Navigation and coordination of fixed-wing unmanned aerial vehicles under mission uncertainty

Wang, X.

DOI
10.4233/uuid:d9752495-9c32-4612-b9c0-e1054c1b764f

Publication date
2022

Document Version
Final published version

Citation (APA)
Wang, X. (2022). *Navigation and coordination of fixed-wing unmanned aerial vehicles under mission uncertainty*. [Dissertation (TU Delft), Delft University of Technology]. https://doi.org/10.4233/uuid:d9752495-9c32-4612-b9c0-e1054c1b764f

NAVIGATION AND COORDINATION OF FIXED-WING UNMANNED AERIAL VEHICLES UNDER MISSION UNCERTAINTY

NAVIGATION AND COORDINATION OF FIXED-WING UNMANNED AERIAL VEHICLES UNDER MISSION UNCERTAINTY

Dissertation

for the purpose of obtaining the degree of doctor
at Delft University of Technology
by the authority of the Rector Magnificus Prof. Dr. Ir. T.H.J.J. van der Hagen,
chair of the Board for Doctorates
to be defended publicly on
Wednesday 16 November 2022 at 10:00 o'clock

by

Ximan WANG

Master of Science in Advanced Control System and Engineering,
The University of Sheffield, The United Kingdom,
born in Taiyuan, China.

Dit proefschrift is goedgekeurd door de

promotor: prof. dr. ir. B. De Schutter, and
promotor: prof. dr. S. Baldi

Samenstelling promotiecommissie:

Rector Magnificus,	voorzitter
Prof. dr. S. Baldi	Technische Universiteit Delft, promotor
Prof. dr. ir. B. De Schutter	Technische Universiteit Delft, promotor
Onafhankelijke leden:	
Prof. dr. Erdal Kayacan	Aarhus University
Prof. dr. Michael Defoort	Polytechnic University of Hauts-de-France
Prof. dr. D. Casalino	Technische Universiteit Delft
Prof. dr. R. Carloni	Rijksuniversiteit Groningen
Prof. dr. ir. G. C. H. E. de Croon	Technische Universiteit Delft

Research described in this thesis was supported by the China Scholarship Council (CSC) under grant 201907720110, and by the Delft Center for Systems and Control (DCSC).

Published and distributed by: Ximan Wang

ISBN 978-94-6384-387-4

Keywords: fixed-wing UAV, vector field,unknown dynamics, adaptive guidance control, formation control.

Printed by RIDDERPRINT in the Netherlands

To my family

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my promoter prof. Bart De Schutter and second promoter prof. Simone Baldi for allowing me to carry out this doctoral research. I am privileged to have Bart as my promoter and I would like to express my sincere gratitude to him. The four years of PhD study would not be possible without the help of Bart.

I wish to express my gratitude to my supervisor prof. Simone. I really appreciated your tireless support and guidance and it was a pleasure working with you. Your patience, modesty, and willingness to make time for me whenever I needed, especially gave me freedom and trust, making me confident in doing my research.

I sincerely thank my PhD committee members, prof. Michael Defoort (Polytechnic University of Hauts-de-France), prof. Raffaella Carloni (University of Groningen), prof. Erdal Kayacan (Aarhus University), prof. Damiano Casalino (TU Delft) and prof. Guido de Croon (TU Delft) for the time and efforts which they spent to thoroughly read my thesis and giving me such valuable and constructing feedback.

I would like to thank Mr. Jun Yang for supporting the visit and experiments. I really enjoyed the productive time in discussion or sitting in the lab that results in the success of the flight tests. I also want to thank Changwei, Xuewei, Yuchen, and prof. Hongwei Xie for their efforts in the experiment.

Since I and my wife arrived in the Netherlands, a lot of my friends helped us to get familiar with the environment, enjoy time in the Netherlands, and even help us to take care of my son. Among these friends are Wenli, Xiang, Jun, Qisong, Yang, Jiawen, Shuang, Na, Chunzi, Hongxin and Jiayu, Xi and Xiaochen, Wei and Tao, Qin and Hui, and Youliang.

Furthermore, I would like to thank all my kind and friendly colleagues. Thank Erica, Marieke, Heleen, Francy, and Martha for organizing attractive additives and supporting our daily work. Thank prof. Peyman encouraged me before Go/Nogo meeting. I would like to thank Jingwei and Jieting for their kind help in my research and daily learning.I would also thanks Dingshan and Yun for the wonderful time while sharing the same office. I also want to thank Xiaoyu, Tian, Ping, Raja, Amin, Mattia, Jianfeng, Yaolu, Shengling, Kanghui, and Leila.

Last, I would like to extend my indebtedness to my parents for their endless love and support to achieve a higher target. I would like to thank my wife, Xiaoqing, for her love, effort, and courage throughout these years. Without you, it would be impossible to finish this thesis. Thanks for your payout for taking care of, and educating our son. I would bless our son Yujie's happiness and health forever.

Ximan Wang
Delft, October 2022.

SUMMARY

Unmanned Aerial Vehicles (UAVs) have been emerging as a promising but challenging platform for studying autonomous and cooperative control. This Ph.D. thesis focuses on fixed-wing UAVs which, with their more efficient aerodynamics, can ensure longer flight durations and more autonomy than multi-rotor UAVs. However, in the current state of the art, limited work has been done on deploying formations of fixed-wing UAVs that can operate autonomously even in the presence of large uncertainties. Uncertainties in fixed-wing UAVs include uncertain wind environments, unmodelled longitudinal/lateral dynamics, uncertain load conditions, uncertain communication conditions among the UAVs, and other uncertain factors.

Within this PhD thesis we develope novel adaptive and distributed guidance approaches for fixed-wing UAVs. The following three aspects are studied:

- **Vector field guidance under uncertainties.**

  The vector field methodology has been proven that effective for path to be following in time-varying wind environments. However, as the methodology relies on the wind being a constant and known parameter, and the UAV course dynamics being known, the autonomous capabilities of this methodology are limited. We embed adaptation mechanisms in the vector field methodology to achieve better operation in uncertain wind environments and with uncertain course dynamics.

- **Distributed formation control with uncertain UAV dynamics.**

  Formation control can be framed as a synchronization problem, and several formation algorithms have been proposed in literature. However, very few actual formations of fixed-wing UAVs have been reported in literature, one reason being the difficulties in handling uncertainty and the heterogeneities of the UAV dynamics. We develop a distributed formation control approach in the framework of model reference adaptive control and we also propose a non-uniform vector field method that changes in both magnitude and direction to achieve formations. We show successful tests of a formation flying with 5-10 fixed-wing UAVs.

- **Testing in the real world to achieve Sim-to-Real transfer**

  All proposed methods are validated in software-in-the-loop and hardware-in-the-loop comparative experiments with the state-of-the-art. Also, with the aim to go from simulation environments to real flight testing, we analyze typical uncertainties occurring during flight testing and provide suitable methods for handling them. Real world testing is achieved via formation flight with 5 fixed-wing UAVs. The results of the real flight testing prove that the performance of the methods we propose overcomes that of the state-of-the-art.

All the methods in this thesis are implemented on open-source flight controllers (mostly PX4, sometimes Ardupilot) and tested in the Gazebo software-in-the-loop and hardware-in-the-loop simulation environment. The results of vector field guidance and formation control are also verified by means of real flights.

SAMENVATTING

Onbemande luchtvaartuigen (UAVs) zijn in opkomst als een veelbelovend maar uitdagend platform voor het bestuderen van autonome en coöperatieve controle. Dit proefschrift richt zich op UAVs met vaste vleugels die, met hun efficiëntere aerodynamica, een langere vluchtduur en meer autonomie kunnen garanderen dan multi-rotor UAVs. In de huidige stand van de techniek is er echter weinig werk verricht aan het inzetten van formaties van UAVs met vaste vleugels die autonoom kunnen opereren, zelfs in de aanwezigheid van grote onzekerheden. Onzekerheden in fixed-wing UAVs omvatten onzekere windomgevingen, niet-gemodelleerde longitudinale/laterale dynamiek, onzekere belastingscondities, onzekere communicatiecondities tussen de UAVs, en andere onzekere factoren.

In dit proefschrift ontwikkelen we nieuwe adaptieve en gedistribueerde geleidingsmethoden voor UAVs met vaste vleugels. De volgende drie aspecten worden bestudeerd:

- **Vectorveldgeleiding onder onzekerheden.**

  De vectorveldmethode is effectief gebleken voor het volgen van een pad in een tijdsafhankelijke windomgeving. Maar omdat de methodologie ervan uitgaat dat de wind een constante en bekende parameter is, en dat de koersdynamiek van de UAV bekend is, zijn de autonome mogelijkheden van deze methodologie beperkt. Wij integreren aanpassingsmechanismen in de vectorveldmethode om een betere werking te verkrijgen in onzekere windomgevingen en met onzekere koersdynamiek.

- **Gedistribueerde formatiecontrole met onzekere UAV dynamica.**

  Formatiecontrole kan worden gezien als een synchronisatieprobleem, en in de literatuur zijn verschillende formatie-algoritmen voorgesteld. Er zijn echter zeer weinig werkelijke formaties van UAVs met vaste vleugels gerapporteerd in de literatuur, onder andere vanwege de moeilijkheden bij het omgaan met onzekerheid en de heterogeniteit van de UAV dynamica. Wij ontwikkelen een aanpak voor gedistribueerde formatiecontrole in het kader van modelreferentie-adaptieve controle en stellen tevens een niet-uniforme vectorveldmethode voor die zowel in grootte als in richting verandert om formaties tot stand te brengen. We tonen succesvolle tests van formatievliegen met 5-10 UAVs met vaste vleugels.

- **Testen in de echte wereld om sim-naar-echt overdracht te bewerkstelligen**

  Alle voorgestelde methoden worden gevalideerd in software-in-the-loop en hardware-in-the-loop vergelijkende experimenten met de stand van de techniek. Met het oog op de overgang van simulatieomgevingen naar echte vliegproeven analyseren we ook de typische onzekerheden die optreden tijdens vliegproeven en bieden we geschikte methoden om ze te behandelen. Tests in de echte wereld worden uitgevoerd via formatievluchten met 5 UAV’s met vaste vleugels. De resultaten van de echte testvluchten bewijzen dat de prestaties van de door ons voorgestelde methoden die van de state-of-the-art overtreffen.

Alle methoden in dit proefschrift zijn geïmplementeerd op open-source vluchtcontrollers (meestal PX4, soms Ardupilot) en getest in de Gazebo software-in-the-loop en hardware-in-the-loop simulatieomgeving. De resultaten van vectorveldbesturing en formatiecontrole worden ook geverifieerd aan de hand van echte vluchten.

CONTENTS

1

INTRODUCTION

1.1. MOTIVATION FOR THE RESEARCH

Unmanned Aerial Vehicles (UAVs) or *drones* have been recently emerging as a promising but challenging platform for studying autonomous and cooperative control. A UAV is an electro-mechanical system that can operate autonomously, or that can be operated by remote control, or a combination of both. Among the various types of UAVs, fixed-wing UAVs have become a very popular type of UAV in the industry and they have been studied in different contexts from academic to commercial, due to their energy-efficient performance while carrying payloads [1, 2]. A fixed-wing UAV typically consists of a much simpler structure in comparison to a rotary-wing UAV (quadrotor, helicopter, tilt rotor aircraft, and so on). This simpler structure typically requires a less complicated maintenance and repair process, thus allowing a longer operational time at a lower cost. Most importantly, this simple structure also ensures more efficient aerodynamics that allow longer flight durations at higher speeds with larger payloads.

However, the dynamics, actuation, take-off, and landing of fixed-wing UAVs involve complicated aerodynamics. In addition, the development of cooperative autonomous navigation techniques with the ability to cope with large mission uncertainties (both in the UAV dynamics and in the surrounding environment) constitutes one of the major research challenges for fixed-wing UAVs from the control theory point of view [3–5].

Therefore, this thesis aims at developing novel adaptive and distributed adaptive approaches that can handle in a systemic way mission uncertainties at the single fixed-wing UAV level and at the multiple UAVs level. The relevance of this research is directly connected to the development of enhanced autonomous navigation systems for robotic autonomous agents accomplishing different tasks and involving a large dynamic working range.

With this vision in mind, in the following we will explain the research goals in more detail.

1.2. RESEARCH GOALS AND MAIN CONTRIBUTIONS

In the current state of the art, limited work has been done on deploying navigation and coordination tools to allow fixed-wing UAVs to operate autonomously even in the presence of large uncertainties. Uncertainties in UAV missions include uncertain wind environments, unmodelled longitudinal/lateral dynamics, and uncertain load conditions, just to name a few. The goal of this thesis is to develop new adaptation mechanisms for navigation and coordination of fixed-wing UAVs. The goals and contributions are summarised in the next three subsections, along three main lines.

1.2.1. VECTOR FIELD PATH FOLLOWING WITH UNCERTAINTIES

Any UAV executes its task primarily by flying along a mission path. As such, the UAV must rely on accurate path-following algorithms, and the capability to plan paths and to follow them accurately is of great importance. The goal of the vector field (VF) approach, which combines geometric reasoning with a sliding mode technique, is to drive the relative path error (also be named as cross-track error in this work) to zero by providing to the UAV a field of desired course angles for each point around the desired path. In a recent comparative survey [6], the VF idea has been shown to be more accurate than other path-following methods, at the price of requiring more parameters to be designed. Most importantly, the VF idea has been proven effective for path-following in windy environments [7]. However, the bottleneck of the standard VF idea is to crucially rely on two assumptions: (1) the wind is known and constant; (2) the course dynamics are known and first-order [7]. These two assumptions imply that wind disturbances, unmodelled dynamics, and limited quality of sensing and actuation, all impose critical limits to the achievable accuracy [8–10]. For example, in most real UAVs the course angle dynamics are much more complex than first-order.

With this open problem in mind, in this thesis, several adaptation mechanisms are developed for improving the path-following performance in actual uncertain wind scenarios and uncertain course dynamics scenarios.

1.2.2. DISTRIBUTED FORMATION CONTROL OF UNCERTAIN MULTI-UAVS SYSTEMS

A team of UAVs can be seen as a multi-agent system, where the agents act in a distributed manner to complete global tasks cooperatively with only local information from neighboring agents, so as to increase flexibility and robustness [11]. In the literature, several algorithms have been proposed for the UAV formation problem, which can be seen as a synchronization problem. Although the majority of works on distributed cooperative control consider known and simple dynamical models, UAVs typically fail to satisfy this situation. Thus, taking into account the inherently nonlinear UAV model uncertainties (e.g. Lagrange dynamics) is of paramount importance in formation control schemes [12–15]. The open problem consists in the existence of nonlinear terms with parametric uncertainties and wind uncertainties, so that classic formation algorithms for linear models cannot be directly used to solve the coordination problem for multi-agent systems with Lagrange dynamics.

With this open problem in mind, in this thesis, a distributed model reference adaptive control approach is proposed to synchronize groups of UAVs. Moreover, the vector field idea for single UAV is extended to multi-UAV formations.

1.2.3. SIM-TO-REAL: FORMATION FLIGHT WITH MULTIPLE FIXED-WING UAVS

As standard in literature, most UAV simulators stop at getting the command course angle. This means that many UAV simulators do not take into account the UAV low-level control, which is crucial for successful testing in the real world (the so-called sim-to-real transfer). With this in mind, this thesis discusses how to improve a UAV simulation platform by adding a low-level controller into the platform. This improvement step is based on an open source autopilot for fixed-wing UAVs called Ardupilot.

Then, with the aim to go from simulation environments to performing real flights, we move step by step towards a real UAV airframe, based on the open-source flight suite PX4. We analyze typical uncertainties occurring during flight testing and we provide suitable methods for handling them. The results of the flight tests prove that the performance of the methods we propose overcomes that of the state-of-the-art.

1.3. STRUCTURE OF THE THESIS

After this introductory chapter, some useful background of fixed-wing UAV control system is given in Chapter 2, comprising fixed-wing UAV modeling, high-level control layer, low-level control layer, and UAV simulators.

In Chapter 3, two adaptive vector field guidance control methods are proposed for handling the lack of knowledge of the wind and of the course dynamics.

In Chapter 4, UAV synchronization via model reference adaptive control is introduced and deployed in a distributed way, in the framework of Euler-Lagrange systems.

In Chapter 5, a vector field formation control law is adopted for teams of fixed-wing UAVs, which relies on an appropriately designed sliding mode control method.

In Chapter 6, the theory of formation control is transferred to the real world via real flight experiments. In particular, we illustrate the approach with a real-life flight experiment involving a formation of 5 UAVs and we provide some consideration on how to optimize the formation vector field methods.

Finally, Chapter 7 concludes the thesis and gives some recommendations for future research.

2

BACKGROUND OF FIXED-WING UNMANNED AERIAL VEHICLES (UAVs)

This chapter will review some key notions of UAV modelling and autopilot design. These notions are crucial towards autonomous control for a single fixed-wing UAV and for a team of fixed-wing UAVs.

A quite general control layout [16] for a fixed-wing UAV is shown in Figure 2.1, which presents an abstraction between the different domains needed to control a UAV. According to this abstraction, the control task can be divided into layers, interconnected by interfaces. The first layer is the path manager: this layer receives the main inputs of the mission, which are typically a set of way points. The way points comprise one or more destinations and/or a map. The path manager layer contains several algorithms used for splitting the points and for creating a desired path on the map. Hence, the straight-line and orbit guidance strategies can be used as primitive paths to follow the set of waypoints. After the path manager layer, the desired path is then sent to the path following layer (often referred to as the *guidance* block) calculates the recommended attitude (also referred to as the commanded attitude in some literature [16]) for the UAV for reach and follow the path. The commanded attitude is sent to the low-level controllers (often referred to as the *autopilot* block) that will compute the actuator commands (for instance, move a flap, or increase the speed of the motors). Finally, the UAV aircraft will execute the actuators commands and several on-board sensors will measure the status of the UAV to close the control loop.

Based on this control layout, this chapter will be organized as follows: we will recall dynamic modeling of fixed-wing UAVs in Sect. 2.1. In Sect. 2.2, a popular path manager algorithm for aerial missions will be introduced. In Sect. 2.3 we present the background of the standard vector field guidance law, which is a classical path following method in guidance control. In Sect. 2.4 the main concepts for attitude control are illustrated, including lateral and longitudinal control. Then, the MATLAB simulation platform, the software-in-the-loop simulator, and the hardware-in-the-loop simulator are described in Sect. 2.5. Finally, the airframe we use in the flight testing is introduced in Sect. 2.6.

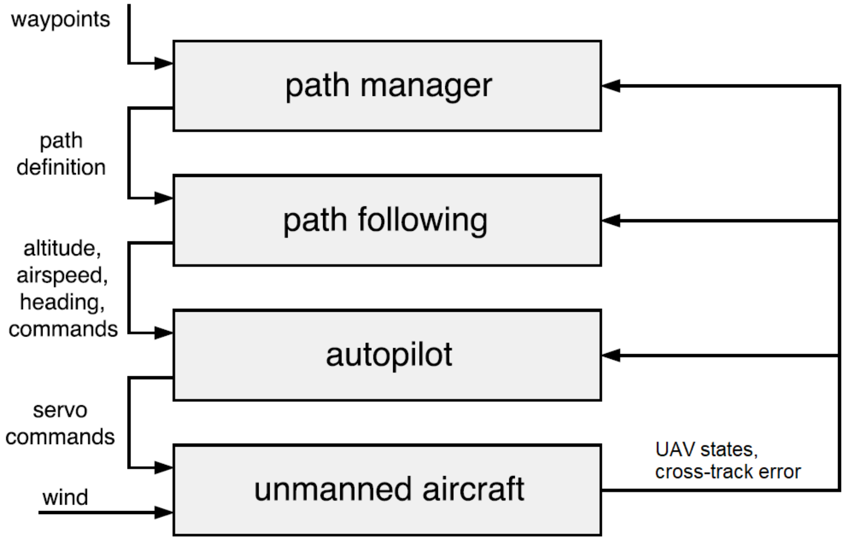


Figure 2.1: General layout for UAV control with autopilot (modified from [16]).

2.1. UAV DYNAMICS

Fixed-wing UAVs can be modelled using 6-DOF Euler-Lagrange equations of motion, which we briefly recall hereafter. More details on the equations can be found in [16, 17].

