

**FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO**

# **Moving Path Following Control System for Fixed-Wing Unmanned Aerial Vehicles**

**Tiago Miguel Monteiro de Oliveira**



Programa Doutoral em Engenharia Electrotécnica e de Computadores

Supervisor: Professor António Pedro Aguiar

Co-Supervisor: Professor Pedro Encarnação

April 21, 2017



# **Moving Path Following Control System for Fixed-Wing Unmanned Aerial Vehicles**

**Tiago Miguel Monteiro de Oliveira**

Programa Doutoral em Engenharia Electrotécnica e de Computadores

April 21, 2017



# Resumo

Esta tese aborda o problema de *Moving Path Following* (MPF) em que um veículo deverá convergir e seguir um determinado caminho geométrico desejado que poderá estar a mover-se de acordo uma especificação em velocidade linear e angular, generalizando desta forma o problema clássico de seguimento de caminhos estacionários. A dinâmica do erro associado é deduzida para o caso geral do seguimento de caminhos que se movem num espaço tridimensional, por veículos aéreos não-tripulados (UAV) de asa fixa. De seguida, diferentes cenários de aplicação para o método MPF são considerados.

Estuda-se primeiro o problema de seguimento de um e de múltiplos alvos no solo, assumindo que o UAV voa a altitude constante. Para essa missão, propõe-se um algoritmo de geração de caminhos juntamente com uma lei de controlo para MPF no espaço bidimensional baseada nos métodos de Lyapunov. São apresentadas garantias formais de convergência, uma métrica de desempenho e resultados dos ensaios em voo.

Aborda-se depois o problema da proteção aérea de colunas de veículos terrestres, utilizando um UAV com uma câmara instalada a bordo cujo campo de visão é inferior ao raio de curvatura mínimo do UAV. A lei de controlo deduzida para o primeiro cenário é utilizada em conjunto com uma estratégia de orientação de um caminho geométrico solidário com o centro da coluna de veículos terrestres, de modo a permitir a um UAV de asa fixa convergir e seguir esse caminho. São apresentadas as condições sob as quais a estratégia proposta resolve o problema de proteção da coluna terrestre. A métrica de desempenho proposta, juntamente com os resultados da simulação numérica, demonstram a eficácia da abordagem adoptada.

Finalmente, é deduzida a nível cinemático uma lei de controlo para o caso geral do problema de MPF a três dimensões, tendo por base a utilização de quaterniões, incluindo-se garantias formais de convergência. A solução proposta é validada por resultados numéricos e de ensaios em voo da sua aplicação à aterragem autónoma num navio em movimento.

Os ensaios em voo realizados no âmbito desta tese tiveram lugar na base aérea da Ota e no aeródromo de Santa Cruz, em Portugal, utilizando o UAV ANTEX-X02.



# Abstract

This thesis introduces the Moving Path Following (MPF) problem where a vehicle is required to converge to and follow a desired geometric path, that moves according to some specified linear and angular velocity, thus generalizing the classical problem of following stationary paths. In the proposed approach to solve the problem, a MPF error dynamical system for the general case of time varying paths in a three-dimensional space is derived for a fixed-wing Unmanned Aerial Vehicle (UAV). Subsequently, distinct application scenarios for the MPF method are considered.

First, the problem of tracking single and multiple targets on the ground using a UAV flying at constant altitude is addressed. To this end, a path generation algorithm is proposed together with a two-dimensional Lyapunov based MPF control law. Formal convergence guarantees, a performance metric and flight tests results are presented.

A second application studied is the problem of convoy protection missions using a fixed-wing UAV with a camera onboard whose field of view is smaller than the UAV minimum turning radius. The previously derived two-dimensional MPF control law is used together with a guidance algorithmic strategy in order to make a fixed-wing UAV converge to and follow a desired geometric moving path that is attached to the convoy center. Conditions under which the proposed strategy solves the convoy protection problem are derived. A performance metric and numerical simulation results demonstrate the effectiveness of the proposed approach.

Finally, a general three-dimensional MPF quaternion-based control law is derived at kinematic level. Formal convergence proofs are provided. The proposed method is validated through numerical and flight test results, considering the mission scenario of autonomous landing on a moving vessel.

The flight tests presented here took place in the Ota air base and the Santa Cruz airfield, in Portugal, using the ANTEX-X02 UAV.





# Acknowledgements

Words cannot express my immense gratitude to the people who have helped me to accomplish the achievement of writing this thesis. I am deeply grateful...

to my supervisors, Professors Pedro Aguiar and Pedro Encarnação, for their insightful comments and thorough suggestions which were crucial to the successful conclusion of this thesis. I am also grateful for their continual availability, patience and friendship, that always made me feel very well supported. Without their permanent support, encouragement and guidance, this thesis would not have been possible;

to the chiefs of the Portuguese Air Force Research group (CIDIFA), Colonel José Morgado, Lieutenant-Colonel Maria Nunes and Major Aurélio Santos, for their confidence in my work;

to Major Carlos Silva and all the Portuguese Air Force flight operations members, namely Sergeants Carlos Bandeiras, Fernando Santos, Paulo Teixeira, Joaquim Gomes, Paulo Mendes and Jorge Fernandes, without whom it would not have been possible to perform the flight tests;

to Major Paula Gonçalves, that given the common challenges we had to face, was always an example of abnegation and perseverance in the arduous path to obtain a Ph.D. degree;

to Major Elói Pereira, for the preliminary revision of this thesis and for sharing his experience as a senior researcher, which allowed me to broaden horizons and put things into perspective;

to Captain Gonçalo Cruz, for being always there when I needed. His friendship, loyalty and support in every aspect of our daily work were essential to achieve this goal;

to Captain Francisco Machado, for sharing the responsibility of teaching Avionic Systems at the Portuguese Air Force Academy, and also for all the disconcerting and useless debates;

to Pedro Marques da Silva and Lieutenant Diogo Silva, for their crucial contributions on the software tools that were essential for the execution of the control algorithms during the flight tests;

to all the members of PITVANT and SEAGULL research projects, in particular to the former Underwater Systems and Technology Laboratory (LSTS) members, Ricardo Bencatel, Carlos Eduardo and Eduardo Marques, for the introduction to the Piccolo autopilot and for the encouragement to proceed with this research work while it was still in its early stage;

to the Portuguese Air Force Academy, where most of this research work took place, for providing all the necessary means that allowed me to be present in several conferences and work meetings;

to the administrative staff of the Faculty of Engineering, University of Porto (FEUP), in particular to José António and Zélia Prior, for their availability in dealing with the non-standard procedures that were needed to include me as a Ph.D. student at FEUP;

to my mother and sister, for showing me an holistic view of life, which helped me through the hard times;

to my wife, Maria Esperança, for her unconditional love and patience, for enduring the necessary absences, and for giving me the best times of my life;

to my children, Helena and Francisco, that were born during the course of this research work and unknowingly gave me a tremendous motivation to conclude it;

to all my friends and family, for their unconditional support, love and education.

To my wife and children, Esperança, Helena and Francisco



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Moving Path Following . . . . .	2
1.2	Thesis contribution . . . . .	7
1.3	Thesis organization . . . . .	8
<b>2</b>	<b>Error space for Moving Path Following</b>	<b>11</b>
2.1	Position error kinematics . . . . .	11
2.2	Orientation error kinematics . . . . .	13
<b>3</b>	<b>2D Moving Path Following control design</b>	<b>17</b>
3.1	2D error space using Euler angles . . . . .	17
3.2	2D Moving Path Following control law . . . . .	19
3.3	Application to single and multiple target tracking problems . . . . .	26
3.3.1	Single target tracking problem formulation as MPF . . . . .	26
3.3.2	Flight test results: Single target tracking . . . . .	28
3.3.3	Multiple targets tracking problem formulation as MPF . . . . .	29
3.3.4	Flight test results: Multiple targets tracking . . . . .	34
3.4	Application to convoy protection problems . . . . .	38
3.4.1	Convoy protection problem formulation . . . . .	38
3.4.2	Convoy protection using MPF . . . . .	39
3.4.3	Numerical simulation results . . . . .	45
<b>4</b>	<b>3D Moving Path Following control design</b>	<b>53</b>
4.1	3D Moving Path Following control law using quaternions . . . . .	53
4.2	Application to following a lemniscate moving in three dimensions . . . . .	56
4.3	Application to autonomous landing on a moving vessel . . . . .	60
<b>5</b>	<b>Conclusions and future work</b>	<b>65</b>
	<b>Appendix A Flight tests architecture</b>	<b>67</b>
	<b>Appendix B Controller implementation details</b>	<b>71</b>
B.1	Path reference frames . . . . .	71
B.1.1	Serret-Frenet frame . . . . .	71
B.1.2	Parallel-transport frame . . . . .	72
B.2	Reference path's parametric equations . . . . .	73
B.3	Position errors . . . . .	74
B.4	Orientation errors . . . . .	75

<b>Appendix C Multiple targets tracking implementation details</b>	<b>77</b>
C.1 Path generation details . . . . .	77
C.2 Algorithm 2 detailed implementation and corresponding “virtual target” position estimation . . . . .	80
C.3 Convergence conditions . . . . .	82
C.4 Performance metric . . . . .	84
C.5 Overall control architecture . . . . .	85
<b>Appendix D Quaternion fundamentals</b>	<b>87</b>
<b>Appendix E Lyapunov stability fundamentals</b>	<b>91</b>
E.1 Stability definitions . . . . .	91
E.2 Lyapunov’s direct method . . . . .	92
E.3 Linear time-varying systems . . . . .	93
E.4 Linearization . . . . .	94
E.5 Barbalat’s Lemma . . . . .	94
<b>References</b>	<b>95</b>

# List of Figures

1.1	Error space frames, illustrating for the case of an UAV. . . . .	3
3.1	Moving Path Following: Error space frames and relevant variables, illustrating the case of a UAV. . . . .	18
3.2	Relationship between the groundspeed vector, the airspeed vector and the wind vector. . . . .	21
3.3	2D MPF numerical simulation example: UAV following a moving straight line that is rotating around a fixed point $p_0 = [0 \ 0]$ with an angular velocity $\omega_{d_e} = 0.025\text{rad/s}$ . . . . .	25
3.4	Single target tracking numerical simulation: Aircraft's trajectory following a target between $t=0\text{s}$ and $t=200\text{s}$ . . . . .	27
3.5	Single target tracking numerical simulation: Position and heading errors. . . . .	27
3.6	Single target tracking flight test: Aircraft's trajectory following a target on the ground (from $t = 0\text{s}$ to $t = 250\text{s}$ ). . . . .	28
3.7	Single target tracking flight test: Commanded and actual bank. . . . .	28
3.8	Single target tracking flight test: Distance and heading errors. . . . .	29
3.9	Dubins path: RS or LS. . . . .	30
3.10	Dubin's path relevant parameters for moving targets, illustrating for the right circular segment case (a), followed by a straight line section (b). . . . .	31
3.11	Multiple targets tracking numerical simulation: An autonomous vehicle tracks three targets heading North at constant speed using Algorithm 1 and the MPF control law given by equation (3.12). . . . .	33
3.12	Multiple targets tracking numerical simulation: Optimal trajectory generation detail. . . . .	34
3.13	Illustration of a UAV intercepting a moving target, heading towards its current estimated interception position. . . . .	34
3.14	Multiple targets tracking flight test: Aircraft's trajectory following multiple targets on the ground. . . . .	35
3.15	Multiple targets tracking flight test: Distance and heading errors. . . . .	36
3.16	Multiple targets tracking flight test: Commanded and actual bank. . . . .	36
3.17	Multiple targets tracking flight test: Actual distance between UAV and current target at time of interception. . . . .	37
3.18	Multiple targets tracking flight test: Communications latency. . . . .	37
3.19	Convoy protection problem formulation illustration. . . . .	39
3.20	Convoy protection using the MPF method: Relevant variables. . . . .	40
3.21	Minimum feasible turning radius $r_c/r_{min}$ for a UAV to follow a lemniscate path contained within a circle of radius $r_c$ , considering: a) $\frac{v_d}{V} = 0.5$ and b) $\frac{v_d}{V} = 1$ . . .	41

3.22	UAV trajectory following a moving lemniscate path with $\frac{v_d}{V} = 0.9$ , $\psi_d = 0$ and $\psi_p = 0$ without imposing any kinematic constraint on the UAV. . . . .	42
3.23	UAV trajectory following a moving lemniscate path with $\frac{v_d}{V} = 0.9$ , $\psi_d = 0$ using Algorithm 3 to control the desired geometric path angular velocity $\dot{\psi}_p = \omega_d$ and orientation angle $\psi_p$ . . . . .	43
3.24	Minimum feasible turning radius $r_c/r_{min}$ for a UAV to follow a circular path considering: a) $\frac{v_d}{V} = 0.5$ and b) $\frac{v_d}{V} = 1$ . . . . .	45
3.25	Lower bound for $\frac{v_d}{V}$ as a function of $\frac{r_c}{r_{min}}$ for the proposed method and the considered benchmark. . . . .	46
3.26	Convoy protection numerical simulation: UAV trajectory following a convoy center (the target in the figure) moving with time-varying linear and angular velocities. . . . .	47
3.27	Convoy protection numerical simulation: UAV's angular velocity. . . . .	48
3.28	Convoy protection numerical simulation: Desired path angular velocity $\omega_d$ and orientation angle $\psi_p$ obtained using Algorithm 3. . . . .	48
3.29	Convoy protection numerical simulation: Position and heading errors. . . . .	49
3.30	Convoy protection performance metric simulation: UAV and convoy center trajectories. . . . .	50
3.31	Convoy protection performance metric simulation: Convoy heading and speed. . . . .	51
4.1	Lemniscate tracking numerical simulation: Path frame origin trajectory (red), desired lemniscate path (blue) and UAV position at sample time instants. . . . .	57
4.2	Lemniscate tracking numerical simulation: UAV trajectory. . . . .	57
4.3	Lemniscate tracking numerical simulation: UAV trajectory as seen by an observer rigidly attached to the path frame origin. . . . .	57
4.4	Lemniscate tracking numerical simulation: Position and orientation errors. . . . .	58
4.5	Lemniscate tracking flight test: UAV's trajectory following a moving lemniscate path, as seen by an observer attached to the inertial frame origin. . . . .	59
4.6	Lemniscate tracking flight test: UAV's trajectory following a moving lemniscate path, as seen by an observer rigidly attached to the path frame origin. . . . .	59
4.7	Lemniscate tracking flight test: UAV's reference commands. . . . .	60
4.8	Lemniscate tracking flight test: UAV's position and orientation errors. . . . .	60
4.9	Autonomous landing flight test: UAV and vessel trajectories. . . . .	61
4.10	Autonomous landing flight test: Simulated vessel trajectory detail. . . . .	61
4.11	Autonomous landing flight test: Position errors. . . . .	62
4.12	Autonomous landing flight test: Orientation errors. . . . .	63
4.13	Autonomous landing flight test: UAV bank and vertical rate reference commands sent to the UAV from the ground control station. . . . .	63
4.14	Autonomous landing flight test: Wind velocity estimate, provided by the autopilot. . . . .	63
4.15	Autonomous landing flight test: UAV's air and ground speed. . . . .	64
4.16	Autonomous landing flight test: Position errors for the five flight tests performed. . . . .	64
A.1	Flight tests framework: a) Portuguese Air Force research facility at the Ota air base; b) ANTEX-X02 UAV; c) Ground control station at the Santa Cruz airfield . . . . .	67
A.2	Flight test operation diagram. . . . .	68
C.1	RS (solid line) and LS (dashed line) Dubins path parameters. . . . .	78
C.2	Angular distance between angles $\eta_s$ and $\eta_e$ for clockwise ( $c_r$ ) and counter clockwise ( $c_l$ ) circles. . . . .	79



C.3	Multiple targets tracking numerical simulation: Increase in the actual time for interception with respect to the optimal time of interception, as a percentage of the optimal time. . . . .	84
C.4	Control system architecture. . . . .	86
D.1	Illustration of Euler's principal theorem. . . . .	88



# List of Tables

3.1	Single target tracking numerical simulation: Controller parameters. . . . .	26
3.2	Proposed method performance metrics. . . . .	51
A.1	ANTEX-X02 main features. . . . .	68
B.1	Reference path's parametric equations. . . . .	73
C.1	Proposed methods performance metrics. . . . .	85



# List of Acronyms

CB	Constant Bearing
GPS	Global Positioning System
LOS	Line of Sight
LQR	Linear Quadratic Regulator
MPF	Moving Path Following
NED	North-East-Down
PCC	Piccolo Command Center
PP	Pure Pursuit
ROS	Robot Operating System
RTK	Real-Time Kinematic
UAV	Unmanned Aerial Vehicle



# Chapter 1

## Introduction

The past two decades have witnessed an enormous increase in the utilization of unmanned robotic vehicles by some of the largest armed forces in the world and governmental organizations [1]. Additionally, fostered by the increasing amount of low-cost sensors, airframes and open software/hardware frameworks available for the general public (strongly driven by the mobile phones industry), the use of robotic vehicles has experienced an exponential growth amongst hobbyists, academic researchers, and industries in the past recent years [1]. In particular, given their potential to perform dull, dirty and dangerous missions, single and/or multiple unmanned aerial vehicles (UAV) have been employed in different mission scenarios such as target search [2], area surveillance [3], mapping [4], perimeter patrol [5], convoy protection [6], target tracking [7] or environmental monitoring [8], just to name a few. Both from a theoretical and from a practical standpoints, these scenarios raise interesting and challenging control problems such as the development of robust control and navigation systems for (optimal) path-planning [9] obstacle and/or collision avoidance [10], motion control of single and/or multiple vehicles in presence of uncertainty [11], nonlinear dynamics [12], partial noisy state measurements [13], disturbances [14], and limited or disrupted communications [15]. Trajectory tracking and path following are among the basic motion control tasks that autonomous aircraft are required to execute, in order to perform the above mentioned missions.

In trajectory tracking, the vehicle follows a given trajectory with time constraints, and thus, trajectory tracking controllers require the vehicle to track a time-parametrized trajectory by commanding its forward speed and orientation [11]. In classical path following, the vehicle is required to converge to and follow a given desired geometric path. To this end, in some approaches, the path following controller computes the distance between the vehicle and the point on the path closest to the vehicle, together with the error between the vehicle's orientation and the desired one (given by the tangent to the path at that point), and tries to reduce both to zero [16]. If there are no time constraints, the vehicle can move with constant speed and achieve smoother convergence to the path [11]. The performance of path-following and trajectory-tracking controllers is analyzed in [17, 18], demonstrating that the task of following a geometric path is less restrictive than the task of tracking a reference signal. Moreover, since the linear position errors computed by a path

following controller are always less than or equal to those involved in trajectory tracking control, the actuation signals are less likely to become saturated [19]. Time constraints for path following methods are used to set a desired speed for a vehicle, acting as a “virtual target” moving along the path [20]. This method solves the problem of singularities that can arise when the vehicle’s desired position is defined as the projection of the real vehicle on the desired path. See also the recent work in [21, 22] on trajectory optimization.

For wheeled mobile vehicles, the first works on path following have been presented in a series of groundbreaking papers by Samson et al. (see for example [23] and the references therein). Since then, several types of control-theoretic techniques have been applied to path following problems for air vehicles [24, 25, 26], underwater and surface maritime vehicles [16, 27, 28, 29] and land vehicles [30, 31]. Some of the well documented techniques are sliding mode control [32, 33, 34], model predictive control [35, 36], back stepping control [37, 38], linear quadratic regulator (LQR) [39, 40], gain scheduling theory [41], adaptive control [42, 43] and dynamic programming [44].

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 thesis, 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, this research work presents 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. The desired geometric path is written with respect to a (so called) path frame that, in general, can move with time-varying linear and angular velocities in relation to an inertial frame. This provides a natural extension of the classical path following methods for stationary paths [16, 24, 25] retaining by design its desirable characteristics, namely smooth convergence to the moving path and the possibility of doing so at constant speed with respect to an inertial coordinate frame.

## 1.1 Moving Path Following

Aiming at formally defining the Moving Path Following problem, consider a local inertial frame  $\{I\} = \{\vec{x}, \vec{y}, \vec{z}\}$  with the  $\vec{x}$  axis pointing North,  $\vec{y}$  East and  $\vec{z}$  Down (this definition is typically referred to as the North-East-Down (NED) with x-North, y-East, and z-Down). Consider also a path frame  $\{P\} = \{\vec{x}_P, \vec{y}_P, \vec{z}_P\}$  and let the origin of  $\{P\}$  expressed in  $\{I\}$  be denoted by  $p_0$ . Let  ${}^P p_d(\ell) = \begin{bmatrix} {}^P p_{d_x}(\ell) & {}^P p_{d_y}(\ell) & {}^P p_{d_z}(\ell) \end{bmatrix}^T$  be a desired planar geometric path parametrized by  $\ell$ , 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 frame. Additionally, let  $v_d = \dot{p}_0 = \begin{bmatrix} v_{d_x} & v_{d_y} & v_{d_z} \end{bmatrix}^T$



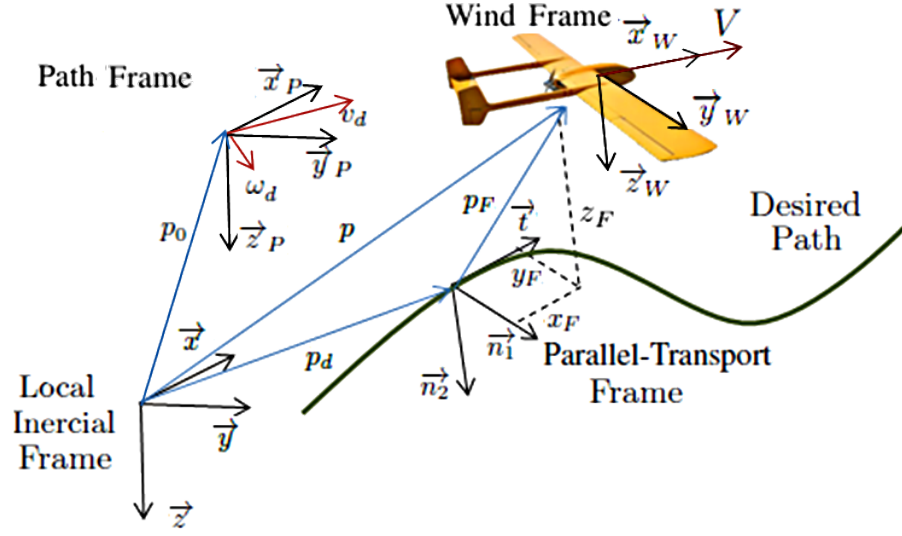


Figure 1.1: Error space frames, illustrating for the case of an UAV.

and  $\omega_d = \begin{bmatrix} \omega_{d_x} & \omega_{d_y} & \omega_{d_z} \end{bmatrix}^T$  be the corresponding linear and angular velocities of the path frame  $\{P\}$ , respectively, expressed in  $\{I\}$ . In practice, the path 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  $\omega_d$  are different from zero. Furthermore, consider the general case where  $v_d$  and  $\omega_d$  can be time varying, i.e., they are explicitly functions of time.

The MPF problem can thus be formulated as follows:

*Given a robotic vehicle moving at a given speed  $V$  and a desired moving path  $\mathcal{P}_d = ({}^P p_d(\ell), p_0, v_d, \omega_d)$ , design a control law that steers and keeps the vehicle on the desired path  $\mathcal{P}_d$ .*

In this thesis, a solution (and conditions under which this is possible) to the two-dimensional MPF problem applied to tracking single and multiple targets on the ground and to convoy protection is presented, considering that the UAV flies at constant altitude. Then, the MPF problem is solved for the general three-dimensional space with an application to the autonomous landing of a UAV on a simulated moving vessel. Other possible applications of the MPF method include air vehicles tracking, thermals soaring [45], gas clouds monitoring (where the time-varying thermal/gas cloud center coordinates may specify the motion of the desired path). The MPF control laws are derived using Lyapunov methods and formal convergence proofs along with numerical and/or flight test results are presented.

The following paragraphs discuss the different approaches present in the literature to the mission scenarios discussed in this thesis.

For single ground target tracking, Lee et al. [46] and Spry et al. [47] describe a controller that switches between two modes according to the relation between the UAV and the target velocities. Dobrokhodov et al. [7] propose 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 [24]. 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 [48, 49]. Chen et al. [49] propose 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 [24]. More recently, Oh et al. [50] propose 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 is 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 [48, 49] (provided that the UAV's ground speed is higher than the target speed). If the target speed is close to the UAV speed, the MPF 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 Section 3.3). When the ground target moves slower than the UAV, the MPF control law makes the UAV to loiter above the target. Additionally, in contrast with the methods proposed in [49, 50], 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 presented in this thesis is not limited to a standoff circle centered at the target position (unlike most of the proposed methods in the literature - see for example [48, 24]) 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 [6, 51]. In order to illustrate these features, in the single target tracking scenario presented in Section 3.3, a fixed wing UAV is required to follow a lemniscate path [52] centered at the actual target position whose angular velocity is the same as the target, keeping the UAV altitude and airspeed constants.

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 [53, 54] and moving targets [55, 56, 57] can be found in the literature. For most interception tasks, the motion of the target is usually unknown in advance. Thus, visual feedback [58] and line-of-sight methods [56], [59] 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 [28, 60] 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 [28]; 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 [28]; 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 [28]. 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 is presented in this thesis. In practice, the proposed path generation algorithm makes the vehicle to track a moving Dubins path [61] (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 thesis 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, explicitly taking into account the UAV's kinematic constraints. Additionally, it is demonstrated 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 presented.

Convoy protection missions using Unmanned Aerial Vehicles (UAV) are an active area of research for both civilian and military applications [1]. In order to perform such tasks, the UAVs are typically equipped with electro-optical sensors with a given resolution and thus, the most suitable trajectory for the UAV depends (among other things) on the required level of image resolution. When missions are performed during long periods of time or when it is expected that the UAV travels long distances, fixed-wing UAVs are typically employed [62]. The work described in [6] addresses the problem of controlling a group of fixed-wing UAVs to provide convoy protection to a group of ground vehicles, where the radius of the circular region of interest (which is determined by the optical sensor's resolution carried by the UAVs) is smaller than the UAVs minimum turning radius. For the case of a single UAV and when the ground convoy of vehicles is restricted to be stationary or moving in straight lines at constant speed, the authors analyse what is the best UAV path in the sense that it maximizes the longest time that the UAV is inside the convoy circular region of interest, and provide a lower bound on the convoy speed (that depends on the ratio between the radius of the convoy circular region of interest and the UAV minimum turning radius) that guarantees continuous convoy protection at all times (i.e., the UAV will always be inside the convoy circular region of interest, despite the UAV's kinematic constraints that are modeled as Dubins vehicles). Notice, however, that these results only apply for a very restricted convoy trajectory case. For a more general target tracking/convoy protection missions, the typical adopted strategy reported in the literature is to make the fixed-wing UAV to be at a given standoff distance from the target/convoy center [7, 24, 49]. In alternative, in [46, 63], lateral and longitudinal orbits (depending on the convoy speed) are proposed in order to perform convoy protection mis-

sions. The method chooses the most adequate desired path depending on the speed ratio between the UAV and the convoy. However, none of these strategies encompasses the case presented in [6] where the radius of the convoy circular region of interest is smaller than the UAV minimum turning radius, and often do not take into account the UAV's kinematic constraints [7].

The problem of convoy protection guaranteeing a continuous time coverage and without the restrictions on the convoy movements considered in [6] is addressed in this thesis. This problem is solved using the MPF method, where a fixed-wing UAV moving at constant ground speed is required to converge to and follow a desired geometric path (contained within the circular region of interest) that is attached to the target/convoy center. Concretely, the proposed strategy is to command the desired path's angular velocity such that the UAV's resulting trajectory complies with the UAV's physical constraints.

In recent years, different methods have been used to address the autonomous landing of UAVs on moving vehicles [64, 65, 66, 67]. In [64], a low-cost real-time kinematic global positioning system (RTK-GPS) is used together with a decoupled longitudinal and lateral guidance strategy using conventional Proportional-Integral-Derivative (PID) and nonlinear controllers respectively. A similar approach is considered in [65] using a differential GPS system. In [66], a fixed-wing UAV is required to land on top of a moving car using decoupled linear controllers to achieve cooperation between the moving car and the UAV. Vision-based systems or a combination of both vision and guidance systems were also proposed to perform an autonomous landing on a moving vehicle [68, 69, 70]. For the particular case of ship landing missions, the heave, roll and yaw motions of a ship can be predicted using artificial neural networks for increased system's performance (see [71] and the references therein). The strategy proposed in this thesis is to attach a desired landing pattern to the moving vessel and make the UAV converge to and track the moving landing pattern. It was assumed that the vessel was equipped with a RTK-GPS and relayed its position and velocity in real time to the UAV.

A common approach with UAV control is to assume that the vehicle has an off-the-shelf inner loop controller (autopilot) 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 [7, 72]. Outer loop control laws are thus derived using a kinematic model of the vehicle that provide the references to the inner control loop. The same approach is adopted in this research work.

The results presented in this thesis were published in a series of papers. The concept of following a path that moves with time-varying linear velocity with respect to an inertial frame (attached to a moving reference target) was presented in a preliminary form in [73, 74]. The MPF problem was formally introduced in [75] for fixed-wing unmanned aircraft vehicles adding an additional degree of freedom to the moving path, allowing it to also move with time-varying angular velocity with respect to an inertial frame. This additional degree of freedom has been explored to perform convoy protection missions in [76] and, combined with a new path-generation algorithm, to address single and multiple targets tracking problems in [77]. In those publications, the proposed MPF method explicitly takes into account the wind estimate (provided by the inner loop

controller) and it is assumed that the UAV flies at constant altitude and airspeed. Using the same ideas presented in [73, 74], a virtual target based path-following guidance system for the execution of coordinated manoeuvres between two marine vehicles using the concept of moving reference paths was presented in [78]. The proposed method considered a similar error space derivation and the same desired geometric paths as the ones considered in [73, 74]. The key difference is that the approach proposed in [78] specifies the vehicle's desired position along the reference (moving) path, which allows to solve the singularity that, depending on the desired path geometry, may arise if the desired path point always corresponds to the closest to the vehicle. This thesis presents an alternative solution to solve this singularity (allowing the UAV to fly with constant airspeed), which is a desirable safety feature for the applications considered in this research work. In particular, this solution contributes to operational safety by preventing a possible stall condition and avoiding sudden thrust bursts necessary to keep up with the moving path. Despite [75] considered a three dimensional error space for the MPF method using Euler angles to parametrize the rotation matrices between reference frames, the derived MPF control law was only applied to planar paths moving with time-varying linear and angular velocities at constant altitude. In the general three dimensional space navigation case, the Euler angles representation of the derived error space involves singularities which should be avoided. In order to address these open issues, a Lyapunov-based singularity-free nonlinear MPF controller for a fixed-wing UAV moving in three dimensions, applying it to the problem of autonomous landing on a moving vessel, was presented in [79].

The concept of following a three dimensional moving path for a quadcopter vehicle is also addressed in [80]. However, it only considered the case of desired paths moving with constant linear and angular velocities with respect to the inertial coordinate frame. To the author's best knowledge, the general 3D MPF error space, considering the case of desired geometric paths moving with time-varying linear and angular velocities, is originally proposed in his research work.

## 1.2 Thesis contribution

The main contributions of this thesis are the formulation and the proposed solutions of a new motion control problem, the moving path following (MPF) problem, in which a vehicle is required to converge to and follow a desired geometric path in a three dimensional space, that moves according to some specified linear and angular velocity, thus generalizing the classical problem of following stationary paths. The MPF method proposed under this research work provides a general guidance and control tool that can be used in different applications in the field of robotics. In particular, it has been employed in several real-world flight tests to fulfil the requirements of the PITVANT [81] and SEAGULL [82] research projects considering distinct mission scenarios, namely target tracking, convoy protection and autonomous landing on moving vehicles.

This thesis presents in an integrated form the main results of publications [77, 75, 76, 79]. Following the results presented in this thesis, other researchers have used MPF control laws for

thermals soaring applications with multiple UAVs [83, 45] and as benchmark to evaluate an alternative solution for the single target tracking problem [84].

In summary, the main contributions of the research presented in this thesis are as follows:

1. The derivation of the general 3D MPF error dynamical system, using quaternions to parametrize the rotation between reference frames;
2. The design of a Lyapunov-based singularity-free nonlinear MPF controller for a fixed-wing UAV moving in three dimensions at constant airspeed;
3. A new path-generation algorithm that combines with the two-dimensional MPF approach for multiple (possible moving) targets, encompassing distinct classical guidance strategies;
4. The derivation of the necessary conditions for the moving path's geometry and linear and angular velocities with respect to the inertial frame that ensure that the proposed two-dimensional MPF method is well posed;
5. Application of the MPF method as a solution to some current problems in UAV control, namely: single and multiple target tracking, convoy aerial protection, and automatic landing on a moving vehicle. This includes several numerical simulations and flight tests using the ANTEX-X02 UAV.

### 1.3 Thesis organization

The thesis is organized as follows:

**Chapter 2** formulates the Moving Path Following problem and derives the general three-dimensional error space that will be used for control design purposes. The position and orientation error dynamics are derived using the parallel-transport frame concept associated with the path to be followed.

**Chapter 3** particularizes the MPF error space for the problem of tracking a single target on the ground by a UAV flying at constant altitude, using Euler angles to parametrize the rotation matrices between reference frames. Then, it describes how the MPF method previously presented can also be applied to multiple targets tracking problems by proposing a new path-generation algorithm that, together with the MPF method, encompasses distinct classical guidance strategies. Formal convergence proofs and flight tests results for the single and multiple targets tracking scenarios are presented.

Subsequently, a convoy protection strategy employing the same 2D MPF control law is proposed, considering the case where the onboard camera's field of view is smaller than the UAV minimum turning radius. This particular problem is addressed using the MPF degree of freedom to set the desired path's angular velocity. A discussion on the desired path's best suited geometric shape for the proposed problem is also included. Illustration and validation

of the proposed strategy is provided through numerical simulations, convergence guarantees and performance metrics.

