Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 50-1 (2017) 6983–6988

Guiding vector field algorithm for a moving path following problem ⋆

Yuri A. Kapitanyuk * **Hector Garcia de Marina** ** **Anton V. Proskurnikov** ***,****,† **Ming Cao** *

* *Engineering and Technology Institute Groningen University of Groningen, The Netherlands e-mail: {i.kapitaniuk, m.cao}@rug.nl*
** *University of Toulouse, France e-mail: hgdemarina@ieee.org*
*** *Delft Center for Systems and Control Delft University of Technology, The Netherlands e-mail: anton.p.1982@ieee.org*
**** *Institute of Problems of Mechanical Engineering of the Russian Academy of Sciences (IPME RAS), St. Petersburg, Russia*
† *ITMO University, St. Petersburg, Russia*

Abstract: This paper presents a guidance algorithm solving the problem of moving path following, that is, steering a mobile robot to a curvilinear path attached to a moving frame. The nonholonomic robot is described by the unicycle-type model under the influence of some constant exogenous disturbance. The desired path may be an arbitrary smooth curve in its implicit form, that is, a level set of some known smooth function. The path following algorithm employs a guiding vector field, whose integral curves converge to the trajectory. Experiments with a real fixed wing unmanned aerial vehicle (UAV) as well as numerical simulations are presented, illustrating the performance of the proposed path following control algorithm.

Keywords: Path following, guiding vector field, mobile robot, motion control, nonlinear systems

1. INTRODUCTION

Automatic guidance of mobile vehicles is a widely studied problem in robotics. In particular, path following is a typical motion control task which requires an autonomous robotic vehicle to converge to a predefined path and follow it at a prescribed speed (Siciliano and Khatib, 2008). There are many possible approaches to solving this problem: the guidance methods using the Serret-Frenet representation at a projective point (Samson, 1992), virtual target (Soetanto et al., 2003), line-of-sight guidance (Fossen and Pettersen, 2014; Caharija et al., 2015) and differential-geometric methods for invariant sets stabilization (Akhtar et al., 2015). Compared to these approaches, the methods based on the tracking of *vector fields* (Nelson et al., 2007; Lawrence et al., 2008) demonstrate better, in many aspects, performance (Caharija et al., 2015; Sujit et al., 2014). The vector field algorithms are widely used in many applications of robotics, such as path-planning (Pamosoaji and Hong, 2013), obstacle avoidance (Hoy et al., 2015) and extremum seeking (Matveev et al., 2016). The main idea is to design a potential vector field, whose integral lines converge to the desired path. In particular, the general description of the vector field for the path following tasks has been presented in the work (Lawrence et al., 2008); however, even for the simplest model of nonholonomic robots, a comprehensive analysis of this guidance algorithm has been carried out only for special trajectories, such as straight-line and circular paths (Nelson et al., 2007).

The traditional problem setup for path following assumes that the path is fixed in space. However, there are applications for which it is useful to follow a path that is attached to an external reference frame that moves independently with respect to a global inertial coordinate frame (Oliveira et al., 2016). In principal, the ability to modify easily the spatial position and the orientation of the entire path may significantly increase flexibility and usability of mobile robots. However, for a number of practical tasks the mobility of a spatial configuration is the inherent feature. The typical example is the tracking of a ground target by a fixed-wing UAV. In performing such missions, since the fixed-wing UAV generally moves faster than the ground target and also this type of vehicles cannot hover in place, the aircraft requires to follow a special moving pattern. For this, the circular *standoff tracking* concept is introduced (Oh et al., 2013). These circular flight formations are recommended for various target tracking applications, in which for each UAV the maximum altitude flight ensures the maximum visibility and the minimum radius turn ensures the minimum distance to the target at the maximum altitude. Many applications, however, require to use more sophisticated curves than combinations of straight lines

⋆ The work was supported in part by the European Research Council (ERC-StG-307207), the Netherlands Organization for Scientific Research (NWO-vidi-14134) and RFBR, grants 17-08-01728, 17-08-00715 and 17-08-01266

10.1016/j.ifacol.2017.08.1340

and circles. Such tasks include atmospheric monitoring by UAVs (Renzaglia et al., 2016) and the ocean sampling by underwater gliders (Paley et al., 2008). In particular, for the problem of the cloud monitoring (Renzaglia et al., 2016) the desired path can be obtained by enclosing the shape of the cloud by a smooth curve. However, to perform this task successfully, one has to take into account a drift of the cloud due to the influence of the wind. Moreover, because of the presence of wind it becomes necessary to consider impact of disturbance in the robot's model.

