POLITECNICO DI TORINO
Repository ISTITUZIONALE

A Guiding Vector-Field Algorithm for Path-Following Control of Nonholonomic Mobile Robots

Original
A Guiding Vector-Field Algorithm for Path-Following Control of Nonholonomic Mobile Robots / Kapitanyuk, Y.A., Proskurnikov, A.V., Cao, M.. - In: IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY. - ISSN 1063-6536. - STAMPA. - 26:4(2018), pp. 1372-1385. [10.1109/TCST.2017.2705059]

Availability:
This version is available at: 11583/2726141 since: 2019-05-17T18:53:56Z

Publisher:
Institute of Electrical and Electronics Engineers Inc.

Published
DOI:10.1109/TCST.2017.2705059

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright
IEEE postprint/Author's Accepted Manuscript

©2018 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collecting works, for resale or lists, or reuse of any copyrighted component of this work in other works.

(Article begins on next page)

27 November 2026

A guiding vector field algorithm for path following control of nonholonomic mobile robots

Yuri A. Kapitanyuk, Anton V. Proskurnikov and Ming Cao

***Abstract*—In this paper we propose an algorithm for path-following control of the nonholonomic mobile robot based on the idea of the *guiding vector field* (GVF). The desired path may be an arbitrary smooth curve in its implicit form, that is, a level set of a predefined smooth function. Using this function and the robot's kinematic model, we design a GVF, whose integral curves converge to the trajectory. A nonlinear motion controller is then proposed which steers the robot along such an integral curve, bringing it to the desired path. We establish global convergence conditions for our algorithm and demonstrate its applicability and performance by experiments with wheeled robots.**

***Index Terms*—Path following, vector field guidance, mobile robot, motion control, nonlinear systems**

I. Introduction

Many applications in industrial and mobile robots are built upon the functionality of following accurately a predefined geometric path [1], [2]. Path following is one of the central problems in automatic guidance of mobile robots, such as aerial vehicles [3], underwater [4] and surface [5] marine crafts, wheeled ground robots and autonomous cars [6], [7]. Although the mathematical models of these robots may differ, the principles of their path following control are similar and can be classified in several major categories.

One approach to steering to the desired path is to fix its time-parametrization; the path is thus treated as a function of time or, equivalently, the trajectory of some reference point. Path steering is then reduced to the *reference point tracking* problem that has been extensively studied in control theory and solved for a very broad class of nonlinear systems [8], [9]. An obvious advantage of the trajectory tracking approach is its applicability to a broad range of paths that can be e.g. non-smooth or self-intersecting. However, this method has unavoidable fundamental limitations, especially when dealing with general nonlinear systems. As shown in [10], unstable zero dynamics generally make it impossible to track a reference trajectory with arbitrary predefined accuracy; more precisely, the integral tracking error is uniformly positive independent of the controller design. Furthermore, in practice the robot's motion along the trajectory can be quite "irregular" due to its oscillations around the reference point. It is impossible to guarantee either path following at a precisely constant speed, or even the motion with a perfectly fixed direction. Attempting to keep close to the reference point, the robot may "overtake" it due to unpredictable disturbances and then turn back.

As has been clearly illustrated in the influential paper [11], these performance limitations of trajectory tracking can be removed by carefully designed *path following* algorithms. Unlike the tracking approach, path following control treats the path as a geometric curve rather than a function of time, dealing with its implicit equations or some time-free parametrization. The algorithm thus becomes "flexible to use a timing law as an additional control variable" [11]; this additional degree of freedom allows to maintain a constant forward speed or any other desired speed profile, which is extremely important e.g. for aerial vehicles, where the lifting force depends on the robot's speed. The *dynamic* controller, maintaining the longitudinal speed, is usually separated from the "geometric" controller, steering the robot to the desired path.

A widely used approach to path-following, originally proposed for car-like wheeled robots [6], [12], assumes the existence of the *projection* point, that is, the closest point on a path, and the robot's capability to measure the distance to it (sometimes referred to as the "cross-track error"). The robot's mathematical model is represented in the Serret-Frenet frame, consisting of the tangent and normal vectors to the trajectory at the projection point. This representation allows to design efficient path following controllers for autonomous wheeled vehicles [6], [7], [12] that eliminate the cross-track error and maintain the desired vehicle's speed along the path. Further development of this approach leads to algorithms for the control of complex unmanned vehicles such as cars with multiple trailers [13] and agricultural tractors [14]. Projection-based *sliding mode* algorithms [7], [14] are capable to cope with uncertainties, caused by the non-trivial geometry of the path, lateral drift of the vehicle and actuator saturations.

The necessity to measure the distance to the track imposes a number of limitations on the path-following algorithm. Even if the nearest point is unique, the robot should either be equipped with special sensors [7] or solve real-time optimization problems to find the cross-track error. In general, the projection point cannot be uniquely determined when e.g. the robot passes a self-intersection point of the path or its position is far from the desired trajectory. A possible way

Yuri Kapitanyuk and Ming Cao are with the ENTEG institute at the University of Groningen, The Netherlands. Anton V. Proskurnikov is with the Delft Center for Systems and Control at Delft University of Technology, Delft, The Netherlands, and also with Institute for Problems of Mechanical Engineering of Russian Academy of Sciences (IPME RAS) and ITMO University, St. Petersburg, Russia. E-mail: y.kapitanyuk@ieee.org, anton.p.1982@ieee.org, m.cao@rug.nl

Y. Kapitanyuk and M. Cao acknowledge the partial support from the European Research Council (ERC-StG-307207). The work of A. Proskurnikov is supported by NWO Domain TTW, the Netherlands, under the project TTW#13712 "From Individual Automated Vehicles to Cooperative Traffic Management - predicting the benefits of automated driving through on-road human behavior assessment and traffic flow models (IAVTRM)".

to avoid these difficulties has been suggested in [15] and is referred to as the "virtual target" approach. The Serret-Frenet frame, assigned to the projection point in the algorithm from [6], [12], can be considered as the body frame of a *virtual target vehicle* to be tracked by the real robot. A modification of this approach, offered in [15], allows the virtual target to have its own dynamics, taken as one of the controller's design parameters. The design from [15] is based on the path following controller from [12], using the model representation in the Serret-Frenet frame and taking the geometry of the path into account. However, the controller from [15] implicitly involves target tracking since the frame has its own dynamics. Avoiding the projection problem, the virtual target approach thus inherits disadvantages of the usual target tracking. In presence of uncertainties the robot may slow down and turn back in order to trace the target's position [3].

A guidance strategy of a human helmsman inspired another path following algorithm, referred to as the *line-of-sight* (LOS) method [3], [5], [16] which is primarily used for air and marine crafts. Maintaining the desired speed of the robot, the LOS algorithm steers its heading along the LOS vector which starts at the robot's center of gravity and ends at the *target* point. This target is located ahead of the robot either on the path [3] or on the line, tangent to the path at the projection point [5]. Unlike the virtual target approach, the target is always chosen at a fixed prescribed distance from the robot, referred to as the *lookahead distance*. The maneuvering characteristics substantially depend on the lookahead distance: the shorter the distance is chosen, the more "aggressively" it steers. A thorough mathematical examination of the LOS method has been carried out in the recent paper [5], establishing the uniform semi-global exponential stability (USGES) property.

Differential-geometric methods for invariant set stabilization [8], [9], [17] have given rise to a broad class of "set-based" [18] path following algorithms. Treating the path as a geometric set, the algorithm is designed to make it invariant and attractive (globally or locally). Typically the path is considered as a set where some (nonlinear) output of the system vanishes, and thus the problem of its stabilization boils down to the output regulation problem. To solve it, various linearization techniques have been proposed [17], [19]–[25]. For stabilization of a closed *strictly convex* curve, an elegant passivity-based method has been established in [18].

In this paper we develop a path following strategy, based on the idea of reference *vector field*. Vector field algorithms are widely used in collision-free navigation and extremum seeking problems [26]–[29]; their efficiency in path following problems has been recently demonstrated in [3], [30]–[32]. A vector field is designed such that its integral curves approach the path asymptotically. Steering the robot along the integral curves, the control algorithm drives it to the desired path. Unlike many path following algorithms that guarantee convergence only in a *sufficiently small* vicinity of the desired path, the vector field algorithm guarantees convergence of any trajectory, which does not encounter the "critical" points where the vector field is degenerate. In particular, in any invariant domain without critical points the convergence to the path can be proved.

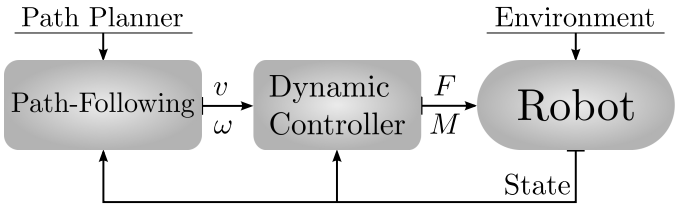

Fig. 1. The structure of motion control systems

For holonomic robots described by a single or double integrator model, a general vector-field algorithm for navigation along a general smooth curve in an n-dimensional space has been discussed in [33]. However, for more realistic nonholonomic vehicles the vector-field algorithms have been studied mainly for straight lines and circular paths [30], [31], [34], where they demonstrate better, in several aspects, performance compared to other approaches [3]. Unlike [3], [30], [31], in this paper we propose and analyze rigorously a vector-field algorithm for guidance of a general nonholonomic robot along a general smooth planar path, given in its implicit form.

The paper is organized as follows. The path following problem is set up in Section II. In Section III we design the guiding vector field and discuss the properties of its integral curves. Section IV offers the path following control algorithm and establishes its main properties. This algorithm is practically validated by experiments with wheeled robots, as discussed in Section V. In Section VI we give a detailed comparison of our algorithm with several alternative approaches to path following controllers design and also discuss its possible extensions.

II. Problem Setup

A widely used technique in path following control is to decompose the controller into an "inner" and an "outer" feedback loops [35]–[37] as illustrated in Fig. 1. The "inner" *dynamic* controller is responsible for maintaining the vector of generalized velocities (in the planar case, the longitudinal speed and turn rate) by controlling the vector of forces and moments. The design of the dynamic controller is based on the robot's mathematical model and may include rejection of external disturbances, e.g. adaptive drift compensators [38]. When the dynamic controller is sufficiently fast and precise, one may consider the speed and turn rate to be the new control inputs, describing thus the robot with a simpler *kinematic* model. The path following algorithm is typically implemented in the "outer" *kinematic* controller, steering the simplified kinematic model to the prescribed path.

The kinematic model of a mobile robot can be *holonomic* or *nonholonomic*. A holonomic robot is not restricted by its angular orientation and able to move in any direction, as exemplified by helicopters, wheeled robots with omni-directional wheels [2] and fully actuated marine vessels at low speeds [16]. For nonholonomic robots only some directions of motion are possible. The simplest model of this type is the *unicycle*, which can move only in the longitudinal direction while the lateral motion is impossible. Examples of robots that can be reduced under certain conditions to the unicycle-type kinematics, include differentially driven wheeled mobile

robots [13], [39], fixed wing aircraft [30], [31], [36], marine vessels at cruise speeds [35] and car-like vehicles, whose rear wheels are not steerable [13], [20], [39]. The most interesting are unicycle models where the speed is restricted to be sufficiently high, making it impossible to reduce the unicycle to a holonomic model via the feedback linearization [40]. The lift force of a fixed-wing UAV, the rudder's yaw moment of a marine craft and the turn rate of a car-like robot depend on the longitudinal speed; at a low speed their manoeuvrability is very limited.