**Chapter 4** uses a quaternion formulation to parametrize the rotation matrices between reference frames to derive a singularity-free MPF kinematic control law that enables a fixed-wing UAV to converge to and follow a given 3D moving path. Formal convergence guarantees are provided along with the obtained flight test results considering two different mission scenarios: an autonomous landing on a moving vessel and tracking of a lemniscate path moving in three dimensions with respect to an inertial frame.

**Chapter 5** discusses the results obtained in this thesis and provides a summary of research problems that deserve further study.

**Appendix A** presents the hardware architecture and the UAV used in the flight tests conducted to show the effectiveness of the control methods proposed in this thesis.

**Appendix B** provides further details regarding the considered reference path's parametric equations, which allows to compute the linear and angular displacements between reference frames, that correspond to the MPF position and orientation error space variables considered in the control design.

**Appendix C** contains a detailed description of the path generation details, convergence conditions, performance metrics and the implemented control system architecture for the multiple targets tracking implementation, presented in Chapter 3.

**Appendix D** reviews the quaternion convention and the corresponding fundamentals that are used in this thesis to derive the general three-dimensional error space and control law presented respectively in chapters 2 and 4.

**Appendix E** provides a basic introduction to the Lyapunov stability theory used in this thesis for the formal stability proofs of the MPF control laws.





## Chapter 2

# Error space for Moving Path Following

This chapter presents the general MPF kinematic model, which is written with respect to the parallel-transport frame<sup>1</sup>, associated to the given reference path.

### 2.1 Position error kinematics

Let  $\{F\} = \{\vec{t}, \vec{n}_1, \vec{n}_2\}$  be the parallel-transport frame associated to the reference path with its orthonormal vectors (see Figure 1.1) satisfying the frame equations [85],

$$\begin{bmatrix} \frac{d\vec{t}}{d\ell} \\ \frac{d\vec{n}_1}{d\ell} \\ \frac{d\vec{n}_2}{d\ell} \end{bmatrix} = \begin{bmatrix} 0 & k_1(\ell) & k_2(\ell) \\ -k_1(\ell) & 0 & 0 \\ -k_2(\ell) & 0 & 0 \end{bmatrix} \begin{bmatrix} \vec{t} \\ \vec{n}_1 \\ \vec{n}_2 \end{bmatrix},$$

where parameters  $k_1(\ell)$  and  $k_2(\ell)$  are related to the path curvature  $\kappa$  and torsion  $\tau$  through [85, 86],

$$\kappa(\ell) = \sqrt{k_1(\ell)^2 + k_2(\ell)^2} \quad (2.1)$$

$$\tau(\ell) = -\frac{d}{d\ell} \left( \arctan \frac{k_2(\ell)}{k_1(\ell)} \right). \quad (2.2)$$

The  $\{I\}$ ,  $\{F\}$  and  $\{P\}$  frames are depicted in Figure 1.1. Additionally, a wind frame  $\{W\} = \{\vec{x}_w, \vec{y}_w, \vec{z}_w\}$  is considered, located at the vehicle center of mass and with its  $\vec{x}_w$ -axis along the direction of the vehicle velocity vector, the  $\vec{y}_w$ -axis parallel to the  $\vec{x} - \vec{y}$  plane, normal to  $\vec{x}_w$ , and pointing to the right of an observer that moves in the same direction of the aircraft, and  $\vec{z}_w$ -axis orthogonal to the previous two (see Figure 1.1). From this definition,  ${}^W v_W$ , the linear velocity of  $\{W\}$  relative to  $\{I\}$  and expressed in  $\{W\}$ , is given by  ${}^W v_W = \begin{bmatrix} V & 0 & 0 \end{bmatrix}^T$ , where  $V$  denotes the vehicle ground speed.

<sup>1</sup>The parallel-transport frame is used to specify a moving frame that is well defined even when the associated reference path has vanishing second derivative. Further details can be found in Appendix B and [25, 85].

The vehicle center of mass coordinates are denoted by  $p = \begin{bmatrix} x & y & z \end{bmatrix}^T$  when expressed in the inertial frame  $\{I\}$  and by  $p_F = \begin{bmatrix} x_F & y_F & z_F \end{bmatrix}^T$  when expressed in the parallel-transport frame. A given path point  ${}^P p_d(\ell)$ , parametrized by  $\ell$ , 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  $p_F = \begin{bmatrix} 0 & y_F & z_F \end{bmatrix}^T$ .

From the previous definition, the velocity of a given path point parametrized by  $\ell$  relative to  $\{I\}$  and expressed in  $\{I\}$  (that will be needed in the sequel) is given by

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

where  $S(\cdot)$  is a skew-symmetric matrix that satisfies  $S(a) b = a \times b$ . Since, by definition,

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

one obtains

$${}^I \dot{p}_d(\ell) = v_d + S(\omega_d) ({}^I p_d(\ell) - p_0). \quad (2.3)$$

The desired angular velocity of the path  $\omega_d$  with respect to the inertial frame  $\{I\}$ , written in the  $\{F\}$  frame, can be computed through

$$\begin{aligned} {}^F \omega_d &= {}^F R_I \omega_d \\ &= \begin{bmatrix} {}^F \omega_{d_x} & {}^F \omega_{d_y} & {}^F \omega_{d_z} \end{bmatrix}^T \end{aligned}$$

where  ${}^F R_I$  is the rotation matrix from  $\{I\}$  to  $\{F\}$ . According to the parallel-transport frame formulas [25], and admitting that the path is also rotating with an angular velocity given by  ${}^F \omega_d$ , the angular velocity of the  $\{F\}$  frame with respect to the inertial frame, written in the  $\{F\}$  frame, is given by

$${}^F \omega_F = \begin{bmatrix} {}^F \omega_{d_x} & -k_2(\ell)\dot{\ell} + {}^F \omega_{d_y} & k_1(\ell)\dot{\ell} + {}^F \omega_{d_z} \end{bmatrix}^T.$$

The linear velocity of  $\{W\}$  relative to  $\{I\}$  and expressed in  $\{I\}$  satisfies

$${}^I v_W = \begin{bmatrix} \dot{x} & \dot{y} & \dot{z} \end{bmatrix} = {}^I R_W {}^W v_W,$$

where  ${}^I R_W$  is the rotation matrix from  $\{W\}$  to  $\{I\}$ .

The position of the UAV in the  $\{I\}$  frame can be written as (Figure 1.1)

$$p = p_d + {}^I R_F p_F \quad (2.4)$$

where  ${}^I R_F$  is the rotation matrix from  $\{F\}$  to  $\{I\}$ . Differentiating (2.4) with respect to time yields

$$\dot{p} = \dot{p}_d + {}^I R_F \dot{p}_F + {}^I R_F S({}^F \omega_F) p_F.$$

Pre-multiplying by  ${}^F R_I$  one obtains

$${}^F R_I \dot{p} = {}^F R_I \dot{p}_d + \dot{p}_F + S({}^F \omega_F) p_F. \quad (2.5)$$

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  ${}^F v_F = \begin{bmatrix} \dot{\ell} & 0 & 0 \end{bmatrix}^T$ , with the velocity of the parallel-transport frame relative to  $\{I\}$ , both expressed in  $\{F\}$ , i.e.

$${}^F R_I \dot{p}_d = {}^F v_F + {}^F R_I \left( v_d + \underbrace{S(\omega_d)(p_d - p_0)}_{v_P} \right), \quad (2.6)$$

where  $(p_d - p_0)$  is the vector from the origin of  $\{P\}$  to the origin of the  $\{F\}$  frame on the path. The path may rotate around  $p_0$ , and thus,  $v_P$  is the linear velocity of  $p_d$ , due to path's angular velocity. Note that  $p_0$  also moves together with the path ( $\dot{p}_0 = v_d$ ), and thus the relative distance between the center of rotation of the path ( $p_0$ ) and each path point remains the same. The left side of (2.5) can be rewritten as

$${}^F R_I \dot{p} = {}^F R_W {}^W v_W. \quad (2.7)$$

Therefore, combining (2.6) with (2.7), equation (2.5) gives the position error kinematics

$$\dot{p}_F = {}^F R_W {}^W v_W - S({}^F \omega_F) p_F - {}^F v_F - {}^F R_I (v_d + S(\omega_d)(p_d - p_0)). \quad (2.8)$$

## 2.2 Orientation error kinematics

In the classical path following algorithms, the goal is to drive the error distance  $p_F$  to zero and orient the UAV such that its velocity vector  ${}^W v_W$  is aligned with the parallel-transport frame tangent direction  $\vec{t}$ . Notice however that since the desired path is now moving with possibly time-varying linear and angular velocities with respect to the inertial frame (respectively  $v_d$  and  $\omega_d$ ), one must take into account the additional velocity component of the  $\{F\}$  frame origin that is now present in the position error kinematics (given by the vector sum  $v_t = v_d + v_P$  in the last term of equation (2.8)). This result imposes a possible non zero steady state velocity component of the origin of the  $\{F\}$  frame along the normals  $\vec{n}_1$  and  $\vec{n}_2$  to the path. Thus, the goal of the moving path following controller is to drive the error distance  $p_F$  to zero and orient the UAV such that the projection of

its velocity vector onto the normal  $\vec{n}_1$  and  $\vec{n}_2$  to the path equals the normal components of the velocity of the parallel-transport frame origin. Note that by imposing this goal to the kinematic path-following, the relative motion between the vehicle and the desired moving path occurs along the tangent  $\vec{t}$  to the path and the classical situation of following paths that are fixed in space [16, 25] is included.

Given that by definition  ${}^F v_F$  is always orthogonal to  $\vec{n}_1$  and  $\vec{n}_2$  and assuming that  $\|v_t\| < \|{}^W v_W\|$  (i.e., the magnitude of the desired moving path velocity is always smaller than the magnitude of the vehicle's velocity), the desired steady state value  ${}^F v_{W_d}$  for the desired displacement between the wind frame  $\{W\}$  and the parallel-transport frame  $\{F\}$  can be computed from the second and third equations of the kinematic model (2.8), by setting  $\dot{p}_F = p_F = 0$ , yielding

$$\Pi_R {}^F v_{W_d} = \Pi_R {}^F R_I (v_d + S(\omega_d) (p_d - p_0)), \quad (2.9)$$

where

$$\Pi_R \triangleq \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Let  $\beta_{n1}$  denote the angle between  $v_t$  and  $\vec{n}_1$  and let  $\beta_{n2}$  denote the angle between  $v_t$  and  $\vec{n}_2$ . Using the previous definitions and imposing a positive sign on the tangent component (along the path) of the desired UAV velocity vector, one finally obtains

$${}^F v_{W_d} = \|{}^W v_W\| \begin{bmatrix} \Upsilon \\ \frac{\|v_t\| \cos(\beta_{n1})}{\|{}^W v_W\|} \\ \frac{\|v_t\| \cos(\beta_{n2})}{\|{}^W v_W\|} \end{bmatrix} \quad (2.10)$$

with

$$\Upsilon = \sqrt{1 - \left( \frac{\|v_t\| \cos(\beta_{n1})}{\|{}^W v_W\|} \right)^2 - \left( \frac{\|v_t\| \cos(\beta_{n2})}{\|{}^W v_W\|} \right)^2}.$$

Consider now a desired wind frame  $\{W_d\}$ , defined similarly to  $\{W\}$ , with its  $\vec{x}_{W_d}$ -axis along the direction of  ${}^F v_{W_d}$ . By definition, it follows that

$${}^F v_{W_d} = {}^F R_{W_d} {}^W v_W \quad (2.11)$$

where  ${}^F R_{W_d}$  parametrizes the desired displacement between the  $\{W\}$  and  $\{F\}$  frames. Note that the case where  $v_d = \omega_d = 0 \Rightarrow v_t = 0$  corresponds to the classical path following case and from the above computations one obtains  ${}^F v_{W_d} = \|{}^W v_W\| \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T$  where clearly (in this classical path following case)  ${}^F R_{W_d} = I_3$ .

Combining (2.9) with (2.11) and differentiating (2.9) with respect to time one obtains

$$\begin{aligned} \Pi_R \left( {}^F \dot{R}_{W_d} {}^W v_W + {}^F R_{W_d} {}^W \dot{v}_W \right) = & \Pi_R \left( {}^F \dot{R}_I (v_d + S(\omega_d) (p_d - p_0)) \right. \\ & \left. + {}^F R_I (\dot{v}_d + S(\dot{\omega}_d) (p_d - p_0) + S(\omega_d) (\dot{p}_d - \dot{p}_0)) \right), \end{aligned}$$

which, using straightforward algebraic manipulations, can be rewritten as

$$\begin{aligned} \Pi_R S({}^F \omega_{W_d}) {}^F v_{W_d} = & \Pi_R \left( - {}^F R_{W_d} {}^W \dot{v}_W - S({}^F \omega_F) {}^F R_I (v_d + S(\omega_d) (p_d - p_0)) \right. \\ & \left. + {}^F R_I (\dot{v}_d + S(\dot{\omega}_d) (p_d - p_0) + S(\omega_d) (\dot{p}_d - \dot{p}_0)) \right). \end{aligned}$$

Note that, by definition, the desired relative roll rate velocity between the  $\{W_d\}$  frame and the  $\{F\}$  frame is equal to zero (i.e.,  ${}^F \omega_{W_{dx}} = 0$ ) and thus, the left-hand side of the above result (that will be needed in the sequel) can be written as

$$S({}^F \omega_{W_d}) {}^F v_{W_d} = \| {}^W v_W \| \Upsilon \begin{bmatrix} 0 & {}^F \omega_{W_{dz}} & -{}^F \omega_{W_{dy}} \end{bmatrix}^T,$$

where  ${}^F \omega_{W_d}$  is the desired relative angular velocity between the wind frame  $\{W\}$  and the  $\{F\}$  frame, written in the  $\{F\}$  frame. Finally, one can use the above result to obtain

$${}^F \omega_{W_d} = \left[ \frac{1}{\| {}^W v_W \| \Upsilon} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \Pi_R S({}^F \omega_{W_d}) {}^F v_{W_d} \right] \in \mathbb{R}^3.$$

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

$${}^W \omega_{W,F;W_d}^r = {}^W \omega_W - {}^W \omega_F - {}^W R_F {}^F \omega_{W_d} \quad (2.12)$$

and it can be shown that

$${}^{W_d} \dot{R}_W = {}^{W_d} R_W S({}^W \omega_{W,F;W_d}^r). \quad (2.13)$$

In summary, the complete MPF error kinematics is given by equations (2.8) and (2.13). Note that for fixed paths, that is with  $v_d = 0$ ,  $\omega_d = 0$ , equations (2.8) and (2.13) resume to the classical path following expressions, which can be found in the works [11, 23, 16]. The goal of the MPF kinematic controller is to converge  $p_F$  to zero and  ${}^{W_d} R_W$  to the identity matrix  $I_3$ .



## Chapter 3

# 2D Moving Path Following control design

This chapter starts by particularizing the error space defined in Chapter 2 to the case where Euler angles 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.

### 3.1 2D error space using Euler angles

For the 2D MPF problem, it follows by definition that  $x_F = z_F = \omega_{d_x} = \omega_{d_y} = 0$ , and the goal of the path following controller 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  $\vec{n}_1$  to the path equals the normal component of the velocity of the parallel-transport frame origin given by the vector sum  $v_d + v_p$  (see Figure 1.1 and equation (2.6)). Thus, the relative motion between the vehicle and the desired moving path occurs along the tangent  $\vec{t}$  to the path. Let  $\psi$  be the yaw angle, that is, the angle between the vehicle velocity vector and the North direction. Since 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 inertial frame through  $\omega = \dot{\psi}$ . Additionally, let  $\psi_p$  be the angle that parametrizes the rotation matrix from  $\{I\}$  to  $\{P\}$  (thus, by definition  $\omega_{d_z} = \dot{\psi}_p$ ) and let  $\psi_f$  be the yaw angle that parametrizes the rotation matrix from  $\{I\}$  to  $\{F\}$ . The angular displacement between the wind frame and the parallel-transport frame is  $\bar{\psi} = \psi - \psi_f$ . Figure 3.1 shows the error space for the 2D moving path following. Taking into account the last notation, the UAV kinematic equations expressed in  $\{I\}$  are given by

$$\begin{aligned}\dot{x} &= V \cos \psi, \\ \dot{y} &= V \sin \psi, \\ \dot{\psi} &= \omega, \quad \text{with } \omega \in [-\omega_{max}, \omega_{max}]\end{aligned}\tag{3.1}$$

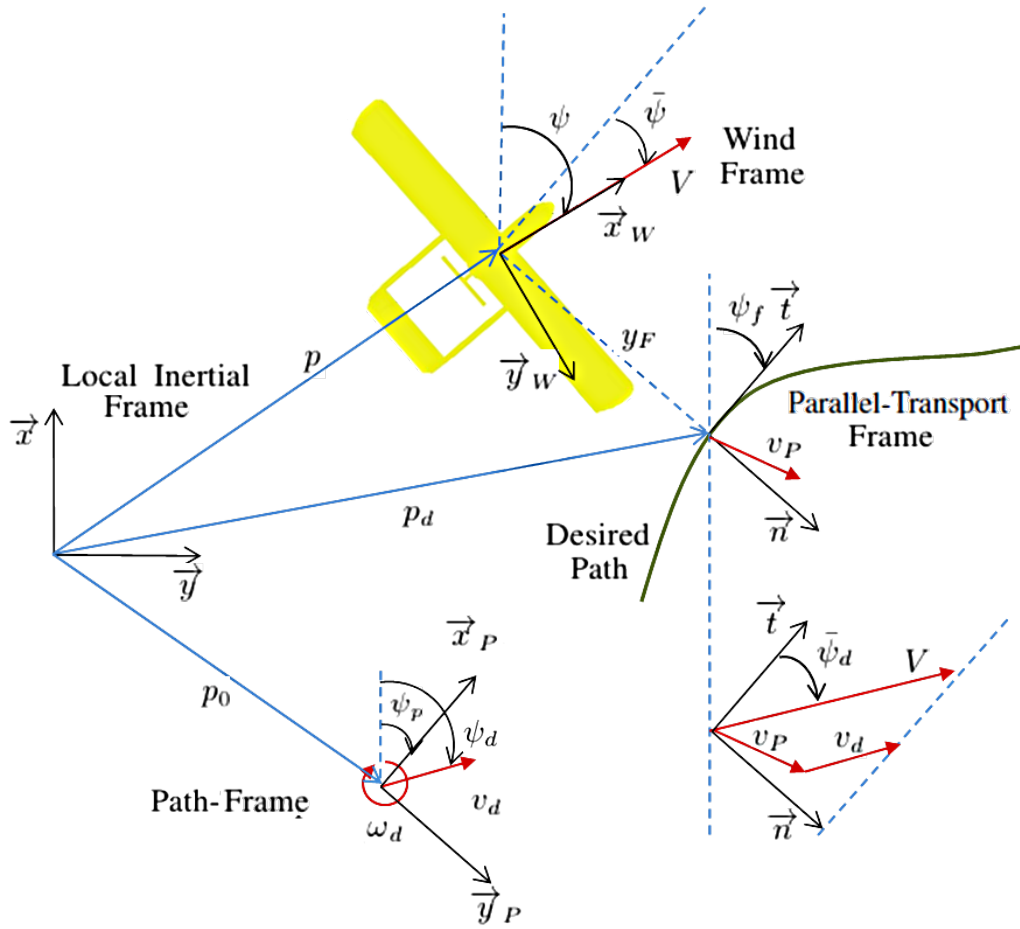


Figure 3.1: Moving Path Following: Error space frames and relevant variables, illustrating the case of a UAV.

where  $\omega_{max}$  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 that is attached to the desired path is described by the following kinematic equations in terms of the total speed  $\|v_d\|$  and the yaw angle  $\psi_d$

$$\begin{aligned} v_{d_x} &= \|v_d\| \cos \psi_d \\ v_{d_y} &= \|v_d\| \sin \psi_d. \end{aligned} \quad (3.2)$$

Therefore, equation (2.8) can be rewritten as

$$\begin{aligned} \begin{bmatrix} 0 \\ \dot{y}_F \end{bmatrix} &= \begin{bmatrix} V \cos \bar{\psi} \\ V \sin \bar{\psi} \end{bmatrix} - \begin{bmatrix} \dot{\ell} \\ 0 \end{bmatrix} - {}^F R_I(\psi_f) \left( \begin{bmatrix} v_{d_x} \\ v_{d_y} \end{bmatrix} + \begin{bmatrix} 0 & -\omega_{d_z} \\ \omega_{d_z} & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \right) \\ &\quad - \begin{bmatrix} -\dot{\ell}(\kappa(\ell)y_F) - \omega_{d_z} y_F \\ 0 \end{bmatrix}, \end{aligned} \quad (3.3)$$



where  $\Delta_x$  and  $\Delta_y$  denote the vector coordinates of  $(p_d - p_0)$ . 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 (2.12))

$$\dot{\bar{\psi}} = \dot{\psi} - \kappa(\ell) \dot{\ell} - \omega_{d_z}. \quad (3.4)$$

The previous result together with (3.3) gives the 2D MPF error kinematic model

$$\begin{aligned} \dot{\ell} &= \frac{V \cos \bar{\psi} - (v_{d_x} - \omega_{d_z} \Delta y) \cos \psi_f - (v_{d_y} + \omega_{d_z} \Delta x) \sin \psi_f + \omega_{d_z} y_F}{1 - \kappa(\ell) y_F} \\ \dot{y}_F &= V \sin \bar{\psi} + (v_{d_x} - \omega_{d_z} \Delta y) \sin \psi_f - (v_{d_y} + \omega_{d_z} \Delta x) \cos \psi_f \\ \dot{\bar{\psi}} &= \dot{\psi} - \kappa(\ell) \dot{\ell} - \omega_{d_z}, \end{aligned} \quad (3.5)$$

where  $\bar{\psi}$  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  $\ell$ ) that corresponds to the inverse of the path's curvature at that point. This singularity could be avoided using a “virtual target” to specify the desired position of the UAV on the path, not necessarily coincident with the projection of the vehicle on the path (and thus  $x_F$  would not necessarily be zero). By choosing the speed of the virtual target along the path ( $\dot{\ell}$ ) it is possible to remove the singularity. Further details of this method can be found in [20, 21]. Section B.3 in Appendix B presents an alternative solution to compute  $\dot{\ell}$  that can be used to ensure that, at the implementation level,  $\dot{\ell}$  is always finite. The advantage of using the approach proposed in section B.3 with respect to [20, 21] is that the UAV can fly at constant airspeed, thus contributing to operational safety by preventing a possible stall condition and avoiding sudden thrust bursts necessary to keep up with the moving path.

The error space for MPF given by equation (3.5) will be used in the sequel to derive the 2D MPF control law for planar paths.

## 3.2 2D Moving Path Following control law

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

Figure 3.1 illustrates a MPF application example where a UAV should follow a planar (horizontal) path moving with linear velocity  $v_d$  and angular velocity  $\omega_d$  with respect to an inertial frame, keeping its altitude constant. Considering the kinematic model (3.5), the desired steady state value  $\bar{\psi}_d$  for  $\bar{\psi}$  can be computed by setting  $\dot{y}_F = 0$  (see also equation (2.9) for the general three dimensional case), which yields

$$\bar{\psi}_d = \arcsin \left( \frac{-(v_{d_x} - \omega_{d_z} \Delta y) \sin \psi_f + (v_{d_y} + \omega_{d_z} \Delta x) \cos \psi_f}{V} \right) \quad (3.6)$$

Note that the numerator of the arcsin argument is the sum of the path 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 (3.6) 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 frame and the parallel-transport frame, given by  $\sqrt{\Delta x^2 + \Delta y^2}$ . More specifically, from (3.6), it must be ensured that

$$|\omega_{d_z}| \sqrt{\Delta x^2 + \Delta y^2} < \frac{V - \|v_d\| \sin(\psi_d - \psi_f)}{|\sin(\psi_f + \arctan(\frac{\Delta y}{\Delta x}))|} \quad \text{and} \quad \|v_d\| < V. \quad (3.7)$$

Notice, for instance, that in the particular case where the path's angular velocity is equal to zero ( $\omega_{d_z} = 0$  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 frame ( $\omega_{d_z} \neq 0$  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  $\frac{V}{|\omega_{d_z}|}$ . These particular cases provide an intuitive interpretation for the conditions in (3.7). Additionally, a MPF control law should drive the lateral distance  $y_F$  and heading error  $\tilde{\psi} = \bar{\psi} - \bar{\psi}_d$  to zero (this goal corresponds to drive  ${}^{W_d}R_W(\tilde{\psi})$  to  $I_3$  as presented in Section 2.2). Thus, considering equation (3.4), the steady state value for the commanded yaw rate  $\dot{\psi}$  (obtained by setting  $\ddot{\psi} = 0$ ) is given by

$$\dot{\psi} = \dot{\bar{\psi}}_d + \kappa(\ell) \dot{\ell} + \omega_{d_z}. \quad (3.8)$$

In order to take into account the vehicle kinematic constraints  $|\omega| < \omega_{max}$  (see equation (3.1)), it is now straightforward to conclude that condition

$$|\dot{\bar{\psi}}_d + \kappa(\ell) \dot{\ell} + \omega_{d_z}| \leq \omega_{max} \quad (3.9)$$

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 3.3) 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<sup>1</sup>. 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

<sup>1</sup>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.

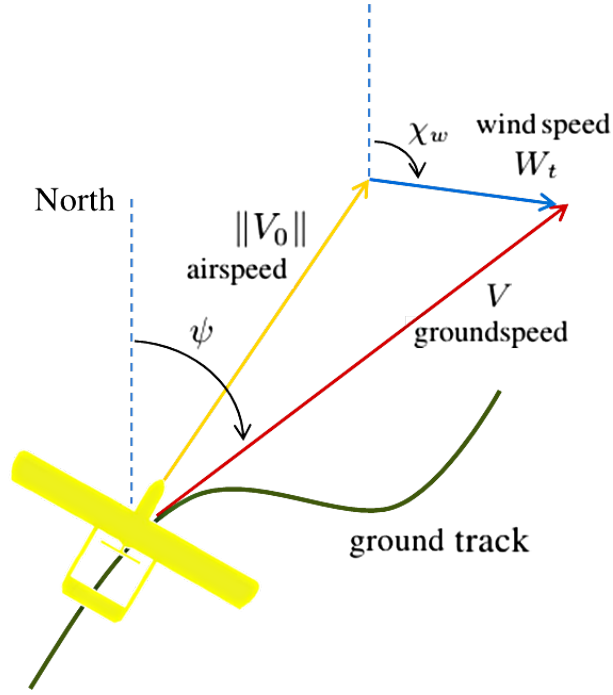


Figure 3.2: Relationship between the groundspeed vector, the airspeed vector and the wind vector. Adapted from [87].

by  $\chi_w$  (both assumed to be constant throughout the flight), one can relate the airspeed vector  $v_0$  and the wind velocity vector  ${}^I v_{wind}$  with the UAV's ground velocity (see Figure 3.2) through [87]

$$v_0 = {}^I R_W(\psi) {}^W v_W - {}^I v_{wind}. \quad (3.10)$$

Taking the squared norm of both sides of (3.10) and solving the resulting quadratic equation for the total ground speed  $V$  one obtains

$$V = \sqrt{\|v_0\|^2 + W_t^2 (\cos^2(\psi - \chi_w) - 1) + W_t \cos(\psi - \chi_w)}.$$

Considering the previous computations, the derivative of  $\tilde{\psi}_d$  with respect to time, that will be necessary in the sequel, assuming that the autopilot is able to keep  $\|v_0\|$  constant, is

$$\begin{aligned} \dot{\tilde{\psi}}_d = & \frac{\rho}{V \sqrt{1 - \left( \frac{-(v_{dx} - \omega_{dz} \Delta y) \sin \psi_f + (v_{dy} + \omega_{dz} \Delta x) \cos \psi_f}{V} \right)^2}} \\ & - \dot{\tilde{\psi}} \frac{\lambda}{V^2 \sqrt{1 - \left( \frac{-(v_{dx} - \omega_{dz} \Delta y) \sin \psi_f + (v_{dy} + \omega_{dz} \Delta x) \cos \psi_f}{V} \right)^2}} \end{aligned} \quad (3.11)$$

where

$$\begin{aligned}\rho &= (-\dot{\psi}_f (v_{dx} - \omega_{dz} \Delta y) + \dot{\omega}_{dz} \Delta x + \omega_{dz} \hat{\Delta}x) \cos \psi_f + (-\dot{\psi}_f (v_{dy} + \omega_{dz} \Delta x) + \dot{\omega}_{dz} \Delta y + \omega_{dz} \hat{\Delta}y) \sin \psi_f \\ &\quad + \|v_d\| \dot{\psi}_d \cos(\psi_d - \psi_f) + \|\dot{v}_d\| \sin(\psi_d - \psi_f) \\ \lambda &= \hat{V} (- (v_{dx} - \omega_{dz} \Delta y) \sin \psi_f + (v_{dy} + \omega_{dz} \Delta x) \cos \psi_f)\end{aligned}$$

with

$$\begin{aligned}\hat{\Delta}x &= \dot{\ell} \cos \psi_f - \omega_{dz} \Delta y \\ \hat{\Delta}y &= \dot{\ell} \sin \psi_f + \omega_{dz} \Delta x \\ \hat{V} &= -W_t \sin(\psi - \chi_w) \left( \frac{W_t \cos(\psi - \chi_w)}{\sqrt{\|v_0\|^2 + W_t^2 (\cos^2(\psi - \chi_w) - 1)}} + 1 \right).\end{aligned}$$

Equation (3.11) can be cast in the compact form

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

with

$$P = \frac{\rho}{V \sqrt{1 - \left( \frac{-(v_{dx} - \omega_{dz} \Delta y) \sin \psi_f + (v_{dy} + \omega_{dz} \Delta x) \cos \psi_f}{V} \right)^2}}$$

and

$$\Lambda = \frac{\lambda}{V^2 \sqrt{1 - \left( \frac{-(v_{dx} - \omega_{dz} \Delta y) \sin \psi_f + (v_{dy} + \omega_{dz} \Delta x) \cos \psi_f}{V} \right)^2}},$$

where it can be shown that  $\Lambda \neq -1$  under the conditions given by equation (3.7).

It is now possible to derive a control law to drive the linear distance  $y_F$  and heading error  $\tilde{\psi} = \bar{\psi} - \tilde{\psi}_d$  to zero. To that end, consider the control law

$$\begin{aligned}\dot{\psi} &= \left( -g_1 \tilde{\psi} + \kappa(\ell) \dot{\ell} + \omega_{dz} + P - g_2 y_F \left( ((v_{dx} - \omega_{dz} \Delta y) \sin \psi_f \right. \right. \\ &\quad \left. \left. - (v_{dy} + \omega_{dz} \Delta x) \cos \psi_f) \frac{1 - \cos \tilde{\psi}}{\tilde{\psi}} + V \cos \bar{\psi}_d \frac{\sin \tilde{\psi}}{\tilde{\psi}} \right) \right) / (1 + \Lambda)\end{aligned}\tag{3.12}$$

and the Lyapunov function

$$V_1 = \frac{1}{2} \left( y_F^2 + \frac{1}{g_2} \tilde{\psi}^2 \right),\tag{3.13}$$

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 (3.1) in closed-loop with the control law (3.12), subject to (3.7) and (3.9). Then, the closed-loop error signals  $\tilde{\psi}$  and  $y_F$  converge to zero as  $t \rightarrow \infty$ . Moreover, for constant ground speed, 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 (3.13) and the Barbalat lemma [88].

Differentiating  $V_1$  with respect to time yields

$$\begin{aligned}\dot{V}_1 &= y_F \dot{y}_F + \frac{1}{g_2} \tilde{\psi} \dot{\tilde{\psi}} \\ &= V y_F \sin \tilde{\psi} \cos \tilde{\psi}_d + V y_F \cos \tilde{\psi} \sin \tilde{\psi}_d + y_F \left( (v_{dx} - \omega_{dz} \Delta y) \sin \psi_f - (v_{dy} + \omega_{dz} \Delta x) \cos \psi_f \right) \\ &\quad + \frac{1}{g_2} \tilde{\psi} (\dot{\tilde{\psi}} - \dot{\tilde{\psi}}_d).\end{aligned}\tag{3.14}$$

Since, by definition (cf. equation (3.6)),

$$V \sin \tilde{\psi}_d + (v_{dx} - \omega_{dz} \Delta y) \sin \psi_f - (v_{dy} + \omega_{dz} \Delta x) \cos \psi_f = 0$$

the previous expression is equivalent to

$$\begin{aligned}\dot{V}_1 &= V y_F \sin \tilde{\psi} \cos \tilde{\psi}_d + \frac{1}{g_2} \tilde{\psi} (\dot{\tilde{\psi}} (1 + \Lambda) - \kappa(\ell) \dot{\ell} - \omega_{dz} - P) \\ &\quad + y_F \left( (v_{dx} - \omega_{dz} \Delta y) \sin \psi_f - (v_{dy} + \omega_{dz} \Delta x) \cos \psi_f \right) (1 - \cos \tilde{\psi}).\end{aligned}$$

The control law (3.12) with  $g_1, g_2 > 0$  makes

$$\dot{V}_1 = -\frac{g_1}{g_2} \tilde{\psi}^2 \leq 0.\tag{3.15}$$

Given the definition of  $V_1$  and the fact that  $\dot{V}_1 \leq 0$ , 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 [88] allows for the conclusion that  $\dot{V}_1$  and consequently  $\tilde{\psi}$  tend to zero as  $t \rightarrow \infty$ . Rewriting (3.12) as

$$\begin{aligned}\dot{\tilde{\psi}} &= -g_1 \tilde{\psi} - g_2 y_F \left( (v_{dx} - \omega_{dz} \Delta y) \sin \psi_f - (v_{dy} + \omega_{dz} \Delta x) \cos \psi_f \right) \frac{1 - \cos \tilde{\psi}}{\tilde{\psi}} \\ &\quad - g_2 V y_F \cos \tilde{\psi}_d \frac{\sin \tilde{\psi}}{\tilde{\psi}},\end{aligned}\tag{3.16}$$

differentiating  $\tilde{\psi}$  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  $\tilde{\psi}$  tends to zero, leads to the conclusion that also  $y_F$  tends to zero as  $t \rightarrow \infty$ .

