

Constructive Time-Varying Vector Fields for Robot Navigation

Adriano M. C. Rezende[✉], Vinicius M. Goncalves[✉], and Luciano C. A. Pimenta[✉]

Abstract—In this work, we present a methodology to compute an artificial time-varying vector field in n dimensions that defines trajectories that converge to and follow a given desired curve. The Euclidean distance function is used to construct the field, which is easily computed from a parametric representation of the curve. The computation of the time feedforward term to compensate for time dependence is such that its norm is limited by the maximum velocity of the curve. This fact allows the normalization of the time-varying vector field such that it has a constant norm without any negative effect for convergence. We present convergence proofs for the proposed normalized time-varying vector field and demonstrate the existence of ultimate bounds in the case bounded disturbances are present. Finally, we show several simulations and experiments with an actual quadrotor to validate our methodology.

Index Terms—Artificial vector fields, autonomous vehicle navigation, motion control, motion and path planning.

I. INTRODUCTION

STRATEGIES based on artificial vector fields have been widely used to guide the high-level control of several types of robots. For instance, one of the most frequent uses of such a strategy is associated with the guidance of fixed-wing unmanned aerial vehicles (UAVs) [1]–[3]. Such approaches have already been discussed in scientific books, such as [4]. Vector fields are also used to control quadcopters, [5]–[7]. Other types of robots that can be controlled with vector fields include wheeled robots [8], [9], autonomous underwater vehicles [10], manipulators [11], and even humanoid-like robots [12]–[14].

In [15], a strategy to generate time-varying vector fields for converging to and circulating a curve in n dimensions is

Manuscript received December 8, 2020; revised May 15, 2021; accepted June 15, 2021. Date of publication July 19, 2021; date of current version April 5, 2022. The work of Adriano M. C. Rezende and Luciano C. A. Pimenta was supported by Brazilian National Research Council (CNPq). This work was supported in part by the project INCT (National Institute of Science and Technology) under Grant Brazilian National Research Council (CNPq) 465755/2014-3, in part by the São Paulo Research Foundation (FAPESP) under Grant 2014/50851-0, and in part by the FAPEMIG (Minas Gerais Research Foundation) under Grant APQ-02144-18, in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) (Finance Code 88887.136349/2017-00), and in part by the CNPq under Grant 315258/2020-9. This article was recommended for publication by Associate Editor M. Schwager and Editor P. Robuffo Giordano upon evaluation of the reviewers' comments. (*Corresponding author: Adriano M. C. Rezende*)

The authors are with the Postgraduate Program in Electrical Engineering, Universidade Federal de Minas Gerais (UFMG), Belo Horizonte 31270-901, Brazil (e-mail: adrianomcr18@gmail.com; vinicius.marianog@gmail.com; lucpim@cpdee.ufmg.br).

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.2021.3093674>.

Digital Object Identifier 10.1109/TRO.2021.3093674

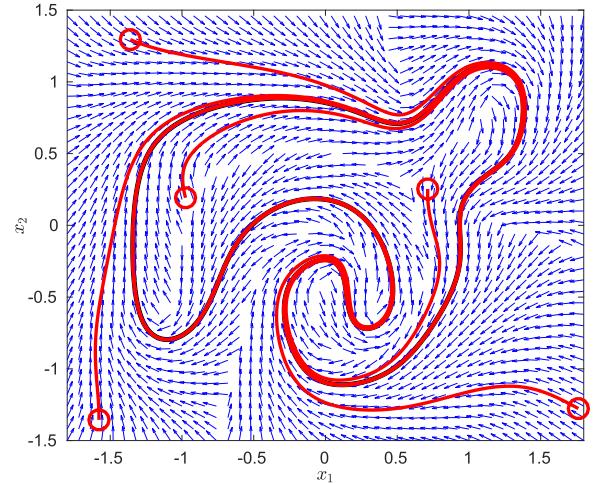


Fig. 1. Paths generated by following the vector field described in this article for a given target curve. The vector field, also shown, is devoid of any stable equilibrium points out of the target curve.

proposed. There, the curve to which the integral lines of the vector field converge is defined by the intersection of $n - 1$ zero-level sets given by $\alpha_i = 0$, $i = 1, \dots, n - 1$, in which $\alpha_i : \mathbb{R}^n \times \mathbb{R}^+ \rightarrow \mathbb{R}$. This approach has the drawback of requiring the design of scalar functions α_i that implicitly define the target curve. For simple curves, such as circles and ellipses, it is easy to obtain these functions, however, for general curves, it is not the case. Methodologies for generating these functions were presented, but the functions α_i generated are prone to induce spurious undesirable stable equilibrium points in the vector field.

In this work, we describe a simpler methodology that can be easily applied given a parametric representation of the curve, instead of using a representation as an intersection of zero-level surfaces. The main feature of this field is that it is computed by using the Euclidean distance between the robot and the curve as an error measurement function. The use of a parametric representation of the curve and the consideration of the distance function make our method simple to implement and allow for easy geometric interpretations of our results. Fig. 1 shows an example of the vector field generated by the methodology described in this article. As in [16], we also present a normalization strategy for time-varying fields, allowing for fields with a fixed norm, a useful feature for practical applications. The normalization problem is not as trivial as the normalization of static fields. This is due to the necessity of maintenance of the computed feedforward component that accounts for the movements of the

curve. For example, in [17], a constant norm vector field is obtained by neglecting the feedforward component. Although stability is obtained, asymptotic stability is lost.

Vector field approaches applied to navigation problems are usually categorized as path controllers instead of trajectory controllers. By path, we mean a unidimensional set $\mathcal{C}(t)$, possibly time-varying, embedded in \mathbb{R}^n . Following a path is a geometric task. By trajectory, we mean a time-dependent reference point $\mathbf{x}_r(t) \in \mathbb{R}^n$, as in [18]. Refer to [19] for more discussion on the differences between path and trajectory controllers. Some works that point to the advantages of a path controller over the trajectory tracking are available in the literature [20]–[22]. In [21], the authors present comparisons of trajectory and path controllers and show that path controllers usually perform better in practical situations. In fact, following a path is a less restrictive task than following a trajectory [20], in the sense the performance is not limited by the energy of the control effort. In [20], the authors demonstrate that, in trajectory tracking, the smallest achievable \mathcal{L}_2 norm of the tracking error is limited by the least amount of control energy required to stabilize the zero dynamics of the error system. This limitation is not present for the task of following a path. This fact is an additional motivation for the study of vector field based strategies.

Indeed, vector fields have been considered by several works to solve the problem of robot path control. In [23], vector fields that converge to circular loiters are developed and convergence proofs are presented with Lyapunov Theory. Fields that converge to different curve shapes are obtained through diffeomorphism. Furthermore, the work in [23] considers only planar curves, for instance, fields in 2-D or fields in 3-D converging to curves with a fixed height. In [24], the authors extend the methodology considering a team of UAVs and moving curves. In [25], a method is developed to make a group of robots converge and circulate a given beacon, which is possibly moving. However, no specific curve is defined to be circulated in this case.

The authors of [26] propose two static 3-D vector field strategies and use it to guide a UAV. The proposed field is based on the tangent vector associated with a given point in the curve and a distance measurement. At that point, their methodology is similar to the one developed in this article. Convergence proofs are also presented. Our strategy differs because we address the problem of time-varying curves and we investigate the analytical properties of the use of the Euclidean distance as a measurement of tracking error.

In [12], vector fields are built based on optimization techniques. In that case, the user can include constraints such as speed limits and obstacle avoidance into the vector field as constraints of the optimization problem. The technique can be used to drive UAVs toward the curve and also to circulate it, which is the problem discussed in this article. However, it requires the user to provide a function that codifies the target curve as a zero-level set. No constructive methodology for generating these curves is provided. The vector field was tested on a humanoid robot. The technique presented in [11] has a similar philosophy, computing the vector field by solving an optimization problem. In that case, the focus is a surgery application.

In [27], the authors develop a method to construct an artificial vector field from a parametric curve. The authors already pointed out the simplicity, and thus practicality, of the use of parametric representations over the implicit representations in [15]. In [27], contraction analysis theory is used to provide convergence proofs for time-invariant curves in generic manifolds. The method proposed in this article gives equivalent results when the manifold is \mathbb{R}^n and the used metrics is the Euclidean distance. Besides, we consider time-varying curves and disturbances in the model.

Vector fields are constructed for star-shaped curves in [28] and [16]. The method is limited to curves embedded in three dimensions. A radial Fourier basis set is used to represent the curves and convergence proofs are presented. Curves represented by a sequence of points are also considered, however, the methodology consists of an analytical approximation. In consequence, the convergence properties of the resulting field are dependent on the behavior of these functions, which can be undesirable.

The recent work in [22] presents a vector field methodology for n dimensions and static curves represented by a parametric equation. The methodology consists of an extension of the n -dimensional space with another dimension, corresponding to the parameter of the curve's representation. Then, based on [15], the authors define zero-level sets in this extended space. A feature reached by this approach is the absence of singularities in the vector field, meaning that there are no points in the space in which the field is not properly defined or is the null vector. Different from our approach, the method in [22] is not invariant to the parametrization and does not consider time-dependent curves.

Vector fields, usually designed for the simple integrator model, are considered as a high-level controller, commonly referred to as a *guidance control*. When a robot with a more complex dynamics is to be guided with a vector field, usually lower level controllers need to be considered. This cascade scheme is known as separated guidance and control (SGC), [29]. In general, these lower level controllers achieve an asymptotic convergence when the field's derivatives are taken into account, [2], [6]. The work in [7] shows that this cascade connection approach is able to control a quadcopter in high speeds. Besides, imperfections of such lower level controllers may be considered as a disturbance on the control inputs of the original integrator model.

Several works that consider the control of robots with artificial vector fields assume that the models are uncertain. This is of great interest because it is important to evaluate the application of the method in real scenarios, in which uncertainties are always present. For instance, in [1] and [2], vector fields are used to control fixed-wing UAVs and the authors show that in the presence of bounded additive disturbances in the model the robot converges to an invariant set around the desired limit cycle. In this article, we use Lyapunov theory to prove the convergence to the curve of a robot described by the simple integrator model. To account for bounded disturbances in the model, we also analyze the perturbed system. As in the previous works, we show the existence of an invariant set around the curve in this case. One interesting fact about this result is that the boundary of the invariant set is defined by a level set of the Euclidean

distance to the curve. It is more intuitive than the invariant set that can be computed by using the α functions of [15], whose size and shape will depend on the analytic properties of the α functions. Besides, the set is isotropic, meaning that the bound of the invariant set has a constant distance to the curve.

Other than the vector field approach, a common strategy to deal with path following uses the idea of a virtual target [29]–[31], a point that moves along the path and guides the design of the controller. This reference is also considered as the look-ahead point [29]. There the Frenet–Serret frames are used, and in this sense, their methodology is similar to ours, since we also use the tangent component of the path to compute the traversal component. Another similarity is that they consider the closest point map (projection) to define the point on the path from which the Frenet–Serret frame will be computed. However, these works are limited to three dimensions [32] and convergence proofs are only available for constant-curvature planar paths. Another advantage of the controller proposed in this article is that it does not have problems when the robot is far from the curve. As pointed out in [33], the Frenet–Serret-based orthogonal projection methods are only valid locally.

The methodology presented here may be considered as an improvement of the work in [15]. Although they both, basically, consist of three components, associated with convergence, circulation, and time feedforward, the current approach has the advantages: simplicity on the representation of the curves; homogeneous convergence pattern, which comes from the use of the Euclidean distance; possibility to generate constant norm vector fields; and absence of undesirable equilibrium points. The use of the Euclidean distance function for generating time-varying vector fields provides pure geometric interpretations of our results. The normalization method, also considered in [16], allows the obtainment of constant norm time-varying fields that converge asymptotically to the desired curve. Our method can also be easily applied to curves represented by sampled points. If a large and smooth sequence of points is available, the method can be easily applied numerically. If only a few way-points are available, the proposed strategy can be applied after a simple polynomial interpolation. Polynomials have been widely used in robotics given their simplicity [34]. For instance, optimum polynomial trajectories for quadcopters are obtained in [35]. Another simple method to consider is the Hermite interpolation used in [36]. Furthermore, two important contributions of this work are as follows: (i) an analysis of the influence of bounded model disturbances, demonstrating robustness in real robot applications; and (ii) a mathematical analysis of the approachability of undesired singularity points for the time-invariant case. Our theory is validated with several simulations that consider: an integrator model; a drone; and a 6 degree of freedom (DOF) manipulator. Moreover, we present outdoor experiments with a physical drone.

II. PROBLEM SETUP

Henceforth, we assume that all the vectors are column vectors. All vector quantities are denoted by a bold symbol, as \mathbf{x} . The

symbol $\|\cdot\|$ denotes the Euclidean norm of a n -dimensional vector. If M is a matrix (or vector), M^T denotes its transpose.

Consider a time-varying curve $\mathcal{C}(t) \subset \mathbb{R}^n, n > 1$. Let $\mathbf{r}(s, t) : \mathbb{R} \times \mathbb{R}^+ \rightarrow \mathbb{R}^n$ be a parametrization for the curve $\mathcal{C}(t)$, in which s is the arc-length¹ parameter and t is time. Let $\mathbf{x} \in \mathbb{R}^n$ be a point. There are many possibilities for the curve $\mathcal{C}(t)$. It could be, for instance, a finite straight line, or two disjoint circles. In order to clarify which kind of curves we will consider in this article, we will use the following assumption.