The method of path following, presented in this paper, is an extension of the guidance algorithm for nonholonomic robots given in (Kapitanyuk et al., 2016; Garcia de Marina et al., 2016b) to the case where a *general C^2-smooth curvilinear path* is attached to an external reference frame that moves uniformly with respect to a global inertial coordinate frame. The robot's motion is influenced by a constant disturbance. The presented algorithm is based on the idea of following a vector field that converges smoothly to the desired path, where the convergence is global under some assumptions. Instead of considering the Euclidean distance, the notion of error is given by the implicit equation of the desired trajectory, making the tracking task much easier to be implemented.

The paper is organized as follows. The path following problem is formulated in Section 2, and Section 3 offers the vector field algorithm for path following. This algorithm is validated by experiments described in Section 4.

2. PROBLEM STATEMENT

A widely used paradigm in path following control is to decompose the controller into an "inner" and an "outer" feedback loop. We will assume throughout this paper the existence of a low level control system that maintains the robot's constant speed. For the sake of clarity and simplicity we do not consider the 3D model here, and focus on the planar motion. Therefore, we consider the unicycle-type model where the longitudinal speed is a predefined constant $s > 0$:

$$\begin{cases} \dot{\bar{p}} = s\bar{m} + \bar{w}, \\ \dot{\alpha} = u. \end{cases} \tag{1}$$

Here $\bar{p} = [x \ y]^\top \in \mathbb{R}^2$ stands for the position of the robot's center of gravity in the inertial Cartesian frame $\mathcal{O}_N$, α is the orientation of the robot in this frame, $\bar{m} = [\cos(\alpha) \ \sin(\alpha)]^\top$ is the unit orientation vector, $\bar{w} \in \mathbb{R}^2$ is a *constant* vector representing the disturbance in the inertial frame $\mathcal{O}_N$. We assume the angular velocity u to be the only control input to the system.

We also assume the existence of a suitable state estimation scheme that provides (i) the position $\bar{p}$ and its derivative $\dot{\bar{p}}$ with respect to $\mathcal{O}_N$, from e.g. a global positioning system (GPS) signal, (ii) the orientation of the robot α with respect to $\mathcal{O}_N$, which can be obtained from a well calibrated compass in areas far away from the Earth's poles and (iii) the instant speed s in the body-fixed frame.

To focus on the ideas of the path following algorithm, we confine ourselves to the simplest case of an external frame motion with a constant translational velocity. For the sake of clarity and simplicity we do not consider the rotational motion of $\mathcal{O}_T$. The kinematics of the external frame

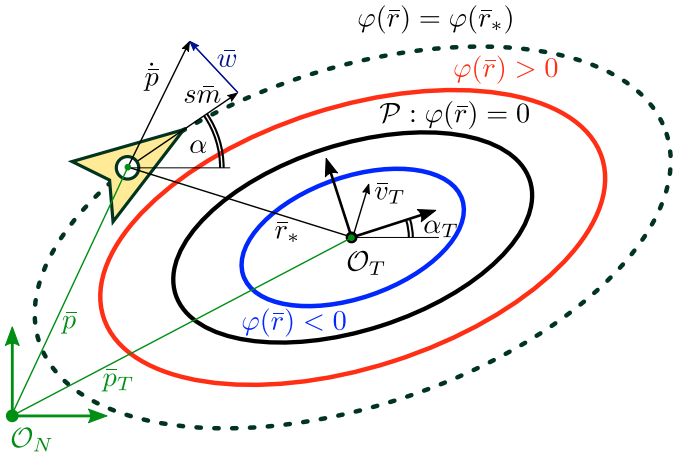

Fig. 1. The robot orientation and level sets of the function $\varphi(\bar{r})$

$\mathcal{O}_T$ with respect to $\mathcal{O}_N$ is represented by the following equations:

$$\begin{cases} \dot{\bar{p}}_T = \bar{v}_T, \\ \dot{\alpha}_T = 0, \end{cases} \tag{2}$$

where $\bar{p}_T = [x_T \ y_T]^\top \in \mathbb{R}^2$ stands for the position of the origin of the external frame $\mathcal{O}_T$ with respect to the frame $\mathcal{O}_N$, $\bar{v}_T \in \mathbb{R}^2$ is the constant vector of the velocity, α_T is the orientation of the external frame $\mathcal{O}_T$ with respect to the frame $\mathcal{O}_N$ and $\dot{\alpha}_T \equiv 0$ is the angular speed of the external frame $\mathcal{O}_T$ with respect to the frame $\mathcal{O}_N$. We assume that $\bar{v}_T$ is known.

