IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Safe vector field for robot navigation in n-dimensions

Arthur H. D. Nunes, Vinicius M. Gonçalves, and Luciano C. A. Pimenta

***Abstract*—In this work, we propose a novel artificial vector field for robot navigation in n-dimensional path-following tasks, designed to ensure safety and convergence with a smoothed control law. Unlike previous methods based on discontinuous Euclidean distance functions, our approach uses a smooth Euclidean-like function to achieve a continuous control law formulation and a field combination to balance the objectives of avoiding obstacles and following the path. This results in a navigation method that follows a target path while preventing robots from approaching obstacles, which can be used in different applications. We provide formal proofs for safety using barrier functions concepts and path convergence via Lyapunov theory. The methodology is validated through extensive numerical simulations and real-world experiments. Those include extrapolations of the methodology in more complex cases, such as quadcopters and multi-robot systems to underline the method's advantages in achieving safe and reliable robot navigation.**

I. INTRODUCTION

SAFE navigation is a timely and important problem in autonomous robotics [1], [2], [3]. As those robots are increasingly being deployed in complex environments, they require navigation solutions that ensure tasks can be performed safely. Defining what it means to be safe is another key point. Usually, it means avoiding obstacles, but it can also mean not colliding with other agents, keeping distance from edges, performing smooth movements, and avoiding dangerous areas. Generally, safety constraints can be mapped as keeping a distance from a forbidden set in the workspace.

The solution to the safe autonomous navigation problem can be applied to a wide range of applications such as surveillance, inspection, search and rescue, urban mobility, autonomous vehicles, among others [4], [5], [6], [7], [8], [9]. Navigation strategies based on vector fields have been discussed in many works [10], [11], [12], [13], [14], [15], [16], [17], [18]. Some of the mentioned works, [13], [14], [18], also consider obstacle avoidance. However, time-varying paths and generic-shaped obstacles have not been considered.

A recent vector field method for time-varying path-following in n-dimensions has been previously proposed in [19]. This vector field is non-conservative as it can guide a robot to follow a closed curve, with this curve being a desired limit-cycle. By using a parametric representation of the target curve, it has several advantages over other approaches, such as [10], [11], [17], [18]. In [19], the presence of obstacles was not considered. We use its result as part of our new method.

In [20], this vector field was extended to allow obstacle avoidance in the formulation by considering obstacle circumnavigation when an imminent collision is detected. The circumnavigation is achieved by using the obstacle "closest point" to the robot. Unfortunately, the method in [20] lacks formal proofs of safety, as the safe distance can be violated in some edge cases. In our method, we solve this issue by providing a vector field with guaranteed safety using barrier functions concepts [21].

Another key point is that none of the aforementioned works is concerned with providing continuous vector field control laws. This may not be an issue if we assume we can instantaneously impose any velocity vector to the robot. However, the underlying dynamics usually do not allow infinite accelerations. This turns out to be an important element when considering real-world applications.

Non-smoothness and undesirable conditions such as local minima are, in general, characteristics of a group of approaches for obstacle avoidance called reactive [22]. They consider navigation free of obstacles in the first moment and take action to avoid a collision when an obstacle appears. The non-smoothness of the reactive approaches often arises due to the usage of Euclidean distance functions, which are not continuous because they require the computation of the global minimum. An alternative to solve this is to use a smooth distance function such as the one presented in [23]. We employ a similar idea in this work while also considering important features that allow us to ensure safety.

On the other hand, there are deliberative approaches that consider some path planning that computes a path free of collisions. In [24], the authors propose a Vector Field Rapidly-exploring Random Tree (VFRRT) [25]. Similarly, in [26] the RRT* together with a vector field is used to generate optimal collision-free motion plans. Based on [26], the work [27] uses the RRT* with a vector field to drive a semi-autonomous robot in the execution of user-defined movements. Although the aforementioned works are deliberative approaches, the methodology presented in [27] does not require global localization, but only a local map and the detection of obstacles.

Other strategies can blend deliberative and reactive elements, as MPC, that repeatedly solves an optimization problem to plan a safe trajectory over a short future horizon. In [28], an MPC-based method was proposed to solve the problem of safe navigation in unknown environments. However, the online optimization required by MPC imposes a significant computational challenge. Even though some recent

Manuscript received: October 24th, 2025; Accepted: December 16th, 2025.

This paper was recommended for publication by Editor Aniket Bera upon evaluation of the Associate Editor and Reviewers comments.

The authors are with the Graduate Program in Electrial Engineering (PPGEE) of the Universidade Federal de Minas Gerais (UFMG), Belo Horizonte, MG, 31270-901, Brazil (emails: arthurhdn@ufmg.br; vinicius.marianog@gmail.com; lucpim@ufmg.br). The first author is affiliated with NUAR Systems LTDA, Contagem, MG, Brazil.

This work was in part supported by the grants CNPq (Brazilian National Research Council) 309925/2023-1 and 301298/2025-4, and FAPEMIG (under grants APQ-02144-18 and APQ-0063023). This work was partially supported by Petrobras/ANP under grants 2023/00494-5 and 2023/00643-0.

Digital Object Identifier (DOI): see top of this page.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

works, such as [29], propose fast solvers for MPC, this class of methods often solves trajectory-tracking problems. In this work, we propose a path-following strategy.

The advantages of reactive approaches are that they are usually simple and fast, so they can be easily applied to unknown environments and have a low computational cost. With these advantages, we propose a path-following method with a reactive obstacle avoidance, while solving the discontinuity problem that may arise from the Euclidean distance function.

In summary, we address the safe robot navigation problem in n-dimensions. The main contributions of the work are:

- Proposition of a smooth distance function that is less than or equal to the Euclidean distance function;
- Proposition of a novel vector field approach capable of following dynamic paths and reactively avoiding generic-shaped dynamic obstacles;
- Formal guarantees of safety using concepts of barrier certificates;
- Extensive validation in simulations and real-robot experiments, including quadcopters and centralized multi-robot systems.

A. Notation

In this work, we use regular symbols for scalars, *e.g.* $D \in \mathbb{R}$, and bold symbols for vectors, *e.g.* $\boldsymbol{D} \in \mathbb{R}^n$. Vectors are assumed to be column vectors.

II. PROBLEM STATEMENT

Let $\boldsymbol{p} \in \mathbb{R}^n$ be the configuration of the robot and $\boldsymbol{v} \in \mathbb{R}^n$ its velocity. Consider also the following single integrator model

$$\dot{\boldsymbol{p}} = \boldsymbol{v}. \tag{1}$$

We aim to control the robot configuration to follow a given target curve $\boldsymbol{\mathcal{C}}(t) \subset \mathbb{R}^n$ with parametric representation $\boldsymbol{c}(s,t)$. The first problem we address is path-following as formalized below:

Problem 1 (Path-following)**.** Considering the system (1), find a control law for $\boldsymbol{v}$ such that the trajectories of the system converge to and follow a given time-varying target path $\boldsymbol{\mathcal{C}}(t) \subset \mathbb{R}^n$ parametrized as $\boldsymbol{c}(s,t)$ with constant speed $v_r > 0$.

However, we must ensure safety during navigation. Let $\boldsymbol{\mathcal{O}}(t) \subset \mathbb{R}^n$ be a forbidden set, also referred to as *obstacle set*. Assume that this set is represented by a finite number of time-varying points $\boldsymbol{o}_i(t) \in \boldsymbol{\mathcal{O}}(t)$, in which the number of points stays the same, N, and each function $\boldsymbol{o}_i(t)$ is differentiable in t.

If parts of this set are perceived between the robot and the target path, the robot must prioritize the safety over the path following, i.e., the robot is allowed to move away from the curve to accomplish the obstacle deviation and then return to it once the avoidance is finished.