Assumption 1: We assume that the curve $\mathcal{C}(t)$ is such that, for all fixed t , it is either an unbounded curve homeomorphic to an infinite straight line (“open curve”) or bounded and homeomorphic to a circle (“closed curve”). ■

The objective is to compute a time-varying vector field $\Psi(\mathbf{x}, t) : \mathbb{R}^n \times \mathbb{R}^+ \rightarrow \mathbb{R}^n$ such that the trajectories of the system $\dot{\mathbf{x}} = \Psi(\mathbf{x}, t)$ converge to and follow the target curve $\mathcal{C}(t)$. The computation of Ψ will be made with basis on the parametric representation of the curve and on the distance to the curve. With this in mind, we then define important functions for our method.

Definition 1: We define the parameters of the closest points on the curve $s^*(\mathbf{x}, t) : \mathbb{R}^n \times \mathbb{R}^+ \rightarrow 2^{\mathbb{R}}$ as

$$s^*(\mathbf{x}, t) = \arg \min_s \|\mathbf{x} - \mathbf{r}(s, t)\|^2. \quad (1)$$

And from it, the *closest points map*, or closest points projection, $\mathbf{x}^*(\mathbf{x}, t) : \mathbb{R}^n \times \mathbb{R}^+ \rightarrow 2^{\mathbb{R}^n}$ as

$$\mathbf{x}^*(\mathbf{x}, t) = \{\mathbf{r}(s, t) \mid s \in s^*(\mathbf{x}, t)\}. \quad (2)$$

Finally, the *distance vectors* $\mathbf{D}(\mathbf{x}, t) : \mathbb{R}^n \times \mathbb{R}^+ \rightarrow 2^{\mathbb{R}^n}$ and the *scalar distance* $D(\mathbf{x}, t) : \mathbb{R}^n \times \mathbb{R}^+ \rightarrow \mathbb{R}^+$ as

$$\mathbf{D}(\mathbf{x}, t) = \{\mathbf{x} - \mathbf{y} \mid \mathbf{y} \in \mathbf{x}^*(\mathbf{x}, t)\} \quad (3)$$

$$D(\mathbf{x}, t) = \|\mathbf{x} - \mathbf{y}\| \text{ for any } \mathbf{y} \in \mathbf{x}^*(\mathbf{x}, t). \quad (4)$$

We will omit the dependence on the variable t in all these functions when the curve is not time-varying. ■

It is important to observe that, for some points (like the center of a circle), the function s^* (and therefore \mathbf{D} and \mathbf{x}^*) is multivalued, there may be more than one point in the curve that has the smallest distance from the point to the curve. Thus, the set s^* contains all the parameters equivalent to the points on $\mathcal{C}(t)$ that are the closest to \mathbf{x} . Each element of the set of vectors $\mathbf{D}(\mathbf{x}, t)$ goes from one of the closest points on $\mathcal{C}(t)$ to the position \mathbf{x} , and the norm of all of them is the scalar distance function $D(\mathbf{x}, t)$. Note that, despite the utilization of a parametric representation of the curve, the functions \mathbf{D} and D are independent of the parametrization used. In fact, the function $\mathbf{x}^*(\mathbf{x}, t)$ is completely defined by purely geometric characteristics of $\mathcal{C}(t)$. Evidently, $D(\mathbf{x}, t)$ and $\mathbf{D}(\mathbf{x}, t)$ are also pure geometric functions. The only dependence the field $\Psi(\mathbf{x}, t)$ will have on the parametrization $r(s, t)$ is on the direction that the curve will be followed.

In the points in which $s^*, \mathbf{x}^*, \mathbf{D}$ are multivalued, our proposed vector field will not be defined. Regarding this aspect, the following definition is required.

¹The use of the arc-length parameter is only for convenience, since it will simplify our mathematical analysis. The methodology does not require this special representation to be implemented.

Definition 2: Let $\mathcal{U} \subseteq \mathbb{R}^n \times \mathbb{R}^+$ be the set of points (\mathbf{x}, t) such that $s^*(\mathbf{x}, t)$ is a singleton (set with only one element). ■

Thus, for $(\mathbf{x}, t) \in \mathcal{U}$, we have that $\mathbf{x}^*(\mathbf{x}, t)$ and $\mathbf{D}(\mathbf{x}, t)$ are also singletons. Henceforth, we will assume that the domain of the function Ψ is \mathcal{U} , and thus, it will be single-valued as well.

The vector field will not be defined for points outside of \mathcal{U} (discontinuity points). We will soon establish that there are few such points, i.e., for any fixed t the set of points \mathbf{x} such that $(\mathbf{x}, t) \notin \mathcal{U}$ has measure zero in \mathbb{R}^n . It will be also mathematically shown that, at least for the time-invariant case, these points are unapproachable if we choose the parameters of the vector field accordingly. Anyway, in the time-variant case or in the presence of disturbances, we can simply choose one of the possible vectors. For instance, the one which is closest to the last velocity vector sent to the robot.

We also need another definition regarding a traversal term, which will be mostly responsible for the curve following.

Definition 3: The curve's tangent vector $\mathbf{T}(s, t) : \mathbb{R} \times \mathbb{R}^+ \rightarrow \mathbb{R}^n$ is defined as

$$\mathbf{T}(s, t) = \frac{\partial \mathbf{r}(s, t)}{\partial s}. \quad (5)$$

The closest point's tangent $\mathbf{T}^*(\mathbf{x}, t) : \mathcal{U} \rightarrow \mathbb{R}^n$ is defined as

$$\mathbf{T}^*(\mathbf{x}, t) = \mathbf{T}(s^*(\mathbf{x}, t), t). \quad (6)$$

We will omit the dependence on the variable t in \mathbf{T}^* when the curve is not time-varying. ■

Note that, unlike s^* , \mathbf{x}^* , and \mathbf{D} , the vector $\mathbf{T}^*(s^*(\mathbf{x}, t), t)$ is not multivalued. Also, since s is the arc length, it holds that $\|\mathbf{T}(s, t)\| = 1$. In practical cases, if one uses a parametrization that is not the arc-length, to obtain the field we aim, the tangent vector must be normalized. In fact, the definition of the normalized tangent vector does not depend on the parametrization either, except for a factor of -1 that may invert the direction in which the curve is traversed. This vector can be seen as the unit vector that spans the tangent space of the set $\mathcal{C}(t)$. A vector field similar to the one presented here could be designed without the normalization of the tangent vector. However, it would not be invariant to the curve parametrization, since the intensity of the traversal component will be tied to the norm of such a vector.

An additional assumption is necessary to establish that the the curve $\mathcal{C}(t)$ is smooth and the tangent vector is well defined.

Assumption 2: We assume that the function $\mathbf{r}(s, t)$ is twice differentiable on s for all t . Furthermore, we assume that $\mathbf{r}(s, t)$ is also differentiable on t , for all fixed s . ■

We may also require that the vector field has a constant norm given by v_r , i.e., $\|\Psi(\mathbf{x}, t)\| = v_r \forall (\mathbf{x}, t) \in \mathcal{U}$. In order to enable that, we must assume that the curve $\mathcal{C}(t)$ is slow-varying on time when compared with the robot speed v_r . For this purpose, we will need another definition.

Definition 4: We define the null space of \mathbf{T}^* projection, $\Pi_{\mathbf{T}}(\mathbf{x}, t) : \mathcal{U} \rightarrow \mathbb{R}^{n \times n}$, as the matrix

$$\Pi_{\mathbf{T}}(\mathbf{x}, t) = I_{n \times n} - \mathbf{T}^*(\mathbf{x}, t)\mathbf{T}^*(\mathbf{x}, t)^T \quad (7)$$

in which $I_{n \times n}$ is the identity matrix of order n . ■

This null-space projection matrix is such that $\Pi_{\mathbf{T}}(\mathbf{x}, t)\mathbf{h}$ is in the (right) null space of $\mathbf{T}^*(\mathbf{x}, t)^T$ for all \mathbf{h} . In other words,

$\Pi_{\mathbf{T}}(\mathbf{x}, t)\mathbf{h}$ removes the component $\mathbf{T}^*(\mathbf{x}, t)$ of the vector \mathbf{h} . With this definition, we can formally assume the following.

Assumption 3: For any point $\mathbf{x} \in \mathbb{R}^n$ at any time $t \in \mathbb{R}^+$ such that $(\mathbf{x}, t) \in \mathcal{U}$, the partial derivative of the distance vector $\mathbf{D}(\mathbf{x}, t)$, defined in (3), with respect to time is finite. We will impose that when we remove the traversal component from this vector the resulting vector has norm smaller than v_r . Formally

$$\max_{(\mathbf{x}, t) \in \mathcal{U}} \left\| \Pi_{\mathbf{T}}(\mathbf{x}, t) \frac{\partial \mathbf{D}}{\partial t}(\mathbf{x}, t) \right\| \equiv v_m < v_r \quad (8)$$

in which v_m is called the maximum “local speed” of $\mathcal{C}(t)$. ■

The intuition behind Assumption 3 is that $\frac{\partial \mathbf{D}}{\partial t}$ can be considered as the velocity for a point \mathbf{x} in the curve at time t to continue in the curve at time $t + \Delta t$. However, any component of this velocity on the tangent of the curve just causes an internal movement. For instance, if the time-varying curve is a circle moving in space, $\frac{\partial \mathbf{D}}{\partial t}$ makes the point move with the curve from $\mathcal{C}(t)$ to $\mathcal{C}(t + \Delta t)$, but the tangent component of this velocity only makes the point rotate inside $\mathcal{C}(t + \Delta t)$. Therefore, we can remove this component.

Note that Assumption 3 is weak, since it is reasonable that a robot with speed v_r cannot track a curve that moves with a speed greater than v_r . In real scenarios, it is also important to emphasize that the curve $\mathcal{C}(t)$ must be feasible to be traversed by the robot at the desired speed v_r . For instance, curves with high curvatures may not be tracked with high velocities as real robots might be subject to mechanical constraints, such as maximum angular velocity.

Essentially, the problem addressed in this article is as follows.

Problem 1: Given a time-varying curve $\mathcal{C}(t)$ embedded in an n -dimensional Euclidean space, under Assumptions 1 and 3, defined in the parametric form $\mathbf{r}(s, t)$, under Assumption 2, find a vector field $\Psi(\mathbf{x}, t) : \mathcal{U} \rightarrow \mathbb{R}^n$ such that $\|\Psi\| = v_r$ and the trajectories of the system $\dot{\mathbf{x}} = \Psi(\mathbf{x}, t)$ converge to and traverse (follow, circulate) $\mathcal{C}(t)$. ■

III. METHODOLOGY

In this section, we will state the solution for Problem 1 defined in the previous section, announce its properties and then proceed to establish them formally. Before stating our solution, two definitions are necessary.

Definition 5: A function $G : \mathbb{R}^+ \rightarrow [0, 1]$ is said to be of G -type if it is Lipschitz continuous, strictly increasing, and $G(0) = 0$, thus, $G(D) > 0$ for $D > 0$. ■

Some simple examples of G -type functions include $G(D) = (2/\pi) \arctan(k_f D)$, for $k_f > 0$ and $G(D) = D/\sqrt{D^2 + k_f^2}$ also for $k_f > 0$.

Definition 6: Let G be a G -type function, we define $H(D) = \sqrt{1 - G(D)^2}$. Henceforth, we will use the definitions

$$\hat{G}(\mathbf{x}, t) \equiv G(D(\mathbf{x}, t)), \quad \hat{H}(\mathbf{x}, t) \equiv H(D(\mathbf{x}, t)). \quad (9)$$

We will omit the dependence on the variable t in \hat{G} and \hat{H} when the curve is not time-varying. ■

Note that $G, H : \mathbb{R}^+ \rightarrow \mathbb{R}^+$, while $\hat{G}, \hat{H} : \mathbb{R}^n \times \mathbb{R}^+ \rightarrow \mathbb{R}^+$. Given these definitions, a solution for Problem 1 is the vector

field $\Psi(\mathbf{x}, t)$ defined by the following components:

$$\begin{aligned}\Psi(\mathbf{x}, t) = & -\eta(\mathbf{x}, t)\hat{G}(\mathbf{x}, t)\frac{\mathbf{D}(\mathbf{x}, t)}{D(\mathbf{x}, t)} \quad (\text{convergence}) \\ & + \eta(\mathbf{x}, t)\hat{H}(\mathbf{x}, t)\mathbf{T}^*(\mathbf{x}, t) \quad (\text{traversal}) \\ & - \Pi_{\mathbf{T}}(\mathbf{x}, t)\frac{\partial \mathbf{D}}{\partial t}(\mathbf{x}, t) \quad (\text{feedforward})\end{aligned}\quad (10)$$

in which $\eta(\mathbf{x}, t) : \mathcal{U} \rightarrow \mathbb{R}$ is a positive scalar computed so that Ψ has a norm equal to v_r .

Note that, since \mathbf{D} and \mathbf{T} are purely geometric functions, the vector field is also purely geometric. The curve's parametrization function is only a convenient tool to enable the computation of Ψ .

Likewise [15], the vector field has three components, each one responsible for a different behavior. The functions \hat{G} and \hat{H} are responsible for tuning the relative predominance of the convergent or traversal components, while η defines the norm of the vector field. Note, however, that the computation of the terms is much more straightforward in comparison to previous works that consider implicit definitions for the target curve, requiring no knowledge of functions α_i .

This vector field has the following features that will be formally shown in the following sections.

