Received: 11 April 2022 | Revised: 6 August 2022 | Accepted: 8 November 2022

DOI: 10.1002/rob.22142

SURVEY ARTICLE

WILEY

A review of path following control strategies for autonomous robotic vehicles: Theory, simulations, and experiments

Nguyen Hung | Francisco Rego | Joao Quintas | Joao Cruz | Marcelo Jacinto | David Souto | Andre Potes | Luis Sebastiao | Antonio Pascoal

Department of Electrical and Computer Engineering, Institute for Systems and Robotics (ISR), Instituto Superior Técnico (IST), University of Lisbon, Lisbon, Portugal

Correspondence
Nguyen Hung, Department of Electrical and Computer Engineering, Institute for Systems and Robotics (ISR), Instituto Superior Técnico (IST), University of Lisbon, Lisbon, Portugal.
Email: nguyen.hung@tecnico.ulisboa.pt

Funding information
FETPROACT; Federación Española de Enfermedades Raras; Horizon 2020 Framework Programme; Fundação para a Ciência e a Tecnologia

Abstract

This article presents an in-depth review of path following control strategies that are applicable to a wide range class of marine, ground, and aerial autonomous robotic vehicles. From a control system standpoint, path following can be formulated as the problem of stabilizing a path following error system that describes the dynamics of position and possibly orientation errors of a vehicle with respect to a path, with the errors defined in an appropriate reference frame. In spite of the large variety of path following methods described in the literature we show that, in principle, most of them can be categorized in two groups: stabilization of the path following error system expressed either in the vehicle's body frame or in a frame attached to a "reference point" moving along the path, such as a Frenet-Serret (F-S) frame or a Parallel Transport (P-T) frame. With this observation, we provide a unified formulation that is simple but general enough to cover many methods available in the literature. We then discuss the advantages and disadvantages of each method, comparing them from the design and implementation standpoint. We further show experimental results of the path following methods obtained from field trials testing with under-actuated and over-actuated autonomous marine vehicles. In addition, we introduce open-source Matlab and Gazebo/ROS simulation toolboxes that are helpful in testing path following methods before their integration in the combined guidance, navigation, and control systems of autonomous vehicles.

KEYWORDS

autonomous cars, autonomous marine vehicles (AMVs), autonomous surface vehicles (ASVs), control, guidance, over-actuated vehicles, path following, under-actuated vehicles, underwater autonomous vehicles (UAVs), unmanned aerial vehicles (UAVs), unmanned ground vehicles (UGVs)

1 | INTRODUCTION

Path-following (PF) is one of the most fundamental tasks to be executed by autonomous vehicles. It consists of driving a vehicle to and maintaining it on a pre-defined path while tracking a path-dependent speed profile. Unlike trajectory tracking, the path is not parameterized by time but rather by any other useful parameter that in some cases may be the path length. Thus, there is more flexibility in making the vehicle first converge to the path smoothly then move along it while tracking a given speed assignment. Path following is useful in many applications where the main objective is to accurately traverse the path, while maintaining a certain speed is a secondary task. Stated in simple terms, it is not required for the vehicle to be at specific positions at specific instants of time, a strong requirement in

trajectory tracking. All that is required is for the vehicle to go through specific points in space while trying to meet speed assignments, but absolute time is not of overriding importance. From a technical standpoint, when compared with trajectory tracking, path following has the potential to exhibit smoother convergence properties and reduced actuator activity (Aguiar & Hespanha, 2007). For these reasons, in a vast number of applications path-following is performed by a variety of heterogeneous vehicles, of which marine vehicles (Fossen, 2011; Nad et al., 2020; Rego et al., 2019), ground vehicles such as Mars rovers (Fox et al., 1997; Helmick et al., 2004; Kanjanawanishkul & Zell, 2009), fixed wing aerial vehicles (Liu et al., 2013; Nelson et al., 2007; Park et al., 2007; Sujit et al., 2014; Yang et al., 2021), autonomous cars (De Luca et al., 1998; Guo et al., 2019; Rokonuzzaman et al., 2021; Snider et al., 2009), and quad-rotors (Rubí et al., 2019) are representative examples.

The task of deriving control strategies to solve the PF problem is technically challenging, specially in the presence of nonholonomic constraints, and often involves the use of a nonlinear control techniques such as backstepping (Encarnacao & Pascoal, 2000; Lapierre et al., 2006), feedback linearization (Akhtar et al., 2012), sliding mode control (Dagci et al., 2003), vector field (Nelson et al., 2007; Wilhelm & Clem, 2019), linear model predictive control (MPC) (Kanjanawanishkul & Zell, 2009; Raffo et al., 2009), and nonlinear MPC (NMPC) (Alessandretti et al., 2013; Guo et al., 2019; Hung et al., 2020; Shibata et al., 2018; Yang et al., 2021). Other path following algorithms exploit learning-based methods such as Learning-based MPC (LB-MPC) (Ostafew et al., 2016; Rokonuzzaman et al., 2020), reinforcement learning-based control (Kamran et al., 2019; Martinsen & Lekkas, 2018; Wang et al., 2021), or geometrical based-approaches like Pure Pursuit (Amidi & Thorpe, 1991; Coulter, 1992), among others. As an overview picture of the work done on this topic, Figure 1 shows the evolution of path following research over the last three decades, highlighting key methods described in the literature.

Due to the proliferation of robotic vehicles and their applications, the past three decades have witnessed the development of a multitude of path-following methods. Therefore, to provide a critical assessment of the collective work done, this paper contains a comprehensive survey, aimed at comparing different PF methods from the design and implementation standpoint and discussing the advantages and disadvantages of each method. In particular, we show that an important classification of path-following algorithms is given by the choice of reference frame in which the path-following error is defined. In general, the latter is defined either in the vehicle's body frame or in a frame attached to a "reference point" moving along the path such as the Frenet–Serret (F-S) frame or the Parallel Transport (P-T) frame.

In the literature, one can find other survey papers of path following methods such as Sujit et al. (2014) for fixed-wing aerial vehicles, Rubí et al. (2019) for quadrotors, or Rokonuzzaman et al. (2021) for autonomous car-like robot. However, the aforementioned survey papers do not describe in detail the theory that supports the methods described and, while they contain simulation results, they do not present a comparison of the performance of the methods in field tests with real vehicles. Moreover, the above surveys focus on specific types of autonomous vehicles, and do not consider some of

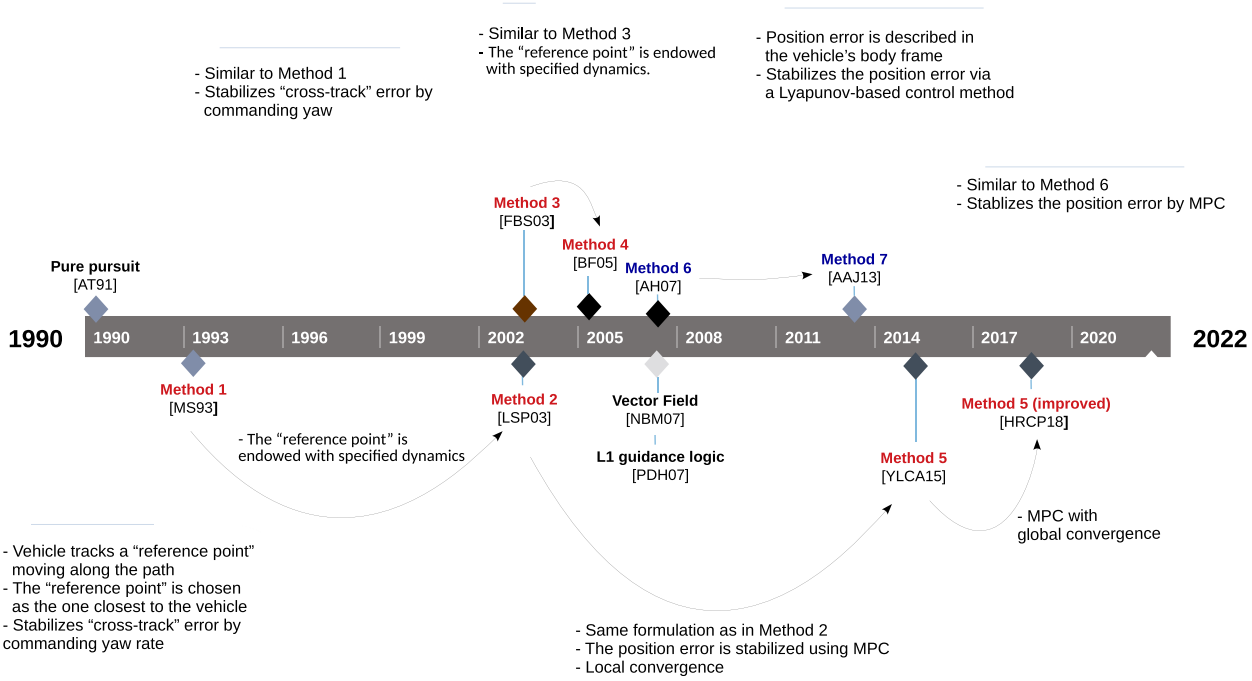

FIGURE 1 A timeline showing the evolution of key path followings methods described the literature. Articles are referenced as in the bibliography. The work referenced to Methods 1–7 is reviewed in great detail in the present paper. [Color figure can be viewed at wileyonlinelibrary.com]

the unique characteristics of under-actuated marine vehicles such as nonholonomic constraints in Rubí et al. (2019) or the requirement that a vehicle track a desired speed profile along the path (Rokonuzzaman et al., 2021; Sujit et al., 2014). In the current review paper, instead of focusing on a particular type of vehicle, we emphasize the common principle underlying path following methods that can be applied and extended to a large class of vehicles, the simplified motion of which can be described by the same class of kinematic models. Furthermore, the present paper also provides a rigorous theoretical proof of the methods reviewed that is absent in the previous surveys. We introduce simulation toolboxes written in Matlab and ROS/Gazebo that are helpful in testing the path following methods and integrating them in guidance, navigation, and control systems. To conclude, we report experimental results with autonomous marine vehicles of the Medusa class (Abreu et al., 2016) that have been widely used in EU projects such as WiMust (Abreu et al., 2016) and MORPH (Caldeira Abreu et al., 2015). In short, the main contributions of this paper include

1. An in-depth review of standard path-following methods in two-dimensional space (2D) explaining in detail the theoretical principles of the different methods.
2. A discussion of the advantages and disadvantages of each method, comparing them from a design and implementation standpoint.
3. A Matlab simulation toolbox and ROS/Gazebo simulation packages for the study of path-following methods.
4. A description of experimental results of field trials at sea with the Medusa under-actuated and over-actuated robotic vehicles, followed by an assessment of the performance obtained in real-life situations.The paper is organized as follows. The path following problem is formulated in Section 2. Section 3 describes path following methods for under-actuated vehicles. Section 4 extends the path following methods to the case where external unknown disturbances occur. Section 5 reviews a path following method developed for fully-actuated vehicles with arbitrary heading. The implementation in Matlab and Gazebo/ROS simulation toolboxes for testing path following methods are introduced in Section 6. Experimental results with autonomous marine vehicles are presented in Section 7. Section 8 provides a discussion on advantages and disadvantages of the path following methods presented and addresses practical issues that arise when the vehicles dynamics are taken into account. Finally, Section 9 contains the main conclusions.

2 | PROBLEM FORMULATION AND COMMON PRINCIPLES OF PATH FOLLOWING METHODS

2.1 | Vehicle kinematic models

Path following refers to the problem of making a vehicle converge to and follow a spatial path while asymptotically tracking a desired speed profile along the path; see Figure 2 that shows an ASV executing a path following maneuver. From a control system standpoint, the structure of a complete path following system is captured in Figure 3a. In this architecture, the outer-loop path following controller implements a guidance strategy, in charge of computing desired references (e.g., linear and angular speeds, or orientations) to steer the vehicle along the path with a desired speed profile U_d. These references act as inputs to autopilots that play the role of inner-loop controllers, in charge of generating suitable forces and torque for the vehicle to track the desired references, thus achieving the path following objectives.

In practice, to simplify the design of a path following controller, it is commonly assumed that the responses the inner-loops are sufficiently fast so that the influence of the latter in the complete system can be neglected. Under these ideal conditions, the path following control system can be simplified by considering the kinematics model only, as shown in Figure 3b. Our purpose in the present paper is to review the core ideas behind existing path following methods in the literature, therefore we primarily focus on those designed for the vehicle kinematics only. We then treat the effect of the inner-loops as an internal disturbances and analyze the robustness of the path following methods under these disturbances accordingly.

The vehicle kinematic models that will be used to derive path following control laws in the present paper will be described next. The vehicle's motion with respect to a path is illustrated (Figure 4). The following notation and nomenclature will be used. The symbol $\{\mathcal{I}\} = \{x_I, y_I\}$ denotes an inertial (global) North-East (NE) frame, where the axis x_I points to the North and the axis y_I points to the East. Let Q be the center of mass of the vehicle and denote by $\mathbf{p} = [x, y]^\top \in \mathbb{R}^2$ the position of Q in $\{\mathcal{I}\}$. Let also $\{\mathcal{B}\} = \{x_\mathcal{B}, y_\mathcal{B}\}$ be a body-fixed frame whose origin is located at Q. In addition, denote by $\mathbf{v} = [u, v]^\top \in \mathbb{R}^2$ the vehicle's velocity vector with respect to the fluid, measured in $\{\mathcal{B}\}$, where u, v are the surge/longitudinal and sway/lateral speeds, respectively. With the above notation, the *"general"* 3-DOF vehicle's kinematic model is described by

$$\begin{aligned} \dot{x} &= u\cos(\psi) - v\sin(\psi) + v_{cx} \\ \dot{y} &= u\sin(\psi) + v\cos(\psi) + v_{cy} \\ \dot{\psi} &= r, \end{aligned} \tag{1}$$

where ψ is the vehicle's heading/yaw angle and r is its heading/yaw rate, v_{cx} and v_{cy} are components of vector $\mathbf{v}_c = [v_{cx}, v_{cy}]^\top \in \mathbb{R}^2$ that

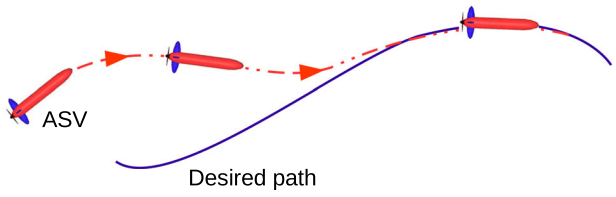

FIGURE 2 Geometric illustration of the path following problem [Color figure can be viewed at wileyonlinelibrary.com]

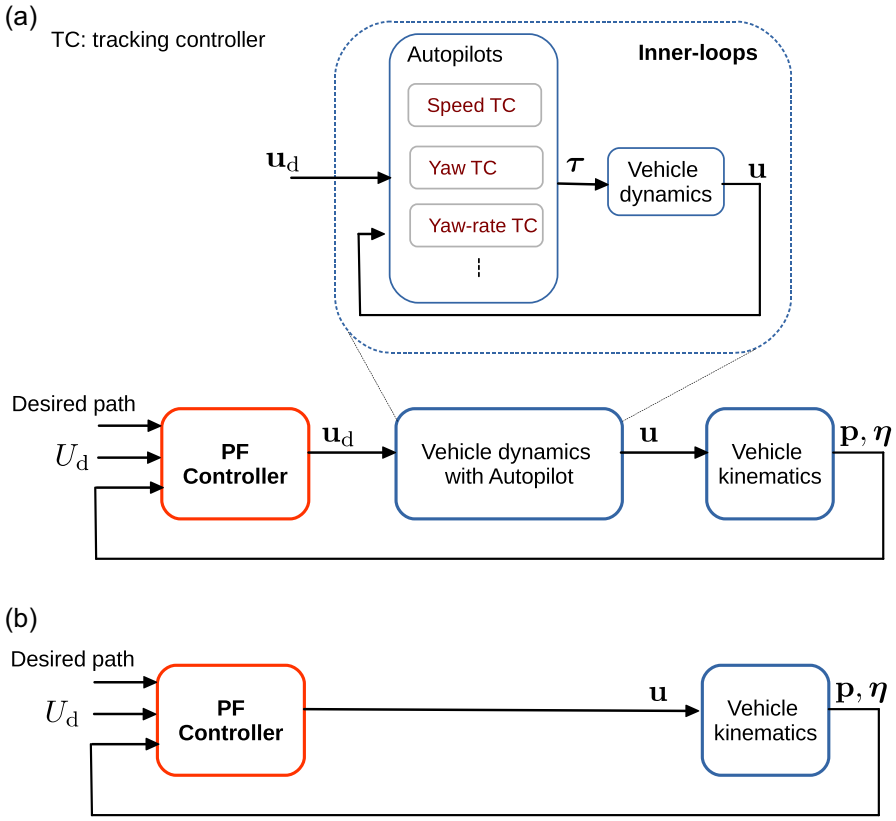

FIGURE 3 Path following control systems; $\mathbf{u}_d$: reference inputs (e.g., desired linear and angular speeds, orientations) for the autopilots; $\mathbf{p}$ and $\boldsymbol{\eta}$: the vehicle's position and orientation, respectively, $\boldsymbol{\tau}$: force and torque, U_d: desired speed profile that the vehicle must track. (a) A complete path following control system and (b) a simplified path following system for PF controller design. PF, path-following. [Color figure can be viewed at wileyonlinelibrary.com]

represents the effect of external unknown disturbances (e.g., ocean current in the case of marine vehicles and wind in the case of aerial vehicles) in $\{\mathcal{I}\}$. For the sake of clarity, in the present paper we consider the following three separate subcases of (1).

2.1.1 | Scenario 1: Under-actuated vehicle without external disturbances

In this scenario we assume that the vehicle is under-actuated and that the vehicle's lateral motion and external disturbances are so small so that they can be neglected, that is, making v, v_{cx}, v_{cy} zero we obtain

$$\begin{aligned} \dot{x} &= u\cos(\psi) \\ \dot{y} &= u\sin(\psi) \\ \dot{\psi} &= r. \end{aligned} \tag{2}$$

We also assume that the longitudinal speed (u) and the heading (ψ) or heading rate (r) can be tracked with good accuracy by inner-loop controllers. Although (2) is a significant simplification of (1) it still captures sufficiently well the behavior of a large class of under-actuated vehicles, including unicycle mobile robots (Micaelli & Samson, 1993; Pyo et al., 2017), fixed-wing UAVs undergoing planar motion (Nelson et al., 2007; Rucco et al., 2015; Zhao et al., 2020), and a wide class of under-actuated autonomous marine vehicles (AMVs). The later include the

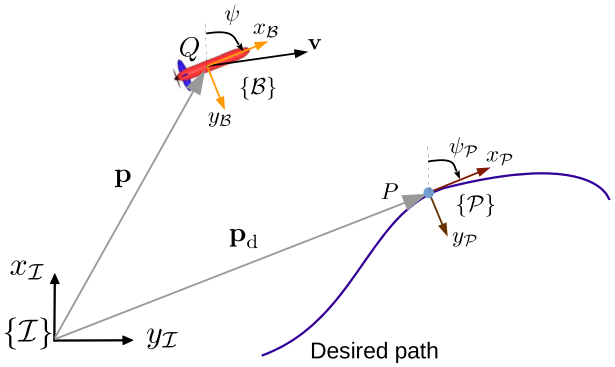

FIGURE 4 Geometric illustration of the path following problem. $\{I\}$, $\{\mathcal{B}\}$, and $\{\mathcal{P}\}$ denote inertial/global, the vehicle body, and path frames, respectively. The symbols $\mathbf{p}$ and $\mathbf{p}_d$ represent the position vectors of the vehicle and a generic point on the path, respectively expressed in $\{\mathcal{I}\}$ [Color figure can be viewed at wileyonlinelibrary.com]

Medusa and Delfim (Abreu et al., 2016) and Charlie (Bibuli et al., 2009) vehicles, for which the sway speed is in practice so small that it can be neglected. Snapshots of several vehicles mentioned above are shown in Figure 5. In the literature, most path following methods are proposed for the kinematic model (2); in accordance, a large part of this paper presented in Section 3 is devoted to their review.

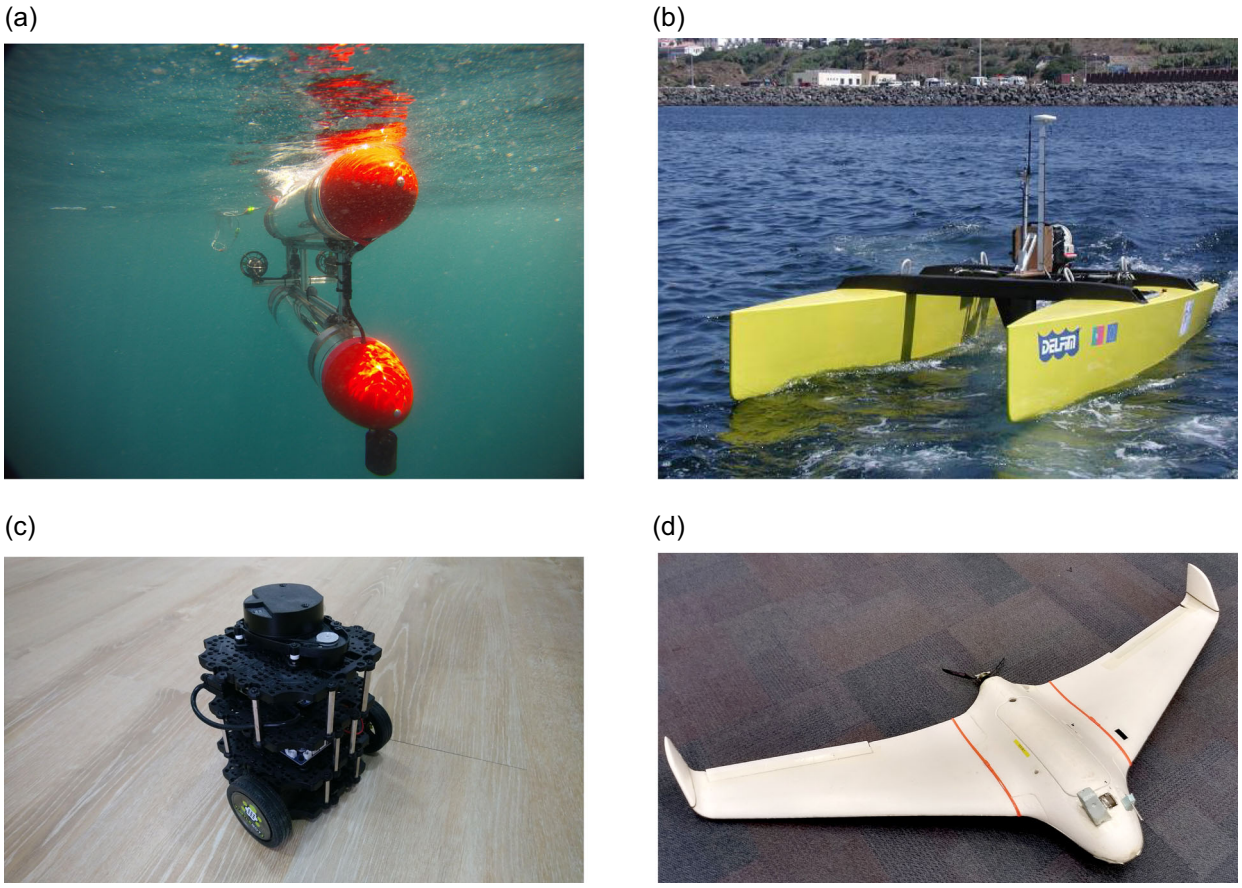

FIGURE 5 Under-actuated robotic vehicles. (a) Under-actuated Medusa (Abreu et al., 2016), (b) Delfim (Alves et al., 2006), (c) Turtlebot Burger (Pyo et al., 2017), and (d) X8 fixed-wing UAV (Yang et al., 2021). [Color figure can be viewed at wileyonlinelibrary.com]

2.1.2 | Scenario 2: Under-actuated vehicle with external disturbances

In this scenario the vehicle motion is influenced significantly by external disturbances that can not be neglected; however, the lateral sway is still small enough to be ignored (i.e., $v = 0$), see (Aguiar & Pascoal, 2002; Caharija et al., 2016; Liu et al., 2013; Yang et al., 2021). Given these assumptions, the vehicle kinematics model is given by

$$\begin{aligned} \dot{x} &= u\cos(\psi) + v_{cx} \\ \dot{y} &= u\sin(\psi) + v_{cy} \\ \dot{\psi} &= r. \end{aligned} \tag{3}$$

Note also that similar to *Scenario 1*, the vehicle is under-actuated. Path following methods developed for this model will be discussed in Section 4.