Let the half-squared distance between the robot configuration $\boldsymbol{p}$ and the set $\boldsymbol{\mathcal{O}}(t)$ be defined as

$$D_{\boldsymbol{\mathcal{O}}}(\boldsymbol{p},t) \equiv \min_{i=1,\dots,N} \frac{1}{2}\|\boldsymbol{p} - \boldsymbol{o}_i(t)\|^2. \tag{2}$$

Also, let $\boldsymbol{o}_*(\boldsymbol{p},t)$ be a witness point such that $D_{\boldsymbol{\mathcal{O}}}(\boldsymbol{p},t) = \frac{1}{2}\|\boldsymbol{p} - \boldsymbol{o}_*(\boldsymbol{p},t)\|^2$:

$$\boldsymbol{o}_*(\boldsymbol{p},t) \in \arg \min_{\boldsymbol{o}_i \in \{\boldsymbol{o}_1,\dots,\boldsymbol{o}_N\}} \frac{1}{2}\|\boldsymbol{p} - \boldsymbol{o}_i(t)\|^2. \tag{3}$$

Then, we state the safety problem to be solved as follows.

Problem 2 (Safety)**.** The robot must avoid the forbidden set by keeping at least a safe distance $\lambda > 0$ from it, *i.e.*

$$\boldsymbol{p}(0) \in \boldsymbol{\mathcal{S}}(0) \implies \boldsymbol{p}(t) \in \boldsymbol{\mathcal{S}}(t) \quad \forall t \geq 0, \tag{4}$$

where $\boldsymbol{\mathcal{S}}(t)$ is a safety set defined using barrier certificates [21]:

$$\boldsymbol{\mathcal{S}}(t) = \{\boldsymbol{p} \in \mathbb{R}^n \mid B^\lambda(\boldsymbol{p},t) \geq 0\}, \tag{5}$$

$$B^\lambda(\boldsymbol{p},t) \equiv D_{\boldsymbol{\mathcal{O}}}(\boldsymbol{p},t) - \frac{\lambda^2}{2}. \tag{6}$$

III. PROPOSED SOLUTION

Given the nature of the problems, it may not be possible to simultaneously solve both 1 and 2 at a given time if obstacles are in the way of the target path, as explained previously. Thus, we employ a hierarchical approach such that we always solve Problem 2 and, when safe, solve Problem 1.

A. Curve vector field

First, we consider following the target path in an obstacle-free scenario. In this case, we directly employ the previously proposed curve vector field $\boldsymbol{\Phi}(\boldsymbol{p},t)$ from [19]. We use the parametric representation of the path $\boldsymbol{c}(s,t)$, for $s \in [0, 2\pi]$, and then find the curve's *witness point*, which can be interpreted as the "closest point" in the curve to the robot

$$\boldsymbol{c}_*(\boldsymbol{p},t) \in \arg \min_{s \in [0,2\pi]} \|\boldsymbol{p} - \boldsymbol{c}(s,t)\|^2. \tag{7}$$

The witness point is used to compute an Euclidean distance vector according to

$$\boldsymbol{D}_{\boldsymbol{\mathcal{C}}}(\boldsymbol{p},t) = \boldsymbol{p} - \boldsymbol{c}_*(\boldsymbol{p},t). \tag{8}$$

As will be clear later, in this work, we smooth the Euclidean distance vector for the *obstacle avoidance vector field*. One could apply the same idea of smoothing the distance function to the *target curve vector field* from [19]. In this case, the trade-off would be to lose precision in following the curve to ensure a smoothed vector field. However, it is more reasonable to add restrictions to the smoothness of the curve [30] itself if you are willing to lose the curve following accuracy, since the curve $\boldsymbol{c}(s,t)$ is a chosen input of the controller.

As assumed in [19], to build the curve vector field, its parametric representation $\boldsymbol{c}(s,t)$ must be at least differentiable at t and differentiable at s. If we aim to ensure smoothness on the vector field, when differentiating it k times, we will be differentiating $\boldsymbol{c}(s,t)$ $k+1$ times. Therefore, the desired smoothness one wants to impose on the curve vector field is directly achieved by designing a smooth target path.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

B. Smooth distance function

Next, we analyze the issue with the Euclidean distance function to the obstacle set and propose a smooth function. Notice that computing the gradient of $D_{\boldsymbol{\mathcal{O}}}(\boldsymbol{p},t)$ with respect to $\boldsymbol{p}$ yields

$$\frac{\partial D_{\boldsymbol{\mathcal{O}}}}{\partial \boldsymbol{p}} = \left(I - \frac{\partial \boldsymbol{o}_*}{\partial \boldsymbol{p}}\right)(\boldsymbol{p} - \boldsymbol{o}_*). \tag{9}$$

Using the Envelope Theorem [31], if $\boldsymbol{o}_*$ is uniquely defined at a given position $\boldsymbol{p}_i$ and is a local minimizer constrained to the set $\boldsymbol{\mathcal{O}}(t)$, small variations $\delta\boldsymbol{p}_i$ do not affect the optimal solution $\boldsymbol{o}_*$. Therefore, $\frac{\partial \boldsymbol{o}_*}{\partial \boldsymbol{p}}(\boldsymbol{p} - \boldsymbol{o}_*) = \boldsymbol{0}$ and (9) reduces to $\boldsymbol{D}_{\boldsymbol{\mathcal{O}}} \equiv \frac{\partial D_{\boldsymbol{\mathcal{O}}}}{\partial \boldsymbol{p}} = \boldsymbol{p} - \boldsymbol{o}_*$, as used in [20].

However, if $\boldsymbol{p}_i$ moves outside of the local region where $\boldsymbol{o}_*$ is well defined or if $\boldsymbol{p}_i$ lies in a boundary where $\boldsymbol{o}_*$ is multi-valued, the Euclidean distance gradient does not exist. This is consistent with the conditions under which the Envelope Theorem holds, as differentiability depends on the uniqueness and smooth dependence of the optimizer on the parameters.

An alternative is to use a smooth distance function. For example, as in [23]. This function approximates the original distance function while ensuring that the approximation function possesses relevant differentiability properties. Here, we employ a similar idea, but with a different smoothing technique. Unlike [23], our approach allows us to ensure safety, given that the smoothed distance will always be smaller than or equal to the real distance, which means that if the real distance is 0, so is the smoothed distance, as we explore in sequence. This turns out to be a key additional element when considering applications where safety is crucial.

Let $D^h_{\boldsymbol{\mathcal{O}}}(\boldsymbol{p},t)$ be the smooth half-squared distance function defined by

$$D^h_{\boldsymbol{\mathcal{O}}}(\boldsymbol{p},t) \equiv \left(\sum_{i=1}^{N}\left(\frac{\|\boldsymbol{p} - \boldsymbol{o}_i(t)\|^2}{2}\right)^{-\frac{1}{h}}\right)^{-h}. \tag{10}$$

The parameter $h > 0$ is a smoothing parameter. The higher its value, the smoother the approximated function, and the poorer the approximation.

Note that the farther an obstacle point is, the smaller its contribution in (10). In practice, such points far from the robot are negligible, allowing approximations of $D^h_{\boldsymbol{\mathcal{O}}}$ computed with measurements around the robot.