- 1) It converges asymptotically to $\mathcal{C}(t)$ for each initial condition $(\mathbf{x}_0, 0) \in \mathcal{U}$. There are no other equilibrium points, stable or unstable (Proposition 2).
- 2) Under Assumption 3, η can always be chosen so that the time-varying vector field has a constant norm: $\|\Psi(\mathbf{x}, t)\| = v_r$ (Lemma 6).
- 3) Points outside of \mathcal{U} form a set of measure zero (Proposition 1). Furthermore, for the normalized time-invariant case, one can choose the parameter \hat{G} such that no points outside of \mathcal{U} are approachable (Proposition 4). Nevertheless, even if these points are reached, which can happen in the case of time-varying curves or if there are disturbances on the system, one can simply choose one of the possible values of $\Psi(\mathbf{x}, t)$ and continue moving along the field. For instance, a natural choice is the closest to the last vector that has been sent to the robot to follow.
- 4) It is robust to norm-bounded additive disturbances in the assumed simple integrator model. In this case, the maximum deviation from the curve is finite and can be computed from the bound of the disturbance (Proposition 3).
- 5) Its convergence behavior is isotropic on space. That means that the weight of the convergent component does not depend on the direction of the vector \mathbf{D} , but only on the distance D (also Proposition 3).

A. Distance function properties

The Euclidean distance function defined in (4) has interesting and useful properties. In this section, they will be presented. We will begin by showing that the set \mathcal{U} is “small.” For this purpose, we begin with a Lemma.

Lemma 1: Consider a point \mathbf{x} , let \mathbf{x}^* be one of the possible closest points on the curve \mathcal{C} to the point \mathbf{x} . Let $0 < \alpha < 1$. Then,

the closest point to the curve of the point $\mathbf{x}_{mid} = (1 - \alpha)\mathbf{x}^* + \alpha\mathbf{x}$ is also \mathbf{x}^* . Furthermore, this point is unique: there is no other point on the curve that has the same distance to \mathbf{x}_{mid} .

Proof: Suppose there is another point on the curve, $\mathbf{x}_{other}^* \neq \mathbf{x}^*$ that is closer to \mathbf{x}_{mid} (that is, \mathbf{x}^* is not the closest point at the curve to \mathbf{x}_{mid}) or has the same distance to \mathbf{x}_{mid} as \mathbf{x}^* (that is, \mathbf{x}^* is not unique). It will be established that this induces a contradiction, and therefore, it is false.

Suppose it is true. Let $\mathbf{u} = \mathbf{x}^* - \mathbf{x}_{mid}$ and $\mathbf{v} = \mathbf{x}_{other}^* - \mathbf{x}^*$. Then, $\|\mathbf{u} + \mathbf{v}\| \leq \|\mathbf{u}\|$. This implies after squaring that (i) $2\mathbf{u}^T \mathbf{v} + \|\mathbf{v}\|^2 \leq 0$, and also that (ii) $\mathbf{u}^T \mathbf{v} < 0$, because $\mathbf{v} \neq \mathbf{0}$ since $\mathbf{x}_{other}^* \neq \mathbf{x}^*$.

Since $0 < \alpha < 1$, we have from (i) and (ii) that $\|\mathbf{v}\|^2 + 2\mathbf{u}^T \mathbf{v}/\alpha < 0$. Summing $\|\mathbf{u}\|^2/\alpha^2$ in both sides it is possible to conclude that (iii) $\|\mathbf{u}/\alpha + \mathbf{v}\| < \|\mathbf{u}\|/\alpha$.

Finally, note that $(\mathbf{x}^* - \mathbf{x}) = \mathbf{u}/\alpha + \mathbf{v}$. Furthermore, that $(\mathbf{x}_{other}^* - \mathbf{x}) = \mathbf{u}/\alpha + \mathbf{v}$. Therefore, (iii) implies that $\|\mathbf{x}_{other}^* - \mathbf{x}\| < \|\mathbf{x}^* - \mathbf{x}\|$, which is contradictory with the fact that \mathbf{x}^* is the closest point at the curve to \mathbf{x} . ■

Now, it is possible to prove that the set is indeed “small.”

Proposition 1: For any curve in \mathbb{R}^n , the set of points that have more than one closest point on the curve has measure zero in \mathbb{R}^n .

Proof: The proof comes from contradiction. Suppose there is a set \mathcal{V} of equidistant points that has a nonzero measure in \mathbb{R}^n . Therefore, it is possible to consider a ball $\mathcal{B} \subseteq \mathcal{V}$ with nonzero volume. Let \mathbf{x} be the center of such a ball, which is also a point with more than one point closest to the curve. Let \mathbf{x}^* be one of such points.

Choose an $0 < \alpha < 1$ such that the point $\mathbf{x}_{mid} = (1 - \alpha)\mathbf{x}^* + \alpha\mathbf{x}$ lies in \mathcal{B} . This is always possible because \mathcal{B} has a nonzero measure. Since \mathbf{x}_{mid} lies in \mathcal{B} , it must also have more than one point closest to the curve. However, Lemma 1 forbids it. ■

Even if this set is small, there is the possibility that the trajectory indeed passes through it. We will establish further in this article sufficient conditions for the time-invariant case that guarantee that this never happens.

We will now derive other properties of the distance function, and its associated functions, which will be important to prove the convergence of the vector field.

Lemma 2: If $s^*(\mathbf{x}, t)$ is defined by (1), the following identity holds $\forall (\mathbf{x}, t) \in \mathcal{U}$

$$\mathbf{T}^*(\mathbf{x}, t)^T \mathbf{D}(\mathbf{x}, t) = \mathbf{T}(s^*(\mathbf{x}, t), t)^T \mathbf{D}(\mathbf{x}, t) = 0. \quad (11)$$

Proof: First, note that, given Assumption 1, the curve $\mathcal{C}(t)$ has no borders. Thus, we can apply the first-order optimality condition to the optimization problem in (1) to conclude that

$$\frac{\partial \mathbf{r}}{\partial s}(s^*(\mathbf{x}, t), t)^T (\mathbf{x} - \mathbf{r}(s^*(\mathbf{x}, t), t)) = 0. \quad (12)$$

The first factor of the dot product in (12) is the tangent vector \mathbf{T} defined in (5). The second factor in (12) is the distance vector $\mathbf{D}(\mathbf{x}, t)$. Using these facts and assuming $(\mathbf{x}, t) \in \mathcal{U}$ into (12), (11) is established. ■

Lemma 3: For each $(\mathbf{x}, t) \in \mathcal{U}$ with $\mathbf{x} \notin \mathcal{C}(t)$, $\nabla D(\mathbf{x}, t)$ is a unit norm vector and is given by $\nabla D(\mathbf{x}, t) = \mathbf{D}(\mathbf{x}, t)/D(\mathbf{x}, t)$.

Proof: For the sake of notation simplicity, we will prove the result for the time-invariant case. Since the gradient is taken in respect to \mathbf{x} , while t is fixed, the proof for the case that the curve is time-variant readily follows: the argument t will play no role.

From the Mises Theorem presented in [37], we have that, for $\mathbf{x} \in \mathbb{R}^n \setminus \mathcal{C}$ the directional derivative $\nabla_{\mathbf{d}} D$ of $D(\mathbf{x})$ on the direction \mathbf{d} is given by

$$\nabla_{\mathbf{d}} D = \inf \left\{ \mathbf{d}^T \left(\frac{\mathbf{x} - \mathbf{y}}{\|\mathbf{x} - \mathbf{y}\|} \right), \mathbf{y} \in \mathbf{x}^*(\mathbf{x}) \right\}. \quad (13)$$

If $\mathbf{x} \in \mathcal{U}$, we have, by definition that $\mathbf{x}^*(\mathbf{x})$ is single-valued, thus, we can write

$$\nabla_{\mathbf{d}} D = \mathbf{d}^T \left(\frac{\mathbf{x} - \mathbf{x}^*(\mathbf{x})}{\|\mathbf{x} - \mathbf{x}^*(\mathbf{x})\|} \right) = \mathbf{d}^T \frac{\mathbf{D}}{D}. \quad (14)$$

We know that $\nabla_{\mathbf{d}} D = \mathbf{d}^T \nabla D$, thus, as a corollary of the Mises theorem, from (14), we have that, $\nabla D = \mathbf{D}/D$. ■

Lemma 4: Let $P : \mathbb{R}^+ \rightarrow \mathbb{R}^+$ be defined by $P = \frac{1}{2} D^2$. For all $(\mathbf{x}, t) \in \mathcal{U}$ the following identity holds:

$$\frac{\partial P}{\partial t} = \nabla P^T \Pi_T \frac{\partial \mathbf{D}}{\partial t}. \quad (15)$$

Proof: First, note that with the chain rule we obtain $\nabla P = D \nabla D$. Using Lemma 3 and noting that ∇P is also defined when $D = 0$, we conclude that

$$\nabla P(\mathbf{x}, t) = \mathbf{D}(\mathbf{x}, t). \quad (16)$$

Now, note that given the definition of Π_T in (7) and that $\nabla P \parallel \mathbf{D} \perp \mathbf{T}$, we have that (15) reduces to

$$\frac{\partial P}{\partial t} = \nabla P^T \frac{\partial \mathbf{D}}{\partial t}. \quad (17)$$

Now, taking the partial derivative of the identity $P = \frac{1}{2} \mathbf{D}^T \mathbf{D}$ with respect to time

$$\frac{\partial P}{\partial t} = \frac{1}{2} \frac{\partial \mathbf{D}^T}{\partial t} \mathbf{D} + \frac{1}{2} \frac{\partial \mathbf{D}^T}{\partial t} \mathbf{D} = \mathbf{D}^T \frac{\partial \mathbf{D}}{\partial t}. \quad (18)$$

Using (16) in (18), we obtain (17), thus, (15) is established. ■

B. Normalized Vector Field Methodology

Let us now discuss the field's three individual components: *convergence*, *traversal*, and *feedforward* terms.

First, a field that converges to the curve $\mathcal{C}(t)$ is sought. Therefore, the *convergent component* $\Psi_G(\mathbf{x}, t) : \mathcal{U} \rightarrow \mathbb{R}^n$ is defined as

$$\Psi_G(\mathbf{x}, t) = -\hat{G}(\mathbf{x}, t) \frac{\nabla P(\mathbf{x}, t)}{\|\nabla P(\mathbf{x}, t)\|} = -\hat{G}(\mathbf{x}, t) \nabla D(\mathbf{x}, t) \quad (19)$$

in which $\hat{G}(\mathbf{x}, t) \equiv G(D(\mathbf{x}, t))$, as in Definition 6, and the second equality is obtained from (16) and Lemma 3. When $D = 0$, the vector ∇D is not defined. However, $\lim_{\mathbf{x} \rightarrow \mathcal{C}(t)} \hat{G}(\mathbf{x}, t) \nabla D(\mathbf{x}, t) = \mathbf{0}$, since $G(0) = 0$ and $\lim_{\mathbf{x} \rightarrow \mathcal{C}(t)} \|\nabla D(\mathbf{x}, t)\| = 1$. Formally, we can define that $\Psi_G = \mathbf{0}$ when $D = 0$. The intuition behind this term is that it always points to the curve, more precisely to the closest point at the curve. Therefore, it guides the point \mathbf{x} toward $\mathcal{C}(t)$.

Now, a field that provides the tracking of the curve is sought. Therefore, the traversal component $\Psi_H(\mathbf{x}, t) : \mathcal{U} \rightarrow \mathbb{R}^n$ is defined as

$$\Psi_H(\mathbf{x}, t) = \hat{H}(\mathbf{x}, t) \mathbf{T}^*(\mathbf{x}, t) \quad (20)$$

in which $\mathbf{T}^*(\mathbf{x}, t)$ is the tangent vector computed according to (6) and \hat{H} is defined as in Definition 6. The intuition behind this component is that, when we are in $\mathcal{C}(t)$ at a point \mathbf{x} , this component is always tangent to the curve. Therefore, it induces \mathbf{x} to follow the curve.

The *time feedforward component* $\Psi_T(\mathbf{x}, t) : \mathcal{U} \rightarrow \mathbb{R}^n$, important in time-dependent scenarios, is defined as

$$\Psi_T(\mathbf{x}, t) = -\Pi_T(\mathbf{x}, t) \frac{\partial \mathbf{D}}{\partial t}(\mathbf{x}, t). \quad (21)$$

It is not as easy to have an intuition for this term, but we will see mathematically that it is necessary to ensure convergence. Furthermore, it is clearly related to time-varying properties of the curve, since for static curves it vanishes.

Fig. 2 illustrates these three components. Except for the norm requirement, $\|\Psi\| = v_r$, we could sum all these components and obtain a field that solves Problem 1.

If it is desirable to have a constant speed, v_r , for time-invariant vector fields, it is usual to normalize the vector field such that a vector Ψ has the same norm for all \mathbf{x} . However, time-varying fields cannot be as easily normalized without compromising convergence. This happens because if we multiply the component Ψ_T by a scaling factor, the vector field will not properly compensate the movement of the curve, thus, the convergence will be affected. We will consider a different normalization method [16]. It will scale only part of the vector field to obtain a constant norm of the composite vector. The component associated with the changes on time of the curve will remain unchanged.

We will divide the three components in two groups. The first one, Ψ_S , composed of Ψ_G and Ψ_H , is not dependent on partial derivatives with respect to time, thus, will be the scalable component. The second one, composed only of Ψ_T , is dependent on the partial derivatives with respect to time, thus, it is called the not scalable component.

We propose the following field:

$$\Psi = \eta \Psi_S + \Psi_T \quad (22)$$

in which η is a scaling factor that will be determined soon.

A Lemma is necessary.

Lemma 5: For all $(\mathbf{x}, t) \in \mathcal{U}$, the vector $\Psi_S = \Psi_G + \Psi_H$ has a unit norm.