2.1.3 | Scenario 3: Fully or over-actuated vehicle

In this scenario we assume that the lateral speed is significant and can not be neglected. However, for simplicity we neglect the effect of external disturbances. With these assumptions, the vehicle kinematic model is given by

$$\begin{aligned} \dot{x} &= u\cos(\psi) - v\sin(\psi) \\ \dot{y} &= u\sin(\psi) + v\cos(\psi) \\ \dot{\psi} &= r. \end{aligned} \tag{4}$$

In this scenario we assume further that the vehicle is fully or over-actuated in that its longitudinal and lateral speeds and heading rate can be controlled simultaneously. Figure 6 shows snapshots of several fully-actuated vehicles that meet these assumptions. For this scenario we are interested the path following problem in which the vehicle is not only required to follow a predefined path but also to maneuver such that its heading tracks an arbitrary heading reference. This scenario will be presented in Section 5.

2.2 | Path parameterization and path frames

Let $\mathcal{P}$ be a spatial path defined in an inertial frame and parametrized by a scalar variable γ (e.g., arc-length of the path). Normally, $\gamma \in \Omega := [a, b]$ where $a, b \in \mathbb{R}$ are values of γ corresponding to the

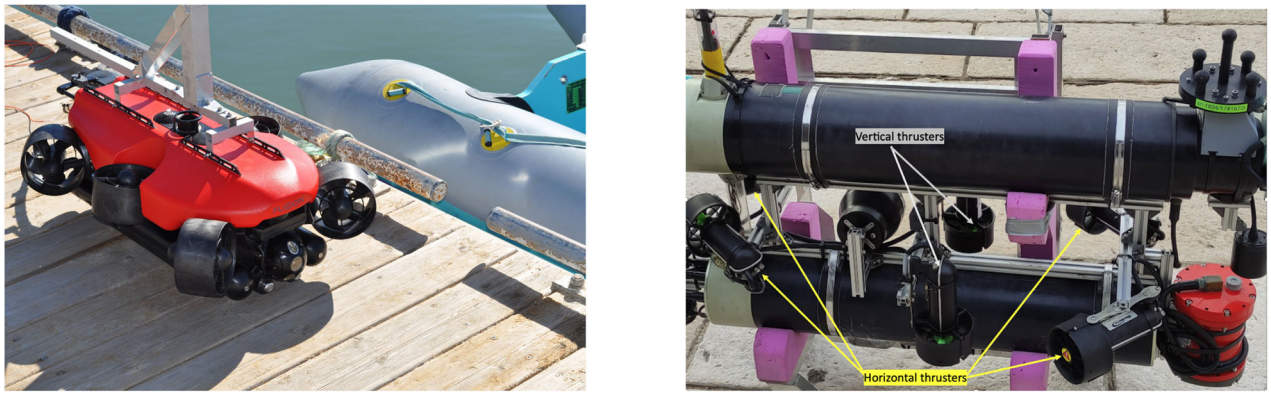

FIGURE 6 Over-actuated robotic vehicles. Left: Fusion vehicle, produced by Strategic Robot Systems company equipped with the Guidance, Navigation, and Control systems developed by IST Lisbon; Right: M-Vector vehicle, developed by IST Lisbon. Notice in both vehicles the existence of 4 thrusters in the horizontal plane (2 at the bow and 2 at the stern, installed at slant angles), capable of imparting directly forces along the longitudinal and lateral axis and torque about the vertical axis. [Color figure can be viewed at wileyonlinelibrary.com]

points at beginning and end of the path. The position of a generic point P on the path in the inertial frame $\{\mathcal{I}\}$ is described by vector

$$\mathbf{p}_d(\gamma) = [x_d(\gamma), y_d(\gamma)]^\top \in \mathbb{R}^2. \tag{5}$$

At P, there are two frames adopted in the literature to formulate the path following problem, that is, to describe the position error between the vehicle and the path. Namely, the *Frenet–Serret* (F-S) and the *Parallel Transport* (P-T) frames that are described next.

2.2.1 | Frenet–Serret frame, Gray et al. (2006)

The F-S frame is used in the path following methods described in Kaminer et al. (2012); Lapierre et al. (2003); Micaelli and Samson (1993). A detailed description of this frame for 3D curves can be found in Gray et al. (2006). In the present paper, because we only consider path following in 2D we simplify the frame for 2D curves as follows, see Figure 7. Formally, let

$$\mathbf{t}(\gamma) = \frac{\mathbf{p}_d'(\gamma)}{\left\|\mathbf{p}_d'(\gamma)\right\|}, \mathbf{n}(\gamma) = \frac{\mathbf{t}'(\gamma)}{\|\mathbf{t}'(\gamma)\|} \tag{6}$$

be the basis vectors defining the F-S frame at the point $\mathbf{p}_d(\gamma)$ where, for every differentiable $\mathbf{f}(x)$, $\mathbf{f}'(x) \triangleq \partial \mathbf{f}(x)/\partial x$. These vectors define the unit tangent and *principle* unit normal respectively to the path at the point determined by γ. The curvature $\kappa(\gamma)$ of the path at that point is given by

$$\kappa(\gamma) = \|\mathbf{t}'(\gamma)\|. \tag{7}$$

As can be seen in the formula for computing the normal vector in (6), the main technical problem with the F-S frame is that it is not well-defined for paths that have a vanishing second derivative (i.e., zero curvature) such as straight lines or nonconvex curves. The other alternative frame, called Parallel-Transport frame, overcomes this limitation and is presented next.

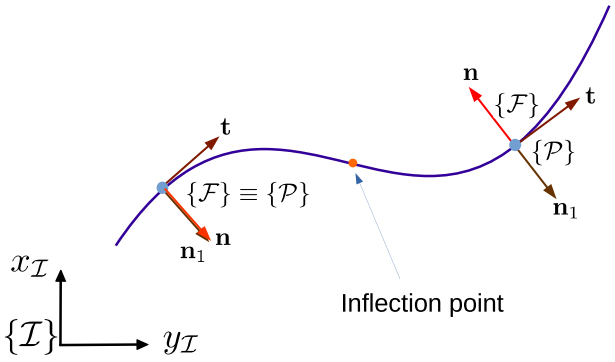

FIGURE 7 An illustration of Frenet-Serret $\{\mathcal{F}\}$ and Parallel Transport $\{\mathcal{P}\}$ frames in 2D. Notice how at the inflection point the normal vector $\mathbf{n}$ of the Frenet–Serret frame inverts its direction. [Color figure can be viewed at wileyonlinelibrary.com]

2.2.2 | Parallel–Transport frame, Hanson and Ma (1995)

This frame was introduced in Hanson and Ma (1995) and used for the first time in the path following method of Kaminer et al. (2010). The P-T frame is based on the observation that, while the tangent vector for a given curve is unique, we may choose any convenient arbitrary normal vector so as to make it perpendicular to the tangent and vary smoothly throughout the path regardless of the curvature (Hanson & Ma, 1995).

In 2D, a simple way to define the P-T frame is as follows. First, specify the tangent basic vector $\mathbf{t}$ as in (6). The second basic vector, called normal vector $\mathbf{n}_1$, is obtained by rotating the tangent vector 90 degree clockwise. This, as shown in Breivik and Fossen (2005), is equivalent to translating $\{\mathcal{I}\}$ to the "reference point" P and then rotating it about the z-axis by the angle

$$\psi_{\mathcal{P}} = \arctan\left(\frac{y_d'(\gamma)}{x_d'(\gamma)}\right). \tag{8}$$

TABLE 1 Principles of path following methods

u	Principles: Stabilizing e in path frames	Principles: Stabilizing e in the body frame
u, ψ	Breivik and Fossen (2005), Fossen et al. (2003), Maurya et al. (2009), Papoulias (1991)	
u, r	Hung et al. (2018), Lapierre et al. (2003), Micaelli and Samson (1993), Yu et al. (2015)	Aguiar and Hespanha (2007), Alessandretti et al. (2013)

The difference between the F-S frame and the P-T frame is illustrated in Figure 7. With the F-S frame the normal component always points to the center of curvature thus, its direction switches at inflection points, while the P-T frame has no such discontinuities. From a path following formulation standpoint, with the F-S frame, the path following error is not well-defined at inflection points because the cross-track error (the position error projected on the normal vector) switches sign, which is not the case with the P-T frame.

Remark 1. Another way of propagating a P-T frame along the path is to use the algorithm proposed in Hanson and Ma (1995). While this algorithm is general and efficient for 3D, it is unnecessarily complicated for 2D curves.

2.3 | Path following formulation

With the concepts and notation described above, the path following problem is stated next, see also Figure 4.

Path following problem in 2D: *Given the 2-D spatial path* $\mathcal{P}$ *described by* (5) *and a vehicle with the kinematics model described by* (2), *derive a feedback control law for the vehicle's inputs* (u, r) *and possibly for* $\dot{\gamma}$ *or* $\ddot{\gamma}$ *so as to fulfill the following tasks:*

1. *Geometric task: steer the position error* $\mathbf{e} \triangleq \mathbf{p} - \mathbf{p}_d$ *s.t.*

$$\lim_{t\to\infty} \mathbf{e}(t) = \mathbf{0}, \tag{9}$$

   where $\mathbf{p}_d$ *is the inertial position of a* "reference point" *P on the path, the temporal evolution of which can be chosen in a number of ways as discussed later.*
2. *Dynamic task: ensure that the vehicle's forward speed tracks a positive desired speed profile* $U_d = U_d(\gamma, t)$, that is

$$\lim_{t\to\infty} u(t) - U_d(t) = 0. \tag{10}$$

In what follows, let $u_{\mathcal{P}}$ be the speed of the "reference point" P with respect to the inertial frame, that is,

$$u_{\mathcal{P}} = \frac{d\mathbf{p}_d}{dt} = \left\|\mathbf{p}_d'(\gamma)\right\| \dot{\gamma}. \tag{11}$$

Should the vehicle achieve precise path following, both the vehicle and the point P will move with the desired speed profile U_d, that is, $u = u_{\mathcal{P}} = U_d$. In this case the dynamics task in (10) is equivalent to requiring

$$\lim_{t\to\infty} \dot{\gamma}(t) - v_d(\gamma, t) = 0, \tag{12}$$

where v_d is the desired speed profile for $\dot{\gamma}$, defined by

$$v_d(\gamma, t) \triangleq \frac{1}{\left\|\mathbf{p}_d'(\gamma)\right\|} U_d(\gamma, t). \tag{13}$$

In path following, the point P on the path plays the role of a "reference point" for the vehicle to track. This point can be chosen as the nearest point to the vehicle (Micaelli & Samson, 1993), that is, the orthogonal projection of the vehicle on the path (in case it is well defined), or can be initialized arbitrarily anywhere on the path with its evolution controlled through $\dot{\gamma}$ (Breivik & Fossen, 2005; Lapierre et al., 2003) or $\ddot{\gamma}$ (Aguiar & Hespanha, 2007) to achieve the path following objectives. In the latter cases, $\dot{\gamma}$ or $\ddot{\gamma}$ are considered as the controlled input of the dynamics of the "reference point," affording an extra freedom in the design of path following controllers.

2.4 | Common principles of path following methods

Although there are a variety of path following methods described in the literature, most of them can be categorized as in Table 1. The first category includes the methods that aimed to stabilize the position error in a frame attached to a *reference point* point moving along the path (e.g., F-S or P-T frames), whereas the second category consists of the methods that aim to stabilize the position error in the vehicle's body frame. The origin of the first method can be traced back to the work of Micaelli and Samson (1993) addressing the path following problem of unicycle-type and two-steering-wheels mobile robots. The core idea in this work was then adopted to develop more advanced path following algorithms in Hung et al. (2018); Lapierre et al. (2003); Yu et al. (2015). Later, we shall see that Line-of-Sight (LOS), a well-known path following method and widely used in marine craft (Fossen et al., 2003) can be categorized in this group as well. The second approach was proposed by Aguiar and Hespanha (2007) and further developed in Alessandretti et al. (2013) to handle the vehicle's input and state constraints.

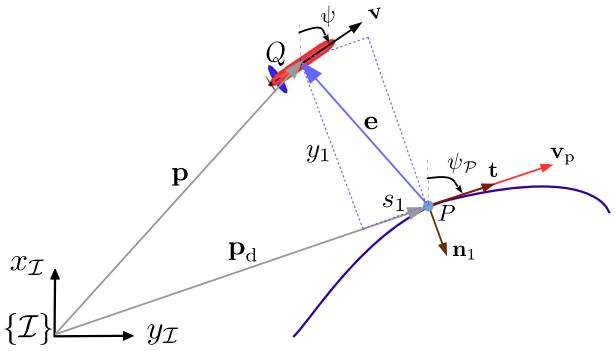

FIGURE 8 A geometric illustration of the methods in Section 3.1. P is the "reference point" that the vehicle must track to achieve path following [Color figure can be viewed at wileyonlinelibrary.com]

3 | BASIC PATH FOLLOWING METHODS

In this section we study path following methods for vehicles whose motion is described by the kinematic model (2). In Section 4 we will extend the methods to the cases when the vehicle motion is subjected to external disturbances.

3.1 | Methods based on stabilizing the path following error in the path frame

In this section we present a "unified formulation" that is simple but general enough to cover the path following methods in Breivik and Fossen (2005); Fossen et al. (2015); Hung et al. (2018); Lapierre et al. (2003, 2006); Micaelli and Samson (1993). The common principle behind these methods can be summarized in two steps:

- Step 1: derive the dynamics of the path following error between the vehicle and the path in a path frame (e.g., F-S or P-T frame)
- Step 2: drive these errors to zero using nonlinear control techniques to achieve path following, that is, make the vehicle converge to and move along the path with a desired speed profile.

To have a *unified formulation*, instead of using the F-S frame as in Lapierre et al. (2003, 2006); Micaelli and Samson (1993) we use the P-T frame. The main advantage of the P-T frame in comparison with the F-S frame was discussed in Section 2.2, that is, it avoids the singularity when the path has a vanishing second derivative, for example, concave paths. Furthermore, the approach that we describe here is different from the one in Lapierre et al. (2003, 2006); Micaelli and Samson (1993) in that the formulation in this section applies to any path that is not necessarily parameterized by the arc-length.

3.2 | Derivation of the path following error

We now derive the dynamics of the path following errors (position and possibly orientation errors) between the vehicle and the path to be stabilized to achieve path following. The formulation is inspired by the work in Breivik and Fossen (2005); Lapierre et al. (2003); Micaelli and Samson (1993) and is presented next. Let P be a point moving along the path that plays the role of a "reference point" for the vehicle to track so as to achieve path following. Let $\{\mathcal{P}\}$ be the P-T frame attached to this point defined by rotating the inertial frame by angle $\psi_{\mathcal{P}}$, where $\psi_{\mathcal{P}}$ is the angle that the tangent vector at P makes with x_I; see Figure 8. Let $\mathbf{e}_{\mathcal{P}} \triangleq [s_1, y_1]^\top \in \mathbb{R}^2$ be a vector defining the position error between the vehicle and the *reference point* P, where s_1 and y_1 are called *along-track* and *cross-track* errors, respectively. This vector can be viewed as the position vector of the vehicle expressed in $\{\mathcal{P}\}$. According to this definition, it is given by

$$\mathbf{e}_{\mathcal{P}} = R_{\mathcal{I}}^{\mathcal{P}}(\psi_{\mathcal{P}})(\mathbf{p} - \mathbf{p}_{\mathrm{d}}), \tag{14}$$

where $\mathbf{p} = [x, y] \in \mathbb{R}^2$ is the position of the vehicle, $\mathbf{p}_{\mathrm{d}}$ is given by (5) and $R_{\mathcal{I}}^{\mathcal{P}} \in SO(2)$ is the rotation matrix from $\{\mathcal{I}\}$ to $\{\mathcal{P}\}$, defined as

$$R_{\mathcal{I}}^{\mathcal{P}}(\psi_{\mathcal{P}}) = \begin{bmatrix} \cos(\psi_{\mathcal{P}}) & \sin(\psi_{\mathcal{P}}) \\ -\sin(\psi_{\mathcal{P}}) & \cos(\psi_{\mathcal{P}}) \end{bmatrix}. \tag{15}$$

Note that $R_{\mathcal{I}}^{\mathcal{P}}(\psi_{\mathcal{P}}) = \left[R_{\mathcal{P}}^{\mathcal{I}}(\psi_{\mathcal{P}})\right]^\top$. It is obvious that if $\mathbf{e}_{\mathcal{P}} \to \mathbf{0}$, then the geometrical task in the path following problem will be solved. Taking the time derivative of (14) yields

$$\dot{\mathbf{e}}_{\mathcal{P}} = \left[\dot{R}_{\mathcal{P}}^{\mathcal{I}}(\psi_{\mathcal{P}})\right]^\top (\mathbf{p} - \mathbf{p}_{\mathrm{d}}) + R_{\mathcal{I}}^{\mathcal{P}}(\psi_{\mathcal{P}})(\dot{\mathbf{p}} - \dot{\mathbf{p}}_{\mathrm{d}}). \tag{16}$$

Applying Lemma A.1 (in the Appendix) for the first term of the previous equation we obtain

$$\dot{\mathbf{e}}_{\mathcal{P}} = -S(\boldsymbol{\omega}_{\mathcal{P}})\mathbf{e}_{\mathcal{P}} + R_{\mathcal{I}}^{\mathcal{P}}(\psi_{\mathcal{P}})\dot{\mathbf{p}} - R_{\mathcal{I}}^{\mathcal{P}}(\psi_{\mathcal{P}})\dot{\mathbf{p}}_{\mathrm{d}}, \tag{17}$$

where $S(\boldsymbol{\omega}_{\mathcal{P}}) \in \mathbb{R}^{2\times 2}$ is a skew symmetric matrix parameterized by $\boldsymbol{\omega}_{\mathcal{P}} = [r_{\mathcal{P}}, 0]^\top \in \mathbb{R}^2$ which is the angular velocity vector of $\{\mathcal{P}\}$ respect to $\{\mathcal{I}\}$, expressed in $\{\mathcal{P}\}$. Note that $r_{\mathcal{P}}$ satisfies the relation

$$r_{\mathcal{P}} = \kappa(\gamma) u_{\mathcal{P}}, \tag{18}$$

where $u_{\mathcal{P}}$ is the total speed of P given by (11) and $\kappa(\gamma)$ is the "*signed*" curvature of the path at P, given by

$$\kappa(\gamma) = \frac{x_{\mathrm{d}}'(\gamma) y_{\mathrm{d}}''(\gamma) - x_{\mathrm{d}}''(\gamma) y_{\mathrm{d}}'(\gamma)}{\left\|\mathbf{p}_{\mathrm{d}}'(\gamma)\right\|^3}. \tag{19}$$

Note also that if γ is the arc-length of the path then $\left\|\mathbf{p}_{\mathrm{d}}'(\gamma)\right\| = 1$. In this case, $u_{\mathcal{P}} = \dot{\gamma}$, that is, the speed of the "reference point" equals the rate of change of the path length. Define

$$\psi_{\mathrm{e}} \triangleq \psi - \psi_{\mathcal{P}} \tag{20}$$

as the orientation error between the vehicle's heading and the tangent to the path. Then,

$$R_I^{\mathcal{P}}(\psi_{\mathcal{P}})\dot{\mathbf{p}} \overset{(2),(15),(20)}{=} \begin{bmatrix} u\cos(\psi_e) \\ u\sin(\psi_e) \end{bmatrix}. \tag{21}$$

Furthermore, letting $\mathbf{v}_{\mathcal{P}} \triangleq [u_{\mathcal{P}}, 0]^\top \in \mathbb{R}^2$ be the velocity of P with respect to $\{\mathcal{I}\}$, expressed in $\{\mathcal{P}\}$, yields

$$R_I^{\mathcal{P}}(\psi_{\mathcal{P}})\dot{\mathbf{p}}_d = \mathbf{v}_{\mathcal{P}}. \tag{22}$$

Substituting (21) and (22) in (17) we obtain the dynamics of the position error as

$$\dot{\mathbf{e}}_{\mathcal{P}} = -S(\boldsymbol{\omega}_{\mathcal{P}})\mathbf{e}_{\mathcal{P}} + \begin{bmatrix} u\cos(\psi_e) \\ u\sin(\psi_e) \end{bmatrix} - \begin{bmatrix} u_{\mathcal{P}} \\ 0 \end{bmatrix}. \tag{23}$$

Furthermore, from (20) the dynamics of the orientation error are given by

$$\dot{\psi}_e \overset{(2),(18)}{=} r - \kappa(\gamma)u_{\mathcal{P}}. \tag{24}$$

At this point, it should be clear that the geometric task in the path following problem, stated in Section 2.3, is equivalent to the problem of stabilizing the position error system (23), that is, making $\mathbf{e}_{\mathcal{P}}(t) \to \mathbf{0}$ as $t \to \infty$. In what follows we will describe a number of path following methods available in the literature that solve this problem. These methods are categorized in Table 2. In *Methods 1* and *3*, the "reference point" is chosen as the orthogonal projection of the center of mass of the vehicle on the path, thus the *along-track* error $s_1(t) = 0$ for all t. In this case, only the *cross-track* y_1 needs to be stabilized to fulfill the geometrical task. In contrast, in the other methods the "reference point" is initialized arbitrarily anywhere on the path and its evolution is controlled by assigning a proper law for $\dot{\gamma}$ so as to make the *cross-track* and *along-track* errors converge to zero. In all of the methods, to fulfill the dynamics task in the path following problem (see 10), the linear speed of the vehicle is assigned with the desired speed profile, that is, $u = U_d$.

Remark 2. In the formulation above, if γ is the arc-length of the path, then in (19) $\left\|\mathbf{p}_d'(\gamma)\right\| = 1$ and thus $u_{\mathcal{P}} = \dot{\gamma}$. In this case, the path following error system composed of (23) and (24) resembles the path following error system developed in Lapierre et al. (2003); see equation (5) in Lapierre et al. (2003). Notice that although parameterizing a path by its arc-length is convenient, this is not always possible; elliptical and sinusoidal curves are typical examples, Gray et al. (2006). In the set-up above, the path parameter γ is not necessarily the arc-length, thus making the formulation presented above applicable to any path.

Remark 3. It is very important to note that to obtain $\dot{\psi}_e$ in (24) ψ_e must be differentiable. This implies that ψ and $\psi_{\mathcal{P}}$ must be differentiable as well. Although from a formulation standpoint there is no problem with this, in practice however, the heading sensors that measure ψ normally return values in $[-\pi, \pi]$ or $[0, 2\pi]$, thus the discontinuities happen when the angle changes between $-\pi$ and π or between 0 and 2π. In this situation, to make ψ and $\psi_{\mathcal{P}}$ continuous and differentiable we should open the domain of these angles to $\mathbb{R}$ before using them in path following algorithms involving in using ψ_e. We suggest a simple algorithm to deal with this situation in the path following toolbox that shall be presented in Section 6.

TABLE 2 Methods proposed to stabilize $\mathbf{e}_{\mathcal{P}}$ to zero

PF methods	Drive $\mathbf{e}_{\mathcal{P}}$ to zero by	References
Method 1	u, r	Micaelli and Samson (1993)
Method 2	$u, r, \dot{\gamma}$	Lapierre et al. (2003)
Method 3[a]	u, ψ	Fossen et al. (2003, 2015), Papoulias (1991)
Method 4	$u, \psi, \dot{\gamma}$	Breivik and Fossen (2005)
Method 5[b]	$u, r, \dot{\gamma}$	Hung et al. (2020), Yu et al. (2015)

[a]Line-of-Sight methods.

