u}^{[i]} \in \mathbb{R}^n$  denotes a vector associated with the  $i$ th robot for  $i \in \mathbb{Z}_1^N$ , and the  $j$ th entry of it is denoted by  $u_j^{[i]}$  for  $j \in \mathbb{Z}_1^n$  (note that  $u_j^{[i]}$  is not boldfaced since it is a scalar). The notation “ $\mathbf{:=}$ ” means “defined to be,” and  $\text{diag}\{\cdot\}$  denotes a (block) diagonal matrix obtained by putting vectors or matrices on the diagonal. If  $\mathcal{A}$  is a finite set, then  $|\mathcal{A}|$  denotes its cardinality (i.e., the number of elements in the set).

### B. Graph Theory

This section follows the convention in [29]. A finite *undirected graph* (or *graph* for short) is a set-theoretic object  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ , where the *vertex set*  $\mathcal{V} := \{1, \dots, N\}$  contains a finite set of elements, called *vertices*, and the *edge set*  $\mathcal{E}$  is a subset of  $\mathcal{V} \times \mathcal{V}$  of which the elements are denoted by  $(i, j)$ , representing the *adjacent* relation between vertices  $i$  and  $j$  for  $i, j \in \mathcal{V}$ . The  $k$ th edge is denoted by  $\mathcal{E}_k = (\mathcal{E}_k^{\text{head}}, \mathcal{E}_k^{\text{tail}})$ , where  $\mathcal{E}_k^{\text{head}}, \mathcal{E}_k^{\text{tail}} \in \mathcal{V}$  are the *head* and *tail* of the edge, respectively. The set of neighboring robots of robot  $i$  is denoted by  $\mathcal{N}_i := \{j \in \mathcal{V} : (i, j) \in \mathcal{E}\}$ . The graph  $\mathcal{G}$  is *connected* if there is a path between any pair of vertices in  $\mathcal{V}$ . The *adjacency matrix*  $A(\mathcal{G})$  of the undirected graph  $\mathcal{G}$  is the symmetric  $N \times N$  matrix encoding the adjacency relationships of vertices, that is,  $[A(\mathcal{G})]_{ij} = 1$  if  $(i, j) \in \mathcal{E}$  and  $[A(\mathcal{G})]_{ij} = 0$  otherwise. The *Laplacian matrix*  $L(\mathcal{G})$  of  $\mathcal{G}$  is the  $N \times N$  matrix defined by  $[L(\mathcal{G})]_{ij} = -a_{ij}$  for  $i \neq j$  and  $[L(\mathcal{G})]_{ii} = \sum_{k=1}^N a_{ik}$  for  $i \in \mathbb{Z}_1^N$ , where  $a_{ij}$  is the  $ij$ th entry of the adjacency matrix. For an undirected graph, we define the entries of the incidence matrix  $B \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{E}|}$  by  $b_{ik} = +1$  if  $i = \mathcal{E}_k^{\text{head}}$ ,  $b_{ik} = -1$  if  $i = \mathcal{E}_k^{\text{tail}}$ , and  $b_{ik} = 0$  otherwise.

The  $N$ -cycle graph  $C_N = (\{1, \dots, N\}, \mathcal{E}_C)$  is the graph where the edge set is  $\mathcal{E}_C = \{(1, 2), (2, 3), \dots, (N-2, N-1), (N-1, N), (N, 1)\}$ . Namely, the  $N$  robots are connected in sequence to form a cycle.

### C. Guiding Vector Field With a Virtual Coordinate for a Single Robot

Guiding vector fields with virtual coordinates for robot navigation enjoy many features, such as guaranteeing global convergence to the desired path and enabling self-intersecting desired path following [27], [30], [31]. We present here a brief introduction. Suppose the (physical) desired path  ${}^{\text{phy}}\mathcal{P}$  is parameterized

by

$$x_1 = f_1(w), \dots, x_n = f_n(w)$$

where  $x_j$  is the  $j$ th coordinate,  $w \in \mathbb{R}$  is the parameter of the path, and  $f_j$  is twice continuously differentiable (i.e.,  $f_j \in C^2$ ) for  $j \in \mathbb{Z}_1^n$ . To derive a corresponding guiding vector field for the desired path, we need to describe the desired path as the intersection of  $n$  hypersurfaces [32], [33]. To this end, taking  $w$  as an additional argument, we define  $n$  surface functions  $\phi_j : \mathbb{R}^{n+1} \rightarrow \mathbb{R}$  as follows:  $\phi_1(\xi) = x_1 - f_1(w), \dots, \phi_n(\xi) = x_n - f_n(w)$ , where  $\xi = (x_1, \dots, x_n, w) \in \mathbb{R}^{n+1}$  is the generalized coordinate with an additional *virtual coordinate*  $w$ . Therefore, the desired path with an additional coordinate is the intersection of the hypersurfaces described by the zero-level set of these functions. Namely

$${}^{\text{hgh}}\mathcal{P} := \{\xi \in \mathbb{R}^{n+1} : \phi_j(\xi) = 0, j \in \mathbb{Z}_1^n\}.$$

The projection of the higher dimensional  ${}^{\text{hgh}}\mathcal{P}$  onto the hyperplane spanned by the first  $n$  coordinates is the original (physical) desired path  ${}^{\text{phy}}\mathcal{P}$ , so we can use the higher dimensional guiding vector field  $\chi : \mathbb{R}^{n+1} \rightarrow \mathbb{R}^{n+1}$  corresponding to  ${}^{\text{hgh}}\mathcal{P}$  to follow the original (physical) desired path  ${}^{\text{phy}}\mathcal{P}$  by using a projection technique. The vector field  $\chi$  is defined by

$$\chi(\xi) = \times(\nabla\phi_1, \dots, \nabla\phi_n) - \sum_{j=1}^n k_j \phi_j \nabla\phi_j \quad (1)$$

where  $k_j$  is a positive gain,  $\nabla\phi_j \in \mathbb{R}^{n+1}$  is the gradient of  $\phi_j$  with respect to its generalized coordinate  $\xi$ , and the first term is the wedge product [27, eq. (5)] of all the gradients  $\nabla\phi_j \in \mathbb{R}^{n+1}, j \in \mathbb{Z}_1^n$  (it degenerates to the cross product if  $n = 2$ ). The physical interpretation of the vector field  $\chi$  is clear: the second term  $-\sum_{j=1}^n k_j \phi_j \nabla\phi_j$  is a weighted sum of all the gradients, which guides the trajectory toward the intersection of the hypersurfaces (i.e., the desired path), while the first term  $\times(\nabla\phi_1, \dots, \nabla\phi_n)$ , being orthogonal to all the gradients  $\nabla\phi_j$  [34, Proposition 7.2.1], provides a propagation direction along the desired path.

*Remark 1:* The virtual coordinate  $w$  can be seen as a “carrot” tracked by robots. Notwithstanding, the dynamics of the “carrot” are in closed loop with the positions of the robots, and thus, it may go backwards on the desired path. The detail is illustrated in [27, Section VII] and its supplementary video. Similarly, another approach to generate vector fields from the parametric equations of desired paths has recently been proposed in [35], which is able to deal with time-varying desired paths and enables the isotropic-in-space convergence. However, the work presented in [35] only focused on the single-robot case. We will show in the sequel how the guiding vector field introduced in this section can be extended for multirobot motion coordination on desired paths and surfaces with collision avoidance. In addition, our approach can deal with self-intersecting desired paths.

## III. DISTRIBUTED MOTION COORDINATION ON DESIRED PATHS USING GUIDING VECTOR FIELDS

In our previous work [27], [30], the higher dimensional guiding vector field in (1) is proved to possess no singular points

where the vector field becomes zero, thanks to the additional dimension (i.e., the additional virtual coordinate  $w$ ). However, like many of other current studies (e.g., [33], [36]), the vector field is used for the guidance of only *one single* robot. We aim to extend the vector field to include a *coordination component* and achieve motion coordination among multiple robots. Since the additional virtual coordinate  $w$  not only helps eliminate singular points but also acts as the path parameter, one idea is to utilize the virtual coordinate  $w$  to coordinate the robots' motions. Thus, the question is: Suppose there are  $N > 1$  robots, and the communication topology is described by an undirected graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ , where  $\mathcal{V} = \{1, \dots, N\}$  represent these  $N$  robots, if  $(i, j) \in \mathcal{E}$ , then information can flow between Robot  $i$  and Robot  $j$ . Based on the higher dimensional vector field in (1), how to design an extra *coordination mechanism* using the virtual coordinates  $w^{[i]}$ ,  $i \in \mathbb{Z}_1^N$ , such that

- 1) each robot can follow their desired paths;
- 2) all robots can coordinate their motions by controlling the virtual coordinates  $w^{[i]}$ ,  $i \in \mathbb{Z}_1^N$ , via *local* information exchange among neighboring robots?

The extra coordination mechanism and the precise meaning of motion coordination will become clear after we present the problem formulation using the dynamical systems theory.

#### A. Mathematical Problem Formulation

Suppose the  $i$ th robot is required to follow a path in  $\mathbb{R}^n$ , parameterized by  $n$  parametric equations

$$x_1^{[i]} = f_1^{[i]}(w^{[i]}) \quad \dots \quad x_n^{[i]} = f_n^{[i]}(w^{[i]}) \quad (2)$$

where  $x_j^{[i]}$  is the  $j$ th coordinate,  $f_j^{[i]} \in C^2$  is the  $j$ th parametric function for the  $i$ th robot,  $i \in \mathbb{Z}_1^N$ ,  $j \in \mathbb{Z}_1^n$ , and  $w^{[i]}$  is the parameter of the desired path. To derive the path-following guiding vector field, we use the parameter  $w^{[i]}$  as an additional virtual coordinate, and the higher dimensional desired path is described by

$$\mathcal{P}^{[i]} := \left\{ \xi^{[i]} \in \mathbb{R}^{n+1} : \phi_1^{[i]}(\xi^{[i]}) = 0, \dots, \phi_n^{[i]}(\xi^{[i]}) = 0 \right\}$$

where  $\xi^{[i]} := (x_1^{[i]}, \dots, x_n^{[i]}, w^{[i]}) \in \mathbb{R}^{n+1}$  denotes the *generalized coordinate* of the  $i$ th robot. Note that the  $(n+1)$ th entry of  $\xi^{[i]}$  is the additional virtual coordinate  $w^{[i]}$ . As before, the surface functions are defined by

$$\phi_j^{[i]}(x_1^{[i]}, \dots, x_n^{[i]}, w^{[i]}) = x_j^{[i]} - f_j^{[i]}(w^{[i]})$$

for  $i \in \mathbb{Z}_1^N$  and  $j \in \mathbb{Z}_1^n$ . Let

$$\Phi^{[i]}(\xi^{[i]}) := (\phi_1^{[i]}(\xi^{[i]}), \dots, \phi_n^{[i]}(\xi^{[i]}))^\top \in \mathbb{R}^n.$$

Observe that  $\xi^{[i]} \in \mathcal{P}^{[i]}$  if and only if  $\|\Phi^{[i]}(\xi^{[i]})\| = 0$ . Therefore, we can use  $\Phi^{[i]}(\xi^{[i]})$  to quantify the distance to the desired path  $\mathcal{P}^{[i]}$ . In the context of path following, we call  $\Phi^{[i]}(\xi^{[i]})$  the *path-following error* to  $\mathcal{P}^{[i]}$ . The aim is to design controllers such that the norm  $\|\Phi^{[i]}(\xi^{[i]})\|$  converges to zero eventually. By combining (1) and (2), we obtain the analytic expression of the *path-following guiding vector field*  $\text{pf}\chi^{[i]} : \mathbb{R}^{n+1} \rightarrow \mathbb{R}^{n+1}$  for

the  $i$ th robot

$$\text{pf}\chi^{[i]}(\xi^{[i]}) = \begin{bmatrix} l(-1)^n f_1^{[i]}'(w^{[i]}) - k_1^{[i]} \phi_1^{[i]}(\xi^{[i]}) \\ \vdots \\ (-1)^n f_n^{[i]}'(w^{[i]}) - k_n^{[i]} \phi_n^{[i]}(\xi^{[i]}) \\ (-1)^n + \sum_{l=1}^n k_l^{[i]} \phi_l^{[i]}(\xi^{[i]}) f_l^{[i]}'(w^{[i]}) \end{bmatrix} \quad (3)$$

for  $i \in \mathbb{Z}_1^N$ , where  $k_j^{[i]} > 0$  are constant gains, and  $f_j^{[i]}'$  are the derivatives of  $f_j^{[i]}$  with respect to the argument  $w^{[i]} \in \mathbb{R}$ .

To achieve coordination in  $w^{[i]}$ , thus indirectly coordinate the positions of robots, we introduce a new concept: a *coordination component*. The coordination component  $\text{cr}\chi^{[i]} : \mathbb{R} \times \mathbb{R}^N \rightarrow \mathbb{R}^{n+1}$  for the  $i$ th robot,  $i \in \mathbb{Z}_1^N$ , is

$$\text{cr}\chi^{[i]}(t, \mathbf{w}) = (0, \dots, 0, c^{[i]}(t, \mathbf{w}))^\top \in \mathbb{R}^{n+1} \quad (4)$$

where  $\mathbf{w} = (w^{[1]}, \dots, w^{[N]})^\top$ , and  $c^{[i]} : \mathbb{R} \times \mathbb{R}^N \rightarrow \mathbb{R}$  is called the *coordination function* to be designed later,<sup>1</sup> which enables coordination among robots through the local interactions via the neighboring virtual coordinates  $w^{[j]}$  for  $j \in \mathcal{N}_i$ . Specifically, we want the virtual coordinates of multiple robots  $w^{[i]}(t) - w^{[j]}(t)$  to converge to  $\Delta^{[i,j]}(t)$  for  $(i, j) \in \mathcal{E}$ , where  $\Delta^{[i,j]}(t) \in \mathbb{R}$  are real-valued continuously differentiable functions, representing the desired differences between  $w^{[i]}(t)$  and  $w^{[j]}(t)$  at time  $t$ , satisfying  $\Delta^{[i,j]}(t) = -\Delta^{[j,i]}(t)$ . It is naturally assumed that  $\Delta^{[i,j]}(t)$  are chosen appropriately such that the resulting formation is feasible<sup>2</sup> at any time  $t \geq 0$ .

We design the  $i$ th *coordinating guiding vector field*  $\mathfrak{X}^{[i]} : \mathbb{R} \times \mathbb{R}^{n+1+N} \rightarrow \mathbb{R}^{n+1}$  to be the weighted sum of the path-following vector field  $\text{pf}\chi^{[i]}$  and the coordination component  $\text{cr}\chi^{[i]}$  as follows:

$$\mathfrak{X}^{[i]}(t, \xi^{[i]}, \mathbf{w}) = \text{pf}\chi^{[i]}(\xi^{[i]}) + k_c \text{cr}\chi^{[i]}(t, \mathbf{w}) \quad (5)$$

where  $k_c > 0$  is a parameter to adjust the contribution of  $\text{pf}\chi^{[i]}$  and  $\text{cr}\chi^{[i]}$  to  $\mathfrak{X}^{[i]}$ . With a larger value of  $k_c$ , the motion coordination is achieved faster. The coordinating guiding vector field  $\mathfrak{X}^{[i]}$  represents the desired moving direction for Robot  $i$ , guiding the robot's motion. Thus, it is imperative to study the guidance result, or precisely, the convergence results of the integral curves of the vector field  $\mathfrak{X}^{[i]}$  for  $i \in \mathbb{Z}_1^N$ . Precisely, we stack all the robot states as a vector  $\xi := (\xi^{[1]}, \dots, \xi^{[N]})^\top \in \mathbb{R}^{(n+1)N}$ , stack all the coordinating guiding vector fields as  $\mathfrak{X}(t, \xi) := (\mathfrak{X}^{[1]}, \dots, \mathfrak{X}^{[N]})^\top \in \mathbb{R}^{(n+1)N}$ , and study the integral curves of  $\mathfrak{X}(t, \xi)$ , that is, the solutions (or trajectories) to the

<sup>1</sup>Although the argument of the coordination function  $c^{[i]}(\cdot, \cdot)$  contains all the virtual coordinates  $w^{[j]}$ ,  $j \in \mathbb{Z}_1^N$ , this is only for a compact and readable notation. The coordination function for each individual robot only requires the knowledge of the neighbors'  $w^{[j]}$  for  $j \in \mathcal{N}_i$ , as shown later.