### Local asymptotic stability

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

$$\begin{aligned} \dot{y}_F &= V \sin(\tilde{\psi}_d + \tilde{\psi}) + (v_{dx} - \omega_{dz} \Delta y) \sin \psi_f - (v_{dy} + \omega_{dz} \Delta x) \cos \psi_f \\ \dot{\tilde{\psi}} &= -g_1 \tilde{\psi} - g_2 y_F \left( (v_{dx} - \omega_{dz} \Delta y) \sin \psi_f \right. \\ &\quad \left. - (v_{dy} + \omega_{dz} \Delta x) \cos \psi_f \right) \frac{1 - \cos \tilde{\psi}}{\tilde{\psi}} - g_2 V y_F \cos \tilde{\psi}_d \frac{\sin \tilde{\psi}}{\tilde{\psi}}, \end{aligned} \quad (3.17)$$

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

$$A(t) = \begin{bmatrix} 0 & V \cos \tilde{\psi}_d \\ -g_2 V \cos \tilde{\psi}_d & -g_1 \end{bmatrix}.$$

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

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

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

$$P(t) = \begin{bmatrix} \frac{g_2 V^2 \cos^2 \tilde{\psi}_d (g_2 + 1) + g_1^2}{2 g_1 g_2 V^2 \cos^2 \tilde{\psi}_d} & \frac{1}{2 g_2 V \cos \tilde{\psi}_d} \\ \frac{1}{2 g_2 V \cos \tilde{\psi}_d} & \frac{g_2 + 1}{2 g_1 g_2} \end{bmatrix}$$

which is bounded below and above by positive definite bounded matrices for every positive constants  $g_1$  and  $g_2 > 0$ , and considering the following Lyapunov function:

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

whose derivative is given by

$$\dot{V}_2 = -e^T (Q(t) + \dot{P}(t)) e \leq -K \|e\|^2,$$

with  $K > 0$ , it is straightforward to conclude that under the proposed control law, the origin of the closed-loop error kinematic model (3.17) is an exponentially stable equilibrium point [88].  $\square$

<sup>2</sup>Note that  $f(t, e)$  depends on the possibly time-varying variables  $v_d$  and  $\omega_d$ .

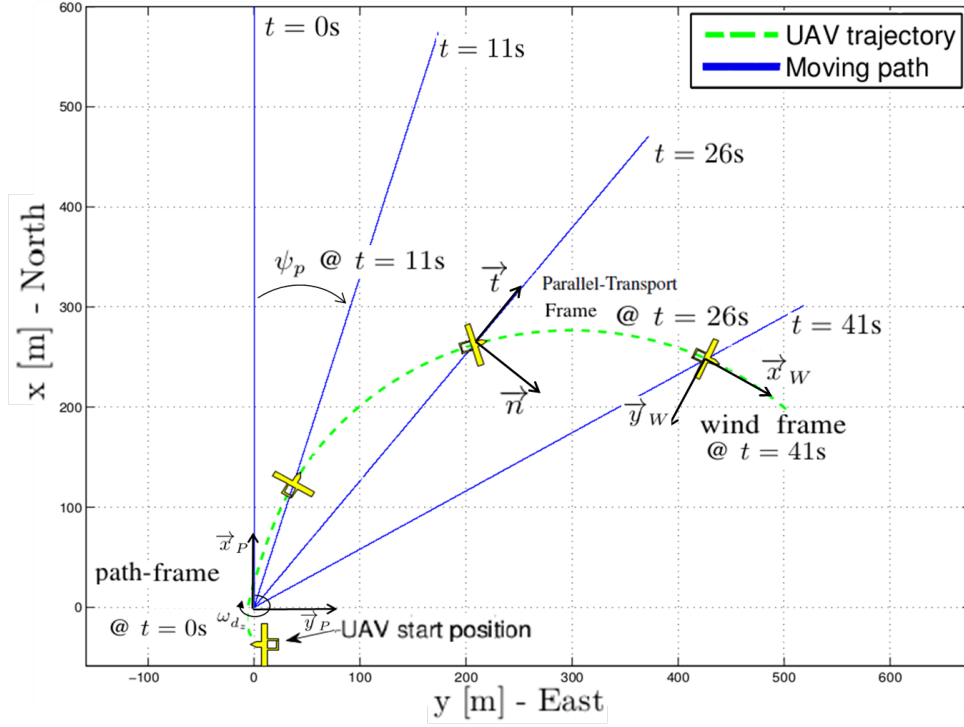


Figure 3.3: 2D MPF numerical simulation example: UAV following a moving straight line that is rotating around a fixed point  $p_0 = [0 \ 0]$  with an angular velocity  $\omega_{d_z} = 0.025 \text{ rad/s}$ .

Figure 3.3 illustrates an example where a UAV flying at  $V = 15 \text{ m/s}$  follows a moving straight line path rotating around a fixed point  $p_0 = [0 \ 0]$  with an angular velocity  $\omega_{d_z} = 0.025 \text{ rad/s}$ . 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  $\psi_p$  between North and the straight line that connects the origin of  $\{I\}$  with  ${}^I p_d(\ell)$  at time instant  $t = 0 \text{ s}$  is zero and increases as  $t$  increases, due to path's angular velocity  $\omega_{d_z}$ . For an observer fixed with the path-frame and standing at the origin  $p_0$  looking towards the positive direction of  $\vec{x}_p$ , the UAV will be moving in a straight line with a decreasing forward speed as the distance between the origin of the path frame and the parallel-transport frame increases. From equation (3.6) 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 parallel-transport frame  $\tilde{\psi}_d$  also increases as the UAV needs to compensate for an increasing normal component of  $v_p$  with respect to the normal to the path  $\vec{n}$ . Thus, in this particular example, the straight line path length is limited to be smaller than  $\frac{V}{|\omega_{d_z}|} = 600 \text{ m}$  (c.f. equations (3.7)) in order to ensure that the MPF is well posed.

### 3.3 Application to single and multiple target tracking problems

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

#### 3.3.1 Single target tracking problem formulation as MPF

Application of the MPF control law to a single ground target tracking scenario is straightforward. A given path with a pre-defined geometry can be attached to the desired target (moving together with it - with possibly time varying linear and angular velocities) and the control law given by equation (3.12) can be used to make the UAV converge to and follow the moving path. To illustrate this application, Figure 3.4 shows a numerical simulation where a UAV flying at 20m/s airspeed is required to track a target by following a 300m width lemniscate path<sup>3</sup> that is moving together with the target, keeping the line that connects the two foci always perpendicular to  $\psi_d$  (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

$$\begin{aligned} (p_{tx}, p_{ty}, \psi_t, \|v_t\|)|_{t=0} &= (0\text{m}, 0\text{m}, 0, 4\text{m/s}) \\ \|\dot{v}_t\| &= 0.2 \sin(0.07t) \text{ m/s}^2 \\ \dot{\psi}_t &= 0.02 \cos(0.03t) \text{ rad/s} \end{aligned} \quad (3.18)$$

where  $v_t$  corresponds to the target velocity,  $\psi_t$  is the target heading, and  $t$  is the simulation time. In order to attach the desired path to the target, the path frame is set with  $p_0|_{t=0} = [p_{tx} \ p_{ty}]|_{t=0}$ ,  $v_d = v_t$  and  $\psi_d = \psi_t$ .

The controller parameters used are listed in Table 3.1.

Table 3.1: Single target tracking numerical simulation: Controller parameters.

$g_1 = 0.22$	$\omega_{d_z} = \dot{\psi}_t$
$g_2 = 0.0002$	$\dot{\omega}_{d_z} = \ddot{\psi}_t = -0.0006 \sin(0.03t)$

Figure 3.5 demonstrates the performance of the overall control system. At time instants  $t = 80\text{s}$  and  $t = 150\text{s}$  a perturbation on the distance and heading errors due to the imposed sudden change on the wind velocity is observed. In Section 3.3.2 flight test results for a single target tracking scenario are presented, thus allowing for a comparison between simulation and experimental results.

<sup>3</sup>For a discussion on the use of a lemniscate instead of another path, please refer to Section 3.4.2



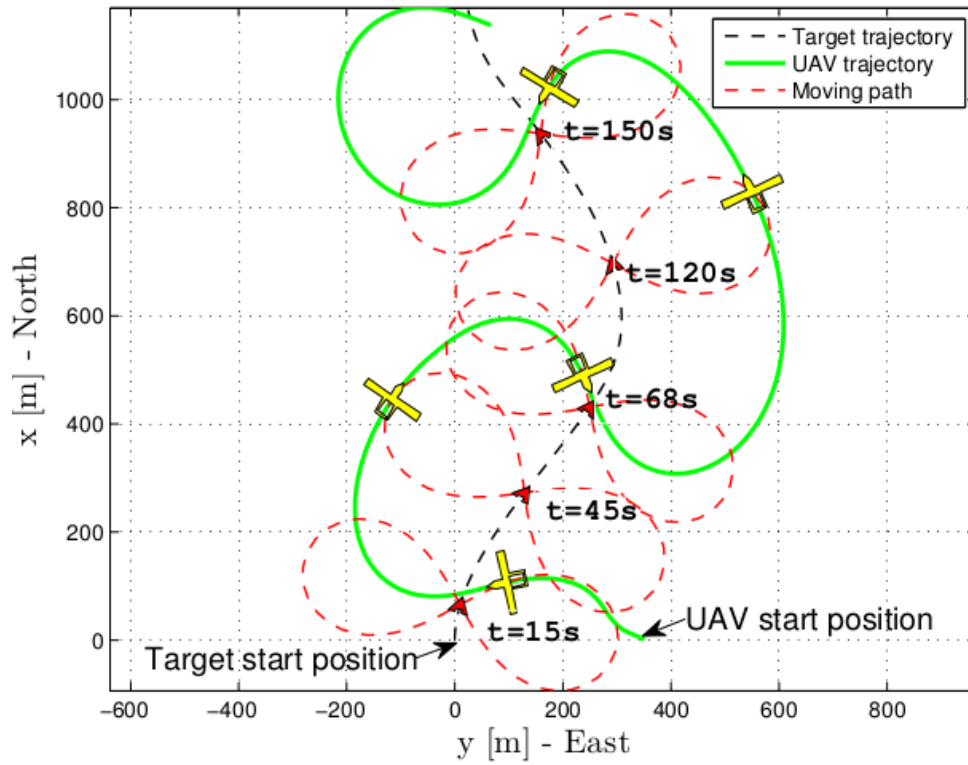


Figure 3.4: Single target tracking numerical simulation: Aircraft's trajectory following a target between  $t=0s$  and  $t=200s$ .

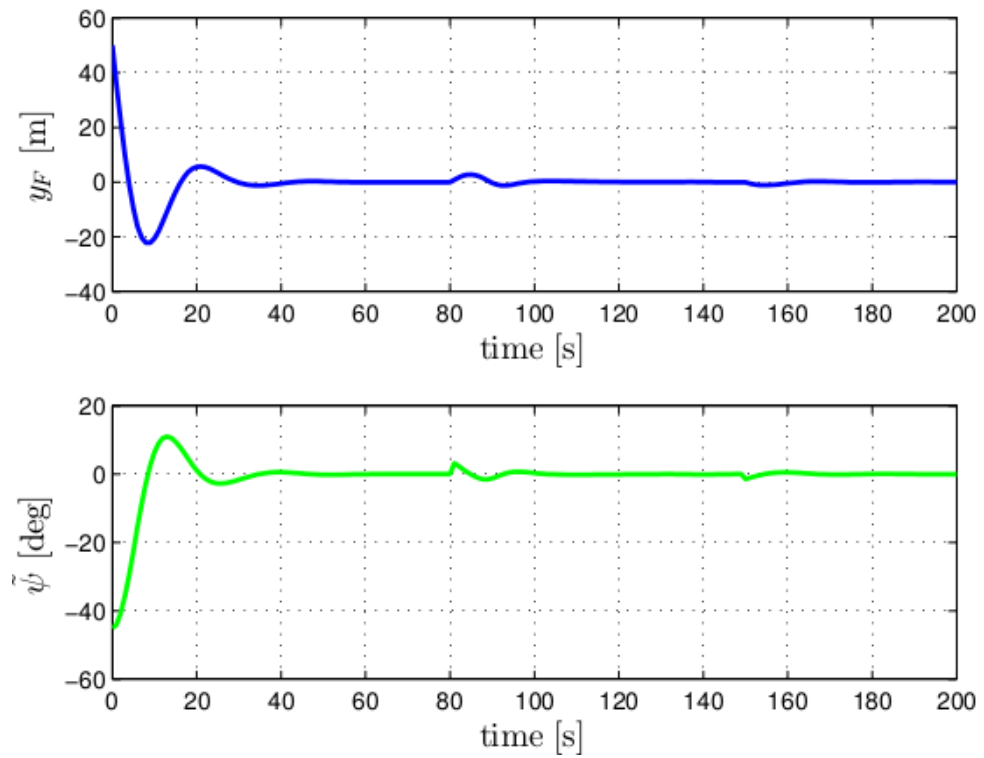


Figure 3.5: Single target tracking numerical simulation: Position and heading errors.

### 3.3.2 Flight test results: Single target tracking

In this experiment, the goal was to make the UAV track a ground vehicle by following a 300m width lemniscate centered at the target position, keeping the line that connects the two foci always perpendicular to  $\psi_t$ . 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 previously presented (cf. equations (3.18)).

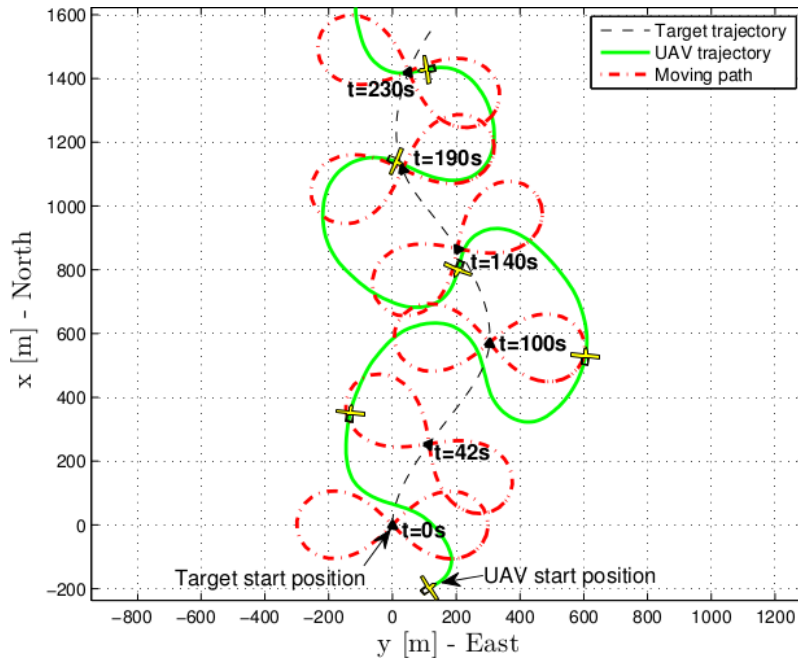


Figure 3.6: Single target tracking flight test: Aircraft's trajectory following a target on the ground (from  $t = 0s$  to  $t = 250s$ ).

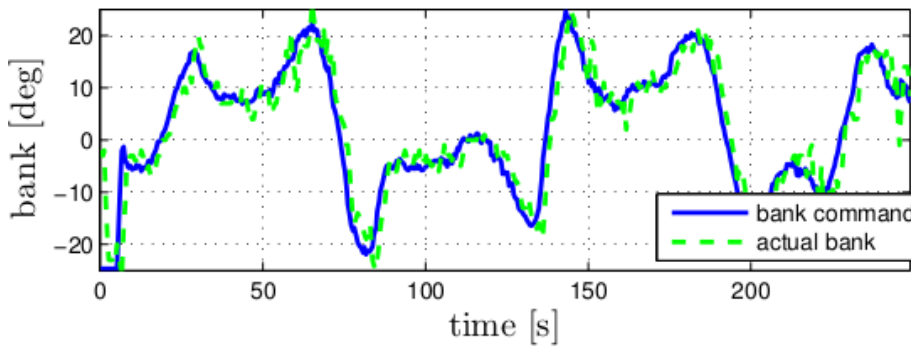


Figure 3.7: Single target tracking flight test: Commanded and actual bank.

The obtained qualitative behaviour of the UAV is similar to the results presented for the numerical simulation. Figure 3.6 shows the overall UAV and target trajectories. The UAV, target and target path positions at sample time instants are also presented. Figure 3.7 shows that the bank command

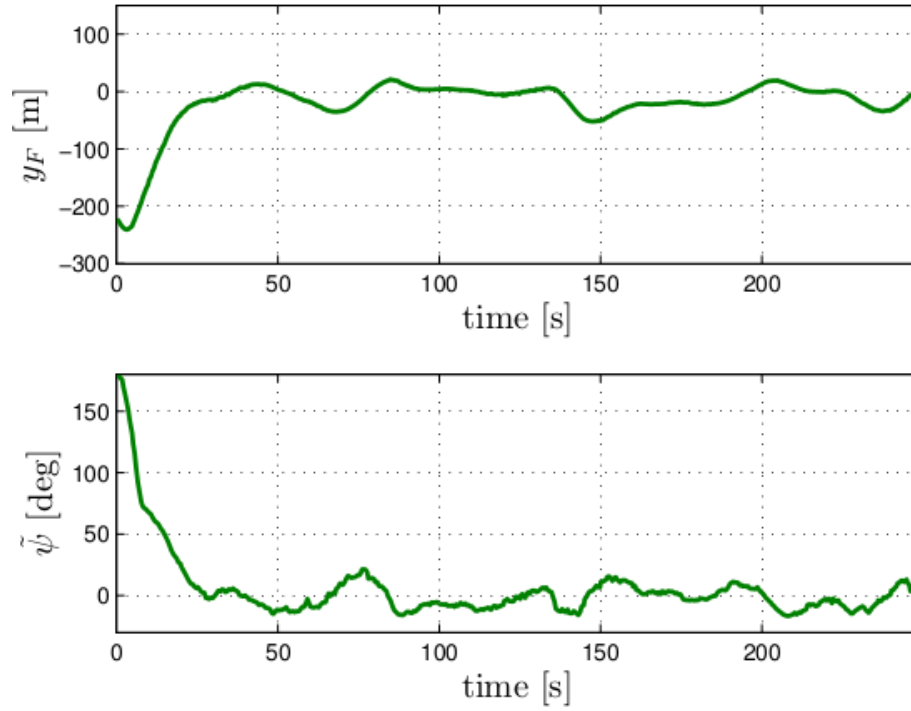


Figure 3.8: Single target tracking flight test: Distance and heading errors.

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 3.8 demonstrate the good performance of the implemented control system.

### 3.3.3 Multiple targets tracking problem formulation as MPF

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  $2r_{min}$  apart, one strategy is to use a Dubins path<sup>4</sup> computed without specifying the course angle at the end point [53]. Figure 3.9 illustrates this strategy for path generation, given the start position  $p_s = (x_s, y_s)$ , the initial course angle  $\psi_s$ , 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 3.9. In particular,

$$d_{P_i} = d_{S_i} + d_{C_i} r_{min} \quad (3.19)$$

<sup>4</sup>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.

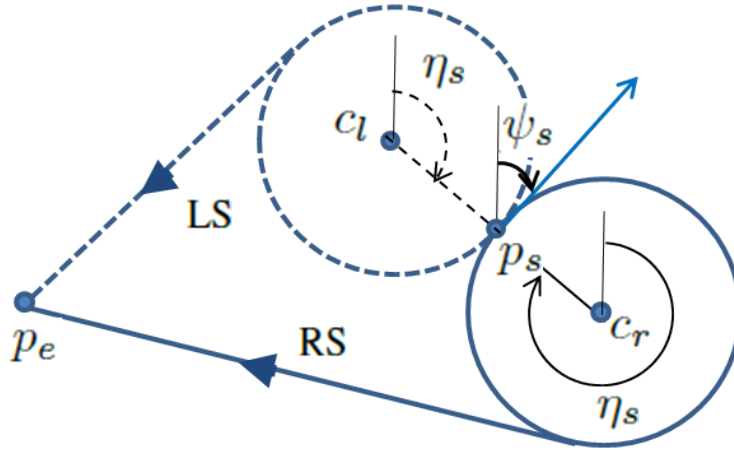


Figure 3.9: Dubins path: RS or LS.

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  $\psi_s$  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  $\psi_s$  (see Figure 3.9);
- From (3.19) 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
- 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  $\eta_s$  (between the line that connects  $c_i$  to the vehicle's initial position,  $p_s$ , and the  $\vec{x}$  axis - see Figure 3.10) 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

$$\eta_{vi} \geq \begin{cases} \langle 2\pi - \eta_e + \eta_s \rangle & \text{if } i = l, \\ \langle 2\pi + \eta_e - \eta_s \rangle & \text{if } i = r \end{cases} \quad (3.20)$$

where  $\eta_e$  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  $\eta_v$  is the angular distance

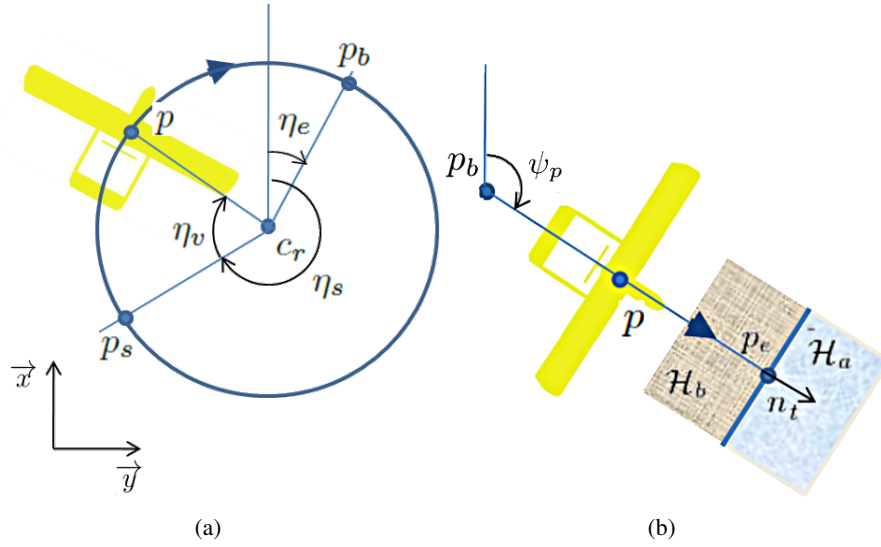


Figure 3.10: Dubin's path relevant parameters for moving targets, illustrating for the right circular segment case (a), followed by a straight line section (b).

between  $\eta_s$  and the vehicle's current position,  $p$  (see Figure 3.10(a)). In (3.20),  $\langle \eta \rangle \triangleq \eta \bmod 2\pi$  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  $p_e$ ,  $\eta_e$  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 3.2.

The angle  $\psi_p$  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 3.10 (b)). Its derivative is given by

$$\dot{\psi}_p = \frac{1}{1 + \left(\frac{\Delta y_t}{\Delta x_t}\right)^2} \frac{\Delta \dot{y}_t \Delta x_t - \Delta y_t \Delta \dot{x}_t}{\Delta x_t^2}. \quad (3.21)$$

Recalling the MPF problem formulation and the illustration example discussed in Section 3.2 (see also Figure 3.3), 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 (3.21), under the assumption that the UAV and the target are initially at least  $2r_{min}$  apart, it can easily be shown that if the target's speed is always smaller than the UAV's speed ( $\|v_t\| < V$ ), the conditions imposed by equations (3.7) and (3.9) always hold (independently from the distance between the UAV and the target for  $t > 0$ ) and thus the 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,

---

**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  $(p, \psi)$ , sequenced target's position  $s_t = (t_1, t_2, \dots, t_m)$  with  $t_j = (x_{t_j}, y_{t_j})$ , and path constraints  $r_{min}$ .

**Output:** Minimal feasible path parameters (RS or LS).

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

**while**  $j \leq m \Rightarrow$  vehicle hasn't visited all the targets **do**

1.  $(p_s, \psi_s) \leftarrow (p, \psi)$ ;
2.  $p_e \leftarrow t_j$ ;
3. Compute the length of two possible paths (RS and LS) between the vehicle's current position  $p_s$  and  $p_e$ :  $\{L_{RS}, L_{LS}\}$ ;
4. Choose the path that has the minimum length:  
 $L \leftarrow \arg \min\{L_{RS}, L_{LS}\}$ ;
5. Compute and output chosen path parameters:  $c_i, \eta_s, \eta_e, \eta_v$  and  $d_{C_i}$ .
- if**  $\langle \eta_v \rangle_i \geq d_{C_i} \Rightarrow$  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:  $p_b, \psi_p, \dot{\psi}_p$ ;
- 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**

---

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 3.10). 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

$$(p - p_e)^T \vec{n}_t \geq 0. \quad (3.22)$$

Algorithm 1 summarizes the above described strategy for a single vehicle to visit a group of moving targets in a given order  $s_t$ . Further implementation details are presented in Appendix C.

Figure 3.11 shows a numerical simulation result where a UAV moving at  $V = 30\text{m/s}$  with minimum turning radius  $r_{min} = 200\text{m}$  tracks three targets heading North at constant speed  $\|v_t\| = 15\text{m/s}$  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 (3.12) to generate its turn rate commands. Additionally, Figure 3.11 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 3.11 is computed considering that the starting position corresponds to the previous autonomous vehicle true interception coordinates as detailed in Figure 3.12. From this observation, it can be inferred that performance may be improved if the target's heading and velocity can be estimated to then obtain the expected interception position. Algorithm 2 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 3.13). A detailed description of Algorithm 2 and formal convergence conditions under which the vehicle reaches its moving target are presented along with a performance metric (with respect to the interception time) in Appendix C.

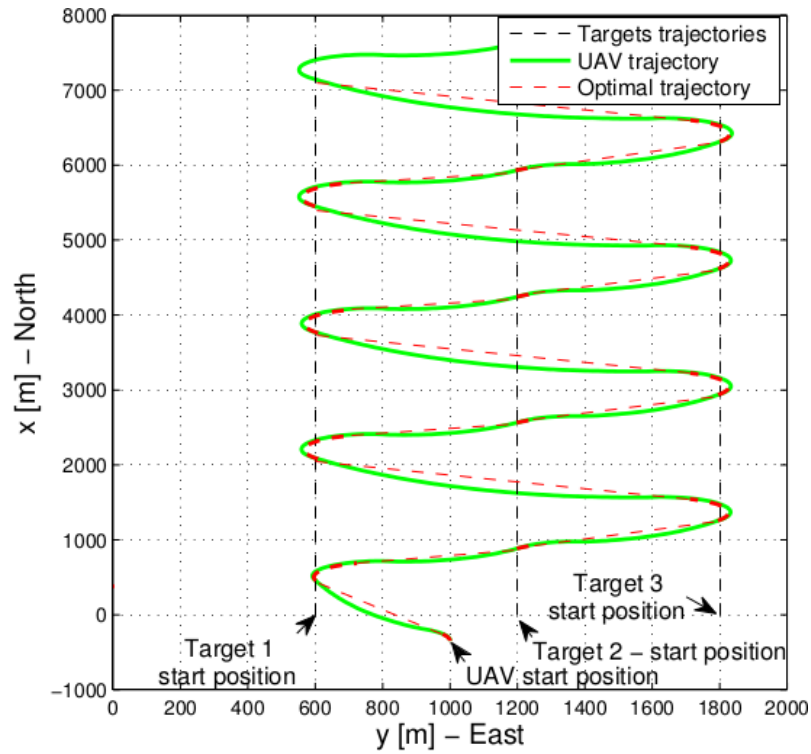


Figure 3.11: Multiple targets tracking numerical simulation: An autonomous vehicle tracks three targets heading North at constant speed using Algorithm 1 and the MPF control law given by equation (3.12).





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

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

$$\psi_t \sim \mathcal{N}(0, 0.03).$$

Distance and angular errors are depicted in Figure 3.15 showing the good performance of the control strategy in the presence of relatively high communication latency (see Figure 3.18). 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 (3.22) was met) are presented in Figure 3.17. 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 onboard camera's footprint (typically onboard vision sensors have footprints wider than  $2r_{min}$ ). The magnitude of 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 C.4 - further details can be found in Appendix C). Figure 3.16 shows that the

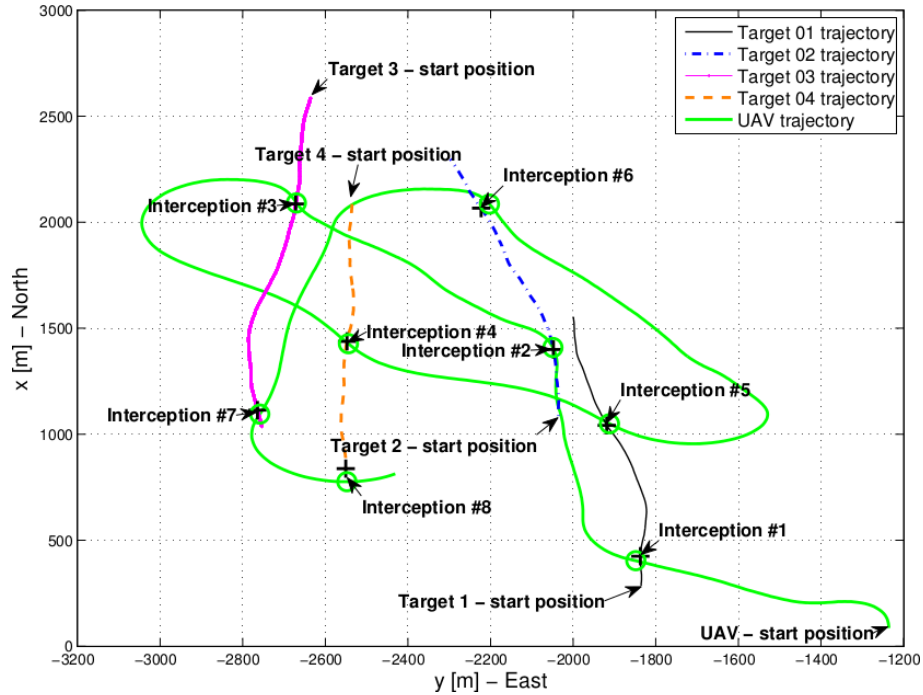


Figure 3.14: Multiple targets tracking flight test: Aircraft's trajectory following multiple targets on the ground.

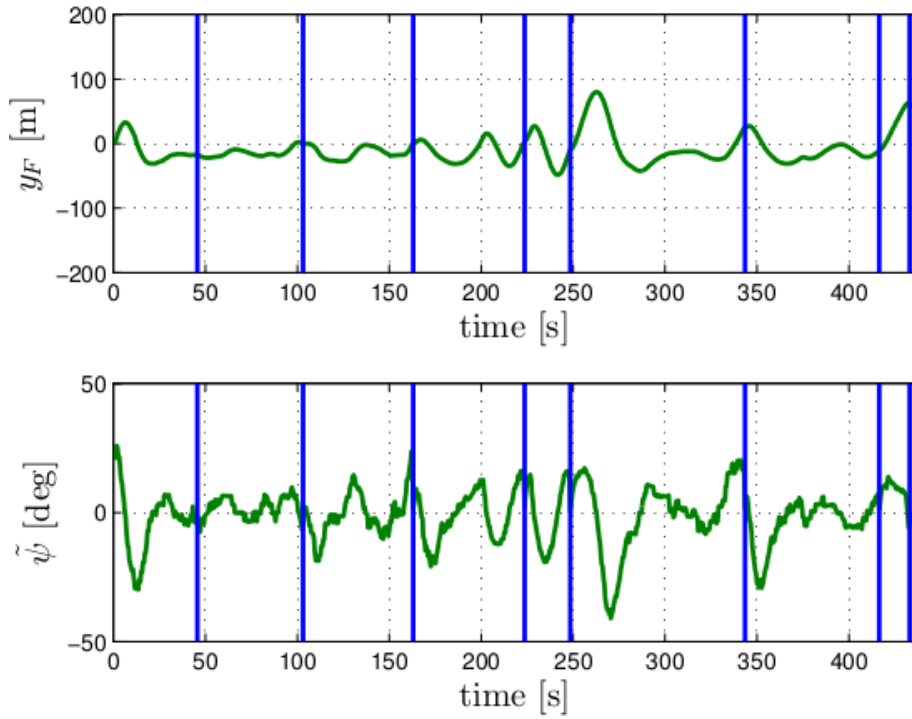


Figure 3.15: Multiple targets tracking flight test: Distance and heading errors. The vertical lines indicate the time instants at which each target interception has occurred.

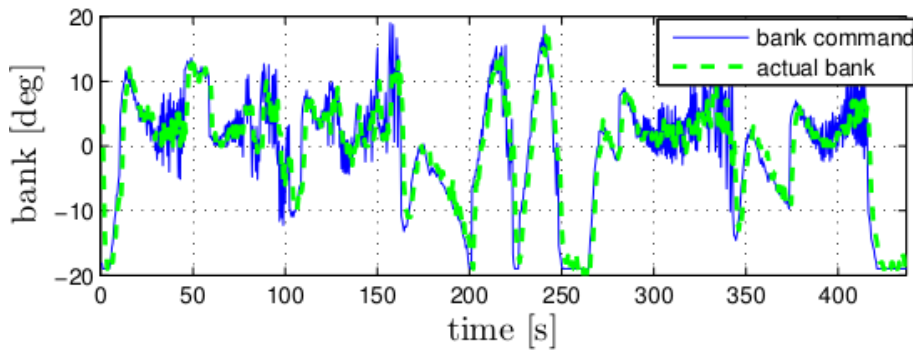


Figure 3.16: Multiple targets tracking flight test: Commanded and actual bank.