A. The robot's model

In this paper, we consider the unicycle-type model where the longitudinal velocity $u_r > 0$ is a predefined constant

$$\dot{\bar{r}} = \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = u_r \bar{m}(\alpha) \in \mathbb{R}^2, \quad \bar{m}(\alpha) = \begin{bmatrix} \cos\alpha \\ \sin\alpha \end{bmatrix}, \qquad \dot{\alpha} = \omega. \tag{1}$$

Here $\bar{r}$ is the position of the robot's center of gravity C in the inertial Cartesian frame of reference $0XY$, α is the robot's orientation in this frame and $\bar{m}(\alpha)$ is the unit orientation vector (see Fig. 2). The only control input to the system (1) is the angular velocity ω, and hence the system is underactuated.

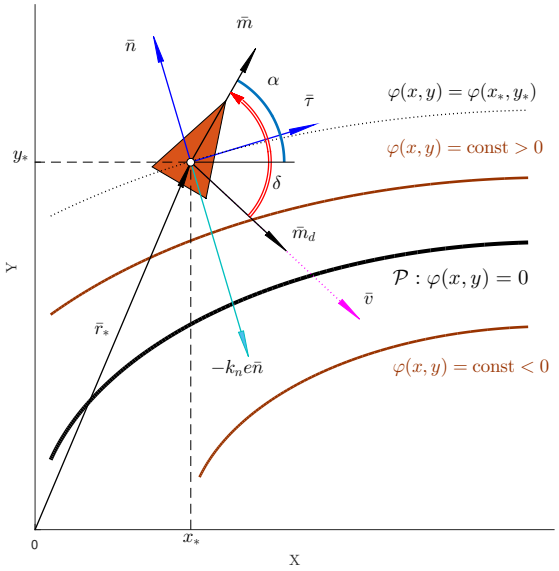

Fig. 2. The robot orientation and level sets of the function $\varphi(x, y)$

B. The description of the desired path

The desired curvilinear path $\mathcal{P}$ is the zero set of a function $\varphi \in C^2(\mathbb{R}^2 \to \mathbb{R})$, i.e. it is described by the implicit equation

$$\mathcal{P} \triangleq \{(x, y) : \varphi(x, y) = 0\} \subset \mathbb{R}^2. \tag{2}$$

The same curve $\mathcal{P}$ may be represented in the implicit form (2) in many ways. The principal restriction imposed by our approach is *regularity* [17]: in some vicinity of $\mathcal{P}$ one has

$$\bar{n}(x, y) \triangleq \nabla\varphi(x, y) = \left[\frac{\partial\varphi(x,y)}{\partial x}; \frac{\partial\varphi(x,y)}{\partial y} \right]^\top \neq 0. \tag{3}$$

As illustrated in Fig. 2, the plane $\mathbb{R}^2$ is covered by the disjoint *level sets* of the function φ, that is, the sets where $\varphi(x, y) = c = const$. If (3) holds, then the vector $\bar{n}(x, y)$ is the *normal* vector to the corresponding level set at the point (x, y). The path $\mathcal{P}$ is one of the level sets, corresponding to $c = 0$; the value $\varphi(x(t), y(t))$ can be considered as a (signed) "distance" from the robot to the path (differing, as usual, from the Euclidean distance), or the *tracking error* [17], [22]. More generally, choosing an arbitrary strictly increasing function $\psi \in C^1(\mathbb{R} \to \mathbb{R})$ with $\psi(0) = 0$ (and thus $\psi(s)s > 0$ for any $s \neq 0$), one may define the tracking error as follows

$$e(x, y) \triangleq \psi[\varphi(x, y)] \in \mathbb{R}. \tag{4}$$

By definition, $e = 0$ if and only if $(x, y) \in \mathcal{P}$. Our goal is to design a path following algorithm, i.e. a causal feedback law $(x(\cdot), y(\cdot), \alpha(\cdot)) \mapsto \omega(\cdot)$, which eliminates the tracking error $|e(t)| \xrightarrow[t\to\infty]{} 0$, bringing thus the robot to the predefined path $\mathcal{P}$. Upon reaching the desired path, the algorithm should "stabilize" the robot on the path, which means that the robot's heading is steered along the tangent vector to the path.

The mapping $\psi(\cdot)$ in (4) is a free parameter of the algorithm. Formally one can get rid of this parameter, replacing φ by the composition $\psi \circ \varphi$. However, it is convenient to distinguish between the path-defining function $\varphi(x, y)$ and the tracking error, depending on the choice of $\psi(s)$. The path representation is usually chosen as simple as possible: for instance, dealing with a straight line, it is natural to choose linear $\varphi(x, y)$, while the circular path is naturally described by $\varphi(x, y) = (x - x_0)^2 + (y - y_0)^2$. At the same time, some mathematical properties of the algorithm, in particular, the region where convergence of the algorithm is guaranteed, depend on the way the tracking error is calculated. As will be discussed, it may be convenient to choose $\psi(\cdot)$ bounded with $|\psi'(s)| \xrightarrow[|s|\to\infty]{} 0$, for instance, $\psi(s) = \arctan s^p$ or $\psi(s) = |s|^p \operatorname{sign} s/(1 + |s|^p)$ with $p \geq 1$. The mentioned two functions, as well as the simplest function $\psi(s) = s$ satisfy the condition, which is henceforth assumed to be valid

$$\sup_{u\in\mathbb{R}} \psi'(\psi^{-1}(u)) < \infty. \tag{5}$$

C. Technical assumptions

Henceforth the following three technical assumptions are adopted, excluding some "pathological" situations. Our first assumption enables one to use of the tracking error $e(x, y)$ as a "signed distance" to the path $\mathcal{P}$ and implies that the *asymptotic* vanishing of the error $e(\bar{r}(t)) \xrightarrow[t\to\infty]{} 0$ entails the convergence to the path in the usual Euclidean metric[1] $\operatorname{dist}(\bar{r}(t), \mathcal{P}) \to 0$.

Assumption 1. *For an arbitrary constant $\varkappa > 0$ one has*

$$\inf\{|e(\bar{r})| : \operatorname{dist}(\bar{r}, \mathcal{P}) \geq \varkappa\} > 0. \tag{6}$$

Our second assumption provides the regularity condition (3) in a sufficiently small vicinity of $\mathcal{P}$.

[1] As usual, the distance from a point $\bar{r}_0$ to the path $\mathcal{P}$ A is $\operatorname{dist}(\bar{r}_0, \mathcal{P}) = \inf\{|\bar{r}_0 - \bar{r}| : \bar{r} \in \mathcal{P}\}$. More generally, the distance between sets A, B is defined as $\operatorname{dist}(A, B) = \inf\{|\bar{r}_1 - \bar{r}_2| : \bar{r}_1 \in A, \bar{r}_2 \in B\}$.

Assumption 2. *The set of* ***critical points****, where $\bar{n}$ vanishes,*

$$\mathcal{C}_0 \triangleq \{(x,y) \in \mathbb{R}^2 : \nabla\varphi(x,y) = 0\},$$

is separated from $\mathcal{P}$ by a positive distance $\operatorname{dist}(\mathcal{C}_0, \mathcal{P}) > 0$.

In the most typical cases where either $\mathcal{P}$ is a closed curve or $\mathcal{C}_0$ is compact, Assumption 2 boils down to the condition $\mathcal{C}_0 \cap \mathcal{P} = \emptyset$. Our final assumption is similar in spirit to Assumption 1 and guarantees that the asymptotic vanishing of the normal vector $\bar{n}(r(t)) \xrightarrow[t\to\infty]{} 0$ is possible only along a trajectory, converging to $\mathcal{C}_0$, i.e. $\operatorname{dist}(\bar{r}(t), \mathcal{C}_0) \to 0$.

Assumption 3. *For an arbitrary constant $\varkappa > 0$ one has*

$$\inf\{|\bar{n}(\bar{r})| : \operatorname{dist}(\bar{r}, \mathcal{C}_0) \geq \varkappa\} > 0. \tag{7}$$

Assumption 3 implies the following useful technical lemma.

Lemma 1. *Consider a Lipschitz vector-function $\bar{r} : [0;\infty) \to \mathbb{R}^2$ such that $\bar{r}(t)$* does not *converge to $\mathcal{C}_0$ as $t \to \infty$, that is, $\limsup_{t\to\infty} d(t) > 0$ where $d(t) = \operatorname{dist}(\bar{r}(t), \mathcal{C}_0)$. Then $\int_0^\infty |\bar{n}(\bar{r}(t))|^p dt = \infty$ for any $p > 0$.*

Proof: It can be easily noticed that $d(\cdot)$ is a Lipschitz function, since as can be easily shown, $|d(t_1) - d(t_2)| \leq |\bar{r}(t_1) - \bar{r}(t_2)| \leq M|t_1 - t_2|\,\forall t_1, t_2 \geq 0$, where $M > 0$ is the Lipschitz constant for $\bar{r}(\cdot)$. Since $\limsup_{t\to\infty} d(t) > 0$, a number $\varepsilon > 0$ and a sequence $t_k \to \infty$ exist such that $d(t_k) \geq 2\varepsilon$ and, therefore, $d(t) \geq \varepsilon$ for $t \in \Delta_k = [t_k; t_k + M^{-1}\varepsilon]$. Passing to subsequences, one may assume without loss of generality that $t_k + M^{-1}\varepsilon < t_{k+1}$ and thus the intervals Δ_k are disjoint. Assumption 3 implies that for sufficiently small $c > 0$ one has $|\bar{n}(\bar{r}(t))| \geq c$ whenever $t \in \Delta_k$. Hence $\int_{\Delta_k} |\bar{n}(\bar{r}(t))|^p dt \geq \varepsilon M^{-1} c^p$ and therefore $\int_0^\infty |\bar{n}(\bar{r}(t))|^p\, dt = \infty$. ∎

III. The Guiding Vector Field and Its Properties

In this section, we construct the Guiding Vector Field (GVF) to be used in the path following control algorithm. We show that the integral curves of this field lead either to the desired path $\mathcal{P}$ or to the critical set $\mathcal{C}_0$. Furthermore, we give efficient criteria, ensuring that the integral curves of the second type are either absent or cover a set of zero measure on $\mathbb{R}^2$.

Besides the normal vector $\bar{n}(x,y) = \nabla\varphi(x,y)$, at each point we consider the *tangent* vector to the level set

$$\bar{\tau}(x,y) = E\bar{n}(x,y), \quad E = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}. \tag{8}$$

If the point is regular (3), the basis $(\bar{\tau}, \bar{n})$ is right-handed oriented[2] (Fig. 2). Our goal is to find such a vector field $\bar{v}(x,y)$, where the absolute tracking error $|e|$ is decreasing along each of its integral curves (unless $e = 0$), and the curves starting on $\mathcal{P}$ do not leave it. We define this vector field by

$$\bar{v}(x,y) \triangleq \bar{\tau}(x,y) - k_n e(x,y)\bar{n}(x,y). \tag{9}$$

The integral curves of the vector field (9) correspond to the trajectories of the autonomous differential equation

$$\frac{d}{dt}\bar{\xi}(t) = \bar{v}(\bar{\xi}(t)) \in \mathbb{R}^2, \quad t \geq 0. \tag{10}$$

[2]In the subsequent constructions, one may replace $\bar{\tau}$ by $-\bar{\tau}$ and E by $-E$, which leads the change of the path following direction.

A. Properties of the integral curves

We notice first that the vector field $\bar{v}$ is C^1-smooth, which implies the *local* existence and uniqueness of solutions of the system (10). Obviously, any point $(x_0, y_0) \in \mathcal{C}_0$ corresponds to the *equilibrium* of (10), which is a trivial single-point integral curve. The uniqueness property implies that non-constant integral curves are free of the critical points; the corresponding solutions, however, may converge to $\mathcal{C}_0$ *asymptotically*. In general, a solution may also escape to infinity in finite time.

The following lemma establishes the principal *dichotomy* property of the integral curves, stating that any curve leads either to the desired path $\mathcal{P}$ or to the critical set $\mathcal{C}_0$.

