

University of Groningen

## Singularity-free guiding vector field for robot navigation

Yao, Weijia; Garcia de Marina, Hector; Lin, Bohuan; Cao, Ming

*Published in:*  
IEEE Transaction on Robotics

*DOI:*  
[10.1109/TRO.2020.3043690](https://doi.org/10.1109/TRO.2020.3043690)

**IMPORTANT NOTE:** You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.

*Document Version*  
Publisher's PDF, also known as Version of record

*Publication date:*  
2021

[Link to publication in University of Groningen/UMCG research database](#)

*Citation for published version (APA):*

Yao, W., Garcia de Marina, H., Lin, B., & Cao, M. (2021). Singularity-free guiding vector field for robot navigation. *IEEE Transaction on Robotics*, 37(4), 1206-1221. <https://doi.org/10.1109/TRO.2020.3043690>

**Copyright**

Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).

The publication may also be distributed here under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license. More information can be found on the University of Groningen website: <https://www.rug.nl/library/open-access/self-archiving-pure/taverne-amendment>.

**Take-down policy**

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

*Downloaded from the University of Groningen/UMCG research database (Pure): <http://www.rug.nl/research/portal>. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.*

# Singularity-Free Guiding Vector Field for Robot Navigation

Weijia Yao , Héctor García de Marina , Bohuan Lin , and Ming Cao 

**Abstract**—In robot navigation tasks, such as unmanned aerial vehicle (UAV) highway traffic monitoring, it is important for a mobile robot to follow a specified desired path. However, most of the existing path-following navigation algorithms cannot guarantee global convergence to desired paths or enable following self-intersected desired paths due to the existence of singular points where navigation algorithms return unreliable or even no solutions. One typical example arises in *vector-field guided path-following (VF-PF) navigation algorithms*. These algorithms are based on a vector field, and the singular points are exactly where the vector field diminishes. Conventional VF-PF algorithms generate a vector field of the same dimensions as those of the space where the desired path lives. In this article, we show that it is mathematically *impossible* for conventional VF-PF algorithms to achieve global convergence to desired paths that are *self-intersected* or even just *simple closed* (precisely, homeomorphic to the unit circle). Motivated by this new impossibility result, we propose a novel method to transform self-intersected or simple closed desired paths to *nonself-intersected* and *unbounded* (precisely, homeomorphic to the real line) counterparts in a *higher dimensional* space. Corresponding to this new desired path, we construct a singularity-free guiding vector field on a higher dimensional space. The integral curves of this new guiding vector field is thus exploited to enable global convergence to the higher dimensional desired path, and therefore the projection of the integral curves on a lower dimensional subspace converge to the physical (lower dimensional) desired path. Rigorous theoretical analysis is carried out for the theoretical results using dynamical systems theory. In addition, we show both by theoretical analysis and numerical simulations that our proposed method is an extension combining conventional VF-PF algorithms and trajectory tracking

algorithms. Finally, to show the practical value of our proposed approach for complex engineering systems, we conduct outdoor experiments with a fixed-wing airplane in windy environment to follow both 2-D and 3-D desired paths.

**Index Terms**—Guidance, navigation, path following, robot motion, singularity, vector field.

## I. INTRODUCTION

SEVERAL robot navigation tasks, such as highway traffic monitoring, underwater pipeline inspection, border patrolling, and area coverage, require the fundamental functionality of following a desired path [1], and new applications are emerging, such as using drones to probe atmospheric phenomena along prescribed paths [2]. The path-following navigation problem has attracted the attention from both the robotics community [3]–[7] and the control community [8]–[13]. In a path-following navigation algorithm, the desired path is usually given in the form of a single connected curve *without temporal information*, and then robots are guided to converge to and move along it with sufficient accuracy. Treating the desired path as a *geometric object* rather than a time-dependent point, path-following navigation algorithms sometimes are able to overcome a number of performance limitations rooted in trajectory tracking [14], e.g., inaccuracy due to unstable zero dynamics [15] and difficulty to maintain constant tracking speed [16]. Moreover, comparing to trajectory tracking algorithms, there is separate interest for the study of path-following navigation algorithms since they are more suitable for some applications, such as fixed-wing aircraft guidance and control [4], [17], [18].

Among different path-following navigation algorithms, those using a *guiding vector field* have been studied widely [3]–[5], [8]–[10], and we refer to these algorithms as *vector-field guided path-following navigation algorithms* (or, *VF-PF algorithms*). The *guidance* feature of the vector field is justified as follows: usually robot *kinematics models* (e.g., single-integrator and double-integrator models [3], [17]) are considered, and the guiding vector field, as its name suggests, provides *guidance* signal inputs to the model. This is valid based on the common assumption that the *robot-specific* inner-loop *dynamics control* can track these guidance signal inputs effectively [16], [19], [20]. Thus, one can simply focus on the guidance layer (i.e., designing a guiding vector field), while considering other control layers separately. Specifically, the guiding vector field is designed such that its integral curves are guaranteed to converge to a predefined geometric desired path. Utilizing the convergence property of the vector field, one can then derive suitable control laws. It is

Manuscript received July 28, 2020; accepted November 5, 2020. Date of publication January 1, 2021; date of current version August 5, 2021. The work of Weijia Yao and Bohuan Lin was supported by the China Scholarship Council. The work of Héctor García de Marina was supported in part by Atracción de Talento under Grant 2019-T2/TIC-13503 from the Government of the Autonomous Community of Madrid, and in part by the Spanish Ministry of Science and Innovation under research Grant RTI2018-098962-B-C21. The work of Ming Cao was supported in part by the European Research Council under Grant ERC-CoG-771687 and in part by the Netherlands Organization for Scientific Research under Grant NWO-vidi-14134. This article was recommended for publication by Associate Editor M. Schwager and Editor P. Robuffo Giordano upon evaluation of the reviewers' comments. (Corresponding author: Weijia Yao.)

Weijia Yao and Ming Cao are with the Institute of Engineering and Technology, University of Groningen, 9747 AG Groningen, the Netherlands (e-mail: weijia.yao.new@gmail.com; m.cao@rug.nl).

Héctor García de Marina is with the Department of Computer Architecture and Automatic Control, Faculty of Physics, Universidad Complutense de Madrid, 28040 Madrid, Spain (e-mail: hgarcia@ucm.es).

Bohuan Lin is with the Bernoulli Institute of Mathematics, Computer Science and Artificial Intelligence, University of Groningen, 9747 AG Groningen, the Netherlands (e-mail: figaro312@sina.com).

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.2020.3043690>.

Digital Object Identifier 10.1109/TRO.2020.3043690

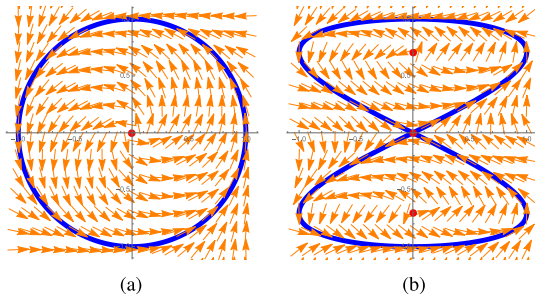


Fig. 1. Normalized vector fields [16] for a circle path described by  $\phi(x, y) = x^2 + y^2 - 1 = 0$  in (a) and a figure “8” path described by  $\phi(x, y) = x^2 - 4y^2(1 - y^2) = 0$  in (b). The red points are the singular points of the (unnormalized) vector fields.

claimed in [18] that VF-PF algorithms demonstrate the lowest cross-track error while requiring the least control effort among several other path-following navigation algorithms. In addition, [21] shows that VF-PF algorithms achieve better path-following accuracy than the integral line-of-sight (ILOS) method [12].

Although the VF-PF algorithms are intuitive and easy to implement, the rigorous analysis remains nontrivial for general desired paths [13], [16], [22], [23]. Significant difficulty in the analysis and application of the VF-PF algorithms arises when there are singular points<sup>1</sup> in the vector field [see Fig. 1(a) and (b)]. In such a case, the convergence of trajectories to the desired path cannot be guaranteed globally, and the normalization of the vector field at those points is not well-defined [3], [13], [16], [22]. In [3], it is assumed that these singular points are repulsive to simplify the analysis, while this assumption is dropped in [16] for a planar desired path and in [13] and [22], for a desired path in 3-D. However, to the best of our knowledge, few efforts have been made on dealing with singular points directly or on eliminating them effectively. Recently, [4] presents a simple treatment of the singular point—the robot does not change its course inside a ball centered at the singular point. Under some conditions, the Lyapunov function evaluated at the exit point is proved to be smaller than that at the entry point. However, these conditions are conservative, since they assume repulsiveness of the singular points, which is not the case, e.g., for saddle points.

Related to the existence of singular points, one of the challenges for the VF-PF navigation problem is to follow a self-intersected desired path. Many existing VF-PF algorithms (e.g., [3], [13], [16], [17], [25]) fail to fulfil this task. This is rooted in the fact that the vector field degenerates to zero at the crossing points of a self-intersected desired path, leading to a zero guidance signal, and thus, a robot can get stuck on the desired path [see Fig. 1(b)]. Due to the existence of singular points on the desired path, some effective VF-PF algorithms such as [13], [16], and [22], become invalid simply because the assumptions are violated in this case. In fact, this task is also challenging for other existing path-following methods, since in the vicinity of the crossing points, many methods are “ill-defined.” For example, the line-of-sight method [26] is not applicable as there is not a

unique projection point in the vicinity of the intersection of the desired path. Indeed, many existing path-following navigation algorithms either focus on simple desired paths such as circles or straight lines [18], [25], [27], or only deal with desired paths that are sufficiently smooth [3], [13], [16], [22]. One might retreat to the virtual-target path-following navigation algorithm [28]. In this method, a virtual target has its own dynamics traveling on the desired path; thus the path-following navigation problem is implicitly converted to a target tracking problem. Although through this conversion, it is possible for a robot to follow a self-intersected desired path, this method is inherently a tracking approach, and thus, may inherit the performance limitations mentioned before, such as limited path-following accuracy.

Another challenging task arising from the VF-PF methods is the description of the desired path, which is crucial for the derivation of the vector field. For generality, the desired path is usually determined by the intersection of several (hyper)surfaces represented by the zero-level sets of some implicit functions [3], [4], [13], [16], [22], [29]. For planar desired paths, for example, the implicit function of a star curve might be as complicated as that in [7], while for desired paths in a higher dimensional space, it is counter-intuitive to create (hyper)surfaces such that the intersection is precisely the desired path, such as a helix. On the other hand, many geometric curves are described by parametric equations [30] rather than implicit functions. It is possible to transform the parametric equations to implicit functions and then derive the vector field, but this might not always be feasible and is computationally expensive. The restrictive characterization of the desired path limits the applicability of VF-PF algorithms to some extent.

In this article, we improve the VF-PF methodology in the sense that we address the aforementioned three issues—the existence of singular points, the obstacle of dealing with self-intersected paths, and the difficulty of representing a generic desired path. Specifically, based on the design of guiding vector fields in [22], we use an intuitive idea to eliminate singular points of the vector field so that global convergence to the desired path, even if self-intersected, is guaranteed. The general idea is to extend the dimensions of the vector field and eliminate singular points simultaneously. This procedure naturally leads to a simple way to transform the descriptions of desired paths from parameterized forms to the intersection of several (hyper)surfaces, which are required in creating a guiding vector field.

It is important to clarify the terminology used throughout this article. In many VF-PF algorithms, the desired path is a geometric object which is not necessarily parameterized. In a precise mathematical language, we assume that the desired path is a 1-D connected differential manifold [13]. Therefore, we can generally classify desired paths into two categories—those homeomorphic to the unit circle  $\mathbb{S}^1$  if they are compact, and those homeomorphic to the real line  $\mathbb{R}$  otherwise [31, Th. 5.27]. This assumption is not a restriction, since many desired paths in practice, such as a circle, an ellipse, a Cassini oval, a spline and a straight line, satisfy this assumption. For ease of exposition, we refer to those desired paths homeomorphic to the unit circle as *simple closed* desired paths, and those homeomorphic to the real line as *unbounded* desired paths. Note that *self-intersected* paths

<sup>1</sup>A point where a vector field becomes zero is called a *singular point* of the vector field [24, p. 219].

do not satisfy this assumption. Nevertheless, we will introduce in the sequel how to *transform* a self-intersected physical desired path to a non-self-intersected and unbounded higher dimensional desired path such that the assumption holds to apply our algorithm.

Here, we summarize the major contributions of our article.

First, we show by rigorous topological analysis that guiding vector fields with the same dimension as the desired path (e.g., [5], [7], [16], [22]) cannot guarantee the global convergence to a simple closed or self-intersected desired path (see Theorem 1 in Section III). With the dichotomy of convergence discussed in the article, this implies that singular points of the vector field *always exist* for a *simple closed* or *self-intersected* desired path regardless of which hypersurfaces one uses to characterize the desired path. This explains why many vector-field guided path-following navigation algorithms in the literature cannot guarantee global convergence in the Euclidean space to a simple closed desired path. We note that excluding singular points is important in practice (e.g., for fixed-wing aircraft guidance and navigation) since degenerated or pathological solutions of system dynamics can be safely avoided. Therefore, this topological obstacle is the primary motivation of the subsequent theoretical development including the introduction of extended dynamics (see Section IV) and the creation of singularity-free guiding vector fields (see Section V).

Secondly, due to the aforementioned topological obstruction, we improve the existing VF-PF algorithms such that all singular points are removed, and global convergence of trajectories to the desired path is rigorously guaranteed (see Sections IV and V). We overcome this topological obstruction by changing the topology of the desired paths. Specifically, we transform a physical *simple closed* or *self-intersected* desired path to a new *unbounded* and *nonself-intersected* desired path in a *higher dimensional space*. We then derive the corresponding guiding vector field on this higher dimensional space, which is guaranteed to have *no* singular points.

Thirdly, our proposed method to create this new singularity-free guiding vector field is proved to enjoy several appealing features (see Section V-B). For example, we provide theoretical guarantees for global exponential convergence of trajectories of system dynamics to the desired path. In addition, the new system dynamics with the singularity-free guiding vector field is robust against perturbation, such as noisy position measurements (see Feature 3 in Section V-B). Moreover, using our proposed method, it becomes straightforward to represent (hyper)surfaces of which the intersection is the new higher dimensional desired path, as long as a parametrization of the physical (lower dimensional) desired path is available (see Feature 2 in Section V-B).

