Moving Path Following for Unmanned Aerial Vehicles with Applications to Single and Multiple Target Tracking Problems

Tiago Oliveira, A. Pedro Aguiar, *Member, IEEE,*
Pedro Encarnação, *Senior Member, IEEE*

***Abstract*—This paper introduces the Moving Path Following (MPF) problem where a vehicle is required to converge to and follow a desired geometric moving path, without a specific temporal specification, thus generalizing the classical path following that only applies to stationary paths. Possible tasks that can be formulated as a MPF problem include tracking terrain/air vehicles and gas clouds monitoring, where the velocity of the target vehicle or cloud specifies the motion of the desired path. We derive an error space for MPF for the general case of time varying paths in a two dimensional space, and subsequently an application is described for the problem of tracking single and multiple targets on the ground using an Unmanned Aerial Vehicle (UAV) flying at constant altitude. To this end, a Lyapunov based MPF control law and a path generation algorithm are proposed together with convergence and performance metric results. Real world flight tests results that took place in OTA air base, Portugal with the ANTEX-X02 UAV demonstrate the effectiveness of the proposed method.**

***Index Terms*—Aerial robotics, target tracking, path following.**

I. Introduction

TRAJECTORY tracking and path following are typical motion control tasks that autonomous robotic vehicles are required to execute. In trajectory tracking, the vehicle should follow a given trajectory with time constraints, while in path following, time constraints are used for the vehicle to track a desired speed assignment along a predefined path [1] or there may be no time constraints and the vehicle can thus, for example, move with constant speed and achieve smoother convergence to the path [2]. For wheeled mobile vehicles, the first works on this subject have been presented in a series of groundbreaking papers by Samson et al. (see for example [3] and the references therein). For path following, a classical approach consists of defining the error space using the Serret-Frenet frame concept [4],[5], associated to the path. The same circle of ideas led to the development of trajectory tracking and path following systems for marine vehicles [1],[6] and Unmanned Aerial Vehicles (UAVs) [7], [8], [9].

Both in classical trajectory tracking and path following the reference trajectory/path is fixed in space. However, there are applications where it is useful to follow a path that is attached to a reference frame that moves with respect to an inertial coordinate frame. A typical example, which will be considered in this paper, is the case of a UAV following a path attached to a vehicle to be tracked, moving with time varying linear and angular velocities with respect to an inertial frame. Note that this problem cannot be directly solved by simply recasting it in a classical path following framework because the target vehicle imposes time constraints and therefore the expected performance (regarding path following objectives) could be affected. Moreover, closed-loop system's stability and robustness would not be ensured. Motivated by this observation, in this paper we present a solution to a new motion control problem, the Moving Path Following (MPF) problem, where the vehicle is required to converge to and follow a desired geometric *moving* path without a specific absolute temporal specification. By further extending the ideas in [10], the MPF method here presented is applied to the general case of desired paths moving with respect to an inertial coordinate frame with time-varying linear and angular velocities, and with non-constant curvature. The MPF method generalizes the classical path following for stationary paths, thus providing a generic tool to follow moving (time-varying) paths that can be applied to several mission scenarios, like thermals soaring, gas clouds monitoring (where the time-varying thermal/gas cloud center coordinates may specify the motion of the desired path) or terrain/air vehicles tracking [11], [12]. Thus, by design, the MPF method retains all the desirable characteristics of the classical path following method, namely smooth convergence to the moving path and the possibility of doing so at constant speed with respect to an inertial coordinate frame.

In this paper, we derive an error space for the general kinematic MPF. Subsequently, an application to the problem of tracking single and multiple targets on the ground using a fixed-wing UAV is described. The standard approach with UAV control is to assume that the vehicle has an off-the-shelf inner loop controller that accepts references at kinematic level (angular rates and linear velocities) and generates the UAV control signals necessary to follow those references in the presence of model uncertainty and external disturbances, like wind [13], [14]. Outer loop control laws are thus derived using

T. Oliveira is with the Portuguese Air Force Research, Development and Innovation Center, and with the Portuguese Air Force Academy, Portugal. tmoliveira@academiafa.edu.pt

A. P. Aguiar is with the Research Center for Systems and Technologies (SYSTEC), Faculty of Engineering of the University of Porto (FEUP), Portugal. pedro.aguiar@fe.up.pt

P. Encarnação is with the UCP - Católica Lisbon School of Business & Economics, Lisbon, Portugal. pme@ucp.pt

This work was supported in part by the Portuguese Air Force Academy, by FEDER and Fundação para a Ciência e a Tecnologia (FCT) under project POCI-01-0145-FEDER-006933/SYSTEC, and by FCT under project UID/GES/00407/2013.

a kinematic model of the vehicle that provide the references to the inner control loop. The same approach is adopted here. A MPF control law is derived using Lyapunov methods, assuming that the UAV flies at constant altitude and airspeed (thus contributing to operational safety and preventing sudden thrust bursts necessary to keep up with the moving path). By explicitly taking into account UAV's physical constraints, this paper formally addresses the necessary conditions for the moving path's geometry and linear and angular velocities with respect to the inertial frame that must be verified in order to ensure that the MPF problem is well posed.

Two example scenarios regarding single and multiple target tracking applications are presented in this paper.

For single ground target tracking, different strategies have been proposed in the literature. Lee et al. [15] and Spry et al. [16] designed a controller that switches between two modes according to the relation between the UAV and the target velocities. Dobrokhodov et al. [13] proposed a vision based target tracking system using a guidance based algorithm. The control objective is to keep the aircraft within a certain range of the target and align the aircraft velocity vector with the perpendicular to the direction of the line that connects the UAV center of mass with the target (perpendicular to the line of sight vector, so that the UAV performs a loiter centered at the target). A Lyapunov guidance vector field strategy is proposed in [8]. A vector field with a stable limit cycle centered at the target position is determined. From it, a scaled Lyapunov guidance vector is computed and added to the target velocity, if known, to provide a heading command to the UAV. However, this may lead to an oscillating behaviour when the target speed or wind speed is close to the UAV speed [9], [17]. Chen et al. [17] proposed the use of a tangent-plus-Lyapunov vector field which includes a switching logic between tangent and Lyapunov vector fields to make convergence to the standoff circle faster than the method presented in [8]. More recently, Oh et al. [18] proposed a differential geometry approach (depending on the initial positions and velocity magnitude ratio between the UAV and the target) to compute a desired tangent direction to a standoff orbit circle around the target position.

MPF can be applied to single target tracking by attaching to the target a path to be followed. The proposed method, by design, eliminates the oscillating behaviour observed in other approaches when the target speed or wind speed are similar to the UAV speed [9], [17] (provided that the UAV's ground speed is higher than the target speed). If the target speed is close to the UAV speed, the control law behaves similarly to a controller that tracks a particular point on a path that moves jointly with the target (further details can be found in [10]). When the ground target moves slower than the UAV, the control law makes the UAV to loiter above the target. Additionally, in contrast with the methods proposed in [17], [18], [19], the same control law is used in all operating conditions, disregarding the relative initial position between the UAV and the target, and their relative speeds. Finally, the MPF method here presented is not limited to a standoff circle centered at the target position (unlike most of the proposed methods in the literature) and allows the use of any geometric path shape (satisfying UAV's physical constraints) attached to the desired target, which may be useful for applications like the ones presented in [20], [21]. For example, in [20] a ground convoy of vehicles is restricted to be stationary or moving in straight lines at constant speed while being protected by a group of UAVs following an optimal path (with respect to continuous convoy protection) above their position. For this scenario, the MPF algorithm would allow an optimal path to be attached to the convoy removing the restrictions on the convoy movements. In order to illustrate these features, in the single target tracking scenario here presented, a fixed wing UAV is required to follow a lemniscate path centered at the actual target position whose angular velocity is the same as the target, keeping the UAV altitude and airspeed constants.

For multiple targets tracking applications, a path generation method is proposed that combines with the MPF algorithm here presented. The multiple target tracking problem is typically decomposed into two phases: the first phase establishes the order in which the targets should be visited and subsequently, it uses a strategy of path generation (or target interception) to successively intercept the targets (by the pre determined order) until all the targets are visited. Solutions for stationary targets [22], [23] and moving targets [24], [25], [26] can be found in the literature. For most interception tasks, the motion of the dynamic target is usually unknown in advance. Thus, visual feedback [27] and line-of-sight methods [25], [28] have been proven as effective approaches. The target interception problem is a classic subject in the area of missile guidance, where three fundamental guidance strategies [29], [30] can be found: 1) Pure Pursuit Guidance (PP), where the interceptor aligns its velocity with the line that connects its position with that of the target. This strategy often results in a tail chase [29]; 2) Line of Sight Guidance (LOS), where the interceptor aligns itself with the line that connects a fixed reference point and the target. This strategy is borrowed from surface-to-air missiles control algorithms in which the target is often illuminated by a beam originated in a ground station [29]; 3) Constant Bearing (CB), where the interceptor aligns its velocity relative to the target with the line that connects its position to the target. This method seeks to reduce the line of sight rotation rate to zero, such that the target is perceived by the interceptor at a constant bearing, and thus the distance to the target is reduced apparently on a direct collision course [29]. We propose a path generation algorithm to solve the interception of multiple targets that together with the MPF guidance and control law behaves similarly to the guidance strategies 2) and 3) described above. In practice, the proposed path generation algorithm makes the vehicle to track a moving Dubins path (with respect to an inertial frame) composed by a fixed circular segment and a moving straight line with a fixed initial position and a moving end point, solidary to the current target position (or to its estimated interception position, assuming it will keep its heading and velocity constant). This paper demonstrates how the MPF approach (combined with the path generation algorithm) provides a general tool that encompasses distinct classical guidance strategies popular in the missile guidance community. A detailed discussion regarding the MPF performance using the proposed path generation

algorithm is also addressed in this paper.

Briefly, this paper's main contributions are: *i)* The formulation of the general MPF problem and derivation of the error space; *ii)* The design of Lyapunov-based nonlinear controllers for the particular case of the MPF for a fixed-wing UAV; *iii)* A new path-generation algorithm using the MPF approach for multiple (possible moving) targets that encompasses distinct classical guidance strategies; and *iv)* Validation and illustration of the proposed approach through field tests using the ANTEX-X02 UAV.

The paper is organized as follows. Section II describes the 2D moving path following error space, and then, Section III particularizes the error space for the problem of tracking a target on the ground by a UAV. Section IV describes how the MPF formulation previously presented can be applied to single and multiple targets tracking problems and, subsequently, Section V presents the flight test architecture used and their results. Finally, Section VI contains the main conclusions and future work. Path generation details for the multiple targets tracking scenario, formal convergence conditions and performance metrics can be found in Appendix. Part of the work regarding the derivation of the MPF error space was presented in preliminary form in [31].

II. Error Space for Moving Path Following

This section presents the MPF problem and formulates the resulting kinematic model written with respect to the Serret-Frenet frame (see definition e.g., [4],[5]) associated to a given reference planar path. Consider a local inertial frame $\{I\} = \{\overrightarrow{x}, \overrightarrow{y}\}$ with the $\overrightarrow{x}$ axis pointing North and $\overrightarrow{y}$ East. Consider also a path-transport frame $\{P\} = \{\overrightarrow{x}_P, \overrightarrow{y}_P\}$ and let the origin of $\{P\}$ expressed in $\{I\}$ be denoted by p_0. Let ${}^P p_d(\ell) = \begin{bmatrix} {}^P p_{d_x}(\ell) & {}^P p_{d_y}(\ell) \end{bmatrix}^T$ be a desired planar geometric path parametrized by ℓ, which for convenience will be assumed to be the path length. Note that for a fixed $\ell \geq 0$, ${}^P p_d(\ell)$ is a point on the path expressed in the path-transport frame. Additionally, let $v_d = \dot{p}_0 = \begin{bmatrix} v_{d_x} & v_{d_y} \end{bmatrix}^T$ and ω_d be the corresponding linear and angular velocities of the path-transport frame $\{P\}$, respectively, expressed in $\{I\}$. The angular velocity ω_d is positive if the $\{P\}$ frame rotates clockwise around the $\overrightarrow{x}_P \times \overrightarrow{y}_P$ direction when observed from the origin of $\{I\}$. In practice, the path-transport frame specifies the desired motion of the path with respect to the inertial frame and a desired geometric path is considered to be a moving path whenever v_d or ω_d are different from zero. Furthermore, we consider the general case where v_d and ω_d can be time varying, i.e., $\dot{\omega}_d \neq 0$ and $\dot{v}_d \neq 0$.

The MPF problem can thus be formulated as follows: Given a robotic vehicle moving at a given speed V and a desired moving path $\mathcal{P}_d = \left({}^P p_d(\ell), p_0, v_d, \omega_d\right)$, design a control law that steers and keeps the vehicle on the desired path $\mathcal{P}_d$.