Let $m \in \mathbb{R}$ be the mass of the UAV, $X_e = [x, y, x]^T \in \mathbb{R}^3$ the inertial position of the body, $V_b = [u, v, w]^T \in \mathbb{R}^3$ the linear velocity of the body expressed in the body frame, $E = [\phi, \theta, \psi]^T \in \mathbb{R}^3$ the Euler angles of the body frame from the inertial frame, $\omega_b = [p, q, r]^T \in \mathbb{R}^3$ the angular velocity of body in the body frame, $F_b \in \mathbb{R}^3$ the net forces acting on the body expressed in body frame, and $M_b \in \mathbb{R}^3$ be the net torques acting on the body expressed in body frame. The inertial frame is an earth-fixed coordinate system with its origin at the defined home location. This coordinate system is sometimes referred to as a North-East-Down (NED) reference frame. The body frame is a vehicle frame. The origin is the center of the mass, the body **x**-axis points out the nose of the airframe, the body **y**-axis points out the right wing, and the body **z**-axis points out the bottom of the airframe.

The equation of motion in the inertial frame is given by Newton's second law of motion as

$$m\dot{V}_e = F_e \tag{2.1}$$

where $V_e \in \mathbb{R}^3$ is the linear velocity of the body in the inertial frame, and $F_e \in \mathbb{R}^3$ are the net forces acting on the body expressed in the inertial frame.

Using standard results, cf. [18, 19], the law of motion in the body frame can be written as

$$m\dot{V}_b + m(\omega_b \times V_b) = F_b. \tag{2.2}$$

Let $\omega_e \in \mathbb{R}^3$ be the angular velocity of the body in the inertial frame, $M_e \in \mathbb{R}^3$ the torques acting on the body expressed in the inertial frame, and $I \in \mathbb{R}^{3\times 3}$ the inertia tensor which is assumed to be constant. For the rotational motion, by Euler's law we can write

$$I\dot{\omega}_e = M_e \tag{2.3}$$

and the dynamics of the rotational velocity in body frame [19] is

$$I\dot{\omega}_b + \omega_b \times I\omega_b = M_b. \tag{2.4}$$

Equations (2.2) and (2.4) constitute the 6 DOF motion equation for a UAV. Both equations can be collected as

$$\begin{bmatrix} m & 0 & 0 & 0 & 0 & 0 \\ 0 & m & 0 & 0 & 0 & 0 \\ 0 & 0 & m & 0 & 0 & 0 \\ 0 & 0 & 0 & I_x & 0 & -I_{xz} \\ 0 & 0 & 0 & 0 & I_y & 0 \\ 0 & 0 & 0 & -I_{xz} & 0 & I_z \end{bmatrix} \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \\ \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} + \begin{bmatrix} 0 & -mr & mq & 0 & 0 & 0 \\ mr & 0 & -mp & 0 & 0 & 0 \\ -mq & mp & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & I_z r - I_{xz} p & -I_y q \\ 0 & 0 & 0 & -I_z r + I_{xz} p & 0 & I_x p - I_{xz} r \\ 0 & 0 & 0 & I_y q & -I_x p + I_{xz} r & 0 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \\ p \\ q \\ r \end{bmatrix} = \begin{bmatrix} F_b \\ M_b \end{bmatrix} \tag{2.5}$$

where the net forces and torques F_b and M_b must include the action of gravity. For simplicity, let us assume that the centre of gravity and the centre of mass are the same which is often the case in UAVs. Thus, the moment due to gravitational action will be neglected.

The gravitational force always acts in the positive **z**-axis direction of the inertial frame

$$F_{ge} = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} \tag{2.6}$$

so that the gravitational force in body frame can be expressed as

$$\begin{aligned} F_{gb} &= R_e^b F_{ge} \\ &= \begin{bmatrix} -mg\sin\theta \\ mg\sin\phi\cos\theta \\ mg\cos\phi\cos\theta \end{bmatrix} \end{aligned} \tag{2.7}$$

Let us now rewrite F_b and M_b as

$$\begin{bmatrix} F_b \\ M_b \end{bmatrix} = \begin{bmatrix} -mg\sin\theta \\ mg\sin\phi\cos\theta \\ mg\cos\phi\cos\theta \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \\ \tau_4 \\ \tau_5 \\ \tau_6 \end{bmatrix}. \tag{2.8}$$

where $[\tau_1, \tau_2, \tau_3]^T$ are the forces acting in the **x**,**y**,**z** axis of the body frame, and $[\tau_4, \tau_5, \tau_6]^T$ are the moments acting in the **x**,**y**,**z** axis of the body frame. Using (2.8) in (2.5), we obtain the Euler Lagrange dynamics for a UAV:

$$\underbrace{\begin{bmatrix} m & 0 & 0 & 0 & 0 & 0 \\ 0 & m & 0 & 0 & 0 & 0 \\ 0 & 0 & m & 0 & 0 & 0 \\ 0 & 0 & 0 & I_x & 0 & -I_{xz} \\ 0 & 0 & 0 & 0 & I_y & 0 \\ 0 & 0 & 0 & -I_{xz} & 0 & I_z \end{bmatrix}}_{D} \underbrace{\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \\ \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix}}_{\ddot{q}} + \underbrace{\begin{bmatrix} 0 & -mr & mq & 0 & 0 & 0 \\ mr & 0 & -mp & 0 & 0 & 0 \\ -mq & mp & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & I_z r - I_{xz} p & -I_y q \\ 0 & 0 & 0 & -I_z r + I_{xz} p & 0 & I_x p - I_{xz} r \\ 0 & 0 & 0 & I_y q & -I_x p + I_{xz} r & 0 \end{bmatrix}}_{C(\dot{q})} \underbrace{\begin{bmatrix} u \\ v \\ w \\ p \\ q \\ r \end{bmatrix}}_{\dot{q}} + \underbrace{\begin{bmatrix} \sin\theta mg \\ -\sin\phi\cos\theta mg \\ -\cos\phi\cos\theta mg \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{g} = \underbrace{\begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \\ \tau_4 \\ \tau_5 \\ \tau_6 \end{bmatrix}}_{\tau} \tag{2.9}$$

or, in state-space representation,

$$\underbrace{\begin{bmatrix} \dot{q} \\ \ddot{q} \end{bmatrix}}_{\dot{x}} = \underbrace{\begin{bmatrix} 0 & \\ 0 & -D^{-1}C \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} q \\ \dot{q} \end{bmatrix}}_{x} + \begin{bmatrix} 0 \\ -D^{-1}g \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ D^{-1} \end{bmatrix}}_{B} \tau \tag{2.10}$$

It is now convenient to rewrite the dynamics to highlight the role of the propulsion and aerodynamics terms. To this purpose, consider the following 12-state model:

$$\begin{aligned} \begin{bmatrix} \dot{p}_n \\ \dot{p}_e \\ \dot{p}_d \end{bmatrix} &= R_v^{b^{-1}}(\phi,\theta,\psi) \begin{bmatrix} u \\ v \\ w \end{bmatrix}, \\ \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} &= \begin{bmatrix} rv - qw \\ pw - ru \\ qu - pv \end{bmatrix} + \frac{1}{m} \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} \\ \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} &= \begin{bmatrix} 1 & \sin(\phi)\tan(\theta) & \cos(\phi)\tan(\theta) \\ 0 & \cos(\phi) & \sin(\phi) \\ 0 & \sin(\phi)/\cos(\theta) & \cos(\psi)/\cos(\theta) \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \\ \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} &= \begin{bmatrix} \Gamma_1 pq - \Gamma_2 qr \\ \Gamma_5 pr - \Gamma_6 (p^2 - r^2) \\ \Gamma_7 pq - \Gamma_1 qr \end{bmatrix} + \begin{bmatrix} \Gamma_3 L + \Gamma_4 N \\ \frac{1}{J_y} M \\ \Gamma_4 L + \Gamma_8 N \end{bmatrix} \end{aligned} \tag{2.11}$$

where the twelve state variables used to derive the equations of motion are summarized in Table 2.1. Let us denote with R_{vb} the rotation matrix from the vehicle to the body frame, located at the center of mass. The body frame unit vector ($\mathbf{i}^b$, $\mathbf{j}^b$, $\mathbf{k}^b$) is defined such that $\mathbf{i}^b$ points out the nose of the airframe, $\mathbf{j}^b$ points out the right wing, and $\mathbf{k}^b$ points through the bottom of the airframe, which are sometimes referred to as the body **x**-axis, the body **y**-axis, and the body **z**-axis, respectively. In (2.11) terms f_x, f_y, and f_z are the forces acting on $\mathbf{i}^b$, $\mathbf{j}^b$ and $\mathbf{k}^b$, respectively (comprising propulsion, aerodynamic, and gravity forces), whereas L, M, N are the rolling, pitching and yawing moments about the same axes, and

$$
\begin{aligned}
\Gamma_1 &= \frac{I_{xz}(I_x - I_y + I_z)}{I_x I_z - I_{xz}^2}, & \Gamma_2 &= \frac{I_z(I_z - I_y) + I_{xz}^2}{I_x I_z - I_{xz}^2} \\
\Gamma_3 &= \frac{I_z}{I_x I_z - I_{xz}^2}, & \Gamma_4 &= \frac{I_{xz}}{I_x I_z - I_{xz}^2}, \\
\Gamma_5 &= \frac{I_z - I_x}{I_y}, & \Gamma_6 &= \frac{I_{xy}}{I_y}, \\
\Gamma_7 &= \frac{(I_x - I_y)I_x + I_{xy}^2}{I_x I_z - I_{xz}^2}, & \Gamma_8 &= \frac{I_x}{I_x I_z - I_{xz}^2},
\end{aligned}
$$

where the I-terms are components of the inertia tensor.

Notice that the Euler angle representation (2.11) is adopted in this thesis (instead of alternatives like quaternions) due to simplicity of analysis. It is acknowledged that a singularity exists in the Euler angle representation when the pitch angle satisfies $\theta = \pm 90°$. Since, $\cos\theta = 0$ when $\theta = 90°$, so that the roll and yaw angles are indistinguishable. However, for the maneuvers considered in this thesis, such singularity will never occur in practice.

Table 2.1: UAV states (F^i is the inertial frame according to the North-East-Down (NED) convention, F^b is the body frame, F^v, F^{v1}, F^{v2} are the vehicle frame and the intermediate vehicle frames arising from the ($\mathbf{k}^i$-$\mathbf{j}^{v1}$-$\mathbf{i}^{v2}$) Euler rotations).

State	Description
p_n	Inertial north position along $\mathbf{i}^i$ in F^i
p_e	Inertial east position along $\mathbf{j}^i$ in F^i
p_d	Inertial down position along $\mathbf{k}^i$ in F^i
u	Body frame velocity along $\mathbf{i}^b$ in F^b
v	Body frame velocity along $\mathbf{j}^b$ in F^b
w	Body frame velocity along $\mathbf{k}^b$ in F^b
ϕ	Roll angle defined with respect to F^{v2}
θ	Pitch angle defined with respect to F^{v1}
ψ	Yaw angle defined with respect to F^v
p	Roll rate along $\mathbf{i}^b$ in F^b
q	Pitch rate along $\mathbf{j}^b$ in F^b
r	Yaw rate along $\mathbf{k}^b$ in F^b

2.2. PATH MANAGER LAYER

Standard aerial missions for UAVs can be of three types: the path manager layer contains some strategies that combine straight-line paths and orbits paths as primitive paths used to synthesize series of waypoints for the UAV [16].

- *Straight-Line Mission:* The UAV needs to follow a straight-line, which is defined by a slope and a point through which the straight-line passes.

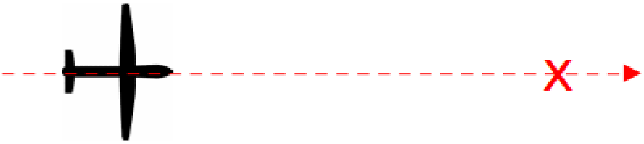


Figure 2.2: Straight-line mission.

- *Loitering Mission:* For this mission, the UAV needs to loiter in an orbit defined by a centre point and an orbit radius.

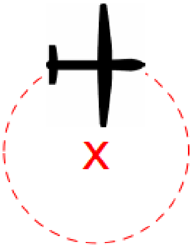


Figure 2.3: Loitering mission.

- *Way Points Mission:* The UAV needs to traverse through (or to fly close-by) a given set of points. The points that define the path are called way points.

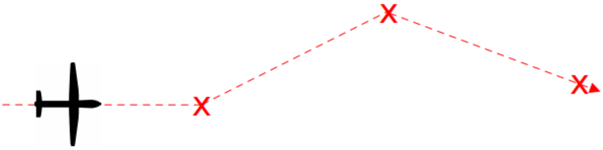


Figure 2.4: Way points mission.

Straight-Line and Loitering Missions are basic paths, which are often referred to as primitives. One typical task of the path manager is to split the way points mission into sub-missions composed of straight-lines and loitering primitives. A simple logic behind this splitting is explained with the help of the example of Figure 2.5. The red cross marks ($\mathbf{W_1}, \mathbf{W_2}, \mathbf{W_3}$) are the way points for the mission. Also, let the turn radius of the mission be R. The mission is split into submissions as:

- *Sub-mission 1 - Straight-Line:* The mission is a straight-line through current location and the point $\mathbf{M_1}$. Let the current position of UAV be $\mathbf{M_L}$; then in vector sense,

$$\mathbf{M_1} = \mathbf{W_1} + \frac{R}{\tan\left(\cos^{-1}\left(\frac{(L-\mathbf{W_1})}{|L-\mathbf{W_1}|}\cdot\frac{L-\mathbf{W_1}+\mathbf{W_2}-\mathbf{W_1}}{|L-\mathbf{W_1}+\mathbf{W_2}-\mathbf{W_1}|}\right)\right)}\frac{(L-\mathbf{W_1})}{|L-\mathbf{W_1}|}. \tag{2.12}$$

- *Sub-mission 2 - Loitering:* A circular orbit is considered centered at $\mathbf{M_2}$ with radius R. In vector sense, $\mathbf{M_2}$ can be written as

$$\mathbf{M_2} = \mathbf{W_1} + \frac{R}{\sin\left(\cos^{-1}\left(\frac{(L-\mathbf{W_2})}{|L-\mathbf{W_2}|}\cdot\frac{L-\mathbf{W_1}+\mathbf{W_2}-\mathbf{W_1}}{|L-\mathbf{W_1}+\mathbf{W_2}-\mathbf{W_1}|}\right)\right)}\frac{L-\mathbf{W_1}+\mathbf{W_2}-\mathbf{W_1}}{|L-\mathbf{W_1}+\mathbf{W_2}-\mathbf{W_1}|}. \tag{2.13}$$

- *Sub-mission 3 - Straight-Line:* The mission is a straight-line through the points $\mathbf{W_1}$ and $\mathbf{M_3}$. In vector sense, $\mathbf{M_3}$ can be written as

$$\mathbf{M_3} = \mathbf{W_2} + \frac{R}{\tan\left(\cos^{-1}\left(\frac{(\mathbf{W_1}-\mathbf{W_2})}{|\mathbf{W_1}-\mathbf{W_2}|}\cdot\frac{\mathbf{W_1}-\mathbf{W_2}+\mathbf{W_3}-\mathbf{W_2}}{|\mathbf{W_1}-\mathbf{W_2}+\mathbf{W_3}-\mathbf{W_2}|}\right)\right)}\frac{(\mathbf{W_1}-\mathbf{W_2})}{|\mathbf{W_1}-\mathbf{W_2}|}. \tag{2.14}$$

- *Sub-mission 4 - Loitering:* A circular orbit centered at $\mathbf{M_4}$ with radius R. In vector sense, $\mathbf{M_4}$ can be written as

$$\mathbf{M_4} = \mathbf{W_2} + \frac{R}{\sin\left(\cos^{-1}\left(\frac{(\mathbf{W_1}-\mathbf{W_2})}{|\mathbf{W_1}-\mathbf{W_2}|}\cdot\frac{\mathbf{W_1}-\mathbf{W_2}+\mathbf{W_3}-\mathbf{W_2}}{|\mathbf{W_1}-\mathbf{W_2}+\mathbf{W_3}-\mathbf{W_2}|}\right)\right)}\frac{\mathbf{W_1}-\mathbf{W_2}+\mathbf{W_3}-\mathbf{W_2}}{|\mathbf{W_1}-\mathbf{W_2}+\mathbf{W_3}-\mathbf{W_2}|}. \tag{2.15}$$

- *Sub-mission 5 - Straight-Line:* The mission is a straight-line through the points $\mathbf{W_2}$ and $\mathbf{W_3}$.

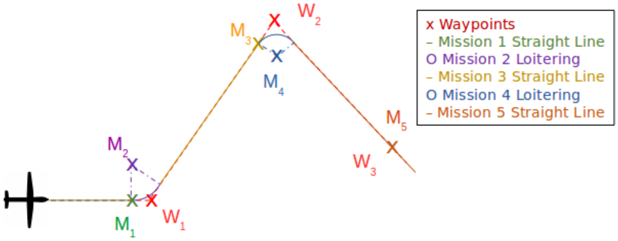


Figure 2.5: Way points mission splitting to sub-missions of straight-line and loitering.

A typical algorithm to plan straight-line and loitering missions is via the Vector Field (VF) approach, which is the topic of the next section.

2.3. PATH FOLLOWING LAYER

Several guidance/path-following techniques have been proposed in the literature, which can be categorized as geometric approaches and control-theoretic approaches [6]. The first class includes the pure pursuit and line-of-sight guidance laws [20–24], which make use of a virtual target point where the UAV is directed to. Control-theoretic techniques include PID, linear quadratic control, sliding-mode control, model predictive control, adaptive control, and their variants [8,25–28]. The *vector field* (VF) approach, originally proposed in [7], has become popular for UAVs and many other unmanned vehicles due to its intuitive combination of geometry and control. The method is based on the generation of a field of desired inertial-referenced course angles for each point around the desired path [29–33]. The goal of the VF approach is to drive the cross-track error (the distance between the current aircraft position and the desired trajectory) to zero using the course angle χ as the control variable. For this reason, it is necessary to provide an appropriate commanded course angle χ_c resulting in the UAV to move towards the path. The course angle χ is the most convenient control variable for this objective, since it is inertial referenced. Considering each point around the desired path, the set of desired course angles is called vector field because it constitutes a set of vectors (relative to the path) of course unit vectors.

2.3.1. THE WIND TRIANGLE

Before introducing the VF guidance control method, let us quickly recall the notion of wind triangle, which is a simple way to describe how the wind affects the aircraft dynamics. Let $\mathbf{V}_g$ be the UAV ground speed relative to the inertial frame and let $\mathbf{V}_a$ be the UAV airspeed. Then, the airspeed $\mathbf{V}_a$, ground speed $\mathbf{V}_g$, and wind speed $\mathbf{V}_w$ are related via the so-called wind triangle (cf. Figure 2.6)

$$\mathbf{V}_a = \mathbf{V}_g - \mathbf{V}_w. \tag{2.16}$$

The wind speed $\mathbf{V}_w$ can be modelled as the composition of a steady-state and a dynamic part

$$\mathbf{V}_w = \mathbf{V}_{w,s} + \mathbf{V}_{w,d}. \tag{2.17}$$

where the dynamic part $\mathbf{V}_{w,d}$ represents wind turbulence, often modeled by passing white noise through appropriate forming filters [34].

By expressing (2.16) in body frame:

$$V_a^b = \begin{bmatrix} u_r \\ v_r \\ w_r \end{bmatrix} = R_w^b \begin{bmatrix} V_a \\ 0 \\ 0 \end{bmatrix} = V_a \begin{bmatrix} \cos\alpha\cos\beta \\ \sin\beta \\ \sin\alpha\cos\beta \end{bmatrix} \tag{2.18}$$

and solving for V_a, α, and β, one obtains:

$$V_a = \left\| V_a^b \right\| = \sqrt{u_r^2 + v_r^2 + w_r^2} \tag{2.19}$$

$$\alpha = \tan^{-1}\left(\frac{w_r}{u_r}\right) \quad \beta = \sin^{-1}\left(\frac{v_r}{\sqrt{u_r^2 + v_r^2 + w_r^2}}\right). \tag{2.20}$$

where u_r, v_r and w_r are relative wind speed projected onto the body frame **x**-axis, **y**-axis and **z**-axis respectively.

Equations (2.19), and (2.20) will be essential when formulating the equations of motion for the UAV. In Figure 2.6, the angle between the wind vector and $\mathbf{i}^i$ is denoted with ψ_w. With the wind triangle in mind, the course angle χ represents the angle between the true north and the projection of $\mathbf{V_g}$ on the horizontal plane ($\mathbf{i}^b$,$\mathbf{j}^b$). The course angle χ constitutes the control variable for the guidance.