<sup>2</sup>If the communication topology does not contain any cycles, then arbitrary values of the desired parametric differences  $\Delta^{[i,j]}$  are feasible as long as  $\Delta^{[i,j]} = -\Delta^{[j,i]}$  is satisfied for  $i, j \in \mathbb{Z}_1^N$ ; otherwise, one needs to examine the (typically geometric) feasibility of the resulting formation.

differential equation

$$\dot{\xi} = \mathfrak{X}(t, \xi) \quad (6)$$

given an initial condition  $\xi_0 \in \mathbb{R}^{(n+1)N}$  at  $t = t_0 \geq 0$ . Note that if the coordination function  $c^{[i]}(t, \mathbf{w})$  in (4) is time-invariant, it is not explicitly dependent on time  $t$ , and we can simply rewrite it to  $c^{[i]}(\mathbf{w})$ . In this case, the system (6) can be rewritten as  $\dot{\xi} = \mathfrak{X}(\xi)$ , which is an autonomous system; otherwise, it is a nonautonomous system [37, Ch. 1], which is more difficult to analyze and left for future work. Throughout this article, we only consider a time-invariant coordination function  $c^{[i]}(\mathbf{w})$  and the autonomous system (6). We can formally formulate the problem as follows.

*Problem 1 (Multirobot path following):* Design the coordinating guiding vector field  $\mathfrak{X}^{[i]}$  in (5) for  $i \in \mathbb{Z}_1^N$  such that the trajectories of (6), given an initial condition  $\xi_0 \in \mathbb{R}^{(n+1)N}$  at  $t = t_0 \geq 0$ , fulfill the following two control objectives.

- 1) (*Path Following*): Robot  $i$ 's path-following error to its desired path  $\mathcal{P}^{[i]}$  converges to zero asymptotically for  $i \in \mathbb{Z}_1^N$ , that is,  $\|\Phi^{[i]}(\xi^{[i]}(t))\| \rightarrow 0$  as  $t \rightarrow \infty$  for  $i \in \mathbb{Z}_1^N$ .
- 2) (*Motion Coordination*): Each robot's motion is coordinated distributedly subject to the communication graph  $\mathcal{G}$  (i.e., Robot  $i$  can communicate with Robot  $j$  if and only if  $(i, j) \in \mathcal{E}$ ) such that their virtual coordinates satisfy  $w^{[i]}(t) - w^{[j]}(t) - \Delta^{[i,j]} \rightarrow 0$  as  $t \rightarrow \infty$  for  $(i, j) \in \mathcal{E}$ .

Given the path-following vector field in (3), we will design the coordination function  $c^{[i]}(\cdot)$  later such that it coordinates the robots' motion but does not affect the path-following performance. We propose the following mild standing assumption.

*Assumption 1:* The communication graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  is undirected and connected.

Assumption 1 implies that if  $(i, j) \in \mathcal{E}$ , then Robot  $i$  and Robot  $j$  can share information bidirectionally, and no robot is isolated from the multirobot system (e.g., a cycle graph satisfies this assumption).

## B. Time-Invariant Coordination Component and Convergence Analysis

Observe that the coordination of the virtual coordinates directly affects the coordination of the positions of the robots implicitly since the virtual coordinate corresponds to the parameter of a desired path. Motivated by this observation, we will design a time-invariant coordination component  $c^{[i]}(\mathbf{w})$  in (4), and analyze the trajectories of (6).

1) *Time-Invariant Coordination Component:* Given the desired path  $\mathcal{P}^{[i]}$ , we can design the *desired parametric differences*  $\Delta^{[i,j]}$  starting from a particular *reference configuration*<sup>3</sup>  $\mathbf{w}^* := (w^{[1]*}, \dots, w^{[N]*})^\top$ . Namely, let  $\Delta^*$  be the stacked vector of  $\Delta^{[i,j]}$ ,  $(i, j) \in \mathcal{E}$ , then  $\Delta^* := D^\top \mathbf{w}^*$ , where  $D \in \mathbb{R}^{N \times |\mathcal{E}|}$  is an incidence matrix obtained by assigning arbitrary orientations to the edges of the undirected graph.<sup>4</sup> Now we propose to employ

<sup>3</sup>Note that  $\mathbf{w}^*$  is unknown by the robots. Instead, Robot  $i$  only knows the desired parametric differences relative to its neighbors, i.e.,  $\Delta^{[i,j]}$  for  $j \in \mathcal{N}_i$ .

<sup>4</sup>Different assignments of the edges' orientations only affect the signs of each entry in the incidence matrix  $D$ , while the desired parametric differences are still the same. In addition, the Laplacian matrix  $L = DD^\top$  introduced later remains the same independent of the orientation assignment.

the following consensus control algorithm [38, p. 25]:

$$c^{[i]} = - \sum_{j \in \mathcal{N}_i} (w^{[i]} - w^{[j]} - \Delta^{[i,j]}) \quad \forall i \in \mathbb{Z}_1^N. \quad (7)$$

Equation (7) can be rewritten in a compact form as

$$c(\mathbf{w}) = -L(\mathbf{w} - \mathbf{w}^*) = -L\tilde{\mathbf{w}} \quad (8)$$

where  $c(\mathbf{w}) = (c^{[1]}(\mathbf{w}), \dots, c^{[N]}(\mathbf{w}))^\top$ ,  $L = L(\mathcal{G}) = DD^\top$  is the Laplacian matrix, and  $\tilde{\mathbf{w}} := \mathbf{w} - \mathbf{w}^*$ . Combining (3), (4), (5), and (7), we attain the coordinating guiding vector field  $\mathfrak{X}^{[i]}$  for  $i \in \mathbb{Z}_1^N$ . The vector field  $\mathfrak{X}^{[i]}$  only takes as inputs Robot  $i$ 's own states and its neighbors' virtual coordinates  $w^{[j]}$  for  $j \in \mathcal{N}_i$ .

*Remark 2:* From (4), (5), and (7), one observes that information exchange only happens in the coordination component  $c^{[i]}(\cdot)$ . Notably, the communication cost is low: Robot  $i$  transmits only a scalar  $w^{[i]}$  to the neighboring Robot  $j \in \mathcal{N}_i$ .

2) *Convergence Analysis:* The convergence analysis of trajectories of (6) is nontrivial given that the right-hand side of (6) is *not* a gradient of any potential function since the path-following vector field in (1) contains a wedge product of all the gradients. In this section, we show that the coordinating guiding vector field (5) enables multiple robots to follow their desired paths while they are coordinated by the virtual coordinates such that  $w^{[i]}(t) - w^{[j]}(t) - \Delta^{[i,j]} \rightarrow 0$  as  $t \rightarrow \infty$  for  $(i, j) \in \mathcal{E}$  as  $t \rightarrow \infty$ .

Define the *composite error vector*  $\mathbf{e}$  to be

$$\mathbf{e} = \left( \Phi^\top, (D^\top \tilde{\mathbf{w}})^\top \right)^\top \in \mathbb{R}^{nN + |\mathcal{E}|} \quad (9)$$

where

$$\Phi := \left( \Phi^{[1]\top}, \dots, \Phi^{[N]\top} \right)^\top \in \mathbb{R}^{nN} \quad (10)$$

and recall that  $D$  is the incidence matrix, which satisfies  $L = DD^\top$ . One needs to prove that  $\mathbf{e}(t) \rightarrow \mathbf{0}$  as  $t \rightarrow \infty$  to show the effectiveness of the coordinating guiding vector field  $\mathfrak{X}^{[i]}$ . This result is formally stated after the following assumption.

*Assumption 2:* The first and second derivatives of  $f_j^{[i]}(\cdot)$  in (2) are bounded for  $i \in \mathbb{Z}_1^N, j \in \mathbb{Z}_1^n$ .

*Theorem 1 (Motion coordination on paths):* Under Assumptions 1 and 2, and given constant desired parametric differences  $\Delta^{[i,j]}$  for  $(i, j) \in \mathcal{E}$ , the coordinating guiding vector fields  $\mathfrak{X}^{[i]}$  for  $i \in \mathbb{Z}_1^N$  designed by combining (3), (4), (5), and (7) solve Problem 1 globally in the sense that the initial composite error  $\|\mathbf{e}(t=0)\|$  in (9) can be arbitrarily large.

*Proof:* See [39, App. A]. ■

## IV. DISTRIBUTED COORDINATED MANEUVERING ON SURFACES

In this section, we extend the previous results such that  $N$  robots can converge to (possibly different) two-dimensional *surfaces*, and maneuver and coordinate their motions according to some parameters of the surfaces. It is clear in the previous sections what is meant by following a (one-dimensional) desired path, but it is perhaps unclear what is meant by coordinated motion on (two-dimensional) surfaces. Therefore, we first clarify the meaning of coordinated motion on surfaces in Section IV-A, and then mathematically formulate the problem in Section IV-B, and derive the coordinating guiding vector field and conduct the convergence analysis in Section IV-C.

### A. Coordinated Motion in a Desired Set

1) *Single-Robot Trajectories in a Desired Set:* For simplicity, we first study trajectories of one robot within a *desired set* of any dimensions. Suppose the desired set is in the following parametric form:

$$x_1 = f_1(w_1, \dots, w_k), \dots, x_n = f_n(w_1, \dots, w_k) \quad (11)$$

where  $w_1, \dots, w_k \in \mathbb{R}$  are  $k$  parameters,  $n \in \mathbb{N}$  is the dimension of the ambient Euclidean space (mainly  $n \in \{2, 3\}$  for applications with mobile robots), and  $f_j, j \in \mathbb{Z}_1^n$ , are twice continuously differentiable functions. Precisely, the desired set described by (11) is  $\mathcal{M} := \{(x_1, \dots, x_n) \in \mathbb{R}^n : (11), w_j \in \mathbb{R}, j \in \mathbb{Z}_1^k\}$ . Generically, if  $k = 1$ , then  $\mathcal{M} = \mathcal{P}$  is a desired path, which is one-dimensional (or roughly speaking, of one degree-of-freedom), while if  $k = 2$ , then  $\mathcal{M}$  is a two-dimensional surface. The  $k$  parameters  $w_j, j \in \mathbb{Z}_1^k$ , can be used as coordinates to localize a point in  $\mathcal{M}$ . Therefore, one can represent the position of a robot in  $\mathcal{M}$  by a  $k$ -tuple denoted by  $(w_1, \dots, w_k)_{\mathcal{M}}$  in the parameter space. This article mainly focuses on parametric paths and surfaces (i.e.,  $k = 1$  or  $k = 2$ ) but the generalization to  $k > 2$  is obvious after we elaborate on the case of  $k = 2$  in this section.

Suppose  $k = 1$  and  $\mathcal{M}$  is a desired path denoted by  $\mathcal{P}$ . It is clear that if a robot starts from the position  $(w_1)_{\mathcal{M}}$  in  $\mathcal{P}$ , then its trajectory (subject to directions) coincides with the desired path  $\mathcal{P}$ , and the speed at which the trajectory extends along the desired path is determined by the time derivative  $\dot{w}_1 \in \mathbb{R}$  along the trajectory. Suppose  $k = 2$  and  $\mathcal{M}$  is a surface denoted by  $\mathcal{S}$ . If a robot starts from the position  $(w_1, w_2)_{\mathcal{M}}$  in  $\mathcal{S}$ , it is the ratio between the different velocities  $\dot{w}_1$  and  $\dot{w}_2$  that determines the eventual trajectory of the robot in  $\mathcal{S}$ .

*Example 1:* Suppose the desired set  $\mathcal{M}$  is the sphere parameterized by

$$\mathcal{S}_{\text{sph}} := \begin{cases} x_1 = \cos(w_1) \cos(w_2) \\ x_2 = \cos(w_1) \sin(w_2) \\ x_3 = \sin(w_1) \end{cases}. \quad (12)$$

If  $\dot{w}_1 = 0$  and  $\dot{w}_2 \neq 0$ , then the robot's trajectory is parallel to the sphere's equator starting from an arbitrary point in  $\mathcal{S}_{\text{sph}}$ .

We will show later that our proposed coordinating guiding vector field allows one to set the values of  $\dot{w}_j$  for  $j \in \mathbb{Z}_1^k$ , so we can guarantee that a robot moves as expected within  $\mathcal{M}$ .

2) *Coordinated Motion of Multiple Robots:* Consider  $N > 1$  robots. The coordinated motion among multiple robots is characterized by the predefined parametric relations  $\Delta_{(\cdot)}^{[i,j]}$ ,  $i \neq j \in \mathbb{Z}_1^N$ , between neighboring robots, for which the neighboring relations are encoded in the undirected communication graph  $\mathcal{G}$ . Note that the subscript in  $\Delta_{(\cdot)}^{[i,j]}$  specifies which parameter of the desired set is considered (in the case of one parameter, this subscript is omitted for simplicity).

In the previous section, we only consider one parameter (i.e.,  $k = 1$ )  $w_1$  to be communicated among robots. In this case, these robots exhibit the behavior of *chasing each other* on one-dimensional desired paths  $\mathcal{P}^{[i]}$ . Dealing with a desired set  $\mathcal{M}$  with  $k > 1$  gives us more freedom for the coordinated motion

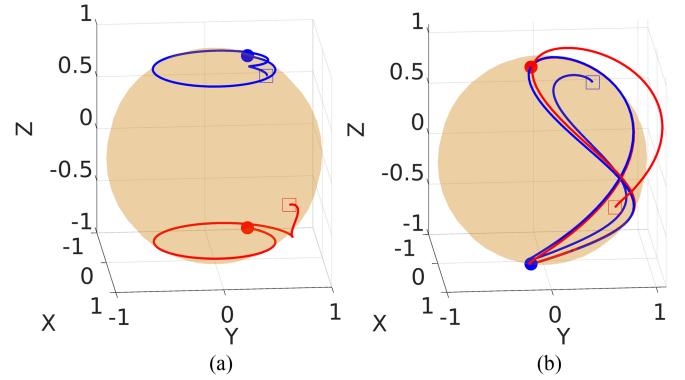


Fig. 1. Simulation results of two robots coordinating their motions on the sphere described by (12). The blue and red dots represent Robot 1 and 2, respectively, and the square symbols represent the initial positions. The desired parametric differences are  $\Delta_1^{[1,2]} = \Delta_2^{[1,2]} = -\pi$ . (a) Desired parametric speeds are  $\dot{w}_1^* = 0$  and  $\dot{w}_2^* = 1$  for both robots. (b) Desired parametric speeds are  $\dot{w}_1^* = \dot{w}_2^* = 1$  for both robots.

of robots. For simplicity, we consider the case of  $k = 2$  (i.e., the desired set  $\mathcal{M}$  is a two-dimensional surface) in the following example.

*Example 2:* Two robots coordinate their motions in the desired set: the sphere described by (12). Specifically, we require that the difference between each of the two parameters,  $w_1^{[1]}$  and  $w_2^{[1]}$ , of Robot 1, and each of those of Robot 2,  $w_1^{[2]}$  and  $w_2^{[2]}$ , is  $-\pi$ . Namely, the desired parametric differences are  $\Delta_1^{[1,2]} = \Delta_2^{[1,2]} = -\pi$ . We also require that, at the steady state, they move at specific parametric speeds: The desired speeds in the first and the second parameter are  $\dot{w}_1^* = 0$  and  $\dot{w}_2^* = 1$ , respectively, in the first scenario, and  $\dot{w}_1^* = \dot{w}_2^* = 1$  in the second scenario for both robots. We use the coordinating guiding vector field that will be introduced later to produce the simulation results shown in Fig. 1.