Proof: From Lemma 2, $\Psi_G^T \Psi_H = -\hat{G} \hat{H} \mathbf{D}^T \mathbf{T}^*/D = 0$. Thus, $\|\Psi_G + \Psi_H\|^2 = \|\Psi_G\|^2 + \|\Psi_H\|^2$. Now, since $\|\mathbf{T}^*\| = 1$ and $\|\mathbf{D}\| = D$, we have that $\|\Psi_G + \Psi_H\|^2 = \hat{G}^2 + \hat{H}^2$. From Definition 6, we have that $\hat{G}^2 + \hat{H}^2 = 1$. ■

Now, using the fact $\|\Psi_S\| = 1$ (Lemma 5), we can see that condition $\|\Psi\|^2 = v_r^2$ reduces to

$$\eta^2 + (2\Psi_S^T \Psi_T) \eta + (\|\Psi_T\|^2 - v_r^2) = 0. \quad (23)$$

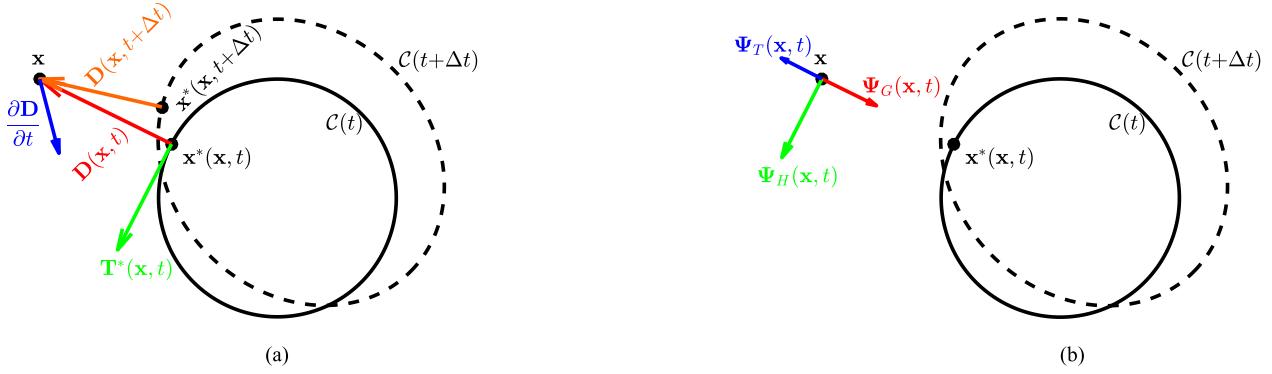


Fig. 2. Example showing the components of Ψ . (a) Some elements necessary to compute the components. (b) Three components.

As will be clear later, in order to ensure $\eta > 0$, the following root of the second-order polynomial in (23) should be considered

$$\eta = -\Psi_S^T \Psi_T + \sqrt{(\Psi_S^T \Psi_T)^2 + v_r^2 - \|\Psi_T\|^2}. \quad (24)$$

Note that $\|\Psi_T\| < v_r$, due to Assumption 3, ensures that η is real.

The normalized vector field Ψ can then be written as

$$\Psi = -\eta G \nabla D + \eta H \mathbf{T}^* - \Pi_T \frac{\partial \mathbf{D}}{\partial t}. \quad (25)$$

Asymptotic convergence for this normalized field will be established, but first, it is necessary to show two lemmas.

Lemma 6: Given Assumption 3, the value of η in (24) is such that

$$0 < v_r - v_m \leq \eta \leq v_r + v_m. \quad (26)$$

Proof: From the fact that $\|\Psi_T\| \leq v_r$, it is clear that $\eta \geq 0$. In addition, from $\|\eta \Psi_S + \Psi_T\| = v_r$, $\|\Psi_S\| = 1$, and $\|\Psi_T\| \leq v_m$ it can be inferred, by the triangle inequality $\|\mathbf{a} + \mathbf{b}\| \leq \|\mathbf{a}\| + \|\mathbf{b}\|$ that

$$v_r = \|\eta \Psi_S + \Psi_T\| \leq \eta \|\Psi_S\| + \|\Psi_T\| \leq \eta + v_m \quad (27)$$

from which $\eta \geq v_r - v_m$ is obtained. Furthermore, using the same information and from the reverse triangle inequality, $\|\mathbf{a} + \mathbf{b}\| \geq \|\mathbf{a}\| - \|\mathbf{b}\|$, we have

$$v_r = \|\eta \Psi_S + \Psi_T\| \geq \eta \|\Psi_S\| - \|\Psi_T\| \geq \eta - v_m \quad (28)$$

from which $\eta \leq v_r + v_m$ is obtained. \blacksquare

Lemma 7: Let $z : \mathbb{R} \rightarrow \mathbb{R}$, a constant $\beta > 0$, and a G -type function G such that the differential equation $\dot{z}(t) = -\beta G(z(t))$ holds, for any initial condition $z(0) = z_0 \geq 0$. Then, $\lim_{t \rightarrow \infty} z(t) = 0$.

Proof: Use the Lyapunov function $V(z) = \frac{z^2}{2}$. We then have that $\dot{V} = -\beta z G(z)$, that, due to the properties of G (Definition 5) is nonpositive and 0 if and only if $z = 0$. \blacksquare

And then, an assumption is necessary.

Assumption 4: Given an initial condition $(\mathbf{x}(t_0), t_0) \in \mathcal{U}$, the trajectories of the system $\dot{\mathbf{x}} = \Psi$ are such that $(\mathbf{x}(t), t) \in \mathcal{U} \forall t > t_0$. \blacksquare

Assumption 4 is only necessary to formalize our next proposition by excluding the points \mathbf{x} such that $\mathbf{x}^*(\mathbf{x}, t)$ is not uniquely

defined. Later, in Section V, we derive sufficient conditions to state that Assumption 4 holds for the time-invariant case. We are now able to state the main result of this section.

Proposition 2: Given Assumption 4, the trajectories of the system $\dot{\mathbf{x}}(t) = \Psi(\mathbf{x}, t)$ converge to $\mathcal{C}(t)$.

Proof: Consider the Lyapunov candidate function $P = \frac{1}{2} D^2$. Given $\dot{\mathbf{x}} = \Psi$, the time derivative of the Lyapunov candidate function P can be written as $\dot{P} = \nabla P^T \Psi + \partial P / \partial t$. Thus, using (25) and recalling that $\nabla P = D \nabla D$

$$\dot{P} = -\eta G \nabla P^T \nabla D + \eta H \nabla P^T \mathbf{T}^* - \nabla P^T \Pi_T \frac{\partial \mathbf{D}}{\partial t} + \frac{\partial P}{\partial t}. \quad (29)$$

From (16) and Lemma 2, it holds that $\mathbf{D}^T \mathbf{T}^* = \nabla P^T \mathbf{T}^* = 0$, thus, the second term in (29) is null. From Lemma 4, it is clear that the last two terms in (29) cancel each other. Now, noting that $\nabla P^T \nabla D = D \nabla D^T \nabla D = D$, (29) becomes

$$\dot{P} = -\eta G D. \quad (30)$$

Now, since $P = \frac{1}{2} D^2$, $\dot{P} = D \dot{D}$. Furthermore, using Lemma 6 and denoting $\beta = v_r - v_m > 0$, we have that

$$\dot{D} \leq -\beta G(D). \quad (31)$$

Now, we use the *strong comparison lemma* for ordinary differential equations (see [38]). It states that if we have two functions $z(t)$ and $D(t)$ such that $\dot{z} = -\beta G(z)$ and $\dot{D} \leq -\beta G(D)$, $D(0) < z(0)$, and G being Lipschitz continuous, then $D(t) < z(t)$ for all (finite) t . Consider any $z(0) > D(0)$, applying Lemma 7, we conclude that $z(t) \rightarrow 0$, and since $D(t) \geq 0$ and $D(t) < z(t)$, this forces $D(t) \rightarrow 0$ as well. \blacksquare

It is important to emphasize that \dot{D} is negative definite and has a bound that depends only on D [see (31)]. Thus, even if the trajectory $\mathbf{x}(t)$ passes through a point outside \mathcal{U} (those that form a set of measure zero), we can simply choose one vector Ψ computed according to one of the possible minimizers \mathbf{x}^* . If we adopt this strategy, independently of the vector Ψ that we choose, the distance D will keep decreasing. Thus, there is no way the trajectory can get “stuck” on any region other than the target curve \mathcal{C} , i.e., there are no equilibrium points outside the target curve.

In Proposition 2, the last two terms would not cancel each other if we had used the standard normalization for the time varying-field. This happens because the term involving ∇P

would be scaled. Consequently, we would not be able to ensure $\dot{P} < 0$. Note also that with Proposition 2 we have that as $t \rightarrow \infty$ the convergence term of (25) vanishes as the robot converges to the curve. Since $\eta > 0$, from Lemma 6, and the feedforward component has no projection onto the tangent direction, the traversal term guarantees that the curve is being followed. In conclusion, (25) resumes the solution of Problem 1.

IV. ROBUSTNESS ANALYSIS

Consider a robot model given by $\dot{\mathbf{x}} = \mathbf{u} + \delta_u$, in which $\mathbf{u} \in \mathbb{R}^n$ is the input signal and $\delta_u \in \mathbb{R}^n$ is a bounded disturbance with limited norm, i.e., $\|\delta_u\| < \Delta_u \forall t \geq 0$. In this section, we present some robustness results based on ultimate boundedness arguments. In order to establish an ultimate bound for the function D in this perturbed system, consider the following Assumption.

Assumption 5: The bound Δ_u is such that $\Delta_u < v_r - v_m$. ■

Now, in the following proposition, we establish the ultimate bound result.

Proposition 3: Given an initial condition $(\mathbf{x}(t_0), t_0) \in \mathcal{U}$, Assumptions 3 and 5, if the trajectories of the perturbed system $\dot{\mathbf{x}}(t) = \mathbf{u} + \delta_u$, with $\|\delta_u\| < \Delta_u$, under the control law $\mathbf{u} = \Psi(\mathbf{x}, t)$, are such that $(\mathbf{x}(t), t) \in \mathcal{U} \forall t > t_0$, then the trajectories of such system converge to the following positive invariant set

$$\mathcal{I}(t) = \left\{ \mathbf{x} \in \mathbb{R}^n \mid D(\mathbf{x}, t) < G^{-1} \left(\frac{\Delta_u}{v_r - v_m} \right) \right\}. \quad (32)$$

Proof: First, note that $(\mathbf{x}(t), t) \in \mathcal{U} \forall t > t_0$, which ensures $\Psi(\mathbf{x}, t)$ is well defined. Assuming the perturbed system, the time derivative of the Lyapunov function P becomes

$$\dot{P} = \nabla P^T \dot{\mathbf{x}} + \frac{\partial P}{\partial t} = \nabla P^T \Psi + \nabla P^T \delta_u + \frac{\partial P}{\partial t}. \quad (33)$$

Following the steps of Proposition 2, we have that $\dot{P} = -\eta G D + \nabla P^T \delta_u$. Let $M = \max \|\delta_u\|$, then, using the fact $\dot{P} = D \dot{D}$, we have

$$\dot{D} \leq -\eta G + M < -\eta G + \Delta_u \quad (34)$$

in which the last inequality is true given that $M < \Delta_u$.

Remember that the function $G \equiv G(D)$ is continuous and strictly increasing, thus, invertible and $G(a) > b$ if and only if $a > G^{-1}(b)$. Thus, given (32), for every $\mathbf{x} \notin \mathcal{I}(t)$ we have

$$G \geq \frac{\Delta_u}{v_r - v_m} \geq \frac{\Delta_u}{\eta}. \quad (35)$$

Using (35) in (34), we conclude

$$\dot{D} \leq -\Delta_u + M < 0 \quad \forall \mathbf{x}(t) \notin \mathcal{I}(t). \quad (36)$$

Given the result in (36), an initial instant t_0 and an initial distance $D_0 = D(t_0)$, we can compute a conservative upper bound time $T(D_0, t_0)$ that ensures $\mathbf{x} \in \mathcal{I}(t)$ for $t > T(D_0, t_0)$. It is defined by

$$T(D_0, t_0) = t_0 + \frac{D_0}{\Delta_u - \max \|\delta_u\|}. \quad (37)$$

Thus

$$t > T(D_0, t_0) \Rightarrow \mathbf{x} \in \mathcal{I}(t). \quad (38)$$

Result in (38) is the statement of the ultimate boundedness of the perturbed system. ■

Note that, according to Proposition 3, the volume of the set \mathcal{I} increases as Δ_u increases, as expected. It is also interesting to observe that the faster is the movement of the curve, represented by the value of v_m , the greater is the volume of \mathcal{I} . If $\Delta_u = 0$, it follows that $\mathcal{I}(t)$ collapses to the curve $\mathcal{C}(t)$.

The result in Proposition 3 clarifies one more geometric interpretation of our result. In the presence of uncertainties, the distance that defines the system's ultimate bound is dependent on the function $G(D)$ and on the maximum velocity error Δ_u . The faster the function G approaches 1 as D increases, smaller is the set \mathcal{I} . It also shows the isotropic property of the field, meaning that deviations from the curve are equally compensated by the convergent component in all directions perpendicular to the tangent vector.

In conclusion, in the presence of limited disturbances, the state \mathbf{x} converges to the set $\mathcal{I}(t)$, which corresponds to a region around the curve $\mathcal{C}(t)$. Given the isotropic property of the field, this region is a circular tube around the curve and its cross-section radius is given by the bound in (32).

V. APPROACHABILITY OF SINGULAR POINT ANALYSIS

Proposition 2 establishes asymptotic stability. However, the field is only defined in \mathcal{U} , because outside this set there is no single definition for the vector field, and therefore, a natural question is whether or not a trajectory $\mathbf{x}(t)$ with $\mathbf{x}(0) \in \mathcal{U}$ reaches these points eventually. It will be established that, under some assumptions, for the time-invariant case no point outside \mathcal{U} is approachable.

Henceforth, we consider the time-invariant vector field $\Psi(\mathbf{x})$ in which we assume, without loss of generality that $v_r = 1$. To handle approachability to points of discontinuity of the vector field, the following convenient definition will be used.