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

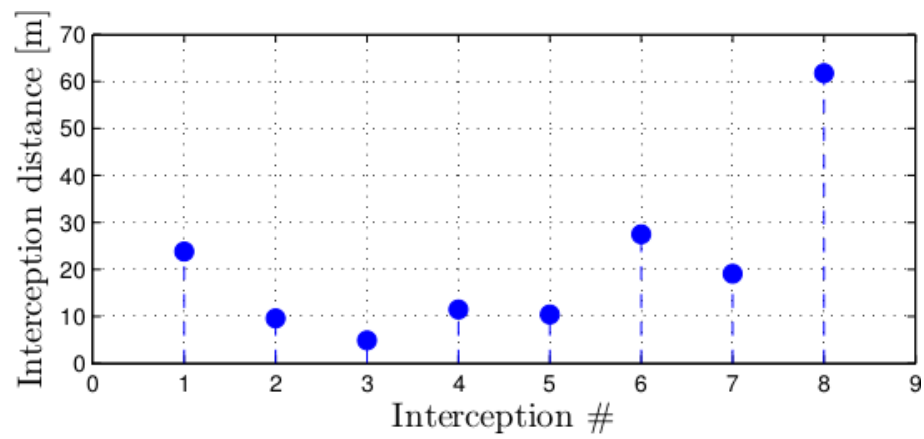


Figure 3.17: Multiple targets tracking flight test: Actual distance between UAV and current target at time of interception.

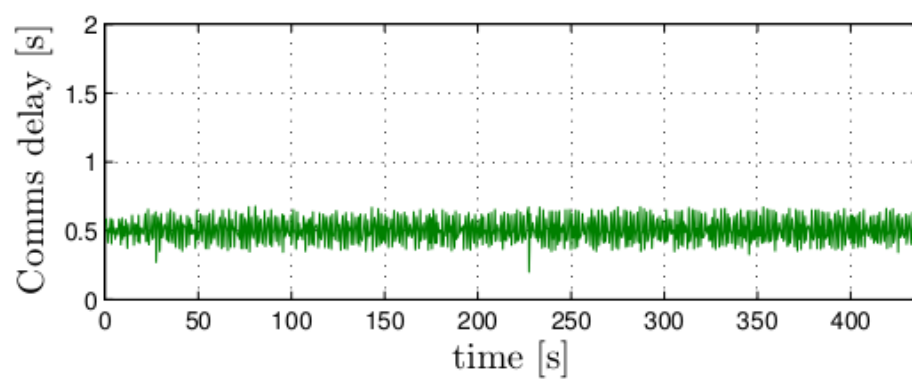


Figure 3.18: Multiple targets tracking flight test: Communications latency.

### 3.4 Application to convoy protection problems

This section describes how the MPF formulation presented in Section 3.2 can be applied to convoy protection problems, using the MPF degree of freedom to set the desired path's angular velocity.

#### 3.4.1 Convoy protection problem formulation

Consider a fixed-wing UAV equipped with an onboard camera that can monitor a disk of radius  $r_c$  on the ground irrespectively of the UAV attitude, i.e., assume that the onboard camera is attached to a gimbaled structure and is always looking down, independently of the attitude of the UAV. For convenience, ground convoys are considered to be a point located at the centroid of the convoys in the  $\vec{x} - \vec{y}$  plane. Similar to [6], it is considered that a successful convoy protection is being achieved when the centroid of the convoy is visible to the UAV at any time, i.e., if the distance between the projection of the UAV onto the  $\vec{x} - \vec{y}$  plane and the centroid of the convoy is less than or equal to  $r_c$  (see Figure 3.19).

Typically, in order to ensure a high level of image resolution, the onboard sensors may have a narrow field of view which, depending on the flight altitude, may impose an observation disk radius  $r_c$  smaller than the minimum turning radius of the UAV  $r_{min}$ . In this case, where  $r_{min} > r_c$ , depending on the convoy trajectory and speed relative to the UAV, it might be possible to fly out of this circle, and therefore not be able to guarantee continuous convoy protection.

The problem addressed in this section can be formalized as follows:

*Consider a convoy that is arbitrarily moving with possibly time-varying linear and angular velocities and a single fixed-wing UAV with a kinematic constrained model given by equations (3.1). Derive a guidance algorithm that sets the UAV speed  $V$  and angular velocity  $\omega$  such that the time that the convoy center remains within the UAV sensor footprint (that is assumed to be smaller than the UAV minimum turning radius) is maximized. Furthermore, provide conditions under which the proposed scheme guarantees continuous convoy protection (i.e., the UAV will always be inside the convoy circular region of interest).*

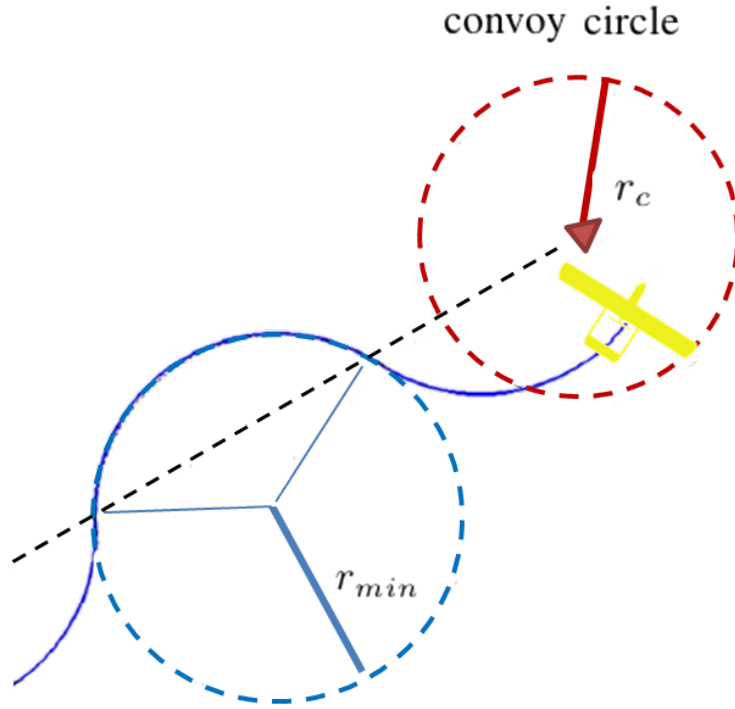


Figure 3.19: Convoy protection problem formulation illustration. The convoy center is represented by the red triangle. Adapted from [6]. The case where the minimum turning radius of the UAV ( $r_{min}$ ) is greater than  $r_c$  is considered.

### 3.4.2 Convoy protection using MPF

Notice from the MPF control law (see equation (3.12)) that  $\dot{\psi}_d$  and  $\dot{\ell}$  also depend on the variables  $v_d$  and  $\omega_d$ . Thus, intuitively, it is possible to command the desired path linear and angular velocities ( $v_d$  and  $\omega_d$ ) to ensure that the condition (3.9) always holds and thus the MPF problem is well posed, for any given path's curvature  $\kappa(\ell)$ . In other words, it is *in principle* possible to make the UAV follow a given desired path geometry (with a given path curvature  $\kappa(\ell)$ ) that would not be possible using the classical path following methods (where  $v_d = \omega_d = 0$ ). This degree of freedom will be explored such that the path movement is chosen to ensure that a UAV running the MPF control law given by equation (3.12) converges to that path, thus providing convoy protection.

Consider a lemniscate path that is contained within a circle of radius  $r_c$  as shown in Figure 3.20. The proposed strategy is to make the UAV follow a moving lemniscate path centered at the target/centroid of convoy position  $p_t$  (thus, by definition  $p_0 = p_t$ ,  $v_d = v_t$  and  $\psi_t = \psi_d$ ), where  $v_t$  and  $\psi_t$  are the target/centroid of convoy velocity and heading respectively. Notice however that by imposing this condition on  $v_d$ , the path's angular velocity  $\omega_d$  is still “free” to be chosen and may be used as a control input.

Under the same assumptions considered in [6], first consider that the convoy moves at a given constant velocity  $v_d$  (thus  $\dot{v}_d = 0$ ) and constant heading (thus  $\dot{\psi}_d = 0$ ). Moreover, in order to provide further insight to the proposed method (that is to command the desired path's angular velocity) first consider the case in which the path's angular velocity is set to zero ( $\dot{\psi}_p = \omega_d = 0$ ).

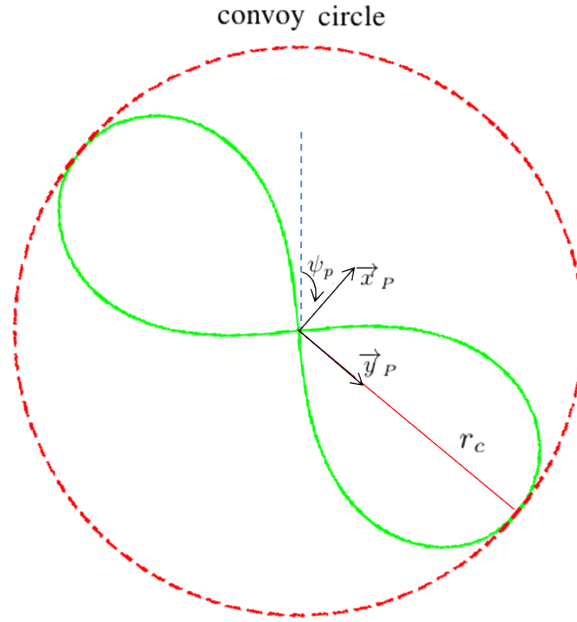


Figure 3.20: Convoy protection using the MPF method: Relevant variables.

Intuitively, as in the case considered in [6], the UAV, due to its kinematic constraints, will not be able to follow the desired lemniscate path unless a given lower bound on the target speed is observed, which can be computed as follows.

Considering the above assumptions, condition (3.9) can be rewritten as

$$\begin{aligned}
 & \left| \kappa(u) \dot{\ell} \left( 1 - \frac{\|v_d\| \cos(\psi_d - \psi_f)}{V \sqrt{1 - \left( \frac{\|v_d\| \sin(\psi_d - \psi_f)}{V} \right)^2}} \right) \right| \leq \omega_{max} \\
 \Leftrightarrow & \left| \frac{r_c}{V} \right| \left| \kappa(u) \dot{\ell} \left( 1 - \frac{\|v_d\| \cos(\psi_d - \psi_f)}{V \sqrt{1 - \left( \frac{\|v_d\| \sin(\psi_d - \psi_f)}{V} \right)^2}} \right) \right| \leq \frac{r_c}{r_{min}} \quad (3.23)
 \end{aligned}$$

where (see equation (3.5))

$$\dot{\ell} = V \left( \sqrt{1 - \left( \left( \frac{\|v_d\|}{V} \right)^2 \sin^2(\psi_d - \psi_f) \right)} - \frac{\|v_d\| \cos(\psi_d - \psi_f)}{V} \right),$$

and the desired lemniscate path curvature  $\kappa$  and tangential angle  $\psi_f$  are (see Appendix B)

$$\kappa(u) = \frac{-3\sqrt{2} \cos(\langle u \rangle)}{r_c \sqrt{3 - 2 \cos(2 \langle u \rangle)}}, \quad (3.24)$$

$$\psi_f(\psi_p, u) = 3 \arctan(\sin(\langle u \rangle)) + \psi_p, \quad (3.25)$$

with  $\langle u \rangle \triangleq u \bmod 2\pi$  and mod is the modulus operator that makes  $u \in [0, 2\pi]$ . The parameter  $u$  is related to the path length  $\ell$  through  $\ell(u) = \sqrt{2}r_c \int_0^u [3 - \cos(2u)]^{-\frac{1}{2}} du$  [52]. Note that from the above definition, a given fixed  $u$  parametrizes a point on the desired path expressed in the path frame.

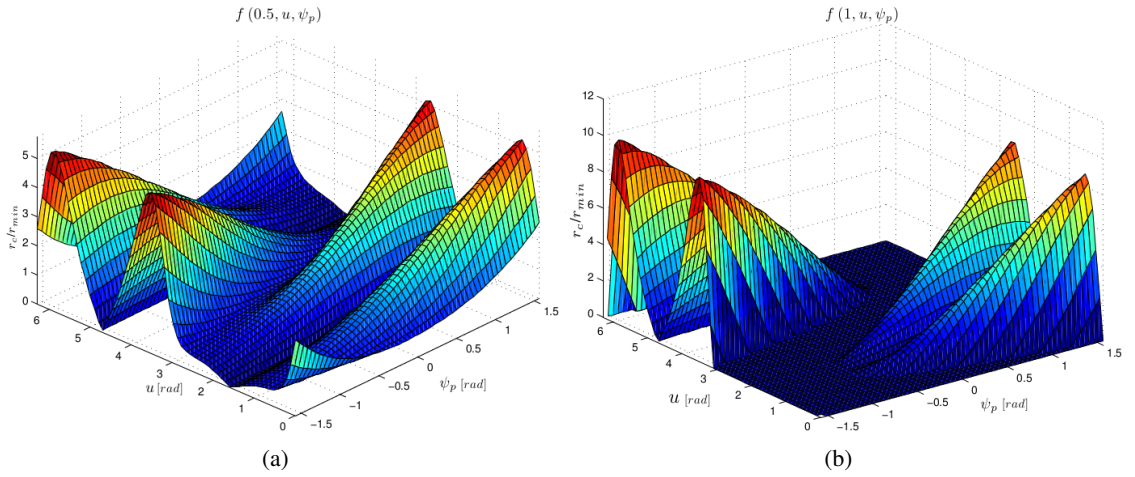


Figure 3.21: Minimum feasible turning radius  $r_c/r_{min}$  for a UAV to follow a lemniscate path contained within a circle of radius  $r_c$ , considering: a)  $\frac{v_d}{V} = 0.5$  and b)  $\frac{v_d}{V} = 1$ .

Let the left-hand side of equation (3.23) be designated by  $f(\frac{v_d}{V}, u, \psi_p)$ . From the considered convoy protection problem formulation, one must ensure that  $f(\frac{v_d}{V}, u, \psi_p) \leq 1$ . A first observation is that the maximum of  $f(\frac{v_d}{V}, u, \psi_p)$  is independent of  $\psi_d$ , which means that the target can move in a straight line in any direction  $\psi_d$  without introducing any additional constraint to the problem.

An analytic solution of equation (3.23) with respect to  $\frac{v_d}{V}$  (in order to compute the lower bound on the target speed) can not be derived. Figure 3.21 illustrates  $f(0.5, u, \psi_p)$  and  $f(1, u, \psi_p)$  for  $u \in [0, 2\pi]$ ,  $\psi_p \in [-\frac{\pi}{2}, \frac{\pi}{2}]$  and  $\frac{v_d}{V}$  equal to 0.5 and 1, respectively. It is assumed, without loss of generality (for the case of a convoy moving in straight line), that  $\psi_d = 0$ . This leads to the following conclusions:

1. For a constant  $\psi_p$  (which follows from the fact that  $\omega_d$  is set to zero), the UAV will increase  $u$  at all the times (because from the path following definition one has  $\dot{u} > 0$ ) and thus one obtains  $f(\frac{v_d}{V}, u, \psi_p) \geq 1$  for every constant  $\psi_p \in [-\frac{\pi}{2}, \frac{\pi}{2}]$ , as it can be seen from Figure 3.21. Therefore, it is not possible for the UAV to follow the desired lemniscate path contained within the circle  $r_c$  keeping the desired path's angular velocity  $\dot{\psi}_p = \omega_d$  always equal

to zero, even for the case where  $\frac{v_d}{V} \approx 1$ <sup>5</sup>;

2. There exists a lower bound on  $\frac{v_d}{V}$  from which there exists a  $u \in [0, 2\pi]$  and a  $\psi_p \in [-\frac{\pi}{2}, \frac{\pi}{2}]$  such that  $f(\frac{v_d}{V}, u, \psi_p) \leq 1$ . Thus, for every path point parametrized by  $u$ , it is possible to compute a desired path orientation  $\psi_p$  as a function of  $u$ , such that from a given lower bound on the  $\frac{v_d}{V}$  ratio,  $\psi_p(u) \rightarrow f(\frac{v_d}{V}, u, \psi_p(u)) \leq 1$ . In practice, one should use the degree of freedom of the path's orientation  $\psi_p$  by controlling the angular velocity  $\dot{\psi}_p = \omega_d$  (subject to the condition (3.9)) in order to make the UAV to be able to follow the moving geometric path.

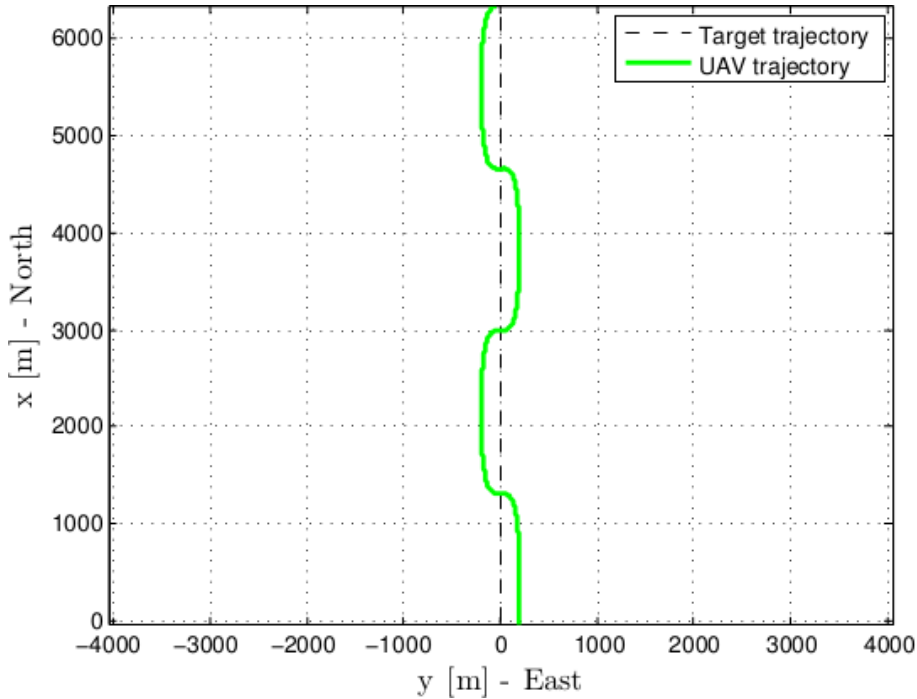


Figure 3.22: UAV trajectory following a moving lemniscate path with  $\frac{v_d}{V} = 0.9$ ,  $\psi_d = 0$  and  $\psi_p = 0$  without imposing any kinematic constraint on the UAV.

Figure 3.22 provides further intuition to the above conclusions. It presents a simulation example where a UAV follows a moving lemniscate path centered at the convoy center that is initially located at the inertial frame origin and starts moving with constant heading and velocity ( $\psi_d = 0$  and  $\frac{v_d}{V} = 0.9$  respectively) towards North. The desired geometric path orientation is kept constant ( $\dot{\psi}_p = \psi_p = 0$ ) and no kinematic constraints for the UAV were imposed. For a long period of time (due to the high ratio  $\frac{v_d}{V}$ ) the UAV flies almost parallel to the target. However, at a given path point, the corresponding path curvature makes the UAV to change its bank angle at a very high rate, making it impossible to comply with its actual kinematic constraints. If one could control the path's angular velocity  $\omega_d$ , it would be possible to smooth the obtained trajectory (considering the same geometric path and  $\frac{v_d}{V}$  ratio) in order to comply with condition (3.9) for all path points

<sup>5</sup>Note that for the case where  $\frac{v_d}{V} = 1$ , one obtains  $\dot{\ell} = 0$  and thus the UAV would follow a given fixed path point and move in a straight line parallel to the target (see also Figure 3.22 and its description).



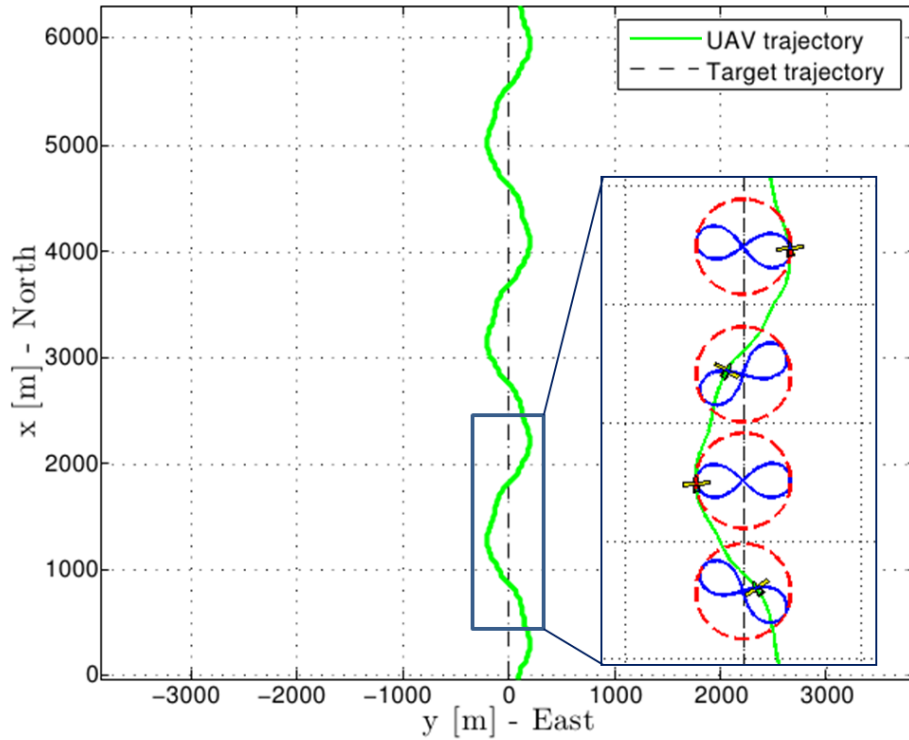


Figure 3.23: UAV trajectory following a moving lemniscate path with  $\frac{v_d}{V} = 0.9$ ,  $\psi_d = 0$  using Algorithm 3 to control the desired geometric path angular velocity  $\dot{\psi}_p = \omega_d$  and orientation angle  $\psi_p$ . The obtained trajectory complies with the UAV kinematic constraints.

parametrized by  $\ell$ . This is exactly the idea proposed to solve the considered convoy protection problem. Algorithm 3 describes how it can be implemented.

Figure 3.23 shows a numerical simulation result considering the same example scenario illustrated in Figure 3.22 (where a UAV follows a moving lemniscate path centered at the target position with  $\frac{v_d}{V} = 0.9$  and  $\psi_d = 0$ ) using Algorithm 3 to control the desired geometric path angular velocity  $\dot{\psi}_p = \omega_d$  and orientation angle  $\psi_p$ . The convoy circle is depicted in red (with a radius  $r_c$  that is equal to the UAV minimum turning radius  $r_{min}$ ) and the desired geometric path (at given sample time instants) is depicted in blue. The resulting UAV's trajectory (depicted in green) complies with its kinematic constraints and the UAV always remains inside the convoy circle<sup>6</sup>.

Under the same assumptions considered above (i.e., target moving with constant heading and speed) consider now the use of a circular path (where by definition  $\kappa(u) = \frac{1}{r_c}$  and  $\psi_f(\psi_p, u) = u + \psi_p + \frac{\pi}{2}$ ) for the proposed strategy. Figure 3.24 shows the minimum feasible turning radius  $r_c/r_{min} = f(\frac{v_d}{V}, u, \psi_p)$  for a UAV to follow a circular path centered at the target position, where  $\frac{v_d}{V} = 0.5$  and  $\frac{v_d}{V} = 1$ . Notice that, starting at a given path point (parametrized by  $u \in [0, 2\pi]$ ), one will always obtain  $f(\frac{v_d}{V}, u, \psi_p) \geq 1$  at a given path point, for every constant  $\psi_p \in [-\frac{\pi}{2}, \frac{\pi}{2}]$ . However, unlike the lemniscate path case, it is not possible to control the desired path's orientation<sup>7</sup>  $\psi_p(u)$  as a function of  $u$  in order to ensure that  $f(\frac{v_d}{V}, u, \psi_p(u)) \leq 1$ . Thus, one can conclude

<sup>6</sup>An illustration video for this simulation can be found in <https://www.youtube.com/watch?v=K7dkK6MwmAY>

<sup>7</sup>Note that the path orientation for the case of a circular path is, in practice, independent of  $\psi_p$ .

**ALGORITHM 3:** Returns the desired geometric path's kinematics.

**Input:** Vehicle start pose  $(p, \psi)$  and velocity  $V$ ; target's position  $p_0$ , velocity  $v_d$  and heading  $\psi_d$ ; desired geometric path parameters  $\kappa(\ell)$  and  $\psi_f(\psi_p, \ell)$ ; vehicle and convoy circle constraints  $r_{min}$  and  $r_c$  respectively.

**Output:** Desired path orientation  $\psi_p$ , angular velocity  $\omega_d$  and angular acceleration  $\dot{\omega}_d$ .

**Initialization:**

1. Set path's initial orientation:  $\psi_p \leftarrow \psi_d \Rightarrow$  the line that connects the two foci of the desired lemniscate path is initially set perpendicular to the target's heading.

**while** target tracking mission is engaged **do**

2. Compute the path parameter  $\ell$  corresponding to the closest to the UAV point of the path (see equation (B.5));

3.  $\omega_{limit} \leftarrow \text{argmax}\{\dot{\ell}(\omega_d)\}$  subject to conditions (3.7) and (3.9) and  $\dot{\ell}(\omega_d) \geq 0 \Rightarrow$  Compute path's desired angular velocity bound that complies with the UAV kinematic constraints;

4. **if** isempty( $\omega_{limit}$ )  $\Rightarrow$  if there is no solution for the desired path's angular velocity that complies with the UAV's kinematic constraints, consider only the geometric constraints given by equation (3.7)

$$\omega_{limit} \leftarrow \frac{V - v_d \sin(\psi_d - \psi_f)}{\sqrt{\Delta x^2 + \Delta y^2} \left| \sin\left(\psi_f + \arctan\left(\frac{\Delta y}{\Delta x}\right)\right) \right|}$$

**end**

5. Compute  $\omega_d$  using a proportional control law

**if**  $u \in [\pi, 2\pi]$

$$\omega_d \leftarrow k_p (\psi_d - \psi_p + \frac{\pi}{6}) \text{ subject to } |\omega_d| \leq |\omega_{limit}|;$$

**else**

$$\omega_d \leftarrow k_p (\psi_d - \psi_p - \frac{\pi}{6}) \text{ subject to } |\omega_d| \leq |\omega_{limit}|;$$

**end**

6.  $\dot{\omega}_d \leftarrow \frac{d\omega_d}{dt} \Rightarrow$  Compute path's angular acceleration;

7.  $\psi_p \leftarrow \int \omega_d dt \Rightarrow$  Compute path's orientation  $\psi_p$ ;

8. Update  $p \Rightarrow$  Update UAV's current position using the MPF control law given by equation (3.12) and the UAV kinematic equations (3.1);

9. Return to 2;

**end**

that, given the considered kinematic constraints, the circular path is not suitable to be used as a solution to the here proposed problem.

The use of a lemniscate path has the following desirable features. First, it allows the UAV to fly over the target/convoy center position periodically, depending on the target's dynamics and the desired path commanded velocity  $\omega_d$ . Additionally, a lemniscate has been shown to be an effective way for an autonomous aircraft to provide surveillance of a slower target [46, 63]. Despite some of the previously mentioned methods propose a change on the desired path width depending on the target and UAV speed ratio [46] or the UAV's kinematic constraints [63], by controlling the lemniscate orientation angle  $\psi_p$  (see Figure 3.20) the solution presented here allows the UAV to stay closer to the target/convoy center, considering the same UAV turn rate constraints (see Figures 3.22 and 3.23). Controlling the desired path's orientation instead of its width also ensures smoother trajectories of the UAV when a change on the target's velocity and/or heading occurs. Additionally the presented solution is, to the author's best knowledge, the only that encompasses the problem where the UAV minimum turning radius is larger than the radius of the convoy circular region of

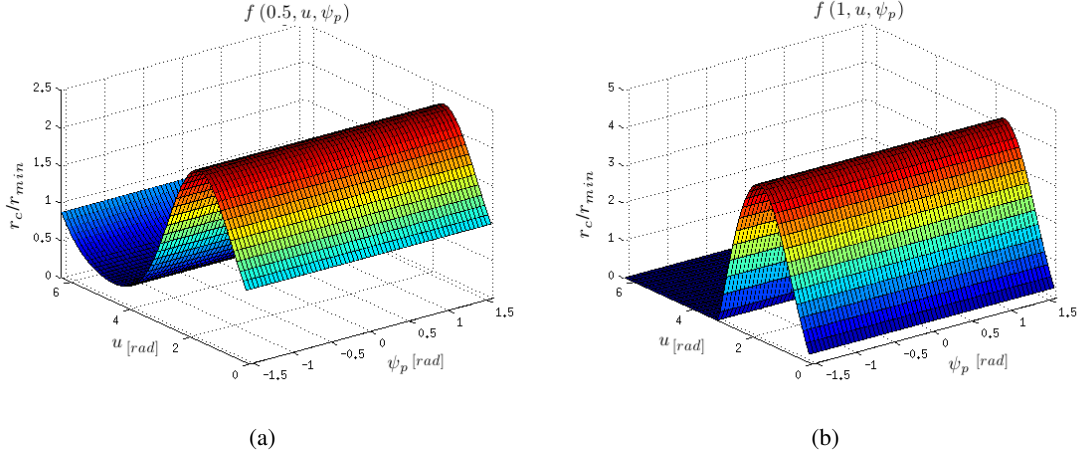


Figure 3.24: Minimum feasible turning radius  $r_c/r_{min}$  for a UAV to follow a circular path considering: a)  $\frac{v_d}{V} = 0.5$  and b)  $\frac{v_d}{V} = 1$ .

interest and explicitly takes into account the UAV kinematic constraints. Furthermore, notice that the desired path's angular velocity  $\omega_d$  is computed in order to maximize the relative velocity of the UAV with respect to the desired path (given by  $\dot{\ell}$  - see Step 3 of Algorithm 3), which in practice minimizes the time between each pass above the convoy center position. Finally, note that in the case of a circular path centered at the target position (which is the most common approach in the literature [89]) the UAV's velocity vector (at a given path point) will be pointing exactly in the opposite direction of the target/convoy center velocity vector. The use of a lemniscate path and Step 5 in Algorithm 3 ensures that this situation never occurs<sup>8</sup> which, in practice, translates into less kinematic constraints due to the desired path's geometry.

The best suited path for the proposed problem is still an open issue. Nonetheless, the presented problem formulation simplifies this task, since a given path geometry to be studied (with a given curvature  $\kappa(u)$  and tangential angle  $\psi_f(\psi_p, u)$ ) can immediately be used by replacing equations (3.24) and (3.25) by those corresponding to the desired path geometry.

### 3.4.3 Numerical simulation results

Three main results for the proposed convoy protection algorithm are presented in this section. First, a lower bound on the convoy/target velocity ratio with respect to the UAV ground speed  $\frac{v_d}{V}$  as a function of the ratio between the convoy circle radius and the UAV minimum turning radius  $\frac{r_c}{r_{min}}$  is compared with the one presented in [6] under the same assumption that the target moves with constant heading and velocity. Then, it is shown how these restrictions on the convoy movements can be relaxed using the proposed method through a simulation example that includes time varying linear and angular convoy velocities. Finally, a performance metric is proposed, considering the case where no lower bound on the convoy speed is imposed.

<sup>8</sup>Note that the angular displacement  $|\psi_d - \psi_p|$  between the target's heading and the desired geometric path is limited to  $\frac{\pi}{6}$ .

Consider a convoy moving with constant heading and velocity. As in the case considered in [6], due to the UAV kinematic constraints, the UAV will not be able to follow the desired lemniscate path within the convoy circle (given that  $r_c < r_{min}$ ) unless a given lower bound on the target/convoy speed relative to the UAV (expressed by the  $\frac{v_d}{V}$  ratio) is achieved.

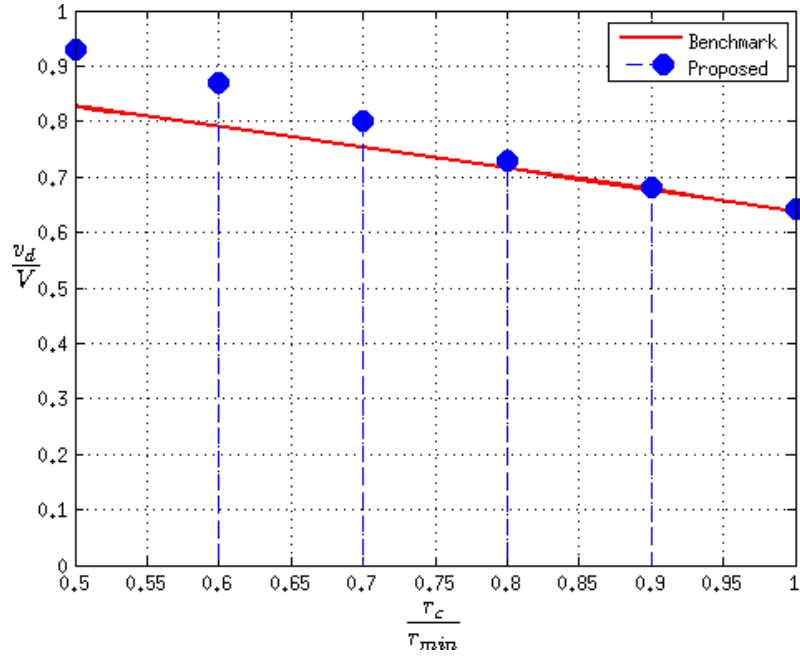


Figure 3.25: Lower bound for  $\frac{v_d}{V}$  as a function of  $\frac{r_c}{r_{min}}$  for the proposed method and the considered benchmark [6].