Let $\{F\} = \{\overrightarrow{t}, \overrightarrow{n}\}$ be the Serret-Frenet frame associated to the desired path and $\kappa(\ell)$ the path curvature. The $\{I\}$, $\{P\}$ and $\{F\}$ frames are depicted in Figure 1. Additionally, a wind frame $\{W\} = \{\overrightarrow{x}_W, \overrightarrow{y}_W\}$ is considered, located at the vehicle's center of mass and with its $\overrightarrow{x}_W$-axis along the direction of the vehicle velocity vector, the $\overrightarrow{y}_W$-axis parallel

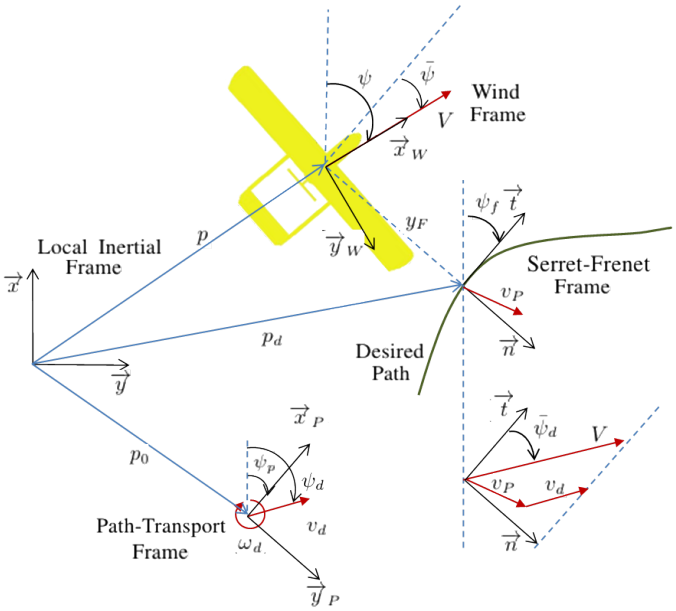


Fig. 1. Moving path following: Error space frames and relevant variables, illustrating the case of a UAV.

to the $\overrightarrow{x} - \overrightarrow{y}$ plane, normal to $\overrightarrow{x}_W$, and pointing to the right of an observer that moves in the same direction of the aircraft (see Figure 1). From this definition, ${}^W v_W$, the linear velocity of $\{W\}$ relative to $\{I\}$ and expressed in $\{W\}$, is given by ${}^W v_W = \begin{bmatrix} V & 0 \end{bmatrix}^T$.

The vehicle center of mass coordinates are denoted by $p = \begin{bmatrix} x & y \end{bmatrix}^T$ when expressed in the inertial frame $\{I\}$ and by $p_F = \begin{bmatrix} x_F & y_F \end{bmatrix}^T$ when expressed in the Serret-Frenet frame. A given path point ${}^P p_d(\ell)$, parametrized by ℓ, can be expressed in the $\{I\}$ frame through

$$ {}^I p_d(\ell) = p_0 + {}^I R_P \, {}^P p_d(\ell), $$

where ${}^I R_P$ is the rotation matrix from $\{P\}$ to $\{I\}$. For convenience, a given path point ${}^I p_d(\ell)$ that is the closest to the vehicle is denoted by p_d. Setting the origin of $\{F\}$ at the path point p_d, it follows that $p_F = \begin{bmatrix} 0 & y_F \end{bmatrix}^T$. From the previous definition, the velocity of a given path point parametrized by ℓ relative to $\{I\}$ and expressed in $\{I\}$ (that will be needed in the sequel) is given by

$$ {}^I \dot{p}_d(\ell) = \dot{p}_0 + {}^I R_P \, {}^P \dot{p}_d(\ell) + S\left(\omega_d\right) \, {}^I R_P \, {}^P p_d(\ell), $$

where $S(.)$ is a skew-symmetric matrix that satisfies

$$ S(a) = \begin{bmatrix} 0 & -a \\ a & 0 \end{bmatrix}. $$

Since by definition,

$$ {}^P \dot{p}_d(\ell) = 0 $$
$$ {}^I R_P \, {}^P p_d(\ell) = {}^I p_d(\ell) - p_0, $$

one obtains

$$ {}^I \dot{p}_d(\ell) = v_d + S\left(\omega_d\right) \left({}^I p_d(\ell) - p_0\right). \tag{1} $$

From the classical path following algorithms [3], the angular velocity of the $\{F\}$ frame with respect to the inertial frame, written in the $\{F\}$ frame, is given by

$$^{F}\omega_{F} = \kappa(\ell)\,\dot{\ell}.$$

Additionally, admitting that the planar path is also rotating with an angular velocity given by ω_d, it is straightforward to verify that

$$^{F}\omega_{F} = \kappa(\ell)\,\dot{\ell} + \omega_d.$$

The position of the UAV in the $\{I\}$ frame can be written as (see Figure 1)

$$p = p_d + {}^{I}R_{F}\,p_F \tag{2}$$

where ${}^{I}R_{F}$ is the rotation matrix from $\{F\}$ to $\{I\}$. Differentiating (2) with respect to time yields

$$\dot{p} = \dot{p}_d + {}^{I}R_{F}\,\dot{p}_F + {}^{I}R_{F}\,S\left({}^{F}\omega_{F}\right)\,p_F.$$

Pre-multiplying by ${}^{F}R_{I}$ one obtains

$${}^{F}R_{I}\dot{p} = {}^{F}R_{I}\,\dot{p}_d + \dot{p}_F + S\left({}^{F}\omega_{F}\right)\,p_F. \tag{3}$$

The linear velocity ${}^{F}R_{I}\,\dot{p}_d$ of a point on the path relative to $\{I\}$ and expressed in $\{F\}$ is the sum of the linear velocity of the point relative to $\{F\}$ given by ${}^{F}v_{F} = \begin{bmatrix} \dot{\ell} & 0 \end{bmatrix}^{T}$, with the velocity of the Serret-Frenet frame relative to $\{I\}$ (see equation (1)), both expressed in $\{F\}$, i.e.,

$${}^{F}R_{I}\,\dot{p}_d = {}^{F}v_{F} + {}^{F}R_{I}\left(v_d + \underbrace{S\left(\omega_d\right)\,\left(p_d - p_0\right)}_{v_P}\right), \tag{4}$$

where $(p_d - p_0) = \begin{bmatrix} \Delta x & \Delta y \end{bmatrix}^{T}$ is the vector from the origin of $\{P\}$ to the origin of the $\{F\}$ frame on the path. The path may rotate around p_0, thus v_P is the linear velocity of p_d due to the path's angular velocity ω_d. The left side of (3) can be rewritten as

$${}^{F}R_{I}\dot{p} = {}^{F}R_{W}\,{}^{W}v_{W}. \tag{5}$$

Therefore, combining (4) with (5), equation (3) gives

$$\begin{aligned}\dot{p}_F \;=\; & {}^{F}R_{W}\,{}^{W}v_{W} - S\left({}^{F}\omega_{F}\right)\,p_F - {}^{F}v_{F} \\ & - {}^{F}R_{I}\,\left(v_d + S\left(\omega_d\right)\,\left(p_d - p_0\right)\right).\end{aligned} \tag{6}$$

Since the moving path's rotation axis is by definition perpendicular to the $\overrightarrow{x} - \overrightarrow{y}$ plane, the angular velocity of the $\{F\}$ frame with respect to the inertial frame, expressed in the $\{W\}$ frame (that will be necessary in the sequel) is

$${}^{W}\omega_{F} = {}^{F}\omega_{F}.$$

The relative angular velocity between the $\{F\}$ frame and the wind frame $\{W\}$, expressed in $\{W\}$, is given by

$${}^{W}\omega^{r}{}_{W,F} = {}^{W}\omega_{W} - {}^{W}\omega_{F} \tag{7}$$

and thus,

$${}^{F}\dot{R}_{W} \;=\; {}^{F}R_{W}\,S\left({}^{W}\omega^{r}{}_{W,F}\right). \tag{8}$$

In summary, the complete MPF kinematic error dynamics is given by equations (6) and (8). Note that for fixed paths, that is with $v_d = 0$, $\omega_d = 0$, (6) and (8) resume to the classical path following expressions, which can be found in the works [2], [3], [6].

III. Moving Path following control law: A generic tool to follow time-varying paths

This section starts by particularizing the error space defined in Section II to the case where the angles ψ_p, ψ_f and $\bar{\psi}$ (see Figure 1) are used to parametrize the rotation matrices between reference frames. Then, an application is made to ground target tracking by a UAV. The control law is derived using Lyapunov methods, assuming that the UAV flies at constant altitude and airspeed.

A. MPF error kinematics

Starting with the path-following controller, the goal is to drive the lateral distance y_F to zero and orient the UAV such that the projection of its velocity vector onto the normal $\overrightarrow{n}$ to the path equals the normal component of the velocity of the Serret-Frenet frame origin given by the vector sum $v_d + v_p$ (see Figure 1 and equation (4)). Thus, the relative motion between the vehicle and the desired moving path occurs along the tangent $\overrightarrow{t}$ to the path. Note that by imposing this goal to the kinematic path-following, the classical situation of following paths that are fixed in space [32], [6], [7] is included. Let ψ_p be the angle that parametrizes the rotation matrix from $\{I\}$ to $\{P\}$ (thus, by definition $\omega_d = \dot{\psi}_p$) and let ψ be the angle between the vehicle velocity vector and the North direction. Additionally, let ψ_f be the yaw angle that parametrizes the rotation matrix from $\{I\}$ to $\{F\}$. The angular displacement between the wind frame and the Serret-Frenet frame is $\bar{\psi} = \psi - \psi_f$. Figure 1 shows the error space for path following. Taking into account the last notation, the UAV kinematic equations expressed in $\{I\}$ are given by

$$\begin{aligned}\dot{x} &= V\,\cos\psi \\ \dot{y} &= V\,\sin\psi. \\ \dot{\psi} &= \omega \qquad \text{with } \omega \in [-\omega_{max}, \omega_{max}]\end{aligned} \tag{9}$$

where ω_{max} represents the bound on the yaw rate, and $r_{min} = \frac{V}{\omega_{max}}$ is the minimum turning radius of the vehicle. Similarly, the movement of the origin of the $\{P\}$ frame is described by the following kinematic equations in terms of the total speed $||v_d||$ and the yaw angle ψ_d

$$\begin{aligned}v_{d_x} &= ||v_d||\,\cos\psi_d \\ v_{d_y} &= ||v_d||\,\sin\psi_d.\end{aligned} \tag{10}$$

Therefore, equation (6) can be rewritten as

$$\begin{aligned}\begin{bmatrix} 0 \\ \dot{y}_F \end{bmatrix} = & \begin{bmatrix} V\,\cos\bar{\psi} \\ V\,\sin\bar{\psi} \end{bmatrix} - \begin{bmatrix} \dot{\ell} \\ 0 \end{bmatrix} \\ & - {}^{F}R_{I}\left(\psi_f\right)\left(\begin{bmatrix} v_{d_x} \\ v_{d_y} \end{bmatrix} + \begin{bmatrix} 0 & -\omega_d \\ \omega_d & 0 \end{bmatrix}\begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}\right) \\ & - \begin{bmatrix} -\dot{\ell}\left(\kappa(\ell)\,y_F\right) - \omega_d\,y_F \\ 0 \end{bmatrix}.\end{aligned} \tag{11}$$

Considering the case where the UAV will be flying at constant altitude, the angular rate $\dot{\psi}$ is related to the angular velocity of the wind frame with respect to the inertial frame, expressed in the wind frame through ${}^{W}\omega_{W} = \dot{\psi}$.

For planar moving paths, it is straightforward to verify that the relative angular velocity between $\{W\}$ and $\{F\}$ is

related to the relative yaw angle rate $\dot{\bar{\psi}}$, and therefore one can conclude that (see also equation (7))

$$\dot{\bar{\psi}} = \dot{\psi} - \kappa(\ell)\,\dot{\ell} - \omega_d. \tag{12}$$

The previous result together with (11) gives the error kinematic model for MPF

$$\begin{aligned} \dot{\ell} &= \frac{V\,\cos\bar{\psi} - (v_{d_x} - \omega_d\,\Delta y)\,\cos\psi_f}{1-\kappa(\ell)\,y_F} \\ &\quad - \frac{\left(v_{d_y} + \omega_d\,\Delta x\right)\,\sin\psi_f - \omega_d\,y_F}{1-\kappa(\ell)\,y_F} \\ \dot{y}_F &= V\,\sin\bar{\psi} + (v_{d_x} - \omega_d\,\Delta y)\,\sin\psi_f - \left(v_{d_y} + \omega_d\,\Delta x\right)\,\cos\psi_f \\ \dot{\bar{\psi}} &= \dot{\psi} - \kappa(\ell)\,\dot{\ell} - \omega_d, \end{aligned} \tag{13}$$

where $\dot{\psi}$ is the control variable for the kinematic controller. It is assumed that $1-\kappa(\ell)\,y_F \neq 0$, which corresponds to the vehicle not being exactly at the distance from the path point p_d (the closest path point to the vehicle - parametrized by ℓ) that corresponds to the inverse of the path's curvature at that point. For a method to avoid this singularity, please refer to [1], [33].