Last but not least, we successfully conduct experiments using a fixed-wing aircraft to verify the effectiveness of our proposed VF-PF algorithm in 3-D (see Section VI), in addition to the experiment with an e-puck robot [32] in our previous preliminary work [33]. This verifies the practical significance of our proposed method for highly complex autonomous vehicles. We also discuss and conclude that our proposed VF-PF algorithm combines and extends features of the conventional VF-PF algorithms and trajectory tracking algorithms (see Section VII). While we

do not claim that our proposed new singularity-free guiding vector field is always superior than traditional trajectory tracking algorithms in every application scenario (such as quadcopter attitude tracking), we emphasize that it significantly improves conventional VF-PF algorithms by providing a global solution and enabling the path-following behavior of complicated or unconventional desired paths (e.g., a self-intersected Lissajous curve). This is imperative and irreplaceable in applications such as fixed-wing aircraft guidance and navigation where convergence to and propagation along a desired path from *every* initial position is required.

This article is a significant extension of our preliminary version in [33] in the following aspects: 1) [33] only considers physical planar desired paths and the *higher dimensional* 3-D vector field, but we generalize all the results to *n*-dimensional; 2) [33] only investigates a *linear transformation operator*, but we introduce a more general *transformation operator* which can be linear or nonlinear in this article. Compared with [33], we present the following five new results.

- 1) We prove a topological theorem regarding the feasibility of global convergence to desired paths, which serves as the primary motivation for the subsequent development.
- 2) We provide rigorous guarantees to justify the features of our proposed approach.
- 3) We provide experiments with a fixed-wing aircraft in 3-D to verify the practicality of the proposed approach for complex engineering systems.
- 4) We elaborate on some important aspects in the implementation of the proposed approach.
- 5) We provide new insightful discussion of our proposed approach and existing algorithms.

Although our previous work in [13] introduces part of the theoretical foundation for this article, it does not solve the singularity problem that will be addressed here.

The remainder of this article is organized as follows. Section II introduces conventional guiding vector fields for path following. In Section III, a theorem about the impossibility of global convergence to simple closed or self-intersected desired paths using the conventional VF-PF navigation algorithm is elaborated. This is the main motivation for the design of higher dimensional guiding vector fields, which will be utilized in Section IV through *extended dynamics*. Based on the previous sections, the construction approach of *singularity-free* guiding vector fields is presented in Section V. In addition, several appealing features of this method are highlighted in this section. Then experiments with a fixed-wing aircraft are conducted to validate the theoretical results in Section VI. Following this, Section VII discusses how our proposed approach can be viewed as a combined extension of VF-PF algorithms and trajectory tracking algorithms. Finally, Section VIII concludes the article and indicates future work. Due to the page limit, some theoretical proofs are provided in the Appendix in the full version [34]. We present here some notations and basic concepts that are used throughout the article.

*Notations:* The notation  $\|\cdot\|$  denotes the Euclidean norm of a vector or the induced matrix two-norm of a matrix, and  $:=$  means “defined to be.” Given a positive integer  $n$ , the *distance*



between a point  $p_0 \in \mathbb{R}^n$  and a set  $\mathcal{S} \subseteq \mathbb{R}^n$  is denoted by  $\text{dist}(p_0, \mathcal{S}) = \text{dist}(\mathcal{S}, p_0) := \inf\{\|p - p_0\| : p \in \mathcal{S}\}$ . The distance between two sets  $\mathcal{A}$  and  $\mathcal{B}$  is  $\text{dist}(\mathcal{A}, \mathcal{B}) = \text{dist}(\mathcal{B}, \mathcal{A}) := \inf\{\|a - b\| : a \in \mathcal{A}, b \in \mathcal{B}\}$ . If  $\xi$  is a differentiable function of time  $t$ , then the derivative of  $\xi$  with respect to time is  $\dot{\xi}$ . The normalization of a vector  $v$  is denoted by  $\hat{v}$  (i.e.,  $\hat{v} = v/\|v\|$ ). The transpose of a vector  $v$  is denoted by  $v^\top$ . Suppose there is a function  $\rho : \mathcal{C} \rightarrow \mathcal{D}$ , where the sets  $\mathcal{C}$  and  $\mathcal{D}$  are nonempty. The image of a subset  $\mathcal{F} \subseteq \mathcal{C}$  under  $\rho$  is the subset  $\rho(\mathcal{F}) \subseteq \mathcal{D}$  defined by  $\rho(\mathcal{F}) := \{\rho(x) \in \mathcal{D} : x \in \mathcal{F}\}$ . Given two functions  $f : X \rightarrow Y, g : Y \rightarrow Z$ ,  $g \circ f$  denotes the composition of these two functions.

**Basic concepts:** A trajectory  $\xi : [0, +\infty) \rightarrow \mathbb{R}^n$  asymptotically converges to a nonempty set  $\mathcal{B} \subseteq \mathbb{R}^n$  if for any  $\epsilon > 0$ , there exists  $T > 0$  such that  $\text{dist}(\xi(t), \mathcal{B}) < \epsilon$  for all  $t > T$ . If the trajectory can only be maximally prolonged to  $t_* < \infty$  [35, Ch. 3], then we say that it converges to the set  $\mathcal{B}$  as  $t$  approaches  $t_*$ , if for any  $\epsilon > 0$ , there exists  $\delta > 0$  such that  $\text{dist}(\xi(t), \mathcal{B}) < \epsilon$  for  $|t - t_*| < \delta$ . Two subsets  $U \subseteq \mathbb{R}^m$  and  $V \subseteq \mathbb{R}^k$  are *homeomorphic*, denoted by  $U \approx V$ , if there exists a continuous bijection  $\varphi : U \rightarrow V$  of which the inverse is also continuous. The map  $\varphi$  is called a *homeomorphism* [31].

## II. GUIDING VECTOR FIELDS FOR PATH FOLLOWING

In this section, we introduce the vector-field guided path-following (VF-PF) navigation problem and guiding vector fields. We formally define the VF-PF navigation problem as follows.

**Problem 1 (VF-PF navigation problem):** Given a desired path  $\mathcal{P} \subseteq \mathbb{R}^n$ , the VF-PF navigation problem is to design a continuously differentiable vector field  $\chi : \mathbb{R}^n \rightarrow \mathbb{R}^n$  for the differential equation  $\dot{\xi}(t) = \chi(\xi(t))$  such that the two conditions as follows are satisfied.

- 1) There exists a neighborhood  $\mathcal{D} \subseteq \mathbb{R}^n$  of the desired path  $\mathcal{P}$  in (1) such that for all initial conditions  $\xi(0) \in \mathcal{D}$ , the distance  $\text{dist}(\xi(t), \mathcal{P})$  between the trajectory  $\xi(t)$  and the desired path  $\mathcal{P}$  approaches zero as time  $t \rightarrow \infty$ ; that is,  $\lim_{t \rightarrow \infty} \text{dist}(\xi(t), \mathcal{P}) = 0$ .
- 2) If a trajectory starts from the desired path, then the trajectory stays on the path for  $t \geq 0$  (i.e.,  $\xi(0) \in \mathcal{P} \Rightarrow \xi(t) \in \mathcal{P}$  for all  $t \geq 0$ ). In addition, the vector field on the desired path is nonzero (i.e.,  $0 \notin \chi(\mathcal{P})$ ).

In this article, we only investigate the guiding vector field in the Euclidean space  $\mathbb{R}^n$ , but it can be extended to a general smooth manifold [36]. For easy understanding of the guiding vector field on  $\mathbb{R}^n$  in the sequel, we first introduce the one defined on the 2-D Euclidean space  $\mathbb{R}^2$ .

### A. Preliminaries on 2-D VF-PF Control

In [16], the desired path  $\mathcal{P}$  is described by the zero-level set of an implicit function

$$\mathcal{P} = \{(x, y) \in \mathbb{R}^2 : \phi(x, y) = 0\} \quad (1)$$

where  $\phi : \mathbb{R}^2 \rightarrow \mathbb{R}$  is twice continuously differentiable. In this description,  $\mathcal{P}$  is a subset of  $\mathbb{R}^2$ . The description is different from some other works where the desired path is a parameterized differentiable curve (e.g., [14]); that is, a differentiable map

$f : I \rightarrow \mathbb{R}^n$  of an open interval  $I = (a, b)$  of the real line  $\mathbb{R}$  into  $\mathbb{R}^n$  [30]. From the definition, we observe that the mathematical object in (1) is actually the *trace* of a parameterized curve  $f$  [30], or the image of a mapping  $f$ . However, this description of the desired path without any parametrization is common in the field of VF-PF navigation [3]–[5], [13], [22], [37]–[40]. One of the advantages is that the vector field can be derived directly from the function  $\phi(\cdot)$ , independent of the specific parametrization of the desired path. Another advantage is that we can replace the calculation of the Euclidean distance<sup>2</sup>  $\text{dist}(\xi, \mathcal{P})$  between a point  $\xi \in \mathbb{R}^2$  and the desired path  $\mathcal{P}$  simply by the value of  $|\phi(\xi)|$ . For simplicity, rather than referring to  $\mathcal{P}$  in (1) as “the trace of a parameterized curve,” we call  $\mathcal{P}$  the desired path throughout the article. In fact, one feature of the VF-PF navigation problem is that the desired path  $\mathcal{P}$  is described by a 1-D connected submanifold, so we have the extra freedom of choosing different analytic expressions (i.e.,  $\phi$ ) for the same desired path.

If the desired path is nonself-intersected, then a valid 2-D vector field  $\chi : \mathbb{R}^2 \rightarrow \mathbb{R}^2$  to solve the VF-PF navigation problem is [16]

$$\chi(x, y) = E \nabla \phi(x, y) - k \psi(\phi(x, y)) \nabla \phi(x, y) \quad (2)$$

where  $E \in SO(2)$  is the 90° rotation matrix<sup>3</sup>  $\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$ , and  $\psi : \mathbb{R} \rightarrow \mathbb{R}$  is a strictly increasing function satisfying  $\psi(0) = 0$ . For simplicity, one can choose  $\psi(\phi(x, y)) = \phi(x, y)$ . The first term of the vector field is “tangential” to the desired path, thus enables a robot to move along the desired path, while the second term of the vector field is perpendicular to the first term, helping the robot move closer to the desired path. Therefore, intuitively, the vector field guides the robot to move toward and along the desired path at the same time. After the preliminaries on the guiding vector field on  $\mathbb{R}^2$  (i.e., 2-D vector field), we are ready to introduce in the next subsection the more abstract guiding vector field on  $\mathbb{R}^n$ , where  $n \geq 2$ .

### B. General Guiding Vector Field

Based on the description of a desired path  $\mathcal{P} \subseteq \mathbb{R}^n$ , the corresponding guiding vector field for path following is derived in this section. Some mild assumptions along with motivation and justification are presented.

**1) Guiding Vector Field Structure:** We introduce the general method of designing a vector field  $\chi : \mathbb{R}^n \rightarrow \mathbb{R}^n$  corresponding to a desired path  $\mathcal{P} \subseteq \mathbb{R}^n$ , where  $n \geq 2$ . The vector field is a generalization of those in [16], [22], while its structure is the same as that in [3]. However, we present crucial assumptions while abandoning the assumption in [3] about the repulsiveness of the set of singular points.

Suppose a desired path in the  $n$ -dimensional Euclidean space is described by the intersection of  $(n - 1)$  hypersurfaces, i.e.,

$$\mathcal{P} = \{\xi \in \mathbb{R}^n : \phi_i(\xi) = 0, i = 1, \dots, n - 1\} \quad (3)$$

<sup>2</sup>This calculation is generally difficult since one needs to find the closest point on the desired path to  $\xi$ ; e.g., it is not trivial for even an ellipse.

<sup>3</sup>In fact, the matrix is  $-E$  in [16], but we use  $E$  for conventional simplicity. This only affects the direction of the motion (forward or backward) on the desired path.

where  $\phi_i : \mathbb{R}^n \rightarrow \mathbb{R}$ ,  $i = 1, \dots, n-1$ , are of differentiability class  $C^2$ . It is naturally assumed that  $\mathcal{P}$  in (3) is *nonempty* and *connected*. We further require the regularity of the desired path as shown later in Assumption 1. For better understanding,  $\phi_i(\cdot) = 0$  can be regarded as  $(n-1)$  constraints, resulting in a one degree-of-freedom desired path.

*Remark 1:* Topologically, the desired path  $\mathcal{P}$  itself is 1-D, independent of the dimensions of the Euclidean space where it lives. However, with slight abuse of terminology and for convenience, the desired path  $\mathcal{P}$  is called an *n-D (or nD) desired path* if it lives in the  $n$ -dimensional Euclidean space  $\mathbb{R}^n$  and not in any lower dimensional subspace  $\mathcal{W} \subseteq \mathbb{R}^n$  (i.e., the *smallest subspace* the desired path lives in). For example, a planar desired path might be defined in the 3-D Euclidean space  $\mathbb{R}^3$ , but we only consider the 2-D subspace  $\mathcal{W} \subseteq \mathbb{R}^2$  where it is contained, and it is thus natural to call it a 2D (or 2-D) desired path rather than a 3D (or 3-D) desired path. Sometimes, for simplicity, we refer to a tangent vector field on the  $n$ -dimensional Euclidean space  $\mathbb{R}^n$  as an *n-dimensional vector field*, and we say that *this vector field is n-dimensional*. ◀

The vector field  $\chi : \mathbb{R}^n \rightarrow \mathbb{R}^n$  is designed as follows:

$$\chi = \times(\nabla\phi_1, \dots, \nabla\phi_{n-1}) - \sum_{i=1}^{n-1} k_i \phi_i \nabla\phi_i \quad (4)$$

where  $\nabla\phi_i$  is the gradient of  $\phi_i$ ,  $k_i > 0$  are constant gains, and  $\times : \mathbb{R}^n \times \dots \times \mathbb{R}^n \rightarrow \mathbb{R}^n$  is the generalized cross product. In particular, let  $p_i = (p_{i1}, \dots, p_{in})^\top \in \mathbb{R}^n$ ,  $i = 1, \dots, n-1$ , and  $\mathbf{b}_j \in \mathbb{R}^n$  be the standard basis column vector with the  $j$ th component being 1 and the other components being 0. Then an intuitive *formal expression* for  $\times(p_1, \dots, p_{n-1})$  is

$$\times(p_1, \dots, p_{n-1}) = \begin{vmatrix} \mathbf{b}_1 & \mathbf{b}_2 & \dots & \mathbf{b}_n \\ p_{11} & p_{12} & \dots & p_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{n-1,1} & p_{n-1,2} & \dots & p_{n-1,n} \end{vmatrix}. \quad (5)$$

In other words,  $\times(p_1, \dots, p_{n-1})$  is obtained by the cofactor expansion along the first row of (5), where  $\mathbf{b}_i$  should initially be regarded as scalars, and in the final evaluation replaced by the basis vectors [41, pp. 241–242]. The generalized cross product has the following property [42, Prop. 7.2.1].