Lemma 1 (Smooth half-squared distance properties)**.** *The smooth half-squared distance function* (10) *has the following properties:*

$$\lim_{h\to+0} D^h_{\boldsymbol{\mathcal{O}}}(\boldsymbol{p},t) = D_{\boldsymbol{\mathcal{O}}}(\boldsymbol{p},t), \tag{11}$$

$$\forall h > 0,\ 0 \leq D^h_{\boldsymbol{\mathcal{O}}}(\boldsymbol{p},t) \leq D_{\boldsymbol{\mathcal{O}}}(\boldsymbol{p},t). \tag{12}$$

Proof. Since $D_{\boldsymbol{\mathcal{O}}}(\boldsymbol{p},t)$ is the minimum, notice that $D_{\boldsymbol{\mathcal{O}}}(\boldsymbol{p},t) \leq 0.5\|\boldsymbol{p} - \boldsymbol{o}_i(t)\|^2\ \ \forall i$. The sum in (10) contains $\left(D_{\boldsymbol{\mathcal{O}}}(\boldsymbol{p},t)\right)^{-\frac{1}{h}}$ such that $\sum_{i=1}^{N}\left(\frac{\|\boldsymbol{p}-\boldsymbol{o}_i(t)\|^2}{2}\right)^{-\frac{1}{h}} \geq \left(D_{\boldsymbol{\mathcal{O}}}(\boldsymbol{p},t)\right)^{-\frac{1}{h}}$. Raising both sides to $-h$ and remembering that $-h < 0$, we find $D^h_{\boldsymbol{\mathcal{O}}}(\boldsymbol{p},t) \leq D_{\boldsymbol{\mathcal{O}}}(\boldsymbol{p},t)$. Additionally, the definition (10) is composed of sums and inverses of positive terms. Thus, leading to $D^h_{\boldsymbol{\mathcal{O}}}(\boldsymbol{p},t) \geq 0$. Combined with the previous result we have that $0 \leq D^h_{\boldsymbol{\mathcal{O}}}(\boldsymbol{p},t) \leq D_{\boldsymbol{\mathcal{O}}}(\boldsymbol{p},t)$.

Now, factoring out $D_{\boldsymbol{\mathcal{O}}}(\boldsymbol{p},t)$ in (10) leads to

$$D^h_{\boldsymbol{\mathcal{O}}}(\boldsymbol{p},t) = D_{\boldsymbol{\mathcal{O}}}(\boldsymbol{p},t)\left(\sum_{i=1}^{N}\left(\frac{\|\boldsymbol{p} - \boldsymbol{o}_i(t)\|^2/2}{D_{\boldsymbol{\mathcal{O}}}(\boldsymbol{p},t)}\right)^{-\frac{1}{h}}\right)^{-h}.$$

Considering the definition of $D_{\boldsymbol{\mathcal{O}}}(\boldsymbol{p},t)$, each term $c_{i,h} = \left(\frac{\|\boldsymbol{p}-\boldsymbol{o}_i(t)\|^2/2}{D_{\boldsymbol{\mathcal{O}}}}\right)^{-\frac{1}{h}}$ inside the sum is such that $0 \leq c_{i,h} \leq 1$ for $h > 0$. As $h \to +0$, the indexes i such that $D_{\boldsymbol{\mathcal{O}}}(\boldsymbol{p},t) = 0.5\|\boldsymbol{p} - \boldsymbol{o}_i(t)\|^2$, which have $c_{i,h} = 1$ and that there exists at least one, maintains being 1 while the others approach 0. Hence, as h approaches $+0$ the term $\sum_{i=1}^{n} c_{i,h}$ is bounded by 1 from below and N from above. Thus, we have $\lim_{h\to+0} D^h_{\boldsymbol{\mathcal{O}}}(\boldsymbol{p},t) = D_{\boldsymbol{\mathcal{O}}}(\boldsymbol{p},t) \cdot \lim_{h\to+0}\left(\sum_{i=1}^{N} c_{i,h}\right)^{-h} = D_{\boldsymbol{\mathcal{O}}}(\boldsymbol{p},t)$. □

Similarly to (5)-(6), let $B^{h,\lambda}(\boldsymbol{p},t)$ be a barrier function in terms of the smoothed distance function, and let $\boldsymbol{\mathcal{S}}^h(t)$ be the smoothed safety set according to

$$B^{h,\lambda}(\boldsymbol{p},t) \equiv D^h_{\boldsymbol{\mathcal{O}}}(\boldsymbol{p},t) - \frac{\lambda^2}{2}, \tag{13}$$

$$\boldsymbol{\mathcal{S}}^h(t) \equiv \{\boldsymbol{p} \in \mathbb{R}^n \mid B^{h,\lambda}(\boldsymbol{p},t) \geq 0\}. \tag{14}$$

Notice that invariance of the smoothed safety set implies invariance of the original safety set. This is a key result that allows us to solve Problem 2 as shown later.

Lemma 2. *If $\boldsymbol{\mathcal{S}}^h(t)$ is invariant, then $\boldsymbol{\mathcal{S}}(t)$ is invariant.*

Proof. Stating that $\boldsymbol{\mathcal{S}}^h(t)$ is invariant means that $B^{h,\lambda}(\boldsymbol{p},t) \geq 0$ and therefore $D^h_{\boldsymbol{\mathcal{O}}} \geq \lambda$. Using Lemma 1 we have $D_{\boldsymbol{\mathcal{O}}} \geq D^h_{\boldsymbol{\mathcal{O}}} \geq \lambda$ which implies that $\boldsymbol{\mathcal{S}}(t)$ is also invariant. □

C. Smooth obstacle vector field

Let $\boldsymbol{\Psi}^h(\boldsymbol{p},t)$ be the avoidance vector field, that will keep the robot inside $\boldsymbol{\mathcal{S}}^h(t)$ (14). Analogously to the definitions from [20], we can define the smoothed distance vector as

$$\boldsymbol{D}^h_{\boldsymbol{\mathcal{O}}}(\boldsymbol{p},t) \equiv \frac{\partial}{\partial \boldsymbol{p}} D^h_{\boldsymbol{\mathcal{O}}}, \tag{15}$$

and the smoothed witness point as

$$\boldsymbol{o}^h_*(\boldsymbol{p},t) \equiv \boldsymbol{p} - \frac{\partial}{\partial \boldsymbol{p}} D^h_{\boldsymbol{\mathcal{O}}}, \tag{16}$$

such that $\boldsymbol{D}^h_{\boldsymbol{\mathcal{O}}}(\boldsymbol{p},t) = \boldsymbol{p} - \boldsymbol{o}^h_*(\boldsymbol{p},t)$.

In order to build the vector field with (15), we consider the following:

Assumption 1. Let $\boldsymbol{\mathcal{D}}(t) = \{\boldsymbol{p} \in \mathbb{R}^n | \boldsymbol{D}^h_{\boldsymbol{\mathcal{O}}}(\boldsymbol{p},t) \neq \boldsymbol{0}\}$ be a set in the workspace such that, for a given time t, the smoothed distance vector is not null. Assume that $\boldsymbol{p}(t) \in \boldsymbol{\mathcal{D}}(t)$.

Since (15) is a smooth function, its zero set is a lower-dimensional manifold that has zero measure. In practice, this does not affect the controller, as shown later in the results.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

To compute the obstacle field, we find the smoothed distance vector with the safe margin λ as

$$\boldsymbol{D}_{\mathcal{O}}^{h,\lambda} \equiv B^{h,\lambda} \frac{\boldsymbol{D}_{\mathcal{O}}^{h}}{\|\boldsymbol{D}_{\mathcal{O}}^{h}\|}. \tag{17}$$

We also need a tangent vector $\boldsymbol{T}_{\mathcal{O}}^{h} \perp \boldsymbol{D}_{\mathcal{O}}^{h}$. This vector defines how the forbidden set will be circumnavigated to be avoided. As used in [20], one option for the tangent vector is to solve the minimization $\arg\min_{\boldsymbol{T}} \frac{1}{2}\|\boldsymbol{T}-\boldsymbol{\Phi}\|^2 \quad \text{s.t.} \quad \boldsymbol{T} \perp \boldsymbol{D}_{\mathcal{O}}^{h}$. The problem with such an approach is that, for any dimension, it can be null when $\boldsymbol{\Phi} \parallel \boldsymbol{D}_{\mathcal{O}}^{h}$. We want this vector $\boldsymbol{T}_{\mathcal{O}}^{h}$ to be always continuous and not null, so that the circulation component in the field $\frac{\boldsymbol{T}_{\mathcal{O}}^{h}}{\|\boldsymbol{T}_{\mathcal{O}}^{h}\|}$ is smooth. However, if n is odd this is impossible due to a topological issue, a consequence of the Hairy Ball Theorem [32], [33].