The error space for MPF given by equation (13) will be used in the sequel to derive the generic MPF control law for planar paths.

B. MPF control law for planar paths

In this section, a generic MPF controller is derived and its stability is proven. A simulation example is then presented to illustrate the MPF method.

Figure 1 illustrates a MPF application example where a UAV should follow a planar (horizontal) path moving with linear velocity v_d and angular velocity ω_d with respect to an inertial frame, keeping its altitude constant. Considering the kinematic model (13), the steady state value $\bar{\psi}_d$ for ψ can be computed by setting $\dot{y}_F = 0$, which yields

$$\bar{\psi}_d = \arcsin\left(\frac{-(v_{d_x} - \omega_d\,\Delta y)\,\sin\psi_f + \left(v_{d_y} + \omega_d\,\Delta x\right)\,\cos\psi_f}{V}\right) \tag{14}$$

Note that the numerator of the arcsin argument is the sum of the path-transport frame speed $\|v_d\|$ with the linear speed $\|v_P\|$ of the origin of $\{F\}$ along the normal to the path. In order to ensure that equation (14) is always well defined, one may have to introduce some restrictions on the chosen path's geometry or *dynamics* since it depends on the relation between the path's linear and angular velocities and also on the distance between the origin of the path-transport frame and the Serret-Frenet frame, given by $\sqrt{\Delta x^2 + \Delta y^2}$. More specifically, from (14), it must be ensured that

$$|\omega_d|\,\sqrt{\Delta x^2 + \Delta y^2} < \frac{V - v_d\,\sin(\psi_d - \psi_f)}{\left|\sin\left(\psi_f + \arctan\left(\frac{\Delta y}{\Delta x}\right)\right)\right|} \quad \text{and}$$

$$v_d < V. \tag{15}$$

Notice, for instance, that in the particular case where the path's angular velocity is equal to zero ($\omega_d = 0$ and $v_d \neq 0$) it is *enough* to ensure that $V > v_d$. Conversely, for the case when the path only rotates around the origin of the path-transport frame ($\omega_d \neq 0$ and $v_d = 0$), one must ensure that the distance between any path point $\mathcal{P}_d$ and its center of rotation, i.e. $\sqrt{\Delta x^2 + \Delta y^2}$, will always be smaller than $\frac{V}{|\omega_d|}$. These particular cases provide an intuitive interpretation for the conditions in (15). Additionally, a MPF control law should drive the lateral distance y_F and heading error $\tilde{\bar{\psi}} = \bar{\psi} - \bar{\psi}_d$ to zero. Thus, considering equation (12), the steady state value for the commanded yaw rate $\dot{\psi}$ (obtained by setting $\dot{\tilde{\bar{\psi}}} = 0$) will be

$$\dot{\psi} = \dot{\bar{\psi}}_d + \kappa(\ell)\,\dot{\ell} + \omega_d. \tag{16}$$

In order to take into account the vehicle kinematic constraints $|\omega| < \omega_{max}$ (see equation (9)), it is now straightforward to conclude that condition

$$|\dot{\bar{\psi}}_d + \kappa(\ell)\,\dot{\ell} + \omega_d| \leq \omega_{max} \tag{17}$$

must also be ensured for the MPF problem to be always well posed. This encompasses the classical path following constraint where condition

$$|\kappa(\ell)| \leq \frac{\omega_{max}}{V}$$

must be verified for the path following problem to be well posed. An illustration example is presented and discussed at the end of this section (see Figure 3) in order to provide a physical intuition for these results.

In order to avoid situations in which the UAV is required to fly near its stall speed, it is desirable to keep the vehicle airspeed (denoted by $\|V_0\|$) constant.[1] Denoting the velocity of the wind relative to $\{I\}$ and expressed in $\{I\}$ by ${}^I v_{wind}$, the wind total speed by $W_t = \|{}^I v_{wind}\|$ and the wind direction by χ_w (both assumed to be constant throughout the flight), one can relate the airspeed vector V_0 and the wind velocity

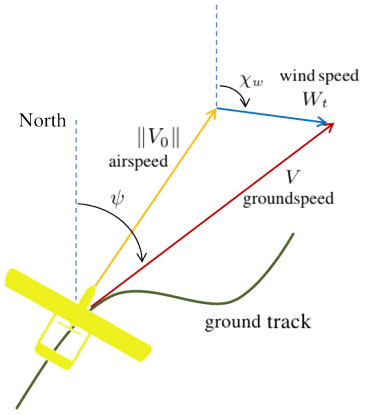


Fig. 2. Relationship between the groundspeed vector, the airspeed vector and the wind vector. Adapted from [34].

[1]Commercial autopilots usually accept airspeed references, expressed in the vehicle body frame, assuming small angles of attack and small sideslip angles with respect to the relative wind.

vector ${}^{I}v_{wind}$ with the UAV's ground velocity (see Figure 2) through [34]

$$V_0 = {}^{I}R_W\left(\psi\right)\ {}^{W}v_W - {}^{I}v_{wind}. \tag{18}$$

Taking the squared norm of both sides of (18) and solving the resulting quadratic equation for the ground speed V one obtains

$$V = \sqrt{\|V_0\|^2 + W_t^2\ (\cos^2\left(\psi - \chi_w\right) - 1)} + W_t\ \cos\left(\psi - \chi_w\right).$$

Considering the previous computations, the derivative of $\bar{\psi}_d$ with respect to time, that will be necessary in the sequel, assuming that the autopilot is able to keep $\|V_0\|$ constant, is

$$\dot{\bar{\psi}}_d = \frac{\rho}{V\sqrt{1-\left(\frac{-\left(v_{d_x}-\omega_d\,\Delta y\right)\,\sin\psi_f+\left(v_{d_y}+\omega_d\,\Delta x\right)\,\cos\psi_f}{V}\right)^2}}$$
$$-\dot{\psi}\,\frac{\lambda}{V^2\sqrt{1-\left(\frac{-\left(v_{d_x}-\omega_d\,\Delta y\right)\,\sin\psi_f+\left(v_{d_y}+\omega_d\,\Delta x\right)\,\cos\psi_f}{V}\right)^2}} \tag{19}$$

where

$$\begin{aligned}\rho &= \left(-\dot{\psi}_f\ \left(v_{d_x}-\omega_d\,\Delta y\right)+\dot{\omega}_d\,\Delta x+\omega_d\hat{\Delta}x\right)\cos\psi_f\\ &+\left(-\dot{\psi}_f\ \left(v_{d_y}+\omega_d\,\Delta x\right)+\dot{\omega}_d\,\Delta y+\omega_d\,\hat{\Delta}y\right)\sin\psi_f\\ &+||v_d||\dot{\psi}_d\cos\left(\psi_d-\psi_f\right)+\dot{v}_d\sin\left(\psi_d-\psi_f\right)\\ \lambda &= \hat{V}\ \left(-\left(v_{d_x}-\omega_d\,\Delta y\right)\,\sin\psi_f+\left(v_{d_y}+\omega_d\,\Delta x\right)\,\cos\psi_f\right)\end{aligned}$$

with

$$\hat{\Delta}x = \dot{\ell}\,\cos\psi_f - \omega_d\,\Delta y$$
$$\hat{\Delta}y = \dot{\ell}\,\sin\psi_f + \omega_d\,\Delta x$$
$$\hat{V} = -W_t\sin\left(\psi-\chi_w\right)\left(\frac{W_t\cos\left(\psi-\chi_w\right)}{\sqrt{\|V_0\|^2+W_t^2\ (\cos^2\left(\psi-\chi_w\right)-1)}}+1\right).$$

Equation (19) can be cast in the compact form

$$\dot{\bar{\psi}}_d = P - \dot{\psi}\,\Lambda,$$

with

$$P = \frac{\rho}{V\sqrt{1-\left(\frac{-\left(v_{d_x}-\omega_d\,\Delta y\right)\,\sin\psi_f+\left(v_{d_y}+\omega_d\,\Delta x\right)\,\cos\psi_f}{V}\right)^2}}$$

and

$$\Lambda = \frac{\lambda}{V^2\sqrt{1-\left(\frac{-\left(v_{d_x}-\omega_d\,\Delta y\right)\,\sin\psi_f+\left(v_{d_y}+\omega_d\,\Delta x\right)\,\cos\psi_f}{V}\right)^2}},$$

where it can be shown that $\Lambda \neq -1$ under the conditions given by equation (15). It is now possible to derive a control law to drive the linear distance y_F and heading error $\tilde{\psi} = \bar{\psi} - \bar{\psi}_d$ to zero. To that end, consider the control law

$$\begin{aligned}\dot{\psi} = \Big(&-g_1\,\tilde{\psi}+\kappa\left(\ell\right)\,\dot{\ell}+\omega_d+P\\ &-g_2\,y_F(((v_{d_x}-\omega_d\,\Delta y)\,\sin\psi_f\\ &-\left(v_{d_y}+\omega_d\,\Delta x\right)\,\cos\psi_f)\,\frac{1-\cos\tilde{\psi}}{\tilde{\psi}}\\ &+V\,\cos\bar{\psi}_d\frac{\sin\tilde{\psi}}{\tilde{\psi}})\Big)/(1+\Lambda)\end{aligned} \tag{20}$$

and the Lyapunov function

$$V_1 = \frac{1}{2}\left(y_F^2+\frac{1}{g_2}\,\tilde{\psi}^2\right), \tag{21}$$

where g_1 and g_2 are positive scalars assigning relative weights between position and orientation errors. The following theorem holds.

Theorem 1

Consider the MPF problem described by the UAV model (9) in closed-loop with the control law (20), subject to (15) and (17). Then, the closed-loop error signals $\tilde{\psi}$ and y_F converge to zero as $t \to \infty$. Moreover, the origin of the closed-loop error kinematic model is an exponentially stable equilibrium point.

Proof

Convergence of the closed-loop errors to zero

Convergence of the errors to zero can be proved from standard Lyapunov arguments using the Lyapunov function (21) and the Barbalat lemma [35].

Differentiating V_1 with respect to time yields

$$\begin{aligned}\dot{V}_1 &= y_F\,\dot{y}_F+\frac{1}{g_2}\,\tilde{\psi}\,\dot{\tilde{\psi}}\\ &= V\,y_F\,\sin\tilde{\psi}\,\cos\bar{\psi}_d+V\,y_F\,\cos\tilde{\psi}\,\sin\bar{\psi}_d+\\ &\quad y_F\left(\left(v_{d_x}-\omega_d\,\Delta y\right)\,\sin\psi_f-\left(v_{d_y}+\omega_d\,\Delta x\right)\,\cos\psi_f\right)\\ &\quad +\frac{1}{g_2}\,\tilde{\psi}\,\left(\dot{\tilde{\psi}}-\dot{\bar{\psi}}_d\right).\end{aligned} \tag{22}$$

Since, by definition (cf. equation (14)),

$$\begin{aligned}&V\,\sin\tilde{\psi}_d+\left(v_{d_x}-\omega_d\,\Delta y\right)\,\sin\psi_f\\ &\quad-\left(v_{d_y}+\omega_d\,\Delta x\right)\,\cos\psi_f=0\end{aligned} \tag{23}$$

the previous expression is equivalent to

$$\dot{V}_1 = V\,y_F\,\sin\tilde{\psi}\,\cos\bar{\psi}_d+\tfrac{1}{g_2}\,\tilde{\psi}\,\left(\dot{\psi}\left(1+\Lambda\right)-\kappa\left(\ell\right)\,\dot{\ell}-\omega_d-P\right)+$$
$$y_F\left(\left(v_{d_x}-\omega_d\,\Delta y\right)\,\sin\psi_f-\left(v_{d_y}+\omega_d\,\Delta x\right)\,\cos\psi_f\right)\left(1-\cos\tilde{\psi}\right).$$

The control law (20) with $g_1,\ g_2 > 0$ makes

$$\dot{V}_1 = -\frac{g_1}{g_2}\tilde{\psi}^2 \leqslant 0. \tag{24}$$

Given the definition of V_1 and the fact that $\dot{V}_1 \leqslant 0$, the errors $\tilde{\psi}$ and y_F are bounded. Computing the second derivative of V_1 one can easily verify that the boundedness of the state variables implies that $\dot{V}_1$ is uniformly continuous. Hence, Barbalat's lemma [35] allows for the conclusion that $\dot{V}_1$ and consequently $\tilde{\psi}$ tend to zero as $t \to \infty$.

Rewritting (20) as

$$\begin{aligned}\dot{\tilde{\psi}} = &-g_1\,\tilde{\psi}-g_2\,y_F((v_{d_x}-\omega_d\,\Delta y)\,\sin\psi_f\\ &-\left(v_{d_y}+\omega_d\,\Delta x\right)\,\cos\psi_f)\,\tfrac{1-\cos\tilde{\psi}}{\tilde{\psi}}-g_2\,V\,y_F\,\cos\bar{\psi}_d\tfrac{\sin\tilde{\psi}}{\tilde{\psi}},\end{aligned} \tag{25}$$

differentiating $\dot{\tilde{\psi}}$ with respect to time, and invoking the boundedness of the variables involved, one can conclude that $\dot{\tilde{\psi}}$ is uniformly continuous. Applying once more Barbalat's lemma to conclude that $\dot{\tilde{\psi}}$ tends to zero, leads to the conclusion that also y_F tends to zero as $t \to \infty$.