Lemma 2. *Let $\bar{\xi}(t)$, $t \in [0; t_*)$ where $t_* \leq \infty$ be a maximally prolonged solution to* (10)*. Then two situations are possible*

1) *either* $\operatorname{dist}(\bar{\xi}(t), \mathcal{P}) \xrightarrow[t\to t_*]{} 0$*, that is, the solution converges to the desired path,*
2) *or $t_* = \infty$ and* $\operatorname{dist}(\bar{\xi}(t), \mathcal{C}_0) \xrightarrow[t\to\infty]{} 0$.

Proof: The proof is based on the Lyapunov function

$$V(x,y) = \frac{1}{2}e(x,y)^2 \geq 0. \tag{11}$$

A straightforward computation shows that $\nabla e(\bar{\xi}) = \psi'(\varphi(\bar{\xi}))\,\nabla\varphi(\bar{\xi}) = \psi'(\psi^{-1}(e(\bar{\xi})))|\bar{n}(\bar{\xi})|$ and hence V is non-increasing along the trajectory $\bar{\xi}(t)$ since its derivative is

$$\begin{aligned} \dot{V}(\bar{\xi}) = e(\bar{\xi})\psi'(\psi^{-1}(e(\bar{\xi})))\bar{n}^\top(\bar{\xi})\bar{v}(\bar{\xi}) \overset{(9)}{=} \\ \overset{(9)}{=} -k_n e(\bar{\xi})^2 \psi'(\psi^{-1}(e(\bar{\xi})))|\bar{n}(\bar{\xi})|^2 \leq 0. \end{aligned} \tag{12}$$

In particular, there exists the limit $e_* = \lim_{t\to t_*} |e(\bar{\xi}(t))| \geq 0$. If $e_* = 0$ then, due to Assumption 1, statement 1) holds: the solution converges to $\mathcal{P}$. Suppose now that $e_* > 0$. We are going to prove that statement 2) is valid. Notice first that $\psi'(\psi^{-1}(e(\bar{\xi}(t))))$ is uniformly bounded and positive; the same holds for $V(\bar{\xi}(t))$. The equality (12) thus implies that $\int_0^{t_*} |\bar{n}(\bar{\xi}(t))|^2 dt < \infty$. Since $|v|^2 = (1 + k_n^2 e^2)|\bar{n}|^2$, (10) implies that $\int_0^{t_*} |\dot{\bar{\xi}}(t))|^2 dt < \infty$. If one had $t_* < \infty$, the Cauchy-Schwartz inequality would imply that

$$|\bar{\xi}(T) - \bar{\xi}(0)|^2 = \left(\int_0^T |\dot{\bar{\xi}}(t)|dt\right)^2 \leq t_* \int_0^{t_*} |\dot{\bar{\xi}}(t)|^2 dt \;\; \forall T < t_*,$$

which contradicts the assumption that $\bar{\xi}(t)$ is a maximally prolonged solution, escaping to infinity as $t \to t_*$. Therefore, $t_* = \infty$; statement 2 now follows from Lemma 1. ∎

A natural question arises on how "large" the set of trajectories is, converging to the critical set $\mathcal{C}_0$. In many practical examples, this set is *finite*. This holds, in particular if $\varphi(x,y) \neq 0$ when $|x| + |y|$ is sufficiently large and its *Hessian matrix*

$$H(x,y) = H(x,y)^\top = \begin{bmatrix} \frac{\partial^2}{\partial x^2}\varphi(x,y) & \frac{\partial^2}{\partial x \partial y}\varphi(x,y) \\ \frac{\partial^2}{\partial x \partial y}\varphi(x,y) & \frac{\partial^2}{\partial y^2}\varphi(x,y) \end{bmatrix} \tag{13}$$

is sign-definite at any point $(x_0, y_0) \in \mathcal{C}_0$, i.e. either $H(x_0, y_0) > 0$ or $H(x_0, y_0) < 0$. In this situation $\mathcal{C}_0$ is bounded (and thus compact) and all its points are isolated, which implies that $\mathcal{C}_0$ is finite. If the set $\mathcal{C}_0$ is finite, the

convergence $\operatorname{dist}(\bar\xi(t),\mathcal{C}_0)\xrightarrow[t\to\infty]{}0$, obviously, means that the solution converges to a critical point: $\bar\xi_*\triangleq\lim_{t\to\infty}\bar\xi(t)\in\mathcal{C}_0$.

Definition 1. *Let $\bar\xi_*\in\mathcal{C}_0$ be an equilibrium point of* (10)*. The* stable manifold *of $\bar\xi_*$, denoted by $W(\bar\xi_*)$, is the set of all points $\bar\xi_0$ such that the solution of* (10)*, starting at $\xi(0)=\xi_0$, exists for all $t\ge0$ and $\bar\xi_*\triangleq\lim_{t\to\infty}\bar\xi(t)$.*

We will use the following corollary of the central manifold theorem (see e.g. Theorem 4.1 and Proposition 4.1 in [41]).

Lemma 3. *If both eigenvalues of the Jacobian matrix $J(\bar\xi_*)=\frac{\partial}{\partial\bar\xi}\bar v(\bar\xi_*)$ are strictly unstable $Re\,\lambda_{1,2}J(\bar\xi_*)>0$, then $W(\bar\xi_*)=\{\bar\xi_*\}$. If $J(\bar\xi_*)$ has at least one strictly unstable eigenvalue, then $W(\bar\xi)$ is a set of zero measure.*

Lemma 3 in turn has the following important corollary.

Corollary 1. *Let the set $\mathcal{C}_0$ be finite and for any of its points $\bar\xi_*\in\mathcal{C}_0$ the matrix $e(\bar\xi_*)H(\bar\xi_*)$ has a* negative[3] *eigenvalue. Then the maximally prolonged solution of* (10) *(possibly, existing on finite interval only) converges to $\mathcal{P}$ for almost all initial conditions $\bar\xi(0)$. If $e(\bar\xi_*)H(\bar\xi_*)<0$ for any $\xi_*\in\mathcal{C}_0$, this convergence takes place whenever $\bar\xi(0)\notin\mathcal{C}_0$.*

Proof: A straightforward computation shows that if $\nabla\varphi(\bar\xi_*)=0$ then $J(\bar\xi_*)=(E-k_ne(\bar\xi_*))H(\bar\xi_*)$ and

$$\operatorname{Tr}J(\bar\xi_*)=-k_ne(\bar\xi_*)\operatorname{Tr}H(\bar\xi_*),$$
$$\det J(\bar\xi_*)=(1+k_n^2e(\bar\xi_*)^2)\det H(\bar\xi_*).$$

It can be easily shown that a symmetric 2×2 matrix M is non-negatively definite $M\ge0$ if and only $\det M=\lambda_1(M)\lambda_2(M)\ge0$ and $\operatorname{Tr}M=\lambda_1(M)+\lambda_2(M)\ge0$. Here $\lambda_i(M)$ stand for the eigenvalues of M. Since $eH(e)$ is *not* non-negatively definite, either $\det J<0$ or $\operatorname{Tr}J>0,\det J\ge0$. In both situations, the matrix $J(\bar\xi_*)$ has at least one strictly unstable eigenvalue. Furthermore, if $eH(e)<0$ then $\operatorname{Tr}J>0,\det J>0$, i.e. both eigenvalues of J are strictly unstable. The statement now follows from Lemma 3. ∎

B. The GVF and the "ideal" motion of the robot

The main idea of the path following controller, designed in the next section, is to steer the robot to the integral curve of the field (9). In other words, when the robot is passing a point $\bar r=(x,y)\in\mathbb{R}^2$, its desired orientation is

$$\bar m_d(x,y)=\frac{1}{|\bar v(x,y)|}\bar v(x,y). \tag{14}$$

The field of unit vectors $\bar m_d(x,y)$, henceforth referred to as the *guiding vector field* (GVF), is defined at any regular point (x,y), where $\bar n\ne0$ and thus $|\bar v|=\sqrt{1+k_n^2e^2}|\bar n|\ne0$. Fig. 2 illustrates the relation between the vectors $\bar r$, $\bar m$, $\bar\tau$, $\bar n$, $\bar v$, $\bar m_d$.

Consider the desired motion of the robot, "ideally" oriented along the integral curves of the GVF at any point. Its position vector $\bar r(t)$ obeys the differential equation

$$\dot{\bar r}(t)=u_r\bar m_d(\bar r(t)),\quad t\ge0. \tag{15}$$

[3]Recall that H is a symmetric matrix, so its eigenvalues are real.

The following lemma is a dual of Lemma 2.

Lemma 4. *Let $\bar r(t)$, $t\in[0;t_*)$ where $t_*\le\infty$ be a maximally prolonged solution to* (15)*. Then two situations are possible*

1) *either $\operatorname{dist}(\bar r(t),\mathcal{C}_0)\xrightarrow[t\to t_*]{}0$,*
2) *or $t_*=\infty$ and $\operatorname{dist}(\bar r(t),\mathcal{P})\xrightarrow[t\to\infty]{}0$ that is, the solution converges to the desired path.*

Proof: If $t_*<\infty$ then the limit $\bar r_*=\bar r(t_*-0)=\bar r(0)+\int_0^{t_*}\bar m_d(\bar r(t))dt$ exists and, since the solution is maximally prolonged, one obviously has $|\bar n(\bar r_*)|=0$, i.e. $\bar r_*\in\mathcal{C}_0$ and statement 1 holds. The case of $t_*=\infty$ is considered similar to the proof of Lemma 2. Introducing the Lyapunov function (11), its derivative is shown to be

$$\dot V(\bar r)=u_re(\bar r)\psi'(\psi^{-1}(e(\bar r)))\bar n^\top(\bar r)\bar m_d(\bar r)\overset{(9)}{=}$$
$$\overset{(9)}{=}-\underbrace{\frac{u_rk_ne(\bar r)^2}{\sqrt{1+k_n^2e(\bar r)^2}}\psi'(\psi^{-1}(e(\bar r)))}_{\Psi(e(\bar r))}|\bar n(\bar r)|\le0. \tag{16}$$

There exists the limit $e_*=\lim_{t\to\infty}|e(\bar r(t))|$. If $e_*=0$, statement 2 holds thanks to Assumption 1. Otherwise, $e_*>0$ and $\Psi(e(\bar r(t)))$ is uniformly positive. In view of (16), one has $\int_0^\infty|\bar n(\bar r(t))|dt<\infty$. Lemma 1 entails now statement 1. ∎

Remark 1. *The equation* (16) *implies that $\dot V\approx-k_nV|\bar n|\theta(e)$, where $\theta(e)\to\theta_0>0$ as $e\to0$. Assumptions 2 and 3 imply that $|\bar n(r)|\ge\varkappa$ as $e\approx0$. Therefore, the desired path $\mathcal{P}$ is* locally *exponentially attractive: if $|e(0)|\le\varepsilon$, where ε is sufficiently small, then $|e(t)|\le e^{-k_nu_r\beta t}|e(0)|$, where $\beta>0$ is a constant, depending on $\psi(\cdot)$ and the normal vector $\bar n(x,y)$ in the vicinity of $\mathcal{P}$. The coefficient k_n corresponds to for the attraction to $\mathcal{P}$. In the limit case $k_n=0$, the path $\mathcal{P}$ is not asymptotically stable; the higher $k_n>0$ one chooses, the stronger is the attraction to the path $\mathcal{P}$ in its small vicinity.*

Obviously, the integral curves of (15) are in one-to-one correspondence with non-equilibrium integral curves of (10). Corollary 1 can now be reformulated as follows.