The robot's position with respect to the frame $\mathcal{O}_T$ can be defined in the following way

$$\bar{r} = R^\top(\alpha_T)(\bar{p} - \bar{p}_T), \quad R(\alpha_T) = \begin{bmatrix} \cos(\alpha_T) & -\sin(\alpha_T) \\ \sin(\alpha_T) & \cos(\alpha_T) \end{bmatrix},$$

where $\bar{r} = [r_x \ r_y]^\top \in \mathbb{R}^2$ is the relative position of the robot with respect to the origin of the frame $\mathcal{O}_T$, $R(\alpha_T)$ is the rotational matrix characterizing the orientation of $\mathcal{O}_T$ with respect to $\mathcal{O}_N$.

A curvilinear desired path $\mathcal{P} \subset \mathbb{R}^2$ is described by the implicit equation

$$\mathcal{P} = \{\bar{r} : \varphi(\bar{r}) = 0\}, \tag{3}$$

where the function $\varphi : \mathbb{R}^2 \to \mathbb{R}$ is C^2-smooth. The same geometric curve $\mathcal{P}$ may be represented in the implicit form (3) in many ways. The principal restrictions imposed by our approach is *regularity*: in some vicinity of $\mathcal{P}$ one has

$$\|\nabla\varphi(\bar{r})\| = \left\| \left[\frac{\partial\varphi(\bar{r})}{\partial r_x}; \frac{\partial\varphi(\bar{r})}{\partial r_y} \right]^\top \right\| \neq 0, \tag{4}$$

where $\|\cdot\|$ is the Euclidean norm of the vector.

As illustrated in Fig. 1, the plane $\mathbb{R}^2$ is covered by the disjoint *level sets* of the function φ, that is, the sets where $\varphi(\bar{r}) = c = const$. The path is one of the level sets, corresponding to $c = 0$; the value $\varphi(\bar{r})$ can be considered as a (signed) "distance" from the robot to the path (differing, as usual, from the Euclidean distance). Generally, consider a strictly increasing C^1-function $\psi : \mathbb{R} \to \mathbb{R}$ with $\psi(0) = 0$ (and thus $\psi(z)z > 0$ for any $z \neq 0$). One may treat

$$e = \psi[\varphi(\bar{r})] \in \mathbb{R}, \tag{5}$$

as the *tracking error*. By definition, $e = 0$ if and only if $\bar{r} \in \mathcal{P}$. Our goal is to design a path following algorithm which eliminates the tracking error $|e(t)| \xrightarrow[t\to\infty]{} 0$, bringing thus the robot to the predefined path $\mathcal{P}$. The mapping $\psi(\cdot)$ in (5) 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 $\varphi(\bar{r})$ and the tracking error, depending on the choice of $\psi(\cdot)$. The path representation is usually chosen to be as simple as possible: for instance, dealing with a straight line, it is natural to choose $\varphi(\bar{r})$ linear, while the circular path is naturally described by $\varphi(\bar{r}) = \bar{r}^\top \bar{r}$. At the same time, some mathematical properties of the algorithm and, in particular, the region where convergence of the algorithm is guaranteed, depend on the way the tracking error is calculated.

Below we consider auxiliary relations which will be used in the further consideration. The derivation of a rotational matrix can be found in the following form

$$\dot{R}(\alpha) = -\dot{\alpha} E R(\alpha), \quad E = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}. \tag{6}$$

Here E is a skew-symmetric matrix with properties

$$EE = -I_2, \quad R^\top E R = E,$$

where I_2 is the 2×2 identity matrix. Similarly, the derivation of the unit orientation vector can be found in the following form

$$\dot{\bar{m}} = -\dot{\alpha} E \bar{m}. \tag{7}$$

Additionally, we have to define the first and second derivatives of the relative position vector $\bar{r}$:

$$\begin{cases} \dot{\bar{r}} = R^\top(\alpha_T)\left(\dot{\bar{p}} - \bar{v}_T\right), \\ \ddot{\bar{r}} = -usER^\top(\alpha_T)\bar{m}. \end{cases} \tag{8}$$

In addition, for the sake of simplicity in the analysis, we consider the following realistic assumption.

Assumption 1. The constant speed s is greater than the sum of the Euclidean norms of the disturbance vector $\bar{w}$ and the velocity vector $\bar{v}_T$, i.e. $s > \|\bar{w}\| + \|\bar{v}_T\|$.

In fact, it is quite straightforward to realize that this assumption is necessary if one wants to reach a desired path from almost every initial position.

3. MAIN RESULT

The path following control presented in this work is based on two steps. The first one is about constructing a guidance vector field such that once the robot is tracking it, the vehicle will converge to $\mathcal{P}$. The second step deals with the task of steering the robot in order to converge to such a guiding vector field.