*Lemma 1 (Orthogonality):* It holds that  $\times(p_1, \dots, p_{n-1})$  is orthogonal to each of the vectors  $p_1, \dots, p_{n-1}$ .

*Remark 2:* Due to Lemma 1, the physical interpretation of the vector field in (4) is clear. The first term  $\times(\nabla\phi_1, \dots, \nabla\phi_{n-1})$ , being perpendicular to each  $\nabla\phi_i$ , provides a tangential direction to each surfaces  $\phi_i(\xi) = 0$ , and hence “pushes” the robot along the desired path. The forward or backward direction of movement along the desired path is determined by the order of the gradient vectors. Thus, if the motion needs to be reversed, it is sufficient to swap any two of these vectors. We call this term the *propagation term*. The latter term  $-\sum_{i=1}^{n-1} k_i \phi_i \nabla\phi_i$  provides a direction toward those surfaces, acting as a “pulling force” to the desired path. We call this term the *converging term*. ◀

To simplify the notations, we define a matrix  $N(\xi) = (\nabla\phi_1(\xi), \dots, \nabla\phi_{n-1}(\xi)) \in \mathbb{R}^{n \times (n-1)}$ , a positive definite gain matrix  $K = \text{diag}\{k_1, \dots, k_{n-1}\} \in \mathbb{R}^{(n-1) \times (n-1)}$  and a  $C^2$

function  $e : \mathbb{R}^n \rightarrow \mathbb{R}^{n-1}$  by stacking  $\phi_i$ ; that is

$$e(\xi) = (\phi_1(\xi), \dots, \phi_{n-1}(\xi))^\top \in \mathbb{R}^{n-1}. \quad (6)$$

In addition, we define  $\nabla_\times \phi : \mathbb{R}^n \rightarrow \mathbb{R}^n$  by  $\xi \in \mathbb{R}^n \mapsto \times(\nabla\phi_1(\xi), \dots, \nabla\phi_{n-1}(\xi))$ . Therefore, the vector field (4) can be compactly written as

$$\chi(\xi) = \nabla_\times \phi(\xi) - N(\xi)K e(\xi). \quad (7)$$

Using this notation, the desired path is equivalent to

$$\mathcal{P} = \{\xi \in \mathbb{R}^n : e(\xi) = 0\}. \quad (8)$$

We call  $e(\xi)$  the *path-following error* or simply *error* between the point  $\xi \in \mathbb{R}^n$  and the desired path  $\mathcal{P}$ . An intuitive example is a 2-D circle described by the zero-level sets of  $e(x, y) = \phi_1(x, y) = x^2 + y^2 - r^2$ . As the point  $(x, y)$  approaches the circle, the norm of the path-following error  $\|e(x, y)\| = |x^2 + y^2 - r^2|$  decreases. When  $\|e(x, y)\| = 0$ , the point is right on the path. The use of  $\|e(\cdot)\|$  is more convenient than that of the distance function  $\text{dist}(\cdot, \mathcal{P})$ . However, there are subtle differences between the norm of the path-following error  $\|e(\cdot)\|$  and the distance  $\text{dist}(\cdot, \mathcal{P})$ ; e.g., when the norm of the path-following error converges to zero, the trajectory might not converge to the desired path [22, Example 3], [43]. Assumptions will be proposed to avoid this pathological situation.

*Remark 3:* The vector field in (4) is basic while effective in VF-PF navigation problems. First, we note that many existing studies only deal with simple desired paths such as a circle or a straight line (or a combination of them) [11], [18], [25], but the vector field in (4) is designed for any *generic* sufficiently smooth desired path. Second, many vector fields in the literature can be seen as *variants* of the vector field in (4). One type of variants are generated by adding  $\phi_i$ -dependent gains to the converging term or (and) the propagation term in (4) [5], [7], [16]. Another type of variants adds time-varying gains or an additional time-varying component [3], [17]. Thus, the vector fields in [5], [7], [16], and [44] can be regarded as 2-D variants of (4), and those in [7], [9], [13], [22], and [45], as 3-D variants of (4). Therefore, the study of the basic vector field in (4) is of great significance. Note that we do not consider time-varying gains or components in the vector field as [3] and [17] do. For one thing, this simplifies the structure of the vector field and facilitates the practical implementation; for another, this clarifies the topological property of these vector fields as studied in Section III. For convenience, we refer to these (time-invariant) vector fields in the literature as *conventional vector fields*. ◀

2) *Assumptions:* To justify using the norm of the path-following error  $\|e(\cdot)\|$  instead of  $\text{dist}(\cdot, \mathcal{P})$ , we need some assumptions that are easily satisfied in practice. These assumptions are based on [16] and [22], but are extended to  $\mathbb{R}^n$ . To this end, we define two sets. The *singular set* consisting of singular points of a vector field is defined as follows:

$$\mathcal{C} = \{\xi \in \mathbb{R}^n : \chi(\xi) = 0\}. \quad (9)$$

Another related set is

$$\mathcal{M} = \{\xi \in \mathbb{R}^n : N(\xi)K e(\xi) = 0\}. \quad (10)$$

It can be proved that  $\mathcal{M} = \mathcal{P} \cup \mathcal{C}$ .

*Lemma 2:* It holds that  $\mathcal{M} = \mathcal{P} \cup \mathcal{C}$ .

*Proof:* See Appendix A-A in the full version [34]. ■

Now we are ready to propose the following assumptions.

*Assumption 1:* There are no singular points on the desired path. More precisely,  $\text{dist}(\mathcal{C}, \mathcal{P}) > 0$ .

*Assumption 2:* In view of (8), as the norm of the path-following error  $\|e(\xi)\|$  approaches zero, the trajectory  $\xi(t)$  approaches the desired path  $\mathcal{P}$ . Similarly, in view of (10), as the “error”  $\|N(\xi)Ke(\xi)\|$  approaches zero, the trajectory  $\xi(t)$  approaches the set  $\mathcal{M}$ .

Assumption 1 leads to Lemma 3 about the topological property of the desired path, which is an extension of [22].

*Lemma 3:* The zero vector  $0 \in \mathbb{R}^{n-1}$  is a regular value of the  $C^2$  function  $e$  in (6), and hence the desired path  $\mathcal{P}$  is a  $C^2$  embedded submanifold<sup>4</sup> in  $\mathbb{R}^n$ .

*Proof:* See Appendix A-B in the full version [34]. ■

*Remark 4:* Henceforth the “regularity” of the desired path is guaranteed; namely, the desired path  $\mathcal{P}$  is assumed to be a 1-D connected manifold, which can generally be classified into those homeomorphic to the unit circle if they are compact, and those homeomorphic the real line otherwise [31, Th. 5.27]. Thus throughout the article, we use the notions of *simple closed desired paths* and *desired paths homeomorphic to the unit circle* interchangeably. The same applies to *unbounded desired paths* and *desired paths homeomorphic to the real line*. Note that self-intersected desired paths do not satisfy Assumption 1, as shown later in Proposition 1, but we will propose a method in Section V to transform them into unbounded and nonself-intersected desired paths, which are then homeomorphic to the real line  $\mathbb{R}$ . ◀

Assumption 2 is satisfied in many practical cases such as many polynomial or trigonometric functions; examples are demonstrated in [3], [4], [16], and [22]. Since the same desired path can be characterized by various choices of  $\phi_i(\cdot)$ , the assumption are crucial to exclude some pathological cases [22, Example 3], [43]. The mathematical formulation of Assumption 2 is presented in Appendix A-C in the full version [34].

### III. ISSUES ON THE GLOBAL CONVERGENCE TO DESIRED PATHS

In this section, we show that, under some conditions, it is not possible to guarantee global convergence to desired paths using the existing VF-PF algorithms as introduced in Section II. More specifically, given a desired path  $\mathcal{P} \subseteq \mathbb{R}^n$  as described in (3), we investigate solutions (trajectories) of the autonomous ordinary differential equation:

$$\dot{\xi}(t) = \chi(\xi(t)) \quad (11)$$

where  $\chi$  is defined in (7). We consider the cases of self-intersected and simple closed desired paths, respectively.

We first show that the crossing points of a self-intersected desired path  $\mathcal{P}$  are singular points of the corresponding vector field  $\chi$  in (4).

<sup>4</sup>See [24, pp. 105, 98–99] for the definitions of a regular value and an embedded submanifold.

*Proposition 1:* If the desired path  $\mathcal{P}$  in (3) is self-intersected, then the crossing points of the desired path are singular points of the vector field  $\chi$  in (4).

*Proof:* See Appendix B-A in the full version [34]. ■

*Remark 5:* This proposition shows that  $0 \in \chi(\mathcal{P})$  when  $\mathcal{P}$  is a self-intersected desired path, and therefore, the VF-PF navigation problem (Problem 1) cannot be addressed as the second requirement about  $0 \notin \chi(\mathcal{P})$  is always violated. Note that Assumption 1 does not hold in this case, but we will propose in the sequel an approach to transform a self-intersected desired path such that Assumption 1 holds. ◀

In Fig. 1(b), for example, the 2-D desired path resembling the figure “8” is self-intersected. It can be numerically calculated that the vector field at the crossing point is zero. This is intuitive in the sense that there is no “preference” for the vector at this point to point to either the left or right portion of the desired path, leaving the only option of zero.

Now, we consider simple closed desired paths. In the planar case, due to the Poincaré–Bendixson theorem [35, Corollary 2.1], there is at least one singular point of the 2-D vector field in the region enclosed by the simple closed desired path. Thus, we can conclude that global convergence to a simple closed planar desired path is not possible. However, this conclusion cannot be trivially generalized to the higher dimensional case since the Poincaré–Bendixson theorem is restricted to the planar case. Nevertheless, we can still reach this conclusion with some topological analysis.

*Proposition 2:* If an  $n$ -D desired path  $\mathcal{P} \subseteq \mathbb{R}^n$  described by (3) is simple closed, under the dynamics (11) where the guiding vector field  $\chi : \mathbb{R}^n \rightarrow \mathbb{R}^n$  is in (4), then it is not possible to guarantee the global convergence of trajectories of (11) to the desired path  $\mathcal{P}$ ; precisely, the domain of attraction of  $\mathcal{P}$  cannot be  $\mathbb{R}^n$ .

*Proof:* See Appendix B-B in the full version [34]. ■

Based on Propositions 1 and 2, we can reach the following key statement about the impossibility of global convergence to some desired paths.

*Theorem 1 (Impossibility of global convergence):* If an  $n$ -D desired path  $\mathcal{P} \subseteq \mathbb{R}^n$  described by (3) is *simple closed* or *self-intersected*, then it is not possible to guarantee the global convergence to the desired path with respect to the dynamics in (11) with the  $n$ -dimensional guiding vector field  $\chi$  in (4); more precisely, the domain of attraction of  $\mathcal{P}$  cannot be  $\mathbb{R}^n$ .

*Proof:* If the desired path  $\mathcal{P}$  is self-intersected, then by Proposition 1, there is at least one singular point on the desired path. Obviously, the path-following problem formulated by Problem 1 cannot be solved. If the desired path  $\mathcal{P}$  is simple closed, then the global convergence to the desired path is impossible by Proposition 2. ■

*Remark 6:* We note that the topological obstacle to global convergence to the desired path roots in two aspects: 1) the geometry of the desired path: being either simple closed or self-intersected; 2) the time-invariance property of the vector field. Although we show this topological obstacle only for the basic vector field in (4), this obstacle also exists for other variants of vector fields as listed in Remark 3. This is because the two aspects that lead to the topological obstruction mentioned



above also hold for these time-invariant vector fields. Note that a state-dependent positive scaling (e.g. the normalization) of vector fields does not affect the topological properties of interest (i.e., the phase portrait, or the convergent results) [46, Prop. 1.14]. ◀

To overcome this topological obstacle and satisfy Assumption 1 even for self-intersected desired paths, we propose a new idea in the sequel to construct unbounded and nonself-intersected desired paths from the originally simple closed or self-intersected desired paths by “cutting” and “stretching” them in a higher dimensional space. Indeed, such a higher dimensional desired path will codify or contain information about the (lower dimensional) physical desired path. Based on the proposed higher dimensional desired paths, we can derive a guiding vector field on this higher dimensional space and show that its singular set is empty. However, to take advantage of the new guiding vector field, we need to *transform* (or *project* in the linear transformation case) its integral curves into a lower dimensional subspace that contains the information of the physical desired path. The details of transformation into another space will be discussed in Section IV, and the detailed construction of a singularity-free guiding vector field on a higher dimensional space will be presented in Section V.

#### IV. EXTENDED DYNAMICS AND CONVERGENCE RESULTS

In this section and the subsequent sections, we consider an  $m$ -dimensional Euclidean space  $\mathbb{R}^m$ , where  $m > n$ . The reason is self-evident as the article develops, but it is not necessary to bother with this difference now. To proceed, we introduce the *extended dynamics* and derive related convergence results. The extended dynamics relates to a *transformation operator* defined as follows.

**Definition 1:** A transformation operator is a function  $G_l : \mathbb{R}^m \rightarrow \mathbb{R}^m$  which is twice continuously differentiable and *globally Lipschitz continuous* with the Lipschitz constant  $l$ .

One can observe that the corresponding Jacobian matrix function of a transformation operator  $\mathbf{D}G_l = \partial G_l / \partial x : \mathbb{R}^m \rightarrow \mathbb{R}^{m \times m}$  is locally Lipschitz continuous, where  $x$  is the argument of  $G_l$ . The transformation operator is able to transform a space into another space (or subspace). One example is a *linear transformation operator* defined by  $G_l(x) = Ax$ , where  $A$  is a nonzero matrix, called the *matrix representation* [47, Remark 6.1.15] of this particular linear transformation operator  $G_l$ . Now we introduce the extended dynamics as follows.