Local asymptotic stability

Under the proposed control law, the closed-loop error kinematic model is given by

$$\begin{aligned}\dot{y}_F &= V\,\sin\left(\bar{\psi}_d+\tilde{\psi}\right)+\left(v_{d_x}-\omega_d\,\Delta y\right)\sin\psi_f\\ &\quad-\left(v_{d_y}+\omega_d\,\Delta x\right)\cos\psi_f\\ \dot{\tilde{\psi}} &= -g_1\,\tilde{\psi}-g_2\,y_F\Big(\left(v_{d_x}-\omega_d\,\Delta y\right)\sin\psi_f\\ &\quad-\left(v_{d_y}+\omega_d\,\Delta x\right)\cos\psi_f\Big)\frac{1-\cos\tilde{\psi}}{\tilde{\psi}}-g_2\,V\,y_F\,\cos\bar{\psi}_d\frac{\sin\tilde{\psi}}{\tilde{\psi}},\end{aligned} \tag{26}$$

which is a nonautonomous[2] nonlinear system of the form $\dot{e}=f\left(t,e\right)$, where $e=\left[\begin{array}{cc} y_F & \tilde{\psi}\end{array}\right]^T$.
The Jacobian matrix $A(t)=\frac{df}{de}\left(t,e\right)|_{e=0}$ is given by

$$A(t)=\left[\begin{array}{cc} 0 & V\cos\bar{\psi}_d\\ -g_2\,V\cos\bar{\psi}_d & -g_1\end{array}\right].$$

One can now propose a positive definite bounded matrix $P(t)=P^T(t)$ that satisfies the Lyapunov equation

$$P(t)\,A(t)+A^T(t)\,P(t)=-Q(t),$$

where $Q(t)=Q^T(t)\ >0$ is a bounded matrix. In particular, by setting

$$P\left(t\right)=\left[\begin{array}{cc}\frac{g_2\,V^2\cos^2\bar{\psi}_d\left(g_2+1\right)+g_1^2}{2\,g_1\,g_2\,V^2\cos^2\bar{\psi}_d} & \frac{1}{2\,g_2\,V\cos\bar{\psi}_d}\\ \frac{1}{2\,g_2\,V\cos\bar{\psi}_d} & \frac{g_2+1}{2\,g_1\,g_2}\end{array}\right]$$

which is a positive definite bounded matrix for every positive constants g_1 and $g_2>0$, one obtains

$$Q(t)=I_2.$$

Thus, considering the following Lyapunov function

$$V_2=e^T\,P(t)\,e,$$

its derivative is given by

$$\dot{V}_2=-e^T\,Q(t)\,e\leqslant-\parallel e\parallel^2,$$

and it is straightforward to conclude that under the proposed control law, the origin of the closed-loop error kinematic model (26) is an exponentially stable equilibrium point [35]. □

Figure 3 illustrates an example where a UAV flying at $\|V_0\|=V=15$m/s follows a moving straight line path rotating around a fixed point $p_0=\left[\begin{array}{cc}0 & 0\end{array}\right]$ with an angular velocity $\omega_d=0.025$rad/s. The controller gains were set to $g_1=1$ and $g_2=0.002$. By assigning a larger weight to orientation errors when compared to position errors, a smoother convergence to the reference paths is achieved. The angle ψ_p between North and the straight line that connects the origin of $\{I\}$ with ${}^Ip_d(\ell)$ at time instant $t=0$s is zero and increases as t increases, due to path's angular velocity ω_d. For an observer fixed with the path-frame and standing at the origin p_0 looking towards the positive direction of $\overrightarrow{x}_P$, the UAV will be moving in a straight line with a decreasing forward speed as the distance between the origin of the path-transport frame and the Serret-Frenet frame increases. From equation (14) one can verify that this is due to the fact that as $\sqrt{\Delta x^2+\Delta y^2}$ increases, the desired heading deviation between the wind frame and the Serret-Frenet frame $\bar{\psi}_d$ also increases as the UAV needs to compensate for an increasing normal component of v_P with respect to the normal to the path $\overrightarrow{n}$. Thus, in this particular example, the straight line path length is limited to be smaller than $\frac{V}{|\omega_d|}=600$m (c.f. equation (15)) in order to ensure that the MPF is well posed.

[2]Note that $f\left(t,e\right)$ depends on the possibly time-varying variables v_d and ω_d.

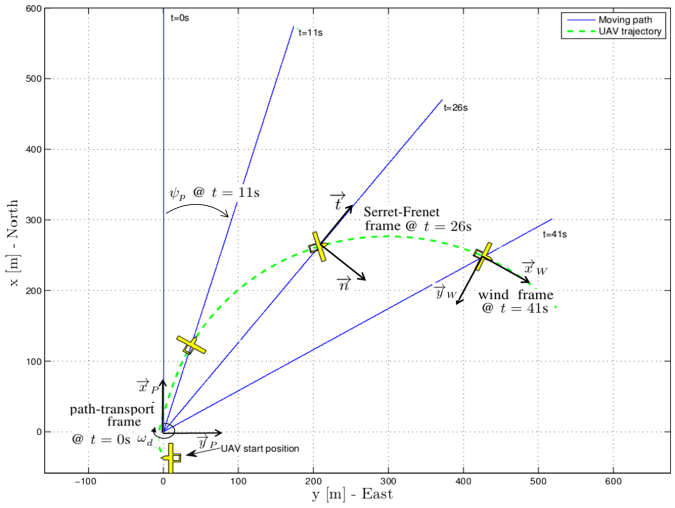


Fig. 3. Numerical simulation example: UAV following a moving straight line that is rotating around a fixed point $p_0=\left[\begin{array}{cc}0 & 0\end{array}\right]$ with an angular velocity $\omega_d=0.025$rad/s.

IV. Single and Multiple Targets Tracking Problem Formulation as MPF

This section describes how the MPF formulation previously presented can be applied to single and multiple targets tracking problems.

A. Single target tracking

Application of the MPF control law to a single ground target tracking scenario is straightforward. A given path with a predefined geometry can be attached to the desired target (moving together with it - with possible time varying linear and angular velocities) and the control law given by equation (20) can be used to make the UAV converge to and follow the moving path. To illustrate this application, Figure 4 shows a numerical simulation where a UAV flying at 20m/s airspeed is required to track a target by following a lemniscate path that is moving together with the target with 300m width, keeping the line that connects the two foci always perpendicular to ψ_d (thus $\dot{\psi}_p=\dot{\psi}_d$). The wind velocity W_t is set to 10m/s (blowing from South) between 80 and 150 seconds (simulation time) and is set to zero otherwise. The target was moving with time-varying linear and angular velocities according to

$$\begin{aligned}\left(p_{t_x},\ p_{t_y},\ \psi_t,\ \|v_t\|\right)|_{t=0} &=(0\text{m},\ 0\text{m},\ 0,\ 4\text{m/s})\\ \|\dot{v}_t\| &=0.2\sin\left(0.07\,t\right)\text{m}/s^2\\ \dot{\psi}_t &=0.02\cos\left(0.03\,t\right)\text{rad}/s\end{aligned} \tag{27}$$

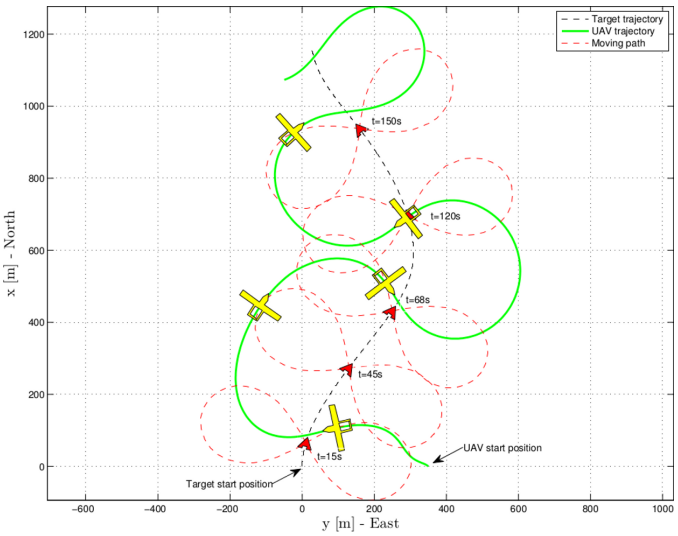


Fig. 4. Numerical simulation - aircraft's trajectory following a target between t=0s and t=200s.

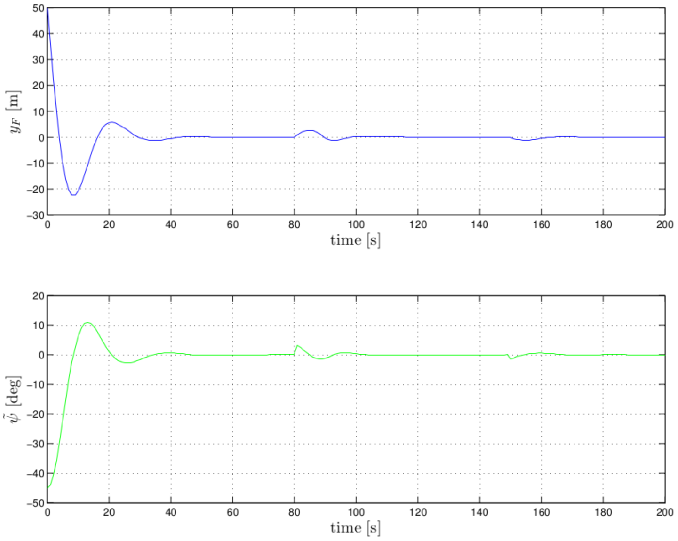


Fig. 5. Numerical simulation - position and heading errors.

where v_t corresponds to the target velocity and ψ_t is the target heading. In order to attach the desired path to the target we set the path fixed frame with $p_0|_{t=0} = \begin{bmatrix} p_{t_x} & p_{t_y} \end{bmatrix}|_{t=0}$, $v_d = v_t$ and $\psi_d = \psi_t$.

The controller parameters used are listed in Table I.

TABLE I
NUMERICAL SIMULATION - CONTROLLER PARAMETERS.

$g_1 = 0.22$	$\omega_d = \dot{\psi}_d$
$g_2 = 0.0002$	$\dot{\omega}_d = \ddot{\psi}_d = -0.0006\sin(0.03\,t)$

Figure 5 demonstrates the performance of the overall control system. At time instants $t = 80$s and $t = 150$s a perturbation on the distance and heading errors due to the imposed sudden change on the wind velocity is observed. Later, in Section V-B we present flight test results for a single target tracking scenario, thus allowing a comparison between simulation and experimental results.

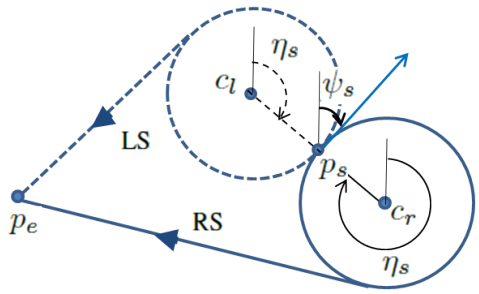


Fig. 6. Dubins path: RS or LS.

B. Multiple targets tracking

Consider now the problem of assigning a desired path to a vehicle that should visit a set of fixed targets whose positions and visiting sequence are known. Assuming that the vehicle and the target are initially at least $2\,r_{min}$ apart, one strategy is to use a Dubins path[3] computed without specifying the course angle at the end point [22]. Figure 6 illustrates this strategy for path generation, given the start position $p_s = (x_s, y_s)$, the initial course angle ψ_s, the minimum radius r_{min} and the end position p_e. For simplicity of notation, a curved segment of radius r_{min} along which the vehicle executes a clockwise maximum curvature rotational motion is denoted by R. L denotes a maximum curvature rotation counterclockwise and the segment along which the vehicle travels straight is denoted by S. To choose from the RS and LS options, it is necessary to compute the path length for these two cases depicted in Figure 6. In particular, we have

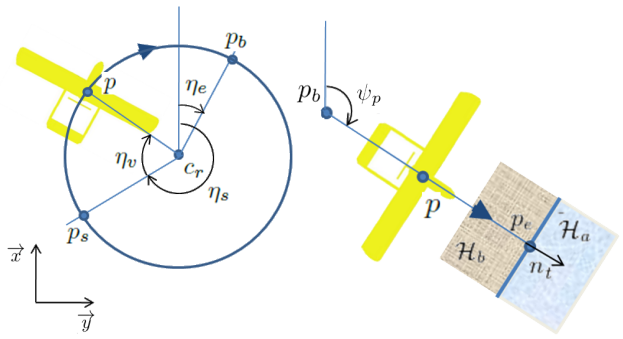
$$d_{P_i} = d_{S_i} + d_{C_i}\, r_{min} \tag{28}$$

where index i equals l or r for L or R curved segments respectively, d_{C_i} is the corresponding arc length and d_{S_i} is the distance along the straight line S.