Let $M \in \mathbb{R}^{n\times n}$ be a matrix. The tangent vector can be defined as the transformation of the Euclidean-like distance vector using M matrix as

$$\boldsymbol{T}_{\mathcal{O}}^{h} \equiv M\boldsymbol{D}_{\mathcal{O}}^{h}, \tag{18}$$

where this transformation must ensure the orthogonality $\boldsymbol{T}_{\mathcal{O}}^{h} \perp \boldsymbol{D}_{\mathcal{O}}^{h}$, which can be expressed as

$$(\boldsymbol{D}_{\mathcal{O}}^{h}) \cdot (M\boldsymbol{D}_{\mathcal{O}}^{h}) = 0, \tag{19}$$

implying that M is skew-symmetric.

To ensure $\|\boldsymbol{T}_{\mathcal{O}}^{h}\| > 0$, we would need to have

$$(M\boldsymbol{D}_{\mathcal{O}}^{h}) \cdot (M\boldsymbol{D}_{\mathcal{O}}^{h}) > 0. \tag{20}$$

which would imply that M is invertible. However, as mentioned previously, both (19)-(20) can be achieved simultaneously only for n even, where the skew-symmetric matrices can be invertible. For n odd, the skew-symmetric matrices M are not invertible, which will lead to $\boldsymbol{T}_{\mathcal{O}}^{h} = \boldsymbol{0}$ for some values of $\boldsymbol{D}_{\mathcal{O}}^{h} \in \mathbb{R}^n$.

This is a limitation of our method that can generate an undefined vector field under specific conditions, and more importantly, a discontinuity.

Assumption 2. We consider that either n is even or that M is projected so that $\boldsymbol{T}_{\mathcal{O}}^{h}$ is never null.

In practice, this is a reasonable assumption. For example, as we experiment later on a flying robot in $n = 3$, we can use

$$M = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{21}$$

as a skew-symmetric matrix that represents a rotation around z-axis. During flight, hardly an obstacle will immediately spawn exactly above or beside the robot such that it will have to circumnavigate it. Usually, the robots approach obstacles in the direction they are flying to, *i.e.,* in the direction of the path. This shows the advantage of our new method when compared to [20]. In addition, for n even, not having null tangent vector issues at all, for n odd, those conditions occur less often than $\boldsymbol{\Phi} \parallel \boldsymbol{D}_{\mathcal{O}}^{h}$.

In the next step, we define the static unit field and feedforward component to deal with dynamical conditions as

$$\boldsymbol{\Psi}_S^h = -G\frac{\boldsymbol{D}_{\mathcal{O}}^{h,\lambda}}{\|\boldsymbol{D}_{\mathcal{O}}^{h,\lambda}\|} + H\frac{\boldsymbol{T}_{\mathcal{O}}^{h}}{\|\boldsymbol{T}_{\mathcal{O}}^{h}\|}, \quad \boldsymbol{\Psi}_T^h = -\frac{\boldsymbol{D}_{\mathcal{O}}^{h}}{\|\boldsymbol{D}_{\mathcal{O}}^{h}\|^2}\frac{\partial D_{\mathcal{O}}^{h}}{\partial t}, \tag{22}$$

where G and H should be defined according to [11], for example

$$G = \frac{2}{\pi}\tan^{-1}(k_G\|\boldsymbol{D}_{\mathcal{O}}^{h,\lambda}\|), k_G > 0, \quad H = \sqrt{1-G^2}. \tag{23}$$

Notice that if $\boldsymbol{D}_{\mathcal{O}}^{h,\lambda} = \boldsymbol{0}$ in (22), then $\lim_{\boldsymbol{D}_{\mathcal{O}}^{h,\lambda}\to 0} -G\boldsymbol{D}_{\mathcal{O}}^{h,\lambda}/\|\boldsymbol{D}_{\mathcal{O}}^{h,\lambda}\| = \boldsymbol{0}$ and (22) still holds.

Finally, the field $\boldsymbol{\Psi}^h(\boldsymbol{p},t)$ is

$$\boldsymbol{\Psi}^h(\boldsymbol{p},t) = \eta(\boldsymbol{p},t)\boldsymbol{\Psi}_S^h(\boldsymbol{p},t) + \boldsymbol{\Psi}_T^h(\boldsymbol{p},t), \tag{24}$$

where

$$\eta = -\boldsymbol{\Psi}_S^h \cdot \boldsymbol{\Psi}_T^h + \sqrt{(\boldsymbol{\Psi}_S^h \cdot \boldsymbol{\Psi}_T^h)^2 + v_r^2 - \|\boldsymbol{\Psi}_T^h\|^2}. \tag{25}$$

Similarly to the vast majority of dynamic obstacle avoidance methods [20], our method requires an estimation of the obstacles' motion. This appears in our method in equation (22) with the term $\frac{\partial D_{\mathcal{O}}^{h}}{\partial t}$. We do not tackle the obstacle motion estimation problem in this work. Instead, we compute this term by means of historical data of the robots local measurements.

Also similar to related works [19], [20] we consider a limited obstacle motion. It is reasonable to not be able to avoid obstacles or follow curves that move faster than the robot. This is captured in the form of the following assumption.

Assumption 3. Consider that $\boldsymbol{\Psi}_T^h$ from (22) is limited in norm by the desired velocity v_r, $\|\boldsymbol{\Psi}_T^h\| \leq v_r$.

In practice, if Assumption 3 does not hold, one can set $\eta = 0$ and this will push the robot to compensate for this high speed and still avoid the obstacles if the actuators limits allow.

The vector field $\boldsymbol{\Psi}^h(\boldsymbol{p},t)$ (24) ensures safety. This intermediate result is described in the following Lemma:

Lemma 3. *Let*

$$C^h(\boldsymbol{p},\boldsymbol{v},t) = \boldsymbol{D}_{\mathcal{O}}^{h}(\boldsymbol{p},t)\cdot\boldsymbol{v} + \frac{\partial D_{\mathcal{O}}^{h}}{\partial t}, \tag{26}$$

such that $C^h(\boldsymbol{p},\boldsymbol{v},t) = \dot{B}^{h,\lambda}$, with $B^{h,\lambda}$ from (13)*. When the barrier function $B^{h,\lambda}$ is zero, the function C^h evaluated at the smooth obstacle vector field (under Assumptions 1,2 and 3) is also zero,* i.e.,

$$B^{h,\lambda}(\boldsymbol{p},t) = 0 \implies C^h(\boldsymbol{p},\boldsymbol{\Psi}^h,t) = 0, \tag{27}$$

ensuring that the robot stays inside $\boldsymbol{\mathcal{S}}^h$ according to the barrier function framework [21].

Proof. The expression (26) at $(\boldsymbol{p},\boldsymbol{\Psi}^h,t)$ expands to

$$\boldsymbol{D}_{\mathcal{O}}^{h}\cdot\left(\frac{-\eta G\boldsymbol{D}_{\mathcal{O}}^{h,\lambda}}{\|\boldsymbol{D}_{\mathcal{O}}^{h,\lambda}\|} + \frac{\eta H\boldsymbol{T}_{\mathcal{O}}^{h}}{\|\boldsymbol{T}_{\mathcal{O}}^{h}\|} - \frac{\boldsymbol{D}_{\mathcal{O}}^{h}}{\|\boldsymbol{D}_{\mathcal{O}}^{h}\|^2}\frac{\partial D_{\mathcal{O}}^{h}}{\partial t}\right) + \frac{\partial D_{\mathcal{O}}^{h}}{\partial t}.$$

At $B^{h,\lambda} = 0$, $\boldsymbol{D}_{\mathcal{O}}^{h,\lambda} = 0$ and the first term is null. Given that $\boldsymbol{D}_{\mathcal{O}}^{h} \parallel \boldsymbol{D}_{\mathcal{O}}^{h,\lambda} \perp \boldsymbol{T}_{\mathcal{O}}^{h}$, the second term is null. And finally, the third and fourth terms cancel each other out. Thus, leading to $C^h(\boldsymbol{p},\boldsymbol{\Psi}^h,t) = 0$. □

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

D. Combined vector field

Lastly, having both the obstacle and curve fields smooth is not enough, as we also need to ensure continuity and study what happens with the combination and switching between the fields. In this part, we propose a formulation for the combined vector field.