**Lemma 4 (Extended dynamics):** Let  $\chi : \mathcal{D} \subseteq \mathbb{R}^m \rightarrow \mathbb{R}^m$  be a vector field that is locally Lipschitz continuous. Given an initial condition  $\xi(0) = \xi_0 \in \mathcal{D}$ , suppose that  $\xi(t)$  is the unique solution to the differential equation  $\dot{\xi}(t) = \chi(\xi(t))$ , then  $(\xi(t), \xi^{\text{trs}}(t)) \in \mathbb{R}^{2m}$ , where  $\xi^{\text{trs}}(t) := G_l(\xi(t))$  and  $G_l$  is a transformation operator, is the unique solution to the following initial value problem:

$$\begin{cases} \dot{\xi}(t) = \chi(\xi(t)) & \xi(0) = \xi_0 \\ \dot{\xi}^{\text{trs}}(t) = \mathbf{D}G_l(\xi(t)) \cdot \chi(\xi(t)) & \xi^{\text{trs}}(0) = G_l(\xi_0) \end{cases} \quad (12)$$

Moreover, if the trajectory  $\xi(t)$  asymptotically converges to some set  $\mathcal{A} \neq \emptyset \subseteq \mathbb{R}^m$ , then  $\xi^{\text{trs}}(t)$  asymptotically converges

to the *transformed set*  $\mathcal{A}^{\text{trs}} := G_l(\mathcal{A}) = \{p \in \mathbb{R}^m : p = G_l(q), q \in \mathcal{A}\}$ .

*Proof:* See Appendix C-A in the full version [34]. ■

We call the ordinary differential equation with the initial condition in (12) the *extended dynamics*. Correspondingly,  $\xi^{\text{trs}}(t) := G_l(\xi(t))$  is called the *transformed solution* or *transformed trajectory* of (12). Before presenting Corollary 1 related to the VF-PF navigation problem, we first define the *transformed desired path* and the *transformed singular set*.

**Definition 2:** The *transformed desired path*  $\mathcal{P}^{\text{trs}}$  of  $\mathcal{P} \subseteq \mathbb{R}^m$  in (8) and the *transformed singular set*  $\mathcal{C}^{\text{trs}}$  of  $\mathcal{C} \subseteq \mathbb{R}^m$  in (9) are defined as follows:

$$\mathcal{P}^{\text{trs}} := G_l(\mathcal{P}) = \{p \in \mathbb{R}^m : p = G_l(q), q \in \mathcal{P}\} \quad (13)$$

$$\mathcal{C}^{\text{trs}} := G_l(\mathcal{C}) = \{p \in \mathbb{R}^m : p = G_l(q), q \in \mathcal{C}\}. \quad (14)$$

In some practical applications, it is desirable to scale the vector field to have a specified constant length. This is useful if a robot takes the vector field as the control input directly and is required to move at a constant speed. In this case, the properties of the integral curves of the scaled vector field are stated in the corollary below. Recall that the solution  $x(t)$  to an initial value problem  $\dot{x} = f(x)$ ,  $x(0) = x_0$ , where  $f(x)$  is sufficiently smooth, is not always possible to be prolonged to infinity. In other words, the solution might only be well-defined in a finite time interval  $[0, t^*)$ , where  $t^* < \infty$  [46]. The time instant  $t^*$  is called the maximal prolonged time of the solution.

**Corollary 1:** Suppose the desired path  $\mathcal{P}$  in (8) is unbounded (i.e.,  $\mathcal{P} \approx \mathbb{R}$ ). Let  $\chi : \mathcal{D} \subseteq \mathbb{R}^m \rightarrow \mathbb{R}^m$  be the vector field defined in (4). Suppose  $\xi(t)$  is the unique solution to the initial value problem  $\dot{\xi}(t) = s\hat{\chi}(\xi(t))$ ,  $\xi(0) = \xi_0 \notin \mathcal{C}$ , where  $s > 0$  is a constant and  $\hat{\cdot}$  is the normalization operator. Consider the following dynamics:

$$\begin{cases} \dot{\xi}(t) = s\hat{\chi}(\xi(t)) & \xi(0) = \xi_0 \notin \mathcal{C} \\ \dot{\xi}^{\text{trs}}(t) = \mathbf{D}G_l \cdot s\hat{\chi}(\xi(t)) & \xi^{\text{trs}}(0) = G_l(\xi_0) \end{cases} \quad (15)$$

where  $G_l$  is a transformation operator. Suppose  $t^* \leq \infty$  is the maximal prolonged time of the transformed solution  $\xi^{\text{trs}}(t)$  to (15). Then  $\xi^{\text{trs}}(t)$  asymptotically converges to the transformed desired path  $\mathcal{P}^{\text{trs}}$  in (13) as  $t \rightarrow \infty$  or the transformed singular set  $\mathcal{C}^{\text{trs}}$  in (14) as  $t \rightarrow t^*$ .

*Proof:* See Appendix C-B in the full version [34]. ■

**Remark 7:** Due to the normalization of the vector field in (15), the right-hand side of the differential equation is not well defined at singular points of the vector field. Therefore, if the transformed singular set  $\mathcal{C}^{\text{trs}}$  is bounded, then the maximal interval to which the transformed trajectory  $\xi^{\text{trs}}(t)$  can be prolonged is only finite when the transformed trajectory  $\xi^{\text{trs}}(t)$  is converging to  $\mathcal{C}^{\text{trs}}$ . This happens when the initial value  $\xi(0)$  is in the invariant manifold of the singular set  $\mathcal{C}$ . ◀

The previous lemma states that the *transformed trajectory* converges to either the *transformed desired path* or the *transformed singular set* for initial conditions  $\xi^{\text{trs}}(0) \in \mathbb{R}^m \setminus G_l(\mathcal{C})$ , while the latter case is undesirable. A preferable situation is where the (transformed) singular set is empty. Moreover, as indicated by Theorem 1, to seek for global convergence, the only possibility is to consider unbounded and nonself-intersected



desired paths (i.e.,  $\mathcal{P} \approx \mathbb{R}$ ). Therefore, we reach the following corollary.

**Corollary 2 (Global convergence to  $\mathcal{P}^{\text{trs}}$ ):** Suppose the desired path  $\mathcal{P}$  in (13) is unbounded (i.e.,  $\mathcal{P} \approx \mathbb{R}$ ). If  $\mathcal{C} = \emptyset$  (equivalently,  $\mathcal{C}^{\text{trs}} = \emptyset$ ), then the transformed trajectory  $\xi^{\text{trs}}(t)$  of (15) globally asymptotically converges to the transformed desired path  $\mathcal{P}^{\text{trs}}$  as  $t \rightarrow \infty$  in the sense that the initial condition  $\xi(0)$  (and hence  $\xi^{\text{trs}}(0)$ ) can be arbitrarily chosen in  $\mathbb{R}^m$ .

As will be shown later, only the second differential equation of (12) or (15) is relevant to the physical robotic system. This corollary thus motivates us to design a (higher dimensional) vector field such that the singular set is empty, in which case global convergence to the (transformed) desired path is guaranteed. In the next section, we will introduce an intuitive idea to “cut” and “stretch” a possibly simple closed or self-intersected physical desired path and create a higher dimensional singularity-free vector field.

## V. HIGH-DIMENSIONAL SINGULARITY-FREE GUIDING VECTOR FIELD CONSTRUCTION

In this section, we explain how to implicitly construct an unbounded desired path from the physical desired path (possibly simple closed or self-intersected) together with a higher dimensional guiding vector field without any singular points (a.k.a, singularity-free guiding vector field).

For simplicity, we restrict the transformation operator  $G_l : \mathbb{R}^m \rightarrow \mathbb{R}^m$  to a linear one defined by  $G_l(x) = P_a x$ , where  $P_a \in \mathbb{R}^{m \times m}$  is a nonzero matrix defined by

$$P_a = I - \hat{a}\hat{a}^\top \quad (16)$$

where  $I$  is the identity matrix of suitable dimensions and  $\hat{a} = a/\|a\| \in \mathbb{R}^m$  is a normalized nonzero vector. In this case,  $G_l$  is actually a linear transformation that projects an arbitrary vector to the hyperplane orthogonal to the given nonzero vector  $a$ , and  $P_a$  is the matrix representation of  $G_l$ . One can observe that the linear transformation  $G_l$  is globally Lipschitz continuous with the Lipschitz constant  $l = \|P_a\| = 1$ , where  $\|\cdot\|$  is the induced matrix two-norm. In addition, the Jacobian is simply  $\mathbf{D}G_l = P_a$ .

Before formulating the problem in the sequel, we define the *coordinate projection function*  $\pi_{(1,\dots,n)} : \mathbb{R}^m \rightarrow \mathbb{R}^n$  as

$$\pi_{(1,\dots,n)}(x_1, \dots, x_n, \dots, x_m) = (x_1, \dots, x_n)$$

where  $m > n$ . In other words, the coordinate projection function  $\pi_{(1,\dots,n)}$  takes only the first  $n$  components of an  $m$ -dimensional vector and generates a lower dimensional one.

**Problem 2:** Given an  $n$ -D physical desired path<sup>5</sup>  $\mathcal{P}^{\text{phy}}$  in  $\mathbb{R}^n$ , we aim to find an  $m$ -D desired path  $\mathcal{P}^{\text{hgh}}$  in  $\mathbb{R}^m$ , where  $m > n$ , which satisfies the following conditions.

- 1) There exist functions  $\phi_i(\cdot)$ ,  $i = 1, \dots, m-1$ , such that  $\mathcal{P}^{\text{hgh}}$  is described by (3).
- 2) The singular set  $\mathcal{C}^{\text{hgh}}$  of the *higher dimensional* vector field  $\chi^{\text{hgh}} : \mathbb{R}^m \rightarrow \mathbb{R}^m$  in (4) corresponding to  $\mathcal{P}^{\text{hgh}}$  is empty.

- 3) There exists a transformation operator  $G_l : \mathbb{R}^m \rightarrow \mathbb{R}^m$  such that  $\pi_{(1,\dots,n)}(\mathcal{P}^{\text{trs}}) = \mathcal{P}^{\text{phy}}$ , where the transformed desired path  $\mathcal{P}^{\text{trs}} = G_l(\mathcal{P}^{\text{hgh}})$ .

**Remark 8:** It is important to distinguish among the *physical desired path*  $\mathcal{P}^{\text{phy}}$ , the *higher dimensional desired path*  $\mathcal{P}^{\text{hgh}}$  and the *transformed desired path*  $\mathcal{P}^{\text{trs}}$ . A major difference is the dimensions of their ambient space; that is,  $\mathcal{P}^{\text{phy}} \subseteq \mathbb{R}^n$ , while  $\mathcal{P}^{\text{hgh}}, \mathcal{P}^{\text{trs}} \subseteq \mathbb{R}^m$  and  $m > n$ . Although the higher dimensional desired path  $\mathcal{P}^{\text{hgh}}$  and the transformed desired path  $\mathcal{P}^{\text{trs}}$  both live in  $\mathbb{R}^m$ , the transformed desired path  $\mathcal{P}^{\text{trs}}$  lives in a subspace  $\mathcal{W} \subseteq \mathbb{R}^m$  *probably* with  $\dim(\mathcal{W}) < m$  since  $\mathcal{P}^{\text{trs}} = G_l(\mathcal{P}^{\text{hgh}})$ . Indeed, for the case of a linear transformation operator in (16), the transformed desired path  $\mathcal{P}^{\text{trs}} = P_a(\mathcal{P}^{\text{hgh}})$  lives in the orthogonal complement subspace  $\mathcal{W}$  of the linear space spanned by the vector  $a$  (i.e.,  $\text{span}\{a\}$ ), and  $\dim(\mathcal{W}) = m - 1 < m$ . ◀

Next, we propose the solution to Problem 2 in Section V-A. Having found the higher dimensional desired path  $\mathcal{P}^{\text{hgh}}$ , then we can directly derive the corresponding vector field  $\chi^{\text{hgh}}$  defined on  $\mathbb{R}^m$  by (4). Some features of the approach illustrated in Section V-A are highlighted in Section V-B.

### A. Construction of a Singularity-Free Guiding Vector Field

Suppose an  $n$ -D physical path  $\mathcal{P}^{\text{phy}}$  is parameterized by

$$x_1 = f_1(w), \dots, x_n = f_n(w) \quad (17)$$

where  $w \in \mathbb{R}$  is the parameter of the desired path and  $f_i \in C^2$ ,  $i = 1, \dots, n$ . We can simply let

$$\phi_1(\xi) = x_1 - f_1(w), \dots, \phi_n(\xi) = x_n - f_n(w) \quad (18)$$

where  $\xi = (x_1, \dots, x_n, w)$  has an additional coordinate  $w$  now and is an  $m$ -dimensional vector, where  $m = n + 1$ . So the  $m$ -D desired path is

$$\begin{aligned} \mathcal{P}^{\text{hgh}} = \{ \xi = (x_1, \dots, x_n, w) \in \mathbb{R}^m : \phi_i(\xi) \\ = 0, i = 1, \dots, n \}. \end{aligned} \quad (19)$$

Thus the first requirement of Problem 2 is met. Intuitively, the new higher dimensional desired path  $\mathcal{P}^{\text{hgh}}$  is obtained by “cutting” and “stretching” the  $n$ -D desired path  $\mathcal{P}^{\text{phy}}$  along the additional virtual  $w$ -axis. From the higher dimensional desired path  $\mathcal{P}^{\text{hgh}} \subseteq \mathbb{R}^m$  in (19), we obtain the corresponding guiding vector field on the higher dimensional space  $\mathbb{R}^m$  by (4)

$$\chi^{\text{hgh}} = \nabla_\times \phi - \sum_{i=1}^n k_i \phi_i \nabla \phi_i.$$

It can be calculated that  $\nabla \phi_i = (0, \dots, 1, \dots, -f'_i(w))^\top$  for  $i = 1, \dots, n$ , where  $f'_i(w) := \frac{df_i(w)}{dw}$  and 1 is the  $i$ th component of the gradient vector. Therefore

$$\nabla_\times \phi = (-1)^n (f'_1(w), \dots, f'_n(w), 1)^\top \in \mathbb{R}^m = \mathbb{R}^{n+1}. \quad (20)$$

It is interesting that the  $m$ th coordinate of this vector is a constant  $(-1)^n$  regardless of the specific parametric form of the desired path. This means that  $\|\nabla_\times \phi(\xi)\| \neq 0$  for  $\xi \in \mathbb{R}^m$  globally. From Lemma 1, we know that the propagation term  $\nabla_\times \phi$  of the vector field is always linearly independent from the converging term  $\sum_{i=1}^n k_i \phi_i \nabla \phi_i$  unless they are zero vectors. However, as we

<sup>5</sup>Recall the notion of an  $n$ -D desired path in Remark 1.

have shown that  $\|\nabla_{\times}\phi\| \neq 0$  in  $\mathbb{R}^m$  globally, this reveals the appealing property that the vector field  $\chi^{\text{hgh}}(\xi) \neq 0$  for any point  $\xi \in \mathbb{R}^m$ , implying that there are no singular points in the higher dimensional space  $\mathbb{R}^m$ ; i.e.,  $\mathcal{C}^{\text{hgh}} = \emptyset$ . Thus, the second requirement of Problem 2 (as well as a related condition in Corollary 2) is satisfied.