3.1 Design of the vector field

Let us first introduce some notation. We define $n(\bar{r}) := \nabla\varphi(\bar{r})$ as the normal vector to the curve corresponding to the level set $\varphi(\bar{r})$ at the point $\bar{r}$ and the tangent vector τ at the same point $\bar{r}$ is given by the rotation

$$\tau(\bar{r}) = En(\bar{r}).$$

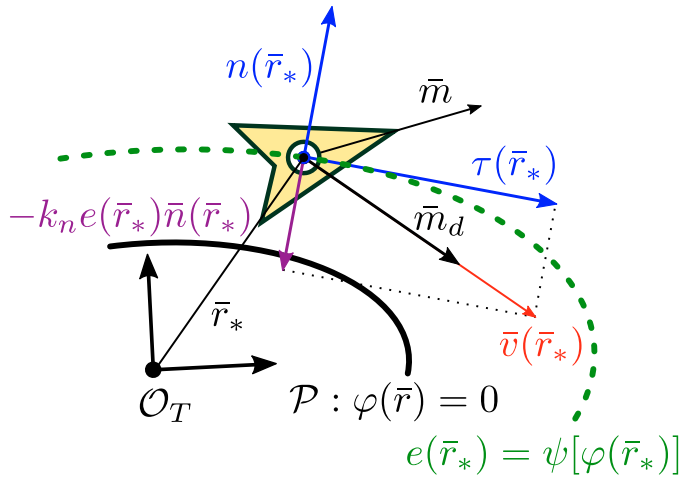

Fig. 2. The construction of the *guiding vector field*

Note that E will determine in which direction $\mathcal{P}$ will be tracked. The basic idea of the path following algorithm is to steer to a direction, in which the tracking error decreases.

We introduce the function

$$V(\bar{r}) = \frac{1}{2} e^2(\bar{r}). \tag{9}$$

Its derivative along the solution of (8) is

$$\dot{V} = e(\bar{r})\psi'\left(\varphi(\bar{r})\right)\nabla\varphi(\bar{r})\dot{\bar{r}} = \psi'\left(\psi^{-1}(e)\right) e\bar{n}^\top \dot{\bar{r}}. \tag{10}$$

If $e \neq 0$, the function V (and hence $|e|$) is decreasing, provided that $e\bar{n}^\top\dot{\bar{r}} < 0$. On the other hand, as $e \approx 0$, one has to guide the robot to steer to the path $\mathcal{P}$ so that $\dot{\bar{r}} \approx \bar{\tau}/\|\bar{\tau}\|$. This motivates to design the heading controller in a way to provide the desired orientation $\dot{\bar{r}} \uparrow\uparrow \bar{v}$, where the vector field $\bar{v}$ is given by

$$\bar{v}(\bar{r}) = \bar{\tau}(\bar{r}) - k_n e(\bar{r})\bar{n}(\bar{r}), \quad k_n = const > 0. \tag{11}$$

In other words, the desired orientation is $\bar{m} = \bar{m}_d$, where

$$\bar{m}_d = \frac{\bar{v}}{\|\bar{v}\|}, \quad \|\bar{v}(\bar{r})\| \neq 0. \tag{12}$$

We call $\bar{m}_d(\bar{r})$ the *guiding vector field*. Fig. (2) illustrates the relations between the vectors $\bar{r}$, $\bar{m}$, $\bar{\tau}$, $\bar{n}$, $\bar{v}$ and $\bar{m}_d$.

The controller offered in the next subsection employs the field vector $\bar{m}_d(\bar{r})$ at any point as well as its derivative along the trajectory $\dot{\bar{m}}_d$. Differentiating (11), one arrives at

$$\dot{\bar{v}} = \left(E - k_n e(\bar{r}) I_2\right) H(\bar{r})\dot{\bar{r}} - k_n \dot{e}(\bar{r})\bar{n}, \tag{13}$$

$$\dot{e}(\bar{r}) = \psi'\left(\varphi(\bar{r})\right)\bar{n}^\top\dot{\bar{r}}, \tag{14}$$

$$\dot{\bar{m}}_d = \frac{d}{dt}\frac{\bar{v}}{\|\bar{v}\|} = -\frac{1}{\|\bar{v}\|} E\bar{m}_d\bar{m}_d^\top E\dot{\bar{v}}. \tag{15}$$

Here $H(\bar{r})$ stands for the Hessian

$$H(\bar{r}) = \begin{bmatrix} \dfrac{\partial^2}{\partial r_x^2}\varphi(\bar{r}) & \dfrac{\partial^2}{\partial r_x \partial r_y}\varphi(\bar{r}) \\ \dfrac{\partial^2}{\partial r_y \partial r_x}\varphi(\bar{r}) & \dfrac{\partial^2}{\partial r_y^2}\varphi(\bar{r}) \end{bmatrix}.$$

Since $\|\bar{m}_d\|^2 = 1$, one has $\bar{m}_d^\top \dot{\bar{m}}_d = 0$, and hence

$$\boxed{\dot{\bar{m}}_d = -\omega_d E\bar{m}_d \Longleftrightarrow \omega_d = -\dot{\bar{m}}_d^\top E\bar{m}_d,} \tag{16}$$

where ω_d is a scalar function that can be explicitly found from (13)-(15).