Definition 7: Given a normalized vector field $\mathbf{f}(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}^n$ the point \mathbf{x} is said to be *approachable* by the direction \mathbf{d} , $\|\mathbf{d}\| = 1$, if the limit

$$\lim_{\epsilon \rightarrow 0+} \mathbf{f}(\mathbf{x} - \epsilon \mathbf{d}) \quad (39)$$

exists and is equal to \mathbf{d} . ■

Fig. 3 shows the intuition behind the result that will be further derived in Proposition 4. In the figure, we have an example curve \mathcal{C} in gray and in blue, we show the set of points not in \mathcal{U} . Let us assume, for the sake of simplicity that the vector field Ψ has only the convergent component ($G = 1, H = 0$) and therefore always points toward the closest point. If point $\mathbf{x}_P \notin \mathcal{U}$ is approachable, the derivative of the trajectory just before reaching the point should be equal to one of the possible values of $\Psi(\mathbf{x}_P)$. Each one of these possible values is attached to a point in $\mathbf{x}^*(\mathbf{x}_P)$. Let \mathbf{w} be one of these points and $\Psi(\mathbf{x}_P; \mathbf{w})$ the corresponding vector. The situation described is shown in the left figure. However, the right figure shows that the situation shown in the left is a contradiction: if we move backward a little in the trajectory (in the opposite

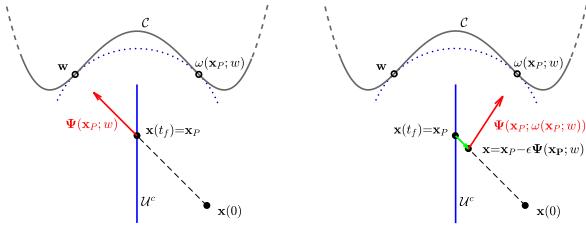


Fig. 3. Intuition behind our approachability analysis. In the figure, we assume, for the sake of simplicity that the vector field Ψ has only the convergent component ($H = 0$) and, therefore, always points toward the closest point. The trajectory cannot pass through the set $\mathcal{U}^c = \mathbb{R}^n \setminus \mathcal{U}$ (in blue), since the vector field just before reaching x_P would have to consider a point $w \in \mathcal{C}$ that is not the closest point.

direction of $\Psi(x_P; w)$, we see that the corresponding vector field is attached to *another* point in x^* : $\omega(x_P; w)$, because when we move backward a little we become closer to this point than to w . As we will show, this different point has an associated Ψ , which is different from the one related to w . Therefore, the trajectory on the left could not have been created by following the vector field. As the picture shows, this is always the case when $G = 1$. The analysis that will be presented in this section will establish sufficient conditions to affirm that these points are not approachable for the case in which $G \neq 1$.

If the point x_P is approachable by the direction d , then by following the vector field, $\dot{x} = f(x)$, it is possible to have a trajectory $x(t)$ and a *finite* time t_P such that $x(t_P) = x_P$ and $\lim_{t \rightarrow t_P^-} \dot{x}(t) = d$. In other words, there is a trajectory that reaches the point x_P from the direction d . If $f(x)$ is continuous in x , then it is trivially approachable by the direction $d = f(x)$, and only from this direction. However, if it is not continuous, it may or may not be approachable. For instance, it is easy to see using Definition 7 that for $f_1(x) = -x/\|x\|$ the point $x = 0$ is approachable by any d (it is possible to reach the point 0 by following the vector field), whereas $f_2(x) = x/\|x\|$ is not approachable by any d : the point $x = 0$ is “unstable.” Thus, the inaccessibility of a point in a vector field can be proved by establishing that it is not approachable by any direction.

We will need the following definition.

Definition 8: Given a normalized vector field $f(x)$, the *limit value set* at a point x , $\mathcal{R}_f(x)$, is the set of all different values the vector field $f(x)$ can have when we approach x for all the possible directions. That is, $\mathcal{R}_f(x) = \cup_{e \in \mathcal{E}(x)} \lim_{\epsilon \rightarrow 0+} \{f(x - \epsilon e)\}$, in which $\mathcal{E}(x) \subset \mathbb{R}^n$ is the set of directions e such that the limit exists. ■

The next result follows from our definitions.

Lemma 8: If a point x is approachable from a direction d , then $d \in \mathcal{R}_f(x)$.

Proof: Follows from the definition of approachability: if x is approachable from a direction d , $\lim_{\epsilon \rightarrow 0+} f(x - \epsilon d) = d$. Since $\mathcal{R}_f(x)$ is the set of *all* vectors that can appear in this limit and d is equal to one of them, the result holds. ■

Definition 9: Consider the time-invariant vector field $\Psi(x)$. For $x_P \notin \mathcal{U}$ and $y \in x^*(x_P)$, we define

$$\Psi(x_P; y) = -\hat{G}(x_P) \left(\frac{x_P - y}{D(x_P)} \right) + \hat{H}(x_P) \mathbf{T}^*(y). \quad (40)$$

The vector $\Psi(x_P; y)$ is just one of the possible values of the vector field $\Psi(x)$ when approaching x_P in different directions, given that $x_P \notin \mathcal{U}$. Thus

$$\mathcal{R}_\Psi(x_P) = \bigcup_{y \in x^*(x_P)} \{\Psi(x_P; y)\}. \quad (41)$$

Equation (41) means that in the neighborhood of point x_P the vectors in $\mathcal{R}_\Psi(x_P)$ are the ones computed by using each possible closest point projection x^* . Clearly, if a point $x \in \mathcal{U}$ we have that $\mathcal{R}_\Psi(x) = \{\Psi(x)\}$.

Lemma 9: Let $x_P \notin \mathcal{U}$. Given the field $\Psi(x)$, if the point x_P is approachable, then it should be by a direction $d = \Psi(x_P; y)$ for $y \in x^*(x_P)$.

Proof: This comes directly from (41) and Lemma 8. ■

Lemma 10: Let $x_P \notin \mathcal{U}$ and $y, z \in x^*(x_P)$. Define

$$\Gamma(x_P; y) = \min_{z \in x^*(x_P)} \frac{2D(x_P)|z - y|^T \mathbf{T}^*(y)|}{\|z - y\|^2} \quad (42)$$

and define $\gamma(x_P; y)$ as one of its minimizers. Then, $\gamma(x_P; y) \neq y$ and Γ is finite.

Proof: Note that $z = y$ produces an indeterminate form. However, there is still the concern that $z \rightarrow y$ when this approximation is possible (e.g., when x_P is the center of a circle). In this case, we note that $\gamma \rightarrow y$ is false, because when $z \rightarrow y$, the value to be minimized goes to infinite once we note that $(z - y)/\|z - y\|$ converges to either $\mathbf{T}^*(y)$ or $-\mathbf{T}^*(y)$ and also that there is still a term $\|z - y\|$ remaining in the denominator. Thus, since we want to minimize, clearly there are choices in which $z \rightarrow y$ is not true that produce a smaller value. This shows that $\gamma \neq y$. Since $\gamma \neq y$, it is easy to see that Γ is finite. In fact, it is less than or equal to $2D(x_P)/\|\gamma - y\|$. ■

Lemma 11: Consider the optimization problem for $y \in x^*(x_P)$

$$\min_{z \in x^*(x_P)} \lim_{\epsilon \rightarrow 0+} \|(x_P - \epsilon \Psi(x_P; y)) - z\|^2. \quad (43)$$

We can conclude the following results about this problem.

- 1) (Unicity): if x_P is approachable by the direction $d = \Psi(x_P; y)$, then the minimizer is unique.
- 2) (Not y): if

$$\frac{\hat{G}(x_P)}{\hat{H}(x_P)} > \Gamma(x_P; y) \quad (44)$$

then y is not a minimizer.

Proof: **Unicity:** Note that (43) is basically the distance between the point $(x_P - \epsilon \Psi(x_P; y))$ and the curve, because in this case, when $\epsilon \rightarrow 0+$, the closest point to the curve must lie on $x^*(x_P)$. Now, by definition, if x_P is approachable by $d = \Psi(x_P; y)$, the limit $\Psi(x_P - \epsilon d)$ must exist. This limit exists if and only if the limit $x^*(x_P - \epsilon d)$ exists. The possible values for this limit are given by the minimizers of (43), therefore, there should be only one minimizer in order to the limit to exist.

Not y : We will show that there is always a $z \neq y$ with a strictly smaller objective function than $z = y$. More specifically, we will show that, given (44), with $z = \gamma(x_P; y) \neq y$ (see Lemma 10), we obtain a smaller value in (43) than with $z = y$. This statement

is equivalent to

$$\|(\mathbf{x}_P - \epsilon \Psi(\mathbf{x}_P; \mathbf{y})) - \mathbf{y}\| > \|(\mathbf{x}_P - \epsilon \Psi(\mathbf{x}_P; \mathbf{y})) - \gamma(\mathbf{x}_P; \mathbf{y})\|. \quad (45)$$

This relation is obtained by comparing the cost function in (43) with $\mathbf{z} = \mathbf{y}$ and with $\mathbf{z} = \gamma(\mathbf{x}_P; \mathbf{y})$. After some simplifications, this statement can be written as

$$(\mathbf{y} - \gamma(\mathbf{x}_P; \mathbf{y}))^T \Psi(\mathbf{x}_P; \mathbf{y}) > 0. \quad (46)$$

Now, let $G_P = \hat{G}(\mathbf{x}_P)$, $H_P = \hat{H}(\mathbf{x}_P)$, and $D_P = D(\mathbf{x}_P)$. Using Definition 9 and after some algebra, we can see that the previous equation is equivalent to

$$G_P \frac{\|\mathbf{y} - \gamma(\mathbf{x}_P; \mathbf{y})\|^2}{2D_P} + H_P (\mathbf{y} - \gamma(\mathbf{x}_P; \mathbf{y}))^T \mathbf{T}^*(\mathbf{y}) > 0. \quad (47)$$

To obtain this, we develop the expression $\|\gamma(\mathbf{x}_P; \mathbf{y}) - \mathbf{y} + \mathbf{y} - \mathbf{x}_P\|^2$ and use $\|\mathbf{y} - \mathbf{x}_P\| = \|\gamma(\mathbf{x}_P; \mathbf{y}) - \mathbf{x}_P\|$ to obtain the equivalence $(\mathbf{x}_P - \mathbf{y})^T (\mathbf{y} - \gamma(\mathbf{x}_P; \mathbf{y})) = -\|\mathbf{y} - \gamma(\mathbf{x}_P; \mathbf{y})\|^2/2$. By noting that D_P and H_P in (47) are positive, we see that the condition in (47) is equivalent to

$$\frac{G_P}{H_P} > 2D_P \frac{(\gamma(\mathbf{x}_P; \mathbf{y}) - \mathbf{y})^T \mathbf{T}^*(\mathbf{y})}{\|\mathbf{y} - \gamma(\mathbf{x}_P; \mathbf{y})\|^2}. \quad (48)$$

This condition is guaranteed by (44). \blacksquare

Definition 10: We define, for $\mathbf{x}_P \notin \mathcal{U}$

$$K(\mathbf{x}_P) = \max_{\mathbf{y} \in \mathbf{x}^*(\mathbf{x}_P)} \Gamma(\mathbf{x}_P; \mathbf{y}). \quad (49)$$

Note that $K(\mathbf{x}_P)$ is finite, since $\Gamma(\mathbf{x}_P; \mathbf{y})$ is finite for all \mathbf{y} (see Lemma 10).

We have then our final result.

Proposition 4: Let $\mathbf{x}_P \notin \mathcal{U}$. Suppose that

$$\frac{\hat{G}(\mathbf{x}_P)}{\hat{H}(\mathbf{x}_P)} > K(\mathbf{x}_P). \quad (50)$$

Then, the point \mathbf{x}_P is not approachable by any direction.

Proof: The proof follows by contradiction. Suppose it is approachable by a direction. Then, it should be by a direction $\mathbf{d} = \Psi(\mathbf{x}_P; \mathbf{w})$ (see Lemma 9) for a $\mathbf{w} \in \mathbf{x}^*(\mathbf{x}_P)$. We will show that assuming approachability of \mathbf{x}_P by $\mathbf{d} = \Psi(\mathbf{x}_P; \mathbf{w})$, for any $\mathbf{w} \in \mathbf{x}^*(\mathbf{x}_P)$, will force a contradiction.

Since we assume that \mathbf{x}_P is approachable by the direction $\mathbf{d} = \Psi(\mathbf{x}_P; \mathbf{w})$, Lemma 11-(Unicity) allows us to conclude that the minimizer of (43) is unique. Let $\omega(\mathbf{x}_P; \mathbf{w})$ be this minimizer.

Thus, inspecting (43), we see that when we approach \mathbf{x}_P by the direction $\mathbf{d} = \Psi(\mathbf{x}_P; \mathbf{w})$, the closest point from the curve to \mathbf{x}_P is $\omega(\mathbf{x}_P; \mathbf{w})$. Henceforth, we will write for the sake of simplicity $\omega = \omega(\mathbf{x}_P; \mathbf{w})$. Therefore

$$\lim_{\epsilon \rightarrow 0+} \Psi(\mathbf{x}_P - \epsilon \Psi(\mathbf{x}_P; \mathbf{w})) = \Psi(\mathbf{x}_P; \omega). \quad (51)$$

It remains to prove that $\Psi(\mathbf{x}_P; \mathbf{w}) \neq \Psi(\mathbf{x}_P; \omega)$, implying that \mathbf{x}_P is *not* approachable by the direction $\mathbf{d} = \Psi(\mathbf{x}_P; \mathbf{w})$. This will force the contradiction, because we assumed that the point is approachable by that direction.