To let the third requirement of Problem 2 be satisfied, we retreat to a linear transformation operator with a matrix representation  $P_a$ . One of the simplest linear transformation operators corresponds to  $a = \mathbf{b}_{n+1} \in \mathbb{R}^m$ , which is a standard basis column vector with the  $(n+1)$ th component being 1 and the other components being 0. This is used to transform an  $m$ -dimensional space to an  $n$ -dimensional subspace by “zeroing” the last coordinate. Specifically, we let  $a = \mathbf{b}_{n+1}$ , then the matrix representation of the linear transformation operator is  $P_a = \begin{bmatrix} I_{n \times n} & \mathbf{0} \\ \mathbf{0} & 0 \end{bmatrix}$ , where  $\mathbf{0}$  are zero vectors of suitable dimensions. Observe that the  $n$ -D desired path  $\mathcal{P}^{\text{phy}} \subseteq \mathbb{R}^n$  is the orthogonal projection of the higher dimensional desired path  $\mathcal{P}^{\text{hgh}} \subseteq \mathbb{R}^m$  on the plane where  $w = 0$ ; i.e.,

$$\pi_{(1,\dots,n)}(\mathcal{P}^{\text{hgh}}) = \pi_{(1,\dots,n)}(\mathcal{P}^{\text{trs}}) = \mathcal{P}^{\text{phy}}.$$

Therefore, the third requirement of Problem 2 is also satisfied. By the construction in (19), the higher dimensional desired path  $\mathcal{P}^{\text{hgh}} \subseteq \mathbb{R}^m$  satisfying all the conditions in Problem 2 is thus found. Ultimately, we can take advantage of the new “well-behaved” guiding vector field  $\chi^{\text{hgh}} \in \mathbb{R}^m$  derived from  $\mathcal{P}^{\text{hgh}} \subseteq \mathbb{R}^m$  as mentioned above. This result is formally stated in the following theorem.

**Theorem 2:** Suppose an  $n$ -D physical desired path  $\mathcal{P}^{\text{phy}} \subseteq \mathbb{R}^n$  is parameterized by (17). If  $\phi_1, \dots, \phi_n$  are chosen as in (18), then there are no singular points in the corresponding guiding vector field  $\chi^{\text{hgh}} : \mathbb{R}^{n+1} \rightarrow \mathbb{R}^{n+1}$  defined on the  $(n+1)$ -dimensional space  $\mathbb{R}^{n+1}$ . Let  $a = \mathbf{b}_{n+1}$  for the linear transformation operator  $P_a$ . Suppose the transformed trajectory of the extended dynamics (15) is  $\xi^{\text{trs}}(t) := (x_1(t), \dots, x_n(t), w(t))^{\top}$ . Then the projected transformed trajectory

$$\xi^{\text{prj}}(t) := \pi_{(1,\dots,n)}(\xi^{\text{trs}}(t)) = (x_1(t), \dots, x_n(t))^{\top}$$

globally asymptotically converges to the physical desired path  $\mathcal{P}^{\text{phy}}$  as  $t \rightarrow \infty$ .

*Proof:* By (4) and (18), the guiding vector field on the  $(n+1)$ -dimensional space  $\mathbb{R}^{n+1}$  is

$$\chi^{\text{hgh}}(x_1, \dots, x_n, w) = \begin{bmatrix} (-1)^n f'_1(w) - k_1 \phi_1 \\ \vdots \\ (-1)^n f'_n(w) - k_n \phi_n \\ (-1)^n + \sum_{i=1}^n k_i \phi_i f'_i(w) \end{bmatrix}. \quad (21)$$

As discussed before, the singular set  $\mathcal{C}^{\text{hgh}} = \emptyset$ . According to Corollary 2,  $\xi^{\text{trs}}(t)$  globally asymptotically converges to the transformed desired path  $\mathcal{P}^{\text{trs}} = G_l(\mathcal{P}^{\text{hgh}}) = P_a(\mathcal{P}^{\text{hgh}})$  as  $t \rightarrow \infty$ . Since  $a^{\top} \xi^{\text{trs}} = a^{\top} P_a \xi = 0$ , the  $(n+1)$ th coordinate  $w(t)$  of the transformed trajectory  $\xi^{\text{trs}}(t)$  is equal to 0, meaning that the transformed trajectory  $\xi^{\text{trs}}(t)$  lies on the subspace

$\mathcal{W} := \{(x_1, \dots, x_{n+1}) \in \mathbb{R}^n : x_{n+1} = 0\}$ . Therefore, the projected transformed trajectory  $\xi^{\text{prj}}(t) = \pi_{(1,\dots,n)}(\xi^{\text{trs}}(t))$  globally asymptotically converges to the physical desired path  $\mathcal{P}^{\text{phy}}$ . ■

**Remark 9:** Note that the proof of convergence to the physical desired path  $\mathcal{P}^{\text{phy}}$  is indirect. The norm of the path-following error  $\|e(\cdot)\| = \|(\phi_1(\cdot), \dots, \phi_n(\cdot))\|$  captures the distance to the higher dimensional desired path  $\mathcal{P}^{\text{hgh}}$ , taking into account the additional coordinate  $w$  as well. It is shown first that in the higher dimensional space  $\mathbb{R}^{n+1}$ , the norm of the path-following error  $\|e(\cdot)\|$  approaches zero asymptotically. Then the convergence to the transformed desired path  $\mathcal{P}^{\text{trs}}$  is obtained from Corollary 1 (or Corollary 2). Due to the special choice of the linear transformation operator  $P_a$ , where  $a = \mathbf{b}_{n+1}$ , the transformed desired path  $\mathcal{P}^{\text{trs}}$  is “almost” the same as the physical desired path  $\mathcal{P}^{\text{phy}}$ , except that it has an additional but constant coordinate  $w(t) \equiv 0$ . ◀

We have shown that, by extending the vector field from  $\mathbb{R}^n$  to  $\mathbb{R}^{n+1}$ , the new guiding vector field does not have any singular points. Therefore, by using the extended dynamics, the convergence to the physical desired path is guaranteed globally. When  $n > 3$ , this case corresponds to some configuration spaces, such as the robot arm joint space in a smooth manifold embedded in  $\mathbb{R}^n$ . See [36] for more details.

## B. Features of the Approach

There are several intriguing features of our proposed approach discussed above in Section V-A. These features are summarized below, and the corresponding theoretical guarantees are presented in Appendix D in the full version [34]. For ease of narration and without loss of generality, we take the case of a 2-D physical desired path  $\mathcal{P}^{\text{phy}} \subseteq \mathbb{R}^2$  for discussion (i.e.,  $n = 2$ ).

**Feature 1:** The corresponding higher dimensional desired path  $\mathcal{P}^{\text{hgh}} = \{\xi \in \mathbb{R}^{2+1} : \phi_1(\xi) = 0, \phi_2(\xi) = 0\}$  is not self-intersected. This is due to the fact that a crossing point must be a singular point (see Proposition 1), but we have shown that there are no singular points in the higher dimensional guiding vector field. In fact, the parameter of the desired path  $w$  in (17) is implicitly transformed to an additional coordinate of the higher dimensional desired path. Thus the physical planar desired path  $\mathcal{P}^{\text{phy}}$  is “cut” and “stretched” into the 3-D Euclidean space, and becomes unbounded and nonself-intersected along the additional dimension. The significance of this feature is that even a self-intersected physical desired path  $\mathcal{P}^{\text{phy}}$  described by (17) can be successfully followed by using the new singularity-free guiding vector field, which in fact corresponds to a nonself-intersected “stretched” desired path  $\mathcal{P}^{\text{hgh}}$ .

**Feature 2:** This approach facilitates the expression of hypersurfaces characterized by implicit functions  $\phi_i$ . Usually, a parameterized form of the desired path is more readily available than the hypersurfaces of which the intersection is the desired path. Therefore, given the parameterized form in (17), we do not need to convert them into  $\phi(x, y) = 0$  and derive the corresponding 2-D vector field. Instead, by simply defining two  $\phi$  functions as in (18), we obtain a singularity-free vector field  $\chi^{\text{hgh}}$  defined on  $\mathbb{R}^{2+1}$ .

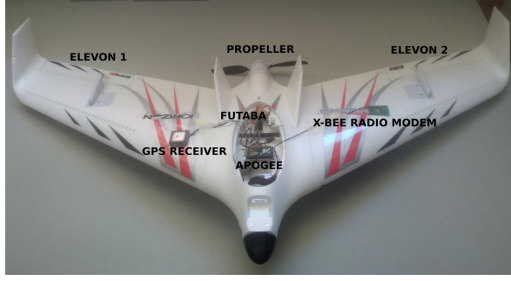


Fig. 2. Autonomous Opterra 1.2 m equipped with Paparazzi's Apogee autopilot. The airframe is built by E-Flite/Horizon Hobby company.

**Feature 3:** One only needs to examine the boundedness of  $|f'_i(z)|$ ,  $i = 1, 2$ , in the vicinity of the higher dimensional desired path  $\mathcal{P}^{\text{hgh}}$  to guarantee both the property of local exponential vanishing of the norm of the path-following error  $\|e\|$  and the property of robustness against disturbance of the system dynamics (11), while these properties usually require more conditions to be satisfied for general vector fields [13]. The theoretical guarantees are shown in Propositions 4 and 5, respectively, in Appendix D in the full version [34].

**Feature 4:** Only Assumption 2 is required. Since the new guiding vector field does not have any singular points, the other assumption, Assumption 1, is vacuously true. This is independent of the specific parametrizations of the desired path in (17). The intuitive Assumption 2 holds for many practical examples; thus one might ignore it in practice [13], [16], [22].

**Feature 5:** The additional virtual coordinate can be used to realize scalable distributed multirobot coordinated path-following navigation by adding a consensus term and guarantee collision avoidance by using a safety barrier certificate [48].

## VI. EXPERIMENTS WITH AN AUTONOMOUS AIRCRAFT

In this section, we demonstrate the effectiveness of our path following approach with an autonomous fixed-wing aircraft. In particular, we verify the tracking of both 2-D and 3-D self-intersected desired paths. All the related software has been developed within the open-source project for autopilots *Paparazzi* [49]. The codes only require the corresponding parametric equations to implement other desired paths.<sup>6</sup>

### A. Autonomous Aircraft and Airfield

For the experiments, we use one Opterra as shown in Fig. 2. Two *elevons* actuate the aircraft at the wings and one motor acts in pushing the configuration. The vehicle's electronics consists of the autopilot *Apogee*, an Ublox GPS receptor, a Futaba receiver, and a X-Bee S1 radio modem. The Apogee's core is an STM32F4 microcontroller where our algorithm runs with a fixed frequency of 50 Hz, and all the relevant data are logged in an SD card at 100 Hz. The ground segment consists of a standard laptop with another X-Bee S1 radio modem to monitor the telemetry and a Futaba transmitter in case of taking over manual control of

the vehicle. The flights took place on July 18, 2020, in Ciudad Real (Spain) with GPS coordinates (39.184535, -4.020797) degrees. The weather forecast reported 36 °C and a South wind of 14 km/h.

### B. Aircraft's Guidance System Design

We employ a decoupled vertical and horizontal model for setting the aircraft's guiding reference signals. In particular, accounting for the nonholonomic lateral constraint of the aircraft, we consider the following unicycle model:

$$\dot{x} = v \cos \theta \quad \dot{y} = v \sin \theta \quad \dot{\theta} = u_\theta \quad \dot{z} = u_z \quad (22)$$

where  $(x, y, z)$  is the 3-D position,  $\theta$  is the heading angle on the XY plane,  $v$  is the ground speed,  $u_\theta$  is the angular velocity control/guiding signal to change the heading, and  $u_z$  is the guiding signal for the climbing velocity. We will show how to design the guiding signals  $u_\theta$  and  $u_z$ , which are injected into the control system of the aircraft that deals with the nontrivial couplings of the lateral and vertical modes. Particularly,  $u_\theta$  is tracked by banking the aircraft depending on the current speed  $v$  and the *pitch* angle to achieve a *coordinated turn*, and  $u_z$  is tracked by controlling the pitch angle and the propulsion to vary the lift and the vertical component of the pushed force coming from the propeller. We leave the reader to check the details of the employed low-level controllers.<sup>7</sup> The experiments will show that our algorithm is compatible with the model (22) and the low-level controller employed in Paparazzi for a fixed-wing aircraft.

Note that the wind has a noticeable impact on the ground speed of the aircraft. Nevertheless, as the experimental results indicate, such a wind speed does not impact the intended performance of the algorithm. In practice, we consider  $\theta$  as the heading angle (given by the velocity vector), not the attitude *yaw* angle. If there is no wind, both angles are the same in our setup. When we consider the heading instead of the yaw for the model (22), the aircraft compensates the lateral wind by *crabbing* such that aerodynamic angle *sideslip* is almost zero.<sup>8</sup>

For following 3-D paths (including 2-D paths at a constant altitude), we will employ a higher dimensional 4-D vector field. The generalized 4-D velocity vector of the aircraft is defined as  $\xi = (\dot{x}, \dot{y}, \dot{z}, \dot{w})^\top$ , where  $(\dot{x}, \dot{y})$  is the actual ground velocity of the aircraft,  $\dot{z}$  is the vertical speed, and  $\dot{w}$  is the velocity in the additional coordinate to be determined. Now we present the control algorithm design; that is, the design of  $u_\theta$  and  $u_z$  in (22) with the following proposition:

**Proposition 3:** Suppose the 3-D physical desired path  $\mathcal{P}^{\text{phy}} \subseteq \mathbb{R}^3$  to follow is parameterized by (17). Then a corresponding 4-D vector field  $\chi : \mathbb{R}^4 \rightarrow \mathbb{R}^4$  can be constructed by Theorem 2. Assume that the vector field satisfies  $\chi_1(\xi)^2 + \chi_2(\xi)^2 \neq 0$  for  $\xi \in \mathbb{R}^4$ , where  $\chi_i$  denotes the  $i$ th entry of  $\chi$ .