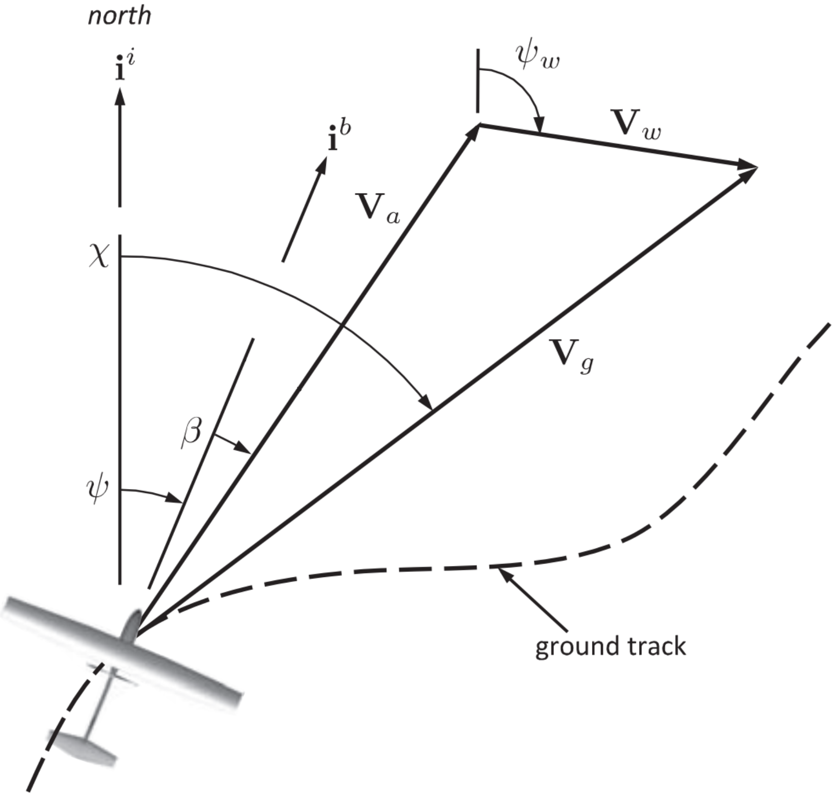


Figure 2.6: The wind triangle (figure adapted from [16]). Here, β represents the side-slip angle of the UAV, χ the course angle, ψ the yaw angle, and ψ_w the wind angle.

2.3.2. STANDARD VECTOR FIELD GUIDANCE LAW

Since the course angle is the most important variable for guidance, any UAV must be equipped with a course-hold loop mechanism, especially in scenarios that request path tracking with high accuracy. The dynamics of the course-hold loop mechanism are typically approximated in the literature (refer to the book [16, Chaps. 9 & 10] or to [7, 35–37]) as a first-order system:

$$\dot{\chi} = \alpha(\chi^c - \chi). \tag{2.21}$$

where χ is the course of the UAV, χ^c is the commanded course from the controller, and α is a known positive constant that defines the response speed of the course-hold loop. It is clear that the value of such a time constant and the validity of the first-order approximation (2.21) depend on how the underlying low-level control law has been tuned.

The vector field is based on specifying a desired course at a certain coordinate, to guide the UAV towards some paths. Two primitive paths are typically considered in the literature:

the straight-line and the orbit path, with fields of desired courses shown in Figure 2.7. More complex paths can be generated as a combination of lines and orbits.

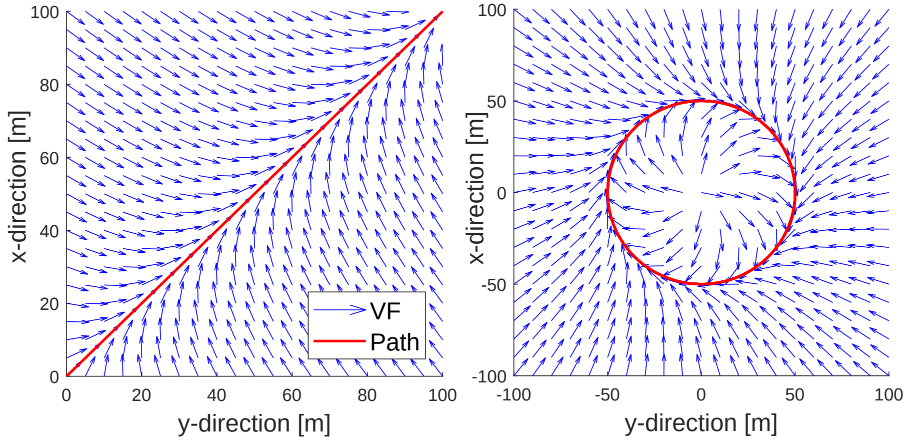


Figure 2.7: Vector fields for straight-line and orbit paths.

STRAIGHT-LINE GUIDANCE

As in [7], let us consider without loss of generality a straight-line parallel to the x-axis. The VF that describes the reference course to drive the UAV on the line is

$$\chi^d(e_{py}) = -\chi_\infty \frac{2}{\pi} \tan^{-1}(k e_{py})) \tag{2.22}$$

where e_{py} is the cross-track error (distance in y-direction), $\chi_\infty \in (0, \frac{\pi}{2}]$, which is the course reference when the error is large, and k is a tuning parameter governing the smoothness of the vector field. The idea of (2.22) is that, when the cross-track error is large, the UAV is supposed to fly almost perpendicularly to the desired line; as the cross-track error decreases, the reference course becomes more and more parallel to the desired line, where the transition is regulated by the parameter k in the function $\tan^{-1}$. The cross-track error e_{py} is obtained according to Figure 2.8 as

$$\mathbf{e}_p = \begin{bmatrix} e_{px} \\ e_{py} \end{bmatrix} = R_i^p (\mathbf{p}^i - \mathbf{r}^i) \tag{2.23}$$

where R_i^p is the rotation matrix from the inertial frame to the path frame, whose $\mathbf{i}$ axis is aligned with the line.

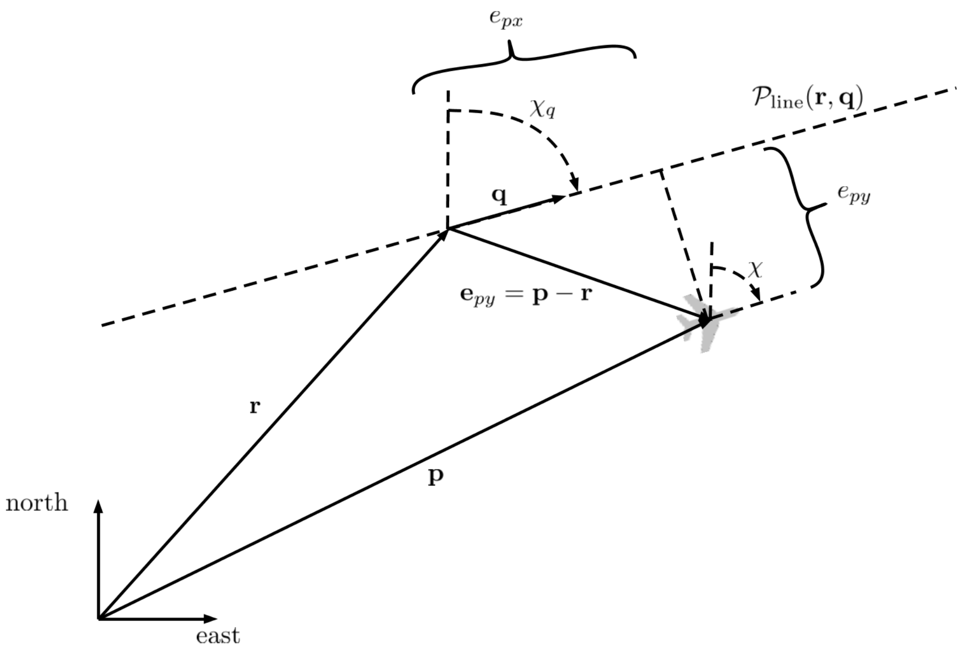


Figure 2.8: Quantities of interest for straight-line path following.

If the straight-line is not parallel to the x-axis as in Figure 2.7, it suffices to use the rotation matrix from the inertial to the path frame. In [7] it is shown that the control law that is able to let $\chi \to \chi^d$ and $e_{py} \to 0$ as $t \to \infty$ is

$$\chi^c = \chi - \chi_\infty \frac{2}{\pi} \frac{\beta_s V_g}{\alpha} \sin(\chi) - \frac{\kappa}{\alpha} \operatorname{sat}\left(\frac{\tilde{\chi}}{\varepsilon}\right) \tag{2.24}$$

where $\tilde{\chi} = \chi - \chi^d$, $\beta_s = k/(1+(ke_{py})^2)$, $V_g = \|\mathbf{V}_g\|$, κ and ε are two parameters governing the control aggressiveness and counteracting a possible chattering in the control action, and

$$\operatorname{sat}(x) = \begin{cases} x & \text{if } |x| < 1, \\ \operatorname{sign}(x) & \text{otherwise.} \end{cases} \tag{2.25}$$

The saturation function acts as a continuous approximation of a sign function, which is typically used in sliding mode control theory. It is well known that the sign function may lead to chattering in the control input [38]. Hence, the continuous saturation function is used in [7] to approximate the behavior of a sign function and avoid to discontinuity in the closed-loop solutions.

ORBIT GUIDANCE

The strategy for orbit guidance builds the course VF around the desired orbit as follows (cf. Figure 2.7):

$$\chi^d(\tilde{d}) = \gamma + \lambda\left(\frac{\pi}{2} + \tan^{-1}(k\tilde{d})\right) \tag{2.26}$$

where $\tilde{d} = d - R$, d is the UAV distance from the orbit center, R is the orbit radius, and γ is the angle between the north and the UAV position with respect to the orbit center. For easiness of analysis, the UAV position is expressed in polar coordinates with $\lambda = 1$ for a clockwise orbit path and -1 for counter-clockwise orbit path. As it can be seen from Figure 2.7, the main intuition of the reference course (2.26) is not very different from the straight-line case: when the cross-track error is large, the UAV is supposed to fly almost perpendicularly to the desired orbit; as the cross-track error decreases, the reference course becomes more and more tangent to the desired orbit, where the transition is regulated by the parameter k in the function $\tan^{-1}$. In [7] it is shown that the control law that is able to let $\chi \to \chi^d$ and $\tilde{d} \to 0$ as $t \to \infty$ is

$$\chi^c = \chi + \frac{V_g}{\alpha d}\sin(\chi - \gamma) + \beta_o \frac{\lambda V_g}{\alpha}\cos(\chi - \gamma) - \frac{\kappa}{\alpha}\,\mathrm{sat}\left(\frac{\tilde{\chi}}{\varepsilon}\right) \tag{2.27}$$

where $\beta_o = k/(1 + (k\tilde{d})^2)$, and the parameters k, κ, ε have a similar meaning as in the straight-line case. The proof of the Lyapunov stability for (2.24) and (2.27) is given in [7] and will not be discussed here. One crucial observation on (2.24) and (2.27) follows.

Remark 2.3.1 *The guidance laws* (2.24) *and* (2.27) *require knowledge of the course time constant* α*, and of the groundspeed* V_g*. Moreover, the groundspeed requires knowledge of the wind. No guidance law has been proposed in the VF literature [29, 35, 39] in the absence of such prior knowledge.*

2.4. AUTOPILOT LAYER

In control of fixed-wing UAVs, one should distinguish at least two levels: the low-level or attitude control law, and the high-level control or guidance/path follower [16]. At the low level, most strategies rely on cascade proportional-integral-derivative (PID) controllers, sometimes enhanced with techniques such as gain scheduling [40]: low-level strategies are usually coded inside the autopilot layer (popular autopilot packages on the market are ArduPilot, Pixhawk, DJI, NAVIO2, AscTec Trinity, just to name a few), that uses measured or observed UAV states (observed position, velocity, specific force, attitude, gyro bias, etc. [41–44]) to control the flying surfaces (aileron, rudder, elevator) and the thrust.

Because the purpose of the autopilot layer is to provide low-level controllers to govern the various UAV states, let us illustrate the main ideas behind lateral and longitudinal UAV control. For the lateral dynamics, the variables of interest are the roll angle ϕ, the roll rate p, the heading angle ψ, and the yaw rate r. The control surfaces used to influence the lateral dynamics are the ailerons δ_a and the rudder δ_r. Ailerons primarily influence the roll rate p; additionally, both the ailerons and the rudder influence the yaw angle ψ.

Similarly, the variables of interest for the longitudinal dynamics are the pitch angle θ, the pitch rate q, the altitude h, and the airspeed V_a. The control signals used to influence the longitudinal dynamics are the elevator δ_e and the throttle δ_t. The elevator is used to directly influence the pitch angle θ. In turn, the pitch angle can be used to manipulate both the altitude and the airspeed. Vice versa, the throttle influences the airspeed and the altitude. Therefore, there are some coupling effects, which will be discussed in more detail in Chapter 3.2.1.

For most flight maneuvers of interest, autopilots are designed with the assumption of decoupled and linear lateral and longitudinal dynamics [16]. In this way, the autopilot design

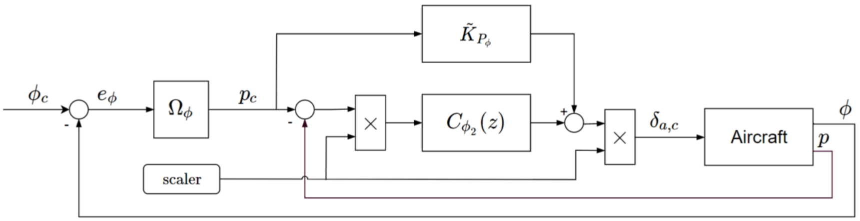


Figure 2.9: Roll control scheme of the UAV. The variables ϕ, ϕ_c are the roll and commanded roll angle, while the variables p, p_c are the roll rate and commanded roll rate. The commanded aileron signal is $\delta_{a,c}$.

significantly simplifies. Using the transfer function formalism, the decoupled linearized dynamics of the UAV are of first and second order:

$$\begin{aligned}
&\text{roll} && \phi(s) = \frac{a_{\phi 2}}{s(s+a_{\phi_1})}\left(\delta_a(s) + \frac{1}{a_{\phi 2}} d_{\phi_2}(s)\right) \\
&\text{pitch} && \theta(s) = \frac{a_{\theta 3}}{s^2 + a_{\theta_2} s + a_{\theta_1}}\left(\delta_e(s) + \frac{1}{a_{\theta 3}} d_{\theta_2}(s)\right) \\
&\text{course} && \chi(s) = \frac{g}{V_g s}\left(\phi(s) + d_\chi(s)\right)
\end{aligned}$$

where s represents the Laplace operator, the terms in d are disturbances coming from the coupled dynamics, the a_{ϕ_*} and a_{θ_*} are constants of the transfer function associated with roll and pitch dynamics, the d_{ϕ_*}, $d_{\theta *}$, and d_{χ_*} are the disturbance signals associated with unmodelled roll, pitch, and course dynamics. All definition for the above variables can be found in [16]. Such first or second order loops allow an effective use of PID control.

Let us focus only on the lateral dynamics, most relevant to path following: the roll controller structure is depicted in Figure 2.9. It consists of two nested loops: the inner one controls the roll rate p; the outer one controls the roll angle ϕ; $C_{\phi_2}(z)$ is a discrete-time PID controller; $\tilde{K}_{P_\phi}$ is a feed-forward gain; in the outer loop there is a proportional controller with gain Ω_ϕ. A similar reasoning applies to the pitch control scheme as shown in Figure 2.10. Details of typical methods to design the low-level controller and on the tuning of the PID controllers can be found in [16, Chap. 6] and on the websites of some open-source autopilot software suites (e.g. PX4, Ardupilot documentation, etc.). Validation of both the roll and the pitch control loops has been performed in the literature (cf. the detailed validation procedure in [16]).

2.5. SIMULATION TOOLS

In order to design realistic path-following tests, we have developed and tested the methods in a software-in-the-loop (SITL) simulator and hardware-in-the-loop (HITL) simulator, which are a MATLAB-based UAV simulation platform and a PX4-Gazebo based simulator, respectively. Both of them include all UAV and environmental dynamics, as briefly described hereafter.

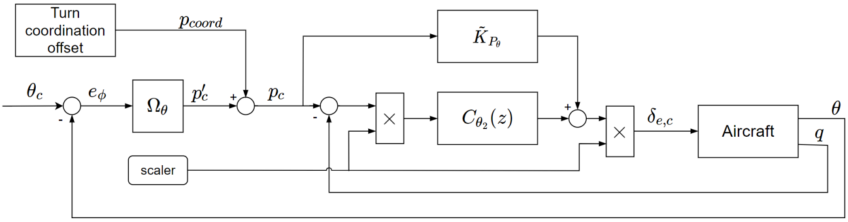


Figure 2.10: Pitch control scheme of the UAV. The variables θ, θ_c are the pitch and commanded pitch angle, while the variables q, q_c are the pitch rate and commanded pitch rate. The commanded elevator signal is $\delta_{e,c}$.

2.5.1. SOFTWARE-IN-THE-LOOP SIMULATION PLATFORM

The fixed-wing UAV and wind dynamics have been implemented in the MATLAB-Simulink environment by means of the Aerospace blockset [45]. With the purpose of testing the algorithms in a realistic UAV simulation platform, a software-in-the-loop UAV platform was developed, i.e. the MATLAB simulator can replicate the low-level control structure of the UAV (i.e. the autopilot layer). A few screenshots of the UAV simulator are in Figure 2.11 and Figure 2.12. In Figure 2.13 the forces and moments contributions are shown on the left. On the right, the block 'Derived Conditions' contains the implementation of the wind dynamics, i.e. the computation of the airspeed, the angle of attack, the side-slip angle, the course angle, and other useful quantities affected by the wind. A visual interface, shown in Figure 2.12, contains in-flight instruments embedded in the simulator, to help analyzing the flight status and to reveal potential errors.

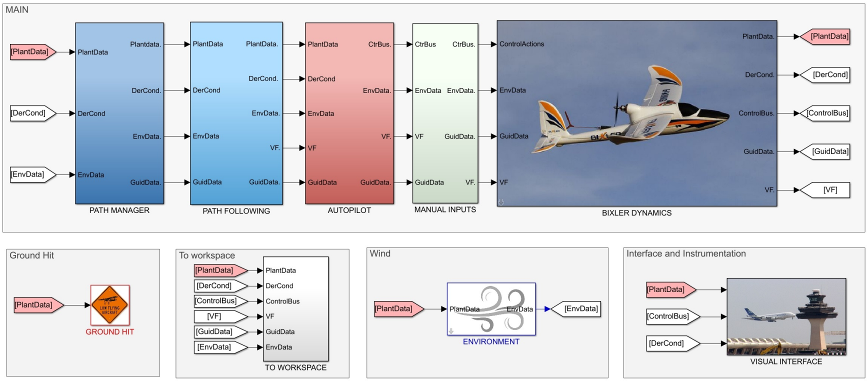


Figure 2.11: UAV simulator (in Matlab-Simulink environment)

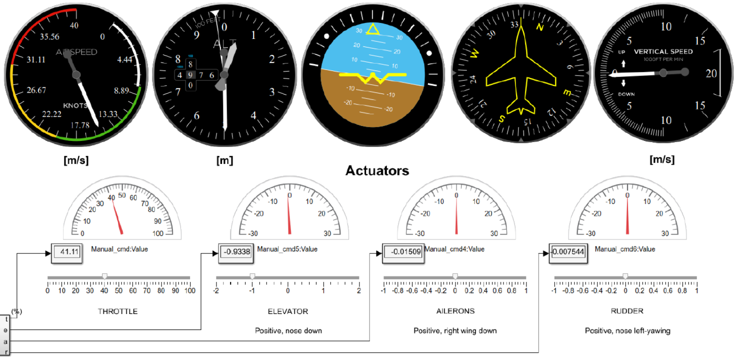


Figure 2.12: Simulink visual interface

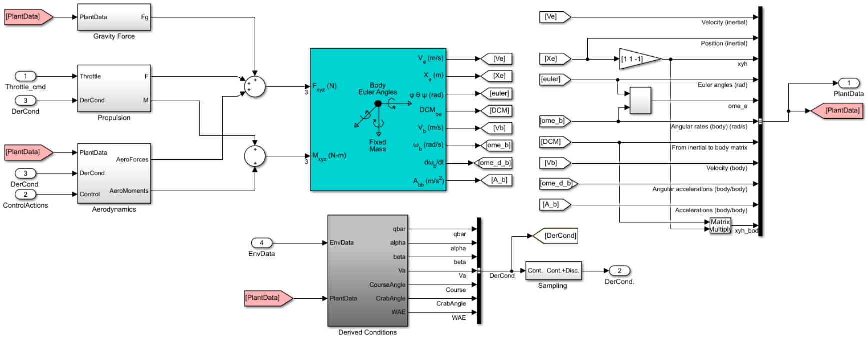


Figure 2.13: Matlab Simulink model for UAV dynamics. The model comprises the forces and moments on the UAV, as well as the airspeed, angle of attack, side-slip angle, and course angle after the effect of the wind.