Let $\text{sat}^{a,b} : \mathbb{R} \to [a, b]$ be a function that saturates its input between a and b, as in

$$\text{sat}^{a,b}(x) \equiv \min(b, \max(a, x)). \tag{28}$$

Then, let $\Theta_1 : \mathbb{R} \to [0, 1]$ be a condition based on the barrier function expressed as

$$\Theta_1(B^{h,\lambda}(\boldsymbol{p}, t)) \equiv \text{sat}^{0,1}\left(\frac{B^{h,\lambda}(\boldsymbol{p}, t) - B_{in}}{B_{in0} - B_{in}}\right), \tag{29}$$

where $B_{in0} > B_{in} > 0$ are blending parameters. The function Θ_1, can be interpreted as a distance condition, where it is 1 if the robot is far away from the obstacles $B^{h,\lambda} > B_{in0}$, it is 0 if the robot is very close to the obstacles $B^{h,\lambda} \leq B_{in}$, and smoothly varies in between.

Let $\Theta_2 : \mathbb{R} \to [0, 1]$ be another condition expressed as

$$\Theta_2(C^h(\boldsymbol{p}, \boldsymbol{\Phi}, t)) \equiv \text{sat}^{0,1}\left(k_B C^h(\boldsymbol{p}, \boldsymbol{\Phi}, t)\right), \tag{30}$$

where $k_B > 0$. The function Θ_2, can be interpreted as an approximation condition. It is 0 if the curve vector field points to directions where the smoothed distance vector field decreases and varies smoothly to 1 as those directions align in the other way.

The condition to dictate the safety of following the curve is defined as an *OR* operation between the two conditions defined previously. This is expressed as the sum

$$\Theta \equiv \text{sat}^{0,1}(\Theta_1 + \Theta_2). \tag{31}$$

Finally, let $\boldsymbol{\mathcal{F}}$ be the vector field defined as

$$\boldsymbol{\mathcal{F}} \equiv \Theta\boldsymbol{\Phi} + (1 - \Theta)\boldsymbol{\Psi}^h. \tag{32}$$

The proposed vector field (32) successfully solves the stated problems, under Assumptions 1,2 and 3. In summary, those assumptions guarantee that the smooth distance vector is not null, the corresponding tangent vector is not null, and limited obstacles velocities, respectively. As mentioned, those are reasonable given that in practice the conditions to make those vectors null are edge cases, hardly achieved, and if the obstacles move too fast, there is still an option to push the actuators and they will be avoided if physically feasible.

Theorem 1 (Combined vector field)**.** *Considering the system* $\dot{\boldsymbol{p}} = \boldsymbol{v}$ (1) *where* $\boldsymbol{v}$ *is the control input chosen as the vector field* $\boldsymbol{\mathcal{F}}(\boldsymbol{p}, t)$ (32) *under Assumptions 1,2 and 3: (I) the safety set* $\boldsymbol{\mathcal{S}}(t)$ *is invariant,* i.e., $\boldsymbol{p}(0) \in \boldsymbol{\mathcal{S}}(0)$*, then* $\boldsymbol{p}(t) \in \boldsymbol{\mathcal{S}}(t),\ \forall t > 0$ *solving Problem 2; and (II) the robot follows the target path, solving Problem 1, when the safety conditions allow,* i.e., *when* $\Theta = 1$*, then* $\boldsymbol{p} \to \boldsymbol{\mathcal{C}}(t)$.

Proof. From (26), for this vector field $\boldsymbol{\mathcal{F}}$, we have that

$$\dot{B}^{h,\lambda} = \Theta \boldsymbol{D}^h_{\boldsymbol{\mathcal{O}}} \cdot \boldsymbol{\Phi} + (1 - \Theta)\boldsymbol{D}^h_{\boldsymbol{\mathcal{O}}} \cdot \boldsymbol{\Psi}^h + \frac{\partial D^h_{\boldsymbol{\mathcal{O}}}}{\partial t}. \tag{33}$$

At $B^{h,\lambda} = 0$, we have $\Theta_1 = 0 \implies \Theta = \Theta_2$. Inspecting (33), we conclude that if $\Theta_2 = 0$, then $\dot{B}^h = C^h(\boldsymbol{p}, \boldsymbol{\Psi}^h, t) = 0$ as ensured by Lemma 3. If $\Theta_2 = 1$, then $\dot{B}^h = C^h(\boldsymbol{p}, \boldsymbol{\Phi}, t) \geq \frac{1}{k_B} \geq 0$ as ensured by the value of Θ_2 itself (30). For values of Θ_2 between zero and one, we must analyze further. Using Lemma 3 result $B^{h,\lambda} = 0 \implies C^h(\boldsymbol{p}, \boldsymbol{\Psi}^h, t) = 0$. Hence, we have that

$$B^{h,\lambda} = 0 \implies \frac{\partial D^h_{\boldsymbol{\mathcal{O}}}}{\partial t} = -\boldsymbol{D}^h_{\boldsymbol{\mathcal{O}}} \cdot \boldsymbol{\Psi}^h. \tag{34}$$

Applying (34) in (33) yields

$$\dot{B}^{h,\lambda} = \Theta_2 \boldsymbol{D}^h_{\boldsymbol{\mathcal{O}}} \cdot \boldsymbol{\Phi} - \Theta_2 \boldsymbol{D}^h_{\boldsymbol{\mathcal{O}}} \cdot \boldsymbol{\Psi}^h, \tag{35}$$

as $\Theta_1 = 0$. Using again the result (34) we have that

$$\Theta_2 = k_B\left(\boldsymbol{D}^h_{\boldsymbol{\mathcal{O}}} \cdot \boldsymbol{\Phi} + \frac{\partial D^h_{\boldsymbol{\mathcal{O}}}}{\partial t}\right) = k_B\left(\boldsymbol{D}^h_{\boldsymbol{\mathcal{O}}} \cdot \boldsymbol{\Phi} - \boldsymbol{D}^h_{\boldsymbol{\mathcal{O}}} \cdot \boldsymbol{\Psi}^h\right). \tag{36}$$

Applying (36) in (35) yields

$$\dot{B}^{h,\lambda} = k_B\left(\boldsymbol{D}^h_{\boldsymbol{\mathcal{O}}} \cdot \boldsymbol{\Phi} - \boldsymbol{D}^h_{\boldsymbol{\mathcal{O}}} \cdot \boldsymbol{\Psi}^h\right)^2 \geq 0, \tag{37}$$

ensuring $\boldsymbol{\mathcal{S}}^h(t)$ is invariant according to the barrier function framework [21]. Then, using Lemma 2, we conclude that $\boldsymbol{\mathcal{S}}(t)$ is invariant and the proposed vector field controller is safe by solving Problem 2.

Now, to show that the robot will follow the curve when the collision avoidance conditions are not active, first we recall that if $\Theta = 1$ then $\boldsymbol{\mathcal{F}} = \boldsymbol{\Phi}$. Next, let $V_p : \mathbb{R}^n \to \mathbb{R}$ be a candidate Lyapunov function, given by $V_p(\boldsymbol{D_\mathcal{C}}) = \frac{1}{2}\boldsymbol{D_\mathcal{C}} \cdot \boldsymbol{D_\mathcal{C}}$, with $\boldsymbol{D_\mathcal{C}}$ from (8). According to Proposition 2 of [19], we have the asymptotic stability ensured by the result

$$\dot{V}_p(\boldsymbol{0}) = 0;\ \dot{V}_p(\boldsymbol{D_\mathcal{C}}) < 0,\ \forall \boldsymbol{D_\mathcal{C}} \neq \boldsymbol{0}. \tag{38}$$

□

IV. RESULTS

We validate our method through extensive numerical simulations and real-world experiments. A detailed visualization including recording and animations is available at the attached video[1]. The code used is available at github.com/ArthurHDN/ral2026.