For this purpose, we observe that (50) implies (44) with $\mathbf{y} = \omega$, since $\omega \in \mathbf{x}^*(\mathbf{x}_P)$. Consequently, Lemma 11-(Not \mathbf{y}) holds

when $\mathbf{y} = \omega$. However, if $\Psi(\mathbf{x}_P; \mathbf{w}) = \Psi(\mathbf{x}_P; \omega)$ were true, we would conclude that ω is also a minimizer for (43) with $\mathbf{y} = \omega$. But Lemma 11-(Not \mathbf{y}) applied when $\mathbf{y} = \omega$ forbids this conclusion, because it states that no minimizer of (43) with $\mathbf{y} = \omega$ could be \mathbf{y} itself, ruling out $\mathbf{y} = \omega$ as a minimizer. \blacksquare

For any $\mathbf{x}_P \notin \mathcal{U}$, (50) can be written, together with the fact that $G^2 + H^2 = 1$, as

$$\hat{G}(\mathbf{x}_P) > \frac{K(\mathbf{x}_P)}{\sqrt{K(\mathbf{x}_P)^2 + 1}}. \quad (52)$$

One could wonder whether or not (52) can always be achievable for any $\mathbf{x}_P \notin \mathcal{U}$, since $\hat{G}(\mathbf{x}_P) \leq 1$. However, the right-hand side of this inequality is always strictly less than 1, because $K(\mathbf{x}_P)$ is finite, therefore, it is always possible to construct a G that achieves this condition.

VI. IMPLEMENTATION AND RESULTS

In this section, we present how our methodology can be easily implemented and also show some simulations and experimental results.

Animations and videos relative to the simulations and experiments are attached and also available.²

A. Implementation Methods

In order to apply the developed methodology to a given parametric representation $\mathbf{r}(s, t)$ of a curve $\mathcal{C}(t)$, the optimization problem in (1) must be solved, i.e., $s^*(\mathbf{x}, t)$ must be found. A two-stage strategy is proposed for solving this problem. In the first stage, m equally spaced samples are computed in the domain of s , i.e., s_k , $k = 1, 2, \dots, m$ such that $s_{k+1} = s_k + \Delta s$. By iterating over s_k the k^* that minimizes $\|\mathbf{x} - \mathbf{r}(s_{k^*}, t)\|$ is selected. In the second stage, a golden search method, which has a logarithmic complexity [39], is used. It assumes the domain $s_{k^*} - \Delta s \leq s \leq s_{k^*} + \Delta s$ and finally obtains $s^*(\mathbf{x}, t)$. The first stage solves the problem of possible local minima, finding the region where the global minimum of (1) is. The second stage refines the solution of the first stage assuming a limited domain of s , in which the problem in (1) is assumed to be convex.

The tangent vector $\mathbf{T}(s^*, t)$ is easily computed with (5). An option is to compute it numerically as $\mathbf{T}(s^*, t) \approx [\mathbf{r}(s^* + \delta s, t) - \mathbf{r}(s^*, t)]/\delta s$, for a sufficiently small $\delta s > 0$. If s is not the arc length (most common case), we still need to normalize $\mathbf{T}(s^*, t)$.

The computation of the time feedforward term requires the computation of $\frac{\partial \mathbf{D}}{\partial t}$. In order to compute it numerically, the optimization problem in (1) is solved once more, now considering the curve at a time $t + \delta t$. Thus, $\frac{\partial \mathbf{D}}{\partial t} = -\frac{\partial \mathbf{r}}{\partial t} \approx [\mathbf{r}(s^*(\mathbf{x}, t), t) - \mathbf{r}(s^*(\mathbf{x}, t + \delta t), t + \delta t)]/\delta t$ can be computed given a small positive δt . In order to avoid computational problems when $D = 0$, the component Ψ_G is computed as $\Psi_G = -GD/(D + \epsilon)$, in which $\epsilon > 0$ is a small number, $\epsilon = 10^{-8}$ for example.

If the curve $\mathcal{C}(t)$ is represented by a sequence of points, the necessary vectors can also be computed. Basically, to compute s_{k^*} , we can use only the first stage described before, since the second stage requires the function $\mathbf{r}(s, t)$. The derivatives

²[Online]. Available: <https://youtu.be/bWHMdUjdZkk>

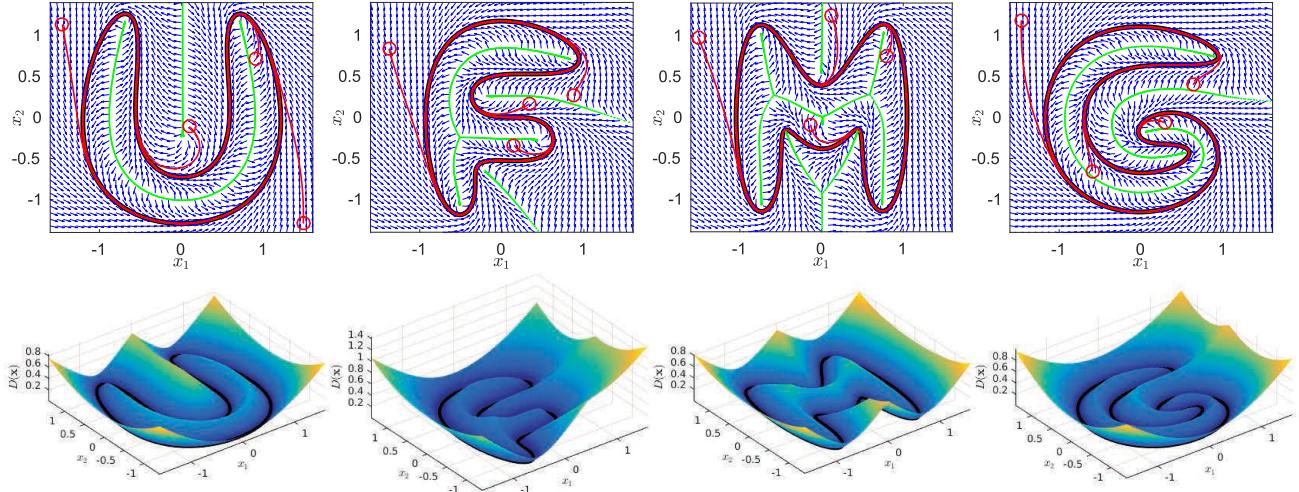


Fig. 4. Four different curve shapes in the form of the letters U, F, M, and G. In the top the curves are in black, the vector field is depicted in blue and in green are the set of points in which there are more than one point closest to the curve or, equivalently, points not in \mathcal{U} . In the bottom, we show the corresponding distance functions.

associated to the computation of \mathbf{T} and $\frac{\partial \mathbf{D}}{\partial t}$ can be estimated from the closest point on $\mathcal{C}(t)$ and its neighbors. If the sequence of points is too sparse, a simple interpolation can be used to obtain a function $\mathbf{r}(s, t)$.

B. Generic Curve Shapes

In order to demonstrate the ability of our theory to deal with generic curve shapes, we first considered four curves in the form of the letters U, F, M, and G. We considered each curve was represented by a sequence of 1000 points and used the numerical strategy presented in Section VI-A. The result is presented in Fig. 4. In the top, we show the 2D static vector fields in blue. In red, we show some trajectories of the system. In green, we show the set of “equidistant points,” those that do not belong to the set \mathcal{U} . In the bottom, we depict the associated distance functions. To compute this field, we used MATLAB R2016a, in a computer with a third generation Intel Core i7 processor. On average, each arrow on the grids of Fig. 4 took 45 μ s to be computed.

C. Real-Robot Experiments

In order to exemplify the developed strategy in a real scenario, two experiments with an actual quadrotor were performed. A DJI Matrice 100 was used and it was equipped with GPS for localization.³ The robot is also equipped with an NVIDIA Jetson Nano computer running Ubuntu 18.04 operational system. The system is completely onboard and the experiment was performed in an outdoor environment. The lower level controllers were based on [6]. The controller was implemented in C++ and run under the Robot Operating System (ROS) environment. In all results, the function $G(D)$ was chosen according to $G = (2/\pi) \operatorname{atan}(k_f D)$, in which k_f is a gain that defines the convergence weight. Given the choice of the function $G(D)$,

³We used the localization packages provided by DJI [Online]. Available: <https://github.com/dji-sdk/Onboard-SDK-ROS>

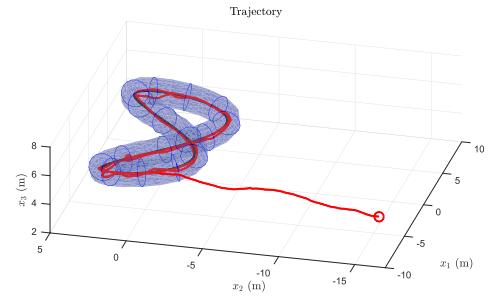


Fig. 5. Trajectory, in red, of the quadrotor following the 8 like curve. The blue tube represents the positive invariant set \mathcal{I} .

it is possible to use the result of Proposition 3 to conclude the following ultimate bound for the distance to the target curve

$$\mathcal{I}(t) = \left\{ \mathbf{x} \in \mathbb{R}^n \mid D(\mathbf{x}, t) \leq \frac{1}{k_f} \tan \left(\frac{\pi}{2} \frac{\Delta_u}{v_r - v_m} \right) \right\}. \quad (53)$$

In the actual robot experiments, the parametric equation of the curve was used, thus, we employed the two-stage algorithm described in Section VI-A.

1) *Eight Curve*: An static 8 like curve was considered to be followed by the quadrotor. The parametric equation that describes it is

$$\mathbf{r}(s, t) = \mathbf{r}(s) = \begin{bmatrix} c_1 \cos(s) \\ c_2 \sin(2s) \\ h_0 + c_3 \sin(s) \end{bmatrix} \quad (54)$$

in which $0 \leq s \leq 2\pi$, $c_1 = 6$ m, $c_2 = 3$ m, $c_3 = 1$ m, and $h_0 = 5$ m. The robot velocity was set to $v_r = 1.2$ m/s. The gain k_f of the function $G(D)$ was set to $k_f = 2.0$ m⁻¹. Note that this curve does not have a self-intersection, it passes through the point $(x_1, x_2) = (0, 0)$ at different heights, $x_3 = h_0 + c_3$ and $x_3 = h_0 - c_3$.

Fig. 5 shows in red the trajectory executed by the robot during 95 s. The initial condition was $\mathbf{x}(0) = [-8.2, -16.1, 4.7]^T$, in

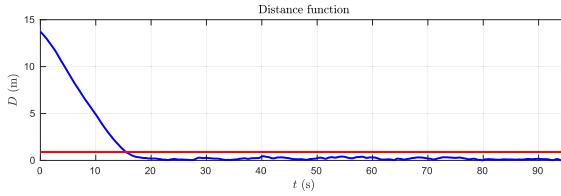


Fig. 6. Distance function associated to the trajectory in Fig. 5 and its ultimate bound.

meters. Since the curve is static $v_m = 0 \text{ m/s}$. Due to practical issues such as measurement errors and robot internal dynamics, $\Delta_u > 0$ and the trajectory of the system does not converge exactly to 0. By analyzing the norm of the difference between the commanded speed Ψ and the performed one $\dot{\mathbf{x}}$, it was possible to estimate $\Delta_u = 0.81 \text{ m/s}$. Using (53), it follows that $\mathcal{I} = \{\mathbf{x} \in \mathbb{R}^3 \mid D(\mathbf{x}) \leq 0.89 \text{ m}\}$. The invariant set \mathcal{I} is illustrated in Fig. 5 by the blue tube around the curve $\mathbf{r}(s)$, in black. The blue tube does not intersect itself, thus, we have a guarantee that, even in the case of uncertainties, when the drone is passing through the “center of the eight,” the closest point will not jump to the other part of the curve.

In Fig. 6, the evolution of the distance function D is presented. The ultimate bound for D is represented by the red line. Note that once D is below the red line it does not go back above. In this experiment, the error after the convergence was kept below 0.45 m, which is in the set \mathcal{I} .

2) *Deforming Saddle*: In order to demonstrate the ability of the strategy to deal with time-varying curves, a saddle-like curve that changes in shape while rotates around the z -axis was considered. The parametric equation that describes this curve is

$$\mathbf{r}(s, t) = R_z(\omega_1 t) \begin{bmatrix} c_1 \cos(s) \\ c_2 \sin(s) \\ h_0 + c_3 \cos(\omega_2 t) \cos(s)^2 \end{bmatrix} \quad (55)$$

in which $0 \leq s \leq 2\pi$ and $R_z(\omega_1 t)$ is a rotation matrix of ω_1 radians around the z -axis. The constants are given by $c_1 = 5 \text{ m}$, $c_2 = 5 \text{ m}$, $c_3 = 3.5 \text{ m}$, $\omega_1 = 0.05 \text{ rad/s}$, $\omega_2 = 0.025 \text{ rad/s}$, and $h_0 = 7 \text{ m}$. In addition, $v_r = 2.1 \text{ m/s}$ and $k_f = 1.8 \text{ m}^{-1}$.

Fig. 7 shows four stages of the drone following the moving and deforming saddle-like curve. The plot corresponds to 192 s of the experiment. Note how the curve, in black, changes between frames. The past trajectory of the system is shown in red. The initial condition was $\mathbf{x}(0) = [-4.8, -4.9, 4.3]^T$, in meters. During the experiment, the maximum value of $\|\Psi_T\|$ was 0.265 m/s, thus, in order to analyze the theory in Section IV, $v_m = 0.265 \text{ m/s}$. As in the previous experiment, it is possible to consider $\Delta_u = 1.1 \text{ m/s}$ by analyzing the difference between the commanded speed and the performed one. Again, using (53), it follows that $\mathcal{I}(t) = \{\mathbf{x} \in \mathbb{R}^3 \mid D(\mathbf{x}, t) \leq 0.74 \text{ m}\}$.