3.2 Path following controller design

Now we are going to present how to make the robot to converge to the guidance vector field defined in (12).

The relative velocity $\dot{\bar{r}}$ can be trivially decomposed as

$$\dot{\bar{r}} = \|\dot{\bar{r}}\|\bar{m}_r, \tag{17}$$

where $\|\dot{\bar{r}}\|$ is an instant relative speed and $\bar{m}_r$ is the unit orientation vector. To find the relation between the relative velocity and the control input u, one has to find the derivation of the vector $\bar{m}_r$ using (8)

$$\dot{\bar{m}}_r = -\frac{1}{\|\dot{\bar{r}}\|} E\bar{m}_r\bar{m}_r^\top E\ddot{\bar{r}} = -u\frac{s}{\|\dot{\bar{r}}\|}\bar{m}_r^\top R^\top(\alpha_T)\bar{m}E\bar{m}_r.$$

Now, it is possible to introduce the new control variable u^* in the following way

$$u^* = u\frac{s}{\|\dot{\bar{r}}\|}\bar{m}_r^\top R^\top(\alpha_T)\bar{m}. \tag{18}$$

Thus, one can rewrite the equations above in the compact form

$$\boxed{\dot{\bar{m}}_r = -u^* E\bar{m}_r.} \tag{19}$$

Comparing with (7) it is clear that the new control input u^* characterizes the angular speed of the vector $\bar{m}_r$. Although Assumption 1 is quite conservative, it guarantees that the changing of coordinates (18) is well-defined.

As we have just discussed, the idea of the path following algorithm is to steer the robot along the guiding vector field $\bar{m}_d$. We introduce the directed angle $\delta \in (-\pi;\pi]$ between $\bar{m}_d$ and $\bar{m}_r$. The function δ is being C^1-smooth on the set of points where $\bar{m}_r \neq -\bar{m}_d$.

Obviously, the vector $\bar{m}_r$ can be decomposed as follows

$$\bar{m}_r = (\cos\delta)\bar{m}_d - (\sin\delta)E\bar{m}_d. \tag{20}$$

As was noticed, $\dot{\bar{m}}_r = -u^*E\bar{m}$, whereas $\dot{\bar{m}}_d = -\omega_d E\bar{m}_d$. By noticing that $\sin\delta = -\bar{m}_r^\top E\bar{m}_d$ and $\cos\delta = \bar{m}_r^\top\bar{m}_d$, at any time $t \geq 0$ when $\bar{m}_r \neq -\bar{m}_d$ (that is, $\delta < \pi$) one has

$$-u^*E\bar{m}_r = \dot{\bar{m}}_r = -\left(\dot\delta + \omega_d\right)E\bar{m}_r \Longleftrightarrow \dot\delta = u^* - \omega_d.$$

More generally, at *any* time the following equality is valid

$$\frac{d}{dt}\sin\delta = -\frac{d}{dt}\bar{m}_r^\top E\bar{m}_d = (u^* - \omega_d)\cos\delta. \tag{21}$$

We are now in a position to describe our path-following algorithm. Calculating ω_d at any point, the control input is

$$\boxed{u^* = \omega_d - k_\delta\delta.} \tag{22}$$

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

When $\delta < \pi$, equality (21) holds and thus

$$\boxed{\dot\delta = u^* - \omega_d = -k_\delta\delta.} \tag{23}$$

Furthermore, even for $\delta(0) = \pi$, one has $u^* - \omega_d = -k_\delta\delta < 0$ as $t \approx t_0$ and hence (23) 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 (22) is equivalent to a very simple proportional controller (23).

Collecting together formulas (16), (18) and (22) we obtain the final form of the control law:

$$\boxed{u = \frac{\|\dot{\bar{r}}\|}{s}\frac{\left(-\dot{\bar{m}}_d^\top E\bar{m}_d - k_\delta\delta\right)}{\bar{m}_r^\top R^\top(\alpha_T)\bar{m}}.} \tag{24}$$

3.3 Properties of the solutions

In this subsection we discuss our main results, concerned with the properties of the resulting solutions. In comparison with the original solution of the path following problem for the driftless nonholonomic model (Kapitanyuk et al., 2016), the additional components such as the constant disturbance vector $\bar{w}$ and the kinematic of the moving frame $\mathcal{O}_T$ have been added. Fortunately, using the changing of coordinates (8) and (18) and under Assumption 1, one can transform a moving path following problem to the standard form considered in the paper (Kapitanyuk et al., 2016). Moreover, the theorems about the existence of solutions are valid for the problem considered in this paper.