Corollary 2. *Let the set $\mathcal{C}_0$ be finite and for any of its points $\bar\xi_*\in\mathcal{C}_0$, the matrix $e(\bar\xi_*)H(\bar\xi_*)$ has a negative eigenvalue. Then for almost all initial conditions $\bar r(0)$ the solutions of* (15) *can be prolonged up to ∞ and converge to $\mathcal{P}$. If $e(\bar\xi_*)H(\bar\xi_*)<0$ for any $\xi_*\in\mathcal{C}_0$, this holds for any $\bar r(0)\notin\mathcal{C}_0$.*

Note that the conditions of Corollary 2 always hold for *strictly convex* function $\varphi(x,y)$ (that is, $H(x,y)>0$ at any point) since at the critical point (if it exists) the function φ attains its *global minimum*. Therefore, at this point (x_*,y_*) one has $\varphi(x_*,y_*)<0$ and thus $e<0$, which implies that $eH<0$. This is exemplified by the function

$$\varphi(x,y)=\frac{(x-x_0)^2}{a^2}+\frac{(y-y_0)^2}{b^2}-1,$$

defining the elliptic path. Fig. 3 demonstrates the corresponding GVF with the unique critical point at the center of ellipse. This critical point is "repulsive" and no trajectory of (15) converges to it. Fig. 4 illustrates the GVF for the Cassini oval, being the zero level set of the non-convex function

$$(x-x_0)^4+(y-y_0)^4-2a^2((x-x_0)^2-(y-y_0)^2)+a^4-b^4.$$

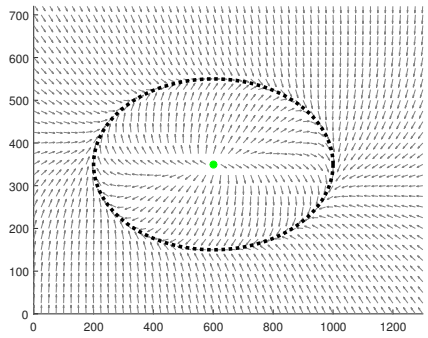

Fig. 3. The GVF for an elliptic path

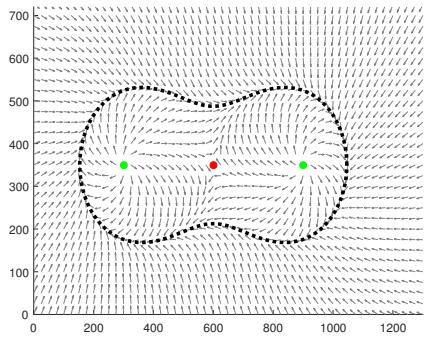

Fig. 4. The GVF for a Cassini oval.

The corresponding set $\mathcal{C}_0$ consists of two "locus points" $(x_0 \pm a, y_0)$ and the "center" (x_0, y_0). At all these points $e < 0$. At the locus points one has $H > 0$, whereas at the center H has one positive and one negative eigenvalue, so the robot potentially can be "trapped" at the center, but the set of corresponding initial conditions has zero measure.

IV. The Path Following Controller Design

In this section, we design the controller, steering the robot to the GVF (14). This controller uses the GVF $\bar{m}_d(x, y)$ at the current robot's position and the function $\omega_d(x, y, \alpha)$, measuring the GVF's "rotation rate" along the robot's trajectory.

A. Preliminaries

We start with the following technical lemma.

Lemma 5. *Let $\bar{m}_d(t) = \bar{m}_d(x(t), y(t))$ stand for the GVF along a trajectory of the robot. Then its derivative $\dot{\bar{m}}_d(t)$ is*

$$\dot{m}_d(t) = -\omega_d(x(t), y(t), \alpha(t))E\bar{m}_d(t), \tag{17}$$

where $\omega_d : \mathbb{R}^3 \to \mathbb{R}$ is continuous and uniquely determined by the functions $\varphi(\cdot)$, $\psi(\cdot)$ and the constant k_n from (9)*.*

Proof: Differentiating the equality $|\bar{m}_d(t)|^2 = 1$, one has that $\dot{\bar{m}}_d(t)^\top \bar{m}_d(t) = 0$, that is, $\dot{\bar{m}}_d(t) \perp \bar{m}_d(t)$. Therefore, the vector $\dot{\bar{m}}_d(t)$ is proportional to the unit vector $Em_d(t)$, so that $\dot{\bar{m}}_d(t) = -\omega_d(t)E\bar{m}_d(t)$ and the scalar multiplier $\omega_d(t)$ can be found from $\omega_d(t) = -\dot{\bar{m}}_d(t)^\top E\bar{m}_d(t)$. It remains to prove that $\omega_d(t)$ in fact depends only on the trajectory $(x(t), y(t), \alpha(t))$, and this dependence is continuous.

Introducing the vector field (9) $\bar{v}(t) = \bar{v}(x(t), y(t))$ and the tracking error $e(t) = e(x(t), y(t))$ along the robot's trajectory, a straightforward computation shows that

$$\begin{gathered}\dot{\bar{m}}_d = \frac{d}{dt}\frac{\bar{v}}{\|\bar{v}\|} = \left(\frac{I_2}{\|\bar{v}\|} - \frac{\bar{v}\bar{v}^\top}{\|\bar{v}\|^3}\right)\dot{\bar{v}},\\ \dot{\bar{v}} = u_r[E - k_n e I_2]H(x, y)\bar{m}(\alpha) - k_n \dot{e}\,\bar{n}(x, y),\\ \dot{e} = u_r\psi'(\varphi(x, y))\,\bar{n}(x, y)^\top \bar{m}(\alpha).\end{gathered} \tag{18}$$

Here I_2 is the 2×2 identity matrix and H is the Hessian (13). Since $\varphi \in C^2$, $\dot{\bar{m}}_d$ and $\bar{m}_d$ continuously depend on the triple $(x(t), y(t), \alpha(t))$, the same holds for $\omega_d = -\dot{\bar{m}}_d^\top E\bar{m}_d$. ■

To clarify the meaning of the function ω_d, suppose for the moment that the robot's speed is $u_r = 1$. If the robot is moving strictly along the integral curves of the GVF, then ω_d is the signed curvature of the robot's trajectory at its current position. In general, ω_d can be treated as the "desired" curvature of the robot's trajectory, which may differ from its real curvature.

B. The algorithm of GVF steering

As was discussed in the foregoing, the idea of the path following algorithm is to steer the robot's orientation along the guiding vector field $\bar{m}_d = \bar{m}_d(x, y)$. We introduce the directed angle $\delta = \delta(x, y, \alpha) \in (-\pi; \pi]$ between $\bar{m}_d$ and $\bar{m}$ (see Fig. 2). The function δ is thus defined at any point (x, y, α) and C^1-smooth at the points where $\bar{m}(\alpha) \neq -\bar{m}_d(x, y)$.

The orientation vector's derivative along the trajectory is

$$\dot{\bar{m}}(t) = \frac{d}{dt}\bar{m}(\alpha(t)) = \omega(t)\begin{bmatrix}-\sin\alpha(t)\\ \cos\alpha(t)\end{bmatrix} = -\omega(t)E\bar{m}(t). \tag{19}$$

On the other hand, $\bar{m}$ may be decomposed (Fig. 2) as

$$\bar{m} = (\cos\delta)\bar{m}_d - (\sin\delta)E\bar{m}_d = [(\cos\delta)I_2 - (\sin\delta)E]\bar{m}_d, \tag{20}$$

At any point where $\delta(t) < \pi$ and thus $\dot{\delta}(t)$ exists, one obtains

$$\begin{aligned}-\omega E\bar{m} &\overset{(14)}{=} \dot{\bar{m}} = -\dot{\delta}\underbrace{[(\sin\delta)I_2 + (\cos\delta)E]\bar{m}_d}_{=E\bar{m}} +\\ &+ [(\cos\delta)I_2 - (\sin\delta)E]\dot{\bar{m}}_d \overset{(17)}{=} -\dot{\delta}E\bar{m} -\\ &- \omega_d\underbrace{[(\cos\delta)I_2 - (\sin\delta)E]E\bar{m}_d}_{=E\bar{m}} = -(\dot{\delta} + \omega_d)E\bar{m},\end{aligned}$$

entailing the following principal relation between δ, ω and ω_d

$$\dot{\delta} = \omega - \omega_d. \tag{21}$$

Furthermore, at *any* time the following equality is valid

$$\frac{d}{dt}\sin\delta = -\frac{d}{dt}\bar{m}^\top E\bar{m}_d = (\omega - \omega_d)\cos\delta. \tag{22}$$

When $\delta(t_0) = \pi$ and $\omega(t) - \omega_d(t) \leq 0$ for $t \approx t_0$, equation (22) entails that $\sin\delta \geq 0$ and hence $\delta = \frac{\pi}{2} + \arcsin(\sin\delta)$ for $t \in [t_0; t_0 + \varepsilon)$, where $\varepsilon > 0$ is sufficiently small. In this situation,

the function $\delta(x(t),y(t),\alpha(t))$ has the right derivative[4] $\dot\delta = D^+\delta$, satisfying (21) as $t\in[t_0;t_0+\varepsilon)$.

We are now in a position to describe our path-following algorithm, employing the GVF "rotation rate" ω_d and the angle of discrepancy between the GVF and the robot's orientation δ

$$\boxed{\omega(t)=\omega_d(x(t),y(t),\alpha(t))-k_\delta\delta(x(t),y(t),\alpha(t)).} \quad (23)$$

Here $k_\delta>0$ is a constant, determining the convergence rate.

When $\delta(x(t),y(t),\alpha(t))<\pi$, the equality (21) holds and thus

$$\boxed{\dot\delta=\omega-\omega_d=-k_\delta\delta.} \quad (24)$$

Furthermore, even for $\delta(x(0),y(0),\alpha(0))=\pi$ one has $\omega-\omega_d=-k_\delta\delta<0$ as $t\approx t_0$ and hence (24) retains its validity at $t=0$, treating $\dot\delta$ as the right derivative D^+f. Thus, considering $\dot\delta$ as a new control input, the algorithm (21) is equivalent to a very simple proportional controller (24), providing, in particular, that $\delta(x(t),y(t),\alpha(t))<\pi\,\forall t>0$.

C. Local existence and convergence of the solutions

In this subsection we examine the properties of the solutions of the closed-loop system (1), (23), rewritten as follows

$$\begin{gathered}\dot x(t)=u_r\cos\alpha(t),\quad \dot y(t)=u_r\sin\alpha(t),\\ \dot\alpha(t)=\omega_d(x(t),y(t),\alpha(t))-k_\delta\delta(x(t),y(t),\alpha(t)).\end{gathered} \quad (25)$$

The right-hand side of (25) is continuous at any point (x_0,y_0,α_0), where $\bar n(x_0,y_0)\neq0$ and $\delta_0=\delta(x_0,y_0,\alpha_0)<\pi$. However, the discontinuity at the points where $\delta_0=\pi$ makes the usual existence theorem [42] inapplicable. To avoid this problem, we consider the equivalent "augmented" system

$$\begin{gathered}\dot x(t)=u_r\cos\alpha(t),\quad \dot y(t)=u_r\sin\alpha(t),\\ \dot\alpha(t)=\omega_d(x(t),y(t),\alpha(t))-k_\delta\delta_*(t),\\ \dot\delta_*(t)=-k_\delta\delta_*(t).\end{gathered} \quad (26)$$

As was discussed in the previous subsection, any solution of (25) satisfies (26) with $\delta_*(t)=\delta(x(t),y(t),\alpha(t))$, and vice versa: choosing a solution $(x(t),y(t),\alpha(t),\delta_*(t))$, where $\delta_*(0)=\delta(x(0),y(0),\alpha(0))\in(-\pi;\pi]$, one has $\delta_*(t)=\delta(x(t),y(t),z(t))$ for any $t\geq0$ due to (24). Unlike (25), the right-hand side of (26) is a C^1-smooth function of (x,y,α,δ_*) at any point where $\bar n(x,y)\neq0$. The standard existence and uniqueness theorem [42] implies the following lemma.