In the next section, we formally define the problem of motion coordination of robots on surfaces and provide the technical detail of the coordinating guiding vector field for surface navigation as employed in Example 2.

### B. Mathematical Problem Formulation

In previous sections, the  $i$ th robot is required to follow a one-dimensional desired path, which is parameterized by one parameter  $w^{[i]}$ . In contrast, to converge to a two-dimensional surface, we need to use two parameters denoted by  $w_1^{[i]}$  and  $w_2^{[i]}$ , respectively. Specifically, suppose the  $i$ th robot is required to converge to a two-dimensional surface  $\mathcal{S}^{[i]} \subseteq \mathbb{R}^n$ , which is parameterized by  $n$  parametric equations

$$x_1^{[i]} = f_1^{[i]}(w_1^{[i]}, w_2^{[i]}) \quad \dots \quad x_n^{[i]} = f_n^{[i]}(w_1^{[i]}, w_2^{[i]}) \quad (13)$$

where  $x_j^{[i]}$  is the  $j$ th coordinate,  $f_j^{[i]} \in C^2$  is the  $j$ th parametric function for the  $i$ th robot, for  $i \in \mathbb{Z}_1^N, j \in \mathbb{Z}_1^n$ , and  $w_1^{[i]}, w_2^{[i]}$  are two parameters of the surface. To derive the corresponding guiding vector field, we use the parameters  $w_1^{[i]}$  and  $w_2^{[i]}$  as two additional virtual coordinates, and the surface  $\mathcal{S}^{[i]}$  is described

by

$$\mathcal{S}^{[i]} := \left\{ \xi^{[i]} \in \mathbb{R}^{n+2} : \phi_1^{[i]}(\xi^{[i]}) = 0, \dots, \phi_n^{[i]}(\xi^{[i]}) = 0 \right\}$$

where  $\xi^{[i]} := (x_1^{[i]}, \dots, x_n^{[i]}, w_1^{[i]}, w_2^{[i]}) \in \mathbb{R}^{n+2}$  denotes the *generalized coordinate* of the  $i$ th robot. Note that the  $(n+1)$ th and the  $(n+2)$ th entries of  $\xi^{[i]}$  are the additional virtual coordinates  $w_1^{[i]}$  and  $w_2^{[i]}$ , respectively. The functions  $\phi_j^{[i]}$  are

$$\phi_j^{[i]}(x_1^{[i]}, \dots, x_n^{[i]}, w_1^{[i]}, w_2^{[i]}) = x_j^{[i]} - f_j^{[i]}(w_1^{[i]}, w_2^{[i]})$$

for  $i \in \mathbb{Z}_1^N$  and  $j \in \mathbb{Z}_1^n$ . We define  $\Phi^{[i]}(\xi^{[i]}) := (\phi_1^{[i]}(\xi^{[i]}), \dots, \phi_n^{[i]}(\xi^{[i]}))^T \in \mathbb{R}^n$ . Observe that  $\xi^{[i]} \in \mathcal{S}^{[i]}$  if and only if  $\|\Phi^{[i]}(\xi^{[i]})\| = 0$ . Therefore, we can similarly use  $\Phi^{[i]}(\xi^{[i]})$ , called the *surface-convergence error*, to quantify the distance to the desired surface  $\mathcal{S}^{[i]}$ .

We use the same structure as the guiding vector field discussed before. However, since the dimensions of the states become  $n+2$ , the original wedge product is not well-defined. Therefore, we need to introduce an *extra vector*  $v = (v_1, \dots, v_{n+2})^T \in \mathbb{R}^{n+2}$  into the wedge product. Specifically, the new guiding vector field  $\text{sf}\chi^{[i]} : \mathbb{R}^{n+2} \rightarrow \mathbb{R}^{n+2}$  corresponding to the surface  $\mathcal{S}^{[i]}$ , called the *surface-navigation vector field*, is

$$\begin{aligned} \text{sf}\chi^{[i]}(\xi^{[i]}) &= \wedge \left( \nabla \phi_1^{[i]}(\xi^{[i]}), \dots, \nabla \phi_n^{[i]}(\xi^{[i]}), v \right) \\ &\quad - \sum_{j=1}^n k_j^{[i]} \phi_j^{[i]}(\xi^{[i]}) \nabla \phi_j^{[i]}(\xi^{[i]}). \end{aligned} \quad (14)$$

In theory, the extra vector  $v \in \mathbb{R}^{n+2}$  can be arbitrarily chosen since the first term  $\wedge(\nabla \phi_1^{[i]}(\xi^{[i]}), \dots, \nabla \phi_n^{[i]}(\xi^{[i]}), v)$  of (14) is always orthogonal to each of the scaled gradient  $k_j^{[i]} \phi_j^{[i]}(\xi^{[i]}) \nabla \phi_j^{[i]}(\xi^{[i]})$  in the second term of (14), and this orthogonality is crucial for showing the convergence of robots' trajectories to the desired surface. However, for simplicity, we choose  $v_1 = \dots = v_n \equiv 0$ , while  $v_{n+1}, v_{n+2}$  can be zero or nonzero constant values. This choice results in the following simple explicit expression of (14):

$$\begin{aligned} \text{sf}\chi^{[i]}(\xi^{[i]}) &= \\ (-1)^n \begin{bmatrix} v_{n+2} \partial_{w_1} f_1^{[i]} - v_{n+1} \partial_{w_2} f_1^{[i]} \\ \vdots \\ v_{n+2} \partial_{w_1} f_n^{[i]} - v_{n+1} \partial_{w_2} f_n^{[i]} \\ v_{n+2} \\ -v_{n+1} \end{bmatrix} &+ \begin{bmatrix} -k_1^{[i]} \phi_1^{[i]} \\ \vdots \\ -k_n^{[i]} \phi_n^{[i]} \\ \sum_{j=1}^n k_j^{[i]} \phi_j^{[i]} \partial_{w_1} f_j^{[i]} \\ \sum_{j=1}^n k_j^{[i]} \phi_j^{[i]} \partial_{w_2} f_j^{[i]} \end{bmatrix} \end{aligned} \quad (15)$$

where  $\partial_{w_1} f_j^{[i]} := \frac{\partial f_j^{[i]}}{\partial w_1^{[i]}}$  and  $\partial_{w_2} f_j^{[i]} := \frac{\partial f_j^{[i]}}{\partial w_2^{[i]}}$  for  $i \in \mathbb{Z}_1^N$  and  $j \in \mathbb{Z}_1^n$ . Thanks to this choice of the extra vector  $v$ , it affects only the last two entries of the first term of (15). Once the robots are on the desired surface, the second term of (15) becomes zero. Therefore, the last two entries of the first term in (15) equals the *parametric speeds*  $\dot{w}_1^{[i]}$  and  $\dot{w}_2^{[i]}$ , which represent how fast and

in which direction the robots navigate on the surface in terms of the parameters  $w_1^{[i]}$  and  $w_2^{[i]}$ .

To achieve coordination in  $w_1^{[i]}$  and  $w_2^{[i]}$ , thus indirectly coordinate the positions of robots, we similarly introduce two time-invariant *coordination components*  $\text{cr}\chi_1^{[i]}, \text{cr}\chi_2^{[i]} : \mathbb{R}^N \rightarrow \mathbb{R}^{n+2}$  for the  $i$ th robot

$$\text{cr}\chi_1^{[i]}(w_1^{[i]}) = (0, \dots, 0, c_1^{[i]}(w_1^{[i]}), 0)^\top \quad (16a)$$

$$\text{cr}\chi_2^{[i]}(w_2^{[i]}) = (0, \dots, 0, 0, c_2^{[i]}(w_2^{[i]}))^\top \quad (16b)$$

where  $w_1^{[i]} = (w_1^{[1]}, \dots, w_1^{[N]})^\top, w_2^{[i]} = (w_2^{[1]}, \dots, w_2^{[N]})^\top$ , and  $c_1^{[i]}, c_2^{[i]} : \mathbb{R}^N \rightarrow \mathbb{R}$  are the time-invariant *coordination functions* to be designed later,<sup>5</sup> which enables coordination among robots through the local interactions via the *neighboring virtual coordinates*  $w_1^{[j]}$  and  $w_2^{[j]}$  for  $j \in \mathcal{N}_i$ . Specifically, we want the virtual coordinates of multiple robots  $w_1^{[i]}(t) - w_1^{[j]}(t)$  and  $w_2^{[i]}(t) - w_2^{[j]}(t)$  to converge to  $\Delta_1^{[i,j]}$  and  $\Delta_2^{[i,j]}$  for  $(i, j) \in \mathcal{E}$ , where  $\Delta_1^{[i,j]}, \Delta_2^{[i,j]} \in \mathbb{R}$  are the desired differences between  $w_1^{[i]}$  and  $w_1^{[j]}$ , satisfying  $\Delta_1^{[i,j]} = -\Delta_1^{[j,i]}$  and  $\Delta_2^{[i,j]} = -\Delta_2^{[j,i]}$ . It is naturally assumed that  $\Delta_1^{[i,j]}$  and  $\Delta_2^{[i,j]}$  are chosen appropriately such that the resulting formation is feasible.

We design the  $i$ th *coordinating guiding vector field*  $\mathfrak{X}^{[i]} : \mathbb{R}^{n+2N} \rightarrow \mathbb{R}^{n+2}$  to be the weighted sum of the surface-navigation vector field  $\text{sf}\chi^{[i]}$  and the coordination components  $\text{cr}\chi_1^{[i]}$  and  $\text{cr}\chi_2^{[i]}$  as follows:

$$\begin{aligned} \mathfrak{X}^{[i]}(\xi^{[i]}, w_1^{[i]}, w_2^{[i]}) &= \text{sf}\chi^{[i]}(\xi^{[i]}) + k_{c1} \text{cr}\chi_1^{[i]}(w_1^{[i]}) \\ &\quad + k_{c2} \text{cr}\chi_2^{[i]}(w_2^{[i]}) \end{aligned} \quad (17)$$

where  $k_{c1}, k_{c2} > 0$  are parameters to adjust the contribution of  $\text{sf}\chi^{[i]}, \text{cr}\chi_1^{[i]}$  and  $\text{cr}\chi_2^{[i]}$  to  $\mathfrak{X}^{[i]}$ .

We stack all the robot states as a vector  $\xi := (\xi^{[1]^\top}, \dots, \xi^{[N]^\top})^\top \in \mathbb{R}^{(n+2)N}$  and stack all the coordinating guiding vector fields as  $\mathfrak{X}(t, \xi) := (\mathfrak{X}^{[1]^\top}, \dots, \mathfrak{X}^{[N]^\top})^\top \in \mathbb{R}^{(n+2)N}$ . We need to investigate the integral curves of  $\mathfrak{X}(\xi)$ , i.e., the trajectories of the differential equation:  $\dot{\xi} = \mathfrak{X}(\xi)$ , given an initial condition  $\xi_0 \in \mathbb{R}^{(n+2)N}$  at  $t = t_0 \geq 0$ . Since the coordination functions in (16) are time-invariant, the system (18) is an autonomous system

$$\dot{\xi} = \mathfrak{X}(\xi). \quad (18)$$

Now we can formally formulate the problem.

**Problem 2 (Multirobot surface navigation):** Design the coordinating guiding vector field  $\mathfrak{X}^{[i]}$  in (17) for  $i \in \mathbb{Z}_1^N$  such that the trajectories of (18), given an initial condition  $\xi_0 \in \mathbb{R}^{(n+2)N}$  at  $t = t_0 \geq 0$ , fulfill the following three control objectives.

- (*Surface convergence*) Robot  $i$ 's surface-convergence errors tend to zero asymptotically for  $i \in \mathbb{Z}_1^N$ . Namely,  $\|\Phi^{[i]}(\xi^{[i]}(t))\| \rightarrow 0$  as  $t \rightarrow \infty$  for  $i \in \mathbb{Z}_1^N$ .

<sup>5</sup>As the first footnote, we do not require the knowledge of all virtual coordinates but only those of the neighbors  $w_1^{[j]}$  or  $w_2^{[j]}$  for  $j \in \mathcal{N}_i$ .

- 2) (*Motion coordination*) Each robot's motion is coordinated distributedly subject to the communication graph  $\mathcal{G}$  (i.e., Robot  $i$  can communicate with Robot  $j$  if and only if  $(i, j) \in \mathcal{E}$ ) such that their additional virtual coordinates satisfy  $w_1^{[i]}(t) - w_1^{[j]}(t) - \Delta_1^{[i,j]} \rightarrow 0$  and  $w_2^{[i]}(t) - w_2^{[j]}(t) - \Delta_2^{[i,j]} \rightarrow 0$  as  $t \rightarrow \infty$  for  $(i, j) \in \mathcal{E}$ .
- 3) (*Surface maneuvering*) Given desired parametric speeds  $\dot{w}_1^* \in \mathbb{R}$  and  $\dot{w}_2^* \in \mathbb{R}$ , the robot motion can achieve these speeds, i.e.,  $\dot{w}_1^{[i]}(t) \rightarrow \dot{w}_1^*$  and  $\dot{w}_2^{[i]}(t) \rightarrow \dot{w}_2^*$  as  $t \rightarrow \infty$  for  $i \in \mathbb{Z}_1^N$ .

Given the surface-navigation vector field in (15), we will design the coordination function  $c_1^{[i]}(\cdot), c_2^{[i]}(\cdot)$  later such that they coordinate the robots' motions but do not affect the surface-navigation performance.

### C. Time-Invariant Coordination Component

In this section, we will design time-invariant coordination components  $c_1^{[i]}(\mathbf{w}_1^{[\cdot]}), c_2^{[i]}(\mathbf{w}_2^{[\cdot]})$  and analyze the trajectories of (18), which is an autonomous system.

1) *Coordination Component*: Given the desired surface  $\mathcal{S}^{[i]}$ , we can design the desired parametric differences  $\Delta_1^{[i,j]}$  and  $\Delta_2^{[i,j]}$  starting from particular reference configurations  $\mathbf{w}_1^* := (\mathbf{w}_1^{[1]*}, \dots, \mathbf{w}_1^{[N]*})^\top$  and  $\mathbf{w}_2^* := (\mathbf{w}_2^{[1]*}, \dots, \mathbf{w}_2^{[N]*})^\top$ . Hence,  $\Delta_1^* = D^\top \mathbf{w}_1^* \in \mathbb{R}^{|\mathcal{E}|}$  and  $\Delta_2^* = D^\top \mathbf{w}_2^* \in \mathbb{R}^{|\mathcal{E}|}$  are the stacked vectors of  $\Delta_1^{[i,j]}, \Delta_2^{[i,j]}, (i, j) \in \mathcal{E}$ , respectively, where  $D \in \mathbb{R}^{N \times |\mathcal{E}|}$  is an incidence matrix. We employ the following consensus control algorithm:

$$c_1^{[i]} = - \sum_{j \in \mathcal{N}_i} \left( w_1^{[i]} - w_1^{[j]} - \Delta_1^{[i,j]} \right) \quad (19a)$$

$$c_2^{[i]} = - \sum_{j \in \mathcal{N}_i} \left( w_2^{[i]} - w_2^{[j]} - \Delta_2^{[i,j]} \right) \quad (19b)$$

for  $i \in \mathbb{Z}_1^N$ . Equations (19) can be rewritten compactly as

$$c_1^{[\cdot]}(\mathbf{w}_1^{[\cdot]}) = -L(\mathbf{w}_1^{[\cdot]} - \mathbf{w}_1^*) = -L\tilde{\mathbf{w}}_1^{[\cdot]} \quad (20a)$$

$$c_2^{[\cdot]}(\mathbf{w}_2^{[\cdot]}) = -L(\mathbf{w}_2^{[\cdot]} - \mathbf{w}_2^*) = -L\tilde{\mathbf{w}}_2^{[\cdot]} \quad (20b)$$

where  $c_1^{[\cdot]}(\mathbf{w}_1^{[\cdot]}) = (c_1^{[1]}(\mathbf{w}_1^{[\cdot]}), \dots, c_1^{[N]}(\mathbf{w}_1^{[\cdot]}))^\top$ ,  $c_2^{[\cdot]}(\mathbf{w}_2^{[\cdot]}) = (c_2^{[1]}(\mathbf{w}_2^{[\cdot]}), \dots, c_2^{[N]}(\mathbf{w}_2^{[\cdot]}))^\top$ ,  $L = L(\mathcal{G})$  is the Laplacian matrix, and  $\tilde{\mathbf{w}}_1^{[\cdot]} = \mathbf{w}_1^{[\cdot]} - \mathbf{w}_1^*$ ,  $\tilde{\mathbf{w}}_2^{[\cdot]} = \mathbf{w}_2^{[\cdot]} - \mathbf{w}_2^*$ . Combining (15), (16), (17), and (19), we attain the coordinating guiding vector field  $\mathfrak{X}^{[i]}$  for  $i \in \mathbb{Z}_1^N$ . The vector field  $\mathfrak{X}^{[i]}$  only takes as inputs Robot  $i$ 's own states and its neighbors' virtual coordinates  $w_1^{[j]}, w_2^{[j]}$  for  $j \in \mathcal{N}_i$ .