4. EXPERIMENTAL VALIDATION

In this section we are going to present two cases. The first result demonstrates the performance of the guidance algorithm in case of the fixed external frame. The second experiment demonstrates the numerical simulation of the moving path following problem.

4.1 The sinusoidal path in the fixed external frame

We have used the fixed wing UAV(in Fig. 3) equipped with the Paparazzi autopilot(Paparazzi, 2003) as the experimental platform. It is about 0.850 kg of mass and 1.2 m of wingspan. The nominal airspeed is $s = 11.5$ m/s. The chosen board for the autopilot includes the usual sensors of three axis gyros, accelerometers, magnetometers and a GPS. Therefore we can measure $\bar{p}$, $\dot{\bar{p}}$ and α. The microcontroller on board is an STMicroelectronics STM32F4. Although there is a logging system on board, the vehicle counts with a serial radio link in order to monitor its status from the ground. The *guiding vector field* algorithm has been programmed as a (guidance) module in Paparazzi and can be combined or integrated easily with other modules in the system. In particular, we have set the periodic frequency of the *guidance vector field module* to 60 Hz. The wind speed was about 4 m/s. The external frame $\mathcal{O}_T$ was fixed, i.e. $\bar{v}_T = [0;0]^\top$. More details about practical implementation of the proposed control algorithm can be found in the paper (Garcia de Marina et al., 2016b).

As a benchmark we chose a sinusoidal path $\mathcal{P}$, namely

$$\varphi(\bar{r}) = \bar{r}_x - A\sin(\omega\bar{r}_y),\ \bar{r} = R^\top(\alpha_T)(\bar{p} - \bar{p}_T), \tag{25}$$

where $\bar{r} = [\bar{r}_x;\ \bar{r}_y]^\top$, $A = 50$ m, $\omega = 0.0251$ rad/m, the position of the fixed external frame $\mathcal{O}_T$ has been chosen as $\bar{p}_T = [37.53;\ 73]^\top$ m and $\alpha_T = -\pi/2$ rad is the

orientation of the frame $\mathcal{O}_T$ with respect to $\mathcal{O}_N$. We choose $k_n = 0.006$, $k_\delta = 0.8$ and define $e(\bar{r}) = \psi(\varphi(\bar{r})) = \varphi(\bar{r})$.

Fig. 4 illustrates both the resulting UAV's trajectory together and the *guiding vector field* (12) for the given path (25). The corresponding tracking error is displayed in Fig. 5.

4.2 The Cassini oval in the moving frame

In this case, we consider the standard unicycle model (1) with the constant speed $s = 1$ and the disturbance vector $\bar{w} = [0.5; 0]^\top$. We define the function $\varphi(\bar{r})$ generating our desired path $\{\mathcal{P} : \varphi(\bar{r}) = 0\}$ as follows

$$\varphi(\bar{r}) = (\bar{r}_x^2 + \bar{r}_y^2)^2 - 2c^2(\bar{r}_x^2 - \bar{r}_y^2) - a^4 + c^4, \qquad (26)$$

where $a = 1.1$, $c = 1.09$ are parameters of the curve. The initial position of the frame $\mathcal{O}_T$ is $\bar{p}_T(0) = [0; -0.5]^\top$ m and corresponding orientation $\alpha_T = 0$ rad; the translational velocity has been chosen as $\bar{v}_T = [0; -0.2]^\top$.

We choose $k_n = 1$, $k_\delta = 10$ and define the tracking error in the following form $e(\bar{r}) = \psi(\varphi(\bar{r})) = \varphi(\bar{r})$. The *guiding vector field* (12) for this case is depicted in Fig. 6.

Fig. 7 illustrates four trajectories (labeled a, b, c and d), corresponding to the initial conditions

$$\begin{aligned} a &: (x = -0.1, y = -0.1, \alpha = -\pi), \\ b &: (x = 1.0, y = 0.0, \alpha = \pi/6), \\ c &: (x = -2.0, y = 0.0, \alpha = -\pi/3), \\ d &: (x = -0.5, y = -0.5, \alpha = \pi/6). \end{aligned} \qquad (27)$$

The corresponding tracking errors are displayed in Fig. 8.