[b]Nonlinear model predictive control-based methods.

3.2.1 | Method 1 (Micaelli & Samson, 1993): Achieve path following by controlling (u, r)

In this section we will describe the first method named as *Method 1*, proposed in Micaelli and Samson (1993), to solve the path following problem. The main objective is to derive a control strategy for (u, r) to drive the position error in the systems (23) and (24) to zero. In this method, the "reference point" is chosen as the orthogonal projection of the real vehicle on the path, that is, the point on the path closest to the vehicle (if it is well-defined). With this strategy the *along-track* error is always zero, that is, $s_1(t) = 0$ for all t. Thus, we only need to stabilize the *cross-track* y_1 to zero to fulfill the geometry task. The dynamics of the *cross-track* error can be written explicitly from (23) as

$$\dot{y}_1 = -r_{\mathcal{P}}s_1 + u\sin(\psi_e) \overset{s_1=0}{=} u\sin(\psi_e). \tag{25}$$

Define

$$\tilde{\psi} = \psi_e - \delta(y_1, u), \tag{26}$$

where $\delta(y_1, u)$ is a time differentiable design function that can be used to shape the manner in which the vehicle approaches the path. The design of $\delta(y_1, u)$ must satisfy the following condition.

Condition 1 For the design of $\delta y_1, u$ (Lapierre et al., 2003; Micaelli & Samson, 1993.

1. $\delta(0, u) = 0$ for all u.
2. $y_1 u \sin(\delta(y_1, u)) \leq 0$ for all y_1, u.

Taking the time derivative of (26) yields

$$\dot{\tilde{\psi}} \overset{(24)}{=} r - \kappa(\gamma) u_{\mathcal{P}} - \dot{\delta}. \tag{27}$$

Notice also that because $s_1(t) = 0$ for all t, $u_{\mathcal{P}}$ in the above equation is obtained by solving the first equation in (23), yielding

$$u_{\mathcal{P}} = u \frac{\cos(\psi_e)}{1 - \kappa(\gamma) y_1}. \tag{28}$$

We obtain the following theorem.

Theorem 3.1. *Consider a system composed of the dynamics of the cross-track error in (25) and the orientation error in (27). Let*

$$u = U_d, \tag{29}$$

where U_d is the positive desired speed profile for the vehicle to track. Further let

$$r = \kappa(\gamma) u_{\mathcal{P}} + \dot{\delta} - k_1 \tilde{\psi} - k_2 y_1 u \frac{\sin(\psi_e) - \sin(\delta)}{\tilde{\psi}}, \tag{30}$$

where $k_1, k_2 > 0$ are tuning parameters, $\kappa(\gamma)$ is defined in (19), ψ_e and $\tilde{\psi}$ are given by (20) and (26), respectively, $u_{\mathcal{P}}$ given by (28). Then, $y_1(t)$ and $\tilde{\psi}(t)$ converge to zero as $t \to \infty$.

Proof. See the Appendix—in Section A.2.1. □

Notice that because of Condition 1 once $y_1, \tilde{\psi} \to 0$, then $\psi_e \to 0$ as well. This is true because we are assuming that the vehicle is under-actuated and has negligible lateral motion (i.e., the total velocity vector is aligned with the body's longitudinal axis). In this situation, the vehicle velocity's vector will align itself with the tangent vector of the P-T frame. To satisfy Condition 1, $\delta(\cdot)$ can be chosen as

$$\delta(y_1, u) = -\theta \tanh(k_\delta y_1 u), \tag{31}$$

where $\theta \in (0, \pi/2)$ and $k_\delta > 0$ (Lapierre et al., 2006). In summary, this path following method can be implemented with Algorithm 1. The implementation of line 3 in the algorithm is equivalent to solving an optimization problem to find γ^*, where

$$\gamma^* = \underset{\gamma \in \Omega}{\operatorname{argmin}} \|\mathbf{p} - \mathbf{p}_d(\gamma)\|, \tag{32}$$

where $\mathbf{p}_d(\gamma)$ is given by (5). Note also that the computation in Algorithm 1 associated with γ uses γ^*, which is obtained by solving (32). In many cases, for example, straight-line or circumference paths, γ^* can be found analytically.

Algorithm 1 PF algorithm using Method 1

1: For every time t do:
2: **procedure** PF CONTROLLER
3: Find P—the point on the path closest to the vehicle
4: Compute y_1 by (14), ψ_e by (20), and $\tilde{\psi}$ by (26)
5: For inner-loop controllers (see Figure 3):
6: - Compute the desired vehicle's forward speed by (29)
7: - Compute the desired vehicle's yaw rate given by (30)
8: **end procedure**

Remark 4. Because the "reference point" on the path for the vehicle to track is chosen closest to the vehicle, there exists a singularity when $y_1 = 1/\kappa(\gamma)$ which stems from (28). This happens for example when the path is a circumference and the vehicle goes through the center of it.

3.2.2 | Method 2 (Lapierre et al., 2003): Achieve path following by controlling $(u, r, \dot{\gamma})$

In this section we will describe the second method, named as *Method 2*, proposed by Lapierre et al. (2003) to solve the path following problem. The main objective is to derive a control law for $(u, r, \dot{\gamma})$ to drive the position and orientation errors in (23) and (24) to zero to achieve path following. In this method, instead of choosing the "reference point" on the path that is closest to the vehicle as in *Method 1*, this point can be initialized anywhere on the path, and its evolution is controlled by $\dot{\gamma}$ (to be defined later) in order to eliminate the *along-track* error (Lapierre et al., 2003). To eliminate the *cross-track* and orientation errors, this method uses the same controller for u, r as in *Method 1*. We now state the main result of this method, followed by a discussion on the intuition behind this result.

Theorem 3.2. *Consider the path following error system composed of (23) and (24). Let $u = U_d$ (as given by (29)), r be given by (30), and*

$$u_{\mathcal{P}} = u \cos(\psi_e) + k_3 s_1, \tag{33}$$

where $k_3 > 0$. Then, $\mathbf{e}_{\mathcal{P}}(t), \psi_e(t) \to 0$ as $t \to \infty$.

Proof. See in Appendix—Section A.2.2. □

Recall that in (33) s_1 is the *along-track* error defined in (14). Because of the relation in (11), the control law for $\dot{\gamma}$ is given by

$$\dot{\gamma} = u_{\mathcal{P}} / \left\|\mathbf{p}_d'(\gamma)\right\|, \tag{34}$$

where $u_{\mathcal{P}}$ is given by (33). In summary, the path following strategy of this method can be implemented as specified by Algorithm 2.

Algorithm 2 PF algorithm using Method 2

1: Initialize $\gamma(0)$
2: For every time t do:
3: **procedure** PF CONTROLLER
4: Compute the position and the orientation errors using (14) and (20).
5: For inner-loop controllers (see Figure 3):
6: - Compute the desired vehicle's forward speed using (29)
7: - Compute the desired vehicle's yaw rate given by (30)
8: Compute $\dot{\gamma}$ in (34), then integrate it to update the value of γ
9: **end procedure**

The control law for $u_{\mathcal{P}}$ in (33) implies that if the vehicle is behind/ahead of the "reference point" ($s_1 < 0/s_1 > 0$) then the "reference point" decreases/increases its speed. Intuitively, it aims to adjust the speed of the "reference point" to coordinate with the vehicle along the tangent axis of the P-T frame so as to reduce the *along-track error* to zero. Recall that in *Method 1* proposed in Micaelli and Samson (1993) this error is always zero because the "reference point" is chosen as the point on the path that is closest to the vehicle. Compared with *Method 1* this strategy has the following advantages: (i) it doesn't require an algorithm to find a point on the path that is closest to the vehicle and (ii) it avoids the singularity that occurs in the method of Micaelli and Samson (1993) when $y_1 = 1/\kappa(\gamma)$.

3.2.3 | Method 3 (Fossen et al., 2015): LOS path following

We now present the third method, named *Method 3*, to solve the path following problem. The main objective is to derive a control law for (u, ψ) to drive the position error in the system (23) to zero to achieve path following. In the literature, this method is known as LOS method and was described in Fossen et al. (2015). Earlier work on LOS methods for straight-lines can be found in Lekkas (2013); Fossen et al. (2003); Papoulias (1991). This method can be also found in Liu et al. (2016). The LOS method is similar to *Method 1* in the sense that the "reference point" is chosen as the orthogonal projection of the vehicle onto the path. Thus, the along *track-error* $s_1(t) = 0$ for all t. However, it is different from *Method 1* in that it derives a control law for the vehicle's heading ψ to achieve path following, instead of the heading rate r as in *Method 1*. We recall from *Method 1* that because $s_1(t) = 0$ the dynamics of *cross-track* are given by (25). In the present method ψ_e is viewed as a "control input" whose control law must be chosen to stabilize the *cross-track* error to zero.

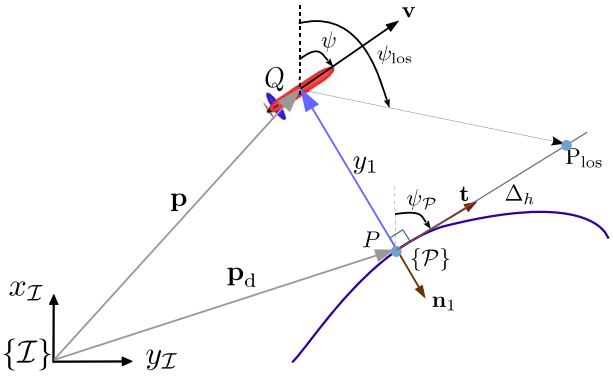

FIGURE 9 Geometric illustration of LOS method, where y_1 is the cross-tracking error [Color figure can be viewed at wileyonlinelibrary.com]

Theorem 3.3. *Consider the cross track error system described by (25). Let the vehicle's forward speed u be given by (29). Let the control law for ψ_e is given by*

$$\psi_e = \arctan\left(-\frac{y_1}{\Delta_h}\right), \tag{35}$$

where $\Delta_h > 0$ is a tuning parameter. Then, $y_1(t) \to 0$ as $t \to \infty$. Because of the relation in (20), the control law for the vehicle's heading is given by

$$\psi = \underbrace{\psi_{\mathcal{P}} + \arctan\left(-\frac{y_1}{\Delta_h}\right)}_{\psi_{los}}. \tag{36}$$

Proof. See Section A.2.3 of the Appendix. □

In the literature, Δ_h is often referred as the *look-ahead distance* (Lekkas, 2013; Fossen et al., 2015) and ψ_{los} is called the light-of-sight angle that the heading of the vehicle should reach to achieve path following. An illustration of the relation between these variables is shown in Figure 9. Notice that Δ_h can be time varying and used to shape the convergence behavior towards the tangent (longitudinal) axis of $\{\mathcal{P}\}$. The larger value of Δ_h, the slower will the convergence be, but this in turn will require less aggressive turning maneuvers in order for the vehicle to reach the path. In summary, the path following *Method 3* can be implemented using Algorithm 3.

Algorithm 3 PF algorithm using Method 3

1: For every time t do:
2: **procedure** PF CONTROLLER
3: Find P—the point on the path closest to the vehicle
4: For inner-loop controllers (see Figure 3):
5: - Compute the desired vehicle's forward speed using (29)
6: - Compute the desired vehicle's heading angle using (36).
7: **end procedure**

FIGURE 10 Stanley robotic car (Thrun et al., 2007) [Color figure can be viewed at wileyonlinelibrary.com]

Remark 5. The control law for the steering angle of the vehicle in (36) can be rewritten in the Body frame as

$$\psi_{\mathcal{B}} = \psi_e + \arctan\left(-\frac{y_1}{\Delta_h}\right), \tag{37}$$

where ψ_e is given by (20).

It is interesting that this control law for the steering angle is equivalent to the one used in the autonomous car called Stanley who won the DARPA Grand Challenge - the competition for autonomous driving in unrehearsed off-road terrain in 2005 (Thrun et al., 2007) (Figure 10).

Remark 6. The cross track error $y_1(t)$ in system (25) can also be driven to zero with the control law ψ_e given by

$$\psi_e = \arcsin\left(\text{sat}\left(-\frac{1}{u}k_1 y_1\right)\right) \tag{38}$$

with any $k_1 > 0$. In (38), $sat(\cdot)$ is a saturation function that returns values in $[-1, 1]$, thus guaranteeing that the arcsin function is well defined. With (38), the dynamics of the cross track error are given by

$$\dot{y}_1 = -u\text{sat}\left(-\frac{1}{u}k_1 y_1\right),$$

from which it is easy to see that $y_1(t)$ converges to zero as $t \to \infty$, Maurya et al. (2009).

3.2.4 | Method 4 (Breivik & Fossen, 2005): Achieve path following by controlling $(u, \psi, \dot{\gamma})$

This path following method is described in Breivik and Fossen (2005) and has a similar principle with that in Lapierre et al. (2003); Rysdyk (2003). The idea behind this method is quite similar to that in *Method 2*; however, instead of achieving path following by controlling the heading rate r, the authors propose a control law for ψ.

Theorem 3.4. *Consider the position error system described by* (23). *Let the vehicle's forward speed* u *be given by* (29), *the speed of the "reference point"* $Pu_{\mathcal{P}}$ *be given by* (33), *and* ψ_e *given by* (35). *Then,* $\mathbf{e}_{\mathcal{P}} = \mathbf{0}$ *is UGAS.*

Proof. See Section A.2.4 in in the Appendix. □

Because of relation (20) the control laws for the vehicle's heading and $\dot{\gamma}$ are given by (36) and (34), respectively. The path following method can be implemented as specified in Algorithm 4. It can be seen from Algorithms 2 and 4 that the two are quite similar except that in the latter the vehicle is guided by controlling its heading, whereas in the former this is done by controlling its heading/yaw rate.

Algorithm 4 PF algorithm using Method 4

1: Initialize $\gamma(0)$
2: For every time t do:
3: **procedure** PF CONTROLLER
4: Compute the position errors s_1, y_1 using (14)
5: For inner-loop controllers (see Figure 3):
6: - Compute the desired vehicle's forward speed using (29)
7: - Compute the desired heading angle using (36).
8: Compute $\dot{\gamma}$ using (34), then integrate it to update the value of γ
9: **end procedure**

Remark 7. If the vehicle's heading in Theorem 3.4 is replaced by (38), then the path following error $\mathbf{e}_{\mathcal{P}}$ converges to zero asymptotically as well. This idea was presented in Ribeiro (2013) and can be proved simply as in the proof of Theorem 3.4.

3.2.5 | Method 5 (Hung et al., 2020; Yu et al., 2015): NMPC-based path following

The path following methods we have studied so far are conceptually simple in terms of design and implementation; however, they do not take into account the vehicle's physical constraints (e.g., maximum and minimum yaw rate). As a consequence, proper care must be taken to ensure that the resulting systems end up operating in a small region where the control law for the unconstrained system does not violate the constraints. To deal with the vehicle's constraints explicitly, in this section we will present an optimization based control strategy called model predictive control to solve the path following problem (Hung et al., 2020, 2018; Yu et al., 2015).

First, define

$$\mathbf{x}_{\mathcal{P}} \triangleq \left[\mathbf{e}_{\mathcal{P}}^{\top}, \psi_e, \gamma\right]^{\top} \in \mathbb{R}^4$$

as the state of the complete path following system where the position error $\mathbf{e}_{\mathcal{P}}$ is defined by (14) and the orientation error is defined by (20). If we assign to the vehicle the

desired speed profile, that is, $u = U_d$ then the dynamics of the complete path following system can be rewritten from (23) and (24) as

$$\dot{\mathbf{x}}_{\mathcal{P}} = \mathbf{f}(\mathbf{x}_{\mathcal{P}}, \mathbf{u}_{\mathcal{P}}) \triangleq \begin{bmatrix} -\left\|\mathbf{p}_d'(\gamma)\right\| v_\gamma (1 - \kappa(\gamma) y_1) + U_d \cos(\psi_e) \\ -\kappa(\gamma) s_1 \left\|\mathbf{p}_d'(\gamma)\right\| v_\gamma + U_d \sin(\psi_e) \\ r - \kappa(\gamma) \left\|\mathbf{p}_d'(\gamma)\right\| v_\gamma \\ v_\gamma \end{bmatrix}, \tag{39}$$

where $\mathbf{u}_{\mathcal{P}} \triangleq [r, v_\gamma]^\top \in \mathbb{R}^2$ is the input of the system, which is constrained in the set $\mathbb{U}_{\mathcal{P}}$ given by

$$\mathbb{U}_{\mathcal{P}} \triangleq \{(r, v_\gamma) : v_{\min} \le v_\gamma \le v_{\max}, |r| \le r_{\max}\}. \tag{40}$$

Here, $r_{\max}$ arises from the physical limitations of the vehicle while $v_{\min}$ and $v_{\max}$ are design parameters. In order for the path following problem to be solvable, the bounds on v_γ can be chosen such that $v_{\min} \le 0$ and

$$v_{\max} > v_d^* \triangleq \max(v_d), \tag{41}$$

where v_d is given by (13). Intuitively, the conditions on the bounds of v_γ ensure that the "reference point" on the path that the vehicle must track has enough speed to coordinate with the vehicle to achieve path following and also to track the desired speed profile v_d. The main objective now is to find an MPC control law for $\mathbf{u}_{\mathcal{P}}$ to stabilize the position orientation errors in the system (39) to zero. To this end, we define a finite horizon open loop optimal control problem (FOCP) that the MPC solves at every sampling time as follows:

Definition 3.5. FOCP-1:

$$\min_{\bar{\mathbf{u}}_{\mathcal{P}}(\cdot)} J_{\mathcal{P}}(\mathbf{x}_{\mathcal{P}}(t), \bar{\mathbf{u}}_{\mathcal{P}}(\cdot)) \tag{42}$$

subject to

$$\dot{\bar{\mathbf{x}}}_{\mathcal{P}}(\tau) = \mathbf{f}_{\mathcal{P}}(\bar{\mathbf{x}}_{\mathcal{P}}(\tau), \bar{\mathbf{u}}_{\mathcal{P}}(\tau)), \tau \in [t, t + T_p], \bar{\mathbf{x}}_{\mathcal{P}}(t) = \mathbf{x}_{\mathcal{P}}(t), \tag{43a}$$

$$\bar{\mathbf{u}}_{\mathcal{P}}(\tau) \in \mathbb{U}_{\mathcal{P}}, \tau \in [t, t + T_p], \tag{43b}$$

$$(\bar{\mathbf{e}}_{\mathcal{P}}(\tau), \bar{\psi}_e(\tau)) \in \mathbb{E}_{\mathcal{P}}, \tau \in [t, t + T_p] \tag{43c}$$

with

$$J_{\mathcal{P}}(\mathbf{x}_{\mathcal{P}}(t), \bar{\mathbf{u}}_{\mathcal{P}}(\cdot)) \triangleq \int_t^{t+T_p} \left\| \begin{bmatrix} \bar{\mathbf{e}}_{\mathcal{P}}(\tau) \\ \bar{\psi}_e(\tau) \end{bmatrix} \right\|_Q + \left\| \bar{\mathbf{u}}_a(\tau) \right\|_R d\tau + F_{\mathcal{P}}(\bar{\mathbf{e}}_{\mathcal{P}}(t + T_p), \bar{\psi}_e(t + T_p)),$$

where $Q \in \mathbb{R}^{3\times3}, R \in \mathbb{R}^{2\times2}$ are positive definite matrices, and the notation $\|\mathbf{x}\|_Q = \mathbf{x}^\top Q \mathbf{x}$ for any $\mathbf{x} \in R^n$ and $Q \in \mathbb{R}^{n\times n}$. The cost associated with the input is defined by

$$\mathbf{u}_a = \begin{bmatrix} U_d \cos(\psi_e) - \left\|\mathbf{p}_d'(\gamma)\right\| v_\gamma \\ r - \kappa(\gamma) \left\|\mathbf{p}_d'(\gamma)\right\| v_\gamma \end{bmatrix}. \tag{44}$$

In the FOCP-1, we use the bar notation to denote the predicted variables and to differentiate them from the actual variables which do not have a bar. Specifically, $\bar{\mathbf{x}}_{\mathcal{P}}(\tau)$ is the predicted trajectory of the state $\mathbf{x}_{\mathcal{P}}$, computed using the dynamic model (39) and the initial condition at the time t, driven by the input $\bar{\mathbf{u}}_{\mathcal{P}}(\tau)$ with $\tau \in [t, t + T_p]$ over the prediction horizon $[t, t + T_p]$. The first cost of $J_{\mathcal{P}}(\cdot)$ is associated with the geometrical task, whereas the choice of $\mathbf{u}_a$ in the argument of the cost function to be minimized is motivated by the fact that once $\mathbf{e}_{\mathcal{P}}$ and $\psi_e \to 0, \mathbf{u}_a \to \mathbf{0}$ as well. $\mathbb{E}_{\mathcal{P}}$ and $F_{\mathcal{P}}$ represent the terminal constraints (the terminal set and the terminal cost, respectively), that should be designed appropriately to guarantee "*recursive feasibility*"[1] and "*stability*" of the MPC scheme (Yu et al., 2015).

In the MPC scheme, the FOCP-1 is repeatedly solved at every discrete sampling instant $t_i = iT_s$, $i \in \mathbb{N}_+$, where T_s is a sampling interval. Let $\bar{\mathbf{u}}_{\mathcal{P}}^*(\tau)$, $\tau \in [t, t + T_p]$, be the optimal solution of the FOCP-1. Then, the MPC control law $\mathbf{u}_{\mathcal{P}}(\cdot)$ is defined by

$$\mathbf{u}_{\mathcal{P}}(t) = \bar{\mathbf{u}}_{\mathcal{P}}^*(t), t \in [t_i, t_i + T_s]. \tag{45}$$

In summary, the MPC strategy for the path following problem can be implemented in Algorithm 5.

Algorithm 5 PF algorithm using Method 5

1: Initialize $\gamma(0)$
2: For every time t do:
3: **procedure** PF CONTROLLER
4: Compute the path following errors s_1, y_1, ψ_e using (14) and (20)
5: Solve the FOCP-1 and apply the MPC control law (45) to obtain optimal r, v_γ
6: For inner-loop controllers (see Figure 3):
7: - Compute the desired vehicle's forward speed using (29)
8: - The optimal r is used as the desired vehicle's heading rate
9: Iterate γ with the optimal input v_γ to update γ
10: **end procedure**

Another way of ensuring stability for the path following error system (39) without using the terminal constraints is to impose a "contractive constraint" in the FOCP-1, see (Hung et al., 2020). The "contractive constraint" in the reference is designed based on the knowledge of an existing global nonlinear stabilizing controller for the path following error system, and is advantageous over the NMPC scheme in Yu et al. (2015) in the sense that the origin of path following error is globally asymptotical stable. However, locally, with the NMPC scheme in Yu et al. (2015), the path following error might converge to zero faster.

[1] An MPC scheme is called recursive feasibility if its associated finite optimal control problem is feasible for all t (Mayne et al., 2000).

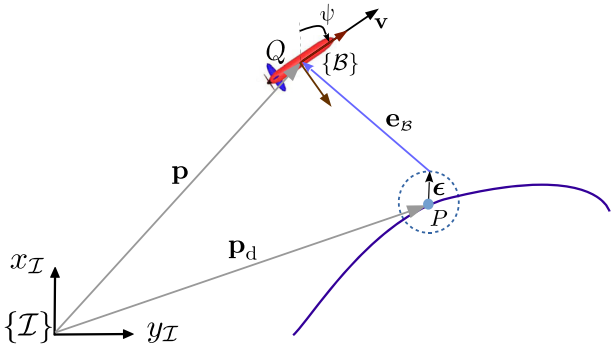

FIGURE 11 Geometric illustration of path following Method 6 (Aguiar & Hespanha, 2007) [Color figure can be viewed at wileyonlinelibrary.com]

Remark 8. While using the terminal constraints in Yu et al. (2015) and the contractive constraint in Hung et al. (2020) are appealing from a theoretical standpoint, for the simplicity in design and implementation in practice they are normally excluded from the finite optimal control problem. In this situation, it is well-known that the convergence of the path following error is guaranteed if the prediction horizon T_p is chosen sufficiently large (Jadbabaie & Hauser, 2005; Mayne et al., 2000). However, it requires effort in tuning the prediction horizon to achieve stability.