*Remark 3:* As in Remark 2, from (16), (17), and (19), one observes that information exchange only involves the coordination components  $c_1^{[i]}, c_2^{[i]}$ , and Robot  $i$  transmits only two scalars  $w_1^{[i]}, w_2^{[i]}$  to the neighboring Robot  $j \in \mathcal{N}_i$ .

2) *Convergence Analysis*: In this section, we show that the coordinating guiding vector field (17) enables multiple robots to navigate to their desired surfaces while they are coordinated by

the virtual coordinates such that  $w_1^{[i]}(t) - w_1^{[j]}(t)$  converges to  $\Delta_1^{[i,j]}$ , and  $w_2^{[i]}(t) - w_2^{[j]}(t)$  converges to  $\Delta_2^{[i,j]}$  for  $(i, j) \in \mathcal{E}$  as  $t \rightarrow \infty$ . To achieve motion coordination on a surface, we impose the following mild assumption.

*Assumption 3:* The first derivatives  $\partial_{w_1} f_j^{[i]}(\cdot) := \frac{\partial f_j^{[i]}}{\partial w_1^{[i]}}$ ,  $\partial_{w_2} f_j^{[i]}(\cdot) := \frac{\partial f_j^{[i]}}{\partial w_2^{[i]}}$  and the second derivatives  $\partial_{w_1}^2 f_j^{[i]}(\cdot), \partial_{w_2}^2 f_j^{[i]}(\cdot), \partial_{w_1} w_2 f_j^{[i]}(\cdot), \partial_{w_2} w_1 f_j^{[i]}(\cdot)$ , where  $\partial_{w_k} w_l f_j^{[i]}(\cdot) := \frac{\partial^2 f_j^{[i]}}{\partial w_l^{[i]} \partial w_k^{[i]}}$  for  $k, l \in \{1, 2\}$ , are bounded for all  $i \in \mathbb{Z}_1^N, j \in \mathbb{Z}_1^n$ .

Similarly, we define the *composite error vector*  $\mathbf{e}$  for the surface navigation to be

$$\mathbf{e} = \left( \Phi^\top, \left( D^\top \tilde{\mathbf{w}}_1^{[\cdot]} \right)^\top, \left( D^\top \tilde{\mathbf{w}}_2^{[\cdot]} \right)^\top \right)^\top \in \mathbb{R}^{nN+2|\mathcal{E}|}. \quad (21)$$

Now we arrive at the following theorem.

*Theorem 2 (Motion coordination on surfaces):* Under Assumptions 1 and 3, and given constant desired parametric differences  $\Delta_1^{[i,j]}, \Delta_2^{[i,j]}$  for  $(i, j) \in \mathcal{E}$ , the coordinating guiding vector fields  $\mathfrak{X}^{[i]}$  for  $i \in \mathbb{Z}_1^N$  designed by combining (15), (16), (17), and (20), and choosing

$$\mathbf{v} = (-1)^{n+1} (0, \dots, 0, \dot{w}_2^*, -\dot{w}_1^*)$$

in (14), solve Problem 2 globally in the sense that the initial composite error  $\|\mathbf{e}(t=0)\|$  in (21) can be arbitrarily large.

*Proof:* See [39, App. B]. ■

*Remark 4:* The above analysis can further be extended for more parameters. For example, to define motion coordination confined in a three-dimensional manifold (e.g., a cube), one may choose three parameters  $w_i, i = 1, 2, 3$ . The design methodology and analysis techniques for the two-parameter case can be directly adopted for this case with more cumbersome notations. In addition, if one uses  $m \geq 2$  parameters, then one also needs to choose  $m-1$  extra vectors  $\mathbf{v}_1, \dots, \mathbf{v}_{m-1}$  as what has been shown in (14) such that the wedge product is well defined.

### V. EXTENDING THE VECTOR FIELD TO INCORPORATE COLLISION AVOIDANCE

The focus of our work is the introduction and rigorous analysis of coordinating guiding vector fields for following paths or navigating surfaces with motion coordination as discussed above. However, to demonstrate the flexibility and practicality of our approach, in this section, we briefly explain how our proposed approach can incorporate an existing collision avoidance algorithm. Namely, we can modify the *nominal* guiding vector field (5) in a minimally invasive way using *safety barrier certificates* [40]. Without loss of generality, we only consider the coordinating guiding vector field for path following. Namely, the number of parameters is 1, and thus, there is only one virtual coordinate for each robot.

For  $i, j \in \mathbb{Z}_1^N, i \neq j$ , we define a function

$$\mathfrak{h}_{ij}(\xi^{[i]}, \xi^{[j]}) = \left\| P(\xi^{[i]} - \xi^{[j]}) \right\|^2 - R^2$$

where  $R > 0$  is the safety distance,  $P$  is the projection matrix  $P = I - \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} \in \mathbb{R}^{(n+1) \times (n+1)}$ , and  $I$  is the identity matrix.

The function  $\mathfrak{h}_{ij}$  reflects whether two robots keep a safety distance. Therefore, we can define the *safety set*  $\mathfrak{S}$

$$\mathfrak{S} = \left\{ \left( \xi^{[1]\top}, \dots, \xi^{[N]\top} \right)^\top \in \mathbb{R}^{(n+1)N} : \mathfrak{h}_{ij} \left( \xi^{[i]}, \xi^{[j]} \right) \geq 0 \quad \forall i \neq j \right\}. \quad (22)$$

Using  $\mathfrak{h}_{ij}$ , we can define a barrier function [41]

$$B_{ij} \left( \xi^{[i]}, \xi^{[j]} \right) := \frac{1}{\mathfrak{h}_{ij} \left( \xi^{[i]}, \xi^{[j]} \right)} \quad (23)$$

for  $i \neq j$ ,  $i, j \in \mathbb{Z}_1^N$ . We want to modify the *nominal vector field*  $\mathfrak{X}^{[i]}$  such that the new vector field denoted by  $\check{\mathfrak{X}}^{[i]}$  also enables collision-avoidance, which is encoded in the safety barrier certificate. Specifically, the new vector field  $\check{\mathfrak{X}}^{[i]}$  is obtained by minimizing  $\frac{1}{2} \sum_{i=1}^N \|\check{\mathfrak{X}}^{[i]} - \mathfrak{X}^{[i]}\|^2$  subject to the constraint  $\dot{B}_{ij} \leq \alpha/B_{ij}$  where  $\alpha > 0$  is a constant, for  $j < i$  and  $i, j \in \mathbb{Z}_1^N$ . Using (23) and the equivalence  $\dot{B}_{ij} = -\dot{\mathfrak{h}}_{ij}/\mathfrak{h}_{ij}^2 = -[2(\xi^{[i]} - \xi^{[j]})^\top P^\top (\check{\mathfrak{X}}^{[i]} - \check{\mathfrak{X}}^{[j]})]/\mathfrak{h}_{ij}^2$ , the constraint  $\dot{B}_{ij} \leq \alpha/B_{ij}$  can be rewritten as

$$(\xi^{[j]} - \xi^{[i]})^\top P^\top \check{\mathfrak{X}}^{[i]} + (\xi^{[i]} - \xi^{[j]})^\top P^\top \check{\mathfrak{X}}^{[j]} \leq \frac{\alpha}{2} \mathfrak{h}_{ij}^3. \quad (24)$$

Therefore, the new vector field  $\check{\mathfrak{X}} := (\check{\mathfrak{X}}^{[1]\top}, \dots, \check{\mathfrak{X}}^{[N]\top})^\top \in \mathbb{R}^{(n+1)N}$  is calculated by the quadratic program as follows:

$$\begin{aligned} \min_{\check{\mathfrak{X}} \in \mathbb{R}^{(n+1)N}} \quad & \frac{1}{2} \sum_{i=1}^N \|\check{\mathfrak{X}}^{[i]} - \mathfrak{X}^{[i]}\|^2 \\ \text{s.t.} \quad & (24) \quad \forall j \in \mathcal{D}_i \end{aligned} \quad (\text{QP1})$$

where  $\mathcal{D}_i$  is the set of robots within distance  $R$  with respect to Robot  $i$ , i.e.,  $\mathcal{D}_i := \{j \in \mathbb{Z}_1^N : \mathfrak{h}_{ij} \leq 0, j \neq i\}$ . Note that the definition of  $\mathcal{D}_i$  is based on geographical distances, different from the neighborhood  $\mathcal{N}_i$ , which is based on a communication graph  $\mathcal{G}$ . If Robot  $i$  keeps safety distances to all other robots (i.e.,  $\mathfrak{h}_{ij} \geq 0$  for all  $j \neq i$ ), then the new vector field  $\check{\mathfrak{X}}^{[i]}$  is the same as the original one  $\mathfrak{X}^{[i]}$ . In addition, since the virtual coordinate is not considered in the constraint (24), there always holds  $\mathfrak{X}_{n+1}^{[i]} = \check{\mathfrak{X}}_{n+1}^{[i]}$  for  $i \in \mathbb{Z}_1^N$ . Note that this optimization problem is *centralized* in the sense that it requires the positions and the nominal vector fields of *all* robots. We can change it to the following *distributed* version:

$$\begin{aligned} \min_{\check{\mathfrak{X}}^{[i]} \in \mathbb{R}^{n+1}} \quad & \frac{1}{2} \|\check{\mathfrak{X}}^{[i]} - \mathfrak{X}^{[i]}\|^2 \\ \text{s.t.} \quad & (\xi^{[j]} - \xi^{[i]})^\top P^\top \check{\mathfrak{X}}^{[i]} \leq \frac{\alpha}{4} \mathfrak{h}_{ij}^3 \quad \forall j \in \mathcal{D}_i \end{aligned} \quad (\text{QP2})$$

for each robot  $i \in \mathbb{Z}_1^N$ . In this optimization problem, Robot  $i$  only needs to measure<sup>6</sup> the positions  $p^{[j]} = P\xi^{[j]}$  of nearby

<sup>6</sup>In practice, the measurement is always inaccurate. However, the robustness property of control barrier functions against perturbation [42] mitigate the consequence of inaccurate measurement.

robots for  $j \in \mathcal{D}_i$  and does *not* need to obtain other robots' nominal vector field  $\mathfrak{X}^{[j]}$ , which is only possible via communication. If all the constraints in (QP2) are satisfied, so are those in (QP1). Therefore, a feasible solution of (QP2) is also a feasible solution of (QP1). Assume that the initial positions of robots are in the safety set  $\mathfrak{S}$ , then it is guaranteed that all the robots will always be in the safety set  $\mathfrak{S}$  [40], and one observes that the quadratic program (QP2) is always feasible since  $\check{\mathfrak{X}}^{[i]} = \mathbf{0}$  remains a trivial solution. This implies that a robot can always keep stationary (i.e.,  $\dot{p}^{[i]} = P\check{\mathfrak{X}}^{[i]} = \mathbf{0}$ ) to maintain a safety distance to other robots even if the nominal vector field  $\mathfrak{X}^{[i]}$  attempted to drive the robot to move closer to others.

*Remark 5:* The incorporation of the collision avoidance behavior may slow down the convergence speed of the path-following/surface-navigation behavior. Robots may not even move if deadlocks persistently appear [40, Sec. VII]. In addition, to make sure that the coordinated path-following/surface-navigation behavior and the collision avoidance behavior are feasible, one should assume that the desired path/surface and the desired parametric distances have been appropriately designed such that when all robots follow the path or navigate the surface separated by the desired parametric distances, the Euclidean distance between each pair of them is greater than the safety distance  $R$ .

## VI. SATURATED CONTROLLER FOR A DUBINS-CAR-LIKE MODEL

If a robot's dynamics can be approximately modeled by the single-integrator model, then the coordinating guiding vector field in (5) can be used directly as the velocity input to the robot. For the unicycle model, one can use feedback linearization to transform it into the single-integrator model [43] to utilize the guiding vector field directly. However, we will design a controller for a Dubins-car-like model without using the feedback linearization technique. Note that the control algorithm design idea in this section is applicable to robot models whose motions are characterized by the robot's orientations, such as the car-like model and the underwater glider model [44]. These models (approximately) represent many different robotic systems in reality, and thus, the design methodology is widely applicable.

Different from the unicycle model, which allows backwards or stationary motion, we use the following Dubins-car-like 3-D model that describes the fixed-wing aircraft dynamics

$$\dot{p}_1^{[i]} = v \cos \theta^{[i]}, \quad \dot{p}_2^{[i]} = v \sin \theta^{[i]}, \quad \dot{p}_3^{[i]} = u_z^{[i]}, \quad \dot{\theta}^{[i]} = u_\theta^{[i]} \quad (28)$$

where  $v$  is a constant speed,  $(p_1^{[i]}, p_2^{[i]}, p_3^{[i]})^\top \in \mathbb{R}^3$  is the position of the  $i$ th aircraft's center of mass,  $\theta^{[i]}$  is the yaw angle, and  $u_z^{[i]}$  and  $u_\theta^{[i]}$  are two control inputs to be designed. Since the essential role of a guiding vector field is to provide the desired yaw angle to guide the flight of a fixed-wing aircraft, the core idea behind the control algorithm design is to align the aircraft's flying direction with that given by the guiding vector field. In the sequel, we consider the case where there are two parameters and design a control algorithm for fixed-wing aircraft to navigate

on a surface.<sup>7</sup> Therefore, the coordinating guiding vector field has two additional coordinates, and thus, one needs to add two additional virtual coordinates  $p_4^{[i]}$  and  $p_5^{[i]}$  such that the aircraft's generalized position is  $(p_1^{[i]}, p_2^{[i]}, p_3^{[i]}, p_4^{[i]}, p_5^{[i]}) \in \mathbb{R}^5$ . Correspondingly, its generalized velocity is  $(\dot{p}_1^{[i]}, \dot{p}_2^{[i]}, \dot{p}_3^{[i]}, \dot{p}_4^{[i]}, \dot{p}_5^{[i]}) = (v \cos \theta^{[i]}, v \sin \theta^{[i]}, u_z^{[i]}, \dot{p}_4^{[i]}, \dot{p}_5^{[i]}),$  where  $\dot{p}_4^{[i]}, \dot{p}_5^{[i]}$  are virtual, acting as extra design freedom. To align the aircraft heading  $v(\cos \theta^{[i]}, \sin \theta^{[i]})$  with the counterpart of the coordinating guiding vector field  $\underline{\mathfrak{X}}^{[i]}$ , we need to "partially normalize" the vector field  $\underline{\mathfrak{X}}^{[i]}$  such that its first two entries form a vector of the same length as  $v$ , i.e.,

$$v\underline{\mathfrak{X}}^{[i]} := v\underline{\mathfrak{X}}^{[i]} / \sqrt{\underline{\mathfrak{X}}_1^{[i]2} + \underline{\mathfrak{X}}_2^{[i]2}}.$$