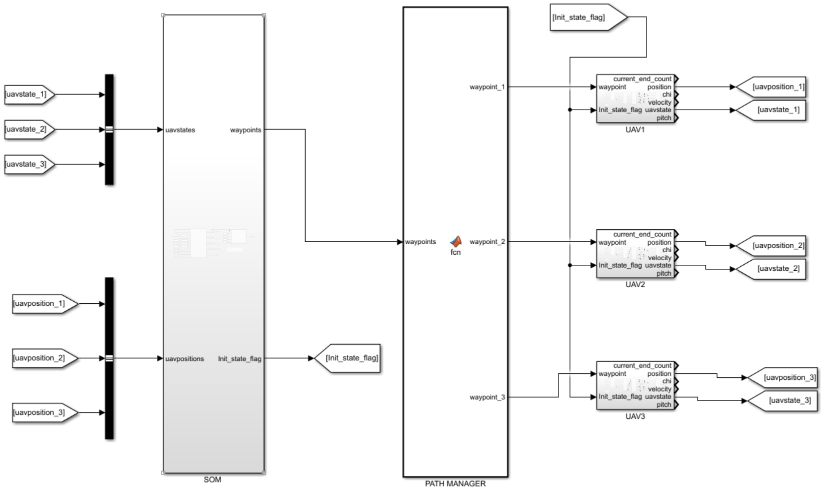


Figure 2.14: Matlab Simulink model for multi-UAVs simulation.

The path manager module is responsible to create complex paths by combining straight-line paths and orbits. The path following tasks for these two simple paths are achieved in the path following module. This module allows to compare several vector field methods, including those proposed in this thesis. The autopilot module contains the code of ArduPilot, a professional autopilot software suite. The code is open-source and it was thus accessed and replicated in Matlab. The module includes all low-level controllers (roll, pitch, altitude, airspeed, side-slip, and course), as well as the Kalman filters used for state estimation. This replication step allowed us to perform simulations with the actual autopilot protocols of the UAV. In BIXLER DYNAMICS we input the geometric characteristics of an actual HobbyKing Bixler UAV airframe; all drag and lift coefficients of the UAV have been derived as look-up tables by means of USAF Digital DATCOM [46]. For this airframe the wind module includes the equations for constant and dynamic wind, so as to generate different environment conditions. More details on a preliminary version of the software-in-the-loop UAV platform can be found in [47].

The simulator also supports additional modules used for multi-UAV simulations, namely a UAV states exchange module. Hence, some UAV formation control algorithms can be tested to achieve multi-UAV control. The number of UAVs that can be tested depends on the computer performance: we have tested 40 UAVs on a laptop. A screenshot of the multi-UAV interface is shown in Figure 2.14.

2.5.2. HARDWARE-IN-THE-LOOP SIMULATION PLATFORM

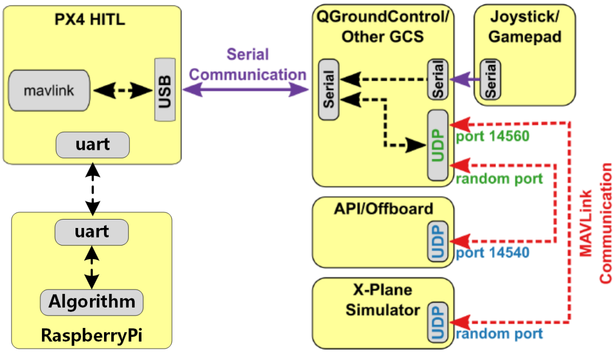


Figure 2.15: Architecture of different modules.

A hardware-in-the-loop UAV platform is set up using the PX4 open-source flight controller with Raspberry Pi 3B+, ROS with a MAVROS communication node (to communicate with PX4), and Gazebo/X-plane as a 3D UAV simulator (cf. Figures 2.15 and 2.16). Along with Ardupilot, PX4 is another popular autopilot suite: its inner-loop dynamics implement a TECS-L1 guidance law, which is an algorithm to control the airspeed (kinetic energy) and the altitude (potential energy) of the UAV. Also, PX4 allows to program in C++ other control laws: in Chapter 3 we programmed both the standard VF and the adaptive VF in the PX4/Raspberry Pi 3B+ hardware.

In this hardware-in-the-loop UAV platform, Gazebo and X-plane can be used not only as 3D simulator for rendering of environments, but also as a physical simulator of the UAV dynamics in 6 degrees of freedom. The UAV model is generated in Gazebo following the tutorial[1]: it is a 1.5 kg standard structure fixed-wing UAV including aileron, rudder, and elevator. The rotor is one puller at the head of the UAV and the airspeed is in the range [10 - 25] m/s. The subsystems are connected as follows: Gazebo or X-plane simulates and visualizes the world environment and the UAV, and it provides the sensor data to PX4; PX4 calculates the guidance commands depending on the embedded algorithm and sends them back to the 3D simulator (Gazebo or X-plane); finally, the 3D simulator delivers the commands to the UAV after simulating the actuator dynamics. As compared to the software-in-the-loop experiments, the hardware-in-the-loop UAV platform is also able to simulate state estimation errors (GPS and IMU measurement errors and the sensor fusion layer), which therefore adds more realism to the experiments.

2.6. REAL-WORLD FIXED-WING UAV AIRFRAME

For real flight testing, we select the MFD crosswind Nimbus Pro fixed-wing UAV (cf. Figure 2.17). Its wingspan is 1950 mm, length is 1287 mm, wing area is 57 dm^2. The weight of the plane is 1.9 kg, and it has a maximum takeoff weight of about 8 kg. The speed range is

---

[1] `http://gazebosim.org/tutorials`

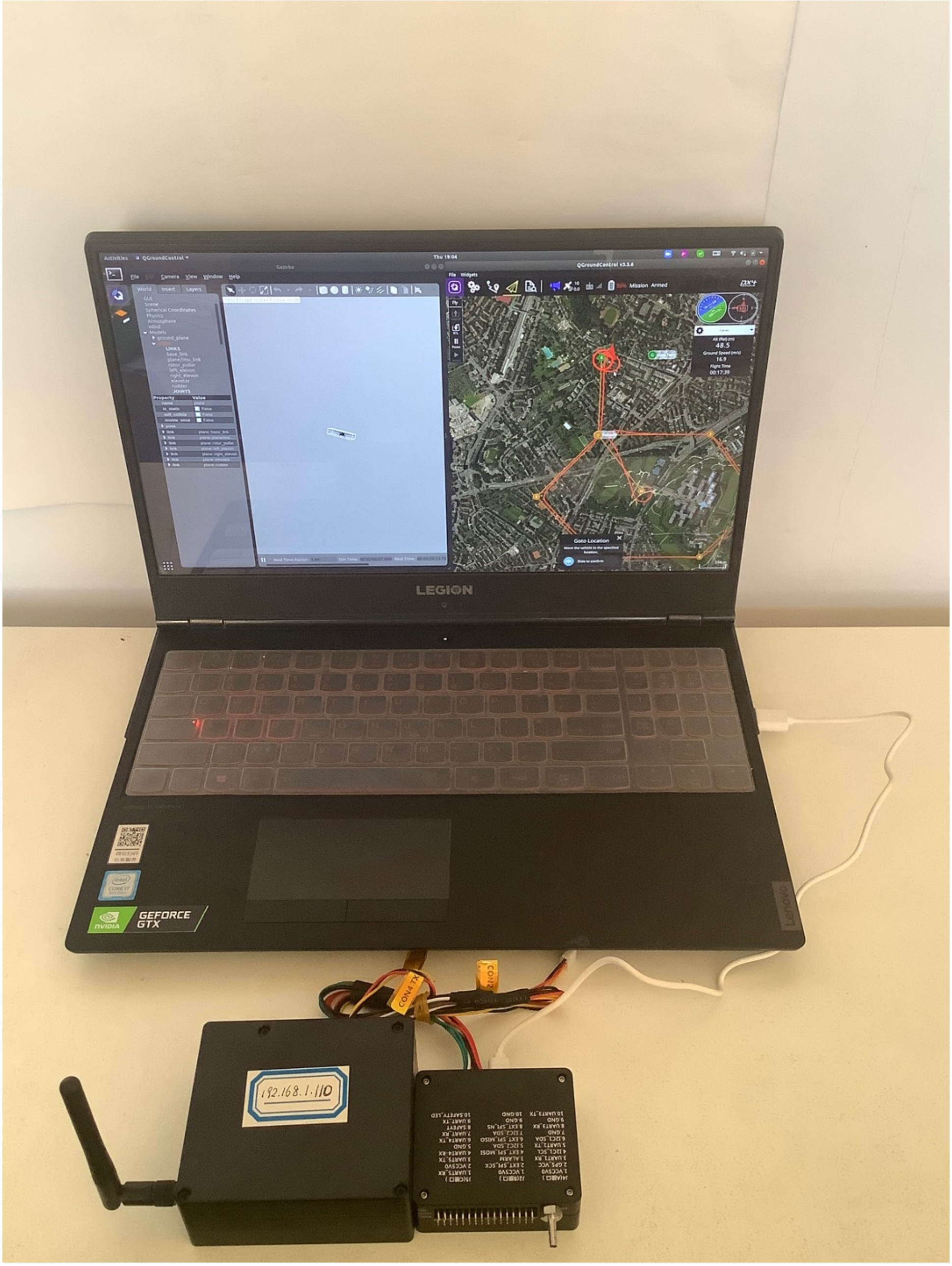


Figure 2.16: Setup for hardware-in-the-loop experiments: a PX4/Raspberry Pi 3B+ controller uses a MAVROS node to share data with a Gazebo 3D simulator

from 12 m/s to 30 m/s. It is a high-performance EPO plane that is easy to assemble in the field and that can take off in a few minutes. With proper setup, it can stay in the air for more than 90 min and travel for more than 100 km.

Figure 2.17: MFD crosswind Nimbus Pro fixed-wing UAV

2.7. SUMMARY

In this chapter, we have recalled the background of the fixed-wing UAV. We have explained how a fixed-wing UAV can be modelled by 6-DOF Euler-Lagrange dynamics. Then we have explained the concepts of path manager layer, the vector field guidance method, and the attitude controller of fixed-wing UAVs layer. A MATLAB simulator and an open-source based simulator have been presented. Finally, the technical specifications of a real flight UAV airframe have been discussed.

3

FIXED-WING UAV VECTOR FIELD GUIDANCE WITH UNCERTAINTIES

3.1. INTRODUCTION

In this chapter, we present several adaptive Vector Field (VF) control laws for fixed-wing UAV path following to compensate for unknown winds speed and direction and/or for unmodelled course angle dynamics. We analyze how unmodelled course angle dynamics arise from the low-level control of the fixed-wing UAV, and we consider two adaptive control laws that suitably handle the uncertainties [48,49]. Furthermore, we provide stability guarantees for the proposed algorithms. Lastly, we simulate our proposed approaches in software-in-the-loop simulator and hardware-in-the-loop simulation environments.

This chapter is structured as follows. In Sect. 3.2, we provide the uncertainties analysis including the wind and UAV course angle dynamics. In Sect. 3.3, we present two adaptive vector field approaches, with stability analysis and simulation evaluation. Lastly, we provide some concluding remarks in Sect. 3.4.

3.2. PROBLEM FORMULATION

Chapter 2 has elaborated that typical guidance laws are designed under the assumptions that some UAV parameters (most notably, roll/pitch/course time constants) are known, course dynamics are linear, longitudinal and lateral motions are not coupled, and the wind vector is known. In particular, we have discussed that the standard VF method crucially relies on two assumptions: known constant wind vector and first-order course dynamics [7]. Both assumptions can be found in all standard VF works [7, 29–33]. In this section, we will explain how unmodelled dynamics naturally arise in practice.

3.2.1. UNCERTAINTY OF UAV COURSE ANGLE DYNAMICS

To the purpose of guidance, the overall UAV dynamics are usually simplified, e.g. by ignoring coupling effects. This simplified setting is partially motivated by the fact that

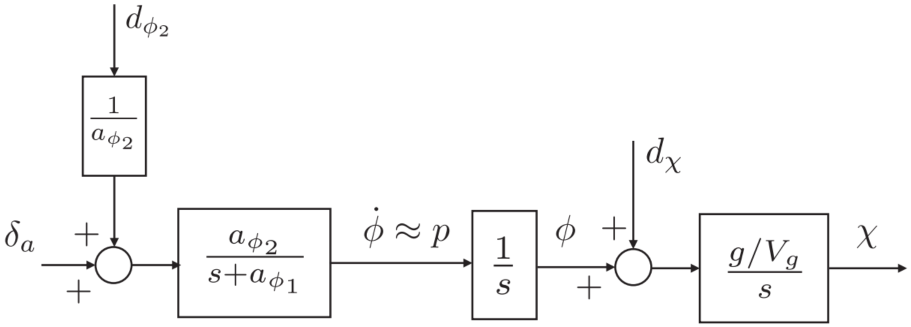


Figure 3.1: Block diagram for simplified course dynamics (adapted from Figure 5.3 in [16])

the control algorithm should reduce the decoupling by disturbance rejection. After this decoupling simplification, the dynamics of the roll angle ϕ can be described by [16, Chap. 5]

$$\dot{\phi} = p + d_{\phi_1} \tag{3.1}$$

where p is the roll rate, and d_{ϕ_1} is an aggregate disturbance:

$$d_{\phi_1} = q \sin\phi \tan\theta + r \cos\phi \tan\theta \tag{3.2}$$

where θ is the pitch angle and r the yaw rate.

After differentiating (3.1), the block diagram in Figure 3.1 can be obtained, showing how the aileron input δ_a and the disturbance d_{ϕ_2} affect the lateral dynamics of the course angle χ:

$$\begin{aligned} d_{\phi_2} \triangleq{} & \dot{d}_{\phi_1} + \Gamma_1 pq - \Gamma_2 qr + \frac{1}{2}\rho {V_a}^2 Sb \cdot \\ & \left[C_{p_0} + C_{p_\beta}\beta - C_{P_p}\frac{b}{2V_a}(d_{\phi_1}) + C_r \frac{br}{2V_a} + C_{p_{\delta_r}}\delta_r \right] \end{aligned} \tag{3.3}$$

where β is the side slip angle, q is the pitch rate, V_a is the airspeed, ρ is the air density, S and b are geometric parameters of the aileron, $\Gamma_{(\cdot)}$ are coefficients related to the inertia matrix of the UAV, and $C_{(\cdot)}$ are coefficients related to the aerodynamics of the UAV. Figure 3.1 and (3.3) clearly show that unmodelled state-dependent terms are aggregated in d_{ϕ_2}, and a similar statement holds for the disturbance d_χ shown in Figure 3.1 (the interested reader can refer to the details in [16, Chap. 6]). These disturbances can take a very complex form and depend on many parameters. However, despite the presence of state-dependent terms, it is common in the literature (refer to [16, Chaps. 9 & 10] or to [6, 7, 35–37]) to assume the disturbance to be *bounded a priori* and the course dynamics to be the *ideal dynamics* (2.21). Two comments follow:

1) The dynamics (2.21) rely on the assumption that longitudinal and lateral dynamics are decoupled. However, [16, Chap. 5.4] discusses how coupling terms arise and should be treated as disturbances. Unfortunately, such disturbances are complex and depend on many parameters including the UAV velocity and lateral commands such

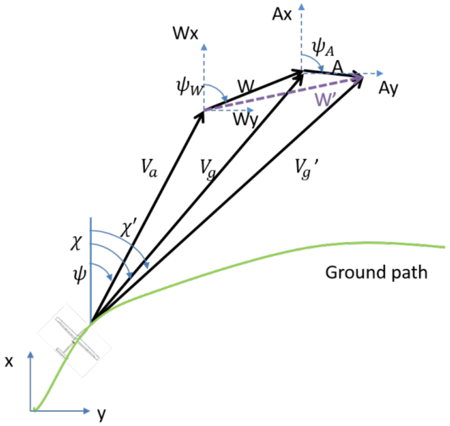


Figure 3.2: The wind triangle for a fixed-wing UAV. Note that calculating the groundspeed V_g or V_g' requires a priori knowledge of the wind.

as rudder angle. Despite this complexity, it is common to treat these disturbances as a priori bounded disturbances affecting the guidance [16, Chap. 10]. In this work, we consider the more realistic course dynamics

$$\dot{\chi} = \alpha(\chi^c - \chi) + \Delta(\chi) \tag{3.4}$$

where $\Delta(\chi)$ is a coupling term acting as state-dependent unmodelled dynamics. A specific description for this unmodelled term will be provided in Sect. 3.3.2 in the framework of adaptive sliding mode control.

2) The steps in [16,37] show how α in (2.21) is *affected in a complex way by aerodynamic coefficients that cannot be perfectly known*, and can even change depending on the altitude and velocity. Therefore, the parameter α in (3.4) should be considered as uncertain or even unknown.

3.2.2. UNCERTAINTY OF WIND DYNAMICS

The relationship between UAV airspeed, ground velocity, and wind velocity is illustrated in Figure 3.2, resulting in the following navigational dynamics of the UAV:

$$\begin{aligned}\dot{x} &= V_a \cos\psi + W\cos\psi_W + A\cos\psi_A = V_g^{'} \cos\chi^{'} \\ \dot{y} &= V_a \sin\psi + W\sin\psi_W + A\sin\psi_A = V_g^{'} \sin\chi^{'}\end{aligned} \tag{3.5}$$

where ψ is the heading angle between airspeed and horizontal axis in the earth frame, V_a is the UAV airspeed, W and A are the amplitude of the constant and the time-varying part of wind, ψ_w and ψ_A are the angle between the constant or time-varying part of the wind velocity and the x axis in the earth frame; x and y are the coordinate of the earth frame. As compared to the standard wind triangle of Figure 2.6, we can see that Figure 3.2 includes possibly time-varying effects, which are typically unknown. As a matter of fact, while the values of W and ψ_w can be obtained from historical data of the wind velocity, the same may not hold for $A(t)$ and $\psi_A(t)$, which should be regarded as disturbances that are changing over time. As a result, one has that V_g is the nominal ground velocity of the UAV and $V_g^{'}$ is the actual but not measurable ground velocity, since the time-varying wind vector influences it.

The overall wind field is denoted with amplitude $W^{'}$ and angle $\psi_{w'}$, which are a combination of constant and time-varying parts:

$$\begin{aligned} W^{'}\cos\psi_{w'} &= W\cos\psi_w + A\cos\psi_A \\ W^{'}\sin\psi_{w'} &= W\sin\psi_w + A\sin\psi_A \end{aligned} \tag{3.6}$$

A third comment follows:

3) The wind introduces another source of uncertainty. The uncertainty in (5.2) is reflected in the fact that *the actual ground speed is not known* since a possibly unknown wind component influences it, as shown in Figure 3.2.

It is worth mentioning that aspects 2) and 3) are overlooked in the standard guidance literature, thus requiring a different guidance approach departing from existing frameworks.

3.3. ADAPTIVE VECTOR FIELD PATH FOLLOWING METHODS

In the following, we will illustrate two adaptive vector field approaches to handle uncertainties. First, we propose a method to handle unknown wind knowledge, which we call adaptive vector field with wind estimator; second, we further improve the first method by handling state-dependent unmodelled terms in the course dynamics, which we call adaptive vector field without a priori knowledge of course dynamics and wind knowledge.

3.3.1. ADAPTIVE VECTOR FIELD WITH WIND ESTIMATOR

Considering the standard vector field in Sect. 2.3 based on the assumption that V_g is not fully known, the idea is to adapt V_g online. In this way, we consider $\hat{V}_g$ in place of V_g, where $\hat{V}_g$ is adapted by an auxiliary differential equation. In the following, we present the estimation mechanism (for straight-line and orbit following), which will be motivated by the corresponding stability analysis.

STRAIGHT-LINE FOLLOWING

The straight-line path following control law (2.24) can be modified as:

$$\chi^c = \chi - \chi_\infty \frac{2}{\pi}\frac{\beta_s \hat{V}_g}{\alpha}\sin(\chi) - \frac{\kappa}{\alpha}\,\mathrm{sat}\left(\frac{\tilde{\chi}}{\varepsilon}\right) \tag{3.7}$$

where the estimation dynamics for a straight-line path is

$$\dot{\hat{V}}_g = \Gamma_{v_{sl}}\mu_{sl}\tilde{\chi}\chi_\infty\beta_s\frac{2}{\pi}\sin(\chi) - \sigma\Gamma_{v_{sl}}\hat{V}_g \tag{3.8}$$

where $\Gamma_{v_{sl}}$ is the estimator gain, μ_{sl} is a weighting term, and σ adds a damping action.