Remark 9. Nowadays there are a variety of tools that can support solving the nonlinear optimization problem like FOCP-1. Typical tools that are widely used in MPC's work include Casadi (Andersson et al., 2019), which was used in the work of Hung et al. (2020) and in the simulation toolbox of the present paper, ACADO (Houska et al., 2011), which was used in the work of Batkovic et al. (2019), or MATLAB optimization toolbox (fmincon function).

3.3 | Methods based on stabilizing the path following error in the body frame

3.3.1 | Method 6 (Aguiar & Hespanha, 2007): Achieve path following by controlling $(u, r, \ddot{\gamma})$

We now describe the second approach to the path following problem. It is different from the methods proposed in Section 3.1 in that the position error between the vehicle and the path is formulated in the vehicle's body frame $\{\mathcal{B}\}$, instead of a path frame, Aguiar and Hespanha (2007); Vanni (2007). A geometric illustration of this path following method is shown in Figure 11. First let P, whose coordinate is specified by $\mathbf{p}_d(\gamma)$, be the "reference point" on the path that the vehicle should track to achieve path following. Define

$$\mathbf{e}_{\mathcal{B}} = R_{\mathcal{I}}^{\mathcal{B}}(\psi)(\mathbf{p} - \mathbf{p}_d) - \boldsymbol{\epsilon} \tag{46}$$

as the position error between the vehicle and the path resolved in the vehicle's body frame $\{\mathcal{B}\}$, where $\boldsymbol{\epsilon}$ is an arbitrarily small nonzero constant vector and

$$R_{\mathcal{I}}^{\mathcal{B}}(\psi) = \begin{bmatrix} \cos(\psi) & \sin(\psi) \\ -\sin(\psi) & \cos(\psi) \end{bmatrix} \tag{47}$$

is the rotation matrix from the inertial frame $\{\mathcal{I}\}$ to the body frame $\{\mathcal{B}\}$. The reason for introducing $\boldsymbol{\epsilon}$ will become clear later. By definition, if $\mathbf{e}_{\mathcal{B}}$ can be driven to zero then the vehicle will converge to the ball centered at the point P with radius $\|\boldsymbol{\epsilon}\|$, which implies that the vehicle will converge to a neighborhood of the path that can be made arbitrarily small by choosing the size of $\boldsymbol{\epsilon}$. Taking the time derivative of (46) and using (2) and the fact that $R_{\mathcal{B}}^{\mathcal{I}}(\psi) = \left[R_{\mathcal{I}}^{\mathcal{B}}(\psi)\right]^{\top}$ yields

$$\dot{\mathbf{e}}_{\mathcal{B}} = \left[\dot{R}_{\mathcal{B}}^{\mathcal{I}}(\psi)\right]^{\top}(\mathbf{p} - \mathbf{p}_d) + R_{\mathcal{I}}^{\mathcal{B}}(\psi)(\dot{\mathbf{p}} - \dot{\mathbf{p}}_d)$$

Using Lemma A.1 in the Appendix for the first term, and expanding the previous equation we obtain

$$\begin{aligned} \dot{\mathbf{e}}_{\mathcal{B}} &= -S(\boldsymbol{\omega})\mathbf{e}_{\mathcal{B}} - S(\boldsymbol{\omega})\boldsymbol{\epsilon} + \mathbf{v} - R_{\mathcal{I}}^{\mathcal{B}}(\psi)\mathbf{p}_d'(\gamma)\dot{\gamma} \\ &= -S(\boldsymbol{\omega})\mathbf{e}_{\mathcal{B}} + \Delta\mathbf{u} - R_{\mathcal{I}}^{\mathcal{B}}(\psi)\mathbf{p}_d'(\gamma)\dot{\gamma}, \end{aligned} \tag{48}$$

where $\boldsymbol{\omega} = [r, 0]^{\top} \in \mathbb{R}^2$ is the angular velocity vector of $\{\mathcal{B}\}$ respect to $\{\mathcal{I}\}$, expressed in $\{\mathcal{B}\}$, $\mathbf{u} = [u, r]$ and $\Delta = \begin{bmatrix} 1 & \epsilon_2 \\ 0 & -\epsilon_1 \end{bmatrix}$. To address the dynamic task stated in (12), define

$$e_\gamma = \dot{\gamma} - v_d \tag{49}$$

as the tracking error for the speed of the path parameter. By definition, if e_γ can be driven to zero then the dynamics task is fulfilled. The dynamics of this error is given by

$$\dot{e}_\gamma = \ddot{\gamma} - \dot{v}_d. \tag{50}$$

Let $\mathbf{x} = \left[\mathbf{e}_{\mathcal{B}}^{\top}, e_\gamma\right]^{\top} \in \mathbb{R}^3$ be the complete path following error vector. From (48) and (50), the dynamics of the path following error vector can be expressed as

$$\dot{\mathbf{x}} = \begin{bmatrix} -S(\boldsymbol{\omega})\mathbf{e}_{\mathcal{B}} + \Delta\mathbf{u} - R_{\mathcal{I}}^{\mathcal{B}}(\psi)\mathbf{p}_d'(\gamma)\dot{\gamma} \\ \ddot{\gamma} - \dot{v}_d \end{bmatrix}. \tag{51}$$

Theorem 3.6. *Consider the path following error system described by (51). Then, the control law for* $\mathbf{u}$ *and* $\ddot{\gamma}$ *given by*

$$\mathbf{u} = \bar{\Delta}\left(R_{\mathcal{I}}^{\mathcal{B}}(\psi)\mathbf{p}_d'(\gamma)v_d - K_p\mathbf{e}_{\mathcal{B}}\right) \tag{52a}$$

$$\ddot{\gamma} = -k_\gamma e_\gamma + \mathbf{e}_{\mathcal{B}}^{\top}R_{\mathcal{I}}^{\mathcal{B}}(\psi)\mathbf{p}_d'(\gamma) + \dot{v}_d, \tag{52b}$$

render the origin of $\mathbf{x}$ *GES, where* $\bar{\Delta} = \Delta^{\top}(\Delta\Delta^{\top})^{-1}$, K_p *is a positive definite matrix with appropriate dimension, and* $k_\gamma > 0$.

In summary, this path following method can be implemented as described in Algorithm 6.

Algorithm 6 PF Algorithm using Method 6

1: Initialize $\gamma(0)$ and $\dot{\gamma}(0)$
2: For every sampling interval, repeat the following procedure:
3: **procedure** PF CONTROLLER
4: Compute the position error $\mathbf{e}_{\mathcal{B}}$ using (46) and the tracking error e_γ using (49).
5: For inner-loop controllers (see Figure 3):
6: - Compute the desired vehicle's forward speed and yaw-rate using (52a)
7: Compute $\ddot{\gamma}$ using (52b), then integrate it to update the value of γ
8: **end procedure**

Proof. See Section A.2.5 in the Appendix. □

Remark 10. In (52) we can control the evolution of the "reference point" by assigning $\dot{\gamma} = v_{\mathrm{d}}$, instead of using the control law for $\ddot{\gamma}$, as in (52b). It can be easily shown that the origin of the path following error system is still GES as well. However, making $\dot{\gamma} = v_{\mathrm{d}}$ implies that the "reference point" moves without taking into consideration the state of the vehicle. In other words, in this case the vehicle tracks a pure trajectory and this might demand more aggressive manoeuvres.

Remark 11. It is worth noticing that this method is not only applicable to the kinematic model (2) but also to the dynamics model of quad-rotor drone. For further details we refer the reader to the work presented in Jacinto (2021).

3.3.2 | Method 7 (Alessandretti et al., 2013): NMPC-based path following

In this section, we will describe the path following NMPC scheme in Alessandretti et al. (2013). This scheme borrows from the formulation in Section 3.3.1. Using this formalism, path following is achieved by deriving an NMPC control law for $(u, r, \dot{\gamma})$ to i) drive the position error $\mathbf{e}_{\mathcal{B}}$ whose dynamics described by (48) to zero and also to ii) ensure that $\dot{\gamma}$ converges to v_{d} to fulfill the dynamics task in (12). For this purpose, define

$$\mathbf{x}_{\mathcal{B}} \triangleq \left[\mathbf{e}_{\mathcal{B}}^{\top}, \psi, \gamma\right]^{\top} \in \mathbb{R}^4, \mathbf{u}_{\mathcal{B}} \triangleq [u, r, v_\gamma]^{\top} \in \mathbb{R}^3 \tag{53}$$

as the state and input of the complete path following system, respectively. Note that similar to the NMPC scheme in *Method 5* we let $v_\gamma = \dot{\gamma}$ for convenience of presentation. Note also that in the present NMPC scheme, the vehicle's speed is naturally optimized through the NMPC scheme, whereas with the NMPC scheme in *Method 5*, it is assigned directly by the desired speed profile U_{d}. Because the vehicle's speed is constrained due to the vehicle's physical limitations, we define a constraint set for $\mathbf{u}_{\mathcal{B}}$ in (53) as

$$\mathbb{U}_{\mathcal{B}} \triangleq \mathbb{U}_{\mathcal{P}} \times \{u : u_{\min} \leq u \leq u_{\max}\}, \tag{54}$$

where $\mathbb{U}_{\mathcal{P}}$ given by (40) and the bounds on the vehicle's speed vary according to the physical constraints of the vehicle. The dynamics of the complete path following state can be rewritten from (2) and (48) as

$$\dot{\mathbf{x}}_{\mathcal{B}} = \mathbf{f}_{\mathcal{B}}(\mathbf{x}_{\mathcal{B}}, \mathbf{u}_{\mathcal{B}}) \begin{bmatrix} -S(\boldsymbol{\omega})\mathbf{e}_{\mathcal{B}} + \Delta\mathbf{u} - R_I^{\mathcal{B}}(\psi)\mathbf{p}_{\mathrm{d}}'(\gamma)v_\gamma \\ r \\ v_\gamma \end{bmatrix}, \tag{55}$$

The objective of the NMPC scheme is to find an optimal control strategy for $\mathbf{u}_{\mathcal{B}}$ to drive the position error $\mathbf{e}_{\mathcal{B}}$ and the speed tracking error $(v_\gamma - v_{\mathrm{d}})$ to zero. To this end, we now define a FOCP that the NMPC solves at every sampling time as follows:

Definition 3.7. FOCP-2:

$$\min_{\bar{\mathbf{u}}_{\mathcal{B}}(\cdot)} J_{\mathcal{B}}(\mathbf{x}_{\mathcal{B}}(t), \bar{\mathbf{u}}_{\mathcal{B}}(\cdot)) \tag{56}$$

subject to

$$\dot{\bar{\mathbf{x}}}_{\mathcal{B}}(\tau) = \mathbf{f}_{\mathcal{B}}(\bar{\mathbf{x}}_{\mathcal{B}}(\tau), \bar{\mathbf{u}}_{\mathcal{B}}(\tau)), \tau \in [t, t + T_{\mathrm{p}}], \bar{\mathbf{x}}_{\mathcal{B}}(t) = \mathbf{x}_{\mathcal{B}}(t), \tag{57a}$$

$$\bar{\mathbf{u}}_{\mathcal{B}}(\tau) \in \mathbb{U}_{\mathcal{B}}, \tau \in [t, t + T_{\mathrm{p}}], \tag{57b}$$

$$\bar{\mathbf{e}}_{\mathcal{B}}(\tau) \in \mathbb{E}_{\mathcal{B}}, \tau \in [t, t + T_{\mathrm{p}}]. \tag{57c}$$

with

$$J_{\mathcal{B}}(\mathbf{x}_{\mathcal{B}}(t), \bar{\mathbf{u}}_{\mathcal{B}}(\cdot)) \triangleq \int_t^{t+T_{\mathrm{p}}} \|\bar{\mathbf{e}}_{\mathcal{B}}(\tau)\|_Q + \|\mathbf{u}_{\mathrm{b}}(\tau)\|_R + \|v_\gamma(\tau) - v_{\mathrm{d}}(\tau)\|_O \, \mathrm{d}\tau + F_{\mathcal{B}}(\bar{\mathbf{e}}_{\mathcal{B}}(t + T_{\mathrm{p}}))$$

where $Q, R, O \succ 0$ and

$$\mathbf{u}_{\mathrm{b}} = \Delta\mathbf{u} - R_I^{\mathcal{B}}(\psi)\mathbf{p}_{\mathrm{d}}'(\gamma)v_\gamma. \tag{58}$$

In the FOCP, we use the bar notation to denote the predicted variables, to differentiate them from the actual variables which do not have a bar. Specifically, $\bar{\mathbf{x}}_{\mathcal{B}}(\tau)$ is the predicted trajectory of $\mathbf{x}_{\mathcal{B}}$, using the dynamic model (55) and their perspective initial condition at time t, driven by the input $\bar{\mathbf{u}}_{\mathcal{B}}(\tau)$ with $\tau \in [t, t + T_{\mathrm{p}}]$ over the prediction horizon T_{p}. $\mathbb{E}_{\mathcal{B}}$ and $F_{\mathcal{B}}$ are called terminal set and terminal cost, respectively, to be designed appropriately to guarantee "*recursive feasibility*" and "*stability*" of the NMPC scheme (Alessandretti et al., 2013). The choice of $\mathbf{u}_{\mathrm{b}}$ in the cost function to be minimized is motivated by the fact that once $\mathbf{e}_{\mathcal{B}} \to \mathbf{0}$, $\mathbf{u}_{\mathrm{b}} \to \mathbf{0}$ as well.

In the NMPC scheme, the FOCP-2 is repeatedly solved at every discrete sampling instant $t_i = iT_{\mathrm{s}}$, $i \in \mathbb{N}_+$, where we recall that T_{s} is a sampling interval. Let $\bar{\mathbf{u}}_{\mathcal{B}}^*(\tau)$, $\tau \in [t, t + T_{\mathrm{p}}]$, be the optimal solution of the FOCP-2. Then, the NMPC control law $\mathbf{u}_{\mathcal{B}}(\cdot)$ is defined as

$$\mathbf{u}_{\mathcal{B}}(t) = \bar{\mathbf{u}}_{\mathcal{B}}^*(t), t \in [t_i, t_i + T_{\mathrm{s}}]. \tag{59}$$

In summary the NMPC scheme can be implemented as in Algorithm 7.

Algorithm 7 PF algorithm using Method 7

1: Initialize $\gamma(0)$
2: For every time t do:
3: **procedure** PF CONTROLLER
4: Compute the path following errors s_1, y_1, ψ_e using (14) and (20)
5: Solve the FOCP-2 and apply the NMPC control law (45) to obtain optimal values of u, r, v_γ
6: For inner-loop controllers (see Figure 3):
7: - the optimal u is used as the desired vehicle's forward speed.
8: - the optimal r is used as the desired vehicle's heading rate.
9: Iterate γ with the optimal input v_γ to update the value of γ
10: **end procedure**

Note that for simplicity of design and implementation we can exclude the terminal constraints above from the finite optimal control problem. In this situation, we recall that the convergence of the path following error is guaranteed if the prediction horizon T_p is chosen sufficiently large (Jadbabaie & Hauser, 2005; Mayne et al., 2000).

4 | PATH FOLLOWING METHODS IN THE PRESENCE OF EXTERNAL DISTURBANCES

In the previous section, we described a number of path following methods for the nominal kinematic model (2). We now discuss how these methods can be extended to the cases when the vehicle maneuvers in an environment where *unknown constant external disturbances*, for example, wind in the case of UAVs (see Liu et al., 2013) or ocean currents in the case of AMVs (see Rego et al., 2019) are present. In this situation, the vehicle kinematic model in (2) can be extended as

$$\begin{aligned} \dot{x} &= u\cos(\psi) + v_{cx} \\ \dot{y} &= u\sin(\psi) + v_{cy} \\ \dot{\psi} &= r \\ \dot{v}_{cx} &= 0,\ \dot{v}_{cy} = 0, \end{aligned} \tag{60}$$

where v_{cx}, v_{cy} are two components of $\mathbf{v}_c$, that is, $\mathbf{v}_c = [v_{cx}, v_{cy}]^\top \in \mathbb{R}^2$ - the vector that describes the influence of the *external unknown disturbance* in the inertial frame. It is important to note that in (2) u is the longitudinal/surge speed measured with respect to the fluid. In the literature, the effect of a *constant disturbance* can be eliminated using two approaches. The first is to add an integral term in the path following control laws. This is simple but only applicable for straight-line paths. The second uses estimates of the disturbances and is applicable to every path. These approaches are presented next.

Remark 12. Note that in the present paper we only address the cases where the external disturbance is a constant. For more general cases where the disturbance is an unknown sinusoidal signal the reader is referred to the work in Ghorbani (2017) and Ghorbani (2021). The idea behind this work is to use an adaptive internal model to estimate the disturbance and then use it in a combination with the path following *Method 6* to cancel out the disturbance effect.

4.1 | Path following with integral terms

The LOS type path following methods presented in Section 3.2.3 can be extended in a simple manner to handle constant external disturbances. For this purpose, we first re-derive the dynamics of the path following error in the presence of an external disturbance. Given the kinematic model in (60) and following the procedure described in Section 3.2, it can be shown that the dynamics of the position error $\mathbf{e}_{\mathcal{P}}$ (expressed in the P-T frame) are given by

$$\dot{\mathbf{e}}_{\mathcal{P}} = -S(\boldsymbol{\omega}_{\mathcal{P}})\mathbf{e}_{\mathcal{P}} + \begin{bmatrix} u\cos(\psi_e) \\ u\sin(\psi_e) \end{bmatrix} - \begin{bmatrix} u_{\mathcal{P}} \\ 0 \end{bmatrix} + R_I^{\mathcal{P}}(\psi_{\mathcal{P}})\mathbf{v}_c. \tag{61}$$

Recall that $\mathbf{e}_{\mathcal{P}} = [s_1, y_1]^\top$ is defined by (14) while the orientation error ψ_e is defined by (20). Note that in the above equation, the last term represents the influence of the external disturbance $\mathbf{v}_c$. We will show that this disturbance can be eliminated by adding an integral term in the LOS methods provided that the following assumptions are satisfied.

Assumption 4.1. The "reference point" on the path for the vehicle to track is chosen as the closet point to the vehicle, that is, the along-track error $s_1(t) = 0$ for all t.

With the above assumption the dynamics of the *cross track* can be rewritten from (61) as

$$\dot{y}_1 = u\sin(\psi_e) + v_{cy}^{\mathcal{P}}, \tag{62}$$

where

$$v_{cy}^{\mathcal{P}} \triangleq -\sin(\psi_{\mathcal{P}})v_{cx} + \cos(\psi_{\mathcal{P}})v_{cy}. \tag{63}$$

is the external disturbance acting along the coordinate $x_{\mathcal{P}}$ of the P-T frame. If $\mathbf{v}_c$ and $\psi_{\mathcal{P}}$ are constant, then the unknown disturbance $v_{cy}^{\mathcal{P}}$ is constant as well. Thus, by adding an integral terms in the LOS methods presented in Section 3.2.3, the effect of $v_{cy}^{\mathcal{P}}$ on the *cross-track* error can be eliminated.

Theorem 4.2. *Consider the dynamics of the cross-track error given by (62) where $v_{cy}^{\mathcal{P}}$ is assumed to constant (i.e., the path is a straight-line). The control law for ψ_e given by*

$$\begin{aligned} \psi_e &= \arctan\left(-\tfrac{y_1 + \sigma y_{\text{int}}}{\Delta_h}\right) \quad \text{and} \\ \dot{y}_{\text{int}} &= \frac{\Delta_h y_1}{(y_1 + \sigma y_{\text{int}})^2 + \Delta_h^2} \end{aligned} \tag{64}$$

with $\Delta_h, \sigma > 0$ are tuning parameters drives $y_1(t)$ to zero as $t \to \infty$. Because of the relation in (20), the control law for the vehicle's heading ψ is given by

$$\psi = \psi_e + \psi_{\mathcal{P}}. \tag{65}$$

where in this situation ψ_e is given by 64.

The guidance law in the theorem is referred as integral line-of-sight (ILOS). The convergence of the *cross-track* error is proved in Caharija et al. (2016).

Another way to reject the disturbance is by adding an integral term to (38), yielding a control law for ψ_e, given by

$$\psi_e = \arcsin\left(\text{sat}\left(-\frac{1}{u}k_1 y_1 - k_2 \int_0^t y_1(\tau)d\tau\right)\right), \tag{66}$$

where k_1, k_2 are design parameters. The rational behind this design is that without saturation the resulting *cross-track* error system (62) is given by

$$\dot{y}_1 = -k_1 y_1 - k_2 \int_0^t y_1(\tau)d\tau + v_{cy}^{\mathcal{P}}. \tag{67}$$

It is well-known that the integral term in (67) is capable of canceling the effect of the constant disturbance $v_{cy}^{\mathcal{P}}$. Further, k_1, k_2 can be chosen so as to obtain a desired natural frequency and damping factor for the above second order system, Maurya et al. (2009).

Although adding an integral terms in the LOS guidance methods is a natural and intuitive way to reject a constant disturbance, the main limitation of this approach is that it can only reject the disturbance completely if the path is a straight-line.

4.2 | Path following with estimation of external disturbances

An alternative approach to eliminate the effect of an external disturbance is to estimate it and then use the estimate in the path following algorithms.

4.2.1 | Disturbance estimation

The disturbance can be estimated using the underlying model described in (60) (Aguiar & Pascoal, 2002). Suppose that the navigation system of the vehicle provides estimate of its position $\mathbf{p} = [x, y]^\top$, longitude/surge speed u wrt. the fluid, and heading ψ. Define

$$\mathbf{x}_c \triangleq \left[\mathbf{p}^\top, \mathbf{v}_c^\top\right]^\top \in \mathbb{R}^4, \mathbf{u} \triangleq [u\cos(\psi), u\sin(\psi)]^\top \in \mathbb{R}^2, \mathbf{y}_c \triangleq \mathbf{p}.$$

From (60) we obtain the linear time invariant system

$$\begin{aligned} \dot{\mathbf{x}}_c &= A_c \mathbf{x}_c + B_c \mathbf{u} \\ \mathbf{y}_c &= C_c \mathbf{x}_c, \end{aligned} \tag{68}$$

where

$$A_c = \begin{bmatrix} 0_{2\times2} & I_{2\times2} \\ 0_{2\times2} & 0_{2\times2} \end{bmatrix} \quad B_c = \begin{bmatrix} I_{2\times2} \\ 0_{2\times2} \end{bmatrix} \quad C_c = [I_{2\times2} \quad 0_{2\times2}].$$

It is easy to check that the observability matrix of the above system computed from the pair (A_c, C_c) is full rank, hence the system (68) is observable, implying that the external disturbance $\mathbf{v}_c$ can be estimated using model (68). Let $\hat{\mathbf{x}}_c$ denote the estimate of $\mathbf{x}_c$. To estimate $\mathbf{v}_c$ (through estimating $\mathbf{x}_c$), one can adopt the estimator given by

$$\begin{aligned} \dot{\hat{\mathbf{x}}}_c &= A_c \hat{\mathbf{x}}_c + B_c \mathbf{u} + K_c(\mathbf{y}_c - \hat{\mathbf{y}}_c) \\ \hat{\mathbf{y}}_c &= C_c \hat{\mathbf{x}}_c, \end{aligned} \tag{69}$$

where $K_c \in \mathbb{R}^{2\times2}$ is the designed matrix, to be defined. Let $\tilde{\mathbf{x}}_c \triangleq \mathbf{x}_c - \hat{\mathbf{x}}_c$ be the estimation error. Then, it follows from (68) and (69) that

$$\dot{\tilde{\mathbf{x}}}_c = (A_c + K_c C_c)\tilde{\mathbf{x}}_c. \tag{70}$$

We obtain the following result.