<sup>7</sup>[Online]. Available: [http://wiki.paparazziuav.org/wiki/Control\\_Loops](http://wiki.paparazziuav.org/wiki/Control_Loops)

<sup>8</sup>Crabbing happens when the inertial velocity makes an angle with the nose heading due to wind. *Slipping* happens when the aerodynamic velocity vector makes an angle (sideslip) with the body ZX plane. Slipping is (almost) always undesirable because it degrades aerodynamic performance. Crabbing is not an issue for the aircraft.

<sup>6</sup>[Online]. Available: [https://github.com/noether/paparazzi/tree/gvf\\_advanced/sw/airborne/modules/guidance/gvf\\_parametric](https://github.com/noether/paparazzi/tree/gvf_advanced/sw/airborne/modules/guidance/gvf_parametric)



Consider the model (22), and let the dynamics of the additional coordinate  $w$  be

$$\dot{w} = \frac{v\chi_4}{\sqrt{\chi_1^2 + \chi_2^2}}. \quad (23)$$

Let the angular velocity control input  $u_\theta$  and the climbing velocity input  $u_z$  be

$$u_\theta = \underbrace{\left( \frac{-1}{\|\hat{\chi}^p\|} \hat{\chi}^{p\top} E J(\chi^p) \dot{\xi} \right)}_{:=\hat{\theta}_d} - k_\theta \hat{h}^\top E \hat{\chi}^p \quad (24a)$$

$$u_z = \frac{v\chi_3}{\sqrt{\chi_1^2 + \chi_2^2}} \quad (24b)$$

where  $k_\theta > 0$  is a gain constant,  $h = (\cos \theta, \sin \theta)^\top$ ,  $E = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$ ,  $\chi^p = (\hat{\chi}_1, \hat{\chi}_2)^\top$  and  $J(\chi^p)$  is the Jacobian matrix of  $\chi^p$  with respect to the generalized position  $\xi = (x, y, z, w)$  and  $\dot{\xi} = (\dot{x}, \dot{y}, \dot{z}, \dot{w})^\top$  is the generalized velocity. Let the angle difference directed from  $\hat{\chi}^p$  to  $\hat{h}$  be denoted by  $\beta \in (-\pi, \pi]$ . If the initial angle difference satisfies  $\beta(0) \in (-\pi, \pi)$ , then it will vanish asymptotically (i.e.,  $\beta(t) \rightarrow 0$ ). Furthermore, the actual robot trajectory  $(x(t), y(t), z(t))$  will converge to the physical desired path  $\mathcal{P}^{\text{phy}}$  asymptotically as  $t \rightarrow \infty$ .

*Proof:* Let  $\chi' := \frac{1}{\sqrt{\chi_1^2 + \chi_2^2}} \chi$  be the scaled 4-D vector field.

We aim to let the generalized robot velocity  $\dot{\xi} = (\dot{x}, \dot{y}, \dot{z}, \dot{w})^\top$  eventually align with and point towards the same direction as the scaled vector field. Specifically, let the orientation error be defined by

$$e_{ori}(t) = \dot{\xi} - v\chi' \stackrel{(23), (24b)}{=} v \begin{bmatrix} \cos \theta - \chi'_1 \\ \sin \theta - \chi'_2 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} h - g \\ \mathbf{0} \end{bmatrix} \in \mathbb{R}^4$$

where  $h = (\cos \theta, \sin \theta)^\top$  and  $g = (\chi'_1, \chi'_2)^\top$ . It is obvious that  $e_{ori} \rightarrow 0$  if and only if  $h - g \rightarrow 0$ . Therefore, it suffices to show that the orientation of  $h$  asymptotically aligns with that of  $g$ . Note that

$$\hat{\chi}^p = \frac{1}{\sqrt{\hat{\chi}_1^2 + \hat{\chi}_2^2}} \begin{bmatrix} \hat{\chi}_1 \\ \hat{\chi}_2 \end{bmatrix} = \frac{1}{\sqrt{\chi_1^2 + \chi_2^2}} \begin{bmatrix} \chi_1 \\ \chi_2 \end{bmatrix} = g$$

and  $\hat{h} = h$ . Therefore, we can define a new orientation error as  $e'_{ori} := \hat{h} - \hat{\chi}^p \in \mathbb{R}^2$ . Choose the Lyapunov function candidate  $V = 1/2 e_{ori}^\top e'_{ori}$  and its time derivative is

$$\begin{aligned} \dot{V} &= e_{ori}^\top \dot{e}'_{ori} = (\dot{\theta} E \hat{h} - \dot{\theta}_d E \hat{\chi}^p)^\top (\hat{h} - \hat{\chi}^p) \\ &= (\dot{\theta} - \dot{\theta}_d) \hat{h}^\top E \hat{\chi}^p \stackrel{(24a)}{=} -k_\theta (\hat{h}^\top E \hat{\chi}^p)^2 \end{aligned} \quad (25)$$

which is negative semidefinite. The second equation makes use of the identities:  $\dot{\hat{h}} = \dot{\theta} E \hat{h}$  and  $\dot{\hat{\chi}}^p = \dot{\theta}_d E \hat{\chi}^p$ , where  $\dot{\theta}_d$  is defined in (24a). The third equation is derived by exploiting the facts that  $E^\top = -E$  and  $a^\top E a = 0$  for any vector  $a \in \mathbb{R}^2$ . Note that  $\dot{V} = 0$  if and only if the angle difference between  $\hat{h}$  and  $\hat{\chi}^p$  is  $\beta = 0$  or  $\beta = \pi$ . Since it is assumed that the initial angle difference  $\beta(t=0) \neq \pi$ , it follows that  $\dot{V}(t=0) < 0$ , and thus, there exists a sufficiently small  $\epsilon > 0$  such that  $V(t=\epsilon) < V(t=0)$ .

It can be shown by contradiction that  $|\beta(t)|$  is monotonically decreasing with respect to time  $t$ .<sup>9</sup> By (25), one observes that  $|\beta(t)|$  and  $V(t)$  tends to 0, implying that the generalized velocity  $\dot{\xi}$  will converge asymptotically to the scaled vector field  $v\chi'$ . Note that the integral curves of the state-dependent positive scaled vector field  $\chi'$  has the same convergence results as those for the original vector field  $\chi$  [46, Prop. 1.14]. Therefore, the generalized trajectory  $(x(t), y(t), z(t), w(t))$  will converge to the higher dimensional desired path  $\mathcal{P}^{\text{high}}$  in (19). From Theorem 2, the actual robot trajectory (i.e., the projected transformed trajectory)  $(x(t), y(t), z(t))$  will converge to the physical desired path  $\mathcal{P}^{\text{phy}}$  asymptotically as  $t \rightarrow \infty$ . ■

We set our aircraft to fly at a constant airspeed (around 12 m/s) while flying at a constant altitude; therefore, we have a bounded speed  $v$  (estimated onboard with an inertial navigation system) when we account for the wind. For tracking 3-D paths, the aircraft will nose down or change the propeller's r.p.m.; nevertheless, the airspeed is also bounded between 9 and 16 m/s. Note that both ground and airspeed are not control/guiding signals; therefore, we do not face any saturation problems regarding these variables.

### C. Accommodating the Guidance to the Aircraft's Dynamics

An arbitrary function  $\phi_i(\cdot)$  in (18), which depends on a specific parametrization  $f_i(\cdot)$ , may result in a highly sensitive coordinate  $w$ . This can lead to considerable vibrations of the guidance signals, due to noisy sensor readings or disturbances of the position, that cannot be tracked effectively by the aircraft.

We propose two approaches, which can be combined to mitigate this practical effect. The first one is to reparameterize the equations for the 3-D desired path  $\mathcal{P}^{\text{phy}}$ ; this does not affect the convergence result. Suppose  $\mathcal{P}^{\text{phy}}$  is reparameterized by

$$x = f_1(g(w)), \quad y = f_2(g(w)), \quad z = f_3(g(w))$$

where  $g: \mathbb{R} \rightarrow \mathbb{R}$  is a smooth bijection with nonzero derivative (i.e.,  $\frac{dg}{dw}(w) \neq 0$  for all  $w \in \mathbb{R}$ ). A simple example of  $g$  is  $g(w) = \beta w$ , where  $\beta$  is a positive constant. This is adopted for the experiments. Let  $\phi_1$ ,  $\phi_2$ , and  $\phi_3$  be chosen as in (18), then the first term of the higher dimensional vector field becomes (for simplicity, the arguments are omitted)

$$\times (\nabla \phi_1, \nabla \phi_2, \nabla \phi_3) = - \left( \frac{df_1}{dg} \frac{dg}{dw}, \frac{df_2}{dg} \frac{dg}{dw}, \frac{df_3}{dg} \frac{dg}{dw}, 1 \right)^\top.$$

To reduce the effect of the “virtual speed” from the fourth coordinate of the equation above, the “gain”  $\frac{dg}{dw}$  can be chosen large such that  $(\frac{df_1}{dg} \cdot \frac{dg}{dw})^2 + (\frac{df_2}{dg} \cdot \frac{dg}{dw})^2 + (\frac{df_3}{dg} \cdot \frac{dg}{dw})^2 \gg 1$ , which implies that  $\|\nabla \phi_1 \times \nabla \phi_2 \times \nabla \phi_3\| \approx \left| \frac{dg}{dw} \right| \sqrt{(\frac{df_1}{dg})^2 + (\frac{df_2}{dg})^2 + (\frac{df_3}{dg})^2}$ . However, from the analytic

<sup>9</sup>Suppose there exist  $0 < t_1 < t_2$  such that  $|\beta(t_1)| < |\beta(t_2)|$ . It can be calculated that  $V(t) = 1 - \cos \beta(t)$ , and thus,  $V(t_1) < V(t_2)$ , contradicting the decreasing property of  $\dot{V}$ . Thus,  $|\beta(t)|$  is indeed monotonically decreasing.

expression of the vector field

$$\chi = \begin{bmatrix} -\frac{dg}{dw} \cdot \frac{df_1}{dg} - k_1 \phi_1 \\ -\frac{dg}{dw} \cdot \frac{df_2}{dg} - k_2 \phi_2 \\ -\frac{dg}{dw} \cdot \frac{df_3}{dg} - k_3 \phi_3 \\ -1 + \frac{dg}{dw} \left( k_1 \phi_1 \frac{df_1}{dg} + k_2 \phi_2 \frac{df_2}{dg} + k_3 \phi_3 \frac{df_3}{dg} \right) \end{bmatrix}$$

one observes that, when  $\|(\phi_1, \phi_2, \phi_3)\|$  is large, (i.e., the aircraft is far from the desired path), the additional coordinate of the vector has also been enlarged approximately by a factor of  $\frac{dg}{dw}$ . Thus, the “gain”  $|\frac{dg}{dw}|$  should not be chosen too large.

The second approach is to scale down the functions  $\phi_i$ . That is, (18) are changed to  $\tilde{\phi}_i(x, y, z, w) = L\phi_i, i = 1, 2, 3$ , where  $L \in (0, 1)$ . The corresponding 3-D vector field is thus changed to

$$\tilde{\chi} = L \begin{bmatrix} -L^2 \frac{df_1}{dw} - k_1 \phi_1 \\ -L^2 \frac{df_2}{dw} - k_2 \phi_2 \\ -L^2 \frac{df_3}{dw} - k_3 \phi_3 \\ -L^2 + k_1 \phi_1 \frac{df_1}{dw} + k_2 \phi_2 \frac{df_2}{dw} + k_3 \phi_3 \frac{df_3}{dw} \end{bmatrix}.$$

The new guiding vector field is scaled down; thus, it helps to lower the sensitivity of the additional coordinate  $w$ .

#### D. Two-Dimensional Trefoil Curve

We start with following a 2-D self-intersected desired path, the trefoil curve, at a constant altitude  $z_o = 50$  m over the ground level. The parametric equations of the trefoil curve are given by

$$\begin{aligned} f_1(w) &= \cos(\beta w \omega_1)(a \cos(\beta w \omega_2) + b) \\ f_2(w) &= \sin(\beta w \omega_1)(a \cos(\beta w \omega_2) + b) \\ f_3(w) &= 0 \end{aligned}$$

where we have set  $\beta = \frac{dg}{dw} = 0.45$  (the “gain” introduced in Section VI-C),  $\omega_1 = 0.02, \omega_2 = 0.03, a = 80$ , and  $b = 160$ . In order to fit into the available flying space, these parametric equations have been rotated by  $\alpha$  and shifted adequately by  $(x_o, y_o)$  in the autopilot, i.e.,

$$\begin{aligned} f_1^*(w) &= \cos(\alpha)f_1(w) - \sin(\alpha)f_2(w) + x_o \\ f_2^*(w) &= \sin(\alpha)f_1(w) + \cos(\alpha)f_2(w) + y_o \\ f_3^*(w) &= f_3(w) + z_o. \end{aligned}$$

Note that the same affine transformation must be done for both  $f_i'$  and  $f_i''$  (needed for the Jacobian of  $\chi$  as we will see shortly). In particular, for the presented flight, we set  $x_o = 79, y_o = -68.10$  and  $z_o = 50$  meters and  $\alpha = 0$ . We set the scaling factor  $L = 0.1$  for the construction of  $\tilde{\phi}_i$  as in Section VI-C, and we choose

the gains  $k_1 = k_2 = k_3 = 0.002$ . We finally set  $k_\theta = 1$  for the control/guidance signal  $u_\theta$  in Proposition 3.

Note that for computing all the control signals (24), we need  $f_i(w)$  and their derivatives  $f_i'(w)$  and  $f_i''(w)$  with respect to  $w$ . For the sake of completeness, we provide the Jacobian  $J(\chi^p)$  in (24a), which is given by  $J(\chi^p) = FJ(\tilde{\chi}) = F(I - \hat{\chi}\hat{\chi}^\top)J(\chi)/\|\chi\|$ , where  $F = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$ , and  $J(\chi)$  is shown in (26). We show the flight results for the trefoil curve in Fig. 3.

#### E. Three-Dimensional Lissajous Curve

We consider the 3-D Lissajous curve described as below:

$$\begin{aligned} f_1(w) &= c_x \cos(\beta w \omega_x + d_x) \\ f_2(w) &= c_y \cos(\beta w \omega_y + d_y) \\ f_3(w) &= c_z \cos(\beta w \omega_z + d_z) \end{aligned}$$

where we have set  $\beta = \frac{dg}{dw} = 0.01, \omega_x = 1, \omega_y = \omega_z = 2, c_x = c_y = 225, c_z = -20, d_x = d_z = 0$ , and  $d_y = \pi/2$ . This selection of parameters gives us an *eight-shaped* desired path that is bent along the vertical axis. As with the trefoil curve, we have added an affine transformation of  $f_i(w), f_i'(w)$  and  $f_i''(w)$  in the autopilot to fit the Lissajous curve into the available flying space. In particular, we have set  $x_o = 79, y_o = -68.10, z_o = 50, \alpha = 0.66$ . Finally, for the construction of  $\tilde{\phi}_i$ , we have chosen  $L = 0.1, k_1 = k_2 = 0.002$  and  $k_3 = 0.0025$ , and we finally set  $k_\theta = 1$  for the control/guiding signal  $u_\theta$  in Proposition 3. We show the flight results in Fig. 4.