As soon as the vehicle arrives at one of the targets, another Dubins path towards the next target is computed using the current course ψ_s and again without specifying the course angle at the new target. This strategy can also be used with moving targets, but in this case the Dubins path that connects the last visited target and the next one will be changing over time by taking into account the current target position, heading and speed. More precisely, the idea is as follows:

- Compute the right and left Dubins path turning circles given the vehicle's current position p_s and course angle ψ_s (see Figure 6);
- From (28) select the curved segment with the minimum path length and set it as the reference path until a criterion (discussed in the sequel) to decide when to abandon the right or left turn at the maximum turning rate and start following a moving straight line that connects the current vehicle position and the target's position is met; and

[3]The shortest path that connects two points in the two-dimensional Euclidean plane $x-y$, considering a constraint on the curvature of the path and that the vehicle moves at constant speed.



(a) Circular section. (b) Straight line section.

Fig. 7. Dubin's path relevant parameters for moving targets, illustrating for the right circular segment case.

- Switch from the current target to the next target when a proximity condition (to be defined) is satisfied.

As in the fixed targets case, the circular section of the generated path does not move. Consequently, the chosen circle center c_i and the angle η_s (between the line that connects c_i to the vehicle's initial position, p_s, and the $\vec{x}$ axis - see Figure 7) do not change over time ($\dot{c}_r = 0$ and $\dot{\eta}_s = 0$). The proposed criterion is to switch to a straight line when

$$\eta_{v_i} \geq \begin{cases} \langle 2\,\pi - \eta_e + \eta_s \rangle & \text{if } i = l, \\ \langle 2\,\pi + \eta_e - \eta_s \rangle & \text{if } i = r \end{cases} \tag{29}$$

where η_e corresponds to the angle between the line that connects c_i to the boundary position p_b (at which the straight line should start to be followed) and the $\vec{x}$ axis and η_v is the angular distance between η_s and the vehicle's current position, p (see Figure 7a). In (29), $\langle \eta \rangle \triangleq \eta \mod 2\,\pi$ and mod is the modulus operator that makes $\eta \in [0\ \ 2\,\pi]$. In this case, the arc length of the chosen circular segment d_{C_i} will be a function of time because p_e, η_e and p_b can be time varying. Note that the straight line segment will have a fixed initial position p_b and a moving end point p_e solidary with the current target's position, which is similar to the MPF example discussed at the end of Section III-B.

The angle ψ_p between this moving straight line and the $\vec{x}$ axis can be computed through $\psi_p = \arctan\left(\frac{\Delta y_t}{\Delta x_t}\right)$, where $(\Delta x_t, \Delta y_t)$ are the coordinates of the vector connecting p_b to p_e in the inertial frame $\{I\}$ (see Figure 7b). Its derivative is given by

$$\dot{\psi}_p = \frac{1}{1 + \left(\frac{\Delta y_t}{\Delta x_t}\right)^2} \frac{\Delta \dot{y}_t\, \Delta x_t - \Delta y_t\, \Delta \dot{x}_t}{\Delta x_t^2}. \tag{30}$$

Recalling the MPF problem formulation and the illustration example discussed in Section III-B (see also Figure 3), it is now possible to define the moving path $\mathcal{P}_d$ to be followed for the case of the multiple target tracking. For the circular segment, ${}^P p_d(\ell)$ components correspond to the circle parametric equations, p_0 contains the circle center coordinates and $v_d = \omega_d = 0$ (thus corresponding to the classical path following case). For the straight line segment, ${}^P p_d(\ell)$ is formed by the straight line parametric equations, $p_0 = p_b$, $v_d = 0$ and $\omega_d = \dot{\psi}_p$. From equation (30), under the assumption that the UAV and the target are initially at least $2\,r_{min}$ apart, it can easily be shown that if the target's speed is always smaller than the UAV's speed $\|v_t\| < V$, then, the conditions imposed by equations (15) and (17) always hold (independently from the distance between the UAV and the target for $t > 0$) and thus de MPF problem is always well posed.

A rule must also be defined to switch from the current target to the next target. To this end, consider the line perpendicular to the vector that connects p_b to the current target position p_e, that passes through the target position. This line divides the plane in two half-planes, the half-plane $\mathcal{H}_b$ of the points "behind the target" and the half-plane $\mathcal{H}_a$ of the points "after the target" (see Figure 7). One approach is to make the vehicle to switch to the next target when it enters the half-plane $\mathcal{H}_a$. This can be formalized as follows. Let $\vec{n}_t$ be the unitary vector

$$\vec{n}_t = (\cos\,\psi_p,\, \sin\,\psi_p)$$

connecting p_b to p_e. The vehicle enters $\mathcal{H}_a$ when

$$(p - p_e)^T\ \vec{n}_t \geqslant 0. \tag{31}$$

Algorithm 1 (in Appendix) summarizes the above described strategy for a single vehicle to visit a group of moving targets in a given order s_t.

Figure 8 shows a numerical simulation result where a UAV moving at $\|V_0\| = V = 30$m/s with minimum turning radius $r_{min} = 200$m tracks three targets heading North at constant speed $\|v_t\| = 15$m/s in a pre-determined order (from target #1 to target #3) using Algorithm 1 to generate path parameters and the MPF control law given by equation (20) to generate its turn rate commands. Additionally, Figure 8 presents the optimal trajectory (the one that minimizes the interception time) that the autonomous vehicle should have adopted, if it had a priori knowledge of the targets trajectories. Note that, in the proposed algorithm, each subsequent optimal interception point presented in Figure 8 is computed considering that the starting position corresponds to the previous autonomous vehicle true interception coordinates as detailed in Figure 9. From this observation, it can be inferred that we may have an improvement of performance if the target's heading and velocity can be estimated to then obtain the expected interception position. Algorithm 1^* implements this idea. It uses the same strategy as Algorithm 1 but now the estimated target interception position is used as a "virtual target" instead of the current target's actual position (see Figure 10). A detailed description of Algorithm 1^* and formal convergence conditions under which the vehicle reaches its moving target are presented along with a perfomance metric (with respect to the interception time) in Appendix.

V. Flight Tests

In this section we first present the flight test architecture used to evaluate the MPF on real target tracking scenarios, namely single and multiple targets tracking. Then, Section V-B presents the field tests for the single target tracking scenario and Section V-C presents the main test results for the case of multiple targets.

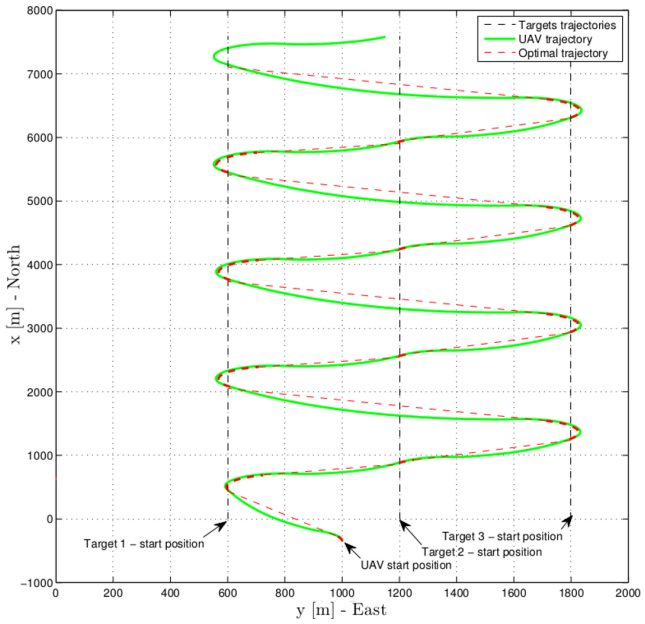


Fig. 8. Numerical simulation: an autonomous vehicle tracks three targets heading North at constant speed using Algorithm 1 and the MPF control law given by equation (20).

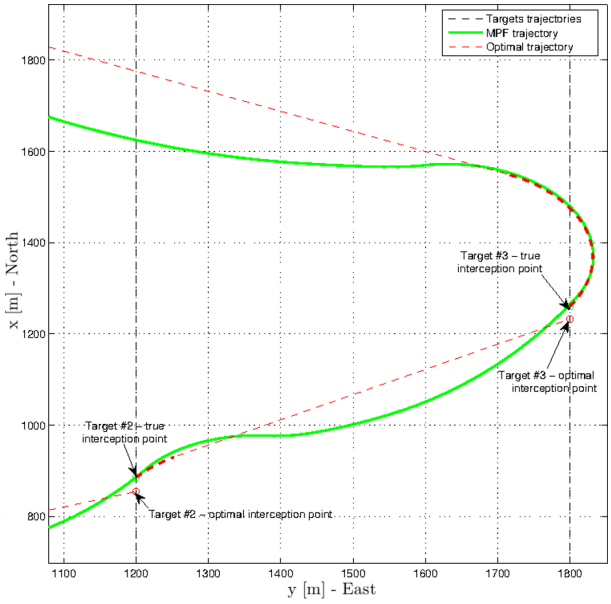


Fig. 9. Numerical simulation: optimal trajectory generation detail.

A. *Flight test architecture*

The ground target tracking algorithm was implemented and tested on the ANTEX-X02 platform (see Figure 11 (a)). This is one of the platforms built from scratch at the Portuguese Air Force Academy Research Center (AFA) and available for tests within the PITVANT[4] and SEAGULL[5] projects. The main characteristics of ANTEX-X02 are listed in Table II. The platform is equipped with a Piccolo II autopilot that plays the role of an inner-loop controller that provides the

[4]http://www.emfa.pt/www/po/unidades/subPagina-10D00-019.005.003.004-pitvant

[5]http://www.criticalsoftware.com/pt/seagull

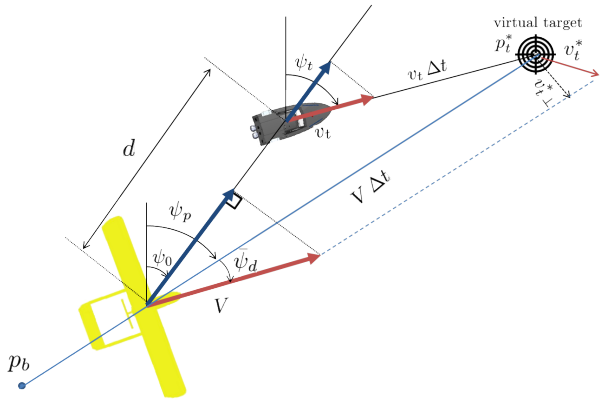


Fig. 10. Illustration of a UAV intercepting a moving target, heading towards its current estimated interception position.

(a) ANTEX-X02 (b) Ground station

Fig. 11. ANTEX-X02 and ground control station at flight test site.

required actuation signals for the UAV's control surface's deflections and the engine power according to the UAV current state that is inferred from the measurements of the on-board sensors, and the reference signal that is transmitted by the ground target tracking algorithm. The autopilot relies on a mathematical model parameterized by the aircraft geometric data and has a built-in wind estimator. Several model and controller parameters can be set by the user [36]. In this work, the parameters collected from more than 100 hours of flight with the ANTEX-X02 were used. The proposed control

TABLE II
ANTEX-X02 - MAIN FEATURES.

Maximum takeoff weight	10kg
Wingspan	2.415m
Payload	4kg
Maximum Speed	100km/h
Autonomy	3h

algorithms for the UAV were implemented on a laptop (Computer 2) connected to the Piccolo Command Center (running on another computer - Computer 1) via an ethernet port to receive the sensor data from the Piccolo autopilot and provide the references to the aircraft as it is illustrated in Figure 12. Robot Operating System (ROS) software [37] was used to establish a convenient communication interface between these two computers. The targets coordinates were generated according to a stochastic signal - see Section V-C - to simplify

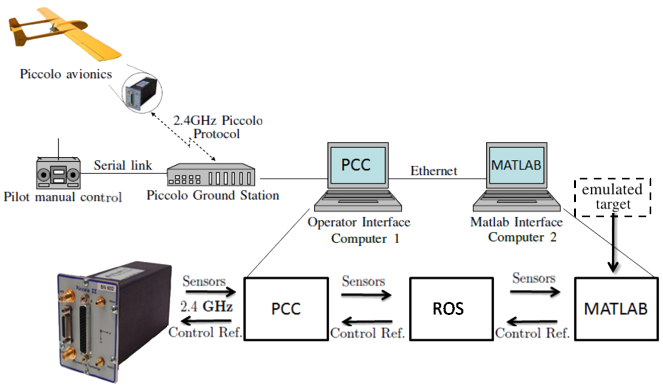


Fig. 12. Fligth test operation frame.

the test implementation[6]. Computer 2, after receiving the sensors data from the Piccolo, computes and provides the Piccolo Command Center (Computer 1) the control references that are then sent to the Piccolo autopilot at a telemetry rate of 2Hz. All flight data was monitored at the base station using the Piccolo Command Center. High gain directional antennas were used at the ground station (see Figure 11 (b)) in order to reduce communications losses. The implemented control system architecture for the multiple targets tracking flight test is presented in Figure 21 and described in detail in Appendix.