Subsequently, we need to design the yaw angular control input  $u_\theta^{[i]}$  such that the aircraft heading  $v(\cos \theta^{[i]}, \sin \theta^{[i]})$  gradually aligns with the vector formed by the first two entries of  $v\underline{\mathfrak{X}}^{[i]}$  [i.e.,  $v(\underline{\mathfrak{X}}_1^{[i]}, \underline{\mathfrak{X}}_2^{[i]})$ ]. For the last three entries of the generalized velocity  $(\dot{p}_1^{[i]}, \dot{p}_2^{[i]}, \dot{p}_3^{[i]}, \dot{p}_4^{[i]}, \dot{p}_5^{[i]})$ , one can simply equate them with those of the "partially normalized" vector field  $v\underline{\mathfrak{X}}^{[i]}$

<sup>7</sup>For the one-parameter case, see [28, Th. 2].

respectively, i.e.,

$$\begin{aligned} \dot{p}_3^{[i]} &= u_z^{[i]} = v\underline{\mathfrak{X}}_3^{[i]} / \sqrt{\underline{\mathfrak{X}}_1^{[i]2} + \underline{\mathfrak{X}}_2^{[i]2}} \\ \dot{p}_4^{[i]} &= v\underline{\mathfrak{X}}_4^{[i]} / \sqrt{\underline{\mathfrak{X}}_1^{[i]2} + \underline{\mathfrak{X}}_2^{[i]2}} \\ \dot{p}_5^{[i]} &= v\underline{\mathfrak{X}}_5^{[i]} / \sqrt{\underline{\mathfrak{X}}_1^{[i]2} + \underline{\mathfrak{X}}_2^{[i]2}}. \end{aligned} \quad (29)$$

This control algorithm design method [27], [31], [45] is extended here to handle the issue with the actuator saturation in the yaw angular control input  $u_\theta^{[i]}$ , as described in the following. First, we define the saturation function  $\text{Sat}_a^b : \mathbb{R} \rightarrow \mathbb{R}$  by  $\text{Sat}_a^b(x) = x$  for  $x \in [a, b]$ ,  $\text{Sat}_a^b(x) = a$  for  $x \in (-\infty, a)$  and  $\text{Sat}_a^b(x) = b$  for  $x \in (b, \infty)$ , where  $a, b \in \mathbb{R}$ ,  $a < b$  are some constants. Although the saturation function  $\text{Sat}_a^b$  is not differentiable, it is Lipschitz continuous. For convenience, we call the time interval when  $\text{Sat}_a^b(x(t)) = b$  the *upper saturation period*, and the time interval when  $\text{Sat}_a^b(x(t)) = a$  the *lower saturation period*. We use the notation  $\bar{\mathbf{v}}$  to denote the normalization of a vector  $\mathbf{v}$  (i.e.,  $\bar{\mathbf{v}} = \mathbf{v}/\|\mathbf{v}\|$ ). We also define

$$\underline{\mathfrak{X}}_p^{[i]} = (\bar{\underline{\mathfrak{X}}}^{[i]}_1, \bar{\underline{\mathfrak{X}}}^{[i]}_2)^\top \quad (30)$$

which is the vector formed by the first two entries of the normalized vector field  $\bar{\underline{\mathfrak{X}}}^{[i]}$ . Therefore,  $\underline{\mathfrak{X}}_p^{[i]}$  represents the orientation given by the vector field  $\underline{\mathfrak{X}}^{[i]}$ . In addition, one can easily calculate that  $\bar{\underline{\mathfrak{X}}}^{[i]} = (\underline{\mathfrak{X}}_1^{[i]}, \underline{\mathfrak{X}}_2^{[i]})$ .

$$\begin{aligned} J \left( {}^{\text{sf}} \chi^{[i]} \right) &= \frac{\partial {}^{\text{sf}} \chi^{[i]}}{\partial \zeta^{[i]}} = \\ &\left[ \begin{array}{ccc|c} -k_1^{[i]} & 0 & 0 & | & -v_{n+2} \partial_{w_1}^{w_1} f_1^{[i]} + v_{n+1} \partial_{w_2}^{w_1} f_1^{[i]} + k_1 \partial_{w_1} f_1^{[i]} \\ 0 & -k_2^{[i]} & 0 & | & -v_{n+2} \partial_{w_1}^{w_1} f_2^{[i]} + v_{n+1} \partial_{w_2}^{w_1} f_2^{[i]} + k_1 \partial_{w_1} f_2^{[i]} \\ 0 & 0 & -k_3^{[i]} & | & -v_{n+2} \partial_{w_1}^{w_1} f_3^{[i]} + v_{n+1} \partial_{w_2}^{w_1} f_3^{[i]} + k_1 \partial_{w_1} f_3^{[i]} \\ \hline k_1^{[i]} \partial_{w_1} f_1^{[i]} & k_2^{[i]} \partial_{w_1} f_2^{[i]} & k_3^{[i]} \partial_{w_1} f_3^{[i]} & | & \sum_{j=1}^3 [k_j^{[i]} \phi_j^{[i]} \partial_{w_1} f_j^{[i]} - k_j^{[i]} (\partial_{w_1} f_j^{[i]})^2] \\ k_1^{[i]} \partial_{w_2} f_1^{[i]} & k_2^{[i]} \partial_{w_2} f_2^{[i]} & k_3^{[i]} \partial_{w_2} f_3^{[i]} & | & \sum_{j=1}^3 [k_j^{[i]} \phi_j^{[i]} \partial_{w_2} f_j^{[i]} - k_j^{[i]} \partial_{w_1} f_j^{[i]} \partial_{w_2} f_j^{[i]}] \\ \end{array} \right. \\ &\quad \left. \begin{array}{c} | \\ -v_{n+2} \partial_{w_1}^{w_2} f_1^{[i]} + v_{n+1} \partial_{w_2}^{w_2} f_1^{[i]} + k_1 \partial_{w_2} f_1^{[i]} \\ -v_{n+2} \partial_{w_1}^{w_2} f_2^{[i]} + v_{n+1} \partial_{w_2}^{w_2} f_2^{[i]} + k_1 \partial_{w_2} f_2^{[i]} \\ -v_{n+2} \partial_{w_1}^{w_2} f_3^{[i]} + v_{n+1} \partial_{w_2}^{w_2} f_3^{[i]} + k_1 \partial_{w_2} f_3^{[i]} \\ \hline \sum_{j=1}^3 [k_j^{[i]} \phi_j^{[i]} \partial_{w_2} f_j^{[i]} - k_j^{[i]} \partial_{w_1} f_j^{[i]} \partial_{w_2} f_j^{[i]}] \\ \sum_{j=1}^3 [k_j^{[i]} \phi_j^{[i]} \partial_{w_2} f_j^{[i]} - k_j^{[i]} (\partial_{w_2} f_j^{[i]})^2] \end{array} \right] \quad (25) \end{aligned}$$

$$J \left( {}^{\text{cr}} \chi_1^{[i]} \right) = \left[ \begin{array}{c|c} \mathbf{0}_{5 \times 3} & \mathbf{0}_{3 \times 2N} \\ \hline a_{11} & \cdots & a_{i(i-1)} & -\sum_{j=1}^N a_{ij} & a_{i(i+1)} & \cdots & a_{iN} & \mathbf{0}_{1 \times N} \\ \hline & \cdots & & \cdots & & & & \cdots & \\ \hline & & & \mathbf{0}_{1 \times 2N} & & & & & \end{array} \right] = \left[ \begin{array}{c|c} & \mathbf{0}_{3 \times 2N} \\ \hline \mathbf{0}_{5 \times 3} & -\mathbf{b}_i^\top L \mathbf{0}_{1 \times N} \\ \hline & \mathbf{0}_{1 \times 2N} \end{array} \right]. \quad (26)$$

$$J \left( {}^{\text{cr}} \chi_2^{[i]} \right) = \left[ \begin{array}{c|c} \mathbf{0}_{5 \times 3} & \mathbf{0}_{4 \times 2N} \\ \hline \mathbf{0}_{1 \times N} & a_{11} \cdots a_{i(i-1)} \\ \hline & -\sum_{j=1}^N a_{ij} & a_{i(i+1)} & \cdots & a_{iN} \end{array} \right] = \left[ \begin{array}{c|c} & \mathbf{0}_{4 \times 2N} \\ \hline \mathbf{0}_{5 \times 3} & -\mathbf{b}_i^\top L \mathbf{0}_{1 \times N} \end{array} \right]. \quad (27)$$

Suppose we are given physical desired surfaces  $\text{phy} \mathcal{S}^{[i]} \subseteq \mathbb{R}^3$  parameterized by (13), where  $n = 3$ , and the coordinating guiding vector field  $\mathfrak{X}^{[i]} : \mathbb{R}^{3+2N} \rightarrow \mathbb{R}^{3+2}$  in (17). Denote the orientation of the aircraft by  $\mathbf{h}^{[i]} = \overline{\mathbf{h}^{[i]}} = (\cos \theta^{[i]}, \sin \theta^{[i]})^\top$ , the (signed) angle difference directed from  $\overline{\mathfrak{X}_p^{[i]}}$  to  $\overline{\mathbf{h}^{[i]}}$  by  $\sigma^{[i]} \in (-\pi, \pi]$ , and define the rotation matrix  $E = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$ .

The following theorem states that the aircraft's orientation  $\overline{\mathbf{h}^{[i]}}$  will converge to that of the vector field  $\overline{\mathfrak{X}_p^{[i]}}$  asymptotically (i.e.,  $\sigma^{[i]}$  converges to zero).

*Theorem 3:* Assume that the vector field satisfies  $\mathfrak{X}_1^{[i]}(\xi^{[i]})^2 + \mathfrak{X}_2^{[i]}(\xi^{[i]})^2 > \gamma > 0$  for  $\xi^{[i]} \in \mathbb{R}^{3+2}$ ,  $i \in \mathbb{Z}_1^N$ , where  $\gamma$  is a positive constant. Let the angular velocity control input  $u_\theta^{[i]}$  in model (28) be

$$\dot{\theta}_d^{[i]} = u_\theta^{[i]} = \text{Sat}_a^b \left( \dot{\theta}_d^{[i]} - k_\theta \overline{\mathbf{h}^{[i]}}^\top E \overline{\mathfrak{X}_p^{[i]}} \right) \quad (31)$$

where

$$\dot{\theta}_d^{[i]} = -\overline{\mathfrak{X}_p^{[i]}}^\top E \dot{\mathfrak{X}_p^{[i]}} / \|\overline{\mathfrak{X}_p^{[i]}}\| \quad (32)$$

$k_\theta > 0$  is a constant, and  $a < 0$ ,  $b > 0$  are constants for the saturation function  $\text{Sat}_a^b$ . If the angle difference  $\sigma^{[i]}$  satisfies the following conditions:

- 1) the initial angle difference  $\sigma^{[i]}(t = 0) \neq \pi$ ;
- 2)  $\sigma^{[i]}(t) \in [0, \pi)$  during the upper saturation period, and  $\sigma^{[i]}(t) \in (-\pi, 0]$  during the lower saturation period, then  $\sigma^{[i]}$  will vanish asymptotically (i.e.,  $\sigma^{[i]}(t) \rightarrow 0$ ).

*Proof:* See [39, App. C]. ■

*Remark 6:* The quantity  $\dot{\mathfrak{X}}_p^{[i]}$  in (32) can be calculated by

$$\dot{\mathfrak{X}}_p^{[i]} = J \left( \mathfrak{X}_p^{[i]} \right) \dot{\zeta}^{[i]} \quad (33)$$

where  $J \left( \mathfrak{X}_p^{[i]} \right) \in \mathbb{R}^{2 \times (3+2N)}$  is the Jacobian matrix of  $\mathfrak{X}_p^{[i]}$  with respect to the generalized position  $\zeta^{[i]} = (\xi^{[i-w]^\top}, w_1^{[i]^\top}, w_2^{[i]^\top})^\top = (x_1^{[i]}, x_2^{[i]}, x_3^{[i]}, w_1^{[i]^\top}, w_2^{[i]^\top})^\top \in \mathbb{R}^{3+2N}$ , with  $\xi^{[i-w]}$  representing the vector obtained by deleting the last two entries (i.e., the virtual coordinates  $w_1^{[i]}$  and  $w_2^{[i]}$ ) of  $\xi^{[i]} \in \mathbb{R}^{3+2}$ . In addition, we can simplify the computation of the Jacobian  $\dot{\mathfrak{X}}_p^{[i]}$  to

$$\begin{aligned} J \left( \mathfrak{X}_p^{[i]} \right) &= F J \left( \overline{\mathfrak{X}_p^{[i]}} \right) = F \left( I - \overline{\mathfrak{X}_p^{[i]}} \overline{\mathfrak{X}_p^{[i]}}^\top \right) J \left( \mathfrak{X}_p^{[i]} \right) / \|\overline{\mathfrak{X}_p^{[i]}}\| \\ &= F \left( I - \overline{\mathfrak{X}_p^{[i]}} \overline{\mathfrak{X}_p^{[i]}}^\top \right) J \left( {}^{\text{sf}} \chi^{[i]} + k_{c1} {}^{\text{cr}} \chi_1^{[i]} \right. \\ &\quad \left. + k_{c2} {}^{\text{cr}} \chi_2^{[i]} \right) / \|\overline{\mathfrak{X}_p^{[i]}}\| \\ &= F \left( I - \overline{\mathfrak{X}_p^{[i]}} \overline{\mathfrak{X}_p^{[i]}}^\top \right) \left( J \left( {}^{\text{sf}} \chi^{[i]} \right) + k_{c1} J \left( {}^{\text{cr}} \chi_1^{[i]} \right) \right. \\ &\quad \left. + k_{c2} J \left( {}^{\text{cr}} \chi_2^{[i]} \right) \right) / \|\overline{\mathfrak{X}_p^{[i]}}\| \end{aligned}$$

where  $F = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$  and the Jacobians  $J({}^{\text{sf}} \chi^{[i]})$ ,  $J({}^{\text{cr}} \chi_1^{[i]})$ , and  $J({}^{\text{cr}} \chi_2^{[i]})$  are shown in (25), (26), and (27) at the

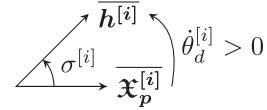


Fig. 2. Signed angle  $\sigma^{[i]}$ .

bottom of the previous page, respectively (where  $b_i$  is the basis vector with the  $i$ th entry being 1). Although (33) contains states  $\dot{w}_1^{[j]}$  and  $\dot{w}_2^{[j]}$  from all robots, we emphasize that *each robot only needs the information  $w_1^{[j]}$ ,  $w_2^{[j]}$ ,  $\dot{w}_1^{[j]}$ , and  $\dot{w}_2^{[j]}$  from its neighbors (i.e.,  $j \in \mathcal{N}_i$ ), and thus the control algorithm is distributed*. This is because in (26) and (27), those terms  $a_{ij}$  become zero if  $j$  is not a neighbor of the  $i$ th robot, and thereby, the corresponding information  $w_1^{[j]}$ ,  $w_2^{[j]}$ ,  $\dot{w}_1^{[j]}$ , and  $\dot{w}_2^{[j]}$  is not required by the  $i$ th robot. In particular, if the communication graph is a cycle, then each robot only needs the information of  $w_1^{[j]}$ ,  $w_2^{[j]}$ ,  $\dot{w}_1^{[j]}$ , and  $\dot{w}_2^{[j]}$  from its two neighbors, regardless of the size of the multirobot system.