Lemma 4.3. *Consider the estimator (69). Then, the origin of the estimation error $\tilde{\mathbf{x}}$ is GES if K is chosen such that $(A_c + K_c C_c)$ is Hurwitz (i.e., the real part of all its eigenvalues are negative).*

We now revisit the path following methods in the previous section and modify them to eliminate the effect of the disturbance by assuming that the disturbance is estimated using the above described method. We show how to achieve this with Method 3 (Section 3.2.3) and Method 6 (Section 3.3.1). The other methods can be extended similarly to deal with external disturbances.

4.2.2 | Method 3 with estimation of external disturbances

Recall that in Method 3 (Section 3.2.3), the "reference point" on the path for the vehicle to track is chosen as the one closest to the vehicle, and therefore the along track error $s_1(t) = 0$ for all t. As a consequence, under the effect of the external disturbance the dynamics of the *cross-track* error satisfy (62). To eliminate the effect of $v_{cy}^{\mathcal{P}}$ in (62) we can add the estimate of the disturbance to the control law (38), yielding

$$\psi = \psi_{\mathcal{P}} + \arcsin\left(\text{sat}\left(-\frac{1}{u}k_1 y_1 - \frac{1}{u}\hat{v}_{cy}^{\mathcal{P}}\right)\right) \tag{71}$$

where

$$\hat{v}_{cy}^{\mathcal{P}} \triangleq -\sin(\psi_{\mathcal{P}})\hat{v}_{cx} + \cos(\psi_{\mathcal{P}})\hat{v}_{cy}. \tag{72}$$

4.2.3 | Method 6 with estimation of the external disturbances

This approach was described in Vanni (2007) and is summarized as follows. With the kinematic model in (60) and following the procedure described in Section 3.3.1, we can show that the dynamics of the position error in the vehicle body frame $\mathbf{e}_{\mathcal{B}}$ (defined by (46)) are given by

$$\dot{\mathbf{e}}_{\mathcal{B}} = -S(\boldsymbol{\omega})\mathbf{e}_{\mathcal{B}} + \Delta\mathbf{u} - R_{\mathcal{I}}^{\mathcal{B}}(\psi)\mathbf{p}_{\mathrm{d}}'(\gamma)\dot{\gamma} + R_{\mathcal{I}}^{\mathcal{B}}(\psi)\mathbf{v}_{\mathrm{c}}. \tag{73}$$

where Δ is a design matrix defined in Section 3.3.1. The last term in the above equation represents the influence of the disturbance on the path following error. Clearly, if $\mathbf{v}_{\mathrm{c}} = 0$ then (73) is the same as (48) (the case without external disturbance). To reject the external disturbance in (73) we modify the control law for the vehicle input in (52a) as

$$\mathbf{u} = \bar{\Delta}\left(R_{\mathcal{I}}^{\mathcal{B}}(\psi)\mathbf{p}_{\mathrm{d}}'(\gamma)v_{\mathrm{d}} - K_{\mathrm{p}}\mathbf{e}_{\mathcal{B}} - R_{\mathcal{I}}^{\mathcal{B}}(\psi)\hat{\mathbf{v}}_{\mathrm{c}}\right), \tag{74}$$

where $\hat{\mathbf{v}}_{\mathrm{c}}$ is the estimate of the external disturbance obtained from estimator (69). Note that the control law for $\ddot{\gamma}$ remains the same as in (52b). With this control law, we obtain the following result.

Lemma 4.4. *Let* $\mathbf{x} = [\mathbf{e}_{\mathcal{B}}, e_{\gamma}]$ *be the path following error whose dynamics are described by* (73) *and* (50). *Let also* $\mathbf{e}_c \triangleq \mathbf{v}_c - \hat{\mathbf{v}}_c$ *be the estimation error of the external disturbance. Then, with the control laws* (74) *and* (52b) *the path following error system composed of* (73) *and* (50) *is input-to-state stable (ISS) with respect to the state* $\mathbf{x}$ *and the input* $\mathbf{e}_c$.

Proof. see Section A.2.6 in the Appendix. □

For the definition of an ISS system, we refer the reader to Definition 7 in Hassan (2002). The ISS property implies that as long as the estimation error $\mathbf{e}_c$ is bounded, then the path following error $\mathbf{x}$ is also bounded. Furthermore, if $\mathbf{e}_c(t) \to \mathbf{0}$ as $t \to \infty$ then $\mathbf{x}(t) \to \mathbf{0}$ as $t \to \infty$.

With the path following controller given in (74) where the ocean current is estimated by (69), the main result in this section is stated in the next theorem.

Theorem 4.5. *Consider the closed-loop path following system composed of the path following error system described by* (73) *and* (50) *and the disturbance estimation system described by* (69).

i. *Let* K_c *be chosen such that* $A_c + K_c C_c$ *is Hurwitz*
ii. *Let* $\mathbf{u}$ *be given by* (74) *and* $\ddot{\gamma}$ *be given by* (52b).

Then, both the path following error $\mathbf{x}(t) = [\mathbf{e}_{\mathcal{B}}(t), e_{\gamma}(t)]$ *and the disturbance estimation error* $\mathbf{e}_{\mathrm{c}}(t)$ *converge to zero as* $t \to \infty$.

Proof. This result is a consequence of lemmas 4.3 and 4.4. □

5 | PATH-FOLLOWING OF FULLY AND OVER-ACTUATED VEHICLES WITH ARBITRARY HEADING

In this section we consider path following for fully-actuated vehicles whose motion is described by the kinematics model (4). We recall that the longitudinal and lateral speeds and heading rate of fully actuated vehicles can be controlled independently. As we shall see, this allows this class of vehicles to follow paths with arbitrary heading assignments.

We start by noticing that the thruster configuration of a vehicle dictates which kinematic variables can be tracked with appropriately designed inner loop controllers. Consider only motions in the 2D plane and let the vehicle be equipped with n horizontal thrusters. Let $\mathbf{f} \coloneqq [f_1, f_2, \ldots f_n]^{\top}$ denote the vector of forces $f_i \in \mathbb{R}^2$; $i = 1, 2, \ldots, n$ generated by thrusters $1, 2, \ldots, n$. Following the SNAME convention, the resulting horizontal forces and torque that they impart to the vehicle can be represented by $F \coloneqq [F_x, F_y]^T$ and N, respectively. Furthermore, $\tau = [F^T, N]^{\top}$ is given by Kf, where $K \in \mathbb{R}^{3\times 2}$ is the thrust allocation matrix that depends on the configuration (both in position and orientation) of the thrusters (Fossen et al., 2009). In what follows we assume that K is full row rank (equal to 3) and n ≥3, with $n = 3$ and $n > 3$ corresponding to a fully actuated and an overactuated vehicle, respectively. In both cases, given a desired vector τ requested by the vehicle's inner loop controllers, and assuming there are no physical limitations on the force vector $\mathbf{f}$, then $\mathbf{f} = K^*\tau$ where K^* is the either the inverse or the Monroe pseudo-inverse of K for $n = 3$ and $n > 3$, respectively (Fossen et al., 2009; Johansen & Fossen, 2013). The more realistic case where there are physical limitations on the force vector $\mathbf{f}$ can be dealt with by solving a constrained optimization problem. Finally, assuming that the force f_i generated by each thruster is approximately given by a quasi steady-state invertible map $f_i = g(u_i)$, where u_i is the control input (e.g., speed of rotation of propeller i), then $u_i = g^{-1}(f_i)K^*\tau$.

For under-actuated vehicles, a proper thruster configuration will only allow for the generation of surge force F_x and yaw torque N, that is, all the elements of the second row of K^* are zero. In this situation, the vehicle's inner loops can control the surge speed u and the yaw ψ or yaw rate r but not sway speed v. For fully actuated vehicles in 2D, since K is full rank, it is also possible to generate a force in sway, Y, which allows for the implementation of an inner loop to control sway speed v. Therefore, we can assign independent values for the total velocity vector $\mathbf{v} = [u, v]^{\top}$ and yaw ψ. The yaw ψ can be assigned independently as a desired value ψ_{d} which may be constant or dependent on the path-following variable γ or time. In view of this, path following with arbitrary heading can be achieved by commanding the total velocity vector in such a way as to make the evolution of $\mathbf{p}$ comply with the path following objectives, that is, converge to a desired path and track a desired velocity profile. Based on the theory exposed in *Method 6*, let $\mathbf{p}_{\mathrm{d}}(\gamma)$ be the "reference point" on the path that the vehicle must track to achieve path following. Define

$$\mathbf{e}_{\mathcal{B}} = R_{\mathcal{I}}^{\mathcal{B}}(\psi)(\mathbf{p} - \mathbf{p}_{\mathrm{d}}) \tag{75}$$

as the position error between the vehicle and the path in the vehicle's body frame $\{\mathcal{B}\}$. Taking the time derivative of (75) and using (4) and the fact that $R_{\mathcal{B}}^{\mathcal{I}}(\psi) = \left[R_{\mathcal{I}}^{\mathcal{B}}(\psi)\right]^{\top}$ yields

$$\begin{aligned}\dot{\mathbf{e}}_{\mathcal{B}} &= \left[\dot{R}_{\mathcal{B}}^{\mathcal{I}}(\psi)\right]^{\top}(\mathbf{p}-\mathbf{p}_{\mathrm{d}}) + R_{\mathcal{I}}^{\mathcal{B}}(\psi)(\dot{\mathbf{p}}-\dot{\mathbf{p}}_{\mathrm{d}}) \\ &= -S(\boldsymbol{\omega})\mathbf{e}_{\mathcal{B}} + \mathbf{v} - R_{\mathcal{I}}^{\mathcal{B}}(\psi)\mathbf{p}_{\mathrm{d}}'(\gamma)\dot{\gamma}.\end{aligned} \tag{76}$$

To address the dynamic task stated in (12), define the speed tracking error of the path parameter e_γ as in (49). If e_γ can be driven to zero then the dynamics task is fulfilled. Let $\mathbf{x} = \left[\mathbf{e}_{\mathcal{B}}^{\top}, e_\gamma\right]^{\top} \in \mathbb{R}^3$ be the complete path following error vector. From (76) and (50), the dynamics of the path following error vector are given by

$$\dot{\mathbf{x}} = \begin{bmatrix} -S(\boldsymbol{\omega})\mathbf{e}_{\mathcal{B}} + \mathbf{v} - R_{\mathcal{I}}^{\mathcal{B}}(\psi)\mathbf{p}_{\mathrm{d}}'(\gamma)\dot{\gamma} \\ \ddot{\gamma} - \dot{v}_{\mathrm{d}} \end{bmatrix}. \tag{77}$$

The main objective now is to derive path following control laws for the vehicle inputs $(u, v, \ddot{\psi})$ to drive $\mathbf{x}$ to zero.

Theorem 5.1. *Consider the path following error system described by (77). Then, the control law for ψ, $\mathbf{v}$, and $\ddot{\psi}$ given by*

$$\psi = \psi_{\mathrm{d}}, \tag{78a}$$

$$\mathbf{v} = R_{\mathcal{I}}^{\mathcal{B}}(\psi)\mathbf{p}_{\mathrm{d}}'(\gamma)v_{\mathrm{d}} - K_{\mathrm{p}}\mathbf{e}_{\mathcal{B}} \tag{78b}$$

$$\ddot{\gamma} = -k_\gamma e_\gamma + \mathbf{e}_{\mathcal{B}}^{\top}R_{\mathcal{I}}^{\mathcal{B}}(\psi)\mathbf{p}_{\mathrm{d}}'(\gamma) + \dot{v}_{\mathrm{d}}, \tag{78c}$$

render the origin of $\mathbf{x}$ GES, where K_{p} is a positive definite matrix with appropriate dimensions, and $k_\gamma > 0$.

Proof. See Appendix - in Section A.2.7. □

In summary, this path following method can be implemented using Algorithm 8.

Algorithm 8 PF Algorithm for fully actuated vehicles

1: Initialize $\gamma(0)$ and $\dot{\gamma}(0)$
2: For every sampling interval, repeat the following procedure:
3: **procedure** PF CONTROLLER
4: Set the yaw angle ψ to the desired value ψ_d as in (78a).
5: Compute position error $\mathbf{e}_{\mathcal{B}}$ using (75) and tracking error e_γ using (49).
6: Compute desired vehicle's velocity $\mathbf{v}$ using (78b)
7: Compute $\ddot{\gamma}$ using (78c), then integrate it to update new value of γ.
8: **end procedure**

6 | MATLAB SIMULATION TOOLBOX AND ROS/GAZEBO IMPLEMENTATION

In the scope of this review paper, both a Matlab toolbox and a set of ROS packages were developed to test and analyze the performance of the path following algorithms described in the previous sections. The end goal is not only to have simulation and field-trial results included in the paper, but also to give the reader tools that will allow for quick testing of the path following algorithms and their integration in guidance, navigation, and control systems.

6.1 | Matlab path following toolbox

The main purpose of the Matlab path following toolbox[2] is to afford the reader a simple and versatile tool to test and access the performance of the path following methods reviewed in the present paper. This toolbox implements the seven methods described in Section 3. The source code is open and therefore anyone is welcome to use, modify or develop new functionalities on top of it. The main components of the toolbox, which implements the seven methods described in Section 3, are illustrated in Figure 12. In what concerns the vehicle's model, with the current version two type of kinematic models were adopted from 2: Type 1, in which the vehicle's inputs are the linear speed and heading rate, and Type 2, in which the vehicle's inputs are the linear speed and heading angle. Model Type 1 is used to test Methods 1, 3, 5, 6, and 7, whereas Model Type 2 is adopted to test Methods 2 and 4. As an example, Figure 13 shows the results of path following simulations using the toolbox for the case of a Bernoulli and a sinusoidal path, coupled with Methods 3 and 1, respectively. An animation video illustrating the performance of the different methods adopted for path following in the case of a Bernoulli path is available in https://youtu.be/XutfsXijHPE.

6.2 | Gazebo/ROS packages

To test the path following algorithms with the Medusa vehicles in a realistic environment. We developed two efficient ROS packages[3] in C++ that are used to run the algorithms on the vehicle's computers. Namely, (i) a paths package for the generation of conveniently parameterized planar paths and (ii) a path following algorithm package. These code libraries communicate with each other according to the diagram in Figure 14, using ROS topics. The paths library implements four simple parametric consisting of: arcs, lines, circles and Bernoulli lemniscates, as illustrated in Figure 15.

The paths library works as a state machine, in that multiple segments that are marked as "composable" can be added together to form more complex shapes. This flexible structure allows for the switching of the path following methods in use and the paths being followed in real time. The complete path following system implemented based on this structure is illustrated in Figure 16.

To allow the reader to perform experiments similar to the ones presented in this paper we provide a realistic simulation environment, resorting to the UUVSimulator Plugin and Gazebo 11 simulator (Manhães et al., 2016), widely used in the robotics community. This makes it possible for the reader to simulate the real conditions found

[2]Link: https://github.com/hungrepo/path-following-Matlab/tree/master/PF-toolbox.
[3]Link: https://github.com/dsor-isr/Paper-PathFollowingSurvey

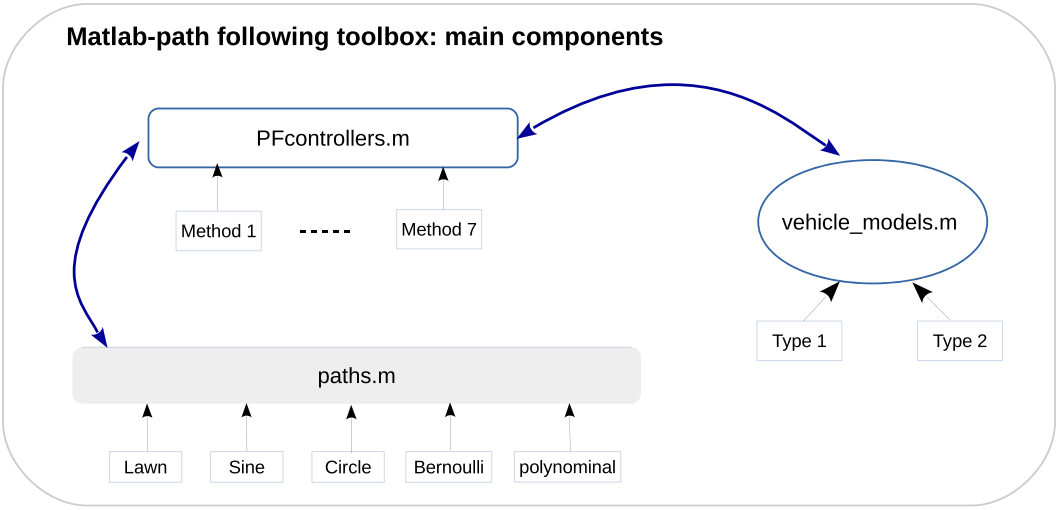

FIGURE 12 Main components of the Matlab path following toolbox [Color figure can be viewed at wileyonlinelibrary.com]

a the Olivais Dock at Parque das Nações, Lisbon, during field trials (Figure 17).

7 | SIMULATIONS AND EXPERIMENTAL RESULTS

In this section, we start by introducing the test set-up adopted for conducting trials with the Medusa class of marine vehicles. This is followed by the presentation of simulation results obtained using the ROS/Gazebo simulator as well as results of field trials.

7.1 | Test set-up

The field tests were conducted with two Medusa class autonomous surface vehicles built at IST. The first is an under-actuated Medusa shown in Figure 5. This type of vehicle has played central roles in many European research projects such as Morph (Caldeira Abreu et al., 2015) and Wimust (Abreu et al., 2016) to support the research in marine science and also in the area of guidance, navigation and control of single and networked multiple autonomous vehicle. The under-actuated Medusa vehicle is equipped with two thrusters located at starboard and portside that allow for the generation of longitudinal forces and torques about the vertical axis. See the corresponding thruster arrangement in Figure 18 (left). This configuration provides two degrees of freedom on the horizontal plane, since the yaw rate and surge speed can be directly controlled, but not the sway speed. The second Medusa, with the thruster configuration shown in Figure 18 (right), is over-actuated in the horizontal plane. This vehicle contains six thrusters, two of which are mounted vertically and four are mounted horizontally in a by-now classical configuration. The thruster configuration in the over-actuated vehicle allows for the generation of longitudinal and lateral forces and torque about the vertical axis. Therefore, this configuration provides three degrees of freedom in the horizontal plane, since yaw rate, surge speed, and sway speed can be controlled directly.

Each vehicle's navigation system builds upon GPS-RTK and AHRS sensors that allow for the computation of its position and orientation. All software modules for navigation and control were implemented using the Robot Operating System (ROS, Melodic version), programmed with C++, and ran on an EPIC computer board (model NANO-PVD5251). More information about the specification of the Medusa class vehicle can be found in Abreu et al. (2016). The implementation of the path following controllers follows the structure depicted in Figure 3. The vehicle's inner-loop controllers include four proportional integrator and derivative (PID)-type controllers to track the references in linear speed, heading, heading rate, and sway (only for over-actuated vehicles) that are generated by the path following controllers.

Snapshots showing the vehicle performing path following can be seen in Figure 19 while its operation can be monitored online with a console shown in Figure 20.

7.2 | Simulation results with the ROS/Gazebo workpackages

Before operating any vehicle in a real world environment, it is common practice to resort to Software-in-the-loop (SITL) simulations to analyze the performance of the controllers developed. Examples of SITL simulations are provided next.

7.2.1 | Results with an under-actuated vehicle

This section describes the results of simulations of an under-actuated vehicle performing a lawnmowing manoeuver using

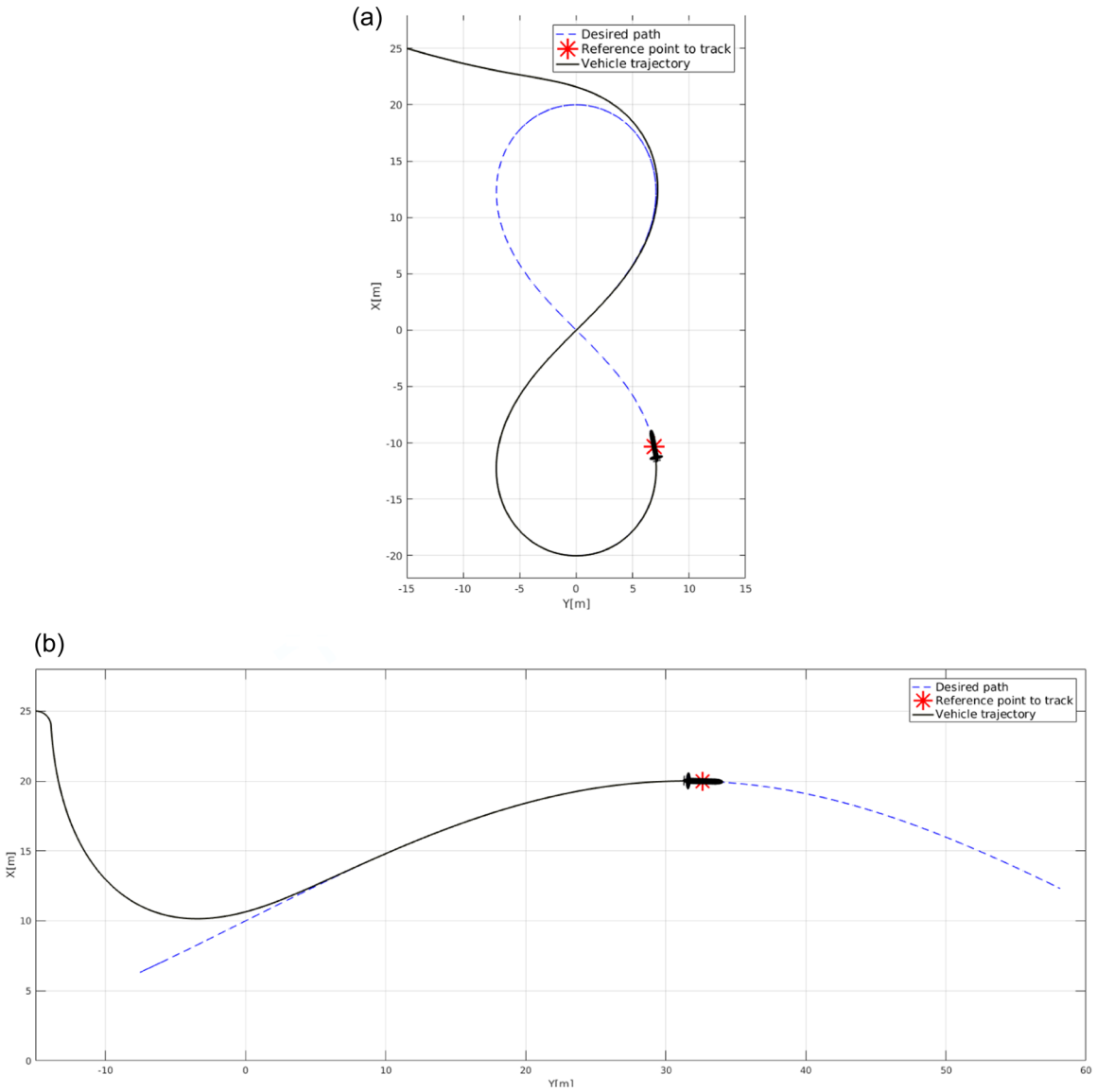

FIGURE 13 Simulation example with different paths and path following methods. (a) Example 1: Bernoulli path with Method 3 and (b) Example 2: Sinusoidal path with Method 1. [Color figure can be viewed at wileyonlinelibrary.com]

Methods 1-4 and 6, in the Gazebo simulator described in Section 6.2. The assigned speed of the vehicle, denoted U_d, is $0.5m/s$. The vehicle paths are shown in Figure 21. We observe some differences in the transient behavior and tracking performance obtained with the different methods. However, we can also observe that the results obtained with Methods 1 and 2 are almost identical. This is a consequence of the fact that in Method 2 the *virtual target* converges fast to the closest point on the path. Ideal comments apply to Methods 3 and 4 because again, in Method 4, the *virtual target* also converges fast to the closest point on the path. It should be noted however that there are practical advantages on the use of virtual targets, such as the lower computational burden. Notice also that for Method 6 the vehicle first rotates in yaw and then accelerates forward, whereas for Methods 1-4, where the surge speed reference does not depend on the yaw, the vehicle rotates with constant surge speed reference.