5. CONCLUSIONS

In this paper we offer an extension of the guiding vector field algorithm for a moving path-following control of nonholonomic robots in presence of disturbances. We considered the problem when a desired path is given with respect to the external coordinate frame that is uniformly moving. The guidance strategy is based on following a reference vector field generated from the implicit representation of the desired path. The results have been experimentally validated, using the fixed wing UAV and numerical simulation.

There are several options on how to extend the presented results. In this work we assumed the existence of a suitable state estimator that provides proper data; however, in practice, it may be impossible. Thus, the aggregation of the proposed guidance controller with observers is an important part of the future considerations. Another direction for the further development is to extend the algorithm to the three-dimensional case. Lastly, we are currently extending and testing the results of this paper for formation control by employing the different level sets of a desired trajectory as an input for consensus algorithms and combining the recent findings in (Garcia de Marina et al., 2016a) for controlling rigid formations.

REFERENCES

Akhtar, A., Nielsen, C., and Waslander, S. (2015). Path following using dynamic transverse feedback linearization for car-like robots. *IEEE Trans. Robotics*, 31(2), 269–279.

Fig. 3. The fixed wing UAV used for the approbation

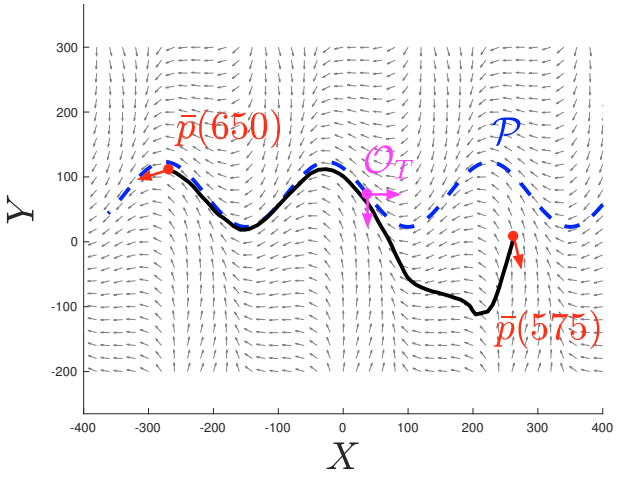

Fig. 4. The resulting UAV's trajectory and the *guiding vector field*

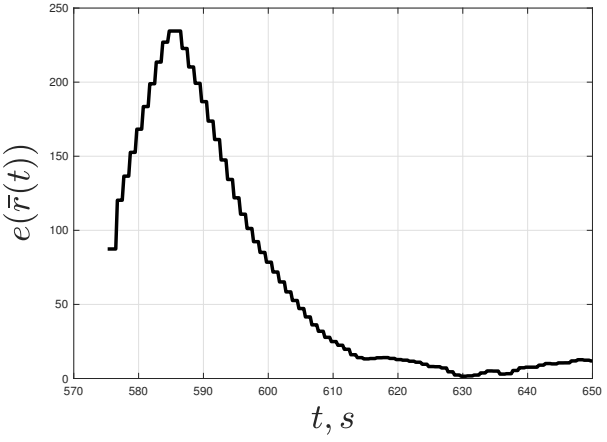

Fig. 5. The dynamics of the tracking error $e(\bar{r}(t))$

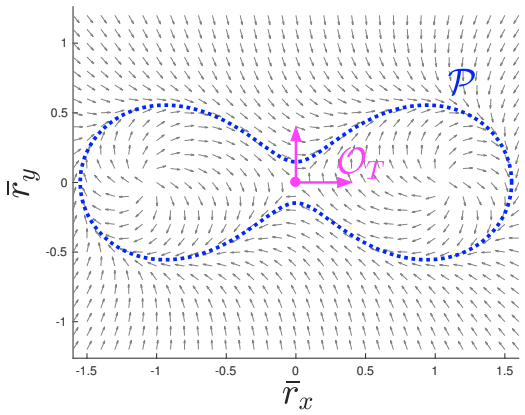

Fig. 6. The *guiding vector field* for the Cassini oval (26)

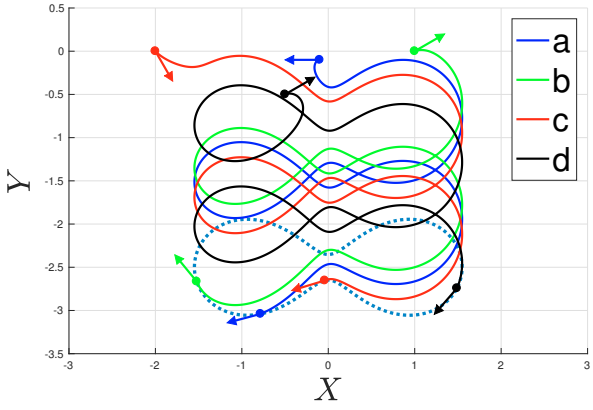

Fig. 7. The resulting robot's trajectories corresponding to the different initial conditions (27)