For adaptive control loops with non-parametric uncertainties, the following is a standard notion of stability [38, Def. 4.6]:

Definition 3.3.1 *The solutions of a nonlinear system* $\dot{x} = f(x)$ *are Uniformly Ultimately Bounded (UUB) with ultimate bound* b *if there exist positive constants* b *and* c *and for every* $a \in (0, c)$, *there is a time* $T(a, b)$ *such that*

$$\|x(0)\| \leq a \Rightarrow \|x(t)\| \leq b, \quad \forall t \geq T(a, b). \tag{3.9}$$

Stability and robustness of the proposed adaptive VF are given in the following theorem in terms of uniform ultimately boundedness:

Theorem 3.3.1 *Consider the course angle dynamics (3.4), where* $|\Delta(\chi)| \leq \bar{\Delta}$ *is a disturbance term with possibly unknown upper bound* $\bar{\Delta}$. *Furthermore, assume that the unknown* V_g *is slowly time-varying with* $|V_g| \leq \bar{V}_g$ *and* $|\dot{V}_g| \leq \bar{\bar{V}}_g$, *for some unknown* $\bar{V}_g$ *and* $\bar{\bar{V}}_g$. *Then, the control laws* (3.7) *(with* $\hat{V}_g$ *in place of* V_g*) and the adaptive law* (3.8) *guarantee uniform ultimate boundedness of* $\left\| [e_{py},\ \rho^{1/2}\tilde{\chi},\ \Gamma_{sl}^{-1/2}\Theta]^{\top} \right\|^2$, *with ultimate bound B as in* (3.13).

Orbit path following

The orbit path following control law (2.27) can be modified as:

$$\chi^c = \chi + \frac{\hat{V}_g}{\alpha d}\sin(\chi - \gamma) + \beta_o \frac{\lambda \hat{V}_g}{\alpha}\cos(\chi - \gamma) - \frac{\kappa}{\alpha}\,\mathrm{sat}\left(\frac{\tilde{\chi}}{\varepsilon}\right) \tag{3.10}$$

where the estimation for an orbit path is

$$\dot{\hat{V}}_g = -\Gamma_{v_o}\mu_o\tilde{\chi}\left(\frac{1}{d}\sin(\chi - \gamma) + \lambda\beta_o\cos(\chi - \gamma)\right) - \sigma\Gamma_{v_o}\hat{V}_g \tag{3.11}$$

where Γ_{v_o}, μ_o, and σ have a similar meaning to the straight-line case.

Theorem 3.3.2 *Consider the same settings as in Theorem 3.3.1, the orbit path control law* (3.10) *(with* $\hat{V}_g$ *in place of* V_g*) and the adaptive law* (3.11) *guarantee uniform ultimate boundedness of* $\left\| [e_{py},\ \rho^{1/2}\tilde{\chi},\ \Gamma_o^{-1/2}\Theta]^{\top} \right\|^2$.

Stability analysis

Because the stability proof of the straight-line and the orbit path are similar, we only provide the analysis for the straight-line case.

Proof. The adaptive law for $\hat{V}_g$ is derived based on the Lyapunov argument below. Let $\Theta = \hat{V}_g - V_g$ be the estimation error. Consider the Lyapunov function $W = W_1 + \rho W_2 + \frac{1}{2}\Gamma_{v_{sl}}^{-1}\Theta^2$, with $W_1 = \frac{1}{2}e_{py}^2$, $W_2 = \frac{1}{2}\tilde{\chi}^2$ whose derivative is

$$\dot{W} = \dot{W}_1 + \rho\dot{W}_2 + \Gamma_{v_{sl}}^{-1}\Theta\dot{\Theta}$$

where ρ is the positive user-designed weight. Substitute (2.24) (with $\hat{V}_g$ in place of V_g) into the derivative of W:

$$\dot{W} = \dot{W}_1 + \rho\tilde{\chi}[\chi_\infty\frac{2\beta_s}{\pi}(\hat{V}_g - V_g)(\sin(\chi) - \kappa\,\mathrm{sat}(\frac{\tilde{\chi}}{\varepsilon})] + \Gamma_{v_{sl}}^{-1}(\hat{V}_g - V_g)(\dot{\hat{V}}_g - \dot{V}_g).$$

Now substitute (3.8)

$$\begin{aligned}\dot{W} &= \dot{W}_1 - \rho\kappa\tilde{\chi}\,\mathrm{sat}(\frac{\tilde{\chi}}{\varepsilon}) + \left[(\dot{\hat{V}}_g - \dot{V}_g)\Gamma_{v_{sl}}^{-1} + \rho\tilde{\chi}\chi_\infty \frac{2\beta_s}{\pi}\sin(\chi)\right](\hat{V}_g - V_g) + \rho\tilde{\chi}\Delta \\ &\leq \dot{W}_1 - \rho\tilde{\chi}\,\mathrm{sat}(\frac{\tilde{\chi}}{\varepsilon}) - \sigma\Theta^2 + \rho\left|\tilde{\chi}\right|\bar{\Delta} - \sigma\Theta(-\Gamma_{v_{sl}}^{-1}\dot{V}_g\sigma^{-1} - V_g).\end{aligned}$$

Using the design condition $\kappa \geq \bar{\Delta}$ we can remove the third and fifth terms of the last inequality when $\frac{\tilde{\chi}}{\varepsilon} \geq 1$ (the analysis for $\frac{\tilde{\chi}}{\varepsilon} < 1$ straightforwardly leads to boundedness, cf. [50]). Furthermore, after applying the inequality $-a^2 + ab \leq -\frac{a^2}{2} + \frac{b^2}{2}$ (valid for any a and b) to the fourth and last term above, we write

$$\dot{W} \leq \dot{W}_1 - \frac{\sigma}{2}\Theta^2 + \frac{\sigma(V_g + \dot{V}_g\Gamma_{v_{sl}}^{-1}\sigma^{-1})^2}{2}. \tag{3.12}$$

Since we assume that the wind changes in a slowly time-varying fashion, the magnitude of $\dot{V}_g$ will be bounded. Also, the ground velocity V_g is bounded. Therefore we can bound $\left|V_g + \dot{V}_g\Gamma_{v_{sl}}^{-1}\sigma^{-1}\right| \leq \Delta_V$, for some unknown Δ_V. Then, using the definition of the Lyapunov function W and the analysis of $\dot{W}_1$ carried out in [7], we have that (3.12) implies

$$\dot{W} \leq -\kappa W - (\bar{\kappa} - \kappa)W + C$$

where $0 < \kappa < \bar{\kappa}$, $\bar{\kappa} = \sigma\Gamma_{v_{sl}}$, and $C = \frac{\sigma\bar{\Delta}_V^2}{2}$, i.e. C is proportional to $\bar{\Delta}_V^2$ (upper bound on the combined variation of the wind). After defining the scalar $B = C/(\bar{\kappa} - \kappa)$ we straightforwardly obtain the bound

$$\left\|\left[e_{py},\ \rho^{1/2}\tilde{\chi},\ \Gamma_{v_{sl}}^{-1/2}\Theta\right]^{\mathrm{T}}\right\|^2 \leq \max\{W(0), B\} \tag{3.13}$$

i.e. e_{py}, $\tilde{\chi}$, Θ converge inside a compact set and stay bounded.

ALGORITHM EVALUATION

In this section, the performance of the adaptive VF method is assessed, as compared to the standard VF method of [7] and to an ideal VF method, with the following wind knowledge:

- Standard VF: guidance (2.24), (2.27) with $V_g(t) = ||\mathbf{V}_a(t) + \mathbf{W}||$ (knowledge of constant wind disturbance);
- Ideal VF: guidance (2.24), (2.27) with $V_g(t) = ||\mathbf{V}_a(t) + \mathbf{W} + \mathbf{A}(t)||$ (knowledge of constant and time-varying disturbance);
- Adaptive VF: guidance (2.24), (2.27) with $V_g(t) = \hat{V}_g(t)$ and (3.8), (3.11) with $\hat{V}_g(0) = ||\mathbf{V}_a(0) + \mathbf{W}||$ (lack of wind knowledge replaced by estimator).

The standard VF knows only the constant wind component, the ideal VF knows the constant and dynamic wind components, while the adaptive VF estimates all components starting from the initial knowledge of the constant wind (we defined it as 'ideal VF' because this approach still relies on simplified course dynamics, which may lead to degraded performance).

Table 3.1: Flight environmental conditions

Scenario	Constant wind	Turbulence	Slowly time-varying wind
#1	$\|\|\mathbf{W}\|\| = 0$ m/s	No	No $\mathbf{A}(t)$
#2	$\|\|\mathbf{W}\|\| = 4$ m/s	No	No $\mathbf{A}(t)$
#3	$\|\|\mathbf{W}\|\| = 4$ m/s	Yes	No $\mathbf{A}(t)$
#4	$\|\|\mathbf{W}\|\| = 4$ m/s	Yes	Yes

Four different wind scenarios have been defined, summarized in Table 3.1, so as to draw conclusions on the effectiveness of adaptation in different conditions. For each scenario, we simulate a straight-line and an orbit path. The constant wind direction is 240 deg. A Dryden dynamic wind model [51] is configured following the its specifications (altitude of 50 m, turbulence intensity of 2.15 m/s on the $\mathbf{i}^b$ and $\mathbf{j}^b$ axes, and 1.4 m/s on the $\mathbf{k}^b$ axis, with wavelengths of 200 m). A slowly time-varying wind part is taken as a sinusoid of 0.01 rad/s, perturbing both the wind magnitude and the wind direction with amplitudes 3 m/s and π rad, respectively.

Two experimental sets are performed:

- Using ideal simplified first-order course angle dynamics;
- Using the more realistic UAV model (with high-order course angle dynamics).

The first set has the purpose of testing the algorithms in the ideal scenario, in such a way to better evaluate how the performance degrades in non-ideal scenarios.

The performance of the standard, adaptive, and ideal VF method are evaluated using the root mean square (RMS) steady-state cross-track error, calculated in the last portion of the path when e_{py} or $\tilde{d}$ have converged. The parameters k, κ, ε, $\Gamma_{v_{sl}}$, and Γ_{v_o} summarized in Table 3.2, have been tuned so as to find a good compromise between convergence speed and no oscillations. The values of Γ_{sl} and Γ_o cannot be selected to high in order to avoid the so-called “high-gain" estimation drift. Meanwhile, the value of σ should be small as long as the because performance is already satisfactory. The main reason why Γ_{sl} is larger than Γ_o is that the straight-line path following presents some lack of persistent excitation to the estimator dynamics, thus requiring a larger gain. The scaling parameters μ_{sl} and μ_o are chosen as the ratio between the initial cross-track error and the maximum course error, i.e. $\mu_{sl} = (e_{py}(0)/\pi)^2$, $\mu_o = (\tilde{d}(0)/\pi)^2$.

Table 3.2: Standard and Adaptive VF parameters.

χ_∞	k	κ	ε	$\Gamma_{v_{sl}}$	Γ_{v_o}	σ
$\pi/2$	0.1 m^{-1}	$\pi/2$ $\mathrm{rad}^2/\mathrm{s}$	1 rad	0.5	0.1	0.001

Experiment: Simplified first-order course dynamics

For the first-order course dynamics (2.21), Table 3.3 reports the RMS steady-state errors in all environmental conditions.

The following observations can be drawn from Table 3.3:

Table 3.3: Vector field RMS steady-state errors in meters (first-order course dynamics).

Straight-line path following

Scenario	Standard VF	Adaptive VF	Ideal VF
#1	0.00	0.00	0.00
#2	0.00	0.00	0.00
#3	0.16	0.12	0.00
#4	0.17	0.12	0.00

Orbit path following

Scenario	Standard VF	Adaptive VF	Ideal VF
#1	0.00	0.00	0.00
#2	0.00	0.00	0.00
#3	0.29	0.14	0.00
#4	0.31	0.14	0.00

- With perfect knowledge of the wind and assuming simplified first-order dynamics, the ideal VF achieves a zero steady-state error in all wind conditions;
- In the absence of wind, or with only constant wind (Scenarios #1 and #2) also the standard and the adaptive VF can achieve a zero steady-state error;
- The adaptive VF outperforms the standard VF in Scenarios #3 and #4 (error reduction $> 20\%$ for the straight-line and $> 50\%$ for the orbit), when unmodelled wind components cannot be accounted by the standard VF.

Experiment: Realistic high-order course dynamics

Using the more realistic UAV model, Table 3.4 reports the RMS steady-state errors in all environmental conditions.

Table 3.4: Vector field RMS steady-state errors in meters (high-order course dynamics).

Straight-line path following

Scenario	Standard VF	Adaptive VF	Ideal VF
#1	0.00	0.00	0.00
#2	0.00	0.00	0.00
#3	0.26	0.25	0.26
#4	0.24	0.24	0.20

Orbit path following

Scenario	Standard VF	Adaptive VF	Ideal VF
#1	0.10	0.00	0.10
#2	0.10	0.00	0.10
#3	0.39	0.21	0.31
#4	1.29	0.80	1.09

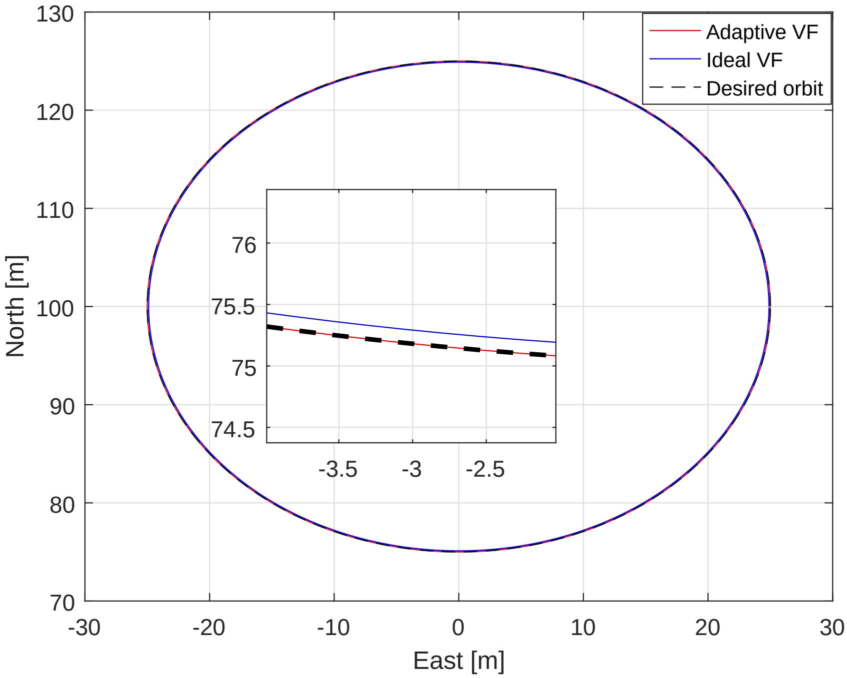


Figure 3.3: Ideal vs Adaptive VF method during orbit following with high-order dynamics (Scenario #1). To highlight the path following error of the ideal VF method, a zoom-in is shown in the small picture.

Table 3.4 demonstrates that the ideal VF has lost its perfect performance shown in Table 3.3. That is, even with full knowledge of the wind, the ideal VF cannot cope with unmodelled dynamics. The following observations are drawn from Table 3.4:

- For the straight-line, the improvement of the adaptive VF is often small. This can be explained by the fact that the unmodelled UAV dynamics are not 'excited' by the straight-line path. The term 'excited' is used in the sense of *persistency of excitation*, a concept well known in adaptive control [4, Sect. 5.2] and referring to the number of sinusoids contained in a signal. The higher the frequency content of a signal flowing across the closed-loop system, the more the unmodelled dynamics of the system will 'manifest' and make the tracking error different from zero, which in turn will activate the adaptive law. Persistency of excitation is reflected in the path-following problem by the fact that the periodic motion induced by the orbit path contains sinusoidal components that are absent in the straight-line case. In fact, the poor excitation of the straight-line path makes all algorithms achieve zero errors in Scenarios #1 and #2 despite the unmodelled dynamics;
- In the orbit scenario, apparently the excitation induced by periodic motion activates the adaptive law and makes the adaptive VF method attain drastic improvements, outperforming not only the standard VF method, but also the ideal VF method.

Remarkably, in Scenario #1 the adaptive VF method achieves a zero tracking error by completely compensating the unmodelled dynamics, as shown in Figure 3.3: on the other hand, the unmodelled dynamics prevent the ideal VF method from perfectly following the orbit (the steady-state error is around 0.11m). Also in Scenario #2 the adaptive VF method drives the error to zero by counteracting the constant wind disturbances, while in Scenarios #3 and #4 the error is reduced by 46% and 61% respectively.

Experiment: On transient performance

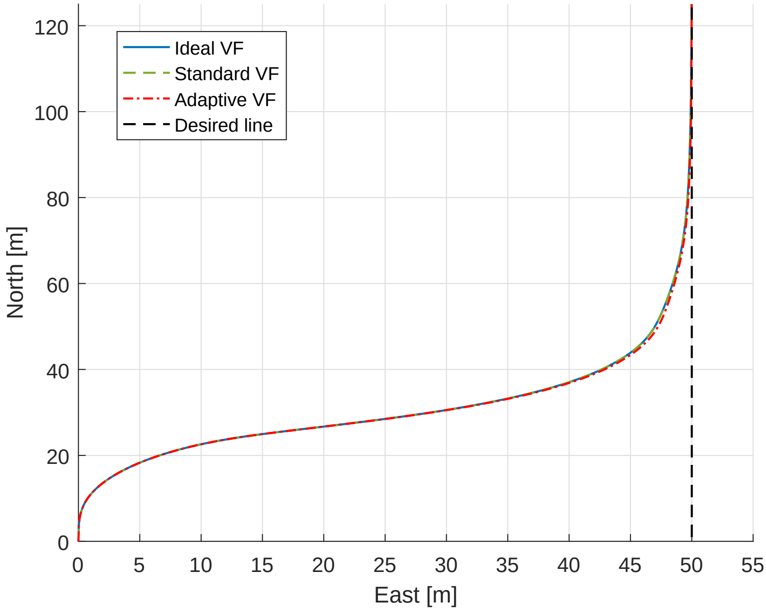


Figure 3.4: Transient of the Ideal, Standard and Adaptive VF method during line following with high-order dynamics (Scenario #2).

To comment on the transient performance of the VF algorithms, Table 3.5 collects the errors calculated from the beginning of the trajectory till when the error is less than 1 meter (the values are higher than the previous tables, because the UAV starts 50 meters away from the desired trajectory). Table 3.5 reveals that the transient performance of the adaptive VF method is close and in most cases slightly better than the non-adaptive versions (cf. Figure 3.4 for straight-line following in Scenario #2). The explanation is twofold: (a) as indicated previously, the initial estimated ground speed in the adaptive VF method results from the vector sum of the airspeed and the constant wind, which is good starting point feasible for implementation (the same a priori knowledge as the standard VF method); (b) the estimator (3.8) and (3.11), being Lyapunov-based, contributes to stability by driving the error to zero. While the transient performance of any adaptive algorithm inevitably benefits from good initial knowledge of the uncertain parameters [4, Sect. 4.3.7], it is worth remarking that, thanks to the estimator (3.8) and (3.11), the a priori knowledge of the standard VF method is *not* requested in the adaptive VF method, giving benefits in unknown wind environments.

Table 3.5: Vector field RMS transient errors in meters (high-order course dynamics).

Straight-line path following

Scenario	Standard VF	Adaptive VF	Ideal VF
#1	23.53	23.52	23.53
#2	26.24	26.20	26.24
#3	26.63	26.58	26.62
#4	27.32	27.08	27.13

Orbit path following

Scenario	Standard VF	Adaptive VF	Ideal VF
#1	31.02	31.01	31.02
#2	32.54	32.53	32.54
#3	33.02	33.00	32.99
#4	33.86	33.85	33.73

3.3.2. ADAPTIVE VECTOR FIELD WITHOUT A PRIORI KNOWLEDGE OF COURSE DYNAMICS AND WIND

The method of Sect. 3.3.1 mainly focuses on unknown wind vector, although it was shown experimentally that the method can also handle unmodelled course dynamics. In this section, we will propose a new guidance law with *no a priori knowledge of the UAV course dynamics and of the wind vector* [52]. The main contributions of this section are:

- Achieving vector-field path following without structural knowledge of the unmodelled coupling effects and without a priori knowledge of the course time constant and of wind amplitude/direction;
- Connect the adaptive vector field method to the uncertainty framework of adaptive sliding mode (cf. [53–56] and references therein), while extending it to consider unmodelled dynamics without a priori constant bound.