Lemma 6. *For any point $\zeta_0=(x_0,y_0,\alpha_0)$, such that $\bar n(x_0,y_0)\neq0$, there exists the unique solution $\zeta(t)=(x(t),y(t),\alpha(t))$ with $\zeta(0)=\zeta_0$. Extending this solution to the maximal existence interval $[0;t_*)$, one either has $t_*=\infty$ or $t_*<\infty$ and $(x(t),y(t))\xrightarrow[t\to t_*]{}(x_*,y_*)\in\mathcal{C}_0$.*

Proof: Reducing the Cauchy problem for the closed-loop system (25) to the Cauchy problem for (26), one shows that the solution exists locally and is unique [42]. Let its maximally prolongable solution $(x(t),y(t),\alpha(t))$ be defined on $\Delta_*\stackrel{\Delta}{=}[0;t_*)$ with $t_*<\infty$. Since $|\dot{\bar r}(t)|=u_r$, the limit exists

$$\bar r_*=\lim_{t\to t_*}\bar r(t)=\bar r(0)+\int_0^{t_*}\dot{\bar r}(t)dt.$$

We are going to show that $\bar r_*\in\mathcal{C}_0$, i.e. $\bar n(\bar r_*)=0$. Suppose, on the contrary, that $|\bar n(\bar r_*)|>0$; therefore, $|\bar v(x(t),y(t))|$ is *uniformly* positive on Δ_*. Using (18), this implies that $\dot{\bar n}_d(t)$, and hence $\omega_d(x(t),y(t),\alpha(t))$ and $\omega(t)$ are uniformly bounded on Δ_*. Thus there exists the finite limit

$$\alpha_*=\lim_{t\to t_*}\alpha(t)=\alpha(0)+\int_0^{t_*}\omega(t)dt,$$

enabling one to define the solution at $t=t_*$ and then to prolong it to $[t_*,t_*+\varepsilon)$, i.e. the solution is not maximally prolonged. This contradiction implies that $\bar n(\bar r_*)=0$. ■

Note that if $\delta(0)=0$, that is, the robot was perfectly oriented along the GVF at the starting moment, (21) implies that $\delta(t)\equiv0$ so that the robot follows the integral curve of the GVF. As was shown in Section III, in this "ideal" situation the robot either approaches the desired path $\mathcal{P}$ or is driven to one of the critical points. The latter situation is practically impossible if the conditions of Corollary 2 are valid since the set of integral curves, leading to the set $\mathcal{C}_0$, has zero measure.

One may consider (25) as a system with *slow-fast* dynamics. Informally, the controller (24) provides the exponential convergence of the robot to an integral curve of the GVF; after this "fast" transient process, the robot "slowly" follows this integral curve and approaches the desired trajectory, unless it is "trapped" in a critical point. Ignoring the "fast dynamics", one may suppose that the statement of Lemma 4 remains valid for a general solution of the system (25). This argument, however, is not mathematically rigorous. Recalling the proofs of Lemmas 2 and 4, one may notice that the central argument was the *non-increasing* property of the Lyapunov function (11). Although the deviation of the robot from the integral curve exponentially decreases due to (24), the non-increasing property, in general, fails; as will be shown below, even if the robot is positioned very close to the desired path, the tracking error may increase. However, this effect does not destroy the dichotomy property (any solution converges either to $\mathcal{P}$ or to $\mathcal{C}_0$) under the following assumption, which usually holds in practice, being valid e.g. if $\varphi(\cdot)$ is a polynomial.

Assumption 4. *There exist $\theta\in(0;k_\delta)$ and $C>0$ such that*

$$|\bar n(\bar r)|\leq Ce^{\theta|\bar r|}\quad \text{as } |\bar r|\to\infty. \quad (27)$$

The latter condition can be relaxed in the case of a closed path; however, we adopt it to consider both bounded and unbounded paths in a unified way. Note that if $\varphi(x,y)$ is a polynomial function, then $k_\delta>0$ can be arbitrarily small.

Henceforth all Assumptions 1-4 are supposed to be valid.

The following theorem, extending Lemma 4 to the case where $\delta(0)\neq0$, is our first main result.

Theorem 1. *For any maximally prolonged solution $(x(t),y(t),\alpha(t))$ of (25), defined for $t\in[0;t_*)$, one of the following statements holds:*

1) either $t_*=\infty$ and $\operatorname{dist}(\bar r(t),\mathcal{P})\xrightarrow[t\to\infty]{}0$, or

[4]By definition, the right derivative of a function $f(t)$ at $t=t_0$ (written $f'(t_0+0)$ or $D^+f(t_0)$) is defined by $D^+f(t_0)=\lim\limits_{t\to t_0+0}\frac{f(t)-f(t_0)}{t-t_0}$.

2) $\operatorname{dist}(\bar r(t),\mathcal{C}_0)\to 0$ *as* $t\to t_*$.

In other words, for any initial condition the algorithm drives the robot to either the desired path $\mathcal{P}$ or $\mathcal{C}_0$.

Proof: In the case where $t_*=\infty$ statement 2 holds due to Lemma 6. Suppose that $t_*=\infty$. Differentiating the function (11) along the trajectories, it can be shown that

$$\begin{aligned}\dot V &= u_r e\psi'(\psi^{-1}(e))\bar n^\top \bar m \overset{(20)}{=}\\ &\overset{(20)}{=} u_r e\psi'(\psi^{-1}(e))\bar n^\top\left[\bar m_d\cos\delta - E\bar m_d\sin\delta\right]\overset{(14)}{=}\\ &\overset{(14)}{=}\frac{u_r e\psi'(\psi^{-1}(e))}{|v|}\bar n^\top\left[\bar v\cos\delta - E\bar v\sin\delta\right]\overset{(9)}{=}\\ &\overset{(9)}{=}\frac{u_r e\psi'(\psi^{-1}(e))}{\sqrt{1+k_n^2e^2}}|\bar n|(-k_n e\cos\delta+\sin\delta)=\\ &=\Phi(e)|\bar n|(-k_n e\cos\delta+\sin\delta).\end{aligned}\tag{28}$$

Here $\Phi(e)$ denotes the bounded, in view of (5), function

$$\Phi(e)\triangleq\frac{u_r e\psi'(\psi^{-1}(e))}{\sqrt{1+k_n^2e^2}}.\tag{29}$$

Since $|\sin\delta|\le|\delta|$, Assumption 4 entails that $\int_0^\infty|\Phi(e(t))\bar n(t)\sin\delta(t)|\,dt<\infty$. Notice now that $(-\Phi(e)e)\le 0$ and $\cos\delta(t)>0$ as t becomes sufficiently large. Thus the integral $I=\int_0^\infty(-e\Phi(e)|\bar n|)\cos\delta\,dt$ exists, being either finite or equal to $-\infty$. This implies, thanks to (28), the existence of $\int_0^\infty\dot V\,dt=\lim_{t\to+\infty}V(t)-V(0)$. Since $V\ge 0$, one has $I>-\infty$ and therefore there exists the limit $e_*=\lim_{t\to\infty}|e(x(t),y(t))|$. If $e_*=0$, then statement 1 holds due to Assumption 1. Otherwise, $e\Phi(e)$ is uniformly positive and thus, recalling that $\cos\delta(t)\to 1$ as $t\to\infty$, one obtains that $\int_0^\infty|\bar n(x(t),y(t))|dt<\infty$, which implies statement 2 thanks to Lemma 1. ∎

Corollary 3. *If* $\mathcal{C}_0=\emptyset$*, then for any initial condition the solution of* (25) *is infinitely prolongable and the algorithm solves the path following problem* $\operatorname{dist}(\bar r(t),\mathcal{P})\xrightarrow[t\to\infty]{}0$.

Corollary 3 is applicable to linear mappings $\varphi(x,y)=ax+by+c$ (with $|a|+|b|\ne 0$) and many other functions, e.g. $\varphi(x,y)=y+f(x)$. These functions, however, usually correspond to *unbounded* desired curves, whereas for closed paths the GVF $\bar m_d$ is usually not globally defined.

The experiments show that under the assumptions of Corollary 2 the robot always "evades" the finite set of critical points and converges to the desired trajectory. This looks very natural since after very fast transient dynamics the robot "almost precisely" follows some integral curve, which leads to $\mathcal{P}$ "almost surely". We formulate the following hypothesis.

Hypothesis. *Under the assumptions of Corollary 2, for almost all initial conditions* $(x(0),y(0),\alpha(0))$ *the robot's trajectory* $(x(t),y(t))$ *converges to the desired path* $\mathcal{P}$.

Whereas the proof of this hypothesis remains a challenging problem, it is possible to guarantee the global existence of the solutions and their convergence to the desired path in some broad *invariant set*, free of the critical points. The corresponding result, which does not rely on the assumptions of Corollary 2, is established in the next subsection.

D. An invariant set, free of critical points

In this subsection we give a sufficient condition, guaranteeing that a solution of (25) does not converge to $\mathcal{C}_0$. This criterion requires the initial condition $(x(0),y(0),\alpha(0))$ to belong to some *invariant* set, free of critical points. Similar restrictions arise in most of the path following algorithms; for example, in the projection-based algorithms the convergence can be rigorously proved only in some region of attraction where the projection to the desired curve is well defined [14].

Assumptions 1 and 2 imply the uniform positivity of the error on $\mathcal{C}_0$, that is, the following inequality holds

$$e_c=\inf\{|e(x,y)|:(x,y)\in\mathcal{C}_0\}>0.\tag{30}$$

Consider the following set

$$\mathcal{M}=\{(x,y,\alpha):\bar n\ne 0,|\delta|<\arctan(k_n e_c),|e|<e_c\}.\tag{31}$$

Recall that k_n is the constant parameter from (9) and $\delta=\delta(x,y,\alpha)$ is the angle between the robot's heading and the vector field direction (see Fig. 2). By definition, $\mathcal{M}\cap\mathcal{C}_0=\emptyset$. The following lemma states that in fact $\mathcal{M}$ is an *invariant* set, i.e. any solution starting in $\mathcal{M}$ remains there.

Theorem 2. *Any solution of* (25)*, starting at* $(x(0),y(0),\alpha(0))\in\mathcal{M}$*, does not leave* $\mathcal{M}$*, is infinitely prolongable and satisfies the inequality*

$$|e(t)|\le\max\{|e(0)|,k_n^{-1}\tan\delta(0)\}<e_c.\tag{32}$$

For such a solution, one has $\operatorname{dist}(\bar r(t),\mathcal{P})\xrightarrow[t\to\infty]{}0$*, i.e. the algorithm* (23) *solves the path following problem in* $\mathcal{M}$.

Proof: Consider a solution $(x(t),y(t),\alpha(t))$, starting at $(x(0),y(0),\alpha(0))\in\mathcal{M}$. Due to (24), one has $|\delta(t)|\le|\delta(0)|<\frac{\pi}{2}\,\forall t\ge 0$, and hence $\cos\delta(t)>0$. By noticing that $\Phi(e)e\ge 0$ and thus $|\Phi(e)|=\Phi(e)\operatorname{sign}e$, one has

$$\begin{aligned}-\dot V &\overset{(28)}{=} -u_r|\Phi(e)|\operatorname{sign}e\,|\bar n|(e\cos\delta+\sin\delta)=\\ &=u_r|\Phi(e)||\bar n|(|e|+\operatorname{sign}e\tan\delta)\cos\delta\ge\\ &\ge u_r|\Phi(e)|\,|\bar n|(|e|-|\tan\delta|)\cos\delta.\end{aligned}$$

In particular, $\dot V\le 0$ whenever $|e|\ge|\tan\delta|$.