The corresponding *cross-track* errors y_1 (defined in Section 3.2) are shown in Figure 23. The along-track errors s_1, defined in Section 3.2, are shown in Figure 22. In the figure, only the data for Methods 2, 4, and 6 are shown since for all the other methods the "reference point" is the orthogonal projection of the vehicle's position on the path, which makes the *along-track* error is equal to zero (Figure 23).

From Figures 22 and 23 one can observe that the vehicle takes approximately the same amount of time to converge to the desired path for all the methods. Conversely, during circular segments we observe that Methods 3 and 4 outperform Methods 1 and 2 due to the impact of the performance of the respective inner loop controllers, yaw rate for Methods 1 and 2 and yaw for Methods 3 and 4. From Figure 22 one can observe that the along track error converges to close to zero for *Methods* 2 and 4 and for *Method 6* stabilizes at $1m$ due to the particular definition of the position error (46).

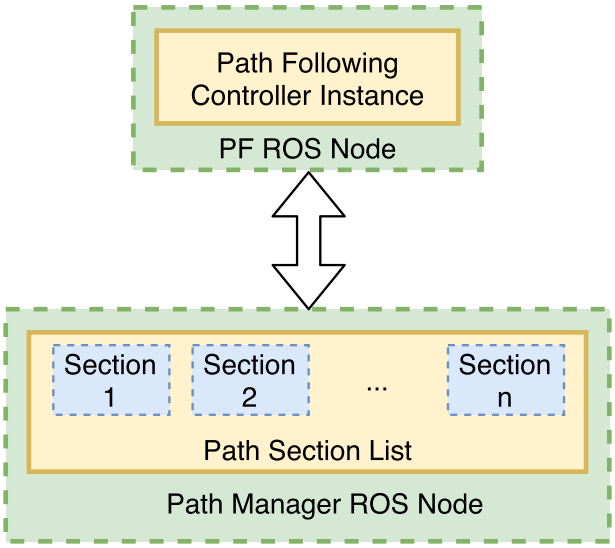

FIGURE 14 ROS node abstraction [Color figure can be viewed at wileyonlinelibrary.com]

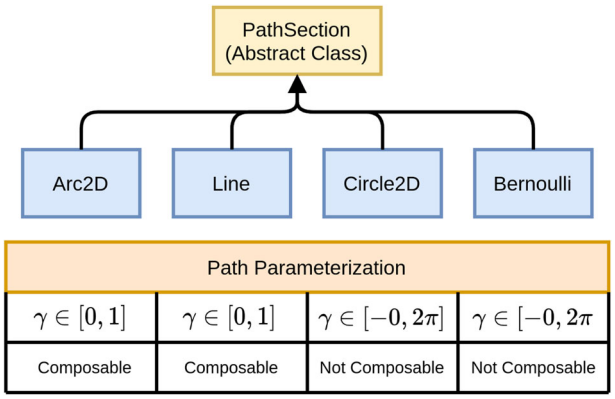

FIGURE 15 Paths library [Color figure can be viewed at wileyonlinelibrary.com]

7.2.2 | Results with an over-actuated vehicle

Figures 24 and 25 show the results of simulating an over-actuated vehicle performing a lawn-mowing maneuver, using the path following method proposed in Section 5. In particular, Figure 24 shows the X-Y path of the vehicle, which was requested to keep its yaw angle offset by 45° clockwise from the tangent to the path.

In Figure 25 it can be observed that the along-track error converges to zero, while the cross-track and heading errors exhibit small static errors in the neighborhood of zero. These errors result from the linear inner-loops implemented, a simplification in the design process which did not take into consideration the impact of the cross-terms in surge, sway and yaw on the dynamics of the vehicle.

7.3 | Experimental results

7.3.1 | Results with an under-actuated vehicle

This section describes the results of experimental tests whereby an under-actuated vehicle was requested to follow the same path repeatedly, using different path following algorithms. As shown in Figure 26, during the tests the vehicle followed a lawnmowing path with a first leg of 30 m heading east, then a half circumference turning clockwise with a radius of 10 m, a second leg of 20 m heading west, another half circumference with a radius of 10*m* turning anticlockwise, and a final leg of 30 m heading east. In another set of tests the vehicle was requested follow a Bernoulli lemniscate with a length of 20 m as shown in Figure 27. In all the tests, the assigned speed of the vehicles U_d is 0.5 m/s. The trials included the use of *Method* 1–4, 6, and the method in Maurya et al. (2009).

Figures 26 and 27 show the paths of the vehicle during the trials, for the different methods. We observe that the vehicles follow their assigned paths and that the transient behavior and tracking performance vary according to the method used.

The *cross-track* errors y_1 (defined in Section 3.2) of the vehicles performing a lawnmowing maneuver using different path following

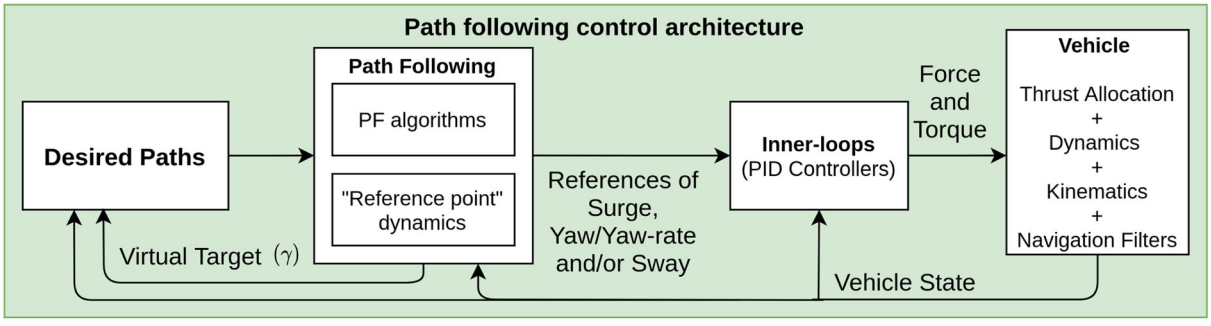

FIGURE 16 Complete path following control system implemented in Medusa vehicles [Color figure can be viewed at wileyonlinelibrary.com]

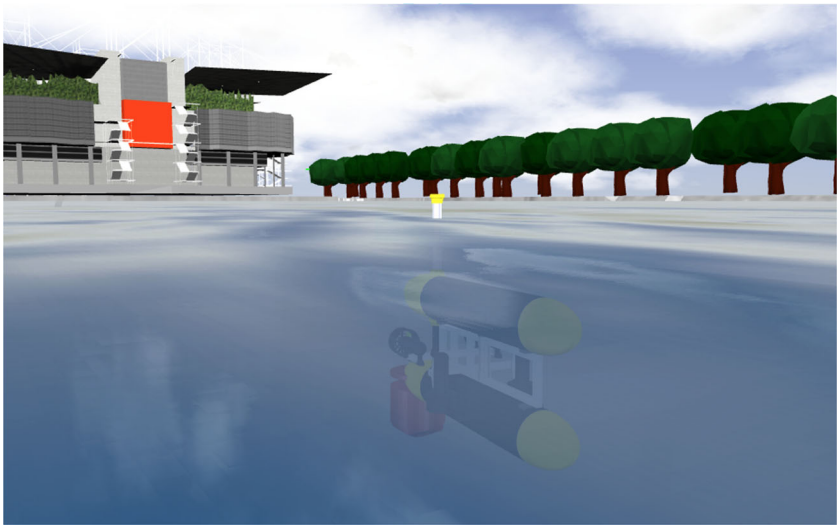

FIGURE 17 Medusa 3-D simulation using Gazebo [Color figure can be viewed at wileyonlinelibrary.com]

FIGURE 18 Top view of the Medusa vehicles showing the thruster configuration. Left: under-actuated Medusa, Right: over-actuated Medusa. [Color figure can be viewed at wileyonlinelibrary.com]

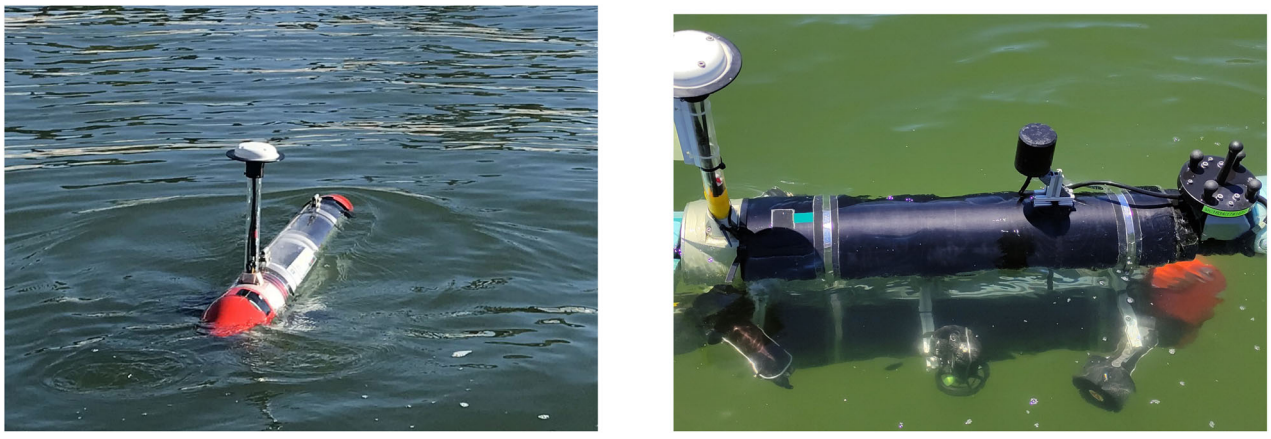

FIGURE 19 The Medusa vehicles performing path following during the trials. Left: under-actuated Medusa (a closer view can be seen in Figure 5a), Right: over-actuated Medusa [Color figure can be viewed at wileyonlinelibrary.com]

methods is shown in Figure 28. Identical results for the case where the path is a Bernoulli lemniscate are shown in 29. From Figures 28 and 29 one can observe that *Methods 3-4* took less time to converge than the other methods.

The along-track errors s_1, defined in Section 3.2, of the vehicle performing a lawnmowing maneuver and following a Bernoulli lemniscate are shown in Figures 30 and 31, respectively. Figures 30 and 31 show the data for *Method 2*, 4 and 6 since for all the other methods the "reference point" is the orthogonal projection of the vehicle's position on the path which makes the *along-track* error equals to zero.

From Figures 30 and 31 one can observe that the along track error converges to close to zero for *Method 2* and 4 and for *Method 6* stabilizes at $1m$ due to the effect of ϵ on the particular

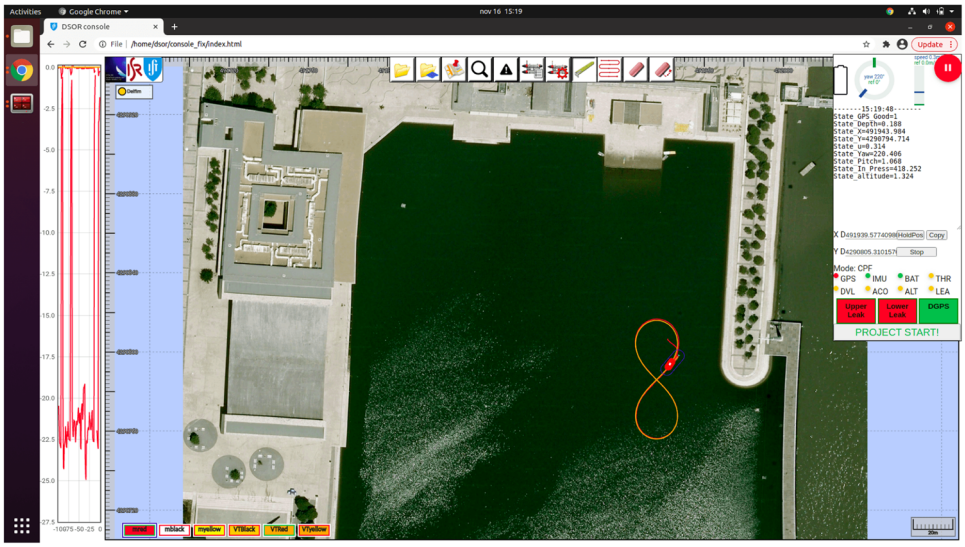

FIGURE 20 Console used to operate the Medusa class vehicles [Color figure can be viewed at wileyonlinelibrary.com]

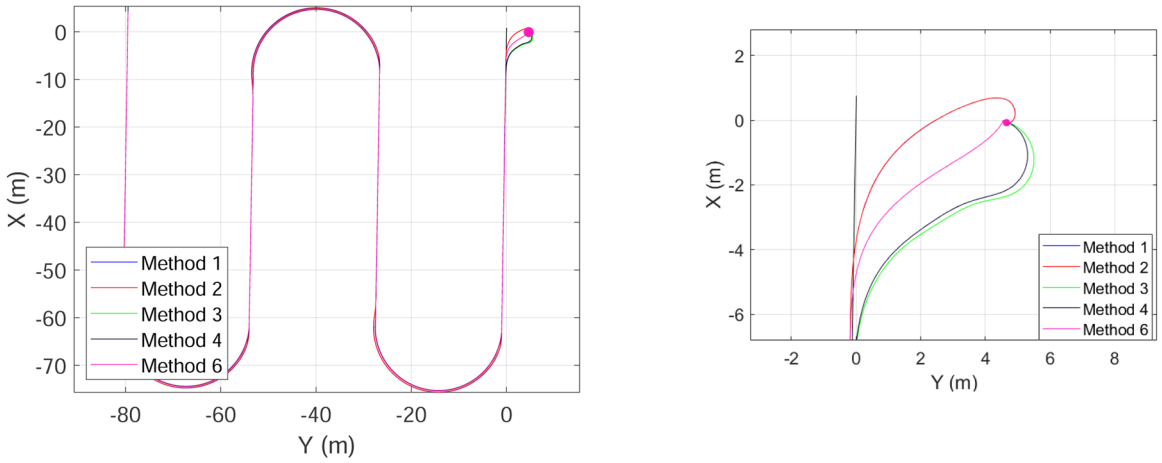

FIGURE 21 Left: vehicle's paths using different path following methods during simulations. Right: Zoom in the paths in transition time. A filled circle indicates the initial position. [Color figure can be viewed at wileyonlinelibrary.com]

definition of the position error, see (46). We can also observe, for the Bernoulli lemniscate mission with *Method 6*, large oscillations at the beginning.

Figures 32 and 33 show the surge speed u of the vehicle for different methods for the lawnmower and Bernoulli lemniscate missions respectively, which clearly indicate that the vehicle asymptotically reaches the desired constant speed profile (0.5 m/s). However they show that the surge speed converged faster to the desired speed for *Method* 3, 4 and the Method in Maurya et al. (2009).

7.3.2 | Results with an over-actuated vehicle

The method for over-actuated vehicles described in Section 5 was tested with a over-actuated Medusa class vehicle following a line due east (along

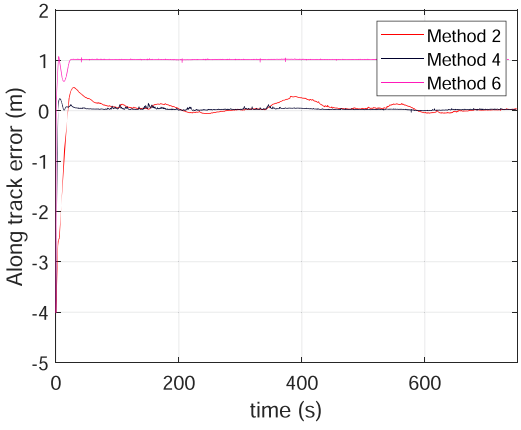

FIGURE 22 *Along-track* errors for different methods during simulations [Color figure can be viewed at wileyonlinelibrary.com]

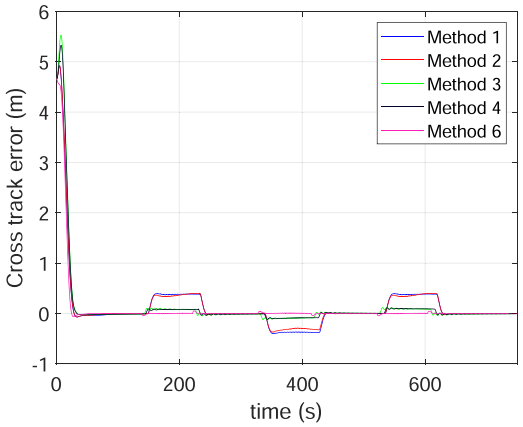

FIGURE 23 *Cross-track* errors for different methods during simulations [Color figure can be viewed at wileyonlinelibrary.com]

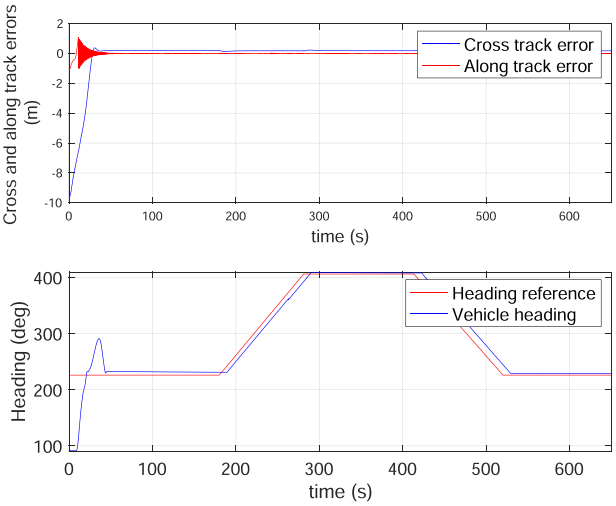

FIGURE 25 *Cross-track and along-track* errors (top); Heading reference and vehicle heading angle (bottom) [Color figure can be viewed at wileyonlinelibrary.com]

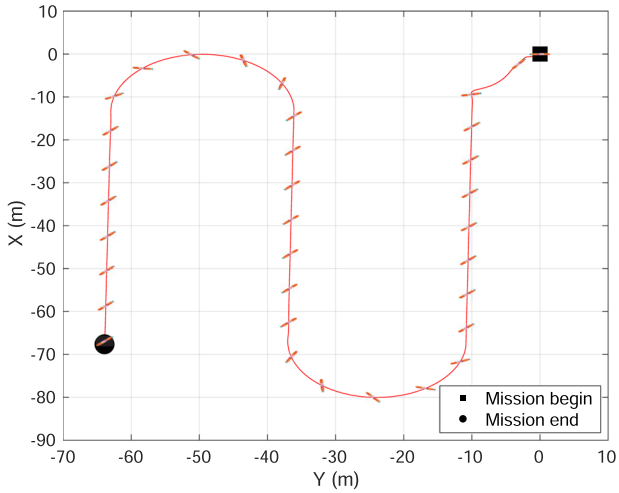

FIGURE 24 Simulated vehicle path using the method described in Section 5 [Color figure can be viewed at wileyonlinelibrary.com]

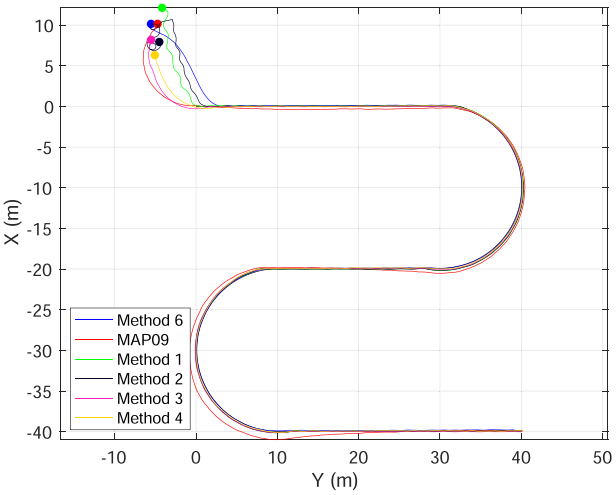

FIGURE 26 Vehicle paths using different path following methods to follow a lawnmower path in real trials. Filled circles are the positions of the vehicle at the beginning of each run. [Color figure can be viewed at wileyonlinelibrary.com]

the Y axis) with a heading ψ of 0°. The obtained path, along and cross track errors, and yaw are shown in Figures 34–36, respectively.

Given the coupling between surge, sway and yaw rate in the dynamics of the vehicle, the tunning of the PID inner loops of the real vehicle is a challenging task and we were not able to obtain adequate performance of circular maneuvers during the field trials with simple linear control laws. From Figure 35 one can observe that the along-track error starts at around 5 m because that is the distance from the initial position of the vehicle to the beginning of the path, but one can observe that the vehicle converges to the beginning of the path in around 70 s. Also one can observe that the cross-track error remains below 1 m during the mission. Finally, from Figure 36 the heading converges to within 15° of the reference. Future work will involve the design of new nonlinear inner loop controllers to achieve better performance.