Due to the complexity of the above derived equations, an analytic solution for equation (3.23) providing the lower bound for  $\frac{v_d}{V}$  as a function of  $\frac{r_c}{r_{min}}$  could not be found. In this section, numerical simulations for specific cases are provided. Figure 3.25 illustrates the lower bound results (depicted in blue) using the following approach:

- For a predefined  $\frac{r_c}{r_{min}}$ , start with an initial guess for  $\frac{v_d}{V}$  and run Algorithm 3 until the UAV has flown over the target position  $n$  times (in this case  $n = 4$  was used). Stop if at any given path point, there is no solution for the desired path's angular velocity that complies with the UAV's kinematic constraints  $\Rightarrow |\omega_{limit}| = \text{empty solution}$  (cf. Step 4 of Algorithm 3);
- Decrease the value of  $\frac{v_d}{V}$  and repeat the previous step until the stop condition is achieved.

For the case where  $\frac{r_c}{r_{min}} \geq 0.8$ , these results are similar to the ones presented in [6] (used as a benchmark - depicted in red). For  $\frac{r_c}{r_{min}} < 0.8$  it can be seen that the obtained lower bound gradually increases (with respect to the benchmark) as  $\frac{r_c}{r_{min}}$  decreases. This is mainly related to the chosen desired path geometry and one can arguably infer that a better performance could be achieved if a more suited path was used. Moreover, note that the considered benchmark only applies to convoys moving with a specific constant heading and speed configuration (hence it corresponds to

the optimal solution) while the computed lower bound for the proposed method always holds for the case of a convoy moving with constant heading and time-varying speed.

Through a simulation, it is now illustrated how the proposed method can be used to relax the convoy movements restrictions described in the literature, allowing it to have time-varying linear and angular velocities. In this simulation, the goal is to make the UAV to track a ground convoy moving with time-varying linear and angular velocities by following a lemniscate path centered at the convoy center position while computing the desired path's angular velocity  $\omega_d$  using Algorithm 3 and the UAV yaw rate  $\dot{\psi}$  using the MPF control law given by equation (3.12), to make the UAV converge to and track the desired moving path. The UAV speed is set to  $V = 20\text{m/s}$ , its minimum turning radius is set to  $r_{min} = 200\text{m}$ , and thus  $\omega_{max} = 0.1\text{rad/s}$ .

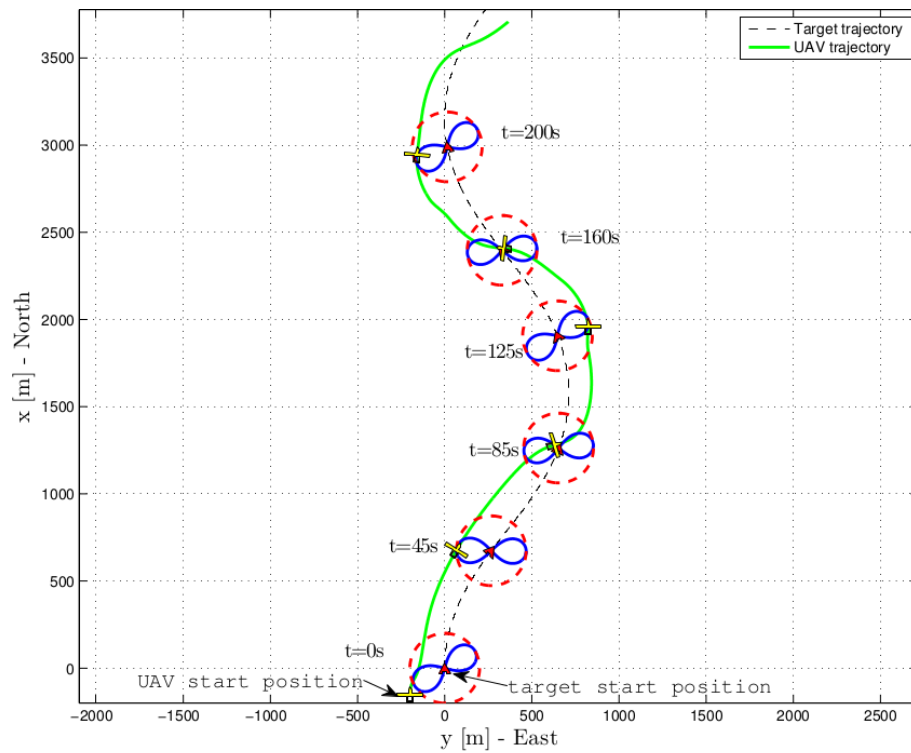


Figure 3.26: Convoy protection numerical simulation: UAV trajectory following a convoy center (the target in the figure) moving with time-varying linear and angular velocities.

The convoy center was moving according to

$$\begin{aligned} (p_{tx}, p_{ty}, \psi_t, \|v_t\|)|_{t=0} &= (0\text{m}, 0\text{m}, 0, 17\text{m/s}) \\ \|\dot{v}_t\| &= 0.01 \sin(-0.07t) \text{m/s}^2 \\ \dot{\psi}_t &= 0.02 \cos(0.03t) \text{rad/s} \end{aligned} \quad (3.26)$$

where  $v_t$  corresponds to the target velocity and  $\psi_t$  is the target heading. In order to attach the desired path to the convoy center the path frame is set with  $p_0|_{t=0} = [p_{tx} \ p_{ty}]|_{t=0}$ ,  $v_d = v_t$  and  $\psi_d = \psi_t$ . The MPF controller gains were set to  $g_1 = 0.22$  and  $g_2 = 0.0002$ .

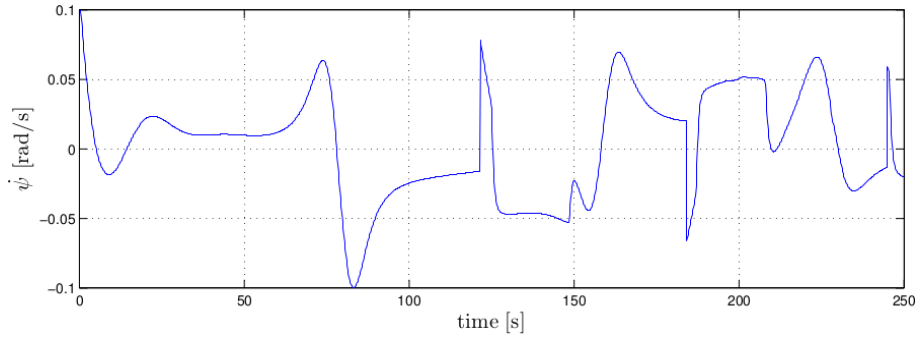


Figure 3.27: Convoy protection numerical simulation: UAV's angular velocity.

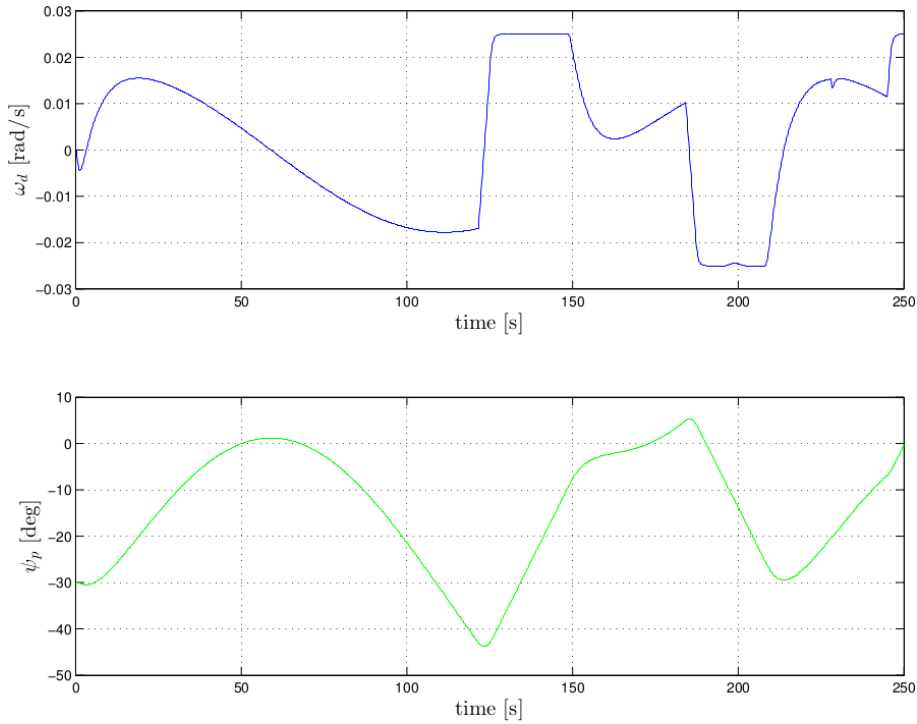


Figure 3.28: Convoy protection numerical simulation: Desired path angular velocity  $\omega_d$  and orientation angle  $\psi_p$  obtained using Algorithm 3.

Figure 3.26 presents the obtained UAV's trajectory. The UAV, convoy center, convoy circle and desired moving path (depicted in blue) positions at sample time instants are also presented. The UAV always remains inside the convoy circle of interest (depicted in red) and its resulting trajectory (depicted in green) complies with its kinematic constraints (i.e.,  $|\dot{\psi}| < \omega_{max}$  - see Figure 3.27).

The desired path angular velocity  $\omega_d$  and orientation angle  $\psi_p$  obtained using Algorithm 3 are presented in Figure 3.28. The distance and angular errors (respectively  $y_F$  and  $\tilde{\psi}$ ) of the UAV with respect to the desired moving path are depicted in Figure 3.29 showing the good performance of the control strategy.

The lower bound presented in Figure 3.25 was obtained by considering the case of a convoy moving with constant heading and speed. Despite it has been demonstrated through a simulation

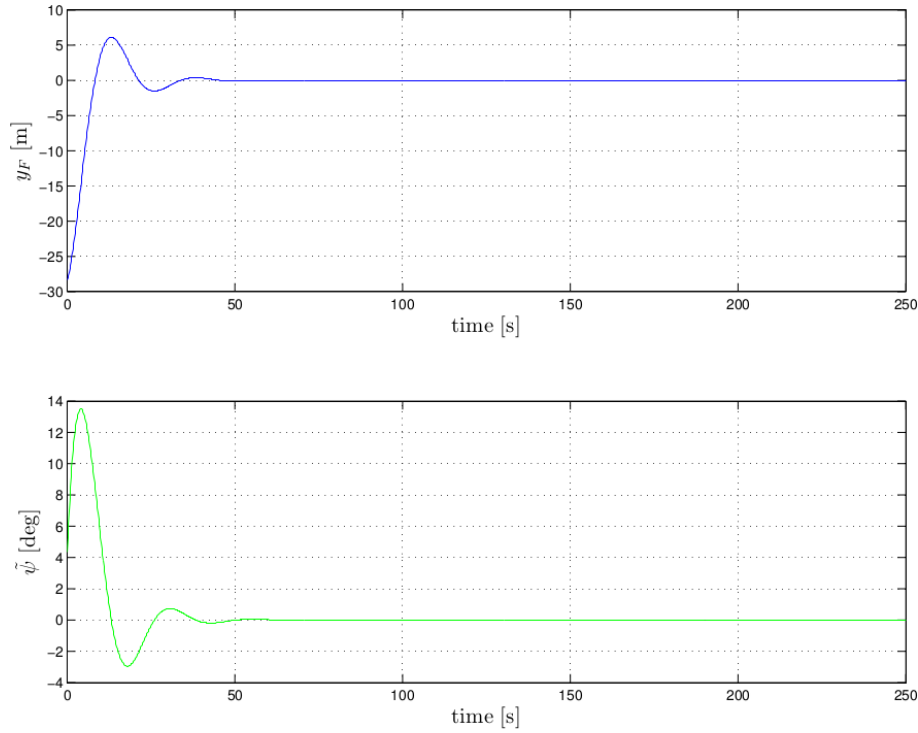


Figure 3.29: Convoy protection numerical simulation: Position and heading errors.

example that the proposed method allows to relax the constraint on the convoy movements, a solution to obtain  $\omega_{limit}$  that complies with the UAV kinematic restrictions (given by conditions (3.7) and (3.9)) for the general case of time-varying heading and speed of the convoy is not so straightforward to compute because it depends on the geometry of the problem and might not always exist. In that case,  $\omega_{limit}$  is computed directly from condition (3.7) considering only the geometric constraints of the problem (see Step 4 of Algorithm 3). In order to ensure a favourable relative geometry (i.e., the relative orientation angle between the desired path and the convoy heading that allows the existence of solution for  $\omega_{limit}$ ) for the problem, it is desirable to have a small amplitude of the path orientation  $\psi_p$  with respect to the convoy velocity vector orientation  $\psi_d$ . Figure 3.21 provides a graphical illustration of this idea. This is implemented by Step 5 of Algorithm 3 where the path orientation angle  $\psi_p$  relative to the target heading  $\psi_d$  is contained within the interval  $[-\frac{\pi}{6}, \frac{\pi}{6}]$ , using a proportional control law with  $k_p = 0.3$ .

Additionally, consider a scenario where the target/convoy kinematics does not always comply with the lower bound for the  $\frac{v_d}{V}$  ratio presented in Figure 3.25, and thus, continuous convoy protection cannot be provided using a single UAV even in the case of a convoy moving with constant heading. Similarly to the strategy proposed in [6] one can consider a multi-UAV coordination approach together with a timing strategy to schedule the UAVs such that, at any time instant, one of the UAVs is inside the convoy circle. This, however, is out of the scope of this thesis. Nevertheless, an interesting performance result in a realistic scenario where no bounds on the convoy movements are considered (except that  $\|v_d\| < V$ ) is to compute the average time that a single fixed-wing UAV remains inside the convoy circle relative to the total simulation time, denoted by

$Av_t$ .

In order to compute the proposed performance metric 500 Monte Carlo simulations were used, and each simulation lasted 300 seconds. The target/convoy started moving with random heading ( $\psi_t|_{t=0}$ ) according to a uniform distribution in the interval  $]-\pi, \pi[$ . The UAV's ground speed was chosen to be  $V = 20\text{m/s}$  and the UAV's minimum turning radius was set to  $r_{min} = 200\text{m}$ . In each simulation, the convoy center initial position  $p_t|_{t=0s}$  was set at the origin of the inertial frame and the UAV initial position was set at a distance  $d = r_{min}$  from the convoy center with the line of sight angle between the line that connects the convoy center of mass  $p_t|_{t=0s}$  to the UAV center of mass  $p_t|_{t=0s}$  and North  $\alpha_{LOS} = \psi_t|_{t=0} - \pi$ . The UAV initial heading  $\psi|_{t=0s}$  was set equal to the convoy initial heading  $\psi|_{t=0s} = \psi_t|_{t=0}$ . 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

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

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

with an output sample time set to 10s (simulation time). Two scenarios were considered for the proposed performance metrics. On the first one, convoy centre's initial speed was set to  $\|v_t\|_{t=0s} = 10\text{m/s}$  and  $\|v_t\|$  was limited to the interval  $[0, 19]$  [m/s]. For the second scenario, the convoy center initial speed was set to  $\|v_t\|_{t=0s} = 16\text{m/s}$  and  $\|v_t\|$  was limited to the interval  $[15, 19]$  [m/s]. The speed bounds imposed on the second scenario took into account the lower bound for the  $v_d/V$  ratio presented in Figure 3.25, thus providing a more adequate performance metric for the proposed strategy. On both scenarios, the  $\frac{r_c}{r_{min}}$  value was set equal to 1.

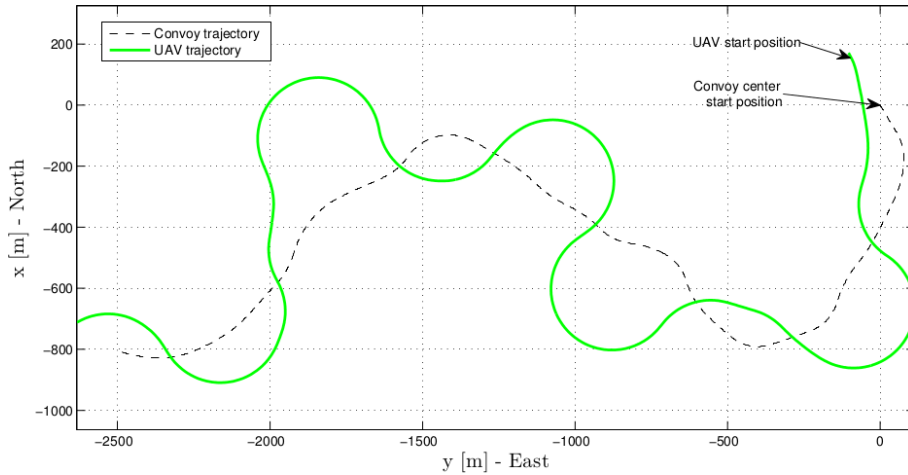


Figure 3.30: Convoy protection performance metric simulation: UAV and convoy center trajectories.

The performance metric was computed through

$$Av_t = \frac{1}{500} \sum_{i=1}^{500} \frac{t_{inside_i}(r_{ci})}{300}$$



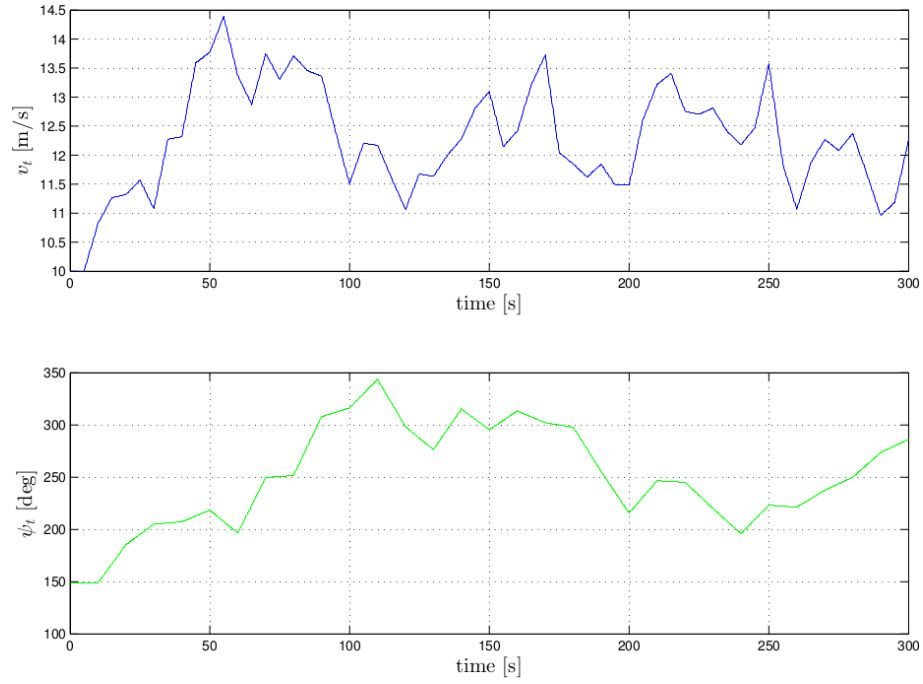


Figure 3.31: Convoy protection performance metric simulation: Convoy heading and speed.

where  $r_{c_i}$  is the convoy circle radius at simulation  $i$  and  $t_{inside_i}$  is total time the UAV remains inside  $r_{c_i}$  during simulation  $i$ .

Figure 3.30 shows the obtained results for the convoy and UAV trajectories (for a particular simulation) using the above defined variables for the first scenario. The corresponding convoy heading and speed are presented in Figure 3.31 illustrating the realistic scenario considered for the performance metric.

Table 3.2: Proposed method performance metrics.

	Scenario 1	Scenario 2
$Av_t$	0.66	0.88

From the obtained results presented in Table 3.2 for the two considered scenarios, one can conclude that the proposed strategy provides a versatile solution for the convoy protection problem with unconstrained movements.



## Chapter 4

# 3D Moving Path Following control design

This chapter formally extends the previously presented MPF methods to the case of desired geometric paths that can be moving in a three dimensional space with time-varying linear and angular velocities with respect to an inertial frame. A quaternion-based MPF control law is derived at kinematic level for a fixed-wing unmanned aircraft vehicle. Formal convergence guarantees and validation using numerical and flight test results are presented, demonstrating the effectiveness of the proposed method.

### 4.1 3D Moving Path Following control law using quaternions

In this section the MPF control law for the general case of desired geometric paths moving with time varying linear and angular velocities in a three dimensional space, using quaternions (see [90, 91]) to parametrize the rotation matrices between reference frames is derived. The use of quaternions allows for removing the singularities associated to the Euler angles parametrization used in the implementations presented in Chapter 3. The adopted quaternion convention and corresponding fundamentals used in this thesis can be found in Appendix D.

Recall from Chapter 2 that  ${}^F v_{W_d}$  and  ${}^F v_W$  correspond respectively to the UAV's desired velocity vector and the actual one, written with respect to the  $\{F\}$  frame. Additionally, recall that a wind frame  $\{W\} = \{\vec{x}_W, \vec{y}_W, \vec{z}_W\}$  is located at the vehicle center of mass and with its  $\vec{x}_W$ -axis along the direction of the vehicle velocity vector, the  $\vec{y}_W$ -axis parallel to the  $\vec{x} - \vec{y}$  plane, normal to  $\vec{x}_W$ , and pointing to the right of an observer that moves in the same direction of the aircraft, and  $\vec{z}_W$ -axis orthogonal to the previous two (see Figure 1.1). The desired wind frame  $\{W_d\}$ , was defined similarly to  $\{W\}$ , with its  $\vec{x}_{W_d}$ -axis along the direction of  ${}^F v_{W_d}$ .

Let the unit quaternion representation [90] that corresponds to the rotation matrix  ${}^{W_d}R_W$  be

given by  $q(t) \triangleq \begin{bmatrix} q_0 & q_v^T \end{bmatrix}^T$ , where

$$\begin{aligned} q_0(t) &\triangleq \cos\left(\frac{\Phi(t)}{2}\right) \\ q_v(t) &\triangleq e_v(t) \sin\left(\frac{\Phi(t)}{2}\right), \end{aligned} \quad (4.1)$$

and  $\Phi(t) \in \mathbb{R}$  and  $e_v(t) = \begin{bmatrix} e_1(t) & e_2(t) & e_3(t) \end{bmatrix}^T \in \mathbb{R}^3$  denote the angle-axis parameters, that are computed as follows

$$\Phi = \arccos\left(\frac{{}^F v_W \cdot {}^F v_{W_d}}{\|{}^F v_W\| \cdot \|{}^F v_{W_d}\|}\right), \quad e_v = \frac{{}^F v_W \times {}^F v_{W_d}}{\|{}^F v_W \times {}^F v_{W_d}\|}.$$

Note from (4.1) that the unit quaternion is subject to the constraint  $q^T q = 1$  and that if the angular displacement ( $\Phi$ ) between  $\{W\}$  and  $\{W_d\}$  (i.e. the attitude error) goes to zero, then so will  $q_v(t)$ .

From the above definitions, one can verify that the angular velocity  ${}^W \omega_{W,F;W_d}^r$  is related to  $\dot{q}(t)$  via the differential equation [90]

$$\dot{q} = \frac{1}{2} B(q) {}^W \omega_{W,F;W_d}^r$$

where

$$B(q) \triangleq \begin{bmatrix} -q_v^T \\ q_0 I_3 - S(q_v) \end{bmatrix} \in \mathbb{R}^{4 \times 3}.$$

Returning to the MPF problem described in Section 1.1, consider now the following Lyapunov function

$$V_3 = g_1 p_F^T p_F + (1 - q_0)^2 + q_v^T q_v,$$

where  $g_1$  is a positive scalar. Recall that the kinematic controller's goal is to drive  $p_F$  to zero,  $q_0$  to 1 and  $q_v$  to zero. Differentiating  $V_3$  with respect to time yields

$$\begin{aligned} \dot{V}_3 &= 2g_1 p_F^T \dot{p}_F + q_v^T {}^W \omega_{W,F;W_d}^r \\ &= 2g_1 p_F^T \dot{p}_F + q_v^T \left( {}^W \omega_W - {}^W \omega_F - {}^W R_{W_d} {}^{W_d} \omega_{W_d} \right), \end{aligned}$$

where  ${}^W \omega_W$  will be used as the control input for the UAV. Since by definition (see equation (2.8))

$$p_F^T (-S({}^F \omega_F) p_F - {}^F v_F) = 0,$$

and using the results given by equation (2.9), one can rewrite  $p_F^T \dot{p}_F$  as

$$\begin{aligned} p_F^T \dot{p}_F &= p_F^T \left( {}^F R_W^W v_W - {}^F R_I (v_d + S(\omega_d) (p_d - p_0)) \right) \\ &= p_F^T \left( {}^F v_W - {}^F v_{W_d} \right). \end{aligned}$$

From the law of cosines [92] it can be shown that

$$\| {}^F v_W - {}^F v_{W_d} \|^2 = 2V^2 (1 - \cos(\Phi)),$$

and thus

$$\begin{aligned} \dot{y}_F &= \sqrt{2V^2 (1 - \cos(\Phi))} \cos \sigma_1 \\ \dot{z}_F &= \sqrt{2V^2 (1 - \cos(\Phi))} \cos \sigma_2, \end{aligned}$$

where  $\sigma_1$  and  $\sigma_2$  are the angles between the vector sum  ${}^F v_{W_d} - {}^F v_W$  and  $\vec{n}_1$  and  $\vec{n}_2$  respectively. Finally, setting the UAV angular velocity to

$${}^W \omega_W = {}^W \omega_F + {}^W R_{W_d} {}^{W_d} \omega_{W_d} - g_2 q_v - g_1 \begin{bmatrix} 0 \\ \frac{1}{q_{v_2}} z_F \dot{z}_F \\ \frac{1}{q_{v_3}} y_F \dot{y}_F \end{bmatrix} \quad (4.2)$$

with  $g_2$  a positive scalar, one obtains

$$\dot{V}_3 = -g_2 q_v^T q_v.$$

Positive scalars  $g_1$  and  $g_2$  assign relative weights between position and orientation errors.

Given the definition of  $V_3$  and the fact that  $\dot{V}_3 \leq 0$ , the errors  $p_F$  and  $q_v$  are bounded. Computing the second derivative of  $V_3$  one can easily verify that the boundedness of the state variables implies that  $\dot{V}_3$  is uniformly continuous. Hence, Barbalat's lemma [88] allows for the conclusion that  $\dot{V}_3$  and consequently  $q_v$  tend to zero as  $t \rightarrow \infty$ .

From the above definitions, rewriting

$$\begin{aligned} \dot{q}_v &= \frac{1}{2} \begin{bmatrix} q_0 I_3 - S(q_v) \end{bmatrix} \left( -g_2 q_v - g_1 \begin{bmatrix} 0 \\ \frac{1}{q_{v_2}} z_F \dot{z}_F \\ \frac{1}{q_{v_3}} y_F \dot{y}_F \end{bmatrix} \right) \\ &= -\frac{1}{2} \begin{bmatrix} q_0 & q_{v_3} & -q_{v_2} \\ -q_{v_3} & q_0 & q_{v_1} \\ q_{v_2} & -q_{v_1} & q_0 \end{bmatrix} \left( \begin{bmatrix} g_2 q_{v_1} \\ g_2 q_{v_2} + g_1 \frac{1}{q_{v_2}} z_F \dot{z}_F \\ g_2 q_{v_3} + g_1 \frac{1}{q_{v_3}} y_F \dot{y}_F \end{bmatrix} \right), \end{aligned}$$

differentiating  $\dot{q}_v$  with respect to time, and invoking the boundedness of the variables involved, one can apply once more Barbalat's lemma to conclude that  $\dot{q}_v$  tends to zero, which leads to the

conclusion that also  $y_F$  and  $z_F$  tends to zero as  $t \rightarrow \infty$ .

## 4.2 Application to following a lemniscate moving in three dimensions

This section presents numerical simulation and flight test results obtained for a UAV following a desired lemniscate path moving with time varying linear and angular velocities with respect to the inertial frame.

Let  $q_P$  be the unit quaternion representation of the path frame orientation  $\{P\}$  with respect to the inertial frame  $\{I\}$  and let  $q_W$  be the unit quaternion representation of the wind frame  $\{W\}$  orientation with respect to the inertial frame. Using the above definitions, one can relate the quaternion rates with their corresponding angular velocities [90] as

$$\begin{aligned}\dot{q}_P &= \frac{1}{2} \begin{bmatrix} 0 & \omega_d^T \end{bmatrix}^T \otimes q_P \\ \dot{q}_W &= \frac{1}{2} q_W \otimes \begin{bmatrix} 0 & {}^W\omega_W^T \end{bmatrix}^T\end{aligned}$$

where  $\otimes$  represents the quaternion product (please refer to Appendix D). For the numerical simulation, the path frame origin and orientation with respect to the inertial frame were initially set to

$$\begin{aligned}p_0|_{t=0s} &= \begin{bmatrix} 0m & 0m & 0m \end{bmatrix}^T \\ q_P|_{t=0s} &= \begin{bmatrix} \cos(\frac{\pi}{8}) & 0 & \sin(\frac{\pi}{8}) & 0 \end{bmatrix}^T,\end{aligned}$$

the wind frame start position and orientation with respect to the inertial frame were set to

$$\begin{aligned}p|_{t=0s} &= \begin{bmatrix} 0m & -150m & 50m \end{bmatrix}^T \\ q_W|_{t=0s} &= \begin{bmatrix} \cos(\frac{\pi}{4}) & 0 & 0 & \sin(\frac{\pi}{4}) \end{bmatrix}^T,\end{aligned}$$

and the desired path linear and angular velocities were set to

$$\begin{aligned}\dot{v}_d &= 0.1 \sin(0.05t) \begin{bmatrix} 1m/s^2 & 1m/s^2 & 0m/s^2 \end{bmatrix} \\ v_d|_{t=0s} &= \begin{bmatrix} 1m/s & 2m/s & -0.2m/s \end{bmatrix}^T \\ \dot{\omega}_d &= -6 \times 10^{-6} \sin(0.03t) \begin{bmatrix} 0rad/s^2 & 10^{-3}rad/s^2 & 1rad/s^2 \end{bmatrix}^T \\ \omega_d|_{t=0s} &= \begin{bmatrix} 0rad/s & 0rad/s & 0rad/s \end{bmatrix}^T,\end{aligned}$$

where  $t$  corresponds to the simulation time.

Figure 4.1 presents the generated path frame origin trajectory with respect to the inertial frame (depicted in red), the desired lemniscate path position at sample time instants (depicted in blue) and the corresponding UAV position at some sample time instants (depicted in yellow). Figures

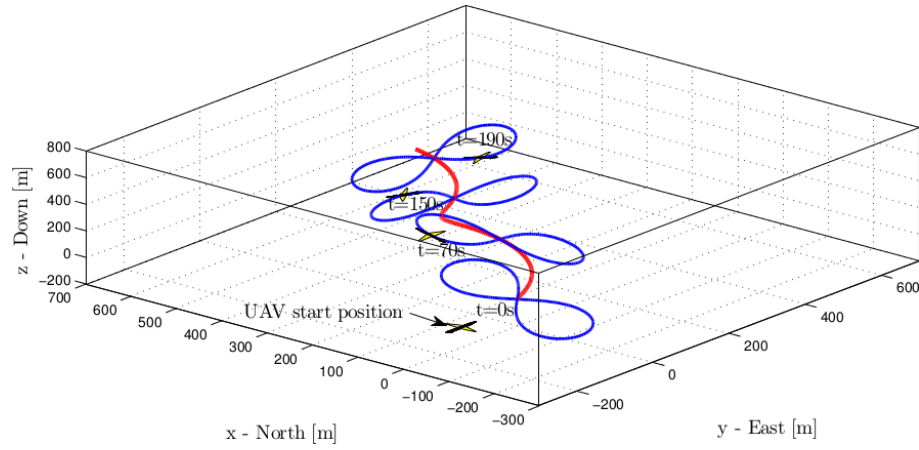


Figure 4.1: Lemniscate tracking numerical simulation: Path frame origin trajectory (red), desired lemniscate path (blue) and UAV position at sample time instants.

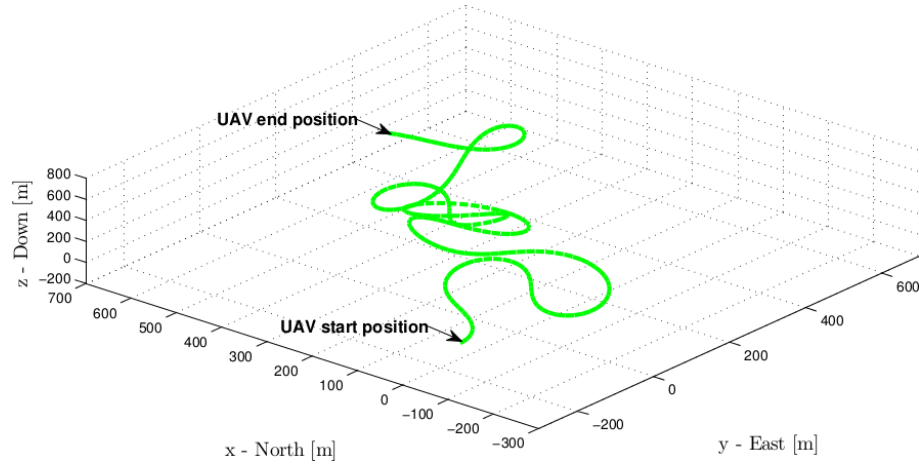


Figure 4.2: Lemniscate tracking numerical simulation: UAV trajectory.