Field tests were conducted at the Portuguese Air Force's UAV test facility, located at Ota, Portugal[7]. For the field tests, a few safety measures were introduced. The bank reference sent to the aircraft was limited to 20°. The telemetry signals from the aircraft were synchronized with the targets data and then fed to the controller to compute the bank reference to the aircraft. In the event of communications loss, the Piccolo assumed the last bank reference for a maximum period of 5 seconds. After that period, the mission would be aborted and the aircraft would be sent to a predefined lost-communication waypoint.

B. Single target tracking

In this experiment, the goal was to make the UAV track a ground vehicle by following a lemniscate centered at the target position with a 300m width, keeping the line that connects the two foci always perpendicular to ψ_t. The UAV was flying at constant altitude - 200m above the ground - with 20m/s commanded airspeed. The wind was blowing from south-east with 3m/s average speed and the controller parameters were set to $g_1 = 0.22$ and $g_2 = 0.00012$. The target trajectory was emulated using the same parameters as in the numerical simulation (cf. equations (27)).

The obtained qualitative behaviour of the UAV is similar to the results presented for the numerical simulation. Figure 13 shows the overall UAV and target trajectories. The UAV, target and path positions at sample time instants are also

[6]For the multiple target tracking scenario, the obtained target trajectories were by construction similar to actual vessels trajectories at the Portuguese coastline - https://www.marinetraffic.com/

[7]http://www.emfa.pt/www/po/unidades/subPagina-10D00-019.001.003.010.003-infraestrutura-de-testes-de-uav

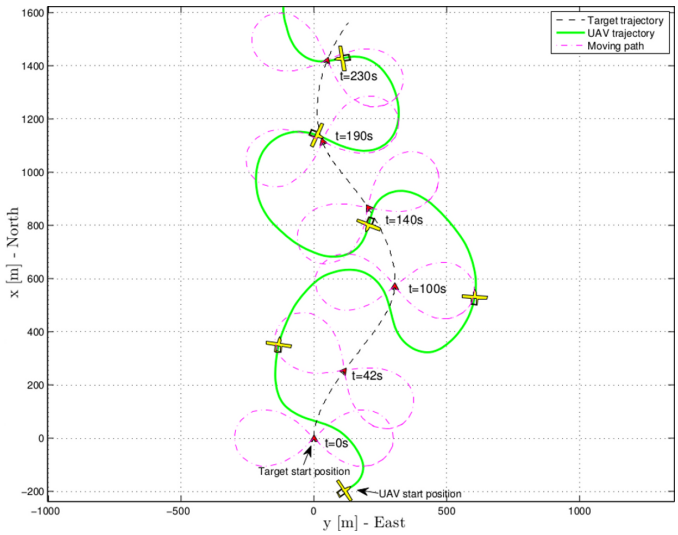


Fig. 13. Aircraft's trajectory following a target on the ground (from $t = 0$s to $t = 250$s).

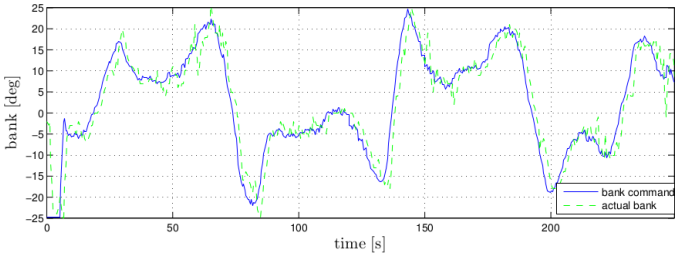


Fig. 14. Commanded and actual bank for the single target tracking scenario.

presented. Figure 14 shows that the bank command and the bank value are kept within the linear region. There is a non negligible delay between the commanded bank and its actual value thus decreasing system performance. However, distance and angular errors plots in Figure 15 demonstrate the good performance of the implemented control system.

C. Multiple targets tracking

In this case, the UAV was required to track four ground vehicles 400m above the ground. During the flight test, the wind was blowing from south-east with 7m/s average speed and wind gusts of about 13m/s. The UAV airspeed reference was set to 23m/s. The controller parameters for this flight test were set to $g_1 = 0.15$ and $g_2 = 0.00009$.

Figure 16 shows the overall vehicle and targets trajectories. Targets #1 and #2 started moving North ($\psi_t|_{t=0} = 0$rad) while targets #3 and #4 started moving South ($\psi_t|_{t=0} = \pi$rad). Targets sequenced order to be visited was predefined as $s_t = [1, 2, 3, 4]$. All targets started moving at 3m/s ($\|v_t\|_{t=0} =$ 3m/s). Both $\|\dot{v}_t\|$ and $\dot{\psi}_t$ were defined as stochastic signals with a normal distribution, namely

$$\|\dot{v}_t\| \sim \mathcal{N}(0, 0.05)$$
$$\dot{\psi}_t \sim \mathcal{N}(0, 0.03).$$

Distance and angular errors are depicted in Figure 17 showing the good performance of the control strategy in the presence

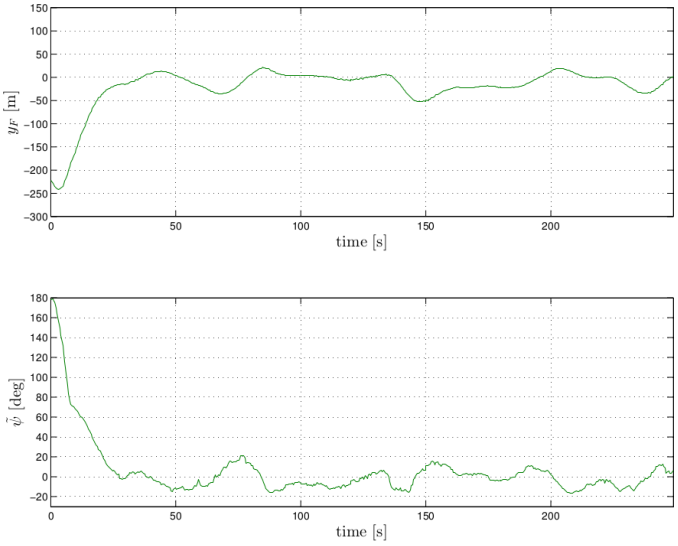


Fig. 15. Distance and heading errors for the single target tracking scenario.

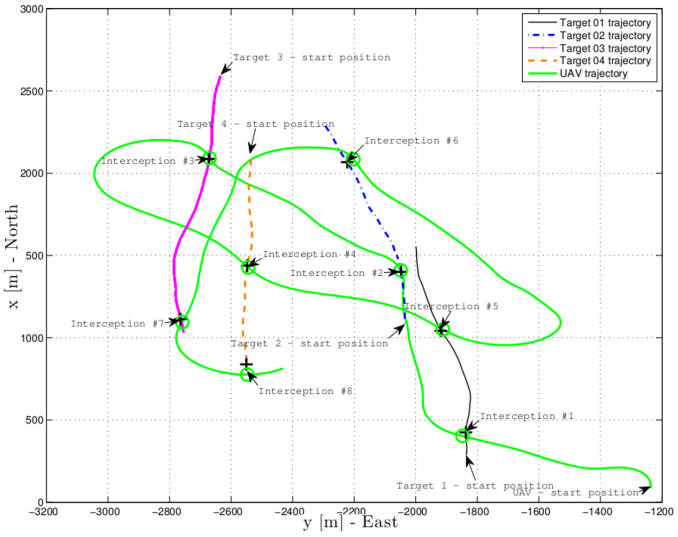


Fig. 16. Aircraft's trajectory following multiple targets on the ground.

of relatively high communication latency (see Figure 20). The vertical lines indicate the time instants at which each target interception has occurred.

The distances between the UAV and the current target at each time of interception (where condition (31) is met) are presented in Figure 19. The interception distances are always below the UAV's minimum turning radius, and thus it is assumed that the current targets at the interception times were always inside the on-board camera's footprint (which typically carry vision sensors that have footprints wider than $2\,r_{min}$). The actual interception distance is mainly due to the delay introduced by the pre-filter used to compute the current target's expected position (see Figure 21 - further details can be found in Appendix). Figure 18 shows that the bank reference sent to the aircraft was saturated for relatively long periods of time, thus decreasing the system's performance. This is particularly noticeable at time instant $t = 250$s. However, the control system proved to be sufficiently robust, tackling all these

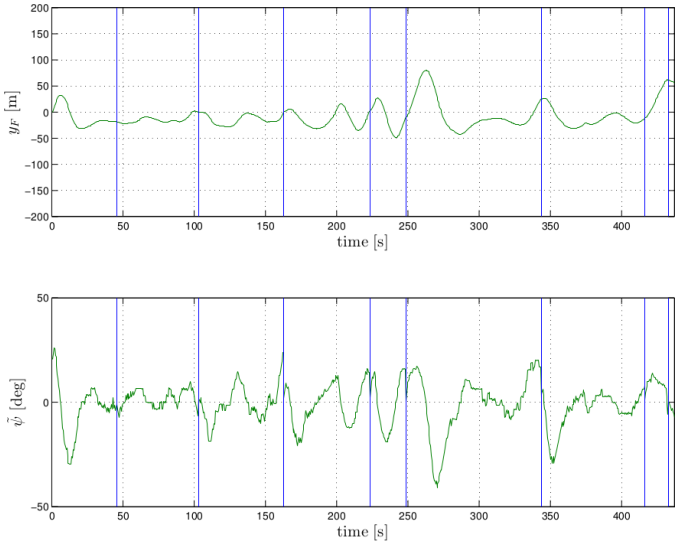


Fig. 17. Distance and heading errors for the multiple targets tracking scenario. The vertical lines indicate the time instants at which each target interception has occurred.

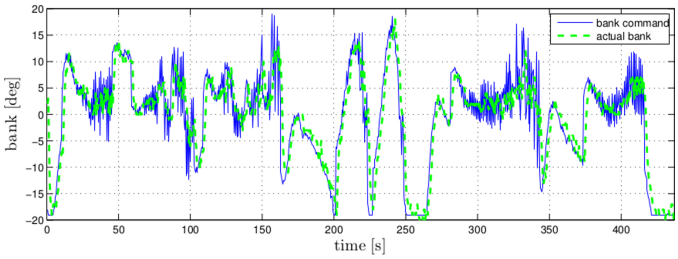


Fig. 18. Commanded and actual bank for the multiple targets tracking scenario.

situations that were not completely taken into account during the control design.

VI. Conclusions

An error space for moving path following was presented, by formally extending the classic path following algorithms to the case of time varying paths in a two dimensional space. The error space derived was used to design a kinematic ground target tracking control law for an Unmanned Aerial Vehicle (UAV) equipped with an autopilot that accepts references at the kinematic level. An application was made to the problem of tracking single and multiple targets on the ground using a fixed wing UAV. Both single and multiple targets tracking scenarios use the same MPF control law and the proposed methods encompass the classical guidance strategies presented in the literature. Formal convergence proofs and performance metrics are provided and flight tests results demonstrate the effectiveness of the proposed methods. Future work will include the flight test of the control law on-board the aircraft. Indeed, system performance can be further improved when the control law is implemented on-board the aircraft (with direct access to the sensors data - allowing for higher control rates, which otherwise are limited by bandwidth constraints). Additionally, the problem of acquiring target's information using passive sensors should also be addressed. We foresee the use of an

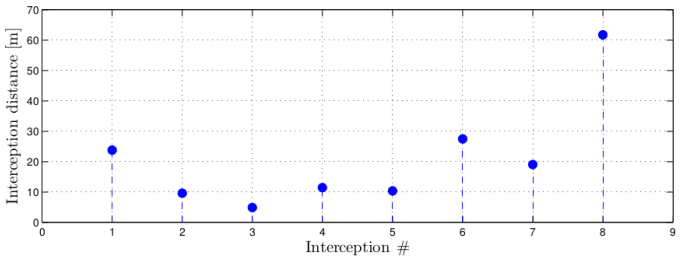


Fig. 19. Actual distance between UAV and current target at time of interception for the multiple targets tracking scenario.

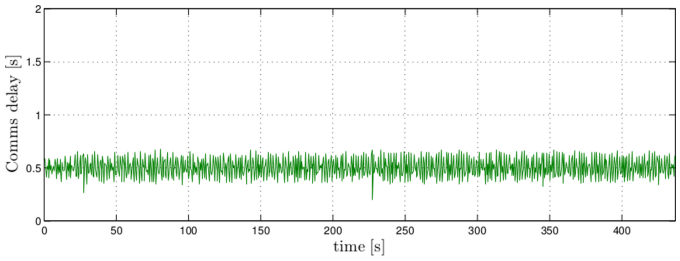


Fig. 20. Communications latency for the multiple targets tracking scenario.