8 | DISCUSSIONS

8.1 | Advantages and disadvantages of the different path following methods

In this section we discuss some of the benefits and drawbacks of the path following methods reviewed in the previous sections and compare them with other methods in the literature such as the vector field (VF) method described in Nelson et al. (2007). A relative comparison in terms of complexity and flexibility is shown in Figure 37 while Table 3 shows the computation times required to run the path following methods described in Section 3. Note that the computation times are recorded while running

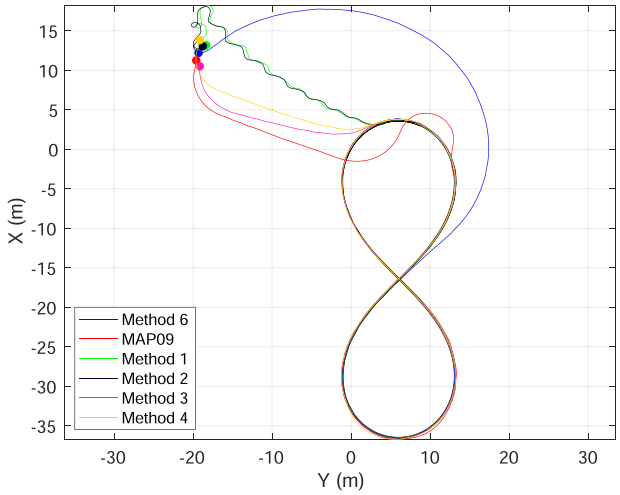

FIGURE 27 Vehicle paths using different path following methods to follow a Bernoulli lemniscate path in real trials. Filled circles are the positions of the vehicle at the beginning of each run. [Color figure can be viewed at wileyonlinelibrary.com]

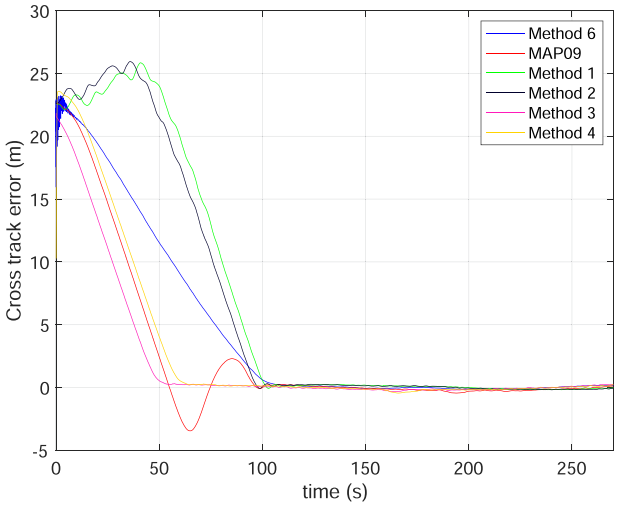

FIGURE 29 *Cross-track* errors for different methods while describing a Bernoulli lemniscate [Color figure can be viewed at wileyonlinelibrary.com]

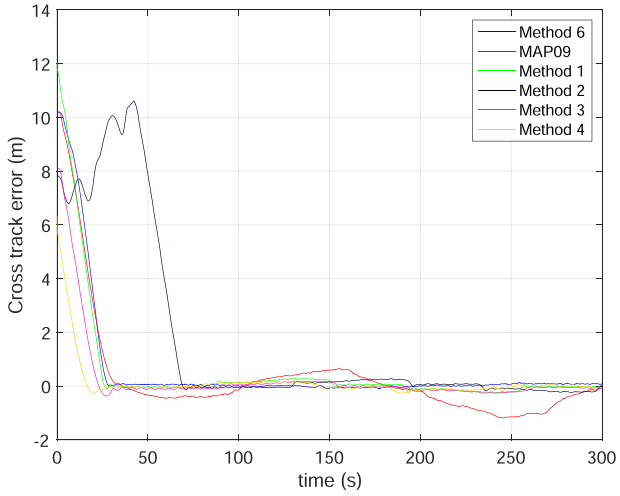

FIGURE 28 *Cross-track* errors for different methods while performing lawnmower maneuvers [Color figure can be viewed at wileyonlinelibrary.com]

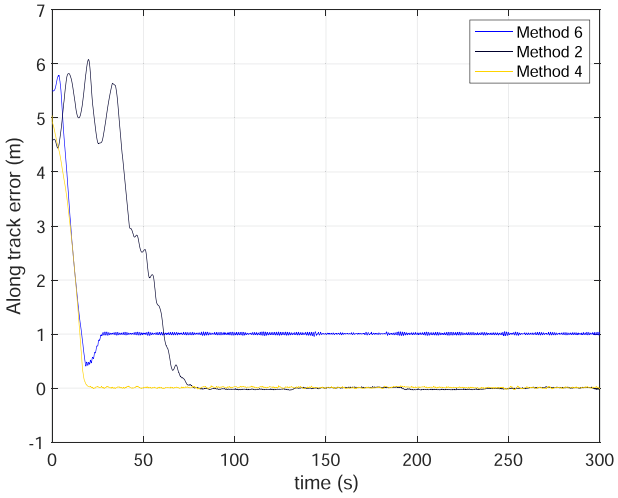

FIGURE 30 *Along-track* errors for different methods while performing lawn-mowing maneuvers [Color figure can be viewed at wileyonlinelibrary.com]

the different path following methods for a Bernoulli path using the Maltab toolbox described in Section 6.1, with a T440 Lenovo Computer.[4]

From an implementation standpoint, Methods 2,4, and 6 are the simplest and can therefore be considered as good candidates to solve path following problems. These methods are elegant in the sense that the "reference point" on the path that the vehicle must track to achieve path following is not necessary the closest to the vehicle. Instead, it can be initialized anywhere on the path, after which its evolution is controlled via $\dot{\gamma}$ or $\ddot{\gamma}$ to attract the vehicle to the path and make it follow that path with a desired speed. Table 3 indicates clearly that these methods require the least computation time. Among these methods, Method 4 is the simplest while Method 2 is the more complicated since it requires to know the knowledge of path's curvature. However, it can be expected that for time varying curvature paths, Method 2 will outperform Method 4.

Methods 1 and 3, on the other hand, are more complex in general, as they consider "reference point" the one closest to the vehicle which, in the case of general paths requires solving an optimization on-line to find this point. For this reason, these methods require more computation time than the Methods 2, 3, 6, as shown in Table 3. Comparing the two methods, it can be said that Method 1 is more complicated as it requires to know the function of curvature

[4]Core™ i7-7500U CPU @ 2.70GHz × 4, RAM:16Gb.

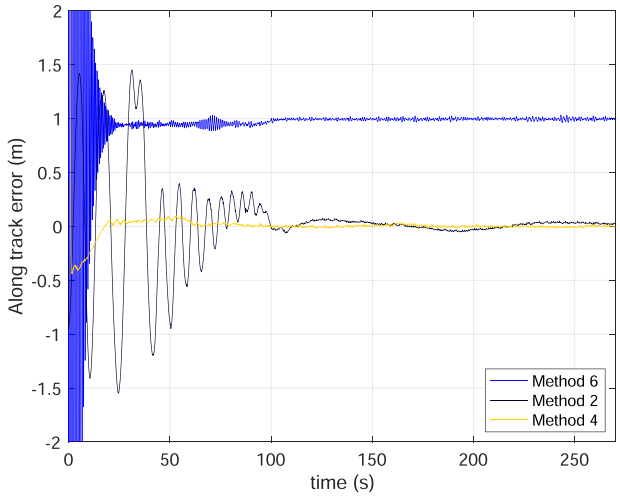

FIGURE 31 *Along-track* errors for different methods while describing a Bernoulli lemniscate [Color figure can be viewed at wileyonlinelibrary.com]

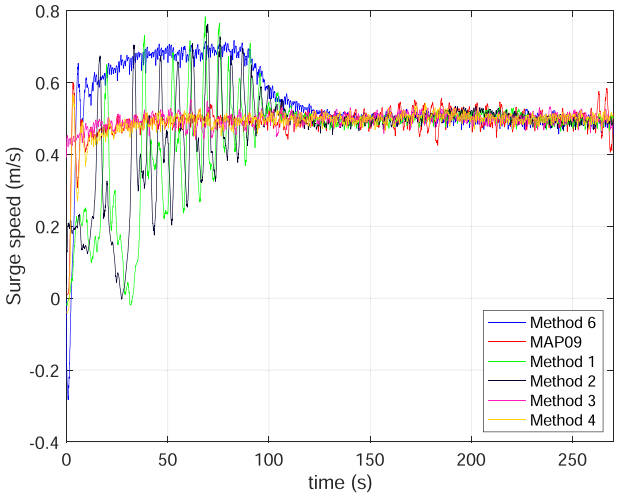

FIGURE 33 Surge speed for different methods while describing a Bernoulli lemniscate [Color figure can be viewed at wileyonlinelibrary.com]

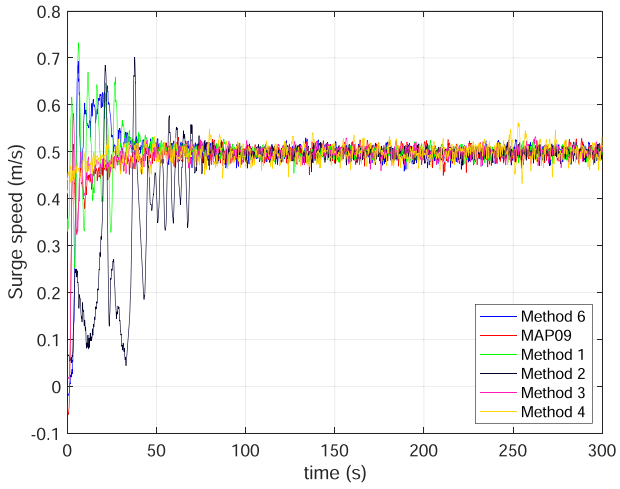

FIGURE 32 Surge speed for different methods while performing lawnmower maneuvers [Color figure can be viewed at wileyonlinelibrary.com]

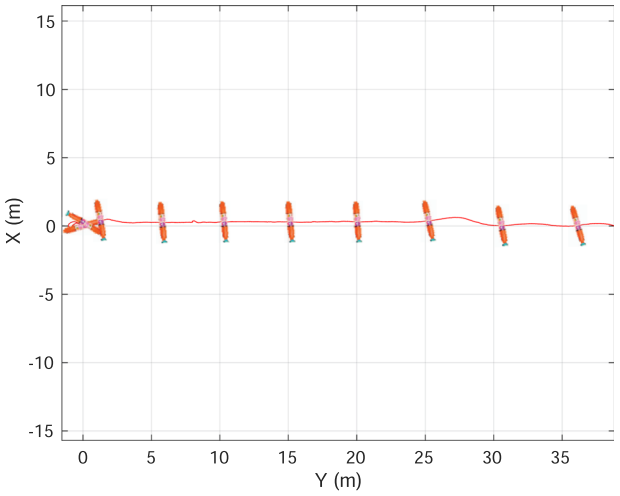

FIGURE 34 Vehicle path using the method described in Section 5. The initial positions of the vehicle is (0,0). [Color figure can be viewed at wileyonlinelibrary.com]

along the path; however, one might expect that it outperforms Method 3 for paths with varying curvature.

VF methods are more flexible than those discussed above in the sense that they can be extended to deal with the problem of obstacle avoidance (Wilhelm & Clem, 2019). However, they are more complex in terms of derivation and proof of stability.

Methods 5 and 7 (NMPC-based approaches) are the most complicated, in terms of design and implementation, since they involve solving an optimization problem online that is nonconvex, due to the kinematic constraints of the vehicles. However, these methods deal with the vehicle's input constraints (e.g., velocity and angular rate) explicitly, an important feature that none of the methods mentioned before has. For this reason, NMPC based-approaches are expected to outperform the others in path following missions that require the vehicle to maneuver more aggressively, pushing the actuation values close to their limits. Another advantage of NMPC-based methods is that they can incorporate easily other tasks such as obstacle avoidance in the path following problem (Shibata et al., 2018).

8.2 | Other issues

As explained before, the main focus of this paper is on path following methods that can be designed by taking into consideration the vehicle kinematics only. It is tacitly assumed that inner control loops

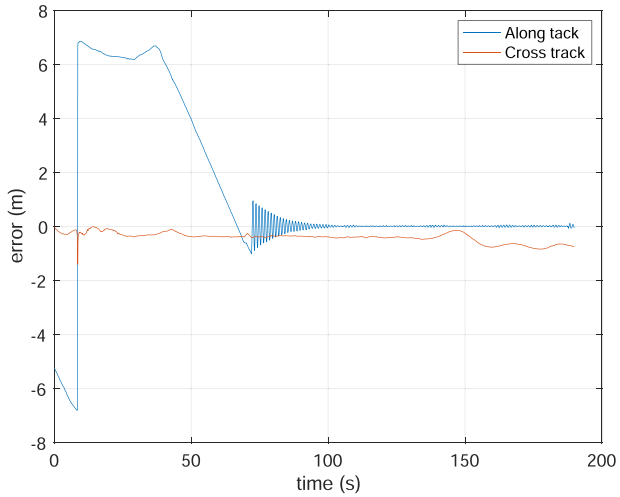

FIGURE 35 Along and cross track errors during the mission of Figure 34 [Color figure can be viewed at wileyonlinelibrary.com]

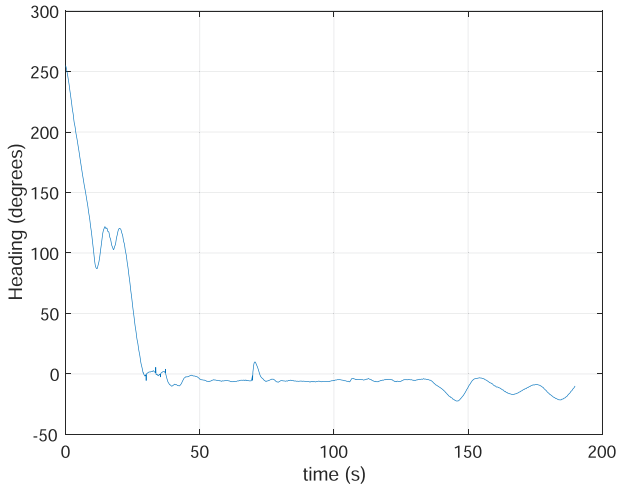

FIGURE 36 Yaw during the mission of Figure 34 [Color figure can be viewed at wileyonlinelibrary.com]

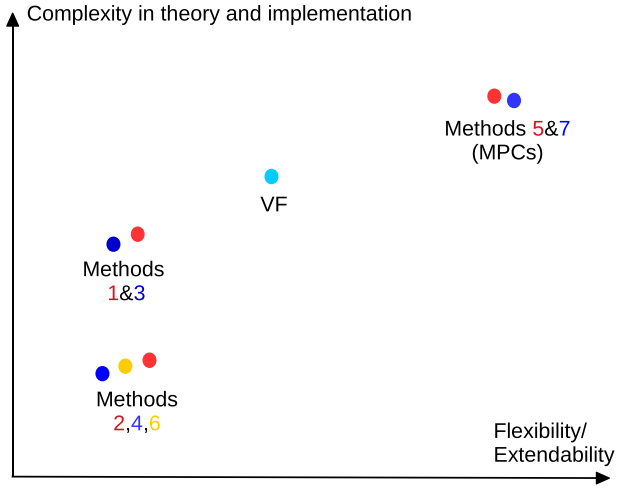

FIGURE 37 A comparison among the path following methods [Color figure can be viewed at wileyonlinelibrary.com]

can be implemented to make the vehicle's relevant variables track with good accuracy the commands issued by the outer path following control loop (e.g., speed, heading, heading rate, etc.). This design approach is widely adopted by control practitioners because it simplifies considerably the design, analysis, and implementation steps. Experimental results reported in the literature (see, e.g., Bibuli et al., 2009; Rego et al., 2019; Thrun et al., 2007) and the results presented in Section 7, suggest that this should be the first candidate approach to solve path following control problems. From a theoretical standpoint, however, it is generally not trivial to guarantee that the path following methods developed at the kinematic level also preserve adequate performance with nonideal tracking inner-loop controllers. With Method 6, the conclusion is in the affirmative as reported in Maurya et al. (2009); Rego et al. (2019); Vanni (2007), where it is shown that the complete path following error system is ISS with respect to the tracking errors associated with speed and yaw-rate tracking controllers. This implies that as long as the tracking errors are bounded the path following error is bounded. Relate results can be found in Kaminer et al. (2017). A similar property may very well hold with the other path following methods summarized in this paper; however, proofs for these methods are not available in the literature.

Another issue with the inner-outer loop approach is that it requires the inner-loop controllers to be fine tuned to guarantee that the complete path following system is stable and and achieves a desired degree of performance; the tuning process can be time-consuming and in many cases it is also expensive. To deal with this problem, the work in Kaminer et al. (2010) suggests augmenting the inner-loop controllers with "Ł1 adaptive controllers," that aim at eliminating the effect of inner-loop uncertainty on the total path following system. See Maalouf et al. (2015) and Xu et al. (2021) for applications of this strategy to an underwater robot and a surface ship, respectively.

Another type of path following methods involves including explicitly the vehicle dynamics in the design process. A typical example of this approach can be found in Aguiar and Hespanha (2007); Lapierre et al. (2006) where the authors employ backstepping techniques to derive control laws for the vehicle force and toque to achieve path following. The main challenge of this approach is that controller design is substantially harder, since the vehicle model adopted in the design phase is far more complex than the kinematic model. In addition, the dynamics of the vehicle normally exhibit large uncertainty in the parameters (e.g., mass, momentums of inertia, hydrodynamic or aerodynamic drag terms, etc.) thus making the design process quite challenging.

Finally, there is also considerable interest in using learning-based methods for the path following problem in recent years. These type of method aims at dealing with vehicle model and environment

TABLE 3 Computation times (ms) required by different path following methods at each sampling cycle (average over 2500 runs)

Method 1	Method 2	Method 3	Method 4	Method 5	Method 6	Method 7
0.365	0.040	0.385	0.053	84.202	0.071	82.662

uncertainty. Along this line, learning-based MPC methods suggested in Ostafew et al. (2016), Rokonuzzaman et al. (2020) seem to be a promising approach while reinforcement learning-based methods described in Kamran et al. (2019), Martinsen and Lekkas (2018), Wang et al. (2021) are also worth considering. Although these methods are powerful, they are far complicated than the methods derived in Section 3. Furthermore these methods lack of stability guarantees and, in the case of RL-based methods, they may not lend themselves to an intuitive interpretation, thus rendering the task of controller tuning difficult.

9 | CONCLUSIONS

Path following of autonomous robotic vehicles is a fairly well understood and established technique for kinematic vehicle models, and a variety of solutions to the path following problem have been published in the literature. Motivated by the need to bring many of the methods under a common umbrella, this paper presented an in-depth review of the topic of path following for autonomous vehicle moving in 2D, showing clearly how many of the existing techniques can be derived under a unified mathematical framework that makes use of nonlinear control theory. The paper provided a rigorous analysis of the main classes of methodologies identified and discussed the advantages and disadvantages of each method, comparing them from a design and implementation standpoint. In addition, the paper described the steps involved in going from theory to practice through the introduction of Matlab and Gazebo/ROS simulation toolboxes that are helpful in testing path following methods before their integration in the combined guidance, navigation, and control systems of autonomous vehicles. Finally, the paper discussed the results of experimental field tests with underactuated and overactuated autonomous marine vehicles performing path following maneuvers.

The simulations and experimental results showed that as long as the inner-loop/dynamic) tracking controllers for speed, heading, or heading rate issued by the (outer loop/kinematic) path following controller are well tuned and there is adequate time scale separation of the inner and outer loops systems, the methods exposed in the paper for kinematic models hold very good potential for real life applications.

For missions involving vehicles for which the inner and outer loop dynamics do not exhibit a clear two scale separation and yet require good path following performance, methods that take explicitly into account the vehicle dynamics in the design of path following strategies may be required. Considerable research has been done in this area, but to the space limitations this survey did not address them. See Lapierre et al. (2006) for a representative early example that resorts to backstepping and vehicle parameter adaptation strategies. The example shows clearly how the explicit intertwining of kinematics and dynamics in path following control system design complicates the design process, yields controllers with increased complexity that may be difficult to implement, and requires knowledge of the vehicle's parameters (such as hydrodynamic or aerodynamic terms) and its actuators, which normally exhibit high degrees of uncertainty. A trending approach to deal with uncertainty in vehicle and actuator models is to use learning-based techniques such as learning-MPC or reinforcement learning for path following. However, existing methods lack formal stability guarantees and fail to capture physically-based intuition that is often crucial in the design and tuning of advanced control systems. Clearly, further research is required to ascertain if such methods can be further developed to obtain proven performance guarantees, a significant step required to make a control systems applicable in practice.

ACKNOWLEDGMENTS

This work was partially funded by Fundação para a Ciência e a Tecnologia (FCT) and FEDER funds through the projects UIDB/50009/2020 and LISBOA-01-0145-FEDER-031411, the H2020-FETPROACT-2020-2 RAMONES project—Radioactivity Monitoring in Ocean Ecosystems (Grant agreement ID: 101017808) and the H2020-MSCA-RISE-2018 ECOBOTICS.SEA project—Bio-inspired Technologies for a Sustainable Marine Ecosystem (Grant agreement ID: 824043).

CONFLICT OF INTEREST

The authors declare no conflict of interest.

DATA AVAILABILITY STATEMENT

Research data are not shared.

ORCID

Antonio Pascoal http://orcid.org/0000-0002-0657-6671

REFERENCES

Abreu, P., Morishita, H., Pascoal, A., Ribeiro, J. & Silva, H. (2016) Marine vehicles with streamers for geotechnical surveys: modeling, positioning, and control. *IFAC-PapersOnLine*, 49(23), 458–464.

Abreu, P.C., Botelho, J., Góis, P., Pascoal, A., Ribeiro, J., Ribeiro, M., Rufino, M., Sebastião, L. & Silva, H. (2016) The medusa class of autonomous marine vehicles and their role in EU projects. In: *OCEANS 2016*. New York: IEEE, pp. 1–10.

Aguiar, A. & Pascoal, A. (2002) Dynamic positioning and way-point tracking of underactuated AUVs in the presence of ocean currents. In: *Proceedings of the 41st IEEE Conference on Decision and Control, 2002*. Vol. 2. pp. 2105–2110.

Aguiar, A.P. & Hespanha, J.P. (2007) Trajectory-tracking and path-following of underactuated autonomous vehicles with parametric modeling uncertainty. *IEEE Transactions on Automatic Control*, 52(8), 1362–1379.

Akhtar, A., Waslander, S.L. & Nielsen, C. (2012) Path following for a quadrotor using dynamic extension and transverse feedback linearization. In: *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*. IEEE, pp. 3551–3556.

Alessandretti, A., Aguiar, A.P. & Jones, C.N. (2013) Trajectory-tracking and path-following controllers for constrained underactuated vehicles using model predictive control. In: *Control Conference (ECC), 2013 European*. IEEE, pp. 1371–1376.

Alves, J., Oliveira, P., Oliveira, R., Pascoal, A., Rufino, M., Sebastiao, L. & Silvestre, C. (2006) Vehicle and mission control of the delfim autonomous surface craft. In: *2006 14th Mediterranean Conference on Control and Automation*. pp. 1–6.

Amidi, O. & Thorpe, C.E. (1991) Integrated mobile robot control. In: *Mobile Robots V*. Vol. 1388. SPIE, pp. 504–523.

Andersson, J.A.E., Gillis, J., Horn, G., Rawlings, J.B. & Diehl, M. (2019) CasADi - A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 11(1), 1–36.

Batkovic, I., Zanon, M., Ali, M. & Falcone, P. (2019) Real-time constrained trajectory planning and vehicle control for proactive autonomous driving with road users. In: *2019 18th European Control Conference (ECC)*. pp. 256–262.

Bibuli, M., Bruzzone, G., Caccia, M. & Lapierre, L. (2009) Path-following algorithms and experiments for an unmanned surface vehicle. *Journal of Field Robotics*, 26(8), 669–688.

Breivik, M. & Fossen, T. I. (2005) Principles of guidance-based path following in 2D and 3D. In: *Proceedings of the 44th IEEE Conference on Decision and Control*. pp. 627–634.

Caharija, W., Pettersen, K.Y., Bibuli, M., Calado, P., Zereik, E., Braga, J., Gravdahl, J.T., Sørensen, A.J., Milovanović, M. & Bruzzone, G. (2016) Integral line-of-sight guidance and control of underactuated marine vehicles: theory, simulations, and experiments. *IEEE Transactions on Control Systems Technology*, 24(5), 1623–1642.

Caldeira Abreu, P., Bayat, M., Botelho, J., Góis, P., Gomes, J., Pascoal, A., Ribeiro, J., Ribeiro, M., Rufino, M., Sebastião, L. & Silva, H. (2015) Cooperative formation control in the scope of the ec morph project: theory and experiments. In: *OCEANS 2015 - Genova*. pp. 1–7.

Coulter, R.C. (1992) *Implementation of the pure pursuit path tracking algorithm*. Technical report, Carnegie-Mellon UNIV Pittsburgh PA Robotics INST.

Dagci, O., Ogras, U. & Ozguner, U. (2003) Path following controller design using sliding mode control theory. In: *Proceedings of the 2003 American Control Conference, 2003*. Vol. 1. pp. 903–908.

De Luca, A., Oriolo, G. & Samson, C. (1998) Feedback control of a nonholonomic car-like robot. *Robot motion planning and control*, 229, 171–253.

Encarnacao, P. & Pascoal, A. (2000) 3D path following for autonomous underwater vehicle. In: *Proceedings of the 39th IEEE Conference on Decision and Control (Cat. No.00CH37187)*. Vol. 3. pp. 2977–2982.

Fossen, T.I. (2011) *Handbook of marine craft hydrodynamics and motion control*. John Wiley & Sons.

Fossen, T.I., Breivik, M. & Skjetne, R. (2003) Line-of-sight path following of underactuated marine craft. *IFAC Proceedings Volumes*, 36(21), 211–216. 6th IFAC Conference on Manoeuvring and Control of Marine Craft (MCMC 2003), Girona, Spain, 17–19 September, 1997.

Fossen, T.I., Johansen, T.A. & Perez, T. (2009) A survey of control allocation methods for underwater vehicles. In: Inzartsev, A.V. (Ed.) *Underwater vehicles*, (pp. 110–128). Rijeka: Intechopen. Ch. 7.