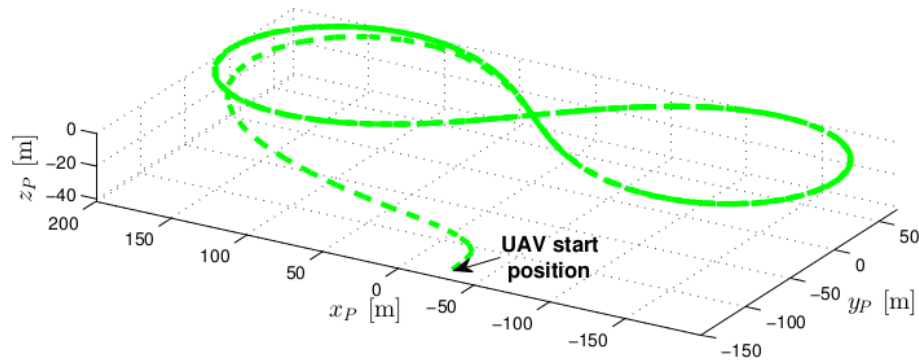


Figure 4.3: Lemniscate tracking numerical simulation: UAV trajectory as seen by an observer rigidly attached to the path frame origin.

4.2 and 4.3 present the obtained UAV trajectory with respect to the inertial frame and as seen by an observer rigidly attached to the path frame origin, respectively<sup>1</sup>.

<sup>1</sup> An illustration video for this simulation can be found in <https://youtu.be/WzeQ0ONhoYA>

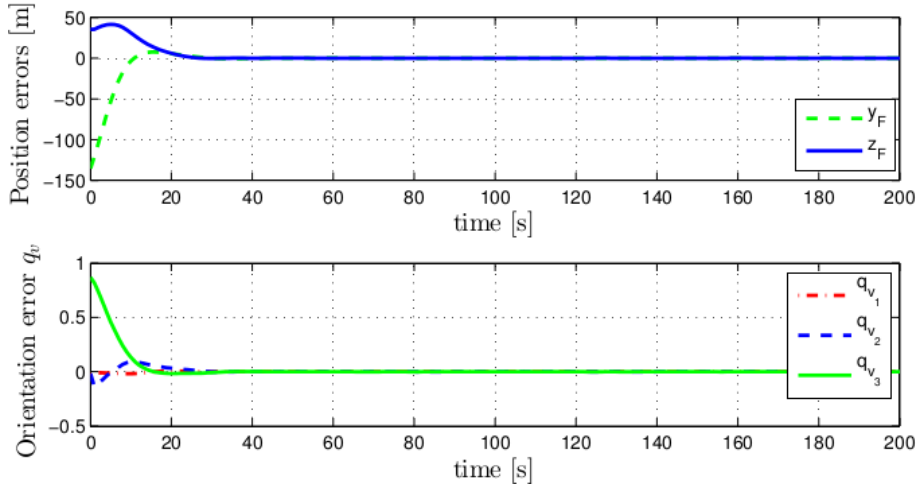


Figure 4.4: Lemniscate tracking numerical simulation: Position and orientation errors.

Position and orientation errors are presented in Figure 4.4 showing the good performance of the proposed controller.

Flight tests were conducted where a UAV was required to track a lemniscate path defined on a plane with five degrees angular displacement with respect to the inertial frame's horizontal plane, rotating around the  $\vec{z}$  axis with constant angular velocity and no forward speed, i.e.,

$$\begin{aligned}
 p_0 &= \begin{bmatrix} 0m & 0m & 0m \end{bmatrix}^T \\
 q_P|_{t=0s} &= \begin{bmatrix} 0.9990 & 0.0436 & 0 & 0 \end{bmatrix}^T \\
 v_d &= \begin{bmatrix} 0m/s & 0m/s & 0m/s \end{bmatrix}^T \\
 \omega_d &= \begin{bmatrix} 0rad/s & 0rad/s & 0.01rad/s \end{bmatrix}^T.
 \end{aligned}$$

This path kinematics clearly imposes a high control effort for the implemented control system, with relatively aggressive manoeuvres.

To compute the bank angle command  $\phi_c$  from the control signal  ${}^W\omega_W = \begin{bmatrix} \dot{\phi} & \dot{\theta} & \dot{\psi} \end{bmatrix}$  given by equation (4.2), the following coordinate turn equation [87] was used

$$\phi_c = \arctan\left(\frac{\dot{\psi}\|v_0\|}{9.81}\right). \quad (4.3)$$

The vertical velocity command  $v_{z_c}$  was computed from the desired pitch rate  $\dot{\theta}$  and current airspeed  $\|v_0\|$ , through

$$v_{z_c} = \|v_0\| \sin(\dot{\theta} d_t), \quad (4.4)$$

where  $d_t$  is the actual time delay between consecutive telemetry messages received from the UAV.

In order to ensure flight safety, the UAV airspeed was commanded to be kept constant at



20m/s, and the UAV bank and vertical velocity reference commands were limited to  $|\phi_c| < 25^\circ$  and  $|v_{z_c}| < 2\text{m/s}$  respectively. At the time of these tests, wind was blowing at about 6m/s South-west.

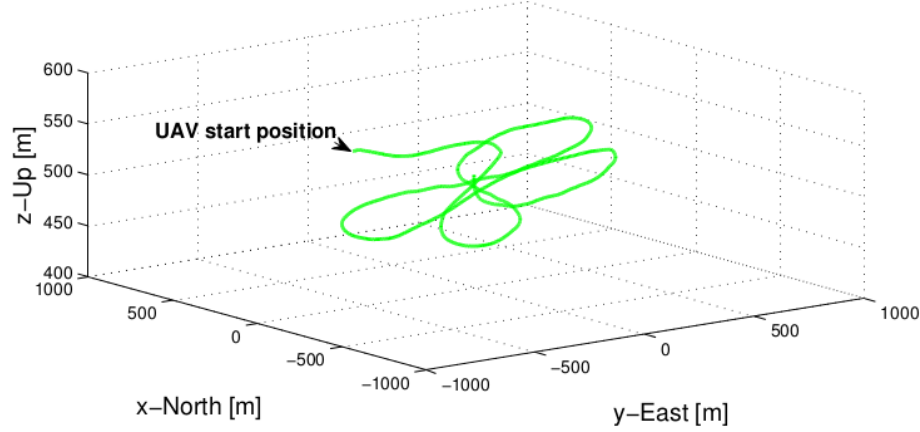


Figure 4.5: Lemniscate tracking flight test: UAV's trajectory following a moving lemniscate path, as seen by an observer attached to the inertial frame origin.

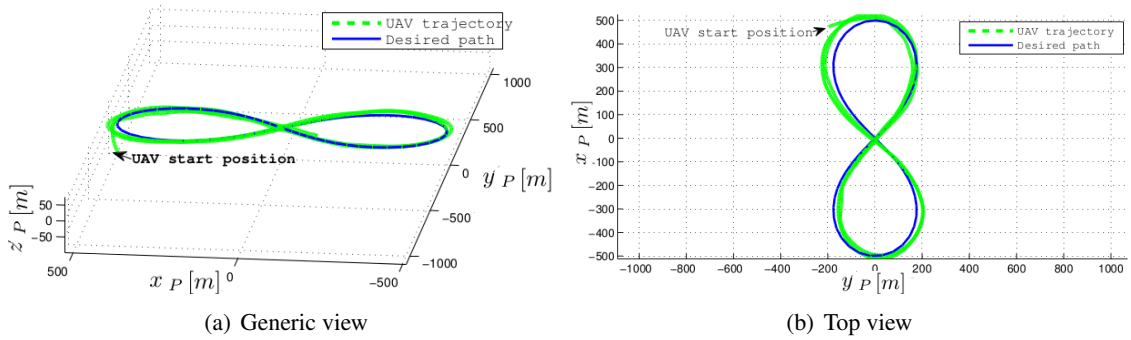


Figure 4.6: Lemniscate tracking flight test: UAV's trajectory following a moving lemniscate path, as seen by an observer rigidly attached to the path frame origin.

Figure 4.5 presents the UAV's trajectory following a moving lemniscate path<sup>2</sup>. The UAV trajectory as seen by an observer rigidly attached to the path frame origin is presented in Figure 4.6. In this flight test, the reference commands (in particular the vertical velocity command - see Figure 4.7) became saturated for some periods of time, as a consequence of the desired path's aggressive kinematics and the flight safety conditions introduced into the controller's output. This issue together with the control architecture used further affected system's performance as it can be observed in Figure 4.8. However, one can still observe a very satisfactory mission execution using the implemented controller as evidenced in Figure 4.6.

<sup>2</sup>An illustration video for this simulation can be found in <https://youtu.be/BqCDPmy1JJQ>

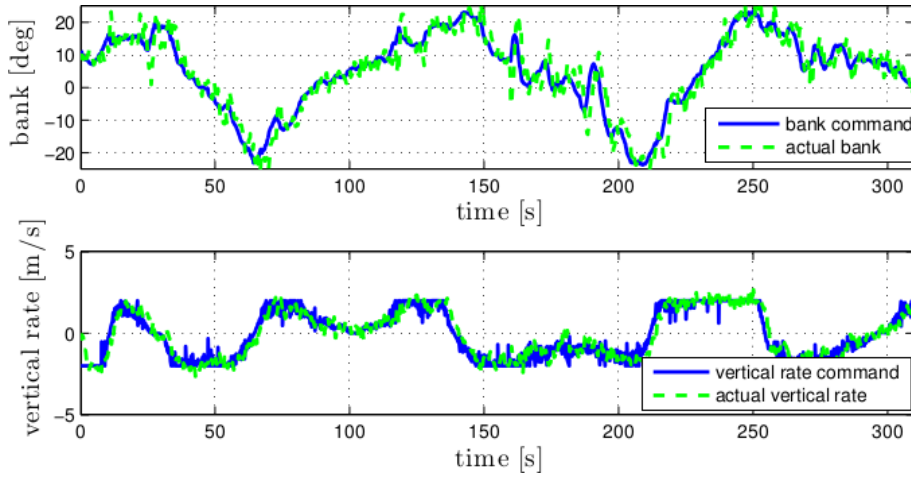


Figure 4.7: Lemniscate tracking flight test: UAV's reference commands.

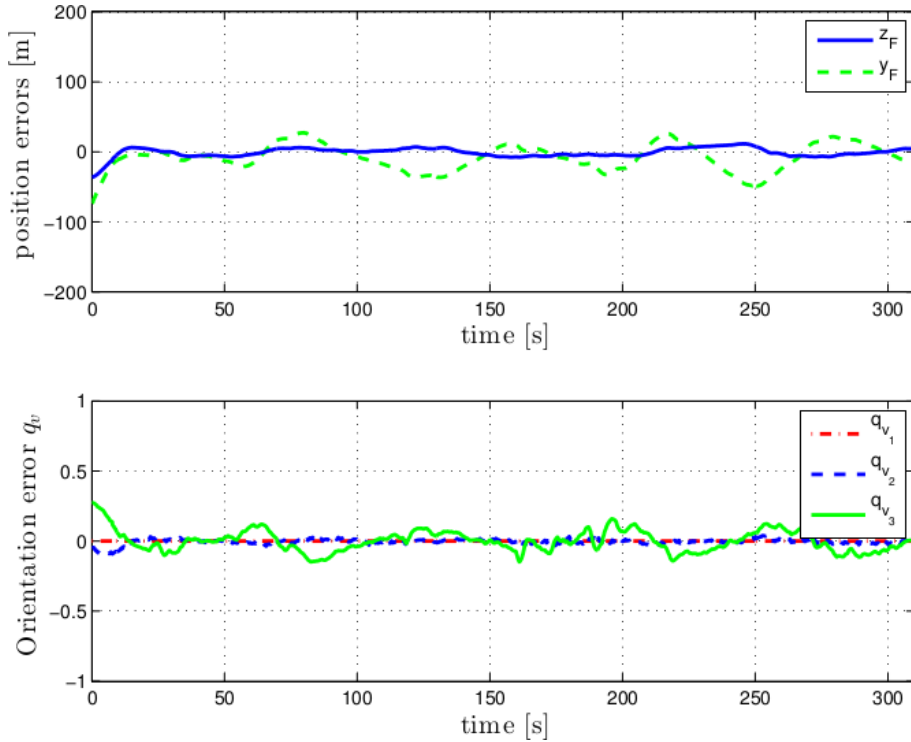


Figure 4.8: Lemniscate tracking flight test: UAV's position and orientation errors.

### 4.3 Application to autonomous landing on a moving vessel

Flight test results were conducted to illustrate the autonomous landing on a simulated moving vessel considered to be equipped with a recovery net, by following a landing pattern attached to the vessel. The same flight safety limits to compute the bank and vertical velocity commands considered in the previous application were used (see also equations (4.3) and (4.4)) and the vessel trajectory was set at 500m above ground level moving along the Ota's air base runway heading.

The path frame origin is located at the simulated vessel position and its orientation (that specifies the desired landing pattern) where set respectively to

$$p_0|_{t=0s} = \begin{bmatrix} 0m & 0m & 0m \end{bmatrix}^T$$

$$q_P = \begin{bmatrix} 0.1088 & -0.0244 & 0.0027 & 0.9938 \end{bmatrix}^T,$$

which corresponds to a (typical) slope of about 3 degrees with respect to the simulated vessel position. To emulate the effect of the ocean waves, the simulated movement of the vessel along the Ota runway with a ground speed of  $\|v_d\| = 6m/s$  was perturbed both horizontally and vertically by considering

$$v_d = 6 \begin{bmatrix} \cos \psi_{vd} \cos \theta_{vd} m/s & \sin \psi_{vd} \cos \theta_{vd} m/s & -\sin \theta_{vd} m/s \end{bmatrix}^T$$

with

$$\begin{bmatrix} \theta_{vd} & \psi_{vd} \end{bmatrix}^T|_{t=0s} = \begin{bmatrix} 0 rad/s & -0.2182 rad/s \end{bmatrix}^T \quad \text{and}$$

$$\begin{bmatrix} \dot{\theta}_{vd} & \dot{\psi}_{vd} \end{bmatrix}^T = \begin{bmatrix} -0.0125 \cos(\frac{1}{3}t) rad/s & 0.02 \cos(\frac{1}{4}t) rad/s \end{bmatrix}^T,$$

where  $t$  corresponds to the actual flight time.

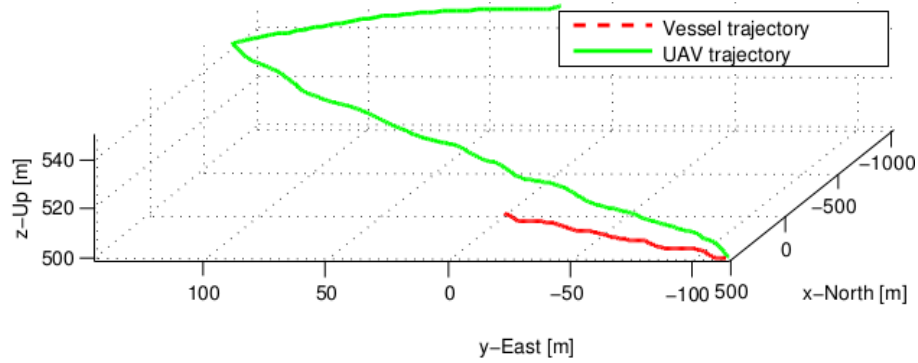


Figure 4.9: Autonomous landing flight test: UAV and vessel trajectories.

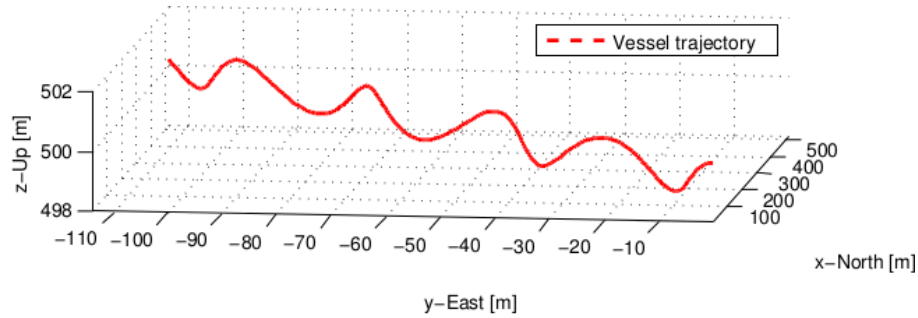


Figure 4.10: Autonomous landing flight test: Simulated vessel trajectory detail.

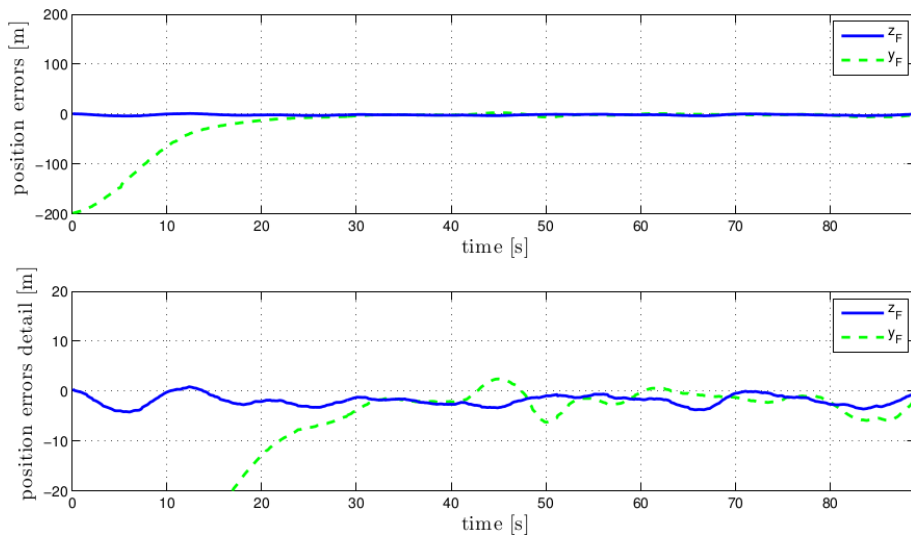


Figure 4.11: Autonomous landing flight test: Position errors.

A total of five flight tests were conducted. Figures 4.9 to 4.14 present the main results obtained for a given flight test, illustrating the good performance of the proposed controller<sup>3</sup>. The results show that the UAV converges and tracks the desired landing pattern attached to the moving vessel (Figures 4.9 and 4.10) with good convergence of position and orientation errors to a small interval around zero (Figures 4.11 and 4.12). On figures 4.11 to 4.13 one can notice a sudden change at the flight test time instant  $t = 47s$ . This was motivated by a change in the wind conditions as can be seen in Figure 4.14 which shows the autopilot wind estimator slowly converging to the new value of the wind after its change. The flight test was performed under temperatures of around 40°C, which made it very difficult to fine tune the UAV's engine. Despite the UAV's air speed command was kept constant throughout the flight, the warm temperatures affected the available thrust and consequently the air speed reference command tracking (see Figure 4.15). The effect of the change of the wind conditions during the flight on the UAV's ground speed can also be observed in Figure 4.15.

In addition to the fact that the command references were being sent from the ground station at a relatively small telemetry rate (5Hz), there is also a non-negligible delay between the reference commands acquired by the autopilot and the UAV's actual attitude that also affects the control system's performance (see Figure 4.13). In order to demonstrate the robustness of the implemented control system (tackling all these situations that were not completely taken into account during control design), Figure 4.16 shows the obtained lateral and vertical position errors for the five flight tests performed. Additionally, for the realistic case of a recovery net with  $5 \times 5m$  length, and considering a landing to be successful if both  $|y_F|$  and  $|z_F|$  are below 2.5m at the end of the desired landing pattern, one would have obtained four successful landings out of the five performed trials (in the unsuccessful trial,  $z_F = -3.06m$  at the end of the landing pattern).

<sup>3</sup>An illustration video for this simulation can be found in <https://youtu.be/50mSVR5n4tQ>

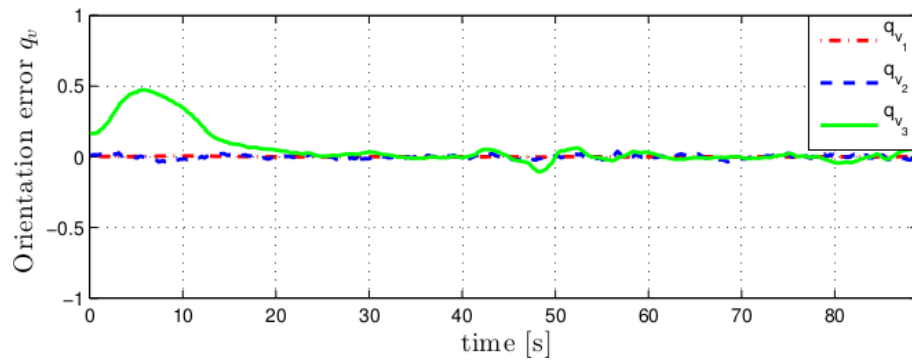


Figure 4.12: Autonomous landing flight test: Orientation errors.

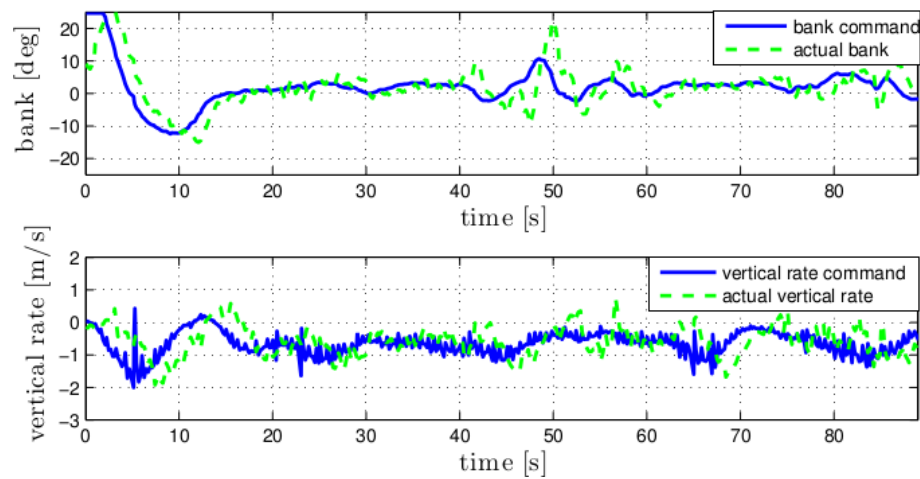


Figure 4.13: Autonomous landing flight test: UAV bank and vertical rate reference commands sent to the UAV from the ground control station.

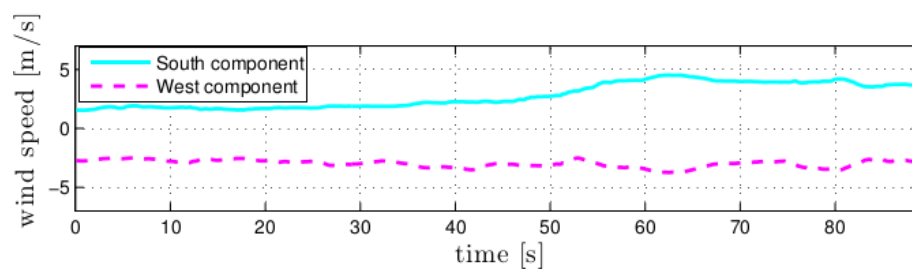


Figure 4.14: Autonomous landing flight test: Wind velocity estimate, provided by the autopilot.

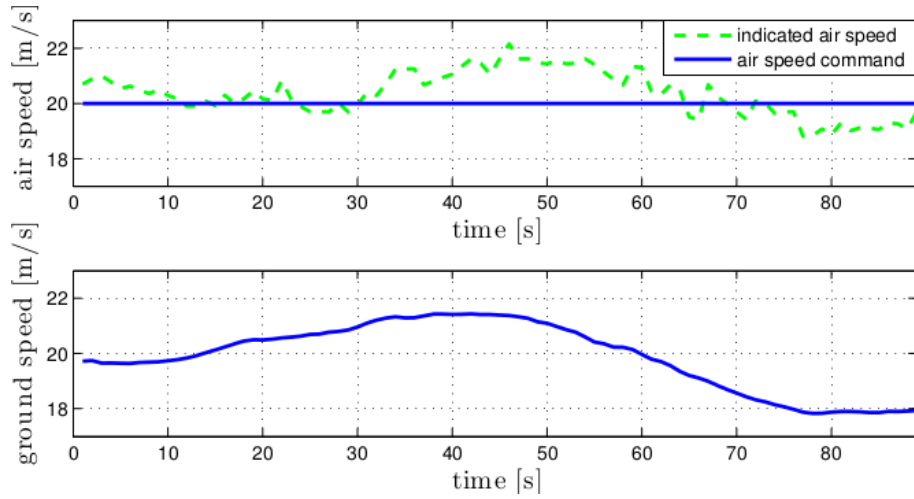


Figure 4.15: Autonomous landing flight test: UAV's air and ground speed.

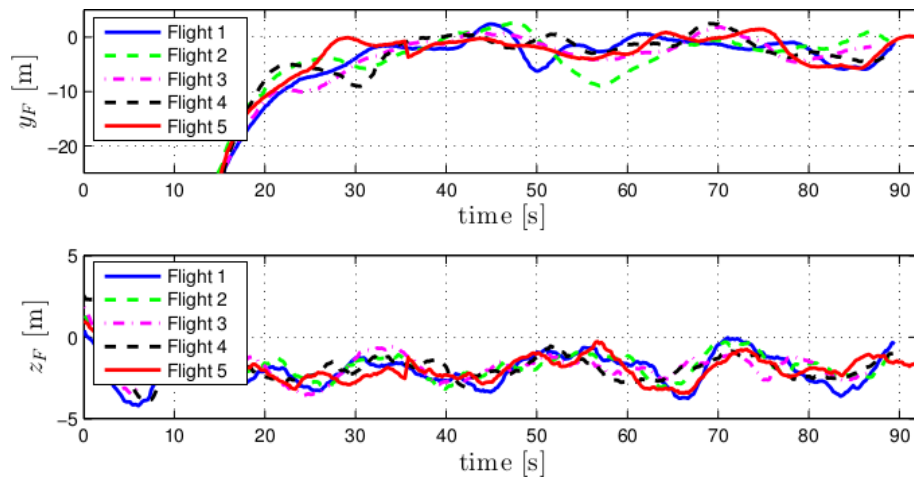


Figure 4.16: Autonomous landing flight test: Position errors for the five flight tests performed.

## Chapter 5

# Conclusions and future work

In this thesis the classical path following problem was extended to the case where the reference path is moving in the three-dimensional space. The so-called Moving Path Following problem was formally defined, error spaces were derived, and Moving Path Following kinematic control laws were proposed allowing an Unmanned Aerial Vehicle equipped with an autopilot to converge to and follow a path that moves either in the two-dimensional or in the three-dimensional space. Formal convergence proofs and performance metrics were provided. Several application scenarios, namely single and multiple ground target tracking, convoy protection, and autonomous landing on a moving vessel, were implemented and tested using the UAV ANTEX-X02 developed by the Portuguese Airforce Academy. The results of these tests have shown that the Moving Path Following method developed constitutes a general guidance and control tool that can successfully be applied to several problems under research within the robotics community, outperforming the solutions obtained by other methods. However, many open issues remain. Convergence proofs assumed that the control system was implemented in continuous time, that no uncertainties were present, and that the autopilot would be able to control the UAV dynamics such that the vehicle would follow the kinematic commands provided by the MPF control laws.

For all the performed flight tests, the MPF control law and related path generation methods were implemented on a laptop at the ground control station. Therefore, system performance can be further improved when the control law is implemented onboard the aircraft. This allows for direct access to the sensors data, and thus for higher control rates, which otherwise are limited by bandwidth constraints. Additionally, all the considered targets' trajectories on the performed flight tests were emulated. Despite typical kinematics for the considered targets were implemented (according to the considered scenario), the problem of acquiring target's information using passive sensors needs also to be addressed.

The performance of the MPF method for the presented convoy protection application (refer to Section 3.4) depends, among other things, on the geometry of the reference path and further work is necessary to study which is the best reference geometric path shape to choose.

For the case where the UAV flies at constant altitude, the necessary conditions for the moving path's geometry and linear and angular velocities with respect to the inertial frame that guarantee

that the MPF problem is always well posed were formally addressed. Similarly to the 2D MPF case, an important issue to consider in the future is the derivation of 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 general 3D MPF problem is well posed (by taking into account UAV's physical constraints).

Another interesting open issue to be explored is the ability to specify the UAV's desired position along the reference (moving) path, as considered in [20]. This would allow to expand the MPF method for the case where multiple vehicles follows the same desired reference moving path in a three dimensional space. The UAV's relative position along the path could then be chosen to maximize the control objective (e.g., ensure that a given target is always on the camera's field of view of at least one UAV) and simultaneously guarantee the necessary separation distance (due to flight safety issues - see for example [93, 67]).

In fact, an attractive and challenging scenario is the deployment of groups of networked UAVs that can interact autonomously with the environment and other vehicles to perform, in the presence of uncertainty and adversity, tasks beyond the ability of individual vehicles. This entails the development of advanced decentralized robust control and navigation systems for (optimal) path-planning, obstacle avoidance, and motion control of single and multiple vehicles that take into account the presence of uncertainty, nonlinear dynamics, partial noisy state measurements and disturbances, faults, and limited and disrupted communications. This thesis shows how single geometric paths, e.g. circles or straight lines, can be combined to form a more complex time-varying path shape. The Moving Path Following method can thus be used as the basic control task that each single autonomous vehicle is required to execute. Envisioned mission scenarios using multiple vehicles are for example perimeter protection/surveillance with respect to a moving reference area (which for example can be set around a large fleet of ground moving vehicles and may also depend on their speed or expected threats along their trajectory).



## Appendix A

### Flight tests architecture

This appendix presents the flight test architecture used to demonstrate the effectiveness of the MPF method on the different scenarios that were considered during this research work. All the MPF



(a)



(b)



(c)

Figure A.1: Flight tests framework: a) Portuguese Air Force research facility at the Ota air base; b) ANTEX-X02 UAV; c) Ground control station at the Santa Cruz airfield

control laws presented in this thesis were implemented and tested on the ANTEX-X02 platform (see Figure A.1 (b)). 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<sup>1</sup> [81] (aiming to

<sup>1</sup><http://www.emfa.pt/www/po/unidades/subPagina-10D00-019.005.003.004-pitvant>

develop UAV systems and concepts of operation) and SEAGULL<sup>2</sup> [82] (aiming to develop efficient solutions to address the challenges of maritime situational knowledge management) research projects. The main characteristics of ANTEX-X02 are listed in Table A.1.

Table A.1: ANTEX-X02 main features.

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

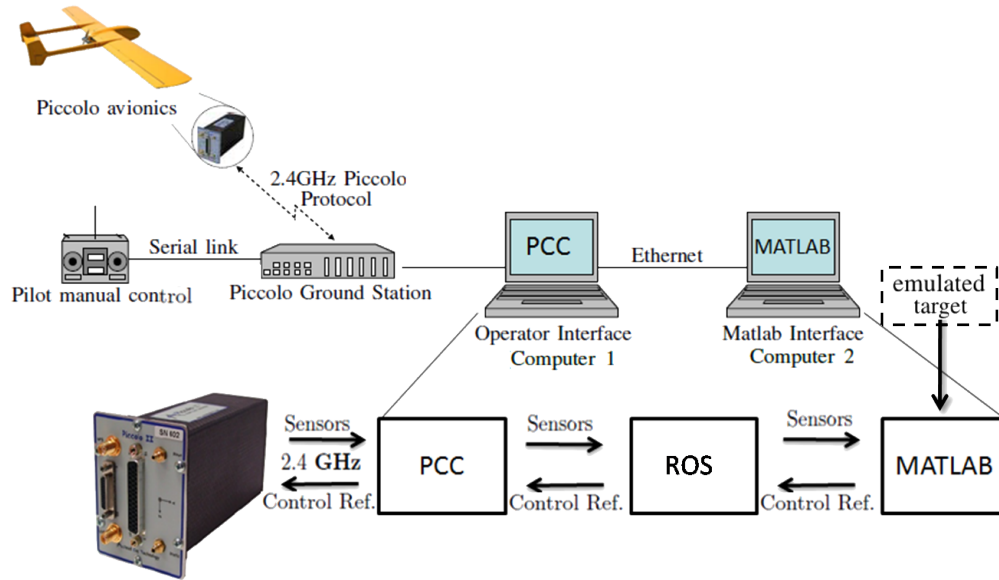


Figure A.2: Flight test operation diagram.

The platform is equipped with a Piccolo II<sup>3</sup> autopilot that plays the role of an inner-loop controller that provides the required actuation signals for the UAV's control surface's deflections and the engine power according to the UAV's current state that is inferred from the measurements of the onboard sensors, and the reference signal that is transmitted by the MPF algorithm. The autopilot relies on a mathematical model parametrized by the aircraft geometric data and has a built-in wind estimator. Several model and controller parameters can be set by the user [94]. In this work, the parameters collected from more than 500 hours of flight with the ANTEX-X02 were used.

The proposed control 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

<sup>2</sup><http://www.criticalsoftware.com/pt/seagull>

<sup>3</sup>[www.cloudcaptech.com](http://www.cloudcaptech.com)

to the aircraft as it is illustrated in Figure A.2. Robot Operating System (ROS) software [95] was used to establish a convenient communication interface between these two computers. The reference target's coordinates were generated according to the corresponding expected kinematics for the considered mission scenario. 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 relatively slow telemetry rate (between 2Hz and 5Hz, depending on the considered application). 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 A.1 (c)) in order to reduce communications losses.

The field tests reported in this thesis were conducted at the Portuguese Air Force's UAV test facility (see Figure A.1 (a)), located in the Ota air base<sup>4</sup> and the Santa Cruz airfield, in Portugal. For the field tests, a few safety measures were introduced. The bank reference sent to the aircraft was limited to  $25^\circ$  and the magnitude of the vertical velocity command was limited to 2m/s. 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 and vertical velocity references 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.