*Remark 7 (Interpretation of Condition 2):* In the proof of the theorem, it has been shown that

$$\dot{\overline{\mathfrak{X}_p^{[i]}}} = \dot{\theta}_d^{[i]} E \overline{\mathfrak{X}_p^{[i]}}.$$

Since  $E \overline{\mathfrak{X}_p^{[i]}}$  is orthogonal to  $\overline{\mathfrak{X}_p^{[i]}}$ , the quantity  $\dot{\theta}_d^{[i]}$  in (31) encodes how fast the vector field changes its orientation along the trajectory of the aircraft (i.e.,  $\dot{\theta}_d^{[i]}$  is the *change rate of the vector field orientation*). Since  $\overline{\mathbf{h}^{[i]}}^\top E \overline{\mathfrak{X}_p^{[i]}} \leq 1$  in (31), saturation may happen due to the possibly large magnitude of the term  $\dot{\theta}_d^{[i]}$ . If upper saturation happens at  $t = t_0$ , the change rate of the vector field orientation  $\dot{\theta}_d^{[i]}$  demands a faster change than the aircraft can achieve; therefore, if  $\sigma^{[i]}(t_0) > 0$ , then the vector field orientation encoded by  $\overline{\mathfrak{X}_p^{[i]}}$  will “chase” the aircraft orientation vector  $\overline{\mathbf{h}^{[i]}}$  such that the angle difference  $\sigma^{[i]}(t)$  is decreasing (see Fig. 2). Thus, as long as  $\sigma^{[i]}(t_0) > 0$ , the Lyapunov function  $V$  is still decreasing. However, if the saturation lasts for a long time such that  $\overline{\mathfrak{X}_p^{[i]}}$  overtakes the aircraft orientation vector  $\overline{\mathbf{h}^{[i]}}$ , the angle  $\sigma^{[i]}(t)$  will become negative and violate Condition 2 in Theorem 3; thus, the decreasing property of the Lyapunov function  $V$  is not necessarily guaranteed; in this case,  $\dot{V} > -k_\theta (\overline{\mathbf{h}^{[i]}}^\top E \overline{\mathfrak{X}_p^{[i]}})^2$ , so  $\dot{V}$  may be negative or positive. Therefore, although Condition 2 in Theorem 3 might be *difficult* to check in practice, it conveys the core message that the saturation, albeit allowed, should not last for a long time. However, this condition is only sufficient (not necessary), while in practice, violating this condition does not immediately entail instability of the algorithm. As the aircraft is guided by the vector field, the aircraft can reorient its heading toward the desired path even if it temporarily deviates due to saturation or other constraints (such as path curvature) as long as the change rate of the vector field orientation  $\dot{\theta}_d^{[i]}$  does not saturate persistently. The subsequent fixed-wing aircraft experiment verifies the effectiveness of the control law.

Nevertheless, we can remove Condition 2 in Theorem 3 by imposing an upper bound on the magnitude of  $\dot{\theta}_d^{[i]}$ , as shown in Corollary 1.

*Corollary 1:* Suppose there exists a positive constant  $d$  satisfying  $d < \min\{-a, b\}$ , where  $a, b$  are the threshold values of the saturation function  $\text{Sat}_a^b(\cdot)$  in Theorem 3 (note that  $a < 0$ ) such that the change rate of the vector field orientation  $|\dot{\theta}_d^{[i]}| \leq d$ . Let  $\bar{k}_\theta := \min\{-a - d, b - d\} > 0$ . If the positive gain  $k_\theta$  in (31) is chosen within the range  $(0, \bar{k}_\theta)$ , then the angle difference  $\sigma^{[i]}$  converges to 0 without requiring Condition 2 in Theorem 3.

*Proof:* See [39, App. D]. ■

To reduce the magnitude of  $\dot{\theta}_d^{[i]}$  and avoid possible saturation, one can scale down the path parameter in the parametric functions in (2) (e.g., by changing  $f_j^{[i]}(w^{[i]})$  to  $f_j^{[i]}(\beta w^{[i]})$ , where  $0 < \beta < 1$ ), or choose another desired path with a smaller curvature. Another approach to avoid input saturation is to add an additional constraint  $|u_\theta^{[i]}| \leq \min\{|a|, |b|\}$  in the quadratic program in Section V.

## VII. SIMULATIONS AND EXPERIMENTS

### A. Simulations

In all simulations, the communication topology is a *cycle graph*, and thus, each robot is only allowed to communicate with its two adjacent neighbors.

In the first simulation, we let  $N = 50$  robots follow a 3-D (i.e.,  $n = 3$ ) *self-intersecting* bent “∞”-shaped curve parameterized by  $x_1^{[i]} = 15 \sin(2w^{[i]})$ ,  $x_2^{[i]} = 30 \sin(w^{[i]})\sqrt{0.5(1 - 0.5 \sin^2(w^{[i]}))}$ , and  $x_3^{[i]} = 5 + 5 \cos(2w^{[i]}) - 2$  for  $i \in \mathbb{Z}_1^N$ . The period of this closed curve is  $T = 2\pi$  and the desired differences between two adjacent robots’ virtual coordinates are  $T/(2N)$ . We construct the desired parametric differences  $\Delta^{[i,j]}$  from the reference  $w^{[i]*} = (i-1)T/(2N)$  for  $i \in \mathbb{Z}_1^N$ . The control gains for the coordinating vector field are:  $k_1^{[i]} = k_2^{[i]} = k_3^{[i]} = 1$ ,  $k_c = 300$  for  $i \in \mathbb{Z}_1^N$ , where  $k_c$  is large to accelerate the motion coordination. As shown in Fig. 3, all robots follow the “∞”-shaped path successfully and keep desired positions (in terms of  $w^{[i]}$ ) between each other. The path-following errors and the coordination errors converge to zero eventually.

In the second simulation, we aim to show that our algorithm is also applicable to path-following control of complicated and *open* curves (i.e., aperiodic curve) and demonstrate its potential application to volume coverage in 3-D. To this end, we choose a 3-D Lissajous curve (i.e.,  $n = 3$ ) with *irrational* coefficients, of which the parametric equations are  $x_1^{[i]} = \cos(n_x w^{[i]}) + m_x$ ,  $x_2^{[i]} = \cos(n_y w^{[i]}) + m_y$ , and  $x_3^{[i]} = \cos(n_z w^{[i]}) + m_z$  for  $i \in \mathbb{Z}_1^N$ , and the coefficients are chosen as  $n_x = \sqrt{2}$ ,  $n_y = 4.1$ ,  $n_z = 7.1$ ,  $m_x = 0.1$ ,  $m_y = 0.7$ ,  $m_z = 0$ . This is an open curve bounded in a cube (as  $n_x$  is irrational). Therefore, it is ideal for a volume coverage task. To clearly illustrate this idea, we choose  $N = 3$  robots and simulate for 100 s. The control gains for the coordinating guiding vector field are:  $k_1^{[i]} = k_2^{[i]} = k_3^{[i]} = k_c = 1$  for  $i \in \mathbb{Z}_1^N$ . We construct the desired parametric differences  $\Delta^{[i,j]}$  from the references  $w^{[i]*}(t) = (i-1)2\pi/N$ ,

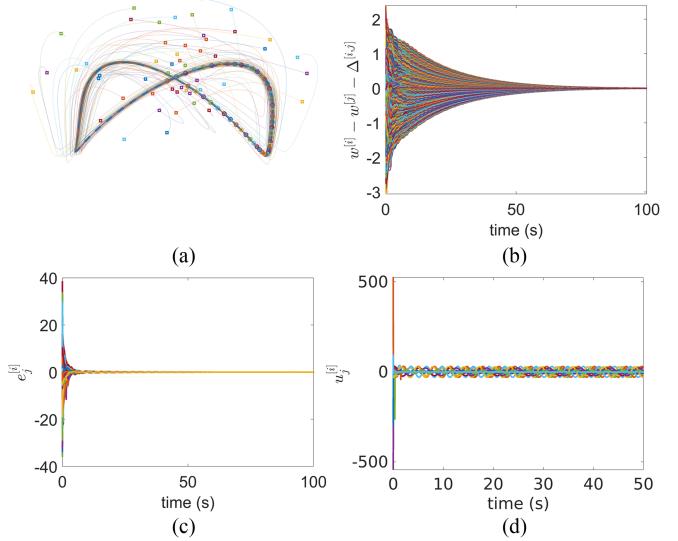


Fig. 3. First simulation results. (a) Trajectories of robots, where squares and circles symbolize the trajectories’ initial and final positions, respectively. (b) Coordination errors  $w^{[i]} - w^{[j]} - \Delta^{[i,j]}$  for  $i, j \in \mathbb{Z}_1^{50}, i < j$ . (c) Path-following errors  $\phi_j^{[i]}$  for  $i \in \mathbb{Z}_1^{50}$  and  $j \in \mathbb{Z}_1^3$ . (d) Control inputs  $u_j^{[i]} := \dot{x}_j^{[i]}$  for  $i = 1, 13, 25, 37, 50$ ,  $j \in \mathbb{Z}_1^{3+1}$ . Only 50 s of data are shown for clarity in (c) and (d).

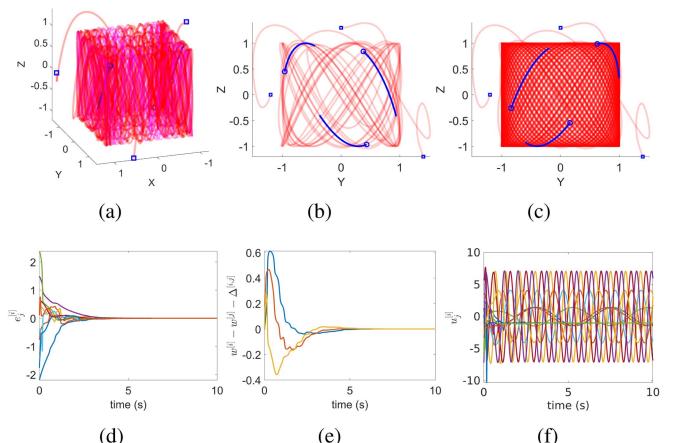


Fig. 4. Second simulation results. Squares and circles symbolize trajectories’ initial and final positions, and the solid blue lines are the trajectories during the last 30 time steps. (a) Trajectories of three robots follow an open Lissajous curve with irrational coefficients. The magenta curve represents part of the Lissajous curve. (b) and (c) Y-Z side views of the trajectories at time 9.4 and 76.8 s, respectively. (d) Path-following errors  $\phi_j^{[i]}$  for  $i \in \mathbb{Z}_1^3$  and  $j \in \mathbb{Z}_1^3$ . (e) Coordination error  $w^{[i]} - w^{[j]} - \Delta^{[i,j]}$  for  $i, j \in \mathbb{Z}_1^3, i < j$ . (f) Control inputs  $u_j^{[i]} := \dot{x}_j^{[i]}$  for  $i \in \mathbb{Z}_1^3$ ,  $j \in \mathbb{Z}_1^{3+1}$ . Only 10 s of data are shown for clarity in (d)–(f).

for  $i \in \mathbb{Z}_1^N$ . Since the Lissajous curve will fill the whole cube as the path parameter  $w^{[i]}$  varies from 0 to infinity, for illustration purpose, we only plot part of the desired path where  $w^{[i]} \in [0, 30\pi]$  (see the magenta curve in Fig. 4). As seen in Fig. 4, the three robots tend to cover the whole volume of the unit cube, and the path-following errors and coordination errors are almost zero after 20 s.

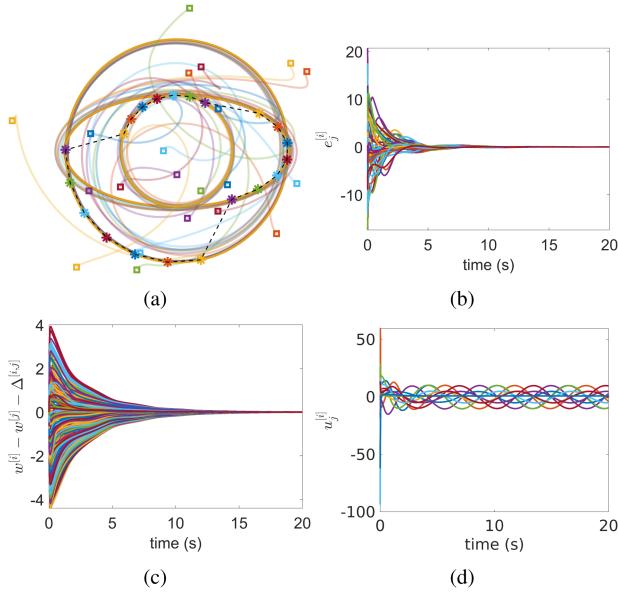


Fig. 5. Third simulation results. (a) Trajectories of robots, where squares and \* symbolize the trajectories' initial and final positions, respectively. The dashed black line shows the communication links between robots (i.e., a cycle graph). (b) Path-following errors  $\phi_j^{[i]}$  for  $i \in \mathbb{Z}_1^{21}$  and  $j \in \mathbb{Z}_2^2$ . (c) Coordination error  $w^{[i]} - w^{[j]} - \Delta^{[i,j]}$  for  $i, j \in \mathbb{Z}_1^{21}, i < j$ . (d) Control inputs  $u_j^{[i]} := \mathfrak{X}_j^{[i]}$  for  $i = 1, 6, 11, 16, 21, j \in \mathbb{Z}_1^{2+1}$ . Only 20 s of data are shown for clarity in (b)–(d).

In the third simulation, we show how different robots can follow different desired paths while they still coordinate their motions to form some formation shapes. We let  $N = 21$  robots follow three different paths, where the first seven robots follow a large circle of radius  $a > 0$  parameterized by  $x_1^{[i]} = a \cos w^{[i]}$ ,  $x_2^{[i]} = a \sin w^{[i]}$  for  $i = 1, \dots, 7$ , the last seven robots follow a small circle of radius  $0 < b < a$  parameterized by  $x_1^{[i]} = b \cos w^{[i]}$ ,  $x_2^{[i]} = b \sin w^{[i]}$  for  $i = 15, \dots, 21$ , and the remaining seven robots follow an ellipse with a semimajor axis  $a$  and a semiminor axis  $b$ , parameterized by  $x_1^{[i]} = a \cos w^{[i]}$ ,  $x_2^{[i]} = b \sin w^{[i]}$  for  $i = 8, \dots, 14$ . These three paths are concentric (see Fig. 5). The robots are coordinated in a distributed way such that they are equally separated in the path parameter  $w^{[i]}$ , i.e., we construct the desired parametric differences  $\Delta^{[i,j]}$  from the references  $w^{[i]*}(t) = (i-1)2\pi/N$ , for  $i \in \mathbb{Z}_1^N$ . Other parameters are chosen as  $a = 10$ ,  $b = 5$ ,  $k_1^{[i]} = k_2^{[i]} = 1$ ,  $k_c = 100$  for  $i \in \mathbb{Z}_1^N$ . An interesting feature is that during the steady-state, these robots will not overlap with each other, since overlapping happens if any two of the robots' virtual coordinates are equal, while the distributed coordination guarantees that the adjacent neighbors satisfy  $|w^{[i]} - w^{[j]}| = 2\pi/N$  for  $(i, j) \in \mathcal{E}$ . These robots successfully generate varying formation shapes: they cluster in different parts of the desired paths.