A. Simulations

1) 2D static: We start with a simulation in 2D space with a static path and obstacles. The goals of this simulation are to build a visualization of the behavior of our method and to compare it with the one proposed in [20] to highlight its advantages. The higher the parameter h, the smoother the path. If h is too low, it eventually reduces to a non-smoothed obstacle field. On the other hand, if h is too high, the robot will

[1]Also available at youtu.be/7Fq1ECjI47E

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

circumnavigate the obstacle at a long distance in a circle-like movement.

In this simulation, we used $v_r = 1\text{m/s}, k_G = 2\text{m}^{-1}, \lambda = 0.1\text{m}, B_{in} = 0.05\text{m}^2, B_{in0} = 0.1\text{m}^2, k_B = 10\text{s/m}^2, h = 0.2$. To compare with [20], we used the same parameters that applied, while mapping $D_{in} = \sqrt{2B_{in} + \lambda^2}$ and $D_{in0} = \sqrt{2B_{in0} + \lambda^2}$.

The starting position of the robot is $\boldsymbol{p}_0 = \begin{bmatrix}0.2 & -0.3\end{bmatrix}^T$ m, which is controlled to follow a curve given by $\boldsymbol{r}(s) = \begin{bmatrix}\cos(s) & \sin(s)\end{bmatrix}^T$ m. Two obstacles are in the way, creating a disjoint non-convex and non-smooth forbidden set. To better study the behavior of avoidance, the matrix M is chosen as a counter-clockwise rotation of 90° in 2D.

Figure 1 shows on the left the vector field of [20] and on the right the proposed one. We can see that the previous method achieves both path-following and obstacle avoidance, where the avoidance is close to the obstacle set. Its movement is not as smooth as we can see in the derivative of the vector field. This is solved using the proposed method, providing smoother movement, at the cost of higher avoidance distances to the obstacles.

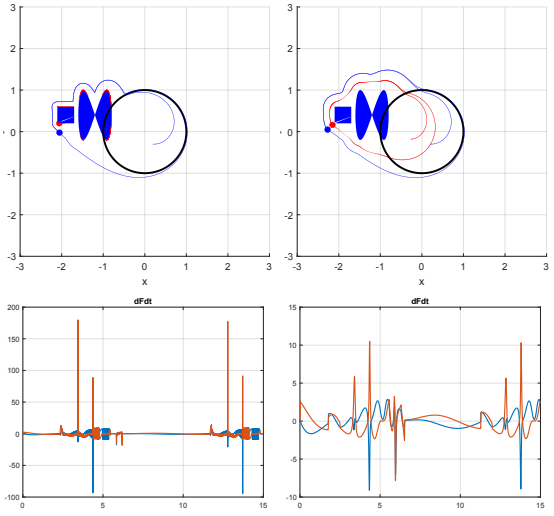

Fig. 1. Comparison of the proposed method (right) with the one from [20] (left) for a 2D static simulation.

2) 2D dynamic: Next, we employ the controller on a similar scenario, but with a dynamic path and obstacles.

The starting position of the robot is $\boldsymbol{p}_0 = \begin{bmatrix}2.2 & -0.3\end{bmatrix}^T$ m, which is controlled to follow a curve given by $\boldsymbol{r}(s,t) = 1.5\begin{bmatrix}\cos(s) + \cos(t/4) & \sin(s)\end{bmatrix}^T$ m. An obstacle is in the way creating a forbidden set. It is a square of 0.8m size at $\begin{bmatrix}\cos(-t/5) & \sin(-t/5)\end{bmatrix}^T$ m that is rotated by t rad counter-clockwise in 2D.

We used the following parameters $h = 0.2, v_r = 1\text{m/s}, k_G = 2\text{m}^{-1}, \lambda = 0.1\text{m}, B_{in} = 0.05\text{m}^2, B_{in0} = 0.1\text{m}^2, k_B = 40\text{s/m}^2$.

Figure 2 again shows a comparison. Both scenarios again achieve path-following and obstacle avoidance. However, the method of [20] slightly avoided the safety set because it did not consider the obstacle velocity under the conditions to follow the curve. Additionally, we again see the non-smoothness of the generated control law that is improved by the proposed controller.

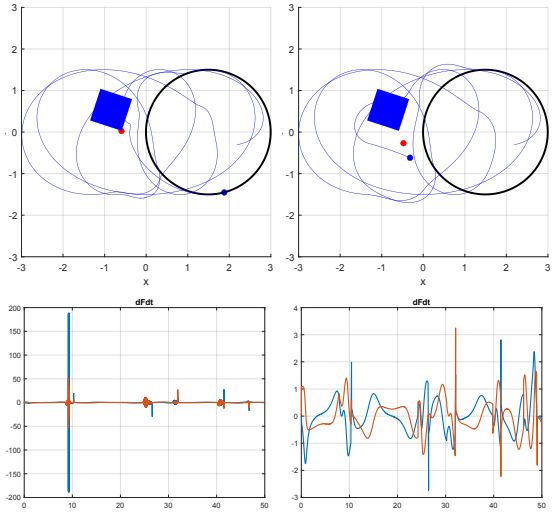

Fig. 2. Comparison of the proposed method (right) with the one from [20] (left) for a 2D dynamic simulation.

3) 3D dynamic: Next, we employ the controller in a $\mathbb{R}^3$ scenario. The starting position of the robot is $\boldsymbol{p}_0 = \begin{bmatrix}-2.2 & -0.3 & -0.1\end{bmatrix}^T$ m, which is controlled to follow a curve given by $\boldsymbol{r}(s,t) = \begin{bmatrix}\cos(s) & \sin(s) & \frac{1}{2}\cos(2s)\cos(t/2) + 1\end{bmatrix}^T$ m. A cubic obstacle is in the way. It has 0.4m side and is centered at $\begin{bmatrix}1 & 0 & 1 + \frac{1}{2}\sin(t/4)\end{bmatrix}^T$ m. Finally, we used M as in (21) and the same controller parameters as the previous simulation.

Figure 3 shows a comparison between the methods. Again, the previous vector field slightly avoided the safety set, and there is a high shattering in a region near $\boldsymbol{\Phi} \| \boldsymbol{D}_o$ given the tangent vector construction. Both issues are solved by our proposed controller.

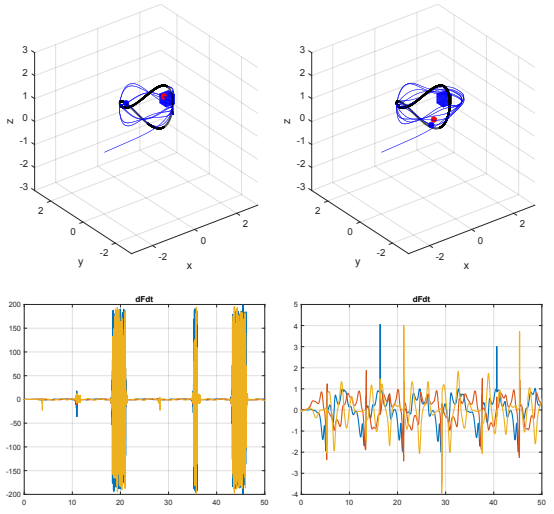

Fig. 3. Comparison of the proposed method (right) with the one from [20] (left) for a 3D dynamic simulation.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

4) Multi-robots in 2D: Next, we employ the controller in a different scenario of multi-agent systems. Although apparently the controller was not designed for those systems, it turns out that it can be applied to them if we express it accordingly.