---

<sup>4</sup><http://www.emfa.pt/www/po/unidades/subPagina-10D00-019.001.003.010.003-infraestrutura-de-testes-de-uav>



## Appendix B

# Controller implementation details

The considered reference paths used in this thesis were straight lines, circumferences, lemniscates (or a combination of both straight lines and circumferences as described in Appendix C). This appendix provides further details regarding the considered reference path's parametric equations, which allows to compute the parallel-transport frame (at the path point that is closest to the UAV) and then determine the linear and angular displacements between the parallel-transport frame and the UAV's wind frame, that correspond to the MPF position and orientation error space variables considered in the control design.

### B.1 Path reference frames

In order to compute the MPF error space variables considered for the control design, the first goal is to define a moving coordinate frame (designated in this thesis by the  $\{F\}$  frame) located at the path point that is closest to the UAV.

Classical textbooks on geometry typically deal with moving frames using the Serret-Frenet frame concept because of its close association with the path's curvature and torsion, which are coordinate-system independent [86]. Unfortunately, the Serret-Frenet frame is undefined when the desired path is even momentarily straight (has vanishing curvature), and exhibits wild swings in orientation around points where the osculating plane's normal has major changes in direction. An alternative approach (the one that was implemented in this research work), is to use the parallel-transport frame method [85], which allows to overcome the above mentioned issues. The following sections borrows heavily from [86] and provides further details regarding the above ideas.

#### B.1.1 Serret-Frenet frame

Consider a desired path  ${}^P p_d(u) = \begin{bmatrix} {}^P p_{d_x}(u) & {}^P p_{d_y}(u) & {}^P p_{d_z}(u) \end{bmatrix}^T$ , parametrized by  $u$ , written in the path frame  $\{P\}$ . If  ${}^P p_d(u)$  is a thrice-differentiable path with non-vanishing second derivative, its tangent  $\vec{t}$ , normal  $\vec{n}$  and binormal  $\vec{b}$  vectors at a given point  $u$  on the path (that specify the

corresponding Serret-Frenet frame at that path point) are given by

$$\begin{aligned}\vec{t}(u) &= \frac{{}^P p'_d(u)}{\|{}^P p'_d(u)\|} \\ \vec{n}(u) &= \frac{\vec{t}'(u)}{\|\vec{t}'(u)\|} \\ \vec{b}(u) &= \vec{t}(u) \times \vec{n}(u).\end{aligned}$$

The Serret-Frenet differential formulas are

$$\frac{d}{du} \begin{bmatrix} \vec{t}(u) \\ \vec{n}(u) \\ \vec{b}(u) \end{bmatrix} = \|{}^P p'_d(u)\| \begin{bmatrix} 0 & k(u) & 0 \\ -k(u) & 0 & \tau(u) \\ 0 & -\tau(u) & 0 \end{bmatrix} \begin{bmatrix} \vec{t}(u) \\ \vec{n}(u) \\ \vec{b}(u) \end{bmatrix}, \quad (\text{B.1})$$

where the path curvature  $\kappa(u)$  and torsion  $\tau(u)$  are given by

$$\begin{aligned}\kappa(u) &= \frac{\|{}^P p'_d(u) \times {}^P p''_d(u)\|}{\|{}^P p'_d(u)\|^3} \\ \tau(u) &= \frac{\|{}^P p'_d(u) \times {}^P p''_d(u) \cdot {}^P p'''_d(u)\|}{\|{}^P p'_d(u) \times {}^P p''_d(u)\|^2}.\end{aligned}$$

( $\cdot$  denotes the dot product of two vectors). Since, by definition, the arc length  $\ell$  of a given desired path  ${}^P p_d(u)$  is

$$\ell(u) = \int_0^u \|{}^P p'_d(\zeta)\| d\zeta,$$

the desired path is often re-parametrized to obtain a natural parametrization with respect to the path length ( ${}^P p_d(\ell) = {}^P p_d(\ell(u))$ ), such that parameter  $u$  in equation (B.1) becomes the path length  $\ell$  and  $\|{}^P p'_d(u)\| = 1$ .

Notice that the above formulation of the Serret-Frenet frame becomes ill-defined if the second derivative of the path becomes zero. Moreover, since the Serret-Frenet frame's normal vector  $\vec{n}$  always points toward the concave side of the desired path, should the reference path be a sinusoid, the direction of the normal would be discontinuous, jumping 180° at the path inflection points. Thus, the principal normal  $\vec{n}$  is not suitable to define a reference frame for path following control purposes.

### B.1.2 Parallel-transport frame

The parallel transport frame is based on the observation that, while  $\vec{t}(u)$  for a given desired path parametrization is uniquely defined, it is possible to arbitrarily choose a complementary basis ( $\vec{n}_1(u), \vec{n}_2(u)$ ) for the remainder of the frame, as long as it defines a plane whose normal is parallel to  $\vec{t}(u)$  at each path point. One hypothesis is to choose the derivatives ( $\vec{n}'_1(u), \vec{n}'_2(u)$ )

depending only on  $\vec{t}(u)$ ,

$$\frac{d}{du} \begin{bmatrix} \vec{t}(u) \\ \vec{n}_1(u) \\ \vec{n}_2(u) \end{bmatrix} = \|P'_{p_d}(u)\| \begin{bmatrix} 0 & k_1(u) & k_2(u) \\ -k_1(u) & 0 & 0 \\ -k_2(u) & 0 & 0 \end{bmatrix} \begin{bmatrix} \vec{t}(u) \\ \vec{n}_1(u) \\ \vec{n}_2(u) \end{bmatrix}, \quad (\text{B.2})$$

solving this way the problems associated to the Serret-Frenet frame. It can be shown that parameters  $k_1(u)$  and  $k_2(u)$  in equation (B.2) are related to the path curvature  $\kappa(u)$  and torsion  $\tau(u)$  through [85, 86],

$$\kappa(u) = \sqrt{k_1(u)^2 + k_2(u)^2} \quad (\text{B.3})$$

$$\tau(u) = -\frac{d}{du} \left( \arctan \frac{k_2(u)}{k_1(u)} \right). \quad (\text{B.4})$$

Similarly to the results in equation (B.1) the desired path can be re-parametrized with respect to the path length ( $^P p_d(\ell) = ^P p_d(\ell(u))$ ), such that parameter  $u$  in equations (B.2) to (B.4) becomes the path length  $\ell$  and  $\|P'_{p_d}(u)\| = 1$ .

## B.2 Reference path's parametric equations

Recall from the MPF formulation presented in Section 1.1 that  $^P p_d(u) = \begin{bmatrix} ^P p_{d_x}(u) & ^P p_{d_y}(u) & ^P p_{d_z}(u) \end{bmatrix}^T$  is a desired geometric path parametrized by  $u$  and that  $^P p_d(u)$  is a point on the path expressed in the path frame  $\{P\}$ .

The parametric equations for each of the reference paths considered in this thesis and their corresponding arc lengths, curvatures and tangent vectors are given in Table B.1 [52, 96],

Table B.1: Reference path's parametric equations.

	Straight line	Circle	Lemniscate
$^P p_{d_x}(u)$	$u$	$r \cos(u)$	$\frac{w \cos(u)}{1 + \sin^2(u)}$
$^P p_{d_y}(u)$	0	$r \sin(u)$	$\frac{w \sin(u) \cos(u)}{1 + \sin^2(u)}$
$^P p_{d_z}(u)$	0	0	0
$^P \vec{t}(u)$	$\begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} -\sin(u) & \cos(u) & 0 \end{bmatrix}$	$\begin{bmatrix} \frac{1-3\sin^2(u)}{(\sin^2(u)+1)^{3/2}} & \frac{3\sin(u)-\sin^3(u)}{(\sin^2(u)+1)^{3/2}} & 0 \end{bmatrix}$
$\ell(u)$	$u$	$ur$	$\sqrt{2} w \int_0^u [3 - \cos(2u)]^{-\frac{1}{2}} du$
$\kappa(u)$	0	$\frac{1}{r}$	$\frac{-3\sqrt{2} \cos(u)}{w \sqrt{3-2\cos(2u)}}$

where  $r$  is the circle radius,  $w$  is the lemniscate width,  $^P \vec{t}(u)$  is the unitary tangent vector to the path at the path point parametrized by  $u$ , written in the  $\{P\}$  frame. Note from Table B.1 that the desired reference path is fixed with respect to the path frame  $\{P\}$  and thus its time-varying position and attitude relative to the inertial frame is specified (respectively) by the linear and angular velocities of the path frame  $\{P\}$  expressed in  $\{I\}$ .

### B.3 Position errors

According to the notation introduced in this thesis, one can write the relative distance  $d$  between the UAV position  $p$  and any path point  ${}^I p_d(u)$  parametrized by  $u$ , written in the inertial frame  $\{I\}$  as

$$\begin{aligned}\|d(u)\| &= \|{}^I p_d(u) - p\| \\ &= \|p_0 + {}^I R_P {}^P p_d(u) - p\|,\end{aligned}$$

where  $p_0$  denotes the origin of the path-frame  $\{P\}$  written in  $\{I\}$  and  ${}^I R_P$  is the rotation matrix that allows to change the reference frame of a given vector from  $\{P\}$  to  $\{I\}$ . In practice, the relative distance  $d$  is a function of  $u$ . A minimization function can be used<sup>1</sup> to find

$$u_{min} = \operatorname{argmin}\{\|d(u)\|\} \quad (\text{B.5})$$

and the corresponding path point  ${}^P p_d(u_{min})$  that is closest to the UAV, given an initial guess solution  $u_0$ . When the MPF controller is firstly engaged (at time instant  $t = 0$ s), the initial guess solution for (B.5) is set to zero ( $u_0 = 0$ ). Then, for each subsequent control loop  $i$  (at time instant  $t_i > 0$ s),  $u_0$  is set to the previous computed solution ( $u_0|_{t_i} = u_{min}|_{t_{i-1}}$ ). This allows to obtain a faster convergence of the minimization algorithm and, more importantly, to obtain continuous solutions for the  $u_{min}$  even when the minimization problem has more than one solution, as it is the case when the path intersects itself, like the lemniscate path considered in this thesis. Furthermore, since  $u_{min}(t)$  is (under conditions (3.7)) an increasing continuous function of time, the derivative of  $\ell$  with respect to time can be computed as (see also Table B.1)

$$\dot{\ell} = \frac{d\ell}{dt} = \frac{d\ell}{du} \dot{u}_{min}. \quad (\text{B.6})$$

Note that equation (B.6) can be used as an alternative to equation (3.5), ensuring in practice at the implementation level that  $\dot{\ell}$  is always finite.

Once the parallel-transport frame origin is determined (and therefore  ${}^P \vec{t}(u_{min})$  is known - see Table B.1) the remainder of the  $\{F\}$  frame, written in the  $\{P\}$  frame, is computed through

$$\begin{aligned}{}^P \vec{n}_2(u) &= \vec{z}_P \\ {}^P \vec{n}_1(u) &= {}^P \vec{n}_2(u) \times {}^P \vec{t}(u),\end{aligned}$$

where  $\vec{z}_P$  is the  $z$  axis of the  $\{P\}$  reference frame. From the above results it is straightforward to

---

<sup>1</sup>In this thesis, the *fminsearch* function from the MATLAB optimization toolbox, that is based on the simplex search method [97], was used.



write the  $\{F\}$  frame basis in the inertial frame  $\{I\}$  as

$$\begin{bmatrix} \vec{I} \\ \vec{n}_1 \\ \vec{n}_2 \end{bmatrix} = {}^I R_P \begin{bmatrix} {}^P \vec{I} \\ {}^P \vec{n}_1 \\ {}^P \vec{n}_2 \end{bmatrix} \triangleq {}^F R_I, \quad (\text{B.7})$$

where  ${}^I R_P$  is the rotation matrix from  $\{P\}$  to  $\{I\}$ .

Finally, the position errors between the UAV's wind frame  $\{W\}$  and the corresponding parallel-transport frame  $\{F\}$  are given by

$$\begin{aligned} y_F &= [p - {}^I p_d(u)] \cdot \vec{n}_1 \\ z_F &= [p - {}^I p_d(u)] \cdot \vec{n}_2. \end{aligned}$$

## B.4 Orientation errors

Consider the UAV velocity vector  ${}^I v_W$  written in the inertial frame  $\{I\} = \{\vec{x}, \vec{y}, \vec{z}\}$  as presented in Chapter 3. The corresponding wind frame  $\{W\} = \{\vec{x}_W, \vec{y}_W, \vec{z}_W\}$  can be computed as follows. First, the  $\vec{x}_W$  vector is aligned with the direction of the UAV's velocity vector  ${}^I v_W$ , written in the  $\{I\}$  frame

$$\vec{x}_W = \frac{{}^I v_W}{\|{}^I v_W\|}. \quad (\text{B.8})$$

Then, the  $\vec{y}_W$ -axis is computed by projecting the  $\vec{x}_W$  vector onto the  $\vec{x} - \vec{y}$  plane and computing its cross product with the  $\vec{z}$ -axis, such that  $\vec{y}_W$  always points to the right of an observer that moves in the same direction of the aircraft, i.e.,

$$\vec{y}_W = \frac{\vec{z} \times (A \vec{x}_W)}{\|\vec{z} \times (A \vec{x}_W)\|}, \quad (\text{B.9})$$

where

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

Finally, the  $\vec{z}_W$ -axis is orthogonal to the previous two:

$$\vec{z}_W = \vec{x}_W \times \vec{y}_W. \quad (\text{B.10})$$

Considering the desired orientation of the UAV's velocity vector  ${}^F v_{W_d}$  written in the  $\{F\}$  frame as given by equation (2.10), one can also obtain the desired wind frame  $\{W_d\} = \{\vec{x}_{W_d}, \vec{y}_{W_d}, \vec{z}_{W_d}\}$  basis vectors by computing first  ${}^I v_{W_d} = {}^F R_I {}^F v_{W_d}$  and then proceed as shown for equations (B.8) to (B.10).

Since by definition,

$$\begin{bmatrix} \vec{x}_W \\ \vec{y}_W \\ \vec{z}_W \end{bmatrix} \triangleq {}^W R_I \quad \text{and} \quad \begin{bmatrix} \vec{x}_{W_d} \\ \vec{y}_{W_d} \\ \vec{z}_{W_d} \end{bmatrix} \triangleq {}^{W_d} R_I, \quad (\text{B.11})$$

equations (B.7) and (B.11) can be combined to obtain the orientation errors considered in Chapter 3 for the 2D MPF controller, given by the Euler angles

$$\begin{aligned} [0, 0, \tilde{\psi}]^T &= \arg({}^F R_W) \\ [0, 0, \tilde{\psi}_d]^T &= \arg({}^F R_{W_d}) \\ \tilde{\psi} &= \tilde{\psi} - \tilde{\psi}_d. \end{aligned}$$

Note from the above equations that a more general parametrization of the orientation error between  $\{W\}$  and  $\{W_d\}$  (given by  $\tilde{\psi}$ ) would be obtained from  $\arg({}^{W_d} R_W)$ , which is precisely the considered approach for the 3D MPF controller presented in Chapter 4 using quaternions to parametrize the rotation matrices between reference frames. Further details regarding the use of quaternions can be found in Appendix D.

## Appendix C

# Multiple targets tracking implementation details

This appendix provides a detailed description of the path generation details, convergence conditions, performance metrics and the control system architecture for the multiple targets tracking implementation. Formal convergence conditions under which the vehicle reaches its moving target using Algorithm 2 are presented in the sequel, along with a performance metric result.

### C.1 Path generation details

This section presents path generation details for the multiple targets tracking scenario.

Given the start position  $p_s = (x_s, y_s)$ , the initial course angle  $\psi_s$ , and the minimum radius  $r_{min}$ , the centres of the right and left turning circles (see Figure 3.9) are given by

$$\begin{aligned} c_r &= p_s + r_{min} \left( \cos \left( \psi_s + \frac{\pi}{2} \right), \sin \left( \psi_s + \frac{\pi}{2} \right) \right) \\ c_l &= p_s + r_{min} \left( \cos \left( \psi_s - \frac{\pi}{2} \right), \sin \left( \psi_s - \frac{\pi}{2} \right) \right). \end{aligned}$$

Let  $p_e = (x_e, y_e)$  be the desired final position, and  $d_i = (p_e - c_i)$  be the vector that connects the desired end position  $p_e$  to the turning circle center coordinates  $c_i$ , where index  $i$  equals  $l$  or  $r$  for L or R circle's centres, respectively.

Figure C.1 presents the geometry for both clockwise (R) (solid lines) and counterclockwise (L) circles (dashed lines), where  $\chi_i$  is the angle between by the line connecting  $p_e$  to  $c_i$  and the  $\vec{x}$  axis. From Figure C.1, it is easy to verify that

$$\vartheta_i = \arcsin \left( \frac{r_{min}}{\|d_i\|} \right).$$

The angle  $\delta_i$  (where index  $i$  equals  $l$  or  $r$ ) that corresponds to the tangent line to the circle centred at  $c_i$ , with radius  $r_{min}$ , and passing through  $p_e$  is given by

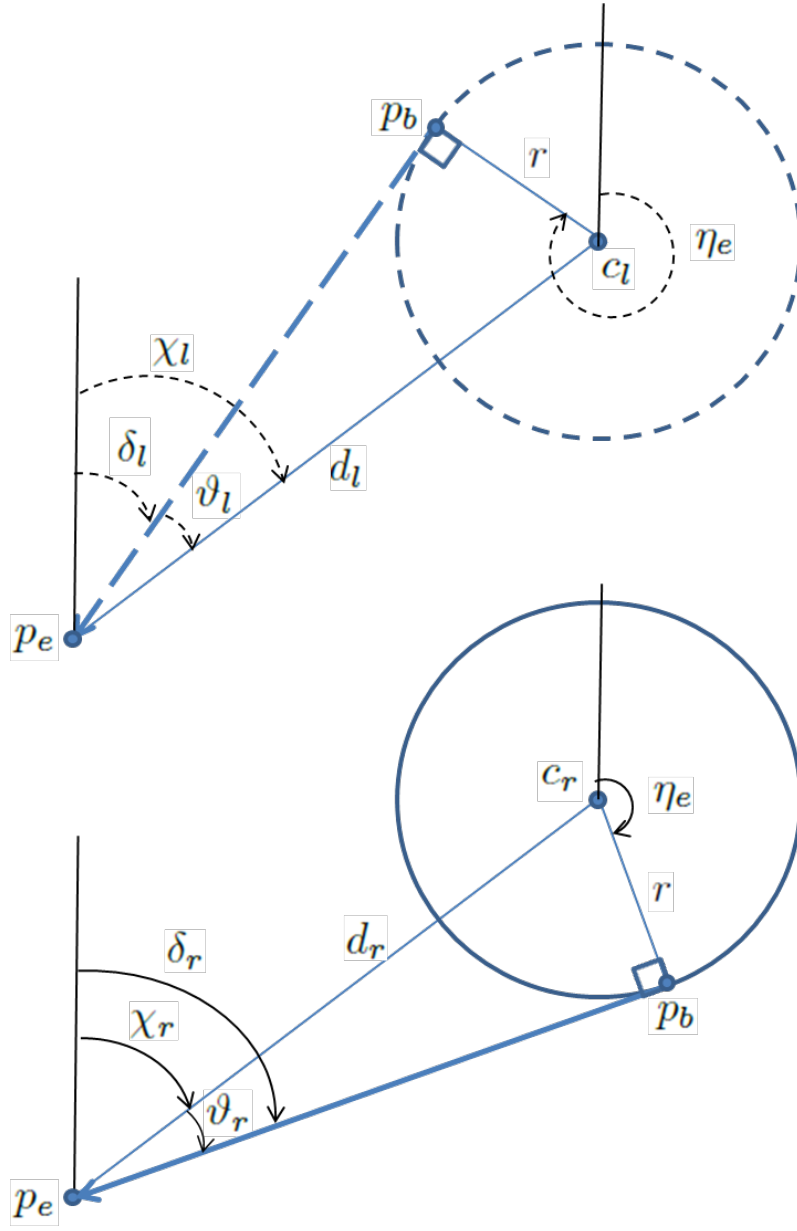


Figure C.1: RS (solid line) and LS (dashed line) Dubins path parameters.

$$\delta_i = \begin{cases} \chi_i - \vartheta_i & \text{if } i = l, \\ \chi_i + \vartheta_i & \text{if } i = r. \end{cases}$$

From straightforward trigonometric relations one can conclude that the angle  $\eta_e$ , as shown in Figure C.1, corresponding to the angle between the line that connects the circle centre  $c_i$  to the

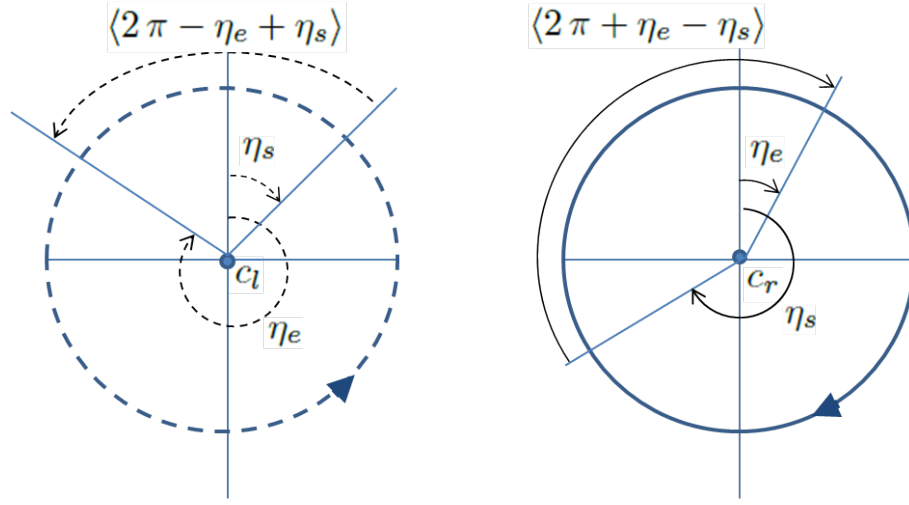


Figure C.2: Angular distance between angles  $\eta_s$  and  $\eta_e$  for clockwise ( $c_r$ ) and counter clockwise ( $c_l$ ) circles.

boundary position  $p_b$ , and the  $\vec{x}$  axis, is given by

$$\langle \eta_e \rangle_i = \begin{cases} \delta_i - \frac{\pi}{2} & \text{if } i = l, \\ \delta_i + \frac{\pi}{2} & \text{if } i = r, \end{cases}$$

where  $\langle \eta \rangle \triangleq \eta \bmod 2\pi$  and mod is the modulus operator that makes  $\eta \in [0, 2\pi]$ . Similarly,  $\eta_s$ , that corresponds to the angle between the line that connects the circle center  $c_i$  to the initial position  $p_s$ , and the  $\vec{x}$  axis can be computed through

$$\langle \eta_s \rangle_i = \begin{cases} \psi_s + \frac{\pi}{2} & \text{if } i = l, \\ \psi_s - \frac{\pi}{2} & \text{if } i = r. \end{cases}$$

Thus, the arc length of the R or L turns is given by

$$d_{C_i} = \begin{cases} \langle 2\pi - \eta_e + \eta_s \rangle & \text{if } i = l, \\ \langle 2\pi + \eta_e - \eta_s \rangle & \text{if } i = r. \end{cases}$$

The distance  $d_{S_i}$  along the straight line that connects  $p_b$  to  $p_e$  can be computed through (see Figure C.2)

$$d_{S_i} = d_i \cos \vartheta_i.$$

Finally, the total Dubins path length is

$$L_{LS} = d_l \cos \vartheta_l + d_{C_l} r_{min}$$

$$L_{RS} = d_r \cos \vartheta_r + d_{C_r} r_{min},$$

which can be cast in the compact form:

$$d_{P_i} = d_{S_i} + d_{C_i} r_{min}.$$

## C.2 Algorithm 2 detailed implementation and corresponding “virtual target” position estimation

The strategy for a single vehicle to visit a group of moving targets in a given order  $s_t$  as proposed in Section 3.3.3 is summarized in Algorithm 1. Algorithm 2 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 3.13). The estimated target interception position for Algorithm 2 is computed as follows.

---

**ALGORITHM 2:** Returns the path parameters for a single vehicle to visit a group of moving targets in a given order.

---

**Input:** Vehicle start pose  $(p, \psi)$ , sequenced virtual target’s position  $s_t^* = (t_1^*, t_2^*, \dots, t_m^*)$  with

$$t_j^* = (x_{t_j}^*, y_{t_j}^*), \text{ and path constraints } r_{min}.$$

**Output:** Minimal feasible path parameters (RS or LS).

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

**while**  $j \leq m \Rightarrow$  vehicle hasn’t visited all the targets **do**

1.  $(p_s, \psi_s) \leftarrow (p, \psi)$ ;
2.  $p_e \leftarrow t_j^*$ ;
3. Compute the length of two possible paths (RS and LS) between the vehicle’s current position  $p_s$  and  $p_e$ :  $\{L_{RS}, L_{LS}\}$ ;
4. Choose the path that has the minimum length:  
 $L \leftarrow \arg \min\{L_{RS}, L_{LS}\}$ ;
5. Compute and output chosen path parameters:  $c_i, \eta_s, \eta_e, \eta_v$  and  $d_{C_i}$ .  
**if**  $\langle \eta_v \rangle_i \geq d_{C_i} \Rightarrow$  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:  $p_b, \psi_p, \psi_p$ ;  
**end**  
 8. Set desired virtual target as the next target  $\Rightarrow j \leftarrow j + 1$ ;  
 9. Return to 1;  
**else**  
 10. Return to 5;  
**end**

**end**

---

Let  $\|v_t\|$  and  $\psi_t$  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  $\Delta t_t$  will be

$$p_t(\Delta t_t) = p_{t_0} + \|v_t\| (\cos \psi_t, \sin \psi_t) \Delta t_t. \quad (\text{C.1})$$

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 (3.19)) as a function of  $\Delta t_t$  as

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

Hence, assuming that the vehicle moves with constant speed  $V$ , the elapsed time between vehicle’s initial and final positions,  $\Delta t_v$ , can be obtained through

$$\Delta t_v = \frac{d_{P_i}(\Delta t_t)}{V}. \quad (\text{C.2})$$

Note that if  $\Delta t_v = \Delta t_t = \Delta t$ , it physically means that both vehicle and target take the same time  $\Delta t$  to reach their final destinations, and thus, they arrive simultaneously to a given “rendezvous” point at coordinates  $p_e = p_t(\Delta t)$ . Variable  $\Delta t$  can be computed making  $\Delta t_v = \Delta t_t = \Delta t$  and solving equation (C.2) for  $\Delta t$ . If the target moves at constant speed  $\|v_t\|$  and heading  $\psi_t$ , 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 ( $\dot{v}_t \neq 0$ ,  $\dot{\psi}_t \neq 0$ ) it is proposed to filter first the “instantaneous” virtual target coordinates given by  $p_t(\Delta t)$  - see equation (C.1) - in order to obtain a filtered output of  $p_t(t)$  that will be sufficiently smooth, with bounded derivatives, such that it can play the role of a reference target complying with the MPF requirements. The filter was adopted from [98] 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

$$\begin{aligned} \dot{x}_1 &= a_1 \tanh(x_2) \\ \dot{x}_2 &= a_2 \tanh(k_1 u - k_1 x_1 - k_2 x_2) \end{aligned} \quad (\text{C.3})$$

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

$$V_4 = \frac{1}{2} e' \begin{bmatrix} \frac{k_1^2}{k_2 a_2} & \frac{k_1}{2 a_2} \\ \frac{k_1}{2 a_2} & \frac{k_2}{a_2} \end{bmatrix} e + \frac{k_1 a_1}{k_2 a_2^2} \int_0^{e_2} \tanh(y) dy$$

where  $e = (e_1, e_2) \triangleq (x_1 - u, x_2)$ , 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 [98]. 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 Chapter 1, depending on the commanded “look ahead distance” for the virtual target position  $\|v_t\| \Delta t$  (cf. equation (C.2)). In the case that  $\Delta t$  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 2 is used), the system behaves similarly to the constant bearing guidance law described in Chapter 1.

### C.3 Convergence conditions

The next result concerns the conditions under which the UAV reaches its moving target when navigating using Algorithm 2 together with the MPF control law. Figure 3.13 presents the considered problem kinematics. It is assumed that the target has limited speed  $0 \leq \|v_t\| \leq V_{t_{max}}$  (where  $V_{t_{max}}$  is the assumed target’s maximum speed) and angular velocity  $|\dot{\psi}_t| \leq \dot{\psi}_{t_{max}}$  (where  $\dot{\psi}_{t_{max}}$  is the assumed target’s maximum angular speed). Considering the general case where 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 (C.3)) will also move with bounded time varying linear and angular velocities. These bounds are the same considered for the actual target, namely  $0 \leq \|v_t^*\| \leq V_{t_{max}}$  and  $|\dot{\psi}_t^*| \leq \dot{\psi}_{t_{max}}$ . 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 (3.21)). From Figure 3.13 it is straightforward to verify that the desired heading deviation  $\bar{\psi}_d$ , (cf. Section 3.2) 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 conclude that

$$|\bar{\psi}_d| \leq \arcsin \left( \frac{V_{t_{max}}}{V} \right). \quad (C.4)$$

Additionally, from Figure 3.13 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  $\psi_t$ ,

$$|\psi_p - \psi_0| \leq \arcsin \left( \frac{V_{t_{max}}}{V} \right). \quad (C.5)$$

The upper bounds in (C.4) and (C.5) will be used in the sequel. Consider the proposed control system architecture and let  $\psi_0$  be the angle between the line of sight vector that connects the UAV to the current target and the  $\vec{x}$  axis,  $\psi_p$  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  $\vec{x}$  axis (thus, in this



particular case,  $\psi_f = \psi_p$ ),  $V$  be the UAV's ground speed,  $v_t$  and  $\psi_t$  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  $V > 2V_{t_{max}}$ .

### Proof

From Figure 3.13 one can conclude that the relative closing velocity between the robot and the target satisfies

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

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

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

Additionally, if the following condition is met

$$\cos(\psi_p - \psi_0 + \bar{\psi}_d) > \frac{V_{t_{max}}}{V}, \quad (\text{C.6})$$

using the transitive property one obtains

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

which makes

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

Rewriting (C.6) yields the following convergence condition

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

which can be combined with (C.4) and (C.5) to obtain

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

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

## C.4 Performance metric

The optimal path (from current target to the next target) can be computed by an exhaustive a posteriori search. For each interception it is thus possible to compare the actual time for interception ( $t_{act}$ ) using the method proposed here with the optimal time of interception ( $t_{opt}$ ), thus assessing the performance of the method. The increase in the actual time for interception with respect to the optimal time, as a percentage of the optimal time for each interception, can be computed through

$$t_{increased}[\%] = \frac{t_{act} - t_{opt}}{t_{opt}} \times 100.$$

Figure C.3 shows the increase in the actual time for interception ( $t_{increased}$ ) for each interception shown in the multiple targets tracking numerical simulation considered in Section 3.3.3, using Algorithm 1.

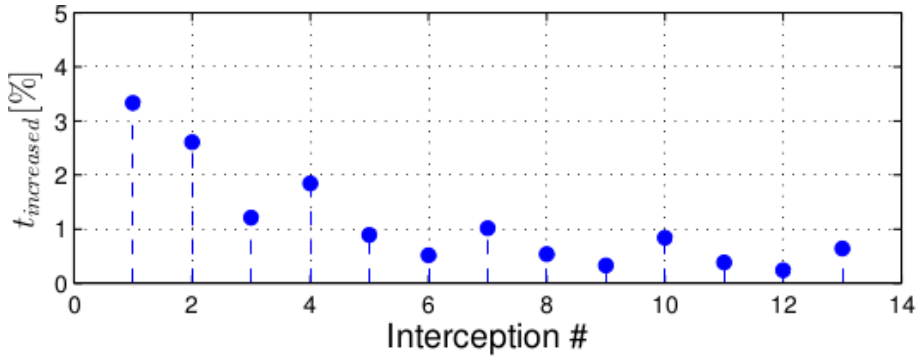


Figure C.3: Multiple targets tracking numerical simulation: Increase in the actual time for interception with respect to the optimal time of interception, as a percentage of the optimal time.

Table C.1 presents the performance (in terms of average time for interception) of each of the proposed methods (Algorithm 1 and Algorithm 2) with respect to the optimal case. The parameters in Table C.1 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  $5000 \times 5000\text{m}$  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  $3\text{m/s}$  ( $\|v_t\|_{t=0} = 3\text{m/s}$ ) 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  $\|v_0\| = 25\text{m/s}$ . The performance metrics values presented in Table C.1 for the proposed method were computed through

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

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

Table C.1: Proposed methods performance metrics.

	Algorithm 1	Algorithm 2
$(\%)t_{opt}$	99.49	99.80

From Table C.1 one can verify that, as previously argued, Algorithm 2 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 [56].

## C.5 Overall control architecture

The implemented control system architecture for the flight test is presented in Figure C.4 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  $\psi_t$ , 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  $\psi_t^*$ , and speed  $\|v_t^*\|$ .

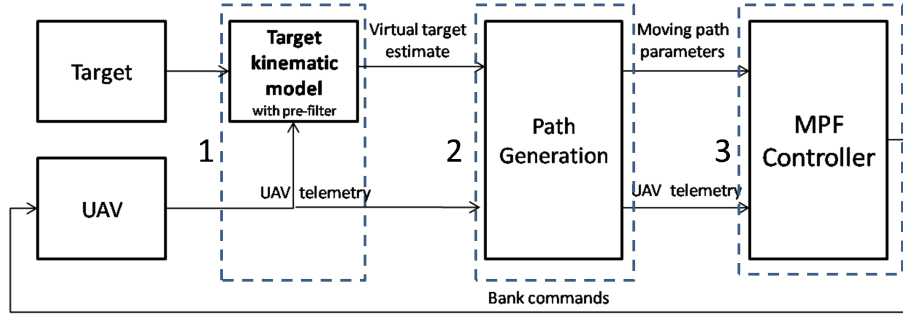


Figure C.4: Control system architecture.