The first contribution is made possible by including estimation in the guidance laws, to compensate the uncertain terms. The second contribution is possible by considering a state-dependent uncertainty bound. Similarly to what has been done before, we will first describe the uncertainty setting, then give the adaptive laws with corresponding stability analysis and numerical validation.

UNCERTAINTY SETTING

The disturbances (3.2)-(3.3) reveal that finding a closed-form form for the term $\Delta(\chi)$ is difficult. We follow an approach motivated by the control-theoretic framework of sliding mode control [57, Assumpt. A2, eq. (8)], showing that for a first-order system $\dot{x} =$

$f(x)+u+\Delta(x)$, nonlinear unmodelled dynamics $\Delta(x)$ can be represented as

$$||\Delta(x)|| \leq c_0 + c_1||x||, \tag{3.14}$$

where c_0, c_1 are some constants. To depart from the ideal assumptions in the literature, the following state dependency of uncertainty Δ is considered:

$$\left|\Delta(\chi)\right| \leq c_0 + c_1\left|\tilde{\chi}+\chi^d\right| \leq \kappa_0 + \kappa_1\left|\tilde{\chi}\right| \tag{3.15}$$

for some scalars κ_0, $\kappa_1 \in \mathbb{R}^+$. We have used (3.14) and the fact that χ^d is bounded by definition. Under the assumption that κ_0, κ_1 are known, the modelling approach (3.15) was proposed in the sliding mode control literature (cf. [57, eq. (8)] and related works) as a way to model complex (state-dependent) disturbances. Notice that (3.15) includes the fact that $\Delta(\chi)$ *may not be bounded a priori by a constant.*

Instead of assuming exact knowledge of α, let us consider a nominal course time constant, named $\hat{\alpha}$, and satisfying

$$\left|\frac{\alpha}{\hat{\alpha}}-1\right| = E < 1 \tag{3.16}$$

for some design constant parameter E. It can be noticed that the uncertainty set (3.16) requires $\hat{\alpha}$ to be not far from the actual α, i.e. $\alpha < (1+E)\hat{\alpha}$.

Since $\alpha > 0$, (3.4) can be written as

$$\overline{\alpha}\dot{\chi} = -\chi + \chi^c + \overline{\Delta}, \tag{3.17}$$

where $\overline{\alpha} \triangleq 1/\alpha, \overline{\Delta} \triangleq \Delta/\alpha$. For control design purposes, the derivative of (2.22) is calculated in [7] as

$$\dot{\chi}^d = -\chi_\infty \frac{2}{\pi}\beta_s V_g \sin(\chi). \tag{3.18}$$

As V_g is unknown, $\dot{\chi}^d$ is not available for control design. Then, observing (3.15), (3.17) we have

$$\left|\overline{\Delta}\right| \leq \kappa_0^* + \kappa_1^*\left|\tilde{\chi}\right|, \tag{3.19}$$

where $\kappa_0^* \triangleq \kappa_0/\alpha, \kappa_1^* \triangleq \kappa_1/\alpha$ are unknown positive constants. For ease of controller design, let us also define $\kappa_2^* \triangleq \overline{\alpha}V_g$, which is also an unknown positive constant.

STRAIGHT-LINE FOLLOWING

Based on the uncertainty structure (3.19), a guidance law is proposed as

$$\chi^c = -\Lambda\tilde{\chi} + \chi - \hat{\kappa}_2\chi_\infty\frac{2}{\pi}\beta_s\sin(\chi) - \rho\,\text{sat}\left(\frac{\tilde{\chi}}{\varepsilon}\right), \tag{3.20a}$$

$$\rho = \hat{\kappa}_0 + \hat{\kappa}_1|\tilde{\chi}|, \tag{3.20b}$$

where $\Lambda \in \mathbb{R}^+$ is a user-defined scalar, and $\hat{\kappa}_i$ are the estimates of κ_i^* $i = 0,1,2$, evaluated via the following adaptive laws:

$$\dot{\hat{\kappa}}_0 = |\tilde{\chi}| - \Gamma_0 \hat{\kappa}_0, \tag{3.21a}$$

$$\dot{\hat{\kappa}}_1 = |\tilde{\chi}|^2 - \Gamma_1 \hat{\kappa}_1, \tag{3.21b}$$

$$\dot{\hat{\kappa}}_2 = \chi_\infty \frac{2}{\pi} \beta_s \sin(\chi) \tilde{\chi} - \Gamma_2 \hat{\kappa}_2, \tag{3.21c}$$

$$\text{with } \hat{\kappa}_i(0) > 0,\ i = 0,1,2, \tag{3.21d}$$

where $\Gamma_i \in \mathbb{R}^+$ are user-defined scalars.

The following result can be derived:

Theorem 3.3.3 *By employing the guidance law (3.20), the resulting trajectories of the UAV (3.17) and the parameters in the adaptive law (3.21) are Uniformly Ultimately Bounded (UUB).*

STABILITY ANALYSIS

Before starting the analysis, let us notice that the combination of the adaptive laws (3.21a)-(3.21b), (3.38a)-(3.38b) and the initial conditions (3.21d),(3.38d) imply that

$$\hat{\kappa}_0(t), \hat{\kappa}_1(t) \geq 0, \forall t \geq 0. \tag{3.22}$$

for both the straight-line and orbit path.

The closed-loop stability in the straight-line case is analysed using the following Lyapunov function:

$$W = \frac{1}{2} \overline{\alpha} \tilde{\chi}^2 + \frac{1}{2} \sum_{i=0}^{2} (\hat{\kappa}_i - \kappa^*)^2. \tag{3.23}$$

Define an overall uncertainty term

$$\Delta^c \triangleq \overline{\Delta} + \kappa_2^* \chi_\infty \frac{2}{\pi} \frac{k}{1 + (k e_{py})^2} \sin(\chi). \tag{3.24}$$

Observing the structure of sat(·) as in (3.20a), the overall stability analysis is carried out for the following two cases, using the common Lyapunov function (3.23).

Case (i): $|\tilde{\chi}| \geq \epsilon$

Using (3.17) and (3.20), the time-derivative of (3.23) yields

$$\begin{aligned}
\dot{W} &= \tilde{\chi}(-\chi + \chi^c + \Delta^c) + \sum_{i=0}^{2} (\hat{\kappa}_i - \kappa_i^*) \dot{\hat{\kappa}}_i \\
&\leq -\Lambda \tilde{\chi}^2 - (\hat{\kappa}_0 - \kappa_0^*)|\tilde{\chi}| - (\hat{\kappa}_1 - \kappa_1^*)|\tilde{\chi}|^2 \\
&\quad - (\hat{\kappa}_2 - \kappa_2^*) \chi_\infty \frac{2}{\pi} \beta_s \tilde{\chi} \sin(\chi) + \sum_{i=0}^{2} (\hat{\kappa}_i - \kappa_i^*) \dot{\hat{\kappa}}_i.
\end{aligned} \tag{3.25}$$

From (3.21a)-(3.21c) we have

$$\sum_{j=0}^{1}(\hat{\kappa}_j-\kappa_j^*)\dot{\hat{\kappa}}_j=(\hat{\kappa}_j-\kappa_j^*)|\tilde{\chi}|^{j+1}-\Gamma_j\hat{\kappa}_j^2+\Gamma_j\hat{\kappa}_j\kappa_j^*, \tag{3.26}$$

$$(\hat{\kappa}_2-\kappa_2^*)\dot{\hat{\kappa}}_2=(\hat{\kappa}_2-\kappa_2^*)\chi_\infty\frac{2}{\pi}\beta_s\tilde{\chi}\sin(\chi)-\Gamma_2\hat{\kappa}_2^2+\Gamma_2\hat{\kappa}_2\kappa_2^*. \tag{3.27}$$

The following simplifications can be made for $i=0,1,2$:

$$\begin{aligned}\hat{\kappa}_i\kappa_i^*-\hat{\kappa}_i^2&=-\left(\frac{\hat{\kappa}_i}{\sqrt{2}}-\frac{\kappa_i^*}{\sqrt{2}}\right)^2-\frac{\hat{\kappa}_i^2}{2}+\frac{\kappa_i^{*2}}{2}\\&\le-\left(\frac{\hat{\kappa}_i}{\sqrt{2}}-\frac{\kappa_i^*}{\sqrt{2}}\right)^2+\frac{\kappa_i^{*2}}{2}.\end{aligned} \tag{3.28}$$

Substituting (3.26)-(3.28) into (3.25) yields

$$\dot{W}\le-\Lambda\tilde{\chi}^2-\sum_{i=0}^{2}\left(\frac{\Gamma_i(\hat{\kappa}_i-\kappa_i^*)^2}{2}-\frac{\Gamma_i\kappa_i^{*2}}{2}\right). \tag{3.29}$$

Using the definition of W in (3.23) yields

$$\dot{W}\le-\bar{\varrho}W+\frac{1}{2}\sum_{i=0}^{2}\Gamma_i\kappa_i^{*2}, \tag{3.30}$$

where $\bar{\varrho}\triangleq\frac{\min_i\{\Lambda,\ \Gamma_i/2\}}{\max\{\bar{\alpha}/2,\ 1/2\}}>0$ by design.

Define a scalar $0<\delta<\bar{\varrho}$. Then, $\dot{W}$ in (3.30) simplifies to

$$\dot{W}\le-\delta W-(\bar{\varrho}-\delta)W+\frac{1}{2}\sum_{i=0}^{2}\Gamma_i\kappa_i^{*2}. \tag{3.31}$$

Defining a scalar $\mathscr{B}_1\triangleq\frac{\sum_{i=0}^{2}\Gamma_i\kappa_i^{*2}}{2(\bar{\varrho}-\delta)}$, it can be noticed that $\dot{W}\le-\delta W$ when $W\ge\mathscr{B}_1$.

Case (ii): $|\tilde{\chi}|<\epsilon$.

Using (3.17) and (3.20), for this case we have

$$\begin{aligned}\dot{W}&\le-\Lambda\tilde{\chi}^2-\rho\frac{|\tilde{\chi}|^2}{\epsilon}+|\bar{\Delta}||\tilde{\chi}|+(\hat{\kappa}_2-\kappa_2^*)\beta_s\tilde{\chi}+\sum_{i=0}^{2}(\hat{\kappa}_i-\kappa_i^*)\dot{\hat{\kappa}}_i\\&\le-\Lambda\tilde{\chi}^2+\kappa_0^*|\tilde{\chi}|+\kappa_1^*|\tilde{\chi}|^2+(\hat{\kappa}_2-\kappa_2^*)\chi_\infty\frac{2}{\pi}\beta_s\sin(\chi)\tilde{\chi}\\&\quad+\sum_{i=0}^{2}(\hat{\kappa}_i-\kappa_i^*)\dot{\hat{\kappa}}_i\end{aligned} \tag{3.32}$$

Then, following the same lines of proof as in Case (i) we have

$$\dot{W}\le-\delta W-(\bar{\varrho}-\delta)W+\frac{1}{2}\sum_{i=0}^{2}\Gamma_i\kappa_i^{*2}+\hat{\kappa}_0|\tilde{\chi}|+\hat{\kappa}_1|\tilde{\chi}|^2. \tag{3.33}$$

In Case (ii) we have $|\tilde{\chi}| < \epsilon$. From (3.21a)-(3.21b) it can be noted that $|\tilde{\chi}| \in \mathscr{L}_\infty \Rightarrow \hat{\kappa}_0, \hat{\kappa}_1 \in \mathscr{L}_\infty$. Therefore, there exists $\varsigma \in \mathbb{R}^+$ such that $(\hat{\kappa}_0|\tilde{\chi}| + \hat{\kappa}_1|\tilde{\chi}|^2) \leq \varsigma$, yielding

$$\dot{W} \leq -\delta W - (\overline{\varrho} - \delta)W + \frac{1}{2}\sum_{i=0}^{2} \Gamma_i {\kappa_i^*}^2 + \varsigma \tag{3.34}$$

and $\dot{W} \leq -\delta W$ holds when $W \geq \mathscr{B}_2 \triangleq \frac{\frac{1}{2}\sum_{i=0}^{2}\Gamma_i {\kappa_i^*}^2 + \varsigma}{\overline{\varrho} - \delta}$.

Observing the results of Cases (i) and (ii) (i.e., (3.31) and (3.34)), we get $\dot{W} \leq -\delta W$ when $W \geq \max\{\mathscr{B}_1,\ \mathscr{B}_2\}$ and the closed-loop system is UUB, implying $\tilde{\chi}, \hat{\kappa}_i \in \mathscr{L}_\infty$ for $i = 0, 1, 2$. Further, the Lyapunov function as in (3.23) yields $W \geq (1/2)\overline{\alpha}\tilde{\chi}^2$. Therefore, following the definition of ultimate bound [38, Sect. 4.8], the ultimate bound $\mathscr{B}_s$ on the straight-line path tracking error $\tilde{\chi}$ is found to be

$$\mathscr{B}_s = \sqrt{\frac{2\max\{\mathscr{B}_1,\ \mathscr{B}_2\}}{\overline{\alpha}}}. \tag{3.35}$$

Tunability: the ultimate bound on the path tracking error can be considered as a performance indicator. From the structures of the error bounds $\mathscr{B}_1$ and $\mathscr{B}_2$ (as below (3.31) and (3.34)), one can derive that a high value of Λ and low values of Γ_i improve tracking accuracy. However, it should be noticed that increasing Λ or decreasing Γ_i result in a higher control input (due to the larger values of ρ): the trade-off between tracking error and control effort is standard in control, and requires to tune these parameters according to the application requirements.

ORBIT FOLLOWING

For control design purposes, the derivative of (2.26) is calculated in [7] as

$$\dot{\chi}^d = V_g\left(\frac{\sin(\chi - \gamma)}{d} + \lambda\beta_o\cos(\chi - \gamma)\right). \tag{3.36}$$

The corresponding guidance law is defined as

$$\chi^c = -\Lambda\tilde{\chi} + \chi + \hat{\kappa}_2\left(\frac{\sin(\chi - \gamma)}{d} + \lambda\beta_o\cos(\chi - \gamma)\right) - \rho\,\mathrm{sat}\left(\frac{\tilde{\chi}}{\varepsilon}\right), \tag{3.37a}$$

$$\rho = \hat{\kappa}_0 + \hat{\kappa}_1|\tilde{\chi}|, \tag{3.37b}$$

with the following adaptive laws:

$$\dot{\hat{\kappa}}_0 = |\tilde{\chi}| - \Gamma_0\hat{\kappa}_0, \tag{3.38a}$$

$$\dot{\hat{\kappa}}_1 = |\tilde{\chi}|^2 - \Gamma_1\hat{\kappa}_1, \tag{3.38b}$$

$$\dot{\hat{\kappa}}_2 = -\left(\frac{\sin(\chi - \gamma)}{d} - \lambda\beta_o\cos(\chi - \gamma)\right)\tilde{\chi} - \Gamma_2\hat{\kappa}_2, \tag{3.38c}$$

$$\text{with } \hat{\kappa}_i(0) > 0,\ i = 0, 1, 2. \tag{3.38d}$$

with similar design parameters as before.

The following result can be derived:

Theorem 3.3.4 *By employing the guidance law (3.37), the resulting trajectories of the UAV (3.17) and the parameters in the adaptive law (3.38) are Uniformly Ultimately Bounded (UUB).*

STABILITY ANALYSIS

The stability analysis for the orbit path follows similar steps as for the straight-line path case, with Lyapunov function (3.23) and *an overall uncertainty term* for the orbit path as

$$\Delta^c \triangleq \overline{\Delta} + \kappa_2^* \left(\frac{\sin(\chi-\gamma)}{d} + \lambda\beta_o \cos(\chi-\gamma) \right) \tag{3.39}$$

Observing the structure of sat(·) as in (3.37a), the overall stability analysis is carried out for the following two cases, using the common Lyapunov function (3.23).

Case (i): $|\tilde{\chi}| \geq \epsilon$. Using (3.17) and (3.36), we get

$$\begin{aligned} \dot{W} &= \tilde{\chi}(-\chi + \chi^c + \Delta^c) + \sum_{i=0}^{2} (\hat{\kappa}_i - \kappa_i^*)\dot{\hat{\kappa}}_i \\ &\leq -\Lambda\tilde{\chi}^2 - (\hat{\kappa}_0 - \kappa_0^*)|\tilde{\chi}| - (\hat{\kappa}_1 - \kappa_1^*)|\tilde{\chi}|^2 + \sum_{i=0}^{2} (\hat{\kappa}_i - \kappa_i^*)\dot{\hat{\kappa}}_i \\ &\quad + (\hat{\kappa}_2 - \kappa_2^*) \left(\frac{\sin(\chi-\gamma)}{d} + \lambda\beta_o \cos(\chi-\gamma) \right) \tilde{\chi}. \end{aligned} \tag{3.40}$$

From (3.38a)-(3.38c) we have

$$\sum_{j=0}^{1} (\hat{\kappa}_j - \kappa_j^*)\dot{\hat{\kappa}}_j = (\hat{\kappa}_j - \kappa_j^*)|\tilde{\chi}|^{j+1} - \Gamma_j \hat{\kappa}_j^2 + \Gamma_j \hat{\kappa}_j \kappa_j^*, \tag{3.41}$$

$$\begin{aligned} (\hat{\kappa}_2 - \kappa_2^*)\dot{\hat{\kappa}}_2 = (\hat{\kappa}_2 - \kappa_2^*)\tilde{\chi} \left(\frac{\sin(\chi-\gamma)}{d} + \lambda\beta_o \cos(\chi-\gamma) \right) \\ - \Gamma_2 \hat{\kappa}_2^2 + \Gamma_2 \hat{\kappa}_2 \kappa_2^*. \end{aligned} \tag{3.42}$$

The same simplifications (3.28) apply to the orbit case, leading along similar steps to

$$\dot{W} \leq -\delta W - (\underline{\varrho} - \delta)W + \frac{1}{2}\sum_{i=0}^{2} \Gamma_i {\kappa_i^*}^2. \tag{3.43}$$

Defining the scalar $\mathscr{B}_1$ as before, we have $\dot{W} \leq -\delta W$ when $W \geq \mathscr{B}_1$.

Case (ii): $|\tilde{\chi}| < \epsilon$. Using (3.17) and (3.36), for this case we get

$$\begin{aligned} \dot{W} &\leq -\Lambda\tilde{\chi}^2 - \rho(|\tilde{\chi}|^2/\epsilon) + |\bar{\Delta}||\tilde{\chi}| + \sum_{i=0}^{2} (\hat{\kappa}_i - \kappa_i^*)\dot{\hat{\kappa}}_i \\ &\quad + (\hat{\kappa}_2 - \kappa_2^*) \left(\frac{\sin(\chi-\gamma)}{d} + \lambda\beta_o \cos(\chi-\gamma) \right) \tilde{\chi} \\ &\leq -\Lambda\tilde{\chi}^2 + \kappa_0^*|\tilde{\chi}| + \kappa_1^*|\tilde{\chi}|^2 + \sum_{i=0}^{2} (\hat{\kappa}_i - \kappa_i^*)\dot{\hat{\kappa}}_i \\ &\quad + (\hat{\kappa}_2 - \kappa_2^*) \left(\frac{\sin(\chi-\gamma)}{d} + \lambda\beta_o \cos(\chi-\gamma) \right) \tilde{\chi}. \end{aligned} \tag{3.44}$$

We obtain that there exists $\varsigma \in \mathbb{R}^+$ such that $(\hat{\kappa}_0|\tilde{\chi}| + \hat{\kappa}_1|\tilde{\chi}|^2) \leq \varsigma$, giving

$$\dot{W} \leq -\delta W - (\underline{\varrho} - \delta)W + \frac{1}{2}\sum_{i=0}^{2} \Gamma_i {\kappa_i^*}^2 + \varsigma \tag{3.45}$$

and $\dot{W} \leq -\delta W$ when $W \geq \mathscr{B}_2$. The results (3.43) and (3.45) reveal that $\dot{W} \leq -\delta W$ when $W \geq \max\{\mathscr{B}_1, \mathscr{B}_2\}$ and the closed loop is UUB, implying $\tilde{\chi}, \hat{\kappa}_i \in \mathscr{L}_\infty$ for $i = 0, 1, 2$.