In the top of Fig. 8, the distance function D is presented. The red line represents the ultimate bound for the function D . After convergence, the distance was kept below 0.58 m. In the bottom, we observe how the norm of the feedforward component, Ψ_T , bounded by v_m is less than v_r .

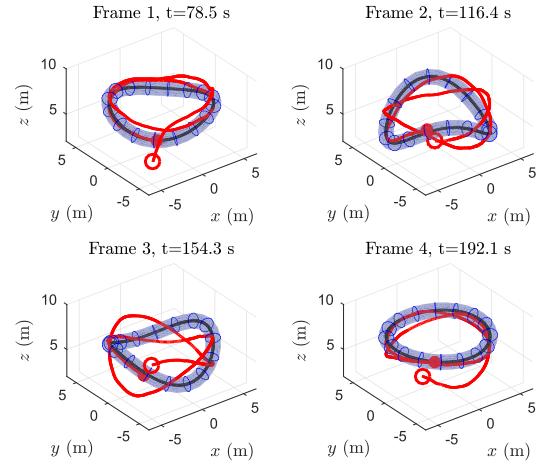


Fig. 7. Trajectory, in red, of the quadrotor following the saddle-like curve in four different instants. The blue tube represents the set $\mathcal{I}(t)$.

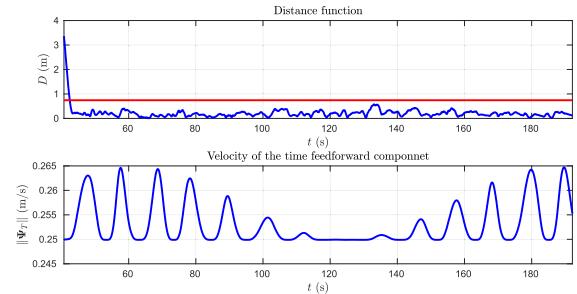


Fig. 8. Distance function associated with the trajectory in Fig. 7 and its ultimate bound (top). Norm of the feedforward component (bottom).

D. Drone Dynamics

In order to exemplify how the vector field can be used to control a robot considering higher order dynamics, we present an example of a drone following a vector field by using the control proposed in [6]. The considered quadcopter model, which assumes the thrust τ and angular velocity ω as control inputs, is given by

$$\dot{\mathbf{p}} = \mathbf{v} \quad (56a)$$

$$\dot{\mathbf{v}} = R\hat{\mathbf{z}} \frac{\tau + \delta_\tau}{m} - g\hat{\mathbf{z}} + \frac{f_d(\mathbf{v})}{m} \quad (56b)$$

$$\dot{R} = RS(\omega + \delta_\omega) \quad (56c)$$

in which \mathbf{p} , \mathbf{v} are states representing position and velocity on the inertial frame, respectively, and R is a rotation matrix representing the vehicle’s orientation. Gravity acceleration and drone’s mass are g and m , respectively, $\hat{\mathbf{z}} = [0 \ 0 \ 1]^T$, $f_d(\mathbf{v})$ is a known drag force and $S(\omega)$ is the skew-symmetric matrix. The variables $\delta_\tau \in \mathbb{R}$ and $\delta_\omega \in \mathbb{R}^3$ are disturbances on the control inputs. Refer to [6], for details on the model and the lower level controllers used.

The considered curve has a complicated shape, and it would be difficult to represent it with functions α_1 , and α_2 , as required by [15], which is the methodology used in [6]. For $0 \leq s \leq 2\pi$, the

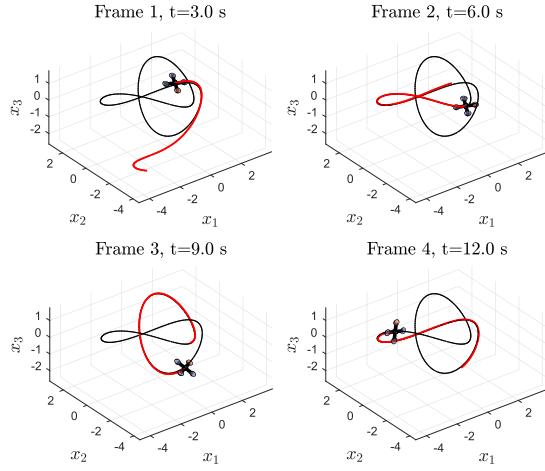


Fig. 9. Trajectory, in red, of the quadrotor following the complex curve in four different instants. The simulation used the controller proposed in [6] with a time step of 25 ms.

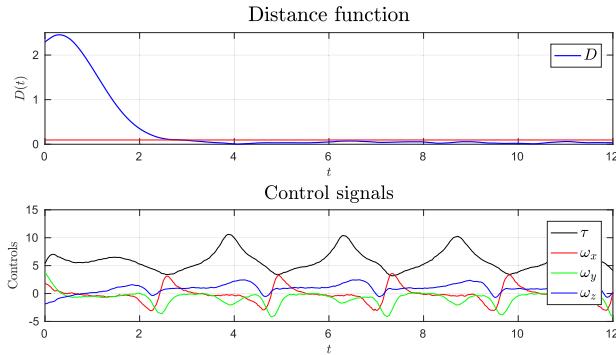


Fig. 10. Distance function associated with the trajectory in Fig. 9 (top) and computed control signals (bottom). The units are N for τ and rad/s for ω_x , ω_y , and ω_z .

parametric equation that describes the curve is

$$\mathbf{r}(s, t) = \mathbf{r}(s) = \begin{bmatrix} \sin(s) + 2 \sin(2s) \\ \cos(s) - 2 \cos(2s) \\ -\sin(3s) \end{bmatrix}. \quad (57)$$

Fig. 9 presents the trajectory performed by the quadcopter in a simulation that considers the third-order model. The simulation time step was 25 ms, $m = 0.5$ kg and we assumed δ_τ and δ_ω as white Gaussian noises with standard deviations of $\sigma_\tau = 1.0$ N and $\sigma_\omega = 0.5$ rad/s, respectively.

In Fig. 10, the distance function D is presented in the top. In the bottom, we observe the control signals for thrust (black), and angular velocities (red, green, and blue). Note that the distance function increases in the beginning. This is due to the initial condition of the higher order system, and after this transient, D converges and stays below 9.6 cm, value computed according to (53) and depicted by the red line on the top figure.

E. Manipulator

We now apply the vector field methodology to control a 6 DOF Kuka KRL4 robot to draw a circle orthogonal to the ground with the tool in the end-effector parallel to the ground. The vector

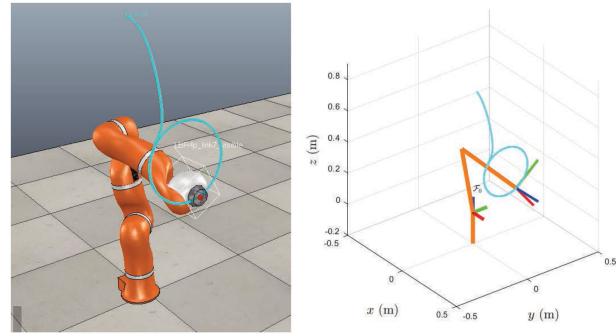


Fig. 11. Scene in CoppeliaSim with the KUKA robot following a curve in the joints space (left). MATLAB plot of the simulation (right).

field was applied directly to the robot's joint space. This implementation requires a simple trick, presented in the Appendix, to compute the distance function, originally designed for the Euclidean space. In the case of the Kuka robot, we can compute the curve C in the joint space by solving an inverse kinematic problem. This can be done by using numerical methods.

To simulate the application of the vector field in the control of the manipulator taking into account its dynamics, we used the CoppeliaSim simulator.⁴ We used an interface that allows the user to submit a reference joint speed computed using the distance field, which is internally converted into torque by a control loop.

The target curve C , mapped into \mathbb{R}^6 using the replication trick (see the Appendix), was defined so that the last link of the robot's kinematic chain performs a circle in a plane orthogonal to the ground. Also, the manipulator's tool should be orthogonal to the circular face defined by the circle. This can be interpreted as a task of welding a circular shape in a plane surface or drawing a circle on a whiteboard with a marker pen (either way, the tool must be aligned with the surface's normal).

Fig. 11 shows on the left, the scene used in the experiment. The cyan trajectory was the one performed by the robot's tool. On the right, we show a plot where we can observe that the tool of the end-effector (blue axis) is indeed parallel to the ground, aligned with the x -axis.

It is important to emphasize that the circle in Fig. 11 is not the curve C , it is only the position of the last link when the joints follow the path in the joints' space. It is a projection of the real curve C by the direct kinematics of the manipulators' tool. The curve C lies in six dimensions, thus, it cannot be directly visualized. In fact, in our example, the robot completes two laps on the circle while completing only one lap on the curve C . In order to show the error in tracking the curve in the joints space, Fig. 12 shows on the top the function D , computed according to the closest point map given in (63). On the bottom of Fig. 12 are the joint angles. The initial condition had all joints with a 0 value. It corresponds to the arm stretched out in a singular configuration. Since our vector field was applied directly on the joints, and no inverse kinematics was necessary to track the vector field, this singular configuration caused no harm to the

⁴[Online]. Available: <https://www.coppeliarobotics.com/>

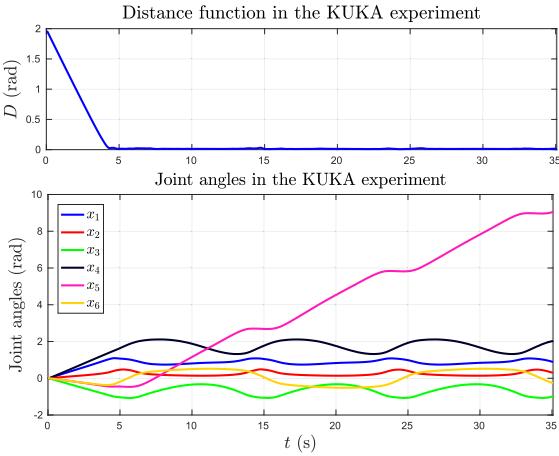


Fig. 12. Distance function in the joints space in the manipulator example on the top. Associated joint angles on the bottom.

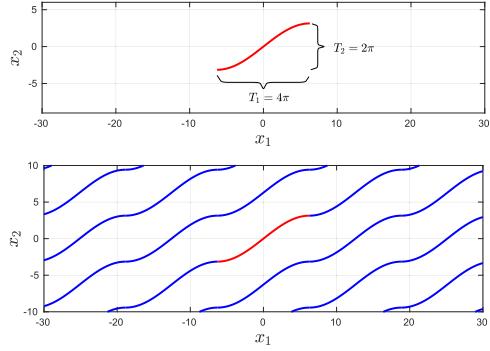


Fig. 13. Replication process for a bidimensional curve (top), bounded in a rectangle with sides $T_1 = 4\pi$ in x_1 and $T_2 = 2\pi$ in x_2 . Note also that the derivatives agree at the end-points. These facts imply that the curve in the bidimensional torus S^2 is closed and differentiable.

task. Note also that joint x_5 reaches values larger than 2π . This also caused no harm, given that we used the replication trick in the Appendix.

VII. CONCLUSION

In this article, we presented a strategy for generating vector fields in n dimensions that converge to and follow time-varying curves. The strategy is based on the Euclidean distance function. We recall the following four important properties/advantages that the vector fields constructed by our method incorporate:

- 1) it can be easily constructed from a parametric representation of the curve or even by a curve represented by a sequence of points;
- 2) its convergence pattern is isotropic on space, which facilitates the calibration of the convergence gain of the field;
- 3) it has a constant norm even for time-variant cases;
- 4) it does not contain equilibrium points.

All these important theoretical and practical features emerge from the use of the Euclidean distance as an error measurement function, instead of a general analytic function.

Disturbance in the simple integrator model was taken into account in order to evaluate the performance of the technique

in real-robot applications, in which imperfections on the lower level controllers and other disturbances are always present. Discontinuities on the field due to equidistant points were formally treated. We validated our strategy with several simulations, including a quadcopter with a more complex dynamics and a manipulator with 6 DOFs. Two outdoor experiments using an actual quadcopter show how the method is suitable for practical applications.

In future works, we intend to address singularities in time-varying scenarios and also curves with self-intersection. The current method to guarantee that a point not in \mathcal{U} is unapproachable requires the computation of the set $\mathbb{R}^n \setminus \mathcal{U}$. This task has a complexity that grows with the number of dimensions n . In the future, we intend to investigate new methods to do this in an efficient manner. A possible approach is to consider the parameter s^* as a state of the field and define its propagation law [22], [31]. We also seek an extension of our method that is able to take into account obstacles in the environment.

APPENDIX

A. Field in the Torus Space

The state-space formed by the rotation of each joint of a robotic manipulator does not have an Euclidean topology: a rotation of θ degrees is the same as one of $\theta + 2\pi k$ for all $k \in \mathbb{Z}$. For this reason, the Euclidean distance is not an appropriate metric for this space. Indeed, a closed curve in the correct topological space of \mathbf{x} (that is of the n -dimensional torus, $S^n = \mathbb{S} \times \mathbb{S} \times \dots \times \mathbb{S}$) can become open and bounded when mapped to the Euclidean space.

This problem can be addressed by a ‘‘replication’’ trick. Suppose we have the target curve \mathcal{C} , $\mathbf{r}(s)$, static in time, as an open, bounded curve in \mathbb{R}^n . Assume that $s = 0$ and $s = s_{\text{end}}$ are the starting and ending parameter values of the curve’s parametrization. To be a closed curve in the n -dimensional torus space, it must hold that for all $i = 1, 2, \dots, n$, $|r_i(s_{\text{end}}) - r_i(0)| = 2\pi k_i \equiv T_i$, for a nonnegative integer k_i (possibly different for all i). If the curve is differentiable, it also holds that $\frac{\partial r_i}{\partial s}(0) = \frac{\partial r_i}{\partial s}(s_{\text{end}})$. Thus, the target curve can be bounded in a hyperrectangle in \mathbb{R}^n whose sides are integer multiples of 2π .