Note that the set $\boldsymbol{\mathcal{O}}$ is more general than only physical objects in *work space*; it can be seen as a forbidden set in the *configuration space* that we want the system to avoid. For instance, consider an obstacle-free environment with $k = 2$ agents in $\mathbb{R}^2$ where $\boldsymbol{p}_1 = \begin{bmatrix} x_1 & y_1 \end{bmatrix}^T$, $\boldsymbol{p}_2 = \begin{bmatrix} x_2 & y_2 \end{bmatrix}^T$ represents each robot position and $\boldsymbol{p} = \begin{bmatrix} \boldsymbol{p}_1^T & \boldsymbol{p}_2^T \end{bmatrix}^T$ represents the system's configuration. If we also want to avoid inter-agent collisions, $\boldsymbol{\mathcal{O}}$ also contains the points in the configuration space of $\mathbf{p}$ where the agents collide. Then $\boldsymbol{\mathcal{O}} = \{\boldsymbol{p} \in \mathbb{R}^4 : p_1 = p_3, p_2 = p_4\}$.

If we consider $k = 3$ agents, this yields $\boldsymbol{\mathcal{O}} = \boldsymbol{\mathcal{O}}_{1,2} \cup \boldsymbol{\mathcal{O}}_{1,3} \cup \boldsymbol{\mathcal{O}}_{2,3} \cup \boldsymbol{\mathcal{O}}_{1,2,3}$, where $\boldsymbol{\mathcal{O}}_{1,2} = \{\boldsymbol{p} \in \mathbb{R}^6 : p_1 = p_3, p_2 = p_4\}$, $\boldsymbol{\mathcal{O}}_{1,3} = \{\boldsymbol{p} \in \mathbb{R}^6 : p_1 = p_5, p_2 = p_6\}$, $\boldsymbol{\mathcal{O}}_{2,3} = \{\boldsymbol{p} \in \mathbb{R}^6 : p_3 = p_5, p_4 = p_6\}$, and $\boldsymbol{\mathcal{O}}_{1,2,3} = \{\boldsymbol{p} \in \mathbb{R}^6 : p_1 = p_3 = p_5, p_2 = p_4 = p_6\}$.

In summary, $\boldsymbol{\mathcal{O}}$ can also represent the imminent collision point between agents. To avoid collisions in real applications, one needs to set λ such that it encompasses the robot's size. Also notice that $\boldsymbol{\mathcal{O}}$ are linear spaces. Therefore, we can easily sample points in this space around the true nearest point $\boldsymbol{o}_*$ to implement the sum in (10).

Figure 4 shows in the left the simulation of $k = 2$ robots and on the right $k = 3$ robots, where the agents were put to follow the same curve with opposite directions to create inter-agent collision points with $\boldsymbol{\mathcal{O}}$ approximated.

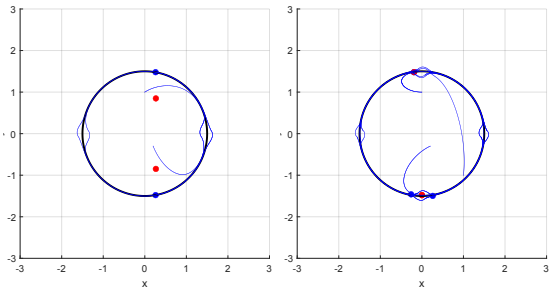

Fig. 4. Visualization of the generated path for multi-agent simulation.

This outlines the generality of the forbidden set avoidance we proposed, allowing us to apply it in different scenarios. On the other hand, the complexity of determining such a set grows with the number of k agents as $\sum_{i=2}^{k} \binom{k}{i} = 2^k - k - 1$. Additionally, this is a centralized controller.

B. Experiments

1) Robotarium: First, we performed experiments using the Robotarium framework [34]. In general, a multi-agent system was controlled to follow curves, while avoiding inter-agent collisions similar to IV-A4 and avoiding going out of the border of the workspace. Each robot has approximately 10cm diameter and we used $\lambda = 12\text{cm}$ to account for the size. The Robotarium framework has a built-in velocity control interface, which directly supports the single-integrator model, commonly used for wheeled ground robots.

Figure 5 shows the results for different numbers of robots and different paths.

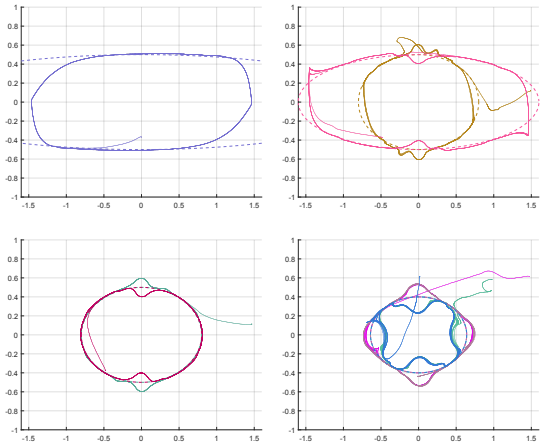

Fig. 5. Visualization of the generated path for Robotarium experiments for different numbers of robots and different paths. Top-left 1 robot, top-right 2 robots, bottom-left 2 robots, and bottom-right 4 robots.

2) Crazyflie: We also performed experiments using Crazyflies nano-quadcopters[2] and the Crazyswarm ROS platform [35]. The pose of the nano-quadcopters was measured using the Optitrack Motion Capture system[3], which is integrated into Crazyswarm and allowed us to input vector field velocity commands. We used the velocity control interface provided by Crazyswarm, which maps our high-level velocity commands to thrust and attitude inputs. The platform supports several inner-loop controllers, with PID being the default option.

Figure 6 shows the evolution of two experiments in which Crazyflies follow a moving target curve (black with shading indicating past positions). The top row presents a single quadcopter (blue trajectory), the moving target curve and a moving foam ball obstacle (red spheres with shading). Each subplot represents a 20-second time window. The bottom row illustrates a second experiment introducing a second quadcopter (magenta trajectory), demonstrating successful inter-agent collision avoidance under similar conditions.

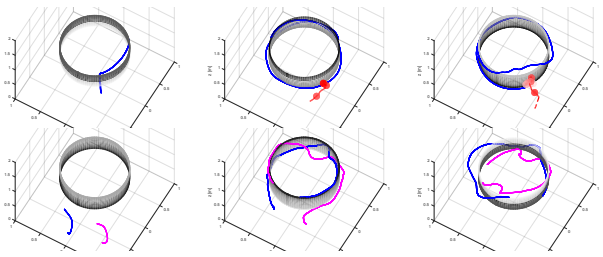

Fig. 6. Crazyflies experiments over time: top row shows single-agent obstacle avoidance; bottom row shows two-agent coordination.

[2]https://www.bitcraze.io/

[3]https://www.optitrack.com/

V. CONCLUSION

In this work, we proposed a novel artificial vector field for safe robot navigation in n-dimensional path-following dynamic tasks. The path following is achieved with a previously proposed vector field. Now, we propose a smooth distance function and a safe and continuous control law, using barrier certificates. Several simulations and real-world experiments are presented to validate the methodology in different scenarios, ranging from simple two-dimensional simulations to multi-agent and quadcopter experiments. The advantages become clear when comparing to a method in the literature, where we achieved continuous control laws and respected the barrier certificates. For future works, we aim to design controllers considering more complex system dynamics.

REFERENCES

[1] H. Durrant-Whyte, "Where am i? a tutorial on mobile vehicle localization," *Industrial Robot: An International Journal*, vol. 21, p. 11–16, Apr. 1994.

[2] B. Siciliano and O. Khatib, *Robotics and the Handbook*, pp. 1–6. Cham: Springer International Publishing, 2016.

[3] Z. Zhang, R. Yang, K. Wu, Z. Xu, J. Liu, L. Mu, Z. Gan, and W. Ding, "Drive in corridors: Enhancing the safety of end-to-end autonomous driving via corridor learning and planning," *IEEE Robotics and Automation Letters*, vol. 10, no. 7, pp. 6728–6735, 2025.