Tunability: following similar lines of Theorem 3.3.3, the ultimate bound $\mathscr{B}_o$ on path tracking error is analogous to (3.35), i.e.

$$\mathscr{B}_o = \sqrt{\frac{2\max\{\mathscr{B}_1, \mathscr{B}_2\}}{\overline{\alpha}}}. \tag{3.46}$$

Therefore, similar trade-offs arise: increasing Λ or decreasing Γ_i results in a smaller ultimate bound, but may result in higher control inputs.

Remark 3.3.1 *Differently from the standard VF method in (2.24) and (2.27), no a priori knowledge of the course time constant, the wind environment, and the unmodelled dynamics is required; the gains $\hat{\kappa}_0$ and $\hat{\kappa}_1$ compensate online the uncertainty term (3.19), stemming from the unmodelled term (3.15); the gain $\hat{\kappa}_2$ plays the role of an estimator for the ground velocity. The course time constant α is estimated jointly via $\hat{\kappa}_0$, $\hat{\kappa}_1$, $\hat{\kappa}_2$ (as κ_0^*, κ_1^*, κ_2^* all contain $1/\alpha$). These estimation actions mark a difference with the standard adaptive-free VF method and with other adaptive-free robust methods.*

Remark 3.3.2 *The adaptive laws in (3.21) and (3.38) reveal that the control gains adjust automatically according to the tracking error, thanks to the effect of the stabilizing leakage terms $-\Gamma_i\hat{\kappa}_i$, $i = 0, 1, 2$. In other words, the adaptive laws keep a balance between increasing the estimates when the error is large, and keeping the estimates bounded. As $\Gamma_0, \Gamma_1, \Gamma_2$ become smaller, adaptation is faster. However, this might lead to larger gains $\hat{\kappa}_0, \hat{\kappa}_1\hat{\kappa}_2$ (i.e. the uncertainty can be overestimated) and high control inputs. This indicates a trade-off between small control inputs and robustness to unmodeled dynamics.*

Remark 3.3.3 *The proposed guidance laws (3.20) and (3.37) share a structure similar to (adaptive) sliding mode control:*

$$\chi^c = \chi - \Lambda\tilde{\chi} + \dot{\chi}^d - \rho\,\mathrm{sat}\left(\frac{\tilde{\chi}}{\varepsilon}\right). \tag{3.47}$$

The main differences are that $\dot{\chi}^d$ is given a priori in adaptive sliding mode control (whereas we include adaptation due to the uncertainty in V_g), and that ρ estimates a constant bound for the uncertainty (whereas we estimate a state-dependent bound).

ALGORITHM EVALUATION

Experiments are carried out on a software-in-the-loop UAV platform where the functionalities of the ArduPilot autopilot are replicated in MATLAB, and on a hardware-in-the-loop UAV platform where PX4 autopilot hardware is connected to a Gazebo/ROS environment.

The experiments offer a way to compare different sliding mode control techniques in view of the following facts:

- The standard VF method is essentially a sliding mode control method that assumes parametric knowledge of the course time constant and the nominal wind environment;

- The ideal VF method is also a sliding mode control method, but with more knowledge of the wind disturbance;

- Our adaptive VF method is an advanced adaptive sliding mode control method without parametric knowledge. Yet, it is different from standard adaptive sliding mode control since the latter still requires nominal parametric knowledge and assumes the uncertainty to be bounded a priori.

Software-in-the-loop experiments

We take the following environmental conditions: constant wind amplitude is $W = 4$ m/s with wind angle $\psi_W = 230°$; and a Dryden turbulence [16, Sect. 4.4]. To draw conclusions on the effectiveness of adaptation in different conditions, all environmental conditions have been combined to obtain three wind scenarios, summarized in Table 3.6.

Just as in the previous simulation, the first-order time constant of the course dynamics can be estimated as $\alpha = 0.4578$. Both the standard and the ideal VF method use this time constant.

The performance of the standard, adaptive, and ideal VF method is first evaluated on primitive paths (straight-line and orbit), using the root mean square (RMS) steady-state tracking error calculated in the last portion of the path when e_{py} or $\tilde{d}$ have converged. The parameters χ^{∞}, k, ε, κ, Γ_0, Γ_1 and Γ_2 in Table 3.7 have been tuned so as to find a good compromise between convergence speed and smooth response.

Table 3.7: Parameters of the guidance laws

χ^{∞}	k	ε	κ	Γ_0, Γ_1	Γ_2
$\pi/2$	0.1 m^{-1}	1 rad	$\pi/2$ rad^2/s	0.01	0.001

Tables 3.8 and 3.9 (straight-line and orbit, respectively) highlight how the proposed adaptive VF method outperforms, in all scenarios, the standard and the ideal VF method. Note that in Scenarios 1 and 2 the standard and the ideal VF method have exactly the same performance since there is no wind perturbation.

Table 3.6: Flight environmental conditions

Scenario	Constant wind	Turbulence
#1	No	No
#2	Yes	No
#3	Yes	Yes

Table 3.8: Straight-line RMS tracking errors (in parentheses is the percentage increase of error against the adaptive VF method)

Scenario	Standard VF RMS error	Ideal VF RMS error	Adaptive VF RMS error
#1	0 **(+0%)**	0 **(+0%)**	0
#2	0.654 (**+38%**)	0.653 (**+38%**)	0.472
#3	0.673 (**+38%**)	0.673 (**+38%**)	0.488

Table 3.9: Orbit RMS tracking errors (in parentheses is the percentage increase of error against the adaptive VF method)

Scenario	Standard VF RMS error	Ideal VF RMS error	Adaptive VF RMS error
#1	0.146 $(+\infty)$	0.146 $(+\infty)$	0
#2	0.776 (**+76%**)	0.776 (**+76%**)	0.441
#3	0.821 (**+89%**)	0.798 (**+84%**)	0.434

For the straight-line case, Figure 3.5 clearly shows that the adaptive VF better counteracts with time the effect of the wind in Scenario 3 (38% improvement). Even with exact knowledge of the wind, the ideal VF method performs quite poorly, due to the inaccurate knowledge of α (the adaptive VF method again gives 38% improvement). Something similar also occurs in Scenario 2 (constant wind).

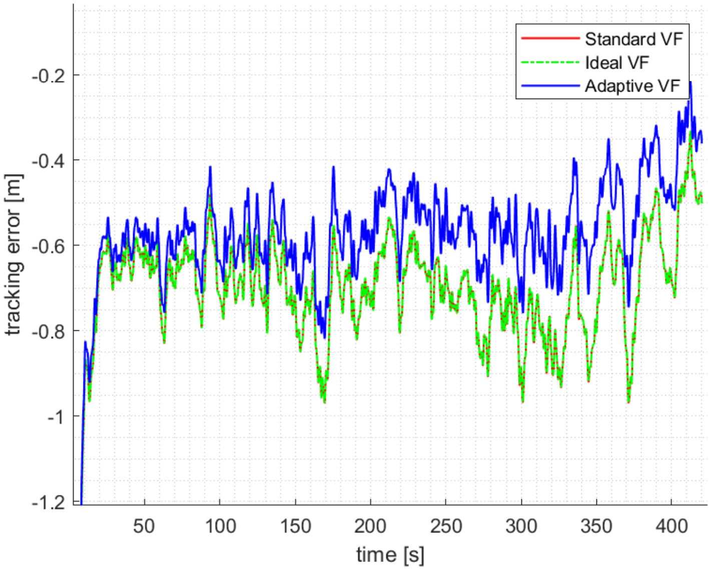


Figure 3.5: Straight-line, Scenario 3: tracking error (the standard and ideal VF method have similar performance and their lines overlap)

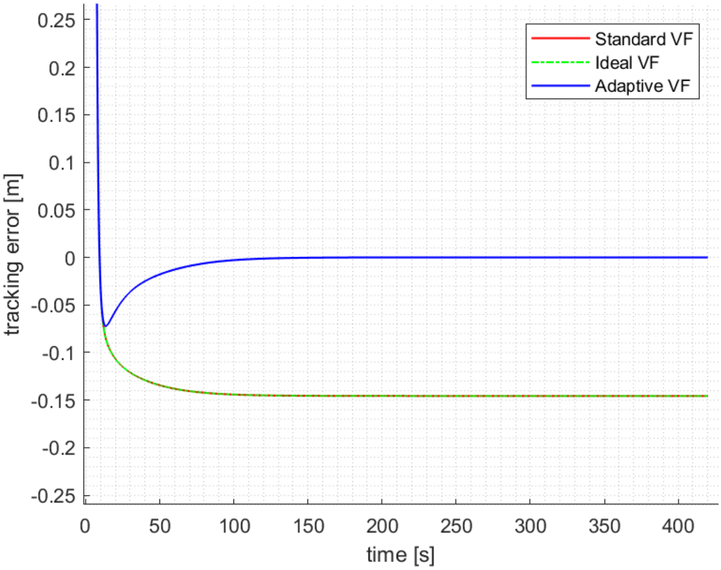


Figure 3.6: Orbit, Scenario 1: tracking error (the standard and ideal VF method have the same performance and their lines overlap)

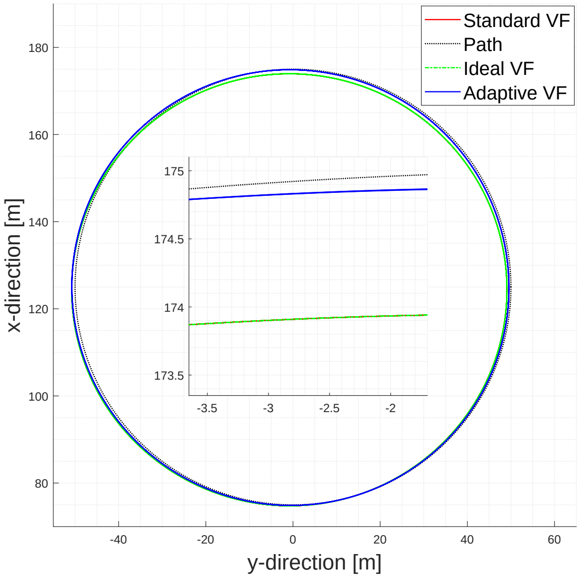


Figure 3.7: Orbit, Scenario 2: path in x-y plane (the small box is a zoom-in to highlight the improved tracking of the adaptive VF method)

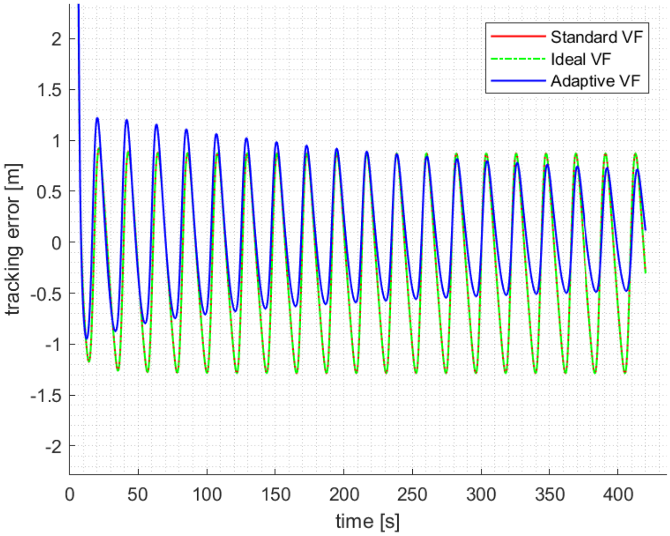


Figure 3.8: Orbit, Scenario 2: tracking error (the standard and ideal VF method have the same performance and their lines overlap)

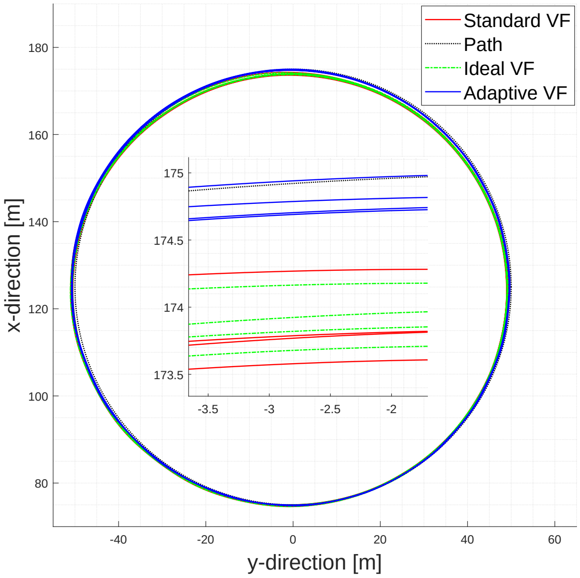


Figure 3.9: Orbit, Scenario 3: path in x-y plane (the small box is a zoom-in to highlight the improved tracking of the adaptive VF method)

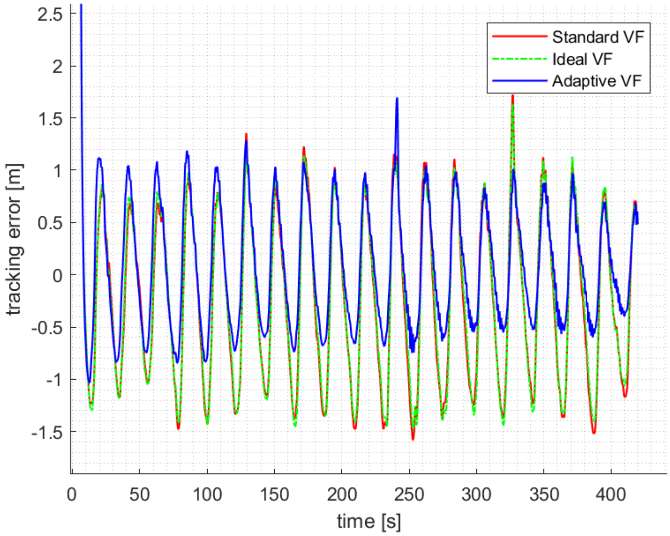


Figure 3.10: Orbit, Scenario 3: tracking error (the standard and ideal VF method have similar performance and their lines almost overlap)

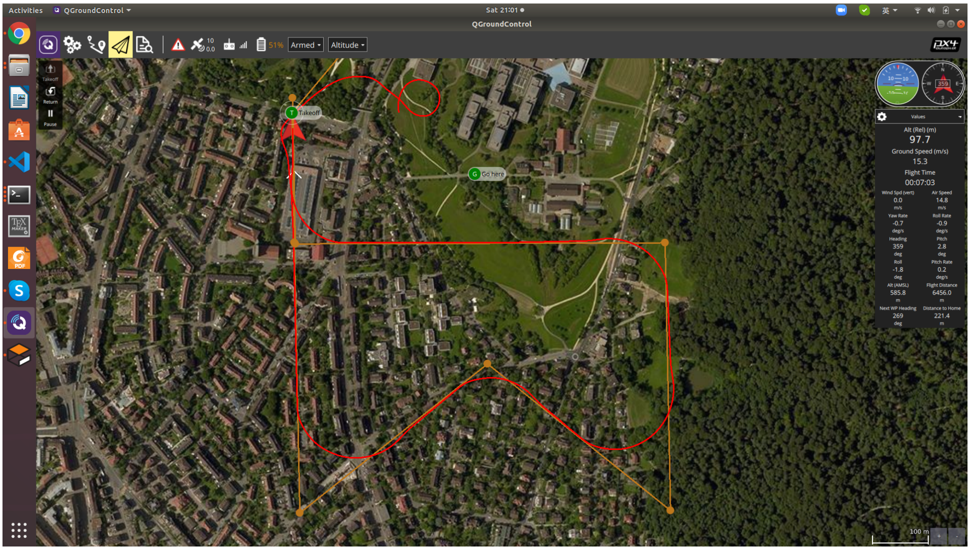
For the orbit case, Figure 3.6 clearly shows that the standard and the ideal VF method have a steady-state tracking error: such error is completely removed by the adaptive VF method. A significant reduction of the tracking error (76-89%) by the adaptive VF method also occurs in Scenarios 2 and 3, which are depicted in Figures 3.7-3.10 in terms of tracking error and path in the x-y plane. In all cases it can be seen that the adaptation mechanism reduces the oscillations of the error: oscillations are present due to the fact that the wind effect changes when the UAV is travelling along the orbit. The adaptive gains $\kappa_0, \kappa_1, \kappa_2$ can verify the boundedness of the gains directly from the adaptive laws (3.21) and (3.38): if the tracking error $\tilde{\chi}$ is bounded, then the $\hat{\kappa}_i$s are bounded using bounded-input-bounded-output notions, as the Γ_is are positive constants.

HARDWARE-IN-THE-LOOP EXPERIMENTS

Similarly to the software-in-the-loop experiments, we define several wind scenarios and paths to test the performance in different environments. We have a scenario with average wind speed 2 m/s with variance of 0.5 m/s and direction $\psi_W = 45°$ (Scenario #4), and a scenario with an average wind speed of 5 m/s, direction $\psi_W = 45°$, variance of 0.5 m/s and gusts up to 7 m/s (Scenario #5). We define three paths: a straight-line path, an orbit path, and a combined path with lines and orbits (cf. Figure 3.11). The results of the guidance laws are shown in Table 3.10 in terms of RMS error. Notice that the standard VF method is implemented in two conditions: one with $\alpha = 0.4578$, and one where α has been carefully tuned so as to improve performance. Because the adaptive VF method is able to improve even over the optimized standard VF method, this further validates the effectiveness of the proposed strategy: even if the optimized α makes the standard VF method at least four times better, still improvements of 3-19% are observed thanks to adaptation. As compared to the Scenario #4, it can be seen that Scenario #5 is quite extreme for the UAV, but still the proposed adaptive VF method outperforms all strategies. The TECS-L1 guidance works good for orbit following under low wind speed (Scenario #4, only 1% degradation) but is less effective for straight-line path and high wind speed , 102-462% degradation).

Table 3.10: 3D simulation RMS tracking errors (in parentheses is the percentage increase of error against the adaptive VF method)

	TECS-L1 RMS error	Standard VF RMS error ($\alpha = 0.4578$)	Standard VF RMS error (optimized α)	Adaptive VF RMS error
Line				
#4	5.161 **(+462%)**	1.419 **(+54%)**	1.036 **(+13%)**	0.919
#5	11.62 **(+102%)**	12.20 **(+112%)**	5.932 **(+3%)**	5.762
Orbit				
#4	2.319 **(+1%)**	2.613 **(+14%)**	2.370 **(+3%)**	2.300
#5	5.810 **(+102%)**	6.105 **(+112%)**	2.966 **(+3%)**	2.881
Combi.				
#4	6.705 **(+333%)**	4.416 **(+185%)**	1.839 **(+19%)**	1.548
#5	18.95 **(+233%)**	16.68 **(+194%)**	5.804 **(+2%)**	5.683



Alt (Rel) (m) 97.7
Ground Speed (m/s) 15.3
Flight Time 00:07:03
Wind Spd (vert) 0.0 m/s
Air Speed 14.8 m/s
Yaw Rate -0.7 deg/s
Roll Rate -0.9 deg/s
Heading 359 deg
Pitch 2.8 deg
Roll -1.8 deg
Pitch Rate 0.2 deg/s
Alt (AMSL) 585.8 m
Flight Distance 6456.0 m
Next WP Heading 269 deg
Distance to Home 221.4 m

Figure 3.11: Combined path, Scenario 5, adaptive VF method

Overall the simulations show that the proposed adaptive VF method, by compensating for the lack of knowledge in course dynamics and wind environment, can bring to improved guidance performance in several wind and path scenarios. It is intuitive to expect that when the autopilot layer is poorly tuned, any guidance algorithm can do little to cope with this situation: investigating this point in an analytic or numerical way could be an interesting future work.

3.4. Summary

The actual performance of model-based path-following methods for Unmanned Aerial Vehicles (UAVs) shows considerable dependence on the knowledge of the wind speed and direction, and on the accuracy of the dynamic model used for design. In this chapter, we analyzed how such uncertainties appear in the UAV dynamics.

In Sect. 3.3.1, we have proposed a method to relax the first-order dynamic assumption in previous literature. We use adaptation of $\hat{V}_g$ in place of the unknown wind speed V_g, where $\hat{V}_g$ is adapted by an auxiliary differential equation. The estimator compensates for the lack of knowledge of the wind vector and for unmodelled course dynamics.

In Sect. 3.3.2, we proposed a method not requiring a priori knowledge of the UAV course time constant, of coupling effects, and of the wind amplitude/direction.

Extensive experiments have shown that the proposed adaptive VF methods are able to compensate for the lack of knowledge on the course dynamics and the wind environment, so as to bring improved guidance performance in several wind and path scenarios.