Suppose this curve is replicated across \mathbb{R}^n by summing to $\mathbf{r}(s)$ constant vectors Δ in which Δ_i is an integer multiple of T_i . See Fig. 13 for an example of this replication. Now, the space \mathbb{R}^n is filled with (disjoint) copies of \mathcal{C} , which we call \mathcal{C}_{rep} and we can now apply the traditional Euclidean distance toward \mathcal{C}_{rep} because it already takes into consideration characteristics of the original topology of the state space (through replication). Note that, after replication, the curve will not be in accordance with Assumption 1, which is now relaxed to incorporate this new type of curve. We now consider curves \mathcal{C} that consist of the union of disjoint curves of the previously considered types.

Of course, it is not necessary to consider directly the (Euclidean) distance toward \mathcal{C}_{rep} , which is formed by disjoint, non-bounded curves, to compute the (Euclidean) distance between an state \mathbf{x} and \mathcal{C}_{rep} . We can compute it from the original version, \mathcal{C} , by solving the following optimization problem not only in the

variable s but also in the integer variables k_i :

$$D(\mathbf{x})^2 = \min_{\substack{s \in [0, s_{\text{end}}] \\ k_i \in \mathbb{Z}}} \sum_{i=1}^n (x_i - r_i(s) - k_i T_i)^2. \quad (58)$$

We can rewrite this problem as

$$D(\mathbf{x})^2 = \min_{s \in [0, s_{\text{end}}]} \sum_{i=1}^n \min_{k_i \in \mathbb{Z}} (x_i - r_i(s) - k_i T_i)^2. \quad (59)$$

We now can easily solve the internal problems analytically. Let $\lfloor u \rfloor$ be the *floor* function and $\text{rem}(\theta, T) = \theta - T \lfloor \theta/T \rfloor$ be the *remainder* function. Define the function $D_S(\theta, T)$ as

$$\min_{k \in \mathbb{Z}} |\theta - kT| = \begin{cases} \text{rem}(\theta, T) & \text{if } \text{rem}(\theta, T) \leq T/2 \\ T - \text{rem}(\theta, T) & \text{otherwise} \end{cases} \quad (60)$$

which is, by the way, a continuous function. Now, we can rewrite $D(\mathbf{x})$ as

$$D(\mathbf{x})^2 = \min_{s \in [0, s_{\text{end}}]} \sum_{i=1}^n D_S(x_i - r_i(s), T_i)^2. \quad (61)$$

Thus, we reduced our problem, a search in infinite unbounded curves, to a search in the original (bounded) curve \mathcal{C} . This construction also induces an expression for the closest point map $\mathbf{x}^*(\mathbf{x})$. Define the function $k_S^*(\theta, T)$ as

$$\arg \min_{k \in \mathbb{Z}} |\theta - kT| = \begin{cases} \lfloor \theta/T \rfloor & \text{if } \text{rem}(\theta, T) < T/2 \\ \lfloor \theta/T \rfloor + 1 & \text{if } \text{rem}(\theta, T) > T/2 \\ \text{either} & \text{if } \text{rem}(\theta, T) = T/2. \end{cases} \quad (62)$$

Then, if $s_{\text{red}}^*(\mathbf{x})$ is the optimal s found by solving the optimization problem in (61), we can write each component of $\mathbf{x}^*(\mathbf{x})$, $x_i^*(\mathbf{x})$, as

$$x_i^*(\mathbf{x}) = r_i(s_{\text{red}}^*(\mathbf{x})) + T_i k_S^*(x_i - r_i(s_{\text{red}}^*(\mathbf{x))), T_i). \quad (63)$$

Note that this construction implies that the source of problematic points, points not in \mathcal{U} , are twofold: first, when $s_{\text{red}}^*(\mathbf{x})$ is not uniquely defined, and second, when $\text{rem}(x_i - r_i, T_i) = T_i/2$, in which case $k_S^*(x_i - r_i, T)$ has two possible choices, implying that there is more than one choice for the closest point.

REFERENCES

- [1] J. L. G. Olavo, G. D. Thums, T. A. Jesus, L. C. A. Pimenta, L. A. B. Torres, and R. M. Palhares, "Robust guidance strategy for target circulation by controlled UAV," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 54, no. 3, pp. 1415–1431, Jun. 2018.
- [2] A. M. C. Rezende, V. M. Gonçalves, G. V. Raffo, and L. C. A. Pimenta, "Robust fixed-wing UAV guidance with circulating artificial vector fields," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 5892–5899.
- [3] S. Zhao, X. Wang, Z. Lin, D. Zhang, and L. Shen, "Integrating vector field approach and input-to-state stability curved path following for unmanned aerial vehicles," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 50, no. 8, pp. 2897–2904, Aug. 2020.
- [4] R. W. Beard and T. W. McLain, *Small Unmanned Aircraft: Theory and Practice*. Princeton, NJ, USA: Princeton Univ. Press, 2012.
- [5] D. Zhou and M. Schwager, "Vector field following for quadrotors using differential flatness," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2014, pp. 6567–6572.
- [6] A. M. C. Rezende, V. M. Gonçalves, A. H. D. Nunes, and L. C. A. Pimenta, "Robust quadcopter control with artificial vector fields," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2020, pp. 6381–6387.
- [7] A. M. C. Rezende, V. R. F. Miranda, H. N. Machado, A. C. B. Chiella, V. M. Gonçalves, and G. M. Freitas, "Autonomous system for a racing quadcopter," in *Proc. 19th Int. Conf. Adv. Robot.*, 2019, pp. 547–552.
- [8] 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.
- [9] J. Kwon, C. Kim, and D. Chwa, "Vector field trajectory tracking control for wheeled mobile robots," in *Proc. IEEE Int. Conf. Ind. Technol.*, 2009, pp. 1–6.
- [10] B. Abdurahman, A. Savvaris, and A. Tsourdos, "A comparison between guidance laws for AUVs using relative kinematics," in *Proc. OCEANS - Aberdeen*, Jun. 2017, pp. 1–6.
- [11] M. M. Marinho, B. V. Adorno, K. Harada, and M. Mitsuishi, "Dynamic active constraints for surgical robots using vector-field inequalities," *IEEE Trans. Robot.*, vol. 35, no. 5, pp. 1166–1185, Oct. 2019.
- [12] V. M. Gonçalves, B. V. Adorno, A. Crosnier, and P. Fraisse, "Stable-by-design kinematic control based on optimization," *IEEE Trans. Robot.*, vol. 36, no. 3, pp. 644–656, Jun. 2020.
- [13] H. J. Asl, T. Narikiyo, and M. Kawanishi, "Prescribed performance velocity field control of robotic exoskeletons with neural network," in *Proc. IEEE Int. Conf. Robot. Biomimetics*, 2017, pp. 2704–2709.
- [14] A. Martínez, B. Lawson, C. Durrough, and M. Goldfarb, "A velocity-field-based controller for assisting leg movement during walking with a bilateral hip and knee lower limb exoskeleton," *IEEE Trans. Robot.*, vol. 35, no. 2, pp. 307–316, Apr. 2019.
- [15] V. M. Gonçalves, 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.
- [16] E. W. Frew and D. Lawrence, "Tracking dynamic star curves using guidance vector fields," *J. Guid., Control, Dyn.*, vol. 40, no. 6, pp. 1488–1495, 2017.
- [17] V. M. Gonçalves *et al.*, "Circulation of curves using vector fields: Actual robot experiments in 2D and 3D workspaces," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2010, pp. 1136–1141.
- [18] T. Lee, M. Leok, and N. H. McClamroch, "Geometric tracking control of a quadrotor UAV on SE(3)," in *Proc. 49th IEEE Conf. Decis. Control*, 2010, pp. 5420–5425.
- [19] B. Rubí, R. Pérez, and B. Morcego, "A survey of path following control strategies for UAVs focused on quadrotors," *J. Intell. Robot. Syst.*, pp. 241–265, Nov. 2019.
- [20] 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.
- [21] A. M. C. Rezende *et al.*, "Indoor localization and navigation control strategies for a mobile robot designed to inspect confined environments," in *Proc. IEEE 16th Int. Conf. Autom. Sci. Eng.*, 2020, pp. 1427–1433.
- [22] W. Yao, H. G. de Marina, B. Lin, and M. Cao, "Singularity-free guiding vector field for robot navigation," *IEEE Trans. Robot.*, pp. 1–16, 2020, doi: [10.1109/TRO.2020.3043690](https://doi.org/10.1109/TRO.2020.3043690).
- [23] D. Lawrence, E. Frew, and W. Pisano, "Lyapunov vector fields for autonomous UAV flight control," in *Proc. AIAA Guid., Navigat. Control Conf. Exhibit*, 2007, pp. 1–22.
- [24] E. W. Frew, D. A. Lawrence, and S. Morris, "Coordinated standoff tracking of moving targets using Lyapunov guidance vector fields," *J. Guid., Control, Dyn.*, vol. 31, no. 2, pp. 290–306, 2008.
- [25] N. Ceccarelli, M. Di Marco, A. Garulli, and A. Giannitrapani, "Collective circular motion of multi-vehicle systems," *Automatica*, vol. 44, no. 12, pp. 3025–3035, 2008.
- [26] Y. Liang, Y. Jia, J. Du, and J. Zhang, "Vector field guidance for three-dimensional curved path following with fixed-wing UAVs," in *Proc. Amer. Control Conf.*, 2015, pp. 1187–1192.
- [27] C. Wu, J. Chen, D. Jeltsema, and C. Dai, "Guidance vector field encoding based on contraction analysis," in *Proc. Eur. Control Conf.*, 2018, pp. 282–287.
- [28] E. W. Frew and D. Lawrence, "Tracking expanding star curves using guidance vector fields," in *Proc. Amer. Control Conf.*, 2012, pp. 1749–1754.
- [29] N. Cho, Y. Kim, and S. Park, "Three-dimensional nonlinear differential geometric path-following guidance law," *J. Guid., Control, Dyn.*, vol. 38, no. 12, pp. 2366–2385, 2015.
- [30] M. Bibuli, G. Bruzzone, M. Caccia, and L. Lapierre, "Path-following algorithms and experiments for an unmanned surface vehicle," *J. Field Robot.*, vol. 26, no. 8, pp. 669–688, 2009.

- [31] L. Lapierre, D. Soetanto, and A. Pascoal, "Nonsingular path following control of a unicycle in the presence of parametric modelling uncertainties," *Int. J. Robust Nonlinear Control: IFAC-Affiliated J.*, vol. 16, no. 10, pp. 485–503, 2006.
- [32] M. Breivik and T. I. Fossen, "Principles of guidance-based path following in 2D and 3D," in *Proc. 44th IEEE Conf. Decis. Control*, 2005, pp. 627–634.
- [33] J. Płaskonka, "Different kinematic path following controllers for a wheeled mobile robot of (2, 0) type," *J. Intell. Robot. Syst.*, vol. 77, no. 3/4, pp. 481–498, 2015.
- [34] C. Richter, A. Bry, and N. Roy, "Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments," in *Robotics Research*. Berlin, Germany: Springer, 2016, pp. 649–666.
- [35] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2011, pp. 2520–2525.
- [36] T. S. Andersen and R. Kristiansen, "Quaternion path-following in three dimensions for a fixed-wing UAV using quaternion blending," in *Proc. IEEE Conf. Control Technol. Appl.*, 2018, pp. 1597–1602.
- [37] L. Zajiček, "Differentiability of the distance function and points of multi-valuedness of the metric projection in banach space," *Czechoslovak Math. J.*, vol. 33, no. 2, pp. 292–308, 1983.
- [38] A. McNabb, "Comparison theorems for differential equations," *J. Math. Anal. Appl.*, vol. 119, no. 1, pp. 417–428, 1986.
- [39] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge, U.K.: Cambridge Univ. Press, 2007.



Adriano M. C. Rezende received the bachelor's and master's degrees in electrical engineering in 2017 and 2019, respectively, from the Universidade Federal de Minas Gerais (UFMG), Belo Horizonte, Brazil, where he is currently working toward the Ph.D. degree in electrical engineering.

During his undergraduate course, he spent a year with Colorado State University (CSU), Fort Collins, CO, USA. He is currently with the development of autonomous navigation strategies for a ground exploration robot from the Instituto Tecnológico Vale (ITV), Belém, Brazil. In 2019, he was classified for the Artificial Intelligence Robotic Racing (AIRR). His current research interests include robot navigation (ground and aerial), motion planning, localization, and control theory.



Vinicius M. Gonçalves received the B.S. degree in control and automation engineering and the Ph.D. degree in control in dioid algebras and discrete event systems from the Universidade Federal de Minas Gerais, Belo Horizonte, Brazil, in 2011 and 2014, respectively.

He is currently an Assistant Professor with the Department of Electrical Engineering, Federal University of Minas Gerais. His current research interests include control, robotics, optimization, and discrete-event systems.



Luciano C. A. Pimenta received the B.S., M.Sc., and Ph.D. degrees in electrical engineering from the Universidade Federal de Minas Gerais (UFMG), Belo Horizonte, Brazil, in 2003, 2005, and 2009, respectively.

From April 2007 to June 2008, he was a Visiting Ph.D. Student with the General Robotics, Automation, Sensing and Perception (GRASP) Laboratory, University of Pennsylvania, Philadelphia, PA, USA. He is currently an Associate Professor with the Department of Electronic Engineering, UFMG. His research interests include robotics, multirobot systems, motion planning, and control theory.