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  $\psi$ , ground speed  $V$  and vehicle constraint  $r_{min}$ ;
- (b) virtual target variables  $p_t^*$ ,  $\psi_t^*$ , and  $\|v_t^*\|$  from previous stage.

Output:

- (a) path parameters  $c_i$ ,  $\eta_s$ ,  $\eta_e$ ,  $p_b$ ,  $\psi_p$  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  $\psi$ , and ground speed  $V$ ;
- (b) path parameters  $c_i$ ,  $\psi_p$ ,  $\dot{\psi}_p$ ,  $\eta_s$  and  $\eta_e$  from previous stage.

Output:

- (a) turn rate command  $\dot{\psi}$  to the UAV.

## Appendix D

# Quaternion fundamentals

In this thesis, the quaternion attitude representation is used to derive the 3D MPF control law. Following [90, 99], the quaternion convention and the corresponding fundamentals that are used in this thesis are summarized in this appendix.

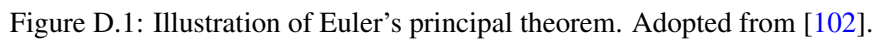
The following theorem has been fundamental in the development of several types of attitude coordinates and is generally referenced to Euler [100, 101].

### Euler's Principal Rotation

*A rigid body or coordinate reference frame can be brought from an arbitrary initial orientation to an arbitrary final orientation by a single rigid rotation through a principal angle  $\Phi$  about the principal axis  $\hat{e}$ , the principal axis being a judicious axis fixed in both the initial and final orientation.*

Thus, one can represent an arbitrary orientation with just a unit vector  $\hat{e}$  and an angle  $\Phi$ , as illustrated in Figure D.1. The unit vector  $\hat{e}$  defines the direction of rotation, and the angle  $\Phi$  is the amount to rotate about this axis to go from the initial to the final reference frame attitude (respectively the  $\{N\}$  and  $\{B\}$  frames depicted in Figure D.1). Vector  $\hat{e}$  will have the same components in both the initial and final orientation of the reference frame. The quaternion attitude representation is based upon this principal.

There are several ways to determine the quaternion corresponding to a given rotation. Historical developments have favoured some conventions over others and a thorough overview of the possible convention choices can be found in [99]. In this thesis, the Hamilton convention was adopted as it is common in the field of robotics (see for example [103, 104], thus taking advantage of the primitives already implemented in many robotics software libraries, such as the Robot Operating System [95] that was used in the flight test architecture in this research work - see Appendix A. The remaining of this appendix borrows in part from [99] and provides an overview of the most significant properties of the adopted quaternion convention.


$$\begin{aligned} q_0 &\triangleq \cos\left(\frac{\Phi}{2}\right) \\ q_{v_1} &\triangleq e_1 \sin\left(\frac{\Phi}{2}\right) \\ q_{v_2} &\triangleq e_2 \sin\left(\frac{\Phi}{2}\right) \\ q_{v_3} &\triangleq e_3 \sin\left(\frac{\Phi}{2}\right) \end{aligned}$$
$$\begin{aligned}\Phi &= 2 \arctan(\|q_v\|, q_0) \\ \hat{e} &= \frac{q_v}{\|q_v\|}.\end{aligned}$$
$$q_A \otimes q_B \triangleq \begin{bmatrix} q_{0A} q_{0B} - \hat{e}_A^T \hat{e}_B \\ q_{0B} \hat{e}_A + q_{0A} \hat{e}_B + \hat{e}_A \times \hat{e}_B \end{bmatrix}. \quad (\text{D.1})$$

Consider now a local frame  $\{L\}$  and a global frame  $\{G\}$ , let  $x_L$  be the image of vector  $x$  written with respect to  $\{L\}$  frame and  $x_G$  be that image written in frame  $\{G\}$ . Moreover, let  $q_{GL}$  and  ${}^G R_L$  be respectively the quaternion and rotation matrix transforming vector  $x$  from frame  $\{L\}$  to frame  $\{G\}$ . Using the adopted Hamilton convention, one can use unit quaternions to act as rotation operators in a way somewhat similar to rotation matrices, namely writing

$$\bar{x}_G = q_{GL} \otimes \bar{x}_L \otimes q_{GL}^* \quad \text{to be compare with} \quad x_G = {}^G R_L x_L.$$

Here, the bar notation  $\bar{x}$  was used to indicate that the vector  $x$  in the left expression is expressed in (pure) quaternion form  $\bar{x} = \begin{bmatrix} 0 & x^T \end{bmatrix}^T \in \mathbb{R}^4$ , thus differentiating it from that on the right. However, this circumstance is most of the times unambiguous and can be derived from the context, especially by the presence of the quaternion product  $\otimes$ . In what follows, this bar will be omitted, writing simply  $x_G = q_{GL} \otimes x_L \otimes q_{GL}^*$ , thus allowing for an additional abuse of notation by writing

$$x_G = q_{GL} \otimes x_L \otimes q_{GL}^* = {}^G R_L x_L.$$

Since both sides of this identity are linear in  $x$ , an expression of the rotation matrix equivalent to the quaternion is found by expanding the left hand side and equating to the corresponding terms on the right, yielding

$${}^G R_L(q_{GL}) = \begin{bmatrix} q_0^2 + q_{v_1}^2 - q_{v_2}^2 - q_{v_3}^2 & 2(q_{v_1}q_{v_2} - q_0q_{v_3}) & 2(q_{v_1}q_{v_3} + q_0q_{v_2}) \\ 2(q_{v_1}q_{v_2} + q_0q_{v_3}) & q_0^2 - q_{v_1}^2 + q_{v_2}^2 - q_{v_3}^2 & 2(q_{v_2}q_{v_3} - q_0q_{v_1}) \\ 2(q_{v_1}q_{v_3} - q_0q_{v_2}) & 2(q_{v_2}q_{v_3} + q_0q_{v_1}) & q_0^2 - q_{v_1}^2 - q_{v_2}^2 + q_{v_3}^2 \end{bmatrix}. \quad (\text{D.2})$$

The opposite conversion, from  $\{G\}$  to  $\{L\}$ , is done with

$$x_L = q_{LG} \otimes x_G \otimes q_{LG}^* \quad \text{or} \quad x_L = {}^L R_G x_G,$$

where

$$q_{LG} = q_{GL}^* \quad \text{and} \quad {}^L R_G = {}^G R_L^T.$$

The rotation matrix  $R(q)$  given by equation (D.2) has the following properties with respect to the quaternion:

$$R\left(\begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}\right) = I \quad (\text{D.3})$$

$$R(-q) = R(q) \quad (\text{D.4})$$

$$R(q^*) = R^T(q) \quad (\text{D.5})$$

$$R(q_A \otimes q_B) = R(q_A) R(q_B) \quad (\text{D.6})$$

from which one can conclude that [99]: 1) the identity quaternion corresponds to the null rotation (D.3); 2) a quaternion and its negative corresponds to the exact same rotation (D.4); 3)

the conjugate quaternion corresponds to the inverse rotation (D.5); and 4) the quaternion product corresponds to consecutive rotations in the same order as rotation matrices do (D.6).

Let  $\omega_L$  be the angular velocity of  $\{L\}$  with respect to  $\{G\}$  written in  $\{L\}$ , and let  $\omega_G$  be that same angular velocity vector written in  $\{G\}$ . The time derivative of the unit quaternion  $q$  is the vector of quaternion rates, which is related to the angular velocity  $\omega_L$  through

$$\begin{aligned}\dot{q}_{GL} &= \frac{1}{2} q_{GL} \otimes \bar{\omega}_L \\ {}^G\dot{R}_L &= {}^G R_L S(\omega_L),\end{aligned}$$

and to the angular velocity  $\omega_G$  through

$$\begin{aligned}\dot{q}_{GL} &= \frac{1}{2} \bar{\omega}_G \otimes q_{GL} \\ {}^G\dot{R}_L &= S(\omega_G) {}^G R_L,\end{aligned}$$

where  $S(\cdot)$  is a skew-symmetric matrix that satisfies  $S(a)b = a \times b$ .



# Appendix E

## Lyapunov stability fundamentals

This appendix provides a review of the Lyapunov stability theory used in this thesis. Further details can be found in [88].

### E.1 Stability definitions

Suppose one would like to analyse the stability of the dynamical system

$$\dot{x} = f(x, t) \quad x(t_0) = x_0 \quad x \in \mathbb{R}^n \quad (\text{E.1})$$

where  $f$  is locally Lipschitz in  $x$  and piecewise continuous in  $t$ . The equilibrium point  $x^* = 0$  of the nonautonomous system (E.1) is

- Stable (in the sense of Lyapunov) at  $t = t_0$  if for any  $\varepsilon > 0$  there exists  $\delta(t_0, \varepsilon) > 0$  such that

$$\|x(t_0)\| < \delta \implies \|x(t)\| < \varepsilon, \quad \forall t \geq t_0 \geq 0 \quad (\text{E.2})$$

- Asymptotically Stable at  $t = t_0$  if  $x^* = 0$  is stable and locally attractive, i.e.,

$$\exists c = c(t_0) : x(t_0) \rightarrow 0 \quad \text{as} \quad t \rightarrow \infty \quad \forall \|x(t_0)\| < c. \quad (\text{E.3})$$

- Unstable if is not stable.

Note that, in the above definitions, stability of a given equilibrium point depends on the initial time instant  $t_0$ . To remove this dependency, the concept of Uniform Stability is introduced: the equilibrium point  $x^* = 0$  of the nonautonomous system (E.1) is

- Uniformly Stable if  $\delta$  is independent of  $t_0$ , such that equation E.2 holds  $\forall t_0$ . Uniform stability of a system guarantees that the equilibrium point is not losing stability.

- Uniformly Asymptotically Stable if  $c$  is independent of  $t_0$  and the convergence is uniformly in  $t_0$ , that is

$$\forall \eta > 0 \exists T = T(\eta) > 0 : \|x(t)\| < \eta, \quad \forall t \geq t_0 + \tau(\eta), \quad \forall \|x(t_0)\| < c \quad (\text{E.4})$$

- Globally Uniformly Asymptotically Stable if the inequality in (E.4) holds for all initial conditions  $x(t_0)$ , that is  $c = \infty$ .

Note that the above definitions for asymptotic stability do not quantify the rate of convergence for the system. A stronger form of stability is the exponential stability which imposes an exponential convergence rate to the equilibrium point. Formally, the equilibrium point  $x^* = 0$  is

- Exponentially Stable if there exists positive constants  $c$ ,  $k$  and  $\lambda$  such that

$$\|x(t)\| \leq k\|x(t_0)\|e^{-\lambda(t-t_0)}, \quad \forall \|x(t_0)\| < c \quad (\text{E.5})$$

- Globally Exponentially Stable if inequality (E.5) is satisfied  $\forall x(t_0)$ .

## E.2 Lyapunov's direct method

Lyapunov's direct method allows to determine the stability of a system without explicitly integrating the differential equation (E.1). Intuitively, the idea is to define the “energy” of the system and infer the stability properties of the system by studying the rate of change of its “energy”. If the system loses energy over time and the energy is never restored, then, eventually, the system will reach some final resting state at a stable equilibrium point. The following paragraphs formalize this idea.

A continuous function  $V : [0, \infty) \times D \rightarrow \mathbb{R}$ , with  $D \subset \mathbb{R}^n$  is Locally Positive Definite if, for some continuous, strictly increasing function  $\alpha : \mathbb{R}_+ \rightarrow \mathbb{R}$

$$V(t, 0) = 0 \quad \text{and} \quad V(t, x) \geq \alpha(\|x\|) \quad \forall x \in D, \quad \forall t \geq 0. \quad (\text{E.6})$$

Considering a positive definite function  $V(t, x)$  as defined in equation (E.6), the equilibrium point  $x^* = 0$  of the nonautonomous system (E.1) is

- Uniformly stable if

$$W_1(x) \leq V(t, x) \leq W_2(x) \quad \text{and} \quad (\text{E.7})$$

$$\frac{dV}{dt} + \frac{dV}{dx}f(t, x) \leq 0, \quad \forall t \geq 0 \quad \forall x \in D, \quad (\text{E.8})$$

where  $W_1(x)$  and  $W_2(x)$  are continuous positive functions on  $D$ .

- Uniformly Asymptotically Stable if, in addition to inequalities (E.7) and (E.8),

$$\frac{dV}{dt} + \frac{dV}{dx}f(t,x) \leq -W_3(x), \quad \forall t \geq 0 \quad \forall x \in D, \quad (\text{E.9})$$

where  $W_3(x)$  is a continuous positive function on  $D$ .

- Globally Uniformly Asymptotically Stable if, in addition to conditions (E.7) to (E.9),  $D = \mathbb{R}^n$  and  $W_1(x)$  is radially unbounded.

The above conditions are sufficient to assess the stability of an equilibrium point of a nonautonomous system. However, a method for determining an adequate Lyapunov function  $V(t,x)$  is not provided and therefore, the search for a Lyapunov function establishing stability of an equilibrium point can be very difficult. Nonetheless, it is worth noticing that, if an equilibrium point is stable, then there exists a function  $V(t,x)$  satisfying one of the above conditions.

Lyapunov's direct method can also be used to verify exponential stability of a nonautonomous system. Considering a positive definite function  $V(t,x)$  as defined in equation (E.6), the equilibrium point  $x^* = 0$  of the nonautonomous system (E.1) is

- Exponentially Stable if there exist positive constants  $k_1, k_2, k_3$  and  $a$  such that

$$k_1 \|x\|^a \leq V(t,x) \leq k_2 \|x\|^a \quad \text{and} \quad (\text{E.10})$$

$$\frac{dV}{dt} + \frac{dV}{dx}f(t,x) \leq -k_3 \|x\|^a \quad (\text{E.11})$$

- Globally Exponentially Stable if inequalities (E.10) and (E.11) hold globally.

### E.3 Linear time-varying systems

Consider the following linear time-varying system

$$\dot{x}(t) = A(t)x(t). \quad (\text{E.12})$$

where  $A(t)$  is continuous and bounded. Let  $x^* = 0$  be an exponential stable equilibrium point of the linear system (E.12) and let  $Q(t) = Q^T(t) > 0$  be continuous and bounded matrix. Then, there is a  $P(t) = P^T(t) \in C^2$  bounded matrix that satisfies

$$-\dot{P}(t) = P(t)A(t) + A^T(t)P(t) + Q(t). \quad (\text{E.13})$$

Moreover,  $V(t,x) = x^T P(t)x$  is a Lyapunov function that satisfies (E.10) and (E.11), with  $a = 2$  and  $f(t,x) = A(t)x(t)$ .

## E.4 Linearization

One can also use the linearization of a system to determine the local stability of the original system using Lyapunov methods. Consider the nonautonomous system (E.1) with the origin as an equilibrium point, where  $f: [0, \infty] \times D \rightarrow \mathbb{R}^n$ ,  $D = \{x \in \mathbb{R}^2: \|x\| < r\}$  is continuously differentiable, and the Jacobian matrix  $\left[\frac{df}{dx}\right]$  is bounded and Lipschitz on  $D$ , uniformly in  $t$ . Let

$$A(t) = \frac{df}{dx}(t, x)|_{x=0}, \quad (\text{E.14})$$

Then, the origin is an exponentially stable equilibrium point of (E.1) if and only if it is an exponentially stable equilibrium point for the linear system (E.12).

## E.5 Barbalat's Lemma

Given a function that tends towards a finite limit as  $t \rightarrow \infty$ , Barbalat's lemma indicates that the derivative itself should have some smoothness. More precisely [105]

**Barbalat Lemma:** If the differentiable function  $f(t)$  has a finite limit as  $t \rightarrow \infty$  and if  $\dot{f}(t)$  is uniformly continuous, then  $\dot{f}(t) \rightarrow 0$  as  $t \rightarrow \infty$ .

Usually, the asymptotic stability of time-varying systems is difficult to analyse because Lyapunov functions with negative definite derivatives are very arduous to find. Barbalat's lemma can be applied to the analysis of dynamic systems using the following immediate corollary [105]

**Lyapunov-Like Lemma:** If a scalar function  $V(t, x)$  satisfies the following conditions

- $V(t, x)$  is lower bounded
- $\dot{V}(t, x)$  is negative semi-definite
- $\dot{V}(t, x)$  is uniformly continuous in time

then  $\dot{V}(t, x) \rightarrow 0$  as  $t \rightarrow \infty$ .

# References

- [1] U. S. D. of Transportation, “Unmanned aircraft system (UAS) service demand 2015 - 2035: Literature review and projections of future usage,” Department of the Air Force, Tech. Rep., 2013.
- [2] E. Galceran and M. Carreras, “A survey on coverage path planning for robotics,” *Robotics and Autonomous Systems*, vol. 61, pp. 1258–1276, 2013.
- [3] B. Grocholsky, J. Keller, V. Kumar, and G. Pappas, “Cooperative air and ground surveillance,” *IEEE Robotics & Automation Magazine*, vol. 13, no. 3, pp. 16–25, 2006.
- [4] M. Nagai, T. Chen, R. Shibasaki, H. Kumagai, and A. Ahmed, “UAV-borne 3-D mapping system by multisensor integration,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 3, pp. 701–708, 2009.
- [5] D. Kingston, R. W. Beard, and R. S. Hol, “Decentralized perimeter surveillance using a team of UAVs,” *IEEE Transactions on Robotics*, vol. 24, pp. 1394–1404, 2008.
- [6] 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.
- [7] 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 Exhibit*, 2006.
- [8] M. Dunbabin and L. Marques, “Robots for environmental monitoring: Significant advancements and applications,” *IEEE Robotics & Automation Magazine*, vol. 19, pp. 24–39, 2012.
- [9] J. Tisdale, Z. Kim, and K. Hedrick, “Autonomous UAV path planning and estimation,” *IEEE Robotics & Automation Magazine*, vol. 16, pp. 35–42, 2009.
- [10] G. Cruz and P. Encarnação, “Obstacle avoidance for unmanned aerial vehicles,” *Journal of Intelligent & Robotic Systems*, vol. 65, pp. 203–217, 2012.
- [11] A. P. 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.
- [12] E. Franco, L. Magni, T. Parisini, M. Polycarpou, and D. Raimondo, “Cooperative constrained control of distributed agents with nonlinear dynamics and delayed information exchange: A stabilizing receding-horizon approach,” *IEEE Transactions on Automatic Control*, vol. 53, pp. 324–338, 2008.

- [13] T. Tomic, K. Schmid, P. Lutz, A. Domel, M. Kassecker, E. Mair, I. Grix, F. Ruess, M. Suppa, and D. Burschka, "Toward a fully autonomous UAV: Research platform for indoor and outdoor urban search and rescue," *IEEE Robotics & Automation Magazine*, vol. 19, pp. 46–56, 2012.
- [14] R. Rysdyk, "Unmanned aerial vehicle path following for target observation in wind," *Journal of Guidance, Control, and Dynamics*, September-October, vol. 29, no. 5, pp. 1092–1100, 2006.
- [15] E. Xargay, I. Kaminer, A. Pascoal, N. Hovakimyan, V. Dobrokhodov, V. Cichella, A. P. Aguiar, and R. Ghabcheloo, "Time-critical cooperative path following of multiple unmanned aerial vehicles over time-varying networks," *Journal of Guidance, Control, and Dynamics*, vol. 36, pp. 499–516, 2013.
- [16] 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.
- [17] A. P. Aguiar, J. Hespanha, and P. Kokotovic, "Path-following for nonminimum phase systems removes performance limitations," *IEEE Transactions on Automatic Control*, vol. 50, no. 2, pp. 234–239, 2005.
- [18] A. P. Aguiar, J. Hespanha, and P. Kokotović, "Performance limitations in reference tracking and path following for nonlinear systems," *Automatica*, vol. 44, no. 3, pp. 598–610, 2008.
- [19] R. Hindman and J. Hauser, "Maneuver modified trajectory tracking," in *Proceedings of MTNS'96 International Symposium on the Mathematical Theory of Networks and Systems*, 1992.
- [20] L. Lapiere, 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.
- [21] A. Rucco, A. P. 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, pp. 236–245.
- [22] ———, "A virtual target approach for trajectory optimization of a general class of constrained vehicles," in *Proceedings of the 54th IEEE Conference on Decision and Control*, 2015.
- [23] C. Canudas de Wit, H. Khenouf, C. Samson, and O. Sordalen, "Nonlinear control design for mobile robots," *Recent Trends in Mobile Robots*, vol. 11, pp. 121–156, 1993.
- [24] 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.
- [25] 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.
- [26] R. Skulstad, C. Syversen, M. Merz, N. Sokolova, T. Fossen, and T. Johansen, "Autonomous net recovery of fixed-wing UAV with single-frequency carrier-phase differential GNSS," *IEEE Aerospace and Electronic Systems Magazine*, vol. 30, no. 5, pp. 18–27, 2015.

- [27] L. Lapierre and D. Soetanto, "Nonlinear path following control of an AUV," *Journal of Oceanic Engineering*, vol. 34, pp. 1734–1744, 2007.
- [28] 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.
- [29] T. Fossen, K. Pettersen, and R. Galeazzi, "Line-of-sight path following for dubins paths with adaptive sideslip compensation of drift forces," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 2, pp. 820–827, 2015.
- [30] J. Wit, C. Crane, and D. Armstrong, "Autonomous ground vehicle path tracking," *Journal of Robotic Systems*, vol. 21, no. 8, pp. 439–449, 2004.
- [31] J. Giesbrecht, D. Mackay, J. Collier, and S. Verret, "Path tracking for unmanned ground vehicle navigation: Implementation and adaptation of the pure pursuit algorithm," DTIC Document, Tech. Rep., 2005.
- [32] A. J. Healey and D. Lienard, "Multivariable sliding mode control for autonomous diving and steering of unmanned underwater vehicles," *IEEE Journal of Oceanic Engineering*, vol. 18, no. 3, pp. 327–339, 1993.
- [33] D. R. Nelson, D. B. Barber, T. W. McLain, and R. W. Beard, "Vector field path following for miniature air vehicles," *IEEE Transactions on Robotics*, vol. 23, no. 3, pp. 519–529, 2007.
- [34] E. Alfaro-Cid, E. McGookin, D. Murray-Smith, and T. Fossen, "Genetic algorithms optimisation of decoupled sliding mode controllers: simulated and real results," *Control Engineering Practice*, vol. 13, no. 6, pp. 739–748, 2005.
- [35] Z. Li, J. Sun, and S. Oh, "Handling roll constraints for path following of marine surface vessels using coordinated rudder and propulsion control," in *Proceedings of the 2010 American Control Conference*, 2010.
- [36] S. Jackson, J. Tisdale, M. Kamgarpour, B. Basso, and K. Hedrick, "Tracking controllers for small UAVs with wind disturbances: Theory and flight results," in *Proceedings of the 47th IEEE Conference on Decision and Control*, 2008.
- [37] T. Fossen and A. Grovlen, "Nonlinear output feedback control of dynamically positioned ships using vectorial observer backstepping," *IEEE transactions on control systems technology*, vol. 6, no. 1, pp. 121–128, 1998.
- [38] P. Encarnação and A. Pascoal, "Combined trajectory tracking and path following: An application to the coordinated control of autonomous marine craft," in *Proceedings of the 40th IEEE Conference on Decision and Control*, 2001.
- [39] B. Wang, X. Dong, and B. M. Chen, "Cascaded control of 3D path following for an unmanned helicopter," in *Proceedings of the IEEE Conference on Cybernetics and Intelligent Systems*, 2010.
- [40] S. Lee, A. Cho, and C. Kee, "Integrated waypoint path generation and following of an unmanned aerial vehicle," *Aircraft Engineering and Aerospace Technology*, vol. 82, no. 5, pp. 296–304, 2010.

- [41] R. Cunha, C. Silvestre, and A. Pascoal, "A path following controller for model-scale helicopters," in *Proceedings of the European Control Conference*, 2003.
- [42] C. Cao, N. Hovakimyan, I. Kaminer, V. Patel, and V. Dobrokhodov, "Stabilization of cascaded systems via L1 adaptive controller with application to a UAV path following problem and flight test results," in *Proceedings of the American Control Conference*, 2007.
- [43] A. P. Aguiar, I. Kaminer, R. Ghabcheloo, A. Pascoal, E. Xargay, N. Hovakimyan, C. Cao, and V. Dobrokhodov, "Coordinated path following of multiple UAVs for time-critical missions in the presence of time-varying communication topologies," *IFAC Proceedings Volumes*, vol. 4, no. 2, pp. 16 015–16 020, 2008.
- [44] J. E. da Silva and J. Sousa, "A dynamic programming approach for the motion control of autonomous vehicles," in *Proceedings of the 49th IEEE Conference on Decision and Control*, 2010.
- [45] R. Bencatel, "Perpetual flight in flow fields," Ph.D. dissertation, University of Porto Engineering School, 2012.
- [46] 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.
- [47] 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.
- [48] R. Wise and R. Rysdyk, "UAV coordination for autonomous target tracking," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2006, pp. 21–24.
- [49] 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, pp. 363–372.
- [50] 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.
- [51] 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.
- [52] E. W. Weisstein. (2016) Lemniscate. From MathWorld—A Wolfram Web Resource. [Online]. Available: <http://mathworld.wolfram.com/Lemniscate.html>
- [53] 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, pp. 359–364.
- [54] 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.
- [55] Z. Tang and U. Özgüner, "Motion planning for multitarget surveillance with mobile sensor agents," *IEEE Transactions on Robotics*, vol. 21, pp. 898–908, 2005.



- [56] 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.
- [57] 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.
- [58] 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.
- [59] M. Breivik and T. Fossen, "Guidance laws for planar motion control," in *IEEE Conference on Decision and Control*, 2008, pp. 570–577.
- [60] J. Swee, "Missile terminal guidance and control against evasive targets," Master's thesis, Naval Postgraduate School, 2000.
- [61] L. Dubins, "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents," *American Journal of Mathematics*, vol. 79, pp. 497–516, 1957.
- [62] D. Barber, J. Redding, T. McLain, R. Beard, and C. Taylor, "Vision-based target geolocation using a fixed-wing miniature air vehicle," *Journal of Intelligent and Robotic Systems*, vol. 47, pp. 361–382, 2006.
- [63] S. Spry, A. Girard, and J. Hedrick, "Convoy protection using multiple unmanned aerial vehicles: organization and coordination," in *Proceedings of the 2005 American Control Conference*, 2005.
- [64] R. Skulstad and C. Syversen, "Low-cost instrumentation system for recovery of fixed-wing UAV in a net," Master's thesis, Norwegian University of Science and Technology, 2014.
- [65] A. Smit, "Autonomous landing of a fixed-wing unmanned aerial vehicle using differential GPS," Master's thesis, Stellenbosch University, 2013.
- [66] T. Muskardin, G. Balmer, L. Persson, S. Wlach, M. Laiacker, A. Ollero, and K. Kondak, "A novel landing system to increase payload capacity and operational availability of high altitude long endurance UAV," in *Proceedings of the International Conference on Unmanned Aircraft Systems (ICUAS)*, 2016.
- [67] A. Rucco, P. Sujit, A. P. Aguiar, and J. Sousa, "Optimal UAV rendezvous on a UGV," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, 2016.
- [68] H. J. Kim, M. Kim, H. Lim, C. Park, S. Yoon, D. Lee, H. Choi, G. Oh, J. Park, and Y. Kim, "Fully autonomous vision-based net-recovery landing system for a fixed-wing UAV," *IEEE/ASME Transactions on Mechatronics*, vol. 18, pp. 1320–1333, 2013.
- [69] K. Wenzel, A. Masselli, and A. Zell, "Automatic take off, tracking and landing of a miniature UAV on a moving carrier vehicle," *Journal of Intelligent & Robotic Systems*, vol. 61, no. 1, pp. 221–238, 2011.
- [70] B. Barber, T. McLain, and B. Edwards, "Vision-based landing of fixed-wing miniature air vehicles," *Journal of Aerospace Computing, Information, and Communication*, vol. 6, no. 3, pp. 207–226, 2009.

- [71] J.-C. Yin, Z.-J. Zou, and F. Xu, "On-line prediction of ship roll motion during maneuvering using sequential learning RBF neural networks," *Ocean Engineering*, vol. 61, pp. 139–147, 2013.
- [72] 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.
- [73] T. Oliveira and P. Encarnação, "Ground target tracking for unmanned aerial vehicles," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, 2010.
- [74] —, "Ground target tracking control system for unmanned aerial vehicles," *Journal of Intelligent and Robotic Systems*, vol. 69, pp. 373–387, 2013.
- [75] T. Oliveira, P. Encarnação, and A. P. Aguiar, "Moving path following for autonomous robotic vehicles," in *Proceedings of the European Control Conference (ECC)*, 2013.
- [76] T. Oliveira, A. P. Aguiar, and P. Encarnação, "A convoy protection strategy using the moving path following method," in *Proceedings of the International Conference on Unmanned Aircraft Systems (ICUAS)*, 2016.
- [77] —, "Moving path following for unmanned aerial vehicles with applications to single and multiple target tracking problems," *IEEE Transactions on Robotics*, vol. 32, no. 5, pp. 1062–1078, 2016.
- [78] E. Zereik, M. Bibuli, M. Caccia, and G. Bruzzone, "Cooperative path-following in a moving path reference framework for autonomous marine vehicles," in *Proceedings of the 21st Mediterranean Conference on Control & Automation*, 2013.
- [79] T. Oliveira, A. P. Aguiar, and P. Encarnação, "Three dimensional moving path following for fixed-wing unmanned aircraft vehicles," in *Submitted to the IEEE International Conference on Robotics and Automation*, 2017.
- [80] J. Wang, I. Kapitaniuk, S. Chepinskiy, D. Liu, and A. Krasnov, "Geometric path following control in a moving frame," in *Proceedings of the 1st IFAC Conference on Modelling, Identification and Control of Nonlinear Systems*, 2015.
- [81] J. Morgado and J. Sousa, "Projecto de investigação e tecnologia em veículos aéreos não-tripulados (PITVANT)," in *UAV's Seminar – DANOTEC EuroDefense Portugal*, 2009.
- [82] M. Marques, P. Dias, N. Santos, V. Lobo, R. Batista, D. Salgueiro, A. P. Aguiar, J. Sousa, A. Bernardino *et al.*, "Unmanned aircraft systems in maritime operations: Challenges addressed in the scope of the SEAGULL project," in *Proceedings of OCEANS*, 2015.
- [83] 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, pp. 3043–3044.
- [84] 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.
- [85] L. Bishop, "There is more than one way to frame a curve," in *Amer. Math. Monthly* 82, March 1975, pp. 246–251.

- [86] A. Hanson and H. Ma, "Parallel transport approach to curve framing," Indiana University, Tech. Rep., 1995.
- [87] R. Beard and T. McLain, *Small unmanned aircraft: Theory and practice*. Princeton University Press, 2012.
- [88] H. Khalil, *Nonlinear Systems*, 3rd ed., 2002, Prentice Hall.
- [89] R. Anderson and D. Milutinovic, "Dubins vehicle tracking of a target with unpredictable trajectory," in *ASME 2011 Dynamic Systems and Control Conference and Bath/ASME Symposium on Fluid Power and Motion Control*, 2011, pp. 675–682.
- [90] J. Diebel, "Representing attitude: Euler angles, unit quaternions, and rotation vectors," *Matrix*, vol. 58, pp. 1–35, 2006.
- [91] A. Lekkas and T. Fossen, "A quaternion-based LOS guidance scheme for path following of AUVs," *IFAC Proceedings Volumes*, vol. 46, no. 33, pp. 245–250, 2013.
- [92] E. W. Weisstein. (2016) Law of cosines. From MathWorld—A Wolfram Web Resource. [Online]. Available: <http://mathworld.wolfram.com/LawofCosines.html>
- [93] S. Quintero, F. Papi, D. Klein, L. Chisci, and J. Hespanha, "Optimal UAV coordination for target tracking using dynamic programming," in *Proceedings of the 49th IEEE Conference on Decision and Control*, 2010.
- [94] *Piccolo Hardware-in-Loop / Software-in-Loop Setup Guide*, Cloud Cap Technology, 2008.
- [95] 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, p. 5.
- [96] E. W. Weisstein. (2016) Curvature. From MathWorld—A Wolfram Web Resource. [Online]. Available: <http://mathworld.wolfram.com/Curvature.html>
- [97] J. Lagarias, J. Reeds, M. Wright, and P. Wright, "Convergence properties of the nelder-mead simplex method in low dimensions," *SIAM Journal on Optimization*, vol. 9, pp. 112–147, 1998.
- [98] A. P. 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, pp. 3858–3863.
- [99] J. Solà, "Quaternion kinematics for the error-state kf," Laboratoire d'Analyse et d'Architecture des Systemes - Centre National de la Recherche Scientifique (LAAS-CNRS), Toulouse, France, Tech. Rep., 2012.
- [100] O. Rodrigues, *Des lois géométriques qui régissent les déplacements d'un système solide dans l'espace: et de la variation des coordonnées provenant de ces déplacements considérés indépendamment des causes qui peuvent les produire*. Publisher not identified, 1840.
- [101] E. Whittaker, *Analytical dynamics of particles and rigid bodies*. University Press, 1952.
- [102] H. Schaub and J. Junkins, *Analytical mechanics of space systems*. AIAA, 2003.

- [103] J. Chou, “Quaternion kinematic and dynamic differential equations,” *IEEE Transactions on Robotics and Automation*, vol. 8, no. 1, pp. 53–64, 1992.
- [104] A. Martinelli, “Vision and IMU data fusion: Closed-form solutions for attitude, speed, absolute scale, and bias determination,” *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 44–60, 2012.
- [105] J. Slotine and W. Li, *Applied nonlinear control*. Prentice-Hall Englewood Cliffs, NJ, 1991.