In the fourth simulation, we show how multiple robots converge to a torus and form a desired pattern while they maneuver on the torus. Specifically, we manually select  $N = 67$  reference points  $(w_1^{[i]*}, w_2^{[i]*})_{i \in \mathbb{Z}_1^N}$  such that they form a pattern resembling the four letters "ICRA." Then based on these reference points,

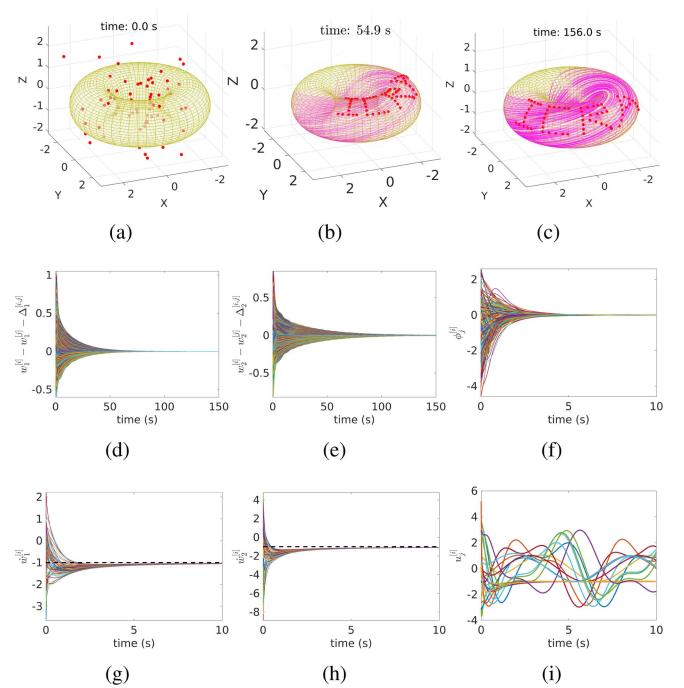


Fig. 6. Fourth simulation results. (a), (b) 67 robots, represented by red dots, converge to and maneuver on a torus while they form the pattern "ICRA" at different time instances. The magenta curves are the trajectories of robots. (c), (d) Coordination errors  $w_1^{[i]} - w_1^{[j]} - \Delta_1^{[i,j]}$  and  $w_2^{[i]} - w_2^{[j]} - \Delta_2^{[i,j]}$  for the two parameters for  $i, j \in \mathbb{Z}_1^{67}, i < j$  converge to zero eventually. (e) Surface-convergence errors  $\phi_j^{[i]}$  for  $i \in \mathbb{Z}_1^{67}$  and  $j \in \mathbb{Z}_1^3$  converge to zero eventually. (f), (g) Parametric velocities  $\dot{w}_1^{[i]}$  and  $\dot{w}_2^{[i]}$  converge to the desired values  $\dot{w}_1^* = \dot{w}_2^* = -1$  denoted by black dashed lines. (h) Control inputs  $u_j^{[i]} := \mathfrak{X}_j^{[i]}$  for  $i = 1, 17, 34, 50, 67, j \in \mathbb{Z}_1^{3+1}$ . Only 10 s of data are shown for clarity in (f)–(i).

one can calculate the desired parametric differences  $\Delta_1^*, \Delta_2^* \in \mathbb{R}^{67}$ . Note that Robot  $i$  only needs to know the desired parametric differences with respect to its neighbors; i.e.,  $\Delta_1^{[i,j]}, \Delta_2^{[i,j]}$  for  $j \in \mathcal{N}_i$ . These 67 robots are required to converge to a torus of which the parametric equations are  $x_1^{[i]} = (2 + \cos w_1^{[i]}) \cos w_2^{[i]}$ ,  $x_2^{[i]} = (2 + \cos w_1^{[i]}) \sin w_2^{[i]}$ ,  $x_3^{[i]} = \sin w_1^{[i]}$  for  $i \in \mathbb{Z}_1^N$  (hence  $n = 3$  in Theorem 2). They are also required to maneuver on the torus in the sense that  $\dot{w}_1^* = \dot{w}_2^* = -1$ . By Theorem 2, the extra vector  $v$  is thus chosen as  $v = (0, 0, 0, -1, 1)$ . Other parameters are chosen as  $k_1^{[i]} = k_2^{[i]} = 1, k_c = 10$  for  $i \in \mathbb{Z}_1^N$ . The simulation results are shown in Fig. 6.

## B. Experiments With a Pair of Fixed-Wing Aircraft

1) *Rendezvous on a 3-D Lissajous Curve:* In this experiment,<sup>8</sup> two autonomous fixed-wing aircraft (Autonomous Opterra 1.2 m similar to the Sonicmodell in Fig. 9) are employed to validate Theorems 1 and 3. The aircraft are equipped with open-source software/hardware components developed with *Paparazzi* [46], an open-source research community for the development of autonomous, mostly aerial,

<sup>8</sup>The code is available at [https://github.com/noether/paparazzi/tree/gvf\\_parametric](https://github.com/noether/paparazzi/tree/gvf_param_multi/sw/airborne/modules/guidance/gvf_parametric).

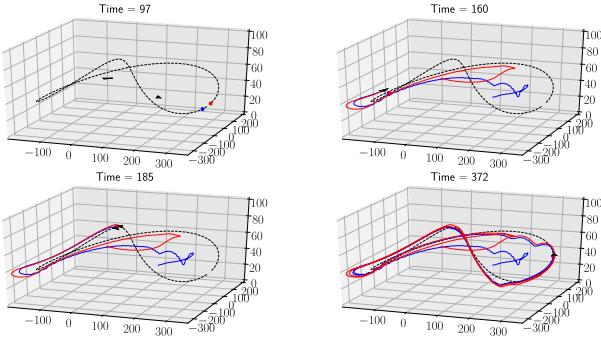


Fig. 7. Two aircraft (blue and red trajectories) fly together and follow a 3-D bent “∞”-shaped path. Although the subparts show that the aircraft’s positions are overlapped, in reality, they are different since the aircraft’s GPS receptors are biased with respect to each other by one meter in the XY plane. The red ( $i = 1$ ) and blue ( $i = 2$ ) dots represent  $(f_1^{[i]}, f_2^{[i]}, f_3^{[i]})$ . The displayed positions are in meters.

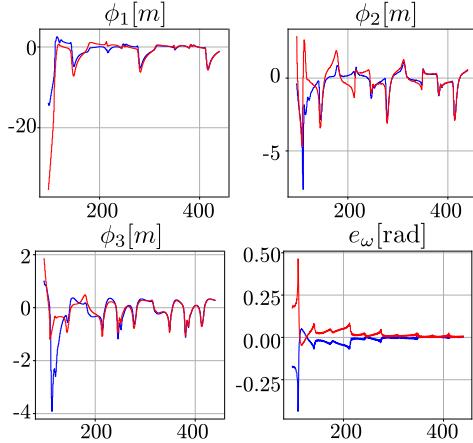


Fig. 8. Path-following errors of the two aircraft (blue and red) to the desired path, and the coordination error  $e_w$  with respect to the desired  $\Delta^{[i,j]} = 0$ . The horizontal axis represent time in seconds.

vehicles. We choose the following 3-D Lissajous curve  $f_1^{[i]}(w) = 225 \cos(w^{[i]}), f_2^{[i]}(w^{[i]}) = 225 \cos(2w^{[i]} + \pi/2), f_3^{[i]}(w^{[i]}) = -20 \cos(2w^{[i]}),$  for  $i = 1, 2,$  which has a bent “∞”-shaped path. The mission requires both aircraft to have  $\Delta^{[1,2]} = \Delta^{[2,1]} = 0$ , i.e., to fly in a tight formation. The overlapping at steady state is avoided by biasing the GPS measurement of one aircraft by a constant distance of 1 m in the horizontal plane, i.e., when the aircraft achieve  $\Delta^{[1,2]} = \Delta^{[2,1]} = 0$ , they are displaced physically. We choose  $k_1^{[i]} = k_2^{[i]} = 0.002, k_3^{[i]} = 0.0025, k_c = 0.01, k_\theta = 1,$  for  $i = 1, 2,$  and the communication frequency is 10 Hz. In the experiment, the weather forecast reported 14 °C and a South wind of 10 km/h. In Fig. 7, the telemetry shows that both aircraft fly in a tight formation and follow the path, and Fig. 8 depicts the path-following and coordination errors.

2) *Rendezvous on a Torus Surface:* Two autonomous fixed-wing aircraft (SonicModell in Fig. 9) are employed to validate Theorems 2 and 3. In particular, we proceed to show the rendezvous of two aircraft

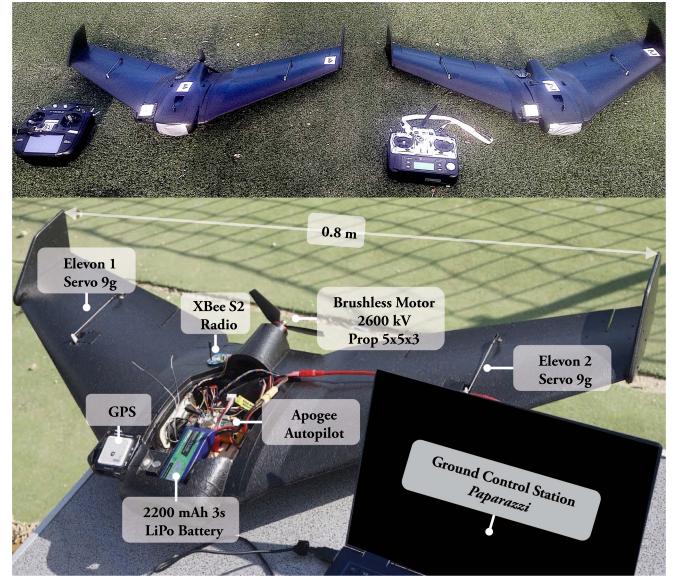


Fig. 9. Two aircraft SonicModell employed for the rendezvous on a torus.

on a torus surface.<sup>9</sup> The torus surface is parameterized with  $f_1^{[i]}(w_1, w_2) = (100 + 5 \cos(w_2^{[i]})) \cos(w_1^{[i]}), f_2^{[i]}(w_1, w_2) = (100 + 5 \cos(w_2^{[i]})) \sin(w_1^{[i]}), f_3^{[i]}(w_1, w_2) = 5 \sin(w_2^{[i]}) + 50$  m, for  $i = 1, 2$ , and we set  $w_2^* = 2w_1^* = 0.01$ , i.e., for each loop in the horizontal dimension, the aircraft will accomplish two loops in the vertical dimension. The rendezvous sets  $\Delta_1^{[1,2]} = \Delta_2^{[1,2]} = 0$ , and we have chosen  $k_1^{[i]} = k_2^{[i]} = k_3^{[i]} = 0.003, k_{c1} = k_{c2} = 0.01, k_\theta = 1$ , for  $i = 1, 2$ , and the air-to-air communication frequency is 10 Hz. In the experiment, the weather forecast reported 16 °C and an East wind of 10 km/h at the radio-control club of Arroyomolinos, Spain. In Fig. 10, the telemetry shows that both aircraft converge to fly together and follows the surface with the planned trajectory eventually. The oscillation around zero for  $\phi_{1,2,3}$  is due to the difficulty on the vertical speed controller for the selected type of aircraft that affects the airspeed. Note that if the horizontal speed changes, then the aircraft wait for each other and they do not track precisely the desired  $\Delta_1^{[1,2]}$  and  $\Delta_2^{[1,2]}$ . Nevertheless, Fig. 11 shows that the error for the desired altitude does not exceed half a meter, in the horizontal dimensions is around a meter, and the errors for the rendezvous are around 1° in  $w_1$  and less than 1° in  $w_2$ .

The two experiments show that once an aircraft flies far ahead of its partner, the algorithm guides the airplane to deviate from the curve to travel more distance, hence, to “wait” for its partner. Nevertheless, these deviations from the desired path are within the order of 1 or 2 m (see Figs. 8 and 11). The generated signals by the algorithm and the actual states of the aircraft can be seen in Fig. 12.

Note that the employed aircraft do *not* control their ground speeds. In fact, they have a reference signal in their throttle to keep a safe airspeed, and the aircraft increase/decrease such

<sup>9</sup>The code is available at [https://github.com/noether/paparazzi/tree/gvf\\_param\\_multi/sw/airborne/modules/guidance/gvf\\_parametric\\_surf](https://github.com/noether/paparazzi/tree/gvf_param_multi/sw/airborne/modules/guidance/gvf_parametric_surf). It is designed such that a generic surface can be added by an experienced user.

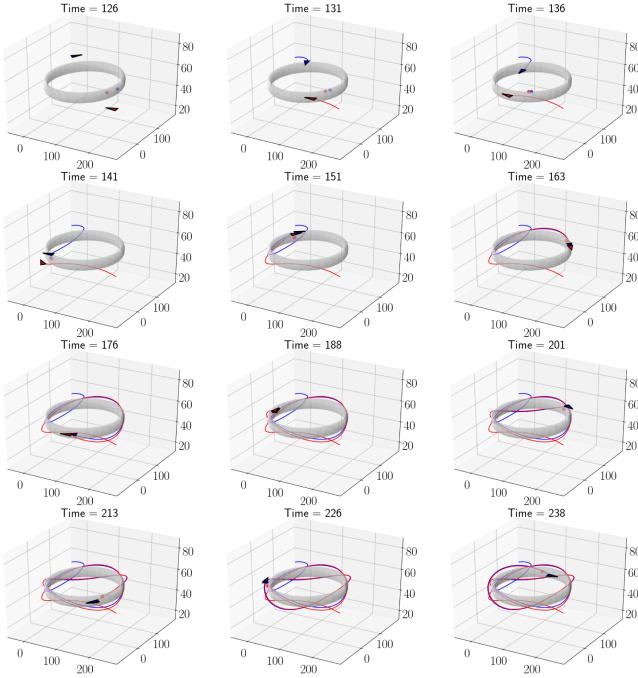


Fig. 10. Two aircraft (blue and red trajectories) rendezvous on a torus surface, where the desired trajectory on the torus corresponds to  $\dot{w}_2^* = 2\dot{w}_1^*$ . Although the synchronization is done on the same surface, one aircraft has its actual position biased in order to avoid collisions during the rendezvous. The red ( $i = 1$ ) and blue ( $i = 2$ ) dots represent  $(f_1^{[i]}, f_2^{[i]}, f_3^{[i]})$ . The displayed positions are in meters.

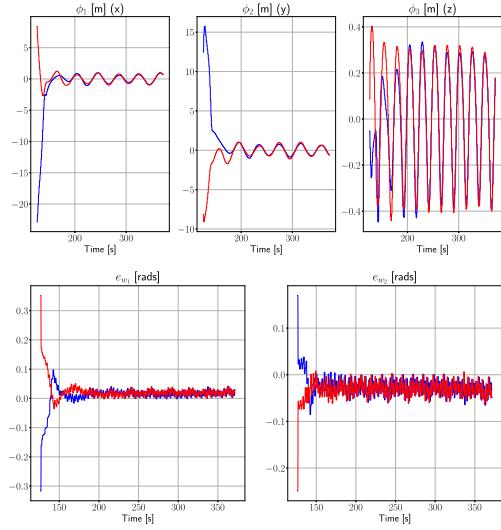


Fig. 11. Path-following errors (in  $XYZ$ ) of the two aircraft (blue and red) to the desired torus surface, and errors  $e_{w1}, e_{w2}$  with respect to the desired  $\Delta_1^{[1,2]} = \Delta_2^{[1,2]} = 0$ . The horizontal axis represents time in seconds.

a reference to ascend/descend. A traditional trajectory tracking algorithm would force the aircraft to track an *open-loop* (i.e., state independent) point; hence, it requires controlling the airspeed/ground speed of the aircraft. Such a requirement is demanding if the aircraft are not equipped with the required sensors or actuators (e.g., spoilers/flaps), and the wind always affects the speed of the airplane. In contrast, our guidance

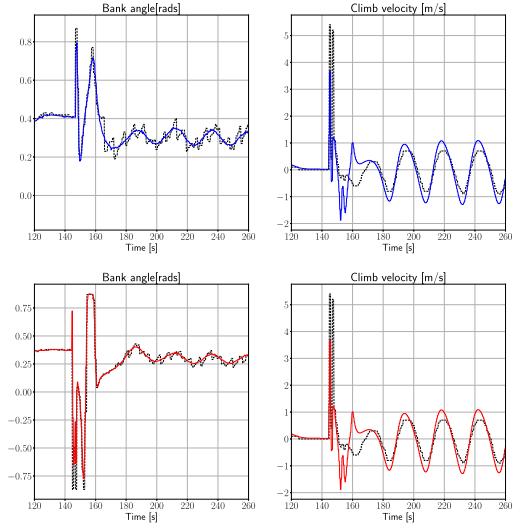


Fig. 12. Black curves are the desired values for the bank angles and the vertical velocities produced by the guiding vector field algorithm for the flight in Fig. 11. The blue and red curves represent the actual states of the first and second aircraft, respectively. For more information on the employed low-level controllers, see [https://wiki.paparazziuav.org/wiki/Control\\_Loops](https://wiki.paparazziuav.org/wiki/Control_Loops).

algorithm is free from such a requirement since the parameter  $w$  is in *closed-loop* with the aircraft state, and thereby adapting automatically to the positions and velocities of the aircraft.