Notice that if $|e(t)|\ge|\tan\delta(t)|$ for any t (where the solution exists), then, obviously $|e(t)|\le|e(0)|$ and thus (32) holds. Suppose now that $|e(t_0)|<|\tan\delta(t_0)|$ at some $t_0\ge 0$. We are going to show that $|e(t)|\le|\tan\delta(t_0)|\le|\tan\delta(0)|$ for any $t\ge t_0$. Indeed, had we $|e(t_1)|>|\tan\delta(t_0)|$ at some point t_1, there would exist $t_*<t_1$ such that $|e(t_*)|=|\tan\delta(t_0)|$ and $|e(t)|>|\tan\delta(t_0)|$ as $t\in(t_*;t_1]$. Using (24), it can be easily shown that $|\tan\delta(t_0)|\ge|\tan\delta(t)|$ for $t\ge t_0$, and hence $|e(t)|>|\tan\delta(t)|$ is non-increasing when $t\in(t_*;t_1]$, which contradicts the assumption that $|e(t_1)|>|\tan\delta(t_0)|=|e(t_*)|$. We have proved that in both cases 1) and 2) the inequality (32) holds at any point where the solution exists; thus the solution stays in $\mathcal{M}$. By definition (30) of e_c, the vector $\bar r(t)$ cannot converge to $\mathcal{C}_0$ in finite or infinite time, i.e. for the considered solution the statement in Theorem 1 holds. ∎

Remark 2. *The condition* $(x(0),y(0),\alpha(0))\in\mathcal{M}$ *restricts the robot to be "properly" headed in the sense that*

$$|\delta(x(0),y(0),\alpha(0))|<\arctan(k_n e_c)<\pi/2.\tag{33}$$

Since $e_c > 0$, (33) is valid for sufficiently large k_n whenever $|\delta| < \pi/2$. In other words, Theorem 2 guarantees convergence to the path from any starting position with $|e(0)| < e_c$ and $|\delta(0)| < \pi/2$, choosing large k_n. Furthermore, if the desired path $\mathcal{P}$ is closed *and the direction of circulation along it is unimportant, the condition $|\delta(0)| < \pi/2$ can be provided by reverting the vector field $\bar{m}_d \mapsto -\bar{m}_d$ (which corresponds to the replacement of $\varphi \mapsto -\varphi$ and $|\delta| \mapsto \pi - |\delta|$) unless $\delta(0) = \pm\pi/2$ (in practice it is never possible).*

Remark 3. *Even if (33) is violated at the starting time $t = 0$, it obviously holds when $t > t_0 = k_\delta^{-1}[\ln|\delta(0)| - \ln\arctan(k_n e_c)]$ thanks to (24). Thus, if one is able to prove that the solution is prolongable up to t_0 and $|e(x(t_0), y(t_0))| < e_c$, Theorem 2 provides the convergence of the robot's position to the desired path.*

Remark 3 suggests the way to relax the restriction on the initial robot's orientation (33). Since the robot moves at the constant speed $u_r > 0$, it covers the distance $u_r t_0$ until its orientation satisfies (33). The path following is guaranteed if $u_r t_0$ is less than the *viability distance* of the initial position, i.e. the distance from it to the set where $|e| \geq e_c$.

Definition 2. *Given a point $\bar{r}_0 = (x_0, y_0)$ with $|e(\bar{r}_0)| < e_c$, the number $d_0 = \inf\{\mathrm{dist}(\bar{r}_0, \bar{r}) : e(\bar{r}) \geq e_c\} > 0$ is said to be its* viability distance.

Theorem 2 and Remark 3 yield in the following.

Corollary 4. *Let the initial position of the robot $\bar{r}(0) = (x(0), y(0))$ with $e(\bar{r}(0)) < e_c$ have the viability distance $d_0 > 0$. If this viability distance satisfies the condition*

$$d_0 > \frac{u_r}{k_\delta} \ln \frac{|\delta(x(0), y(0), \alpha(0))|}{\arctan(k_n e_c)}, \tag{34}$$

then the solution of (25) gets into the set $\mathcal{M}$ in finite time, and is prolongable up to ∞ satisfying $\mathrm{dist}(\bar{r}(t), \mathcal{P}) \xrightarrow[t\to\infty]{} 0$.

Note that (34) holds for any $\alpha(0)$ under the assumption

$$d_0 > \frac{u_r}{k_\delta} \ln \frac{\pi}{\arctan(k_n e_c)}. \tag{35}$$

The condition (35) gives an estimate for the region, starting in which the robot necessarily converges to the desired path $\mathcal{P}$. Taking the original orientation relative to the field into account, this estimate can be tightened by using (34). Typically d_0 is *uniformly* positive in the vicinity of $\mathcal{P}$ (this holds, for instance, when $\mathcal{P}$ is a closed curve). Thus (35) guarantees the algorithm to converge in the vicinity of $\mathcal{P}$, choosing k_δ/u_r sufficiently large. As was mentioned, practical experiments with natural trajectories, satisfying the assumptions of Corollary 2, show that the robot is always attracted to the desired path, although the mathematical proof of this remains a non-trivial task.

Remark 4. *Although explicit calculation of the viability distance is complicated, conditions (34) and (35) in fact require only its* lower *bound. Such a bound can be explicitly obtained, for instance, if the error is Lipschitz*

$$|\nabla e(\bar{r})| = \psi'(\varphi(\bar{r}))|\bar{n}(\bar{r})| \leq c = const \quad \forall x, y. \tag{36}$$

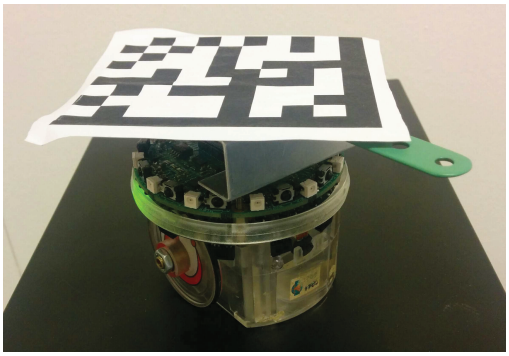

Fig. 5. The E-puck robot with marker on top, used in experiments

If the condition (36) holds, then $|e(\bar{r}) - e(\bar{r}_0)| \leq c|\bar{r} - \bar{r}_0|$ and hence the viability distance of $\bar{r}_0$ is estimated as follows

$$d_0 \geq (e_c - e(x_0, y_0))/c.$$

Condition (36) can often be provided under an appropriate choice of $\psi(\cdot)$. For instance, when $\varphi(x, y) = (x - x_0)^2 + (y - y_0)^2 - R^2$ (circular path), one can choose $\psi(s) = \arctan s$ so that $\psi'(s) = 1/(1 + s^2)$. More generally, if $|\varphi(\bar{r})| \geq C_1|\bar{r}|^\beta$ and $|\nabla\varphi(r)| \leq C_2|\bar{r}|^\gamma$ as $|\bar{r}| \to \infty$, then ∇e is globally bounded; we choose $\psi(s) = \arctan(s^p)$, where $p > 0$ is sufficiently large so that $p - 1 + \gamma \leq 2\beta p$.

V. Experimental validation

In this section the experiments with real robots are reported. In one of these experiments, the desired path is an ellipse, and the other experiments deals with the more sophisticated Cassini oval (see Section III).

We test the results using the E-puck mobile robotic platform [43]. The experimental setup consists of the differential wheeled E-puck robot, a server computer, an overhead camera and a communication module. The robot is identified by a data matrix marker on its top (see Fig. 5). The position of the central point and the orientation of the marker are recognized by a vision algorithm running at a control computer connected to the overhead camera. The available workspace is a planar area of 2.6x2 meters covered by the image of 1280x720 Px (henceforth the acronym Px is used for pixels). For convenience we operate with pixels as coordinates. The PC runs real-time calculation of the control actions based on the pose information of the robot and computes the control inputs for the robot. The results of computation as the desired angular and linear velocities of the robot are translated into the commands for its left and right wheels. The commands from the control computer to the robot are sent via Bluetooth at the fixed frequency of 20 Hz.

We verify the guidance algorithm for two closed trajectories, followed by the robot in the *clockwise* direction. In both cases the forward velocity was set to $u_r = 50$ Px/s, and we set the parameters of the algorithm $k_n = 3$ and $k_\delta = 2$. We choose $\psi(s) = s$, so that $e(x, y) = \varphi(x, y)$. Below we describe the trajectories and show the numerical data from the experiments.

A. Circulation along the ellipse

For our first experiment, we choose the elliptic trajectory, defined by the function

$$\varphi(x,y) = k_s\left(\frac{(x-x_0)^2}{p^2} + \frac{(y-y_0)^2}{q^2} - R^2\right). \qquad (37)$$

Obviously, the level curves of φ are ellipses. The path $\mathcal{P}$ corresponds to the ellipse with semiaxes pR and qR, centered at the point (x_0, y_0).

For the experiment we choose $x_0 = 600$Px, $y_0 = 350$Px, $R = 400$Px, and the semiaxes scale factors $p = 1$, $q = 0.5$ (see Fig. 6). To provide $\varphi(x,y) \in [-5;5]$ in the working area, we choose the scaling factor $k_s = 10^{-5}$. The robots's trajectories (labeled a, b, c and d) under four different initial conditions are shown in Fig. 6. The respective starting conditions are

$$\begin{aligned} a &: (x = 472, y = 311, \alpha = 0.0768),\\ b &: (x = \ 30, y = 555, \alpha = 0.0278),\\ c &: (x = 408, y = 369, \alpha = 2.1515),\\ d &: (x = \ 78, y = 133, \alpha = 4.0419) \end{aligned} \qquad (38)$$

(the coordinates x, y are in pixels, and α is in radians). Fig. 7 illustrates the dynamics of the tracking error $\varphi(x(t), y(t))$.

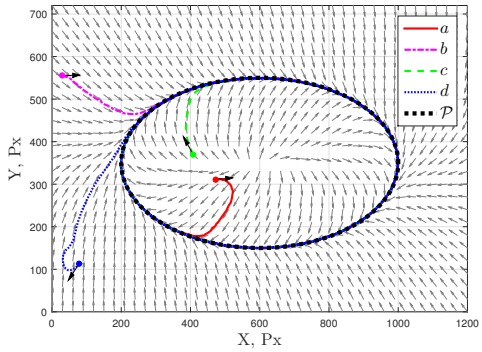

Fig. 6. Elliptical path: the vector field and robot's trajectories.

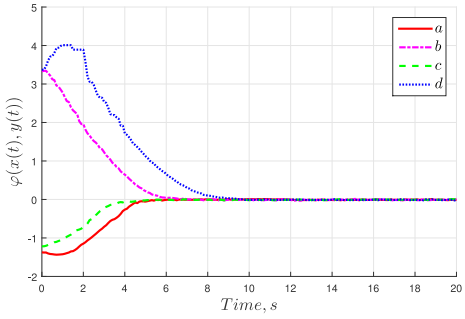

Fig. 7. Elliptical path: the dynamics of the tracking error $e = \varphi(x,y)$.

In this example the set of critical points is $\mathcal{C}_0 = \{(x_0, y_0)\}$, which corresponds to $e(x_0, y_0) = -k_s R^2$ and hence $e_c = R^2$. The set $\{(x,y) : |e(x,y)| < e_c\}$ consists of all points, where

$$0 < \frac{(x-x_0)^2}{p^2} + \frac{(y-y_0)^2}{q^2} < 2R^2. \qquad (39)$$

Theorem 2 guarantees that if the robot starts at any of these points and (33) holds, then it converges to the desired path. Using the geometry of the specific set (39), Remark 3 allows to prove convergence for many other initial conditions: it is clear, for instance, that if the robot starts from some point, lying outside the ellipse and sufficiently far from it, then it enters into the domain (39) with a "proper" orientation, satisfying (33), and thus the problem of path following is solved. The practical experiments, however, show that the robot converges to the path from *any* point, except for the ellipse' center, independent of the initial robot's orientation, whereas the mathematically rigorous proof remains an open problem.