### VII. DISCUSSION: PATH FOLLOWING OR TRAJECTORY TRACKING?

In this section, we show that our proposed higher dimensional VF-PF algorithm is an extension that combines elements from both conventional VF-PF algorithms (e.g., [16], see Remark 3) and trajectory tracking algorithms (e.g., [50, p. 506]). While our generated guiding vector field is the standard output for the path-following approach, we will argue that our algorithm can also be seen as a fair extension of a trajectory tracking approach. Therefore, our algorithm, to some extent, combines and extends elements from both approaches. For ease of explanation and without loss of generality, we restrict our focus to a physical planar desired path in  $\mathbb{R}^2$ ; that is,  $\mathcal{P}^{\text{phy}} \subseteq \mathbb{R}^2$ .

Compared to trajectory tracking algorithms, a similarity exists in the sense that the additional coordinate  $w$  in the proposed VF-PF algorithms acts like the time variable in trajectory tracking algorithms. However, our approach is an extension in the sense that the time-like variable is in fact state-dependent. In trajectory tracking algorithms, a desired trajectory  $(x_d(t), y_d(t))$  is given. Then, at any time instant  $t$ , the algorithm aims to decrease the

$$J(\chi) = L \begin{bmatrix} -k_1 L & 0 & 0 & -f_1''(\beta w)L^2\beta^2 + k_1\beta L f_1'(\beta w) \\ 0 & -k_2 L & 0 & -f_2''(\beta w)L^2\beta^2 + k_2\beta L f_2'(\beta w) \\ 0 & 0 & -k_3 L & -f_3''(\beta w)L^2\beta^2 + k_3\beta L f_3'(\beta w) \\ k_1\beta L f_1'(\beta w) & k_2\beta L f_2'(\beta w) & k_3\beta L f_3'(\beta w) & \beta^2 \sum_{i=1}^3 [k_i \phi_i f_i''(\beta w) - k_i L f_i'^2(\beta w)] \end{bmatrix} \quad (26)$$

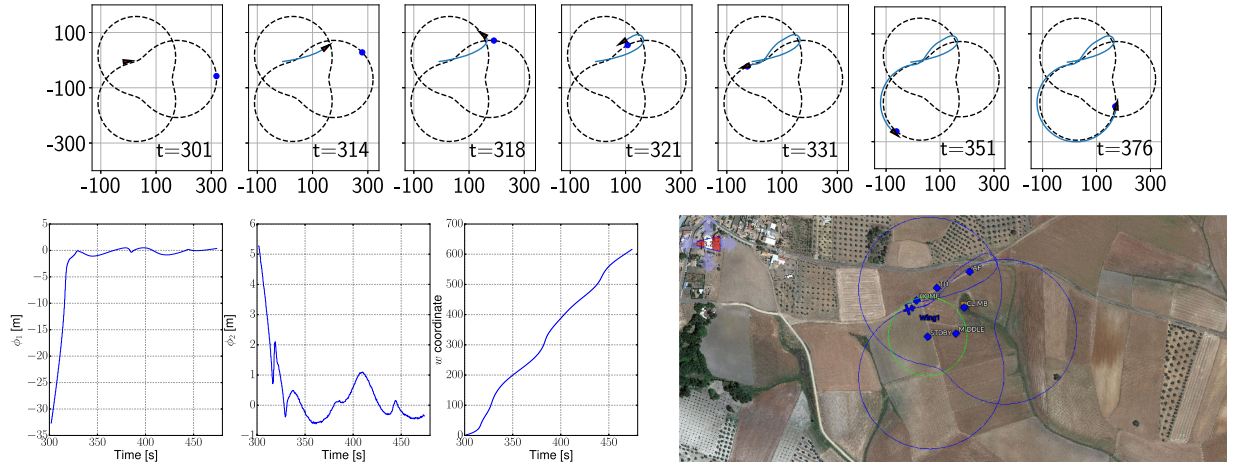


Fig. 3. Flight results of the trefoil path tracking at a constant altitude. On the low-right side, we show a screenshot of the ground control station, where the green circle denotes the *stand-by* trajectory just before starting the tracking of the trefoil. On top, we show the first minute of the flight once the tracking of the trefoil started. All the tracking is done at the constant altitude of 50 meters, and the evolution of the coordinate  $z$  is thus omitted. We can see how the blue dot, i.e.,  $(f_1(w), f_2(w))$  goes and *waits* for the aircraft at time  $t = 321$ . Afterward, the vehicle converges to the desired path as the first two low-left plots indicate with  $\phi_1$  and  $\phi_2$  staying around 0 meters. The third plot shows the evolution of the virtual coordinate  $w$ . We can see how  $w$  does not follow a constant growth, but it varies since it is in closed-loop with the position of the aircraft to facilitate the convergence to the path. Coming back to the ground control station, we can see in blue color the described 2-D trajectory of the aircraft, showing one period of the trefoil.

distance to the *desired trajectory point*  $(x_d(t), y_d(t))$ , which moves as time  $t$  advances. Note that the dynamics of the *desired trajectory point*  $(x_d(t), y_d(t))$  is *open-loop* in the sense that it does not depend on the current states of the robot, but only depends on time  $t$ . From (18), if we let  $\phi_i = 0$ ,  $i = 1, 2$ , then we may call the point  $(f_1(w(\xi(t))), f_2(w(\xi(t))))$  the *guiding point*, since it always stays on the desired path and may be regarded as the counterpart of the *desired trajectory point* in trajectory tracking algorithms. But as we will show later, the *guiding point* is essentially different from the *desired trajectory point*. Note that the *guiding point*  $(f_1(w(\xi(t))), f_2(w(\xi(t))))$  in our VF-PF algorithm depends on the evolution of the additional coordinate  $w(\xi(t))$ , of which the dynamics is state-dependent as shown in (23). This might be roughly regarded as a *closed-loop* version of the *desired trajectory point*. An intuitive consequence of this difference is that the *desired trajectory point*  $(x_d(t), y_d(t))$  in trajectory tracking algorithms always moves *unidirectionally* along the desired trajectory as  $t$  monotonically increases, while the *guiding point* can move *bidirectionally* along the desired path, subject to the current state (i.e., the robot position). In fact, when the initial position of the *guiding point*  $(f_1(w(\xi(0))), f_2(w(\xi(0))))$  is far from the initial position of the robot, the *guiding point* “proactively” moves towards the robot along the desired path to accelerate the path-following process. This feature, along with better robustness against perturbation in some cases, are experimentally studied in our previous work [33, Sec. VII]. To illustrate this closed-loop feature more intuitively, after the robot has successfully followed the desired path, we manually move the robot far away from the desired path and keep it stationary (to mimic the situation of erroneous localization and operation failure of the robot). As is clear in the supplementary video, although the robot is kept stationary, the *guiding point*  $(f_1(w(\xi(t))), f_2(w(\xi(t))))$  can still move in the reverse direction to approach the robot along the desired path

such that the path-following error decreases, and the guiding point eventually stops at some place on the desired path. After that, the guiding point does not move until the robot is released to move again.

In existing VF-PF algorithms, a 2-D vector field on  $\mathbb{R}^2$  is created for guiding the robot movement (see Remark 3). However, as we aim to create a higher dimensional (i.e., 3-D) vector field, our approach can be roughly regarded as utilizing an infinite number of layers of projected 2-D vector fields, and thus, might be seen as a *dynamic* 2-D vector field. The dynamic property is due to the dynamics of the additional coordinate  $w$ . For example, consider a circular desired path parameterized by

$$x = f_1(w) = \cos(4w) \quad y = f_2(w) = \sin(4w)$$

where  $w \in \mathbb{R}$  is the parameter. In conventional VF-PF algorithms, a 2-D vector field can be created, as shown in Fig. 1(a), but there exists a singular point at the center of the circle. Nevertheless, using our approach, we can generate a singularity-free 3-D vector field, as illustrated in Fig. 5. For clarity of visualization, we plot the 3-D vectors, which originate from three planes where the  $w$  values are 0, 0.6, and 1.4, respectively. For each value of the additional coordinate  $w$ , we can obtain a projected 2-D vector field, as shown in Fig. 6. Therefore, we can observe that these 2-D vector fields change dynamically as  $w$  varies. As a result of the dynamics of  $w$ , the *guiding point*  $(f_1(w(\xi(t))), f_2(w(\xi(t))))$  moves along the 2-D desired path (not necessarily unidirectionally). Again, we note that this point is not the same as the *desired trajectory point* in trajectory tracking algorithms since the integral curves of the 2-D vector field do not converge to this point, as can be seen graphically from Fig. 6 or analytically from the expression of the vector field in (7): the second term leads to convergence to the *guiding point*, while the first term “deviates” this convergence, since it controls the propagation along the *higher dimensional* desired path.



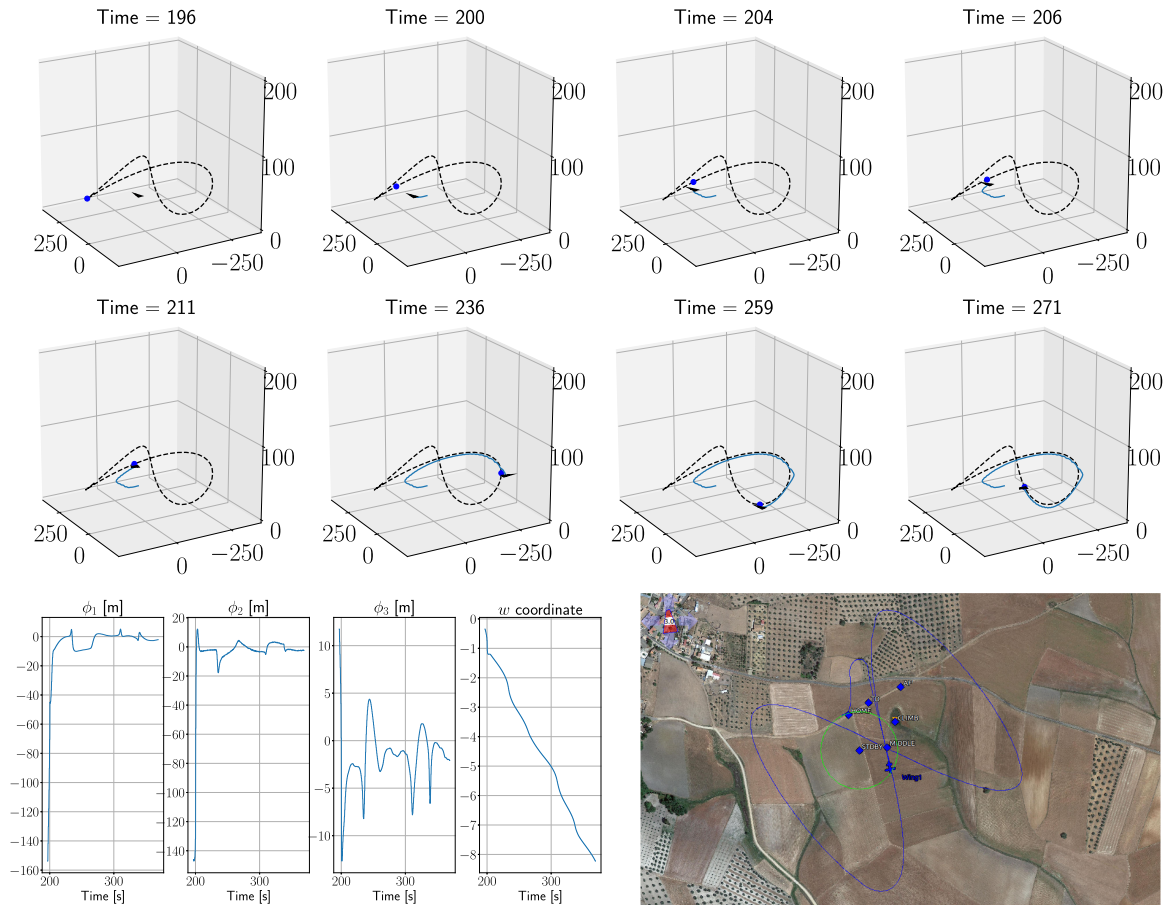


Fig. 4. Flight results of the 3-D Lissajous path tracking. On the low-right side, we show a screenshot of the ground control station, where the green circle denotes the *stand-by* trajectory just before starting the tracking of the Lissajous figure. In the 3-D plots on top, we show the first seconds of the flight once the tracking of the Lissajous figure started. We can see how the blue dot, i.e.,  $(f_1(w), f_2(w), f_3(w))$  travels quickly and *waits* for the aircraft at time  $t = 204$ . This quick traveling of  $w$  can be seen at the beginning of the fourth plot in the bottom row. Afterward, the vehicle stays on the desired path as the three bottom-left plots indicate with  $\phi_1$ ,  $\phi_2$ , and  $\phi_3$  staying around 0 meters. The aircraft has been trimmed to fly at a constant altitude. However, this path requires the vehicle to track a sinusoidal ascending/descending trajectory, and any disturbance (e.g., changing wind) makes the aircraft sensitive to track a climbing/descending speed different from zero accurately. In addition, the Lissajous curve demands aggressive turnings a bit beyond the capabilities of the aircraft, in particular, when the aircraft is descending and achieving maximum speed. We can see how  $w$  does not follow a constant growth, but it varies since it is in closed-loop with the position of the aircraft to facilitate the convergence to the path. Coming back to the ground control station, we can see in blue color the described 2-D (projected) trajectory of the aircraft, showing one period of the Lissajous figure. In particular, we can see that the aircraft passes by the *middle waypoint* corresponding to the highest point of the desired track.