Fossen, T.I., Pettersen, K.Y. & Galeazzi, R. (2015) Line-of-sight path following for dubins paths with adaptive sideslip compensation of drift forces. *IEEE Transactions on Control Systems Technology*, 23(2), 820–827.

Fox, D., Burgard, W. & Thrun, S. (1997) The dynamic window approach to collision avoidance. *IEEE Robotics Automation Magazine*, 4(1), 23–33.

Ghorbani, M. (2017) Some notes on trajectory tracking and path following control of marine vehicles. IST Technical Report.

Ghorbani, M. (2021) A wall-following controller design method for underactuated AUVs: an adaptive internal model approach. *IFAC-PapersOnLine*, 54(16), 314–319. 13th IFAC Conference on Control Applications in Marine Systems, Robotics, and Vehicles CAMS 2021.

Gray, A., Abbena, E. & Salamon, S. (2006) *Modern differential geometry of curves and surfaces with Mathematica®*, 3rd edition.

Guo, H., Cao, D., Chen, H., Sun, Z. & Hu, Y. (2019) Model predictive path following control for autonomous cars considering a measurable disturbance: implementation, testing, and verification. *Mechanical Systems and Signal Processing*, 118, 41–60.

Hanson, A.J. & Ma, H. (1995) *Parallel transport approach to curve framing*. Technical report.

Hassan, K. (2002) *Nonlinear systems*, 3rd edition. Hoboken, NJ: Prentice Hall.

Helmick, D.M., Cheng, Y., Clouse, D. S., Matthies, L.H. & Roumeliotis, S.I. (2004) Path following using visual odometry for a mars rover in high-slip environments. In: *2004 IEEE Aerospace Conference Proceedings (IEEE Cat. No. 04TH8720)*. Vol. 2. IEEE, pp. 772–789.

Houska, B., Ferreau, H. & Diehl, M. (2011) ACADO toolkit—an open source framework for automatic control and dynamic optimization. *Optimal Control Applications and Methods*, 32(3), 298–312.

Hung, N.T., Pascoal, A.M. & Johansen, T.A. (2020) Cooperative path following of constrained autonomous vehicles with model predictive control and event triggered communications. *International Journal of Robust and Nonlinear Control*, 30, 2644–2670.

Hung, N.T., Rego, F., Crasta, N. & Pascoal, A. (2018) Input-constrained path following for autonomous marine vehicles with a global region of attraction. In: *11th IFAC Conference on Control Applications in Marine Systems, Robotics, and Vehicles CAMS 2018*. Vol. 51. pp. 348–353.

Jacinto, M.F. (2021) Cooperative motion control of aerial and marine vehicles for environmental applications. Master's thesis, Instituto Superior Tecnico, Lisboa. https://fenix.tecnico.ulisboa.pt/cursos/meec/dissertacao/1972678479054971

Jadbabaie, A. & Hauser, J. (2005) On the stability of receding horizon control with a general terminal cost. *IEEE Transactions on Automatic Control*, 50(5), 674–678.

Johansen, T.A. & Fossen, T.I. (2013) Control allocation—a survey. *Automatica*, 49(5), 1087–1103.

Kaminer, I., Pascoal, A., Xargay, E., Hovakimyan, N., Cao, C. & Dobrokhodov, V. (2010) Path following for small unmanned aerial vehicles using l1 adaptive augmentation of commercial autopilots. *Journal of Guidance, Control, and Dynamics*, 33(2), 550–564.

Kaminer, I., Pascoal, A., Xargay, E., Hovakimyan, N., Cichella, V. & Dobrokhodov, V. (2017) *Time-critical cooperative control of autonomous air vehicles*. Butterworth-Heinemann.

Kaminer, I., Yakimenko, O., Dobrokhodov, V., Pascoal, A., Hovakimyan, N., Patel, V., Cao, C. & Young, A. (2012) Coordinated path following for time-critical missions of multiple UAVs via L1 adaptive output feedback controllers. AIAA.

Kamran, D., Zhu, J. & Lauer, M. (2019) Learning path tracking for real car-like mobile robots from simulation. In: 2019 European Conference on Mobile Robots (ECMR). pp. 1–6.

Kanjanawanishkul, K. & Zell, A. (2009) Path following for an omnidirectional mobile robot based on model predictive control. In: *2009*

IEEE International Conference on Robotics and Automation. pp. 3341–3346.

Lapierre, L., Soetanto, D. & Pascoal, A. (2003) Nonlinear path following with applications to the control of autonomous underwater vehicles. In: *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*. Vol. 2. IEEE, pp. 1256–1261.

Lapierre, L., Soetanto, D. & Pascoal, A. (2006) Nonsingular path following control of a unicycle in the presence of parametric modelling uncertainties. *International Journal of Robust and Nonlinear Control*, 16(10), 485–503.

Lekkas, A.M. (2013) *Advanced in marine robotics* (Ch. 5). Chisinau: Lambert Academic Publishing.

Liu, C., McAree, O. & Chen, W.-H. (2013) Path-following control for small fixed-wing unmanned aerial vehicles under wind disturbances. *International Journal of Robust and Nonlinear Control*, 23(15), 1682–1698.

Liu, L., Wang, D. & Peng, Z. (2016) Path following of marine surface vehicles with dynamical uncertainty and time-varying ocean disturbances. *Neurocomputing*, 173, 799–808.

Maalouf, D., Chemori, A. & Creuze, V. (2015) L1 adaptive depth and pitch control of an underwater vehicle with real-time experiments. *Ocean Engineering*, 98, 66–77.

Manhães, M.M.M., Scherer, S.A., Voss, M., Douat, L.R. & Rauschenbach, T. (2016) UUV simulator: A gazebo-based package for underwater intervention and multi-robot simulation. In: *OCEANS 2016 MTS/IEEE Monterey*. IEEE.

Martinsen, A.B. & Lekkas, A.M. (2018) Straight-path following for underactuated marine vessels using deep reinforcement learning. *IFAC-PapersOnLine*, 51(29), 329–334. 11th IFAC Conference on Control Applications in Marine Systems, Robotics, and Vehicles CAMS 2018.

Maurya, P., Aguiar, A.P. & Pascoal, A. (2009) Marine vehicle path following using inner-outer loop control. *IFAC Proceedings Volumes*, 42(18), 38–43. 8th IFAC Conference on Manoeuvring and Control of Marine Craft.

Mayne, D.Q., Rawlings, J.B., Rao, C.V. & Scokaert, P.O. (2000) Constrained model predictive control: stability and optimality. *Automatica*, 36(6), 789–814.

Micaelli, A. & Samson, C. (1993) Trajectory tracking for unicycle-type and two-steering-wheels mobile robots. INRIA.

Nad, D., Mandic, F. & Miskovic, N. (2020) Using autonomous underwater vehicles for diver tracking and navigation aiding. *Journal of Marine Science and Engineering*, 8(6). https://doi.org/10.3390/jmse8060413

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

Ostafew, C.J., Schoellig, A.P., Barfoot, T.D. & Collier, J. (2016) Learning-based nonlinear model predictive control to improve vision-based mobile robot path tracking. *Journal of Field Robotics*, 33(1), 133–152.

Papoulias, F.A. (1991) Bifurcation analysis of line of sight vehicle guidance using sliding modes. *International Journal of Bifurcation and Chaos*, 01(04), 849–865.

Park, S., Deyst, J. & How, J.P. (2007) Performance and lyapunov stability of a nonlinear path following guidance method. *Journal of Guidance, Control, and Dynamics*, 30(6), 1718–1728.

Pyo, Y., Cho, H., Jung, L. & Lim, D. (2017) *ROS Robot Programming (English)*. Lake Forest: ROBOTIS.

Raffo, G.V., Gomes, G.K., Normey-Rico, J. E., Kelber, C.R. & Becker, L.B. (2009) A predictive controller for autonomous vehicle path tracking. *IEEE Transactions on Intelligent Transportation Systems*, 10(1), 92–102.

Rego, F.C., Hung, N.T., Jones, C.N., Pascoal, A.M. & Aguiar, A.P. (2019) Cooperative path-following control with logic-based communications: Theory and practice. Navigation and Control of Autonomous Marine Vehicles, chapter 8. IET.

Ribeiro, M.F.S. (2013) Computer-based cooperative motion planning, programming, and control of autonomous robotic vehicles. Master's thesis, Instituto Superior Tecnico, Av. Rovisco Pais 1, 1049-001 Lisboa.

Rokonuzzaman, M., Mohajer, N., Nahavandi, S. & Mohamed, S. (2020) Learning-based model predictive control for path tracking control of autonomous vehicle. In: *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. pp. 2913–2918.

Rokonuzzaman, M., Mohajer, N., Nahavandi, S. & Mohamed, S. (2021) Review and performance evaluation of path tracking controllers of autonomous vehicles. *IET Intelligent Transport Systems*, 15(5), 646–670.

Rubí, B., Pérez, R. & Morcego, B. (2019) A survey of path following control strategies for UAVs focused on quadrotors. *Journal of Intelligent & Robotic Systems*, 98, 241–265. https://doi.org/10.1007/s10846-019-01085-z

Rucco, A., Aguiar, A.P., Fontes, F.A., Pereira, F.L. & de Sousa, J.B. (2015) A model predictive control-based architecture for cooperative path-following of multiple unmanned aerial vehicles. In: *Developments in Model-Based Optimization and Control*. Springer, pp. 141–160.

Rysdyk, R. (2003) UAV path following for constant line-of-sight.2nd AIAA Conference and Workshop & Exhibit. AIAA

Shibata, K., Shibata, N., Nonaka, K. & Sekiguchi, K. (2018) Model predictive obstacle avoidance control for vehicles with automatic velocity suppression using artificial potential field. *IFAC-PapersOnLine*, 51(20), 313–318. 6th IFAC Conference on Nonlinear Model Predictive Control NMPC 2018.

Snider, J.M. (2009) *Automatic steering methods for autonomous automobile path tracking*. Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RITR-09-08.

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

Thrun, S., Montemerlo, M., Dahlkamp, H., Stavens, D., Aron, A., Diebel, J., Fong, P., Gale, J., Halpenny, M., Hoffmann, G., Lau, K., Oakley, C., Palatucci, M., Pratt, V., Stang, P., Strohband, S., Dupont, C., Jendrossek, L.-E., Koelen, C., Markey, C., Rummel, C., van Niekerk, J., Jensen, E., Alessandrini, P., Bradski, G., Davies, B., Ettinger, S., Kaehler, A., Nefian, A. & Mahoney, P. (2007) *Stanley: The Robot That Won the DARPA Grand Challenge*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 1–43.

Vanni, F.V. (2007) Coordinated motion control of multiple autonomous underwater vehicles. Master's thesis, Instituto Superior Tecnico, Lisboa.

Wang, C., Du, J., Wang, J. & Ren, Y. (2021) AUV path following control using deep reinforcement learning under the influence of ocean currents. In: *2021 5th International Conference on Digital Signal Processing*. ICDSP 2021. New York: Association for Computing Machinery, pp. 225–231.

Wilhelm, J.P. & Clem, G. (2019) Vector field UAV guidance for path following and obstacle avoidance with minimal deviation. *Journal of Guidance, Control, and Dynamics*, 42(8), 1848–1856.

Xu, H., Oliveira, P. & Guedes Soares, C. (2021) L1 adaptive backstepping control for path-following of underactuated marine surface ships. *European Journal of Control*, 58, 357–372.

Yang, J., Liu, C., Coombes, M., Yan, Y. & Chen, W.-H. (2021) Optimal path following for small fixed-wing UAVs under wind disturbances. *IEEE Transactions on Control Systems Technology*, 29(3), 996–1008.

Yu, S., Li, X., Chen, H. & Allgöwer, F. (2015) Nonlinear model predictive control for path following problems. *International Journal of Robust and Nonlinear Control*, 25(8), 1168–1182.

Zhao, S., Wang, X., Chen, H. & Wang, Y. (2020) Cooperative path following control of fixed-wing unmanned aerial vehicles with collision avoidance. *Journal of Intelligent & Robotic Systems*, 100(3), 1569–1581.

How to cite this article: Hung N., Rego F., Quintas J., Cruz J., Jacinto M., Souto D. et al. (2022) A review of path following control strategies for autonomous robotic vehicles: Theory, simulations, and experiments. *Journal of Field Robotics*, 1–33. https://doi.org/10.1002/rob.22142

APPENDIX A

A.1 | Basic results

The following lemma was used in the paper.

Lemma A.1 (Rotation Matrix Differential Equation, Theorem 2.2, Fossen, 2011). *Let* $R_A^B \in SO(n)$[5] $(n = 2, 3)$ *be the rotation matrix from frame* $\{A\}$ *to frame* $\{B\}$. *Then,*

$$\dot{R}_A^B = R_A^B S\left(\boldsymbol{\omega}_{A/B}^A\right), \tag{A1}$$

where $S(\boldsymbol{\omega}_{A/B}^A)$ *is a Skew-symmetric matrix*[6] *and* $\boldsymbol{\omega}_{A/B}^A \in \mathbb{R}^n$ *is the angular velocity vector of* $\{A\}$ *respect to* $\{B\}$, *expressed in* $\{A\}$.

A.2 | Proofs

A.2.1 | Proof of Theorem 3.1

Consider the Lyapunov function candidate V_1 defined by

$$V_1 = \frac{1}{2}y_1^2 + \frac{1}{2k_2}\tilde{\psi}^2 \tag{A2}$$

Taking its time derivative yields

$$\dot{V}_1 = y_1\dot{y}_1 + \frac{1}{k_2}\tilde{\psi}\dot{\tilde{\psi}} \overset{(25),(27),(30)}{=} y_1 u \sin\delta - \frac{k_1}{k_2}\tilde{\psi}^2. \tag{A3}$$

Because of Condition 1, $\dot{V}_1 \le 0$ for all t. Furthermore, the above equation shows that $\dot{V}_1$ is uniform continuous and therefore, using Barbalat's Lemma (Hassan, 2002), we conclude that $\dot{V}_1(t)$ converges to zero as $t \to \infty$. In view of (A3) and Condition 1, this implies that $y_1(t), \tilde{\psi}(t) \to 0$ as $t \to \infty$.

A.2.2 | Proof of Theorem 3.2

The proof can be done by employing the Lyapunov function candidate given by

$$V_2 = \frac{1}{2}\mathbf{e}_{\mathcal{P}}^2 + \frac{1}{2k_2}\tilde{\psi}^2. \tag{A4}$$

Taking its time derivative yields

$$\dot{V}_2 = \mathbf{e}_{\mathcal{P}}^\top\dot{\mathbf{e}}_{\mathcal{P}} - \frac{1}{k_2}\tilde{\psi}\dot{\tilde{\psi}} \overset{(23),(27)}{=} \mathbf{e}_{\mathcal{P}}^\top\begin{bmatrix} u\cos(\psi_e) - u_{\mathcal{P}} \\ u\sin(\psi_e)\end{bmatrix} + \frac{1}{k_2}\tilde{\psi}(r - \kappa(\gamma)u_{\mathcal{P}} - \dot{\delta}). \tag{A5}$$

In the above, we have used the fact that for any Skew-symmetric matrix S, $\mathbf{x}^\top S\mathbf{x} = 0$ for all $\mathbf{x}$, thus $\mathbf{e}_{\mathcal{P}}^\top S(\boldsymbol{\omega}_{\mathcal{P}})\mathbf{e}_{\mathcal{P}} = 0$. Substituting r given by (30) and $u_{\mathcal{P}}$ given by (33) in $\dot{V}_2$, we obtain

$$\dot{V}_2 = -k_3 s_1^2 + y_1 u\sin\delta - \frac{k_1}{k_2}\tilde{\psi}^2. \tag{A6}$$

Due to Condition 1, we conclude that $\dot{V}_2 \le 0$ for all t. Furthermore, the above equation shows that $\dot{V}_2$ is uniform continuous; thus, invoking Barbalat's lemma (Hassan, 2002), $\dot{V}_2(t)$ converges to 0 as $t \to \infty$. In view of (A6) this implies that $s_1(t), y_1(t), \tilde{\psi}(t) \to 0$ as $t \to \infty$. Notice that because of Condition 1 once $y_1, \tilde{\psi} \to 0$, $\psi_e \to 0$ as well.

A.2.3 | Proof of Theorem 3.3

Consider the Lyapunov function candidate V_3, given by $V_3 = y_1^2/2$. Taking its time derivative yields

$$\dot{V}_3 = y_1\dot{y}_1 \overset{(25),(35)}{=} -u\frac{y_1^2}{\sqrt{y_1^2 + \Delta_h^2}} \overset{(29)}{=} -U_d\frac{y_1^2}{\sqrt{y_1^2 + \Delta_h^2}}$$

Since U_d (the desired speed profile for the vehicle to track) is always positive, $\dot{V}_3 \le 0$ for all y_1. It can be seen that $\dot{V}_3$ is uniform continuous, and therefore invoking Barbalat's lemma (Hassan, 2002) $\dot{V}_3(t)$ converges to 0 as $t \to \infty$. This implies that $y_1(t)$ converges to zero as $t \to \infty$.

A.2.4 | Proof of Theorem 3.4

First, substituting $u_{\mathcal{P}}$ given by (33) and ψ_e given by (35) in (23) yields the dynamics of the position error in the resulting closed-loop system as

$$\dot{\mathbf{e}}_{\mathcal{P}} = -S(\boldsymbol{\omega}_{\mathcal{P}})\mathbf{e}_{\mathcal{P}} - \begin{bmatrix} -k_3 s_1 \\ -uy_1/\sqrt{y_1^2 + \Delta_h^2}\end{bmatrix}. \tag{A7}$$

Consider the Lyapunov function candidate, given by $V_4(\mathbf{e}_{\mathcal{P}}) = \frac{1}{2}\mathbf{e}_{\mathcal{P}}^\top\mathbf{e}_{\mathcal{P}}$. Taking its time derivative yields

$$\dot{V}_4 = \mathbf{e}_{\mathcal{P}}^\top\dot{\mathbf{e}}_{\mathcal{P}} \overset{(A7)}{=} -k_3 s_1^2 - u\frac{y_1^2}{\sqrt{y_1^2 + \Delta_h^2}} \overset{(29)}{=} -k_3 s_1^2 - U_d\frac{y_1^2}{\sqrt{y_1^2 + \Delta_h^2}}. \tag{A8}$$

Note that we used the fact that $\mathbf{e}_{\mathcal{P}}^\top S(\boldsymbol{\omega}_{\mathcal{P}})\mathbf{e}_{\mathcal{P}} = 0$ for all $\mathbf{e}_{\mathcal{P}}$ because S is a Skew-symmetric matrix. Furthermore, because $u = U_d$, which is always positive, we conclude that $\dot{V}_4 < 0$ for $\mathbf{e}_{\mathcal{P}}$. Since u is in general a

[5] A special orthogonal group with dimension n, defined as $SO(n) = \{R \in \mathbb{R}^{n\times n} : RR^\top = R^\top R = I_n, \det R = 1\}$

[6] A square matrix S is called Skew-symmetric matrix iff $S^\top = -S$.

function of time then the closed-loop position error system (A7) is nonautonomous, therefore we conclude that the origin of $\mathbf{e}_{\mathcal{P}}$ is UGAS.

A.2.5 | Proof of Theorem 3.6

Consider a Lyapunov function candidate for the path following system (51), given by

$$V_6(\mathbf{x}) = \frac{1}{2}\|\mathbf{x}\|^2 = \frac{1}{2}\mathbf{e}_{\mathcal{B}}^{\top}\mathbf{e}_{\mathcal{B}} + \frac{1}{2}e_{\gamma}^2. \tag{A9}$$

Taking its time derivative yields

$$\begin{aligned}\dot{V}_6 &= \mathbf{e}_{\mathcal{B}}^{\top}\dot{\mathbf{e}}_{\mathcal{B}} + e_{\gamma}\dot{e}_{\gamma}\\ &\overset{(48),(50)}{=} \mathbf{e}_{\mathcal{B}}^{\top}\left(-S(\boldsymbol{\omega})\mathbf{e}_{\mathcal{B}} + \Delta\mathbf{u} - R_{\mathcal{I}}^{\mathcal{B}}(\psi)\mathbf{p}_{\mathrm{d}}'(\gamma)\dot{\gamma}\right) + e_{\gamma}(\ddot{\gamma} - \dot{v}_{\mathrm{d}}).\\ &\overset{(52),(49)}{=} -\mathbf{e}_{\mathcal{B}}^{\top}K_{\mathrm{p}}\mathbf{e}_{\mathcal{B}} - k_{\gamma}e_{\gamma}^2.\end{aligned} \tag{A10}$$

Let $K = \mathrm{diag}(K_{\mathrm{p}}, k_{\gamma})$, then $\dot{V}_6 = -\mathbf{x}^{\top}K\mathbf{x} \le -\lambda_{\min}(K)\|\mathbf{x}\|^2$ for all $\mathbf{x} \neq \mathbf{0}$. Thus, we conclude that the origin of $\mathbf{x}$ is GES.

A.2.6 | Proof of Lemma 4.4

Consider again the Lyapunov function candidate V_6 given by (A12). Substituting $\mathbf{u}$ in (74) and $\ddot{\gamma}$ in (52b) in (73) and (50) yields

$$\begin{aligned}\dot{V}_6 &= \mathbf{e}_{\mathcal{B}}^{\top}\dot{\mathbf{e}}_{\mathcal{B}} + e_{\gamma}\dot{e}_{\gamma}\\ &= -\mathbf{e}_{\mathcal{B}}^{\top}K_{\mathrm{p}}\mathbf{e}_{\mathcal{B}} - k_{\gamma}e_{\gamma}^2 - \mathbf{e}_{\mathcal{B}}^{\top}R_{\mathcal{I}}^{\mathcal{B}}(\psi)\mathbf{e}_c\\ &\le -\lambda_{\min}(K)\|\mathbf{x}\|^2 + \|\mathbf{x}\|\|\mathbf{e}_c\|\end{aligned} \tag{A11}$$

Thus, invoking Theorem 4.19 in Hassan (2002) we conclude that the path following error system is ISS respect to the input $\mathbf{e}_c$.

A.2.7 | Proof of Theorem 5.1

Consider a Lyapunov function candidate for the path following system (77), given by

$$V_F(\mathbf{x}) = \frac{1}{2}\|\mathbf{x}\|^2 = \frac{1}{2}\mathbf{e}_{\mathcal{B}}^{\top}\mathbf{e}_{\mathcal{B}} + \frac{1}{2}e_{\gamma}^2. \tag{A12}$$

Taking its time derivative yields

$$\begin{aligned}\dot{V}_F &= \mathbf{e}_{\mathcal{B}}^{\top}\dot{\mathbf{e}}_{\mathcal{B}} + e_{\gamma}\dot{e}_{\gamma}\\ &\overset{(48),(50)}{=} \mathbf{e}_{\mathcal{B}}^{\top}\left(-S(\boldsymbol{\omega})\mathbf{e}_{\mathcal{B}} + \mathbf{v} - R_{\mathcal{I}}^{\mathcal{B}}(\psi)\mathbf{p}_{\mathrm{d}}'(\gamma)\dot{\gamma}\right) + e_{\gamma}(\ddot{\gamma} - \dot{v}_{\mathrm{d}}).\\ &\overset{(78),(49)}{=} -\mathbf{e}_{\mathcal{B}}^{\top}K_{\mathrm{p}}\mathbf{e}_{\mathcal{B}} - k_{\gamma}e_{\gamma}^2.\end{aligned} \tag{A13}$$

Let $K = \mathrm{diag}(K_{\mathrm{p}}, k_{\gamma})$, then $\dot{V}_F = -\mathbf{x}^{\top}K\mathbf{x} \le -\lambda_{\min}(K)\|\mathbf{x}\|^2$ for all $\mathbf{x} \neq \mathbf{0}$. Thus, we conclude that the origin of $\mathbf{x}$ is GES.