## VIII. CONCLUSION AND FUTURE WORK

In this article, we address the problem of multirobot coordinated navigation using a novel guiding vector field. The proposed approach enables an arbitrary number of robots to follow or navigate possibly different desired paths or surfaces and achieve motion coordination in a distributed way. Specifically, we derive a time-invariant coordinating guiding vector field to rigorously guarantee the convergence and motion coordination on desired paths or surfaces. This is achieved by exploiting the higher dimensional guiding vector field with path or surface parameters as virtual coordinates and employing distributed consensus algorithms to render the virtual coordinate differences (i.e., parametric differences) converging to the predefined ones. Based on the coordinating guiding vector field, a control law is designed for a nonholonomic Dubins-car-like robot model, which considers the actuation saturation. Moreover, we also elaborate on how to effectively integrate the coordinating guiding vector field with a safety barrier certificate to realize collision avoidance among robots. Extensive simulation examples verify the effectiveness of our approach and showcase possible practical applications. Furthermore, we conduct outdoor experiments with multiple fixed-wing aircraft to demonstrate our algorithm's practical value, including its robustness against wind perturbation, actuation saturation, etc. There are many other fascinating features of our approach as outlined at the end of Section I-B, such as the scalability and the low computational cost. Note that the (communication) graph topology is assumed to be fixed in this work, but our algorithm is still effective under time-varying or switching graph topologies if additional

assumptions/conditions are satisfied (e.g., [47, Th. 9], [48, Th. 2.31, Corollary 2.32]).

Our work opens multiple directions for future studies. 1) Since the motion coordination is quantified in terms of path or surface parameters, it is reasonable that the robots might not keep equal or manually specified Euclidean distances if the parametric equations are highly nonlinear (e.g., trigonometric equations). Therefore, one may investigate a transformation method between the parameter space and the Euclidean space such that the motion coordination can be quantified in terms of Euclidean distances. 2) One may study time-varying coordinating guiding vector fields to track moving desired paths or time-varying desired parametric differences  $\Delta^{[i,j]}(t)$ . 3) The deadlock problem may arise in generating a new vector field incorporating collision avoidance in Section V. This problem has been partially addressed in [40, Sec. VII], while the full solution is worth further investigation.

## REFERENCES

- [1] A. Prorok, M. Malencia, L. Carlone, G. S. Sukhatme, B. M. Sadler, and V. Kumar, "Beyond robustness: A taxonomy of approaches towards resilient multi-robot systems," 2021.
- [2] G.-Z. Yang et al., "The grand challenges of science robotics," *Sci. Robot.*, vol. 3, no. 14, 2018, Art. no. eaar 7650.
- [3] P. B. Sujit, S. Saripalli, and J. B. Sousa, "Unmanned aerial vehicle path following: A survey and analysis of algorithms for fixed-wing unmanned aerial vehicles," *IEEE Control Syst.*, vol. 34, no. 1, pp. 42–59, Feb. 2014.
- [4] A. Ratnoo, P. Sujit, and M. Kothari, "Adaptive optimal path following for high wind flights," *IFAC Proc. Vol.*, vol. 44, no. 1, pp. 12 985–12 990, 2011.
- [5] S. Park, J. Deyst, and J. P. How, "Performance and Lyapunov stability of a nonlinear path following guidance method," *J. Guidance, Control, Dyn.*, vol. 30, no. 6, pp. 1718–1728, 2007.
- [6] W. Caharija, K. Y. Pettersen, P. Calado, and J. Braga, "A comparison between the ILOS guidance and the vector field guidance," *IFAC-PapersOnLine*, vol. 48, no. 16, pp. 89–94, 2015.
- [7] Y. A. Kapitanyuk, A. V. Proskurnikov, and M. Cao, "A guiding vector-field algorithm for path-following control of nonholonomic mobile robots," *IEEE Trans. Control Syst. Technol.*, vol. 26, no. 4, pp. 1372–1385, Jul. 2018.
- [8] H. G. De Marina, Z. Sun, M. Bronz, and G. Hattenberger, "Circular formation control of fixed-wing UAVs with constant speeds," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 5298–5303.
- [9] K. Nakai and K. Uchiyama, "Vector fields for UAV guidance using potential function method for formation flight," in *Proc. AIAA Guidance, Navig., Control Conf.*, 2013, Paper 4626.
- [10] L. C. Pimenta, G. A. Pereira, M. M. Gonçalves, N. Michael, M. Turpin, and V. Kumar, "Decentralized controllers for perimeter surveillance with teams of aerial robots," *Adv. Robot.*, vol. 27, no. 9, pp. 697–709, 2013.
- [11] X. Wang, S. Baldi, X. Feng, C. Wu, H. Xie, and B. De Schutter, "A fixed-wing UAV formation algorithm based on vector field guidance," *IEEE Trans. Automat. Sci. Eng.*, early access, doi: [10.1109/TASE.2022.3144672](https://doi.org/10.1109/TASE.2022.3144672).
- [12] J. Ghommam, H. Mehrjerdi, M. Saad, and F. Mnif, "Formation path following control of unicycle-type mobile robots," *Robot. Auton. Syst.*, vol. 58, no. 5, pp. 727–736, 2010.
- [13] 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, 2007.
- [14] L. Sabattini, C. Secchi, M. Cocetti, A. Levрatti, and C. Fantuzzi, "Implementation of coordinated complex dynamic behaviors in multi-robot systems," *IEEE Trans. Robot.*, vol. 31, no. 4, pp. 1018–1032, Aug. 2015.
- [15] H. Chen, Y. Cong, X. Wang, X. Xu, and L. Shen, "Coordinated path-following control of fixed-wing unmanned aerial vehicles," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 52, no. 4, pp. 2540–2554, Apr. 2022.
- [16] Y. Wang, D. Wang, and S. Zhu, "Cooperative moving path following for multiple fixed-wing unmanned aerial vehicles with speed constraints," *Automatica J. IFAC*, vol. 100, pp. 82–89, 2019.
- [17] A. Doosthoseini and C. Nielsen, "Coordinated path following for unicycles: A nested invariant sets approach," *Automatica J. IFAC*, vol. 60, pp. 17–29, 2015.
- [18] L. A. V. Reyes and H. G. Tanner, "Flocking, formation control, and path following for a group of mobile robots," *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 4, pp. 1268–1282, Jul. 2015.
- [19] A. H. Mong-ying and V. Kumar, "Pattern generation with multiple robots," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2006, pp. 2442–2447.
- [20] A. H. Mong-ying, S. Loizou, and V. Kumar, "Stabilization of multiple robots on stable orbits via local sensing," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2007, pp. 2312–2317.
- [21] M. Iqbal, S.-i. Azuma, J. Leth, and T. D. Ngo, "Circular and concentric formation of kinematic unicycles," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2019, pp. 1–6.
- [22] Z. Sun, H. G. de Marina, G. S. Seyboth, B. D. O. Anderson, and C. Yu, "Circular formation control of multiple unicycle-type agents with nonidentical constant speeds," *IEEE Trans. Control Syst. Technol.*, vol. 27, no. 1, pp. 192–205, Jan. 2019.
- [23] C. Wang, G. Xie, and M. Cao, "Forming circle formations of anonymous mobile agents with order preservation," *IEEE Trans. Autom. Control*, vol. 58, no. 12, pp. 3248–3254, Dec. 2013.
- [24] H. Lu, W. Yao, and L. Chen, "Distributed multi-robot circumnavigation with dynamic spacing and time delay," *J. Intell. Robotic Syst.*, vol. 99, no. 1, pp. 165–182, 2020.
- [25] W. Yao, H. Lu, Z. Zeng, J. Xiao, and Z. Zheng, "Distributed static and dynamic circumnavigation control with arbitrary spacings for a heterogeneous multi-robot system," *J. Intell. Robotic Syst.*, vol. 94, no. 3/4, pp. 883–905, 2019.
- [26] A. Franchi, P. Stegagno, and G. Oriolo, "Decentralized multi-robot encirclement of a 3D target with guaranteed collision avoidance," *Auton. Robots*, vol. 40, no. 2, pp. 245–265, 2016.
- [27] W. Yao, H. G. de Marina, B. Lin, and M. Cao, "Singularity-free guiding vector field for robot navigation," *IEEE Trans. Robot.*, vol. 37, no. 4, pp. 1206–1221, Aug. 2021.
- [28] W. Yao, H. G. de Marina, Z. Sun, and M. Cao, "Distributed coordinated path following using guiding vector fields," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 10 030–10 037.
- [29] M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks*, vol. 33. Princeton, NJ, USA: Princeton University Press, 2010.
- [30] W. Yao, H. G. de Marina, and M. Cao, "Vector field guided path following control: Singularity elimination and global convergence," in *Proc. IEEE Conf. Decis. Control*, 2020, pp. 1543–1549.
- [31] W. Yao, "Guiding vector fields for robot motion control," Ph.D. dissertation, Fac. Sci. Eng., Univ. Groningen, Groningen, The Netherlands, Oct. 2021.
- [32] W. Yao, Y. A. Kapitanyuk, and M. Cao, "Robotic path following in 3D using a guiding vector field," in *Proc. IEEE Conf. Decis. Control*, 2018, pp. 4475–4480.
- [33] V. M. Goncalves, L. C. A. Pimenta, C. A. Maia, B. C. O. Dutra, and G. A. S. Pereira, "Vector fields for robot navigation along time-varying curves in n-dimensions," *IEEE Trans. Robot.*, vol. 26, no. 4, pp. 647–659, Aug. 2010.
- [34] A. Galbis and M. Maestre, *Vector Analysis Versus Vector Calculus*. New York, NY, USA: Springer, 2012.
- [35] A. M. C. Rezende, V. M. Goncalves, and L. C. A. Pimenta, "Constructive time-varying vector fields for robot navigation," *IEEE Trans. Robot.*, vol. 38, no. 2, pp. 852–867, Apr. 2022.
- [36] D. R. Nelson, D. B. Barber, T. W. McLain, and R. W. Beard, "Vector field path following for miniature air vehicles," *IEEE Trans. Robot.*, vol. 23, no. 3, pp. 519–529, Jun. 2007.
- [37] H. Khalil, *Nonlinear Systems*, 3rd ed. Hoboken, NJ, USA: Prentice-Hall, 2002.
- [38] W. Ren and R. W. Beard, *Distributed Consensus in Multi-Vehicle Cooperative Control*, vol. 27. London, U.K.: Springer, 2008.
- [39] W. Yao, H. G. de Marina, Z. Sun, and M. Cao, "Guiding vector fields for the distributed motion coordination of mobile robots," 2022, arXiv:2209.09478. [Online]. Available: <https://arxiv.org/abs/2209.09478>
- [40] L. Wang, A. D. Ames, and M. Egerstedt, "Safety barrier certificates for collisions-free multirobot systems," *IEEE Trans. Robot.*, vol. 33, no. 3, pp. 661–674, Jun. 2017.
- [41] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Trans. Autom. Control*, vol. 62, no. 8, pp. 3861–3876, Aug. 2017.

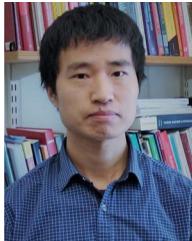
- [42] X. Xu, P. Tabuada, J. W. Grizzle, and A. D. Ames, "Robustness of control barrier functions for safety critical control," *IFAC-PapersOnLine*, vol. 48, no. 27, pp. 54–61, 2015.
- [43] X. Yun and Y. Yamamoto, "On feedback linearization of mobile robots," Dept. Comput. Inf. Sci., Univ. Pennsylvania, Philadelphia, PA, USA, Tech. Rep. MS-CIS-92-45, 1992.
- [44] B. Siciliano and O. Khatib, *Springer Handbook of Robotics*. Secaucus, NJ, USA: Springer-Verlag, 2007.
- [45] A. M. Rezende, V. M. Gonçalves, G. V. Raffo, and L. C. Pimenta, "Robust fixed-wing UAV guidance with circulating artificial vector fields," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 5892–5899.
- [46] B. Gati, "Open source autopilot for academic research—the paparazzi system," in *Proc. IEEE Amer. Control Conf.*, 2013, pp. 1478–1481.
- [47] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 1520–1533, Sep. 2004.
- [48] W. Ren and R. W. Beard, *Distrib. Consensus in Multi-vehicle Cooperative Control* (Communications and Control Eng. Series), E. D. Sontag, M. Thoma, A. Isidori, and J. H. van Schuppen, Eds., London, U.K.: Springer, 2008.



**Weijia Yao** (Member, IEEE) received the Ph.D. degree with distinction *cum laude* in systems and control theory from the University of Groningen, Groningen, The Netherlands, in 2021.

His research interests include nonlinear systems and control, robotics, multiagent systems, and game theory.

Dr. Yao was a finalist for the Best Conference Paper Award at ICRA in 2021, a finalist for the Georges Giralt Ph.D. Award in 2022, and the recipient of the Outstanding Master Degree Dissertation Award of Hunan province, China, in 2020.



**Zhiyong Sun** received the Ph.D. degree in control engineering from The Australian National University (ANU), Canberra, ACT, Australia, in February 2017.

He was a Research Fellow/Lecturer with the Research School of Engineering, ANU, from 2017 to 2018. From June 2018 to January 2020, he was a Postdoctoral Researcher with the Department of Automatic Control, Lund University, Lund, Sweden. Since January 2020, he has been with the Eindhoven University of Technology, Eindhoven, The Netherlands, as an Assistant Professor. His research interests include multiagent systems, autonomous robotics, and distributed control and optimization.



**Ming Cao** (Fellow, IEEE) received the bachelor's and master's degrees from Tsinghua University, Beijing, China, in 1999 and 2002, respectively, and the Ph.D. degree from Yale University, New Haven, CT, USA, in 2007, all in electrical engineering.

Since 2016, he has been a Professor of networks and robotics with the Engineering and Technology Institute, University of Groningen, Groningen, The Netherlands, where he started as an Assistant Professor in 2008. From 2007 to 2008, he was a Research Associate with Princeton University, Princeton, NJ, USA. He was a Research Intern in 2006 with the IBM T. J. Watson Research Center, USA. His research interests include autonomous robots and multiagent systems, complex networks, and decision-making processes.

Dr. Cao was the 2017 and inaugural recipient of the Manfred Thoma Medal from the International Federation of Automatic Control and the 2016 recipient of the European Control Award sponsored by the European Control Association. He is a Senior Editor for *Systems and Control Letters*, an Associate Editor for *IEEE TRANSACTIONS ON AUTOMATIC CONTROL*, *IEEE TRANSACTIONS OF CONTROL ON NETWORK SYSTEMS*, and *IEEE Robotics and Automation Magazine*, and was an Associate Editor for *IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS* and *IEEE Circuits and Systems Magazine*. He is a member of the IFAC Conference Board and a Vice Chair of the IFAC Technical Committee on Large-Scale Complex Systems.



**Héctor García de Marina** received the Ph.D. degree in systems and control from the University of Groningen, The Netherlands, in 2016.

He was a Postdoctoral Research Associate with the Ecole Nationale de l'aviation Civile, Toulouse, France, and an Assistant Professor with the Unmanned Aerial Systems Center, University of Southern Denmark, Odense, Denmark. Since 2022, he has been a Ramón y Cajal Researcher with the Department of Computer Engineering, Automation and Robotics, and with CITIC, Universidad de Granada, Spain. He is an Associate Editor for *IEEE TRANSACTIONS ON ROBOTICS*. His current research interests include multiagent systems and the design of guidance navigation and control systems for autonomous vehicles.