Automatic Identification System (AIS) to track actual targets (vessels) on the Portuguese coastline.

VII. Acknowledgements

The authors would like to acknowledge the contribution of Captain Elói Pereira, Captain Gonçalo Cruz and Pedro Marques da Silva, former Captain of the Portuguese Air Force. The authors are also in debt to every member of the SEAGULL and PITVANT teams.

Appendix

This section provides a detailed description of Algorithm 1, Algorithm 1^* and the implemented control system architecture. Formal convergence conditions under which the vehicle reaches its moving target using Algorithm 1^* are presented in the sequel, along with a performance metric result.

A. Path generation Algorithm 1 *and "virtual target" estimation for Algorithm* 1^*

The strategy for a single vehicle to visit a group of moving targets in a given order s_t as proposed in Section IV-B is summarized in Algorithm 1. Algorithm 1^* implements the idea of using the estimated target interception position as a "virtual target" instead of the current target's actual position as in Algorithm 1 (see Figure 10). The estimated target interception position for Algorithm 1^* is computed as follows.

Let $\|v_t\|$ and ψ_t be the current target speed and course angle respectively. Additionally, let p_{t_0} be its current center of mass coordinates. Thus, assuming that the target will keep moving in a straight line at constant speed, its position after time Δt_t will be

$$p_t\left(\Delta t_t\right) = p_{t_0} + \|v_t\| \left(\cos\psi_t,\, \sin\psi_t\right) \Delta t_t. \tag{32}$$

ALGORITHM 1: Returns the path parameters for a single vehicle to visit a group of moving targets in a given order.

Input: Vehicle start pose (p, ψ), sequenced target's position $s_t = (t_1, t_2, \ldots, t_m)$ with $t_j = \left(x_{t_j}, y_{t_j}\right)$, and path constraints r_{min}.
Output: Minimal feasible path parameters (RS or LS).

Initialization: Set $j = 1 \Rightarrow$ select first target;

while $j \leqslant m \Rightarrow$ *vehicle hasn't visited all the targets* **do**
1. $(p_1, \psi_1) \leftarrow (p, \psi)$;
2. $p_2 \leftarrow t_j$;
3. Compute the length of two possible paths (RS and LS) between the vehicle's current position p_1 and p_2: $\{L_{RS}, L_{LS}\}$;
4. Choose the path that has the minimum length: $L \leftarrow \arg\min\{L_{RS}, L_{LS}\}$;
5. Compute and output chosen path parameters: c_i, η_s, η_e, η_v and d_{C_i}.

if $\langle \eta_v \rangle_i \geqslant d_{C_i} \Rightarrow$ vehicle has reached end of circular section.
6. Set $p_b = p \Rightarrow$ Set straight line start position.
while $p \notin \mathcal{H}_a \Rightarrow$*vehicle hasn't reached the end of straight line* **do**
7. Compute and output straight line segment path parameters: p_b, ψ_p, $\dot{\psi}_p$.
end
8. Set desired target as the next target $\Rightarrow j \leftarrow j + 1$;
9. Return to 1.
else
10. Return to 5.
end
end

Consequently, under the same assumption that the target will keep moving in a straight line at constant speed, one can write the autonomous vehicle's total path length d_{P_i} (see equation (28)) as a function of Δt_t as

$$d_{P_i}\left(\Delta t_t\right) = d_{S_i}\left(\Delta t_t\right) + d_{C_i}\left(\Delta t_t\right) r_{min}.$$

Hence, assuming that the vehicle moves with constant speed V, the elapsed time between vehicle's initial and final positions, Δt_v, can be obtained through

$$\Delta t_v = \frac{d_{P_i}\left(\Delta t_t\right)}{V}. \tag{33}$$

Note that if $\Delta t_v = \Delta t_t = \Delta t$, it physically means that both vehicle and target take the same time Δt to reach their final destinations, and thus, they arrive simultaneously to a given "rendezvous" point at coordinates $p_e = p_t\left(\Delta t\right)$. Variable Δt can be computed making $\Delta t_v = \Delta t_t = \Delta t$ and solving equation (33) for Δt. If the target moves at constant speed $\|v_t\|$ and heading ψ_t, the "virtual target" coordinates will be kept constant and therefore the actual path travelled by the vehicle will correspond to the optimal path.

For the general case of targets moving with time varying velocity and/or heading ($\dot{v}_t \neq 0$, $\dot{\psi}_t \neq 0$) we propose the following procedure that consists in filtering first the "instantaneous" virtual target coordinates given by $p_t\left(\Delta t\right)$ - see equation (32) - in order to obtain a filtered output of $p_t\left(t\right)$ that will be sufficiently smooth with bounded derivatives,

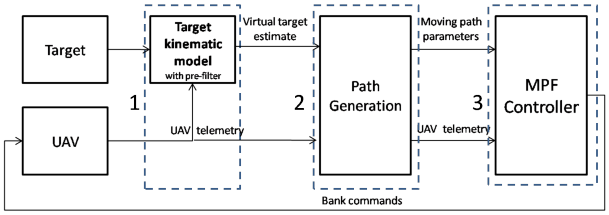


Fig. 21. Control system architecture.

and therefore it will play the role of a reference target to comply with the MPF requirements. The filter was adopted from [38] and consists of a cascade of two second-order non-linear filters. The input of the first pre-filter is the output position estimate of the target vehicle from $p_t\,(\Delta t)$, and the output of the second pre-filter corresponds to the filtered target interception position, p_t^* (the input of the path generation algorithm). The state space representation of each filter can be written as

$$\begin{aligned} \dot{x}_1 &= a_1\ \tanh\left(x_2\right) \\ \dot{x}_2 &= a_2\ \tanh\left(k_1\, u - k_1\, x_1 - k_2\, x_2\right) \end{aligned} \tag{34}$$

where it is assumed that k_1, k_2, a_1,and a_2 are scalar positive constants such that $k_2\, a_2 \ > \ k_1\, a_1$. By linearization, it is straightforward to show that the pre-filter is locally input to state stable. Invoking LaSalle's Principle and using the Lyapunov function given by

$$V = \frac{1}{2}\, e' \begin{bmatrix} \frac{k_1^2}{k_2\, a_2} & \frac{k_1}{2\, a_2} \\ \frac{k_1}{2\, a_2} & \frac{k_2}{a_2} \end{bmatrix} e + \frac{k_1\, a_1}{k_2\, a_2^2} \int_0^{e_2} \tanh\left(y\right)\, \mathrm{d}y$$

where $e = (e_1, e_2) := (x_1 - u,\ x_2)$, it can be proven that for constant input commands ($\dot{u} = 0$), the equilibrium point $x_1 = u$, $x_2 = 0$ is globally asymptotically stable [38]. Thus, the filtered target interception position $p_t^*(t)$ will converge to the "instantaneous" target interception position $p_t(t)$ if p_t is constant, otherwise it will stay close (depending on the rate of p_t).

The proposed strategy to intercept multiple targets behaves similarly to the guidance strategies LOS and CB described in Section I, depending on the commanded "look ahead distance" for the virtual target position $\|v_t\|\,\Delta t$ (cf. equation (33)). In the case that Δt is set to zero (and thus Algorithm 1 is used) the proposed strategy is identical to the missile line of sight guidance. Otherwise (when Algorithm 1^* is used), the system behaves similarly to the constant bearing guidance law described in Section I.

B. Overall control architecture

The implemented control system architecture for the flight test is presented in Figure 21 encompassing three stages:

1) From target and UAV pose updates, filter the estimated interception position to ensure sufficiently smooth position estimates and boundedness of their derivatives.
   Inputs:
   a) UAV center of mass coordinates $p = \begin{bmatrix} x & y \end{bmatrix}^T$ and ground speed V;
   b) target's current position p_{t_0}, heading ψ_t, and speed $\|v_t\|$;
   c) pre-filter parameters a_1, a_2, k_1 and k_2.
   Output:
   a) filtered target interception position estimate p_t^*, heading ψ_t^*, and speed $\|v_t^*\|$.
2) Use the path generation algorithm to generate the time varying (moving) path that connects the UAV's current position to the filtered target interception estimate;
   Inputs:
   a) UAV center of mass coordinates $p = \begin{bmatrix} x & y \end{bmatrix}^T$, course angle ψ, ground speed V and vehicle constraint r_{min};
   b) virtual target variables p_t^*, ψ_t^*, and $\|v_t^*\|$ from previous stage;
   Output:
   a) path parameters c_i, η_s, η_e, p_b, ψ_p and $\dot{\psi}_p$.
3) Use the MPF algorithm to allow the UAV to converge to the current target.
   Inputs:
   a) UAV center of mass coordinates $p = \begin{bmatrix} x & y \end{bmatrix}^T$, course angle ψ, and ground speed V;
   b) path parameters c_i, ψ_p, $\dot{\psi}_p$, η_s and η_e from previous stage;
   Output:
   a) turn rate command $\dot{\psi}$ to the UAV.

C. Convergence conditions

The next result concerns the conditions under which the UAV reaches its moving target when navigating using Algorithm 1^* together with the MPF control law. Figure 10 presents the considered problem kinematics. It is assumed that the target has limited speed $0 \leqslant \|v_t\| \leqslant V_{t_{max}}$ (where $V_{t_{max}}$ is the assumed target's maximum speed) and angular velocity $|\dot{\psi}_t| \leqslant \dot{\psi}_{t_{max}}$ (where $\dot{\psi}_{t_{max}}$ is the assumed target's maximum angular speed). Considering the general case were the target moves with bounded time varying linear and angular velocity, the virtual target position $p_t^*(t)$ output from the implemented pre-filter (equation (34)) will also move with bounded time varying linear and angular velocities. These bounds are the same considered for the actual target, namely $0 \leqslant \|v_t^*\| \leqslant V_{t_{max}}$ and $|\dot{\psi}_t^*| \leqslant \dot{\psi}_{t_{max}}$. Thus, the path's straight line segment that connects the fixed position p_b to the virtual target position p_t^* may be moving with a given angular velocity $\dot{\psi}_p$ (see equation (30)). From Figure 10 it is straightforward to verify that the desired heading deviation $\bar{\psi}_d$, (cf. Section II) is

$$\bar{\psi}_d = \arcsin\left(\frac{v_{t\,\perp}^*}{V}\right),$$

where $v_{t\,\perp}^*$ is the normal component of v_t^* with respect to the straight line segment that connects p_b to p_t^* and thus one can compute the upper bound for $\bar{\psi}_d$ as

$$|\bar{\psi}_d| \leqslant \arcsin\left(\frac{V_{t_{max}}}{V}\right). \tag{35}$$

Additionally, from Figure 10 one can also verify that

$$V \sin(\psi_p - \psi_0) = v_t \sin(\psi_t - \psi_0)$$

and thus, for any target's orientation ψ_t, one can write a bound for $|\psi_p - \psi_0|$ as

$$|\psi_p - \psi_0| \leqslant \arcsin\left(\frac{V_{t_{max}}}{V}\right). \tag{36}$$

The upper bounds in (35) and (36) will be used in the sequel. Consider the proposed control system architecture and let ψ_0 be the angle between the line of sight vector that connects the UAV to the current target and the $\overrightarrow{x}$ axis, ψ_p be the angle between the path's straight line segment that connects the fixed position p_b to the virtual target position p_t^*, and the $\overrightarrow{x}$ axis (thus, in this particukar case, $\psi_f = \psi_p$), V be the UAV's ground speed, v_t and ψ_t be the current target's linear velocity and heading, respectively, and finally let $\bar{\psi}_d$ be the desired heading deviation. The following theorem holds.

***Theorem** 2*

The autonomous vehicle navigating under the MPF control law reaches the moving target in finite time for any target velocity and orientation when $V > 2\,V_{t_{max}}$.

Proof

From Figure 10 one can conclude that the relative closing velocity between the robot and the target satisfies

$$\dot{d} = -V \cos\left(\psi_p - \psi_0 + \bar{\psi}_d\right) + \|v_t\| \cos(\psi_t - \psi_0).$$

Note that, by definition, angle $\left(\psi_p - \psi_0 + \bar{\psi}_d\right)$ lies always in the interval $\left]-\frac{\pi}{2}, \frac{\pi}{2}\right[$. Since the target has limited speed $0 \leqslant \|v_t\| \leqslant V_{t_{max}}$, for any target's orientation ψ_t, one can write

$$\frac{\|v_t\| \cos(\psi_t - \psi_0)}{V} < \frac{V_{t_{max}}}{V}.$$

Additionally, if the following condition is met

$$\cos\left(\psi_p - \psi_0 + \bar{\psi}_d\right) > \frac{V_{t_{max}}}{V}, \tag{37}$$

using the transitive property one obtains

$$\cos\left(\psi_p - \psi_0 + \bar{\psi}_d\right) > \frac{\|v_t\| \cos(\psi_t - \psi_0)}{V},$$

which makes

$$\dot{d} = -V \cos\left(\psi_p - \psi_0 + \bar{\psi}_d\right) + \|v_t\| \cos(\psi_t - \psi_0) < 0.$$

Rewriting (37) yields the following convergence condition

$$|\psi_p - \psi_0 + \bar{\psi}_d| < \arccos\left(\frac{V_{t_{max}}}{V}\right), \tag{38}$$

which can be combined with (35) and (36) to obtain

$$2 \arcsin\left(\frac{V_{t_{max}}}{V}\right) < \arccos\left(\frac{V_{t_{max}}}{V}\right). \tag{39}$$

Thus, one can conclude that if $V > 2\,V_{t_{max}}$ this inequality is always satisfied and the robot will converge to the target in finite time for any target's speed or orientation. □

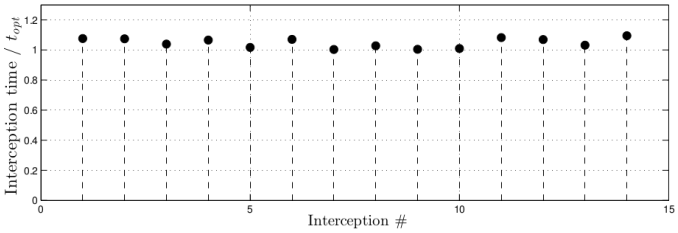


Fig. 22. Numerical simulation example: ratio between actual time for interception and optimal time for interception.