B. Circulation along the Cassini oval

For our second experiment the path chosen is referred to as the *Cassini oval* (Fig. 8) and determined by the function

$$\varphi(x,y) = k_s\left[\left(\Delta x^2 + \Delta y^2\right)^2 - 2q^2\left(\Delta x^2 - \Delta y^2\right) - p^4 + q^4\right],$$
$$\Delta x = (x - x_0),\ \ \Delta y = (y - y_0).$$

We choose here $k_s = 10^{-10}$, $p = 330$Px and $q = 300$Px. The oval is centered at $(x_0, y_0) = (600, 350)$Px.

Fig. 8 illustrates four trajectories (labeled a, b, c and d), corresponding to the initial conditions

$$\begin{aligned} a &: (x = 233, y = 184, \alpha = 2.9287),\\ b &: (x = 106, y = 202, \alpha = 4.2487),\\ c &: (x = 355, y = 343, \alpha = 5.4071),\\ d &: (x = 503, y = 619, \alpha = 0.1022). \end{aligned} \qquad (40)$$

(the coordinates x, y are in pixels, and α is in radians). The corresponding tracking errors are displayed in Fig. 9.

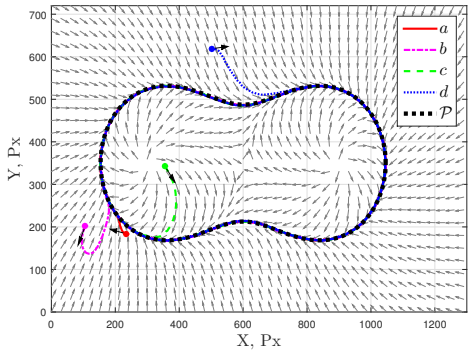

Fig. 8. Cassini oval: the vector field and robot's trajectories.

As discussed in Section III, the set of critical points is $\mathcal{C}_0 = \{(x_0 \pm q, y_0), (x_0, y_0)\}$. If $q^2 < p^2 < 2q^2$ (such inequalities are valid in our example) then it can be checked that $e_c = |e(x_0, y_0)| = p^4 - q^4$, and the set $\{(x,y) : |e(x,y)| < e_c\}$ is a domain, shown in Fig. 10. Theorem 2 guarantees that starting at any of these points with the orientation satisfying (33), the robot converges to the desired path. Similar to the case of the elliptic path, Remark 3 allows to prove convergence for many other initial conditions; experiments show that in fact the robot converges to the desired path from any non-critical point.

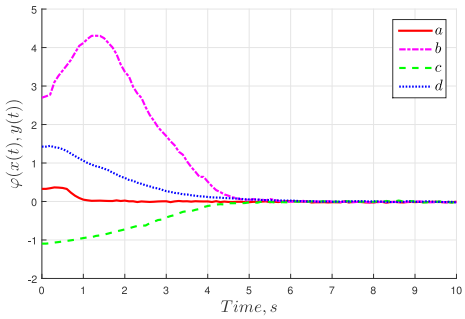

Fig. 9. Cassini oval: the dynamics of the tracking error $e = \varphi(x, y)$.

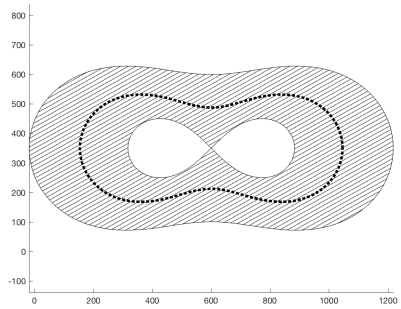

Fig. 10. Cassini oval: the set $\{(x, y) : |e(x, y)| < e_c\}$.

VI. Discussion

In this section, we compare the proposed method with the existing path following control algorithms. Potential extensions of the proposed method are also discussed.

A. The GVF method vs. other path following algorithms

It should be noticed that many approaches to path following, discussed in Introduction, are not applicable to general smooth paths of non-constant curvature and nonholonomic robots; for example, various virtual target approaches [15] typically require to control the robot's velocity. We compare the GVF method with two commonly used algorithms [3]: the projection-based line-of-sight (LOS) method [5], [16] and the "nonlinear guidance law" (NGL) [3], [44].

The LOS approach assumes the existence of the unique *projection* of the robot location $\bar{r}$ onto the path $\mathcal{P}$, that is, the point P closest to $\bar{r}$ in the Euclidean metric. The distance to the path $d = d(\bar{r}, P)$ serves as the tracking (or "cross-track") error. The desired robot's orientation α_{LOS} is the absolute bearing to the unique point P_Δ (Fig. 11), such that the vector $\overrightarrow{PP_\Delta}$ has length $\Delta > 0$ and is tangent to $\mathcal{P}$ at the projection point. The direction of this vector is the desired direction of path following (in our example, the robot circulates along the path clockwise). The controller (23) is replaced by

$$\omega_{LOS} = c(P)u_r - k_{LOS}\left(\alpha - \alpha_{LOS}\right),$$

where $k_{LOS} > 0$ is a constant gain and $c(P)$ is the curvature of the curve $\mathcal{P}$ at the projection point P. The parameters of the algorithm are the *lookahead distance* Δ and the gain k_{LOS}.

The second path following algorithm to be compared with the GVF method is the "nonlinear guidance law" (NGL) [3], [44]. At the current robot position $\bar{r}$, draw a circle of radius R. The circle intersects the path $\mathcal{P}$ at two points q, q'. It is supposed that the robot moves sufficiently close to the desired path and the direction of the path following is fixed, so it is always possible to choose the intersection point, lying *ahead* of the robot on the path $\mathcal{P}$ (in Fig. 11, this is point q). The absolute bearing α_R to this point is the desired direction of motion. The angular velocity controller is designed as

$$\omega_R = -k_R\left(\alpha - \alpha_R\right).$$

The constants $R > 0, k_R > 0$ are the algorithm's parameters.

For the sake of comparison, we will use the same elliptic path, determined by the function (37) and the same parameters of the GVF algorithm as in Subsect. V-A. The initial position and heading of the robot are

$$x = 200; y = 450; \alpha = 0.0278.$$

It is clear that any path following controller is very sensitive to the parameter choice, and varying the parameters one can always make the convergence rate faster or slower. To compare the behavior of different algorithms, one thus needs to choose parameters, providing approximately the same convergence rate; in our situation, the dynamics of the Euclidean distance $d(t) = \mathrm{dist}(\bar{r}(t), \mathcal{P})$ is similar to the GVF method, choosing $k_{LOS} = 2$ and $\Delta = 70$ in the LOS method and $k_R = 2$ and $R = 40$ in the NGL algorithm (Fig. 12). As one can see, the GVF method gives much better transient behavior than the other algorithms, which have visible overshoots. To eliminate the overshoot of the LOS method, one has to increase the lookahead distance, but in this case the convergence to the desired path becomes slower. The NGL method has the largest overshoot; using this method, it is also impossible to eliminate completely the tracking error since the controller has no information about the changing curvature of the path. To decrease the overshoot, one has to increase the radius R; this however amplifies the oscillations in the tracking error d. Thus the GVF method demonstrates better performance than the considered alternative methods.

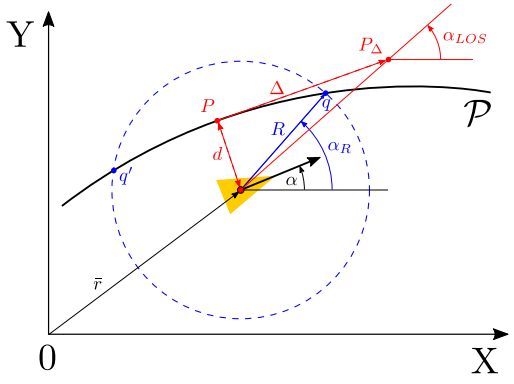

Fig. 11. Geometric objects, employed by the LOS (red) and the NGL (blue) path following algorithms

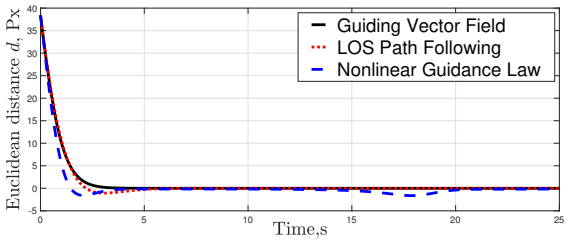

Fig. 12. The dynamic of the euclidean distance d

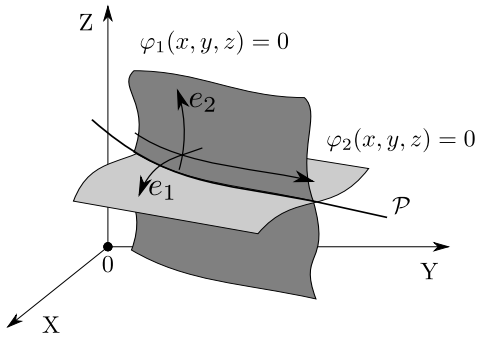

Fig. 13. The desired path as the intersection of two surfaces

B. The previous works on GVF algorithms

The vector-field path following algorithms have been mainly considered for straight lines and circular paths [3], [31]. The GVF designed for these paths is the same as constructed in Section III; the only difference is the path following controller: unlike [31], in this paper we do not consider sliding-mode algorithms and confine ourselves to simple linear controllers. A general approach to the Lyapunov-based GVF design has been suggested in [30]. The desired path is supposed to be the set of the global minima of some predefined Lyapunov function; e.g. for the path (2), our usual Lyapunov function (11) can be chosen. Departing from this Lyapunov function, a broad class of GVF has been suggested, including (14) as a special case. The paper [30] also offers a methodology of proving the convergence of integral curves to the desired path (considered in our Section III) and design of path following controllers. For the case of unicycle-like robot the control algorithm, suggested in [30, Eq. (35)], is equivalent to our controller (24). Describing a general framework, the paper [30] focuses however on some specific problems of flight control, namely, following a *loiter circle* (planar circle at the fixed altitude) or some other planar path, which can be obtained from a loiter circle by a smooth "warping" transformation, preserving the vector field's properties. Many assumptions adopted in [30] are restrictive, e.g. the Lyapunov function should be radially unbounded, excluding straight lines and other unbounded paths from the consideration. The analysis of the closed-loop system in [30] is not fully rigorous as it completely ignores the problem of the solution existence up to ∞, which becomes non-trivial when $\mathcal{C}_0 \neq \emptyset$.

Unlike the previous papers [3], [30], [31], we consider a general smooth path (2), which can be closed or unbounded; for closed trajectories, we do not restrict the Lyapunov function to be radially unbounded. We provide a mathematically rigorous analysis of our algorithm and specify an explicit set of initial conditions, starting at which the robot evades the critical points and converges to the desired trajectory.

C. Further extensions of our GVF method

The results presented in the paper can be extended in several directions. First, the idea of GVF steering can be applied to more general models than the simplest unicycle robot, considered in this paper.

Second, the algorithm can be extended to the three-dimensional case. Some special results in this direction have been reported in [24], [30]. The desired curvilinear path $\mathcal{P}$ in the three dimensional space can be describing as an intersection of two surfaces (Fig. 13)

$$\mathcal{P} \triangleq \{(x,y,z) : \varphi_1(x,y,z) = 0 \wedge \varphi_2(x,y,z) = 0\} \subset \mathbb{R}^3,$$

where $\varphi_i \in C^2(\mathbb{R}^3 \to \mathbb{R}), i = 1,2$. The main design idea is that if the robot simultaneously approaches both surfaces, eventually it will be brought to the prescribed path $\mathcal{P}$. Thus, we may define the guiding vector field by

$$\bar{v} = \bar{\tau} - k_{n1} e_1 \bar{n}_1 - k_{n2} e_2 \bar{n}_2, \quad \bar{\tau} = \bar{n}_1 \times \bar{n}_2$$

where $\bar{n}_i = \nabla\varphi_i$ and $e_i = \varphi_i(x,y,z)$ are the "tracking errors" $(i = 1,2)$. Subsequent design of the controller, steering the robot to the GVF, is similar to the planar case (two angles are controlled instead of a single angle). The technical analysis of this algorithm requires however some extra assumptions of non-degeneracy and is beyond the scope of this paper; the convergence rate of the path following algorithm appears to be very sensitive to the choice of φ_1 and φ_2.