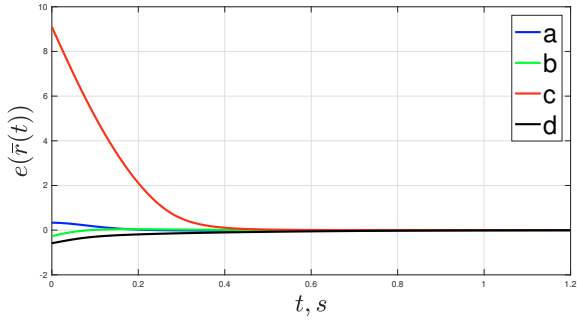

Fig. 8. The dynamics of the tracking error $e(\bar{r}(t))$

Caharija, W., Pettersen, K.Y., Calado, P., and Braga, J. (2015). A comparison between the ILOS guidance and the vector field guidance. *IFAC-PapersOnLine*, 48(16), 89 – 94.

Fossen, T.I. and Pettersen, K.Y. (2014). On uniform semiglobal exponential stability (USGES) of proportional line-of-sight guidance laws. *Automatica*, 50(11), 2912 – 2917.

Garcia de Marina, H., Jayawardhana, B., and Cao, M. (2016a). Distributed rotational and translational maneuvering of rigid formations and their applications. *IEEE Transactions on Robotics*, 32(3), 684–697.

Garcia de Marina, H., Kapitanyuk, Y.A., Bronz, M., Hattenberger, G., and Cao, M. (2016b). Guidance algorithm for smooth trajectory tracking of a fixed wing UAV flying in wind flows. *arXiv: 1610.02797*.

Hoy, M., Matveev, A.S., and Savkin, A.V. (2015). Algorithms for collision-free navigation of mobile robots in complex cluttered environments: a survey. *Robotica*, 33(03), 463–497.

Kapitanyuk, Y.A., Proskurnikov, A.V., and Cao, M. (2016). A guiding vector field algorithm for path following control of nonholonomic mobile robots (under review in IEEE Trans. Control Techn.). *arXiv: 1610.04391*.

Lawrence, D.A., Frew, E.W., and Pisano, W.J. (2008). Lyapunov vector fields for autonomous unmanned aircraft flight control. *Journal of Guidance, Control, and Dynamics*, 31(5), 1220–1229.

Matveev, A.S., Hoy, M.C., and Savkin, A.V. (2016). Extremum seeking navigation without derivative estimation of a mobile robot in a dynamic environmental field. *IEEE Transactions on Control Systems Technology*, 24(3), 1084–1091.

Nelson, D., Barber, D., McLain, T., and Beard, R. (2007). Vector field path following for miniature air vehicles. *IEEE Trans. Robotics*, 23(3), 519–529.

Oh, H., Kim, S., Shin, H.S., Tsourdos, A., and White, B. (2013). Coordinated standoff tracking of groups of moving targets using multiple UAVs. In *Control Automation (MED), 2013 21st Mediterranean Conference on*, 969–977.

Oliveira, T., Aguiar, A.P., and Encarnao, P. (2016). Moving path following for unmanned aerial vehicles with applications to single and multiple target tracking problems. *IEEE Transactions on Robotics*, 32(5), 1062–1078.

Paley, D.A., Zhang, F., and Leonard, N.E. (2008). Cooperative control for ocean sampling: The glider coordinated control system. *IEEE Transactions on Control Systems Technology*, 16(4), 735–744.

Pamosoaji, A.K. and Hong, K.S. (2013). A path-planning algorithm using vector potential functions in triangular regions. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 43(4), 832–842.

Paparazzi (2003). Uav open-source project. URL `http://wiki.paparazziuav.org/`.

Renzaglia, A., Reymann, C., and Lacroix, S. (2016). Monitoring the evolution of clouds with UAVs. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 278–283.

Samson, C. (1992). Path following and time-varying feedback stabilization of a wheeled mobile robot. In *Proc. of the ICARCV'92, Singapore*.

Siciliano, B. and Khatib, O. (eds.) (2008). *Springer Handbook of Robotics*. Springer.

Soetanto, D., Lapierre, L., and Pascoal, A. (2003). Adaptive, non-singular path-following control of dynamic wheeled robots. In *Proc. of 42nd IEEE Conf. on Decision and Control*, volume 2, 1765–1770.

Sujit, P., Saripalli, S., and Borges Sousa, J. (2014). Unmanned aerial vehicle path following: A survey and analysis of algorithms for fixed-wing unmanned aerial vehicless. *IEEE Control Systems*, 34(1), 42–59.