D. Performance metric

To assess the performance of the proposed method, computation of the optimal path (from current target to the next target) can be achieved by an exhaustive a posteriori search. For each interception it is possible to compute its corresponding optimal path and compare the actual time for interception using our proposed method with the optimal time of interception (see Figure 22). Table III presents the performance (in terms of average time for interception) of each of the proposed methods (Algorithm 1 and Algorithm 1*) with respect to the optimal case. The parameters in Table III were obtained from 500 Monte Carlo simulations. In each simulation, the number of targets was randomly generated according to a uniform distribution in the interval $[3, 10]$. Target's initial positions were also randomly generated according to a uniform distribution over a squared area of 5000 × 5000m centred at the origin of the inertial frame. Target's sequence to be visited was taken to be the same as the target's generation order. All targets started moving at 3m/s ($\|v_t\||_{t=0} = 3$m/s) with random heading ($\psi_t|_{t=0}$) according to a uniform distribution in the interval $]-\pi, \pi[$. Both $\dot{v}_t$ and $\dot{\psi}_t$ were defined as stochastic signals with a normal distribution with a predefined mean and standard deviation, namely

$$\|\dot{v}_t\| \sim \mathcal{N}(0, 0.05)$$
$$\dot{\psi}_t \sim \mathcal{N}(0, 0.03)$$

and the target's speed $\|v_t\|$ was limited to the interval $[0, 8]$ [m/s]. Each simulation lasted 500 seconds and the UAV constant airspeed was chosen to be $\|V_0\| = 25$m/s. The performance metrics values presented in Table III for the proposed method were computed through

$$\frac{1}{500} \sum_{i=1}^{500} \sum_{j=1}^{n_i} \frac{1}{n_i} \frac{t_{j|i}}{t_{opt_{j|i}}}$$

where n_i is the total number of target interceptions at simulation i, $t_{j|i}$ is the j^{th} interception time at simulation i and $t_{opt_{j|i}}$ is the corresponding optimal time for interception obtained from a posteriori computations.

TABLE III
PROPOSED METHODS PERFORMANCE METRICS.

	Algorithm 1	Algorithm 1*
$(\%)t_{opt}$	99.49	99.80

From Table III one can verify that, as previously argued, Algorithm 1* improves Algorithm's 1 performance. Finally,

since the proposed MPF algorithm behaves similarly to a missile guidance algorithm (considering multiple targets tracking applications), it is possible to arguably infer that the vehicle using the proposed method for multiple targets tracking applications will intercept the current target with a performance similar to the classical missile guidance algorithms described in [25].

REFERENCES

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

[2] A. Aguiar and J. Hespanha, "Trajectory-tracking and path-following of underactuated autonomous vehicles with parametric modeling uncertainty," *IEEE Transactions on Automatic Control*, vol. 52, pp. 1362–1379, 2007.

[3] C. Canudas de Wit, H. Khennouf, C. Samson, and O. Sordalen, "Nonlinear control design for mobile robots," *Recent Trends in Mobile Robots*, vol. 11, pp. 121–156, 1993.

[4] L. Bishop, "There is more than one way to frame a curve," in *Amer. Math*. Monthly 82, March 1975, pp. 246–251.

[5] I. Kaminer, O. Yakimenko, A. Pascoal, and R. Ghabcheloo, "Path generation, path following and coordinated control for time-critical missions of multiple UAVs," in *Proceedings of the 2006 American Control Conference*, 2006, pp. 4906–4913.

[6] P. Encarnação and A. Pascoal, "3D path following for autonomous underwater vehicles," in *Proceedings of the 39th Conference on Decision and Control*, 2000, pp. 2977 – 2982.

[7] I. Kaminer, A. Pascoal, C. Cao, and V. Dobrokhodov, "Path following for unmanned aerial vehicles using L1 adaptive augmentation of commercial autopilots," *Journal of Guidance, Control and Dynamics*, vol. 33, pp. 550–564, 2010.

[8] E. Frew, D. Lawrence, C. Dixon, J. Elston, and J. Pisano, "Lyapunov guidance vector fields for unmanned aircraft applications," in *Proceedings of the 2007 American Control Conference*, 2007, pp. 371–376.

[9] R. Wise and R. Rysdyk, "UAV coordination for autonomous target tracking," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2006.

[10] T. Oliveira and P. Encarnação, "Ground target tracking control system for unmanned aerial vehicles," *Journal of Intelligent and Robotic Systems*, vol. 69, pp. 373–387, 2013.

[11] M. Zhang and H. Liu, "Game-theoretical persistent tracking of a moving target using a unicycle-type mobile vehicle," *IEEE Transactions on Industrial Electronics*, vol. 61, pp. 6222–6233, 2014.

[12] H. Dias, P. Calado, R. Bencatel, R. Gomes, S. Ferreira, and J. Sousa, "Operations with multiple unmanned systems," in *International Conference on Intelligent Robots and Systems (IROS)*, 2012.

[13] O. Dobrokhodov, I. Kaminer, K. Jones, and R. Ghabcheloo, "Vision-based tracking and motion estimation for moving targets using small UAVs," in *Proceedings of the AIAA Guidance Navigation and Control Conference and Exibit*, 2006.

[14] W. Ren and R. Beard, "Trajectory tracking for unmanned air vehicles with velocity and heading rate constraints," *IEEE Transactions on Control Systems Technology*, vol. 12, pp. 706–716, 2004.

[15] J. Lee, R. Huang, A. Vaughn, X. Xiao, K. Hedrick, M. Zennaro, and R. Sengupta, "Strategies of path-planning for a UAV to track a ground vehicle," in *Proceedings of the 2nd annual Autonomous Intelligent Networks and Systems Conference*, 2003.

[16] S. Spry, A. Vaughn, and X. Xiao, "A vehicle following methodology for UAV formations," in *Proceedings of the 4th International Conference on Cooperative Control and Optimization*, 2003.

[17] H. Chen and K. Chang, "Tracking with UAV using tangent-plus-Lyapunov vector field guidance," in *Proceedings of the 12th International Conference on Information Fusion*, 2009.

[18] H. Oh, S. Kim, H. Shin, B. White, A. Tsourdos, and C. Rabbath, "Rendezvous and standoff target tracking guidance using differential geometry," *Journal of Intelligent and Robotic Systems*, vol. 69, pp. 389–405, 2013.

[19] T. Oliveira and P. Encarnação, "Ground target tracking for unmanned aerial vehicles," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, 2010.

[20] X. Ding, A. Rahmani, and M. Egerstedt, "Multi-UAV convoy protection: an optimal approach to path planning and coordination," *IEEE Transactions on Robotics*, vol. 26, pp. 256–268, 2010.

[21] S. Smith, M. Schwager, and D. Rus, "Persistent robotic tasks: Monitoring and sweeping in changing environments," *IEEE Transactions on Robotics*, vol. 28, pp. 410–426, 2012.

[22] S. Rathinam, P. Almeida, Z. Kim, S. Jackson, A. Tinka, W. Grossman, and R. Sengupta, "Autonomous searching and tracking of a river using an UAV," in *Proceedings of the 2007 American Control Conference*, 2007.

[23] K. Savla, E. Frazzoli, and F. Bullo, "Traveling salesperson problems for the Dubins vehicle," *IEEE Transactions on Automatic Control*, vol. 53, pp. 1378–1391, 2008.

[24] Z. Tang and U. Özgüner, "Motion planning for multitarget surveillance with mobile sensor agents," *IEEE Transactions on Robotics*, vol. 21, pp. 898–908, 2005.

[25] F. Belkhouche, B. Belkhouche, and P. Rastgoufard, "Line of sight robot navigation toward a moving goal," *IEEE Transactions on System, Man, and Cybernetics*, vol. 36, pp. 255–267, 2006.

[26] F. Morbidi and G. Mariottini, "Active target tracking and cooperative localization for teams of aerial vehicles," *IEEE Transactions Control Systems Technology*, vol. 21, pp. 1694–1707, 2013.

[27] L. Freda and G. Oriolo, "Vision-based interception of a moving target with a nonholonomic mobile robot," *Robotics and Autonomous Systems*, vol. 55, pp. 419–432, 2007.

[28] M. Breivik and T. Fossen, "Guidance laws for planar motion control," in *IEEE Conference on Decision and Control*, 2008.

[29] M. Breivik, V. Hovstein, and T. Fossen, "Straight-line target tracking for unmanned surface vehicles," *Modeling, Identification and Control*, vol. 29, no. 4, pp. 131–149, 2008.

[30] J. Swee, "Missile terminal guidance and control against evasive targets," Master's thesis, Naval Posgraduate School, 2000.

[31] T. Oliveira, P. Encarnação, and A. Aguiar, "Moving path following for autonomous robotic vehicles," in *Proceedings of the European Control Conference (ECC)*, 2013.

[32] C. Samson, "Path following and time-varying feedback stabilization of wheeled mobile robots," in *Proceedings of the International Conference on Advanced Robotics and Computer Vision*, vol. 13, 1992.

[33] A. Rucco, A. Aguiar, and J. Hauser, "Trajectory optimization for constrained UAVs: a virtual target vehicle approach," in *Proceedings of the International Conference on Unmanned Aircraft Systems (ICUAS)*, 2015.

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

[35] H. Khalil, *Nonlinear Systems*, 3rd ed., 2002, Prentice Hall.

[36] *Piccolo Hardware-in-Loop / Software-in-Loop Setup Guide*, Cloud Cap Technology, 2008.

[37] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. B. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: An open-source robot operating system," in *ICRA Workshop on Open Source Software*, 2009.

[38] A. Aguiar, L. Cremean, and J. Hespanha, "Position tracking for a nonlinear underactuated hovercraft: Controller design and experimental results," in *Proceedings of 42nd IEEE Conference on Decision and Control*, 2003.

Tiago Oliveira is a Captain of the Portuguese Air Force. He is a lecturer at the Portuguese Air Force Academy and a researcher at the Portuguese Air Force Research, Development and Innovation Center, working in the area of nonlinear control oriented to Unmanned Aerial Vehicles (UAV) applications. He has received the M.Sc. degree in Electrical and Computer Engineering in 2009 from the Portuguese Air Force Academy and is currently a Ph.D. candidate at Faculty of Engineering, University of Porto, Portugal. His research interests include nonlinear robust guidance control and collaborative control of autonomous robotic vehicles.

A. Pedro Aguiar received the Licenciatura, M.S. and Ph.D. in Electrical and Computer Engineering from the Instituto Superior Técnico (IST), University of Lisbon, Portugal in 1994, 1998 and 2002, respectively. Currently, Dr. Aguiar holds an Associate Professor position with the Department of Electrical and Computer Engineering (DEEC), Faculty of Engineering, University of Porto (FEUP). From 2002 to 2005, he was a post-doctoral researcher at the Center for Control, Dynamical-Systems, and Computation at the University of California, Santa Barbara (UCSB). From 2005 to 2012, he was a senior researcher with the Institute for Systems and Robotics at IST (ISR/IST), and an invited assistant professor with the Department of Electrical and Computer Engineering, IST. His research interests include modeling, control, navigation, and guidance of autonomous robotic vehicles, nonlinear control, switched and hybrid systems, tracking, path-following, performance limitations, nonlinear observers, optimization, networked control, and coordinated/cooperative control of multiple autonomous robotic vehicles.

Pedro Encarnação is an Adjunct Assistant Professor at UCP - Católica Lisbon School of Business & Economics. He has received the "Licenciatura" degree in 1994, the M.Sc. degree in 1998, and the Ph.D. degree in 2002, all in Electrical and Computer Engineering from the Instituto Superior Técnico, Technical University of Lisbon, Portugal. His current research interests include Rehabilitation Engineering, namely the use of robotic systems in the rehabilitation of children with disabilities, and nonlinear control of autonomous vehicles.