Third, the results can be extended to time-varying vector fields [30], allowing to work with some types of transformations (such as translation, rotation and scaling) of the predefined trajectory and enabling to use the GVF approach for stand-off tracking of slowly moving targets [45]. For instance, it is possible to formulate the problem of moving path following, that is, steering a mobile robot to a curvilinear path attached to a moving frame. Using the change of coordinates, one can transform the original task to the form considered in this paper; however, the longitudinal velocity with respect to the moving frame does not, in general, remain constant. Whereas the case of time-varying speed is uncovered by this paper and requires further analysis, the practical experiments with a fixed-wing UAV [46] have demonstrated satisfactory performance of the proposed path following control algorithm in this situation.

Fourth, it is important to consider external disturbances, unavoidable in practice and leading, in particular, to the lateral drift of the robot. Some preliminary results on using draft compensators in parallel with the path following controller are reported in the works [34], [47].

VII. Conclusions

In this paper, we have proposed a new algorithm for path following control of nonholonomic robots exploiting the idea

of a guiding vector field. Unlike the existing results, the desired path can be an arbitrary smooth curve in its implicit form, i.e. the zero set of a smooth function. We have examined mathematical properties of the algorithm and given global conditions for following asymptotically the desired path. The results have been experimentally validated, using the E-puck wheeled robots.

REFERENCES

[1] T. Kurfess, Ed., *Robotics and Automation Handbook*. CRC Press, 2005.

[2] B. Siciliano and O. Khatib, Eds., *Springer Handbook of Robotics*. Springer, 2008.

[3] P. Sujit, S. Saripalli, and J. Borges Sousa, "Unmanned aerial vehicle path following: A survey and analysis of algorithms for fixed-wing unmanned aerial vehicless," *IEEE Control Systems*, vol. 34, no. 1, pp. 42–59, 2014.

[4] E. Peymania and T. I. Fossen, "Path following of underwater robots using Lagrange multipliers," *Robotics Autonomous Syst.*, vol. 67, pp. 44–52, 2015.

[5] T. I. Fossen and K. Y. Pettersen, "On uniform semiglobal exponential stability (USGES) of proportional line-of-sight guidance laws," *Automatica*, vol. 50, no. 11, pp. 2912 – 2917, 2014.

[6] C. Samson, "Path following and time-varying feedback stabilization of a wheeled mobile robot," in *Proc. of the ICARCV'92*, vol. 13, 1992, pp. 1.1–1.5.

[7] J. Ackermann, J. Guldner, W. Sienel, R. Steinhauser, and V. I. Utkin, "Linear and nonlinear controller design for robust automatic steering," *IEEE Trans. Control Syst. Tech.*, vol. 3, no. 1, pp. 132–143, 1995.

[8] A. Isidori, *Nonlinear Control Systems*, 3rd ed., M. Thoma, E. D. Sontag, B. W. Dickinson, A. Fettweis, J. L. Massey, and J. W. Modestino, Eds. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 1995.

[9] L. Marconi and A. Isidori, "Mixed internal model-based and feedforward control for robust tracking in nonlinear systems," *Automatica*, vol. 36, no. 7, pp. 993 – 1000, 2000.

[10] M. M. Seron, J. H. Braslavsky, P. V. Kokotovic, and D. Q. Mayne, "Feedback limitations in nonlinear systems: From Bode integrals to cheap control," *IEEE Trans. Autom. Control*, vol. 44, no. 4, pp. 829–833, 1999.

[11] A. Aguiar, J. Hespanha, and P. Kokotovic, "Path-following for nonminimum phase systems removes performance limitations," *IEEE Trans. Autom. Control*, vol. 50, no. 2, pp. 234–239, 2005.

[12] A. Micaelli, C. Samson *et al.*, "Trajectory tracking for unicycle-type and two-steering-wheels mobile robots," *Tech. Rep. No. 2097, INRIA*, 1993.

[13] C. Samson, "Control of chained systems application to path following and time-varying point-stabilization of mobile robots," *IEEE Trans. Autom. Control*, vol. 40, no. 1, pp. 64–77, 1995.

[14] A. S. Matveev, M. Hoy, J. Katupitiya, and A. V. Savkin, "Nonlinear sliding mode control of an unmanned agricultural tractor in the presence of sliding and control saturation," *Robotics Autonomous Syst.*, vol. 61, no. 9, pp. 973 – 987, 2013.

[15] D. Soetanto, L. Lapierre, and A. Pascoal, "Adaptive, non-singular path-following control of dynamic wheeled robots," in *Proc. of 42nd IEEE Conf. on Decision and Control*, vol. 2, 2003, pp. 1765–1770.

[16] M. Breivik and T. Fossen, "Principles of guidance-based path following in 2D and 3D," in *Proc. of 44th IEEE Conf. on Decision and Control and European Control Conf. CDC-ECC'05*, 2005, pp. 627–634.

[17] A. Fradkov, I. Miroshnik, and V. Nikiforov, *Nonlinear and Adaptive Control of Complex Systems*, ser. Mathematics and Its Applications. Springer, 1999.

[18] M. I. El-Hawwary and M. Maggiore, "Case studies on passivity-based stabilisation of closed sets," *International Journal of Control*, vol. 84, no. 2, pp. 336–350, 2011.

[19] A. Hladio, C. Nielsen, and D. Wang, "Path following for a class of mechanical systems," *IEEE Trans. Control Syst. Tech.*, vol. 21, no. 6, pp. 2380–2390, 2013.

[20] A. Akhtar, C. Nielsen, and S. Waslander, "Path following using dynamic transverse feedback linearization for car-like robots," *IEEE Trans. Robotics*, vol. 31, no. 2, pp. 269–279, 2015.

[21] S. Kolyubin and A. Shiriaev, "Dynamics-consistent motion planning for underactuated ships using virtual holonomic constraints," in *Proc. of IEEE OCEANS*, 2014, pp. 1–7.

[22] Y. Kapitanyuk and S. Chepinsky, "Control of mobile robot following a piecewise-smooth path," *Gyroscopy and Navigation*, vol. 4, no. 4, pp. 198–203, 2013.

[23] Y. Kapitanyuk, S. Chepinskiy, and A. Kapitonov, "Geometric path following control of a rigid body based on the stabilization of sets," in *Proc. of the 19th IFAC World Congress*, 2014, pp. 7342–7347.

[24] J. Wang, I. A. Kapitaniuk, S. A. Chepinskiy, D. Liu, and A. J. Krasnov, "Geometric path following control in a moving frame," *IFAC-PapersOnLine*, vol. 48, no. 11, pp. 150 – 155, 2015.

[25] R. Gill, D. Kulić, and C. Nielsen, "Spline path following for redundant mechanical systems," *IEEE Trans. Robotics*, vol. 31, no. 6, pp. 1378–1392, 2015.

[26] D. Panagou and V. Kumar, "Cooperative visibility maintenance for leader-follower formations in obstacle environments," *IEEE Trans. Robotics*, vol. 30, no. 4, pp. 831–844, 2014.

[27] M. Hoy, A. Matveev, and A. V. Savkin, "Algorithms for collision-free navigation of mobile robots in complex cluttered environments: a survey," *Robotica*, vol. 33, no. 3, pp. 463–497, 2015.

[28] A. S. Matveev, K. S. Ovchinnikov, and A. V. Savkin, "A method of reactive 3D navigation for a tight surface scan by a nonholonomic mobile robot," *Automatica*, vol. 75, pp. 119–126, 2017.

[29] A. S. Matveev, M. C. Hoy, and A. V. Savkin, "Extremum seeking navigation without derivative estimation of a mobile robot in a dynamic environmental field," *IEEE Trans. Control Syst. Tech.*, vol. 24, no. 3, pp. 1084–1091, 2016.

[30] 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, no. 5, pp. 1220–1229, 2008.

[31] D. Nelson, D. Barber, T. McLain, and R. Beard, "Vector field path following for miniature air vehicles," *IEEE Trans. Robotics*, vol. 23, no. 3, pp. 519–529, 2007.

[32] L. Valbuena Reyes and H. Tanner, "Flocking, formation control, and path following for a group of mobile robots," *IEEE Trans. Control Syst. Tech.*, vol. 23, no. 4, pp. 1268–1282, 2015.

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

[34] R. Beard, J. Ferrin, and J. Humpherys, "Fixed wing UAV path following in wind with input constraints," *IEEE Trans. Control Syst. Tech.*, vol. 22, no. 6, pp. 2103–2117, 2014.

[35] T. I. Fossen, *Guidance and control of ocean vehicles*. John Wiley & Sons Ltd, 1994.

[36] R. W. Beard and T. W. McLain, *Small unmanned aircraft: Theory and practice*. Princeton University Press, 2012.

[37] A. Tsourdos, B. White, and M. Shanmugavel, *Cooperative Path Planning of Unmanned Aerial Vehicles*, ser. Aerospace Series. Wiley, 2010.

[38] T. I. Fossen, K. Y. Pettersen, and R. Galeazzi, "Line-of-sight path following for Dubins paths with adaptive sideslip compensation of drift forces," *IEEE Trans. Control Syst. Tech.*, vol. 23, no. 2, pp. 820–827, 2015.

[39] P. Morin and C. Samson, "Trajectory tracking for non-holonomic vehicles: overview and case study," in *Proc. of 4th Int. Workshop on Robot Motion and Control RoMoCo'04*, 2004, pp. 139–153.

[40] G. Oriolo, A. D. Luca, and M. Vendittelli, "WMR control via dynamic feedback linearization: Design, implementation, and experimental validation," *IEEE Trans. Control Syst. Tech.*, vol. 10, no. 6, pp. 835–852, 2002.

[41] R. Potrie and P. Monzón, "Local implications of almost global stability," *Dynam. Syst.*, vol. 24, no. 1, pp. 109–115, 2009.

[42] H. Khalil, *Nonlinear systems*. Englewood Cliffs, NJ: Prentice-Hall, 1996.

[43] F. Mondada, M. Bonani, X. Raemy, J. Pugh, C. Cianci, A. Klaptocz, S. Magnenat, J.-C. Zufferey, D. Floreano, and A. Martinoli, "The e-puck, a robot designed for education in engineering," in *Proc. of the 9th Conf. on Auton. Robot Syst. and Competitions*, 2009, pp. 59–65.

[44] S. Park, J. Deyst, and J. P. How, "Performance and Lyapunov stability of a nonlinear path following guidance method," *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 6, pp. 1718–1728, 2007.

[45] H. Oh, S. Kim, H. Shin, and A. Tsourdos, "Coordinated standoff tracking of moving target groups using multiple UAVs," *IEEE Trans. Aerospace and Electronic Syst.*, vol. 51, no. 2, pp. 1501–1514, 2015.

[46] Y. A. Kapitanyuk, H. G. de Marina, A. V. Proskurnikov, and M. Cao, "Guiding vector field algorithm for a moving path following problem," *accepted to 20th IFAC 2017 World Congress*, 2017.

[47] H. G. de Marina, Y. A. Kapitanyuk, M. Bronz, G. Hattenberger, and M. Cao, "Guidance algorithm for smooth trajectory tracking of a fixed wing UAV flying in wind flows," *accepted to IEEE Int. Conf. Robotics and Automation (ICRA)*, 2017.