[4] A. Williams and O. Yakimenko, "Persistent mobile aerial surveillance platform using intelligent battery health management and drone swapping," in *2018 4th International Conference on Control, Automation and Robotics (ICCAR)*, pp. 237–246, 2018.

[5] V. Duggal, M. Sukhwani, K. Bipin, G. S. Reddy, and K. M. Krishna, "Plantation monitoring and yield estimation using autonomous quadcopter for precision agriculture," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5121–5127, 2016.

[6] M. Rabah, A. Rohan, S. A. S. Mohamed, and S.-H. Kim, "Autonomous moving target-tracking for a UAV quadcopter based on fuzzy-pi," *IEEE Access*, vol. 7, pp. 38407–38419, 2019.

[7] S. E. Eid and S. Sham Dol, "Design and development of lightweight-high endurance unmanned aerial vehicle for offshore search and rescue operation," in *2019 Advances in Science and Engineering Technology International Conferences (ASET)*, pp. 1–5, 2019.

[8] V. R. Miranda, A. Rezende, T. L. Rocha, H. Azpúrua, L. C. Pimenta, and G. M. Freitas, "Autonomous navigation system for a delivery drone," *Journal of Control, Automation and Electrical Systems*, vol. 33, no. 1, pp. 141–155, 2022.

[9] A. M. Rezende, V. R. Miranda, H. N. Machado, A. C. Chiella, V. M. Gonçalves, and G. M. Freitas, "Autonomous system for a racing quadcopter," in *2019 19th International Conference on Advanced Robotics (ICAR)*, pp. 1–6, IEEE, 2019.

[10] W. Yao, H. G. de Marina, B. Lin, and M. Cao, "Singularity-Free Guiding Vector Field for Robot Navigation," *IEEE Transactions on Robotics*, vol. 37, pp. 1206–1221, Aug. 2021.

[11] V. M. Gonçalves, L. C. Pimenta, C. A. Maia, B. C. Dutra, and G. A. Pereira, "Vector fields for robot navigation along time-varying curves in n-dimensions," *IEEE Transactions on Robotics*, vol. 26, no. 4, pp. 647–659, 2010.

[12] V. M. Gonçalves, L. C. Pimenta, C. A. Maia, G. A. Pereira, B. C. Dutra, N. Michael, J. Fink, and V. Kumar, "Circulation of curves using vector fields: actual robot experiments in 2D and 3D workspaces," in *2010 IEEE International Conference on Robotics and Automation*, pp. 1136–1141, IEEE, 2010.

[13] W. Yao, B. Lin, B. D. O. Anderson, and M. Cao, "Guiding vector fields for following occluded paths," *IEEE Transactions on Automatic Control*, vol. 67, no. 8, pp. 4091–4106, 2022.

[14] W. Yao, B. Lin, and M. Cao, "Integrated Path Following and Collision Avoidance Using a Composite Vector Field," in *2019 IEEE 58th Conference on Decision and Control (CDC)*, IEEE, Dec. 2019.

[15] D. A. Lawrence, E. W. Frew, and W. J. Pisano, "Lyapunov Vector Fields for Autonomous Unmanned Aircraft Flight Control," *Journal of Guidance, Control, and Dynamics*, vol. 31, pp. 1220–1229, Sept. 2008.

[16] D. Nelson, D. Barber, T. McLain, and R. Beard, "Vector Field Path Following for Miniature Air Vehicles," *IEEE Transactions on Robotics*, vol. 23, pp. 519–529, June 2007.

[17] A. M. Rezende, V. M. Gonçalves, A. H. Nunes, and L. C. Pimenta, "Robust quadcopter control with artificial vector fields," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6381–6387, IEEE, 2020.

[18] L. A. Pereira, A. H. Nunes, A. M. Rezende, V. M. Gonçalves, G. V. Raffo, and L. C. Pimenta, "Collision-free vector field guidance and MPC for a fixed-wing UAV," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 176–182, IEEE, 2021.

[19] A. M. C. Rezende, V. M. Goncalves, and L. C. A. Pimenta, "Constructive time-varying vector fields for robot navigation," *IEEE Transactions on Robotics*, vol. 38, no. 2, pp. 852–867, 2021.

[20] A. H. D. Nunes, A. M. C. Rezende, G. P. Cruz, G. M. Freitas, V. M. Gonçalves, and L. C. A. Pimenta, "Vector field for curve tracking with obstacle avoidance," in *2022 IEEE 61st Conference on Decision and Control (CDC)*, pp. 2031–2038, 2022.

[21] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *2019 18th European Control Conference (ECC)*, pp. 3420–3431, 2019.

[22] I. Meijer, M. Pantic, H. Oleynikova, and R. Siegwart, "Pushing the limits of reactive navigation: Learning to escape local minima," *IEEE Robotics and Automation Letters*, vol. 10, no. 7, pp. 6792–6799, 2025.

[23] V. M. Gonçalves, A. Tzes, F. Khorrami, and P. Fraisse, "Smooth distances for second-order kinematic robot control," *IEEE Transactions on Robotics*, vol. 40, pp. 2950–2966, 2024.

[24] A. Jahn and L. C. Pimenta, "Sampling Based Path Planning and Vector Fields for Curve Tracking by UAVs," in *2016 XIII Latin American Robotics Symposium and IV Brazilian Robotics Symposium (LARS/SBR)*, IEEE, Oct. 2016.

[25] I. Ko, B. Kim, and F. C. Park, "Randomized path planning on vector fields," *The International Journal of Robotics Research*, vol. 33, pp. 1664–1682, Oct. 2014.

[26] G. A. S. Pereira, S. Choudhury, and S. Scherer, "A framework for optimal repairing of vector field-based motion plans," in *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*, IEEE, June 2016.

[27] G. A. S. Pereira and E. J. R. Freitas, "Navigation of Semi-autonomous Service Robots Using Local Information and Anytime Motion Planners," *Robotica*, vol. 38, pp. 2080–2098, Jan. 2020.

[28] I. B. P. Nascimento, B. S. Rego, L. C. A. Pimenta, and G. V. Raffo, "Nmpc strategy for safe robot navigation in unknown environments using polynomial zonotopes," in *2023 62nd IEEE Conference on Decision and Control (CDC)*, pp. 7100–7105, 2023.

[29] B. E. Jackson, T. Punnoose, D. Neamati, K. Tracy, R. Jitosho, and Z. Manchester, "Altro-c: A fast solver for conic model-predictive control," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7357–7364, 2021.

[30] E. J. Freitas, M. W. Cohen, A. A. Neto, F. G. Guimarães, and L. C. Pimenta, "DE3D-NURBS: A differential evolution-based 3D path-planner integrating kinematic constraints and obstacle avoidance," *Knowledge-Based Systems*, vol. 300, p. 112084, 2024.

[31] P. Milgrom and I. Segal, "Envelope theorems for arbitrary choice sets," *Econometrica*, vol. 70, no. 2, pp. 583–601, 2002.

[32] V. M. Gonçalves, P. Krishnamurthy, A. Tzes, and F. Khorrami, "Control barrier functions with circulation inequalities," *IEEE Transactions on Control Systems Technology*, vol. 32, no. 4, pp. 1426–1441, 2024.

[33] M. Eisenberg and R. D. Guy, "A proof of the hairy ball theorem," *American Mathematical Monthly*, vol. 86, pp. 571–574, 1979.

[34] S. Wilson, P. Glotfelter, L. Wang, S. Mayya, G. Notomista, M. Mote, and M. Egerstedt, "The robotarium: Globally impactful opportunities, challenges, and lessons learned in remote-access, distributed control of multirobot systems," *IEEE Control Systems Magazine*, vol. 40, no. 1, pp. 26–44, 2020.

[35] J. A. Preiss*, W. Hönig*, G. S. Sukhatme, and N. Ayanian, "Crazyswarm: A large nano-quadcopter swarm," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3299–3304, IEEE, 2017.