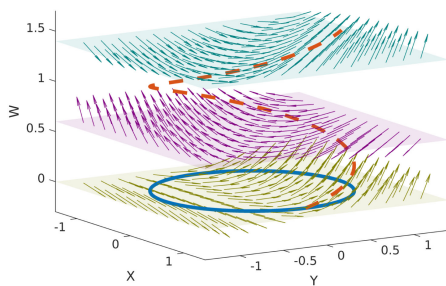


Fig. 5. Three layers of the 3-D vector field corresponding to a circle. The solid line is the 2-D desired path while the dashed line is the corresponding 3-D (unbounded) desired path. Three layers of the 3-D vector field evaluated at  $w = 0, 0.6, 1.4$ , respectively, are illustrated.

In many existing VF-PF algorithms, the desired path is usually not parameterized but is described by the intersection of (hyper)surfaces, while the latter case might be restrictive in describing more complicated desired paths. However, our

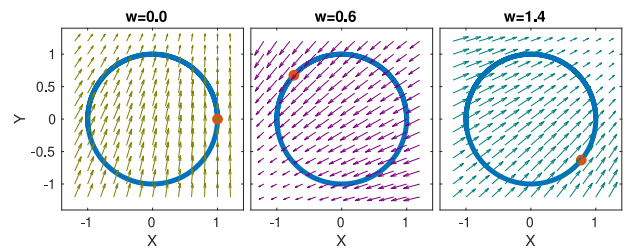


Fig. 6. Projected 2-D vector field corresponding to  $w = 0, 0.6, 1.4$ , respectively. The solid line is the projected 2-D desired path. The solid dots represent the guiding point  $(\cos(4w), \sin(4w))$ .

approach enables the possibility to use a parameterized desired path directly in the design of a higher dimensional vector field. Our approach thus extends the flexibility of conventional VF-PF algorithms. The desired path can now be described by either the intersection of (hyper)surfaces or parameterized functions. In the latter case, the parametric equations can

be easily converted using (17)–(19) and leads to a higher dimensional desired path and singularity-free guiding vector field. Theoretically, the parametrization is not instrumental, since it is only utilized to derive the expressions of functions  $\phi_i$ , of which the zero-level sets are interpreted as (hyper)surfaces. The subsequent derivation of the vector field is based on  $\phi_i$ , independent of the specific parametrization of the desired path.

## VIII. CONCLUSION

In this article, we first showed that the integral curves of a time-invariant continuously differential vector field as in (4) cannot guarantee global converge to desired paths which are simple closed (i.e., homeomorphic to the unit circle) or self-intersected. Motivated by this general topological result, we proposed a novel approach to create unbounded desired paths from simple closed or self-intersected ones, and constructed a singularity-free higher dimensional guiding vector field. One of the advantages of this approach was that global convergence to the desired paths, which could be even self-intersected, was now rigorously guaranteed. This was achieved by the introduction of a transformation operator and the extended dynamics. Another advantage was that, given a parameterized desired path, we could easily describe the (hyper)surfaces as the zero-level set of some implicit functions, and then the proposed vector field on a higher dimensional space could be directly constructed. This increased the applicability of conventional VF-PF algorithms. In addition, we highlighted five features of our approach, with rigorous theoretic guarantees. We also showed that our approach was a combined extension of both conventional VF-PF algorithms and trajectory tracking algorithms. Finally, we conducted outdoor experiments with a fixed-wing aircraft under wind perturbation to validate the theoretical results and demonstrated the practical effectiveness for complex robotic systems.

Due to the additional coordinates of the vector field, it was difficult for a robot to follow the desired path at a constant speed. This may be solved by replacing the normalization of the vector field  $\hat{\chi}$  by the “partial normalization”, i.e.,  $\tilde{\chi} := \chi / \|\chi_p\|$ , where  $\chi_p$  is the vector composed of only the first  $(n - 1)$  components of  $\chi$ . It is our future work to use the singularity-free guiding vector field to solve collision avoidance problems (preliminary results are reported in [44]), and determine the parameterization of the desired path to achieve the optimal path-following accuracy.

## ACKNOWLEDGMENT

The authors would like to thank M. Bronz and G. Hattenberger from the *Paparazzi* project for their assistance and feedback. The authors would also like to thank Y. Yu and K. Miller for the discussions.

## REFERENCES

- [1] B. Siciliano and O. Khatib, *Springer Handbook of Robotics*. Secaucus, NJ, USA: Springer-Verlag, 2007.
- [2] S. Lacroix *et al.*, “Fleets of enduring drones to probe atmospheric phenomena with clouds,” in *Proc. EGU General Assem. Conf. Abstr.*, 2016, vol. 18.
- [3] 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.
- [4] 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.
- [5] M. M. Michałek and T. Gawron, “VFO path following control with guarantees of positionally constrained transients for unicycle-like robots with constrained control input,” *J. Intell. Robot. Syst.*, vol. 89, no. 1/2, pp. 191–210, 2018.
- [6] 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.*, Sep. 2017, pp. 5298–5303.
- [7] Y. Liang and Y. Jia, “Combined vector field approach for 2D and 3D arbitrary twice differentiable curved path following with constrained UAVs,” *J. Intell. Robot. Syst.*, vol. 83, no. 1, pp. 133–160, 2016.
- [8] Y. A. Kapitanyuk, S. A. Chepinskiy, and A. A. Kapitonov, “Geometric path following control of a rigid body based on the stabilization of sets,” *IFAC Proc. Volumes*, vol. 47, no. 3, pp. 7342–7347, 2014.
- [9] K. Łakomy and M. M. Michałek, “The VFO path-following kinematic controller for robotic vehicles moving in a 3D space,” in *Proc. 11th Int. Workshop Robot Motion Control*, 2017, pp. 263–268.
- [10] M. Michałek and K. Kozłowski, “Vector-field-orientation feedback control method for a differentially driven vehicle,” *IEEE Trans. Control Syst. Technol.*, vol. 18, no. 1, pp. 45–65, Jan. 2010.
- [11] W. Caharija *et al.*, “Integral line-of-sight guidance and control of under-actuated marine vehicles: Theory, simulations, and experiments,” *IEEE Trans. Control Syst. Technol.*, vol. 24, no. 5, pp. 1623–1642, Sep. 2016.
- [12] E. Borhaug, A. Pavlov, and K. Y. Pettersen, “Integral LOS control for path following of underactuated marine surface vessels in the presence of constant ocean currents,” in *Proc. 47th IEEE Conf. Decis. Control.*, 2008, pp. 4984–4991.
- [13] W. Yao and M. Cao, “Path following control in 3D using a vector field,” *Automatica*, vol. 117, 2020, Art. no. 108957.
- [14] A. P. Aguiar, J. P. Hespanha, and P. V. Kokotović, “Performance limitations in reference tracking and path following for nonlinear systems,” *Automatica*, vol. 44, no. 3, pp. 598–610, 2008.
- [15] M. M. Seron, J. H. Braslavsky, P. V. Kokotovic, and D. Q. Mayne, “Feedback limitations in nonlinear systems: From bode integrals to cheap control,” *IEEE Trans. Autom. Control*, vol. 44, no. 4, pp. 829–833, Apr. 1999.
- [16] 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.
- [17] D. A. Lawrence, E. W. Frew, and W. J. Pisano, “Lyapunov vector fields for autonomous unmanned aircraft flight control,” *J. Guid., Control, Dyn.*, vol. 31, no. 5, pp. 1220–1229, 2008.
- [18] 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.
- [19] W. F. Phillips, *Mechanics of Flight*. Hoboken, NJ, USA: Wiley, 2004.
- [20] B. Rubí, R. Pérez, and B. Morcego, “A survey of path following control strategies for UAVs focused on quadrotors,” *J. Intell. Robot. Syst.*, vol. 98, pp. 1–25, 2019.
- [21] 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.
- [22] 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.
- [23] H. G. De Marina, Y. A. Kapitanyuk, M. Bronz, G. Hattenberger, and M. Cao, “Guidance algorithm for smooth trajectory tracking of a fixed wing UAV flying in wind flows,” in *Proc. IEEE Int. Conf. Robot. Autom.*, 2017, pp. 5740–5745.
- [24] J. Lee, *Introduction to Smooth Manifolds*, (Graduate Texts in Mathematics Series, 2nd ed), Berlin, Germany: Springer, 2015.
- [25] 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.
- [26] T. I. Fossen, M. Breivik, and R. Skjetne, “Line-of-sight path following of underactuated marine craft,” *IFAC Proc. Volumes*, vol. 36, no. 21, pp. 211–216, 2003.
- [27] S. Zhu, D. Wang, and C. B. Low, “Ground target tracking using UAV with input constraints,” *J. Intell. Robot. Syst.*, vol. 69, no. 1, pp. 417–429, Jan. 2013.

- [28] D. Soetanto, L. Lapierre, and A. Pascoal, "Adaptive, non-singular path-following control of dynamic wheeled robots," in *Proc. 42nd IEEE Conf. Decis. Control*, 2003, vol. 2, pp. 1765–1770.
- [29] A. Sgorbissa and R. Zaccaria, "3D path following with no bounds on the path curvature through surface intersection," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Systems.*, 2010, pp. 4029–4035.
- [30] M. P. Do Carmo, *Differential Geometry of Curves and Surfaces: Revised and Updated Second Edition*. New York, NY, USA: Courier Dover Publications, 2016.
- [31] J. Lee, *Introduction to Topological Manifolds*. vol. 202. Berlin, Germany: Springer-Verlag, 2010.
- [32] F. Mondada *et al.*, "The e-puck, a robot designed for education in engineering," in *Proc. 9th Conf. Auton. Robot Syst. Competitions*, 2009, pp. 59–65.
- [33] W. Yao, H. G. de Marina, and M. Cao, "Vector field guided path following control: Singularity elimination and global convergence," in *Proc. IEEE 59th Conf. Decis. Control*, 2020, *arXiv:2003.10012*.
- [34] W. Yao, H. G. de Marina, B. Lin, and M. Cao, "Singularity-free guiding vector field for robot navigation," 2020. [Online]. Available: <https://arxiv.org/abs/2012.01826>
- [35] H. Khalil, *Nonlinear Systems*, 3rd ed. Englewood Cliffs, NJ, USA: Prentice Hall, 2002.
- [36] W. Yao, B. Lin, B. D. O. Anderson, and M. Cao, "Topological analysis of vector-field guided path following on manifolds," submitted for publication. [Online]. Available: [http://tiny.cc/yao\\_topo](http://tiny.cc/yao_topo)
- [37] L. Consolini, M. Maggiore, C. Nielsen, and M. Tosques, "Path following for the PVTOL aircraft," *Automatica*, vol. 46, no. 8, pp. 1284–1296, 2010.
- [38] A. Morro, A. Sgorbissa, and R. Zaccaria, "Path following for unicycle robots with an arbitrary path curvature," *IEEE Trans. Robot.*, vol. 27, no. 5, pp. 1016–1023, Oct. 2011.
- [39] K. D. Do, "Global output-feedback path-following control of unicycle-type mobile robots: A level curve approach," *Robot. Auton. Syst.*, vol. 74, pp. 229–242, 2015.
- [40] Y.-Y. Chen and Y.-P. Tian, "A curve extension design for coordinated path following control of unicycles along given convex loops," *Int. J. Control*, vol. 84, no. 10, pp. 1729–1745, 2011.
- [41] J. B. Fraleigh and A. B. Raymond, *Linear Algebra*, 3rd ed. Reading, MA, USA: Addison-Wesley, 1995.
- [42] A. Galbis and M. Maestre, *Vector Analysis Versus Vector Calculus*. Berlin, Germany: Springer-Verlag, 2012.
- [43] W. Yao, B. Lin, B. D. O. Anderson, and M. Cao, "Refining dichotomy convergence in vector-field guided path following control," submitted for publication. [Online]. Available: [http://tiny.cc/yao\\_ecc21](http://tiny.cc/yao_ecc21)
- [44] W. Yao, B. Lin, and M. Cao, "Integrated path following and collision avoidance using a composite vector field," in *Proc. IEEE 58th Conf. Decis. Control*, 2019, pp. 250–255.
- [45] Y. A. Kapitanyuk, H. G. de Marina, A. V. Proskurnikov, and M. Cao, "Guiding vector field algorithm for a moving path following problem," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 6983–6988, 2017.
- [46] C. Chicone, *Ordinary Differential Equations With Applications*. vol. 34. Berlin, Germany: Springer-Verlag, 2006, .
- [47] T. Tao, *Analysis II, Texts and Readings in Mathematics*, vol. 38. Berlin, Germany: Springer, 2015.
- [48] W. Yao, H. G. de Marina, Z. Sun, and M. Cao, "Distributed coordinated path following using guiding vector fields," submitted for publication. [Online]. Available: [http://tiny.cc/yao\\_icra21](http://tiny.cc/yao_icra21)
- [49] B. Gati, "Open source autopilot for academic research-the Paparazzi system," in *Proc. Amer. Control Conf.*, 2013, pp. 1478–1481.
- [50] B. Siciliano, L. Sciacivico, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*. Berlin, Germany: Springer-Verlag, 2010.



**Weijia Yao** received the bachelor's and master's degrees in control science and engineering from the National University of Defense Technology, Changsha, China, in 2015 and 2017, respectively. He is currently working toward the Ph.D. degree in systems and control theory with the Engineering and Technology Institute (ENTEG), University of Groningen, Groningen, the Netherlands.

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

Mr. Yao was the recipient of the Outstanding Master Degree Dissertation award of Hunan province, China, in 2020.



**Héctor García de Marina** received the Ph.D. degree in systems and control from the University of Groningen, 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. He is currently a Researcher with the Department of Computer Architecture and Automatic Control, Faculty of Physics, Universidad Complutense de Madrid, Madrid, Spain.

His current research interests include multiagent systems and the design of guidance navigation and control systems for autonomous vehicles.



**Bohuan Lin** received the bachelor's degree in theoretical and applied mechanics and the bachelor's and master's degrees in mathematics from Sun Yat-sen University, Guangzhou, China, in 2013, 2014, and 2016, respectively. He is currently working toward the Ph.D. degree in integrable systems (Hamiltonian/non-Hamiltonian) with the Bernoulli Institute of the University of Groningen, Groningen, the Netherlands.



**Ming Cao** received the bachelor's and the 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 Systems and Control with the Engineering and Technology Institute (ENTEG), University of Groningen, Groningen, the Netherlands, where he started as a tenure-track Assistant Professor in 2008. From September 2007 to August 2008, he was a Postdoctoral Research

Associate with the Department of Mechanical and Aerospace Engineering, Princeton University, Princeton, NJ, USA. He worked as a Research Intern during the summer of 2006 with the Mathematical Sciences Department, IBM T. J. Watson Research Center, Yorktown Heights, NY, USA. His research interests include autonomous agents and multiagent systems, complex networks, and decision-making processes.

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