4

ADAPTIVE FORMATION CONTROL FOR FIXED-WING UAVS

4.1. INTRODUCTION

In recent years, the synchronization of multi-agent systems has been an emerging research direction drawing the attention of the control community. Synchronization represents a potential solution for coordination of large-scale networked systems [58, 59], encompassing spacecraft attitude control [60], sensor networks [61], smart buildings and smart grids [62, 63], unmanned aerial, ground and underwater vehicles [64–66]. To coordinate multi-agent systems, either a centralized approach or a distributed approach can be adopted. The centralized approach introduces a central node that utilizes the information stemming from all agents: standard ground station control is usually implemented in a centralized way. On the contrary, the distributed approach uses, for each agent, a controller that utilizes local information, e.g. neighbors' information. The distributed approach provides advantages due to its applicability in the presence of communication constraints, for example, when continuous communication with the ground station may be impossible [2, 67, 68]. The research in distributed control can be grouped into several directions that may be overlapping, such as synchronization (sometimes referred to as consensus or rendezvous when the synchronizing behavior is constant), distributed formation control (sometimes referred to as flocking in the presence of collision avoidance capabilities), distributed optimization and estimation [69].

In this chapter, distributed synchronization is studied as a way to control formations of uncertain agents with Euler–Lagrange (EL) dynamics. Synchronization is to be intended as the whole network to converge to a common trajectory. Clearly, this requires the state of some agent to track the state of another agent with which a communication link exists. In this sense, achieving synchronization requires to solve a tracking problem among connected agents. Based on the modelling and autopilot design for fixed-wing UAVs presented in Chapter 2, we will test the algorithm in the software-in-the-loop simulator and show the capability of handling parametric uncertainty in the UAV structure (uncertain mass and inertia) [47].

The rest of the chapter is organized as follows: in Sect. 4.2 the problem will be introduced. In Sect. 4.3, an algorithm for adaptive vector field path following is given, followed by the proposed adaptive formation control method. Sect. 4.4 illustrates the approach with software-in-the-loop simulation tests. Finally, in Sect. 4.5 some concluding remarks are presented.

4.2. PROBLEM FORMULATION

A multi-UAV formation control system typically consists of a cooperative guidance system and a cooperative control system. The cooperative control system aims to control the attitude in the inner loop, and the cooperative guidance system is used to control the position in the outer loop, where the output of the outer loop is used as the input of the inner loop. Synchronization is achieved when the attitudes converge to a common value and the positions converge to a common value (up to some formation gaps). Furthermore, synchronization is added adaptive when it is achieved even in the presence of uncertain system dynamics and/or uncertain environmental dynamics.

Given these challenges, this work is driven by the following research questions: how to cope with parametric uncertainties (uncertain mass and inertia) in UAV dynamics? How to account for the autopilot low-level control when testing path-following algorithms? How to scale the path-following problem to teams of UAVs?

Let us start by formulating the problem. The main variables behind the formation control law are collected in Table 4.1, and their explanation is given hereafter.

Table 4.1: Variables for adaptive formation control law.

Variable	Description
A_m, B_m	Reference dynamics
K_p, K_v	Reference gains
P	Lyapunov matrix
Γ	Adaptive gain
$\hat{D}_i, \hat{C}_i, \hat{g}_i$	Estimated dynamics of UAV_i
$\Theta_{Di}, \Theta_{Ci}, \Theta_{gi}$	Estimated gains of UAV_i
$\widehat{D_i D_{ji}}, \widehat{D_i D_j C_{ji}}$	Estimated dynamics between UAV_i and UAV_j
$\Theta_{D_i D_j}, \Theta_{D_i D_j D_j}$	Estimated gains between UAV_i and UAV_j

4.2.1. TEAM OF UAVS AS A MULTI-AGENT SYSTEM

In Section 2.1, we have introduced the UAV dynamics in the framework of Euler-Lagrange dynamics. When we have multiple UAVs, we can consider that team as a multi-agent system having dynamics as

$$D_i(q_i)\ddot{q}_i + C_i(q_i, \dot{q}_i)\dot{q}_i + g_i(q_i) = \tau_i, \quad i = \{1, ..., N\} \tag{4.1}$$

where q_i is as in (2.9) where the subscript $i \in \{1, ..., N\}$ uniquely identifies each agent in the team of N UAVs, the term $D_i(q_i)\ddot{q}_i$ is proportional to the second derivatives of the generalized coordinates, the term $C_i(q_i, \dot{q}_i)\dot{q}_i$ is the vector of centrifugal/Coriolis forces, proportional to the first derivatives of the generalized coordinates, the term $g_i(q_i)$ is the vector of potential forces, and the term τ_i represents the external force applied to the system. Throughout this work the following assumptions, standard in literature [70], will be adopted:

Assumption 4.2.1 *The inertia matrix $D_i(q_i)$ is symmetric positive definite, and both $D_i(q_i)$ and $D_i^{-1}(q_i)$ are uniformly bounded.*

Assumption 4.2.2 *There is an independent control input for each degree of freedom of the system.*

Assumption 4.2.3 *All the parameters of interest such as link masses, moments of inertia, etc. appear in a linear-in-the parameter form, i.e. as coefficients of known functions of the generalized coordinates.*

Remark 4.2.1 *For most EL agents of practical interest, like robotic manipulators and mobile vehicles, Assumptions 4.2.1 and 4.2.3 hold [70]. Assumption 4.2.2 is clearly restrictive and implies that the system is fully actuated, which is not always the case in practice. For under-actuated EL systems, a control allocator should be put in place to transform the input τ_i into the actual inputs to the system: this will introduce unmodeled dynamics, which can be handled by modifying the proposed methodology in a robust adaptive sense [71]. While this is a relevant practical aspect, in this chapter we focus for compactness on fully actuated EL dynamics. Chapter 5 will be specifically focused on fixed-wing UAVs and address the inevitable under-actuation.*

4.2.2. PRELIMINARIES ON COMMUNICATION GRAPHS

Most formation control algorithms are graph-based approaches. This means that the agents are linked to each other via a *communication graph* that describes the allowed information flow (cf. Figure 4.1). The graph is composed of nodes and directed edges (represented by arrows). The nodes in the graph can be classified into three types:

- *Path Planner Node:* This node, typically indicated as node 0, decides the path for the complete set of UAVs. The node does not receive information from any other node (UAV) and also generates the dynamics to which all other nodes should synchronize. Thus, the node is also called *pinner node* in some literature.
- *Leader Node:* Each leader node in the formation has access to the data from the pinner node.
- *Follower Nodes:* Each follower node in the formation has only access to the data from nodes other than the pinner node (e.g. the data from a leader or the data from another follower).

The communication graph describing the allowed information flow between all the nodes, pinner excluded, is completely defined by the pair $G = (V, E)$, where $V = \{1, ..., N\}$ is a finite non-empty set of nodes, and $E \subseteq V \times V$ is a set of pairs of nodes, called edges. To include the presence of the pinner in the network we define $\bar{G} = \{V, E, T\}$, where $T \subseteq V$ is the set of those nodes, called *target nodes*, that receive information from the pinner. Let us introduce the *adjacency matrix* $A = [a_{ij}] \in \mathbb{R}^{N\times N}$ of a directed communication graph, which is defined as $a_{ii} = 0$ and $a_{ij} = 1$, if $(i, j) \in E$, where $i \neq j$. In addition, we define a vector, the *target vector* $M = [a_{j0}] \in \mathbb{R}^N$, to describe the directed communication of the pinner with the target nodes. The target matrix is defined as $a_{j0} = 1$ if $j \in T$ and $a_{j0} = 0$ otherwise.

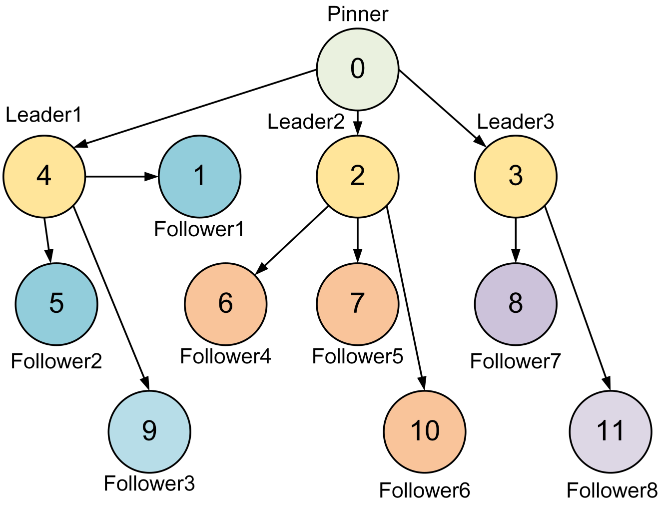


Figure 4.1: Example of communication graph for UAV formation control.

Control Problem: Given a hierarchical network $\bar{G}$ of EL heterogeneous uncertain UAVs, and a pinner with state $(q_0, \dot{q}_0)$, we want to find a distributed strategy for the inputs τ_i that respects the communication graph, that does not require knowledge of the EL matrices, and that leads to synchronization of the network, i.e. $[q_i, \dot{q}_i] \to [q_0, \dot{q}_0]$ as $t \to \infty$ for every UAV i.

4.3. ADAPTIVE UAVS FORMATION CONTROL LAW

4.3.1. REFERENCE DYNAMICS FOR LEADER/FOLLOWER SYNCHRONIZATION

Let us start by formulating some reference dynamics:

$$\begin{bmatrix} \dot{q}_0 \\ \ddot{q}_0 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & \mathbb{1} \\ -K_p & -K_v \end{bmatrix}}_{A_m} \underbrace{\begin{bmatrix} q_0 \\ \dot{q}_0 \end{bmatrix}}_{x_m} + \underbrace{\begin{bmatrix} 0 \\ \mathbb{1} \end{bmatrix}}_{B_m} r \tag{4.2}$$

where $q_0, \dot{q}_0 \in \mathbb{R}^n$ is the state of the reference model, K_p, K_v can be interpreted as the proportional and derivative gains of a multivariable PD controller, and $r = \ddot{q}^d + K_v \dot{q}^d + K_p q^d$ is a control input.

If the guidance method in the path planner layer is based on a vector field approach, it is clear that this does not ensure the dynamics to be as (4.2). Thus, the path planner layer also needs to generate some reference dynamics to which all UAVs in the formation should synchronize. From (2.9), the EL dynamics of a UAV can be expressed as

$$D_i(q_i)\ddot{q}_i + C_i(q_i,\dot{q}_i)\dot{q}_i + g_i(q_i) = \tau_i \tag{4.3}$$

On using an inverse dynamic based controller of the form in (4.4) we obtain the dynamics as in (4.2) if

$$\tau_i = D_i(q_i)a_i + C_i(q_i,\dot{q}_i)\dot{q}_i + g_i(q_i) \tag{4.4}$$

where the term a_i is defined as

$$a_i = \ddot{q}_i{}^d - K_v\dot{e}_i - K_p e_i \tag{4.5}$$

with $e_i = q_i - q_i{}^d$.

Substituting (4.4) in (4.3), we obtain the error dynamics as

$$\ddot{e}_i + K_v\dot{e}_i + K_p e_i = 0 \tag{4.6}$$

which can be re-written in state space form as,

$$\begin{bmatrix}\dot{e}_i\\ \ddot{e}_i\end{bmatrix} = \begin{bmatrix}0 & \mathbb{1}\\ -K_p & -K_v\end{bmatrix}\begin{bmatrix}e_i\\ \dot{e}_i\end{bmatrix} \tag{4.7}$$

which is equivalent to (4.2) after some rearrangement. Since K_p, K_v are positive gains and same for all UAVs, by construction the state matrix in (4.7) is Hurwitz. This implies as $t \to \infty$, $e_i \to 0$ [72] i.e. $q_i \to q_i{}^d$.

In the path planner node, we run the dynamics in (4.2) virtually with $q_i{}^d$, $\dot{q}_i{}^d$,and $\ddot{q}_i{}^d$ as the inertial measurements (trajectories, velocities, and accelerations) of the path planner UAV. By this method, the reference states x_m will be in close match to the states of the path planner UAV and have the dynamics in (4.2). The reference states x_m are further transmitted to the leader nodes for leader synchronization.

4.3.2. ADAPTIVE SYNCHRONIZATION OF LEADER DYNAMICS TO REFERENCE DYNAMICS

For easiness of presentation, let us now use the specific formation in Figure 4.2 (node 0 is the pinner node with reference dynamics, nodes 1 and 2 are the leader nodes, nodes 3 and 4 are the follower node) to present the main design steps. The first step is to let leader node 1 synchronize to node 0, and likewise for leader node 2. Any leader dynamics in the form (2.10) can be written in the state-space form

$$\begin{bmatrix}\dot{q}_1\\ \ddot{q}_1\end{bmatrix} = \underbrace{\begin{bmatrix}0 & \mathbb{1}\\ 0 & -D_1^{-1}C_1\end{bmatrix}}_{A_1}\underbrace{\begin{bmatrix}q_1\\ \dot{q}_1\end{bmatrix}}_{x_1} + \begin{bmatrix}0\\ -D_1^{-1}g_1\end{bmatrix} + \underbrace{\begin{bmatrix}0\\ D_1^{-1}\end{bmatrix}}_{B_1}\tau_1 \tag{4.8}$$

where the dependence of the matrices on q_1, $\dot{q}_1$ will be omitted whenever obvious.

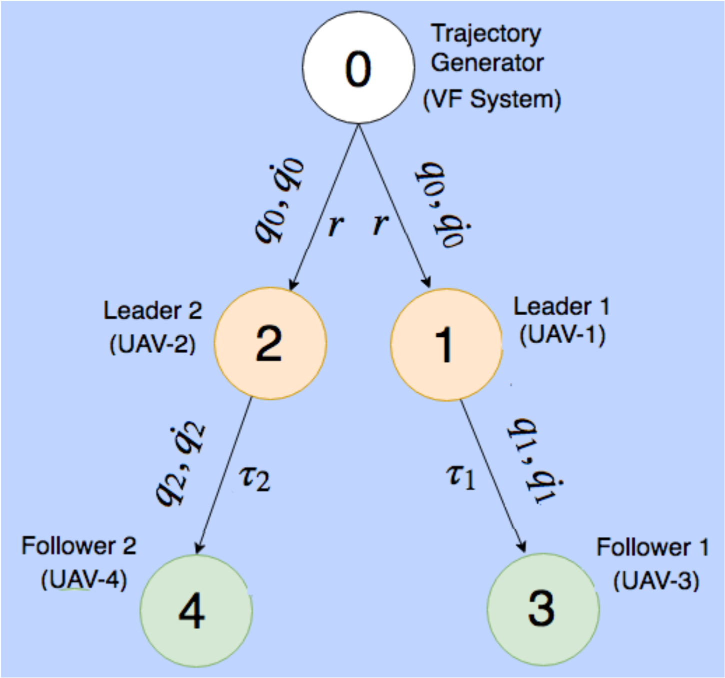


Figure 4.2: Communication graph with V formation.

The main idea to synchronize the leader dynamics to the reference dynamics is to formulate a nonlinear version of the model reference adaptive control method [73, 74], by designing a controller to match the leader dynamics in state-space form like (4.8) to the reference model dynamics (4.2). To this purpose, we propose a controller of the form

$$\tau_1^* = \underbrace{\begin{bmatrix} \bar{K}_1^{*\prime} & \bar{\bar{K}}_1^{*\prime} \end{bmatrix}}_{K_1^{*\prime}} \begin{bmatrix} q_1 \\ \dot{q}_1 \end{bmatrix} + G_1^{*\prime} + L_1^{*\prime} r \tag{4.9}$$

where the superscript * indicates an ideal controller whose gains possibly require the knowledge of the system dynamics. The following proposition tells how to find such matching gains.

Proposition 4.3.1 *There exists an ideal control law in the form of* (4.9) *that matches the leader dynamics in state-space form* (4.8) *to the reference model dynamics* (4.2)*, and whose control gains* $\bar{K}_1^*, \bar{\bar{K}}_1^*, L_1^*$*, and* G_1^* *are*

$$\begin{aligned} \bar{K}_1^{*\prime} &= -D_1 K_p & L_1^{*\prime} &= D_1 \\ \bar{\bar{K}}_1^{*\prime} &= -D_1 K_v + C_1 & G_i^{*\prime} &= g_1. \end{aligned} \tag{4.10}$$

Proof. By direct substitution of (4.9) into (4.8), we have the leader closed-loop dynamics

$$\begin{bmatrix} \dot{q}_1 \\ \ddot{q}_1 \end{bmatrix} = \begin{bmatrix} 0 & \mathbb{1} \\ D_1^{-1}\bar{K}_1^{*\prime} & D_1^{-1}(\bar{\bar{K}}_1^{*\prime} - C_1) \end{bmatrix} \begin{bmatrix} q_1 \\ \dot{q}_1 \end{bmatrix} + \begin{bmatrix} 0 \\ -D_1^{-1}(g_1 - G_1^{*\prime}) \end{bmatrix} + \begin{bmatrix} 0 \\ D_1^{-1}L_1^{*\prime} \end{bmatrix} r. \tag{4.11}$$

We see that Proposition 4.3.1 is verified for the ideal control law

$$\tau_1^* = -D_1K_pq_1 - D_1K_v\dot{q}_1 + C_1\dot{q}_1 + g_1 + D_1r \tag{4.12}$$

from which we derive the control gains in (4.10). This concludes the proof.

With reference to the formation given in Figure 4.2 (UAV 1 and UAV 2 are leader nodes), we propose the controllers for these UAVs:

$$\begin{aligned} \tau_1 &= \underbrace{\Theta'_{D1}\xi_{D1}}_{\hat{D}_1}(-K_pq_1 - K_v\dot{q}_1 + r) + \underbrace{\Theta'_{C1}\xi_{C1}}_{\hat{C}_1}\dot{q}_1 + \underbrace{\Theta'_{g1}\xi_{g1}}_{\hat{g}_1} \\ \tau_2 &= \underbrace{\Theta'_{D2}\xi_{D2}}_{\hat{D}_2}(-K_pq_2 - K_v\dot{q}_2 + r) + \underbrace{\Theta'_{C2}\xi_{C2}}_{\hat{C}_2}\dot{q}_2 + \underbrace{\Theta'_{g2}\xi_{g2}}_{\hat{g}_2} \end{aligned} \tag{4.13}$$

where the estimates $\hat{D}_1$, $\hat{C}_1$, $\hat{g}_1$ and $\hat{D}_2$, $\hat{C}_2$, $\hat{g}_2$ of the ideal matrices have been split in a linear-in-the-parameter form (i.e. any dynamic term is split as $\Theta'\xi(q,\dot{q})$ for some unknown parameter Θ and some known state-dependent regressor $\xi(q,\dot{q})$).

In fact, in view of Assumption 4.2.3, an appropriate linear-in-the-parameter form $D_1 = \Theta^{*\prime}_{D_1}\xi_{D_1}$, $C_1 = \Theta^{*\prime}_{C_1}\xi_{C_1}$ and $g_1 = \Theta^{*\prime}_{g_1}\xi_{g_1}$ can always be found. A specific form of regressand Θ and regressor ξ will be derived later in Sect. 4.4 for the example of UAVs. Let us define the error $e_1 = x_1 - x_m$, whose dynamics are

$$\begin{aligned} \dot{e}_1 &= A_me_1 + B_1(\tilde{\bar{K}}_1'q_1 + \tilde{\bar{\bar{K}}}_1'\dot{q}_1 + \tilde{G}_1' + \tilde{L}_1'r) \\ &= A_me_1 + B_1(\tilde{\Theta}'_{D_1}\xi_{D_1}(-K_pq_1 - K_v\dot{q}_1 + r) + \tilde{\Theta}'_{C_1}\xi_{C_1}\dot{q}_1 + \tilde{\Theta}'_{g_1}\xi_{g_1}) \end{aligned} \tag{4.14}$$

where $\tilde{\bar{K}}_1 = \bar{K}_1 - \bar{K}_1^*$, $\tilde{\bar{\bar{K}}}_1 = \bar{\bar{K}}_1 - \bar{\bar{K}}_1^*$, $\tilde{L}_1 = L_1 - L_1^*$, $\tilde{\Theta}_{D_1} = \Theta_{D_1} - \Theta^*_{D_1}$, $\tilde{\Theta}_{C_1} = \Theta_{C_1} - \Theta^*_{C_1}$, and $\tilde{\Theta}_{g_1} = \Theta_{g_1} - \Theta^*_{g_1}$. The following theorem provides the synchronization result between the leader and the reference model[1].

Theorem 4.3.1 *Consider the reference model* (4.2)*, the unknown leader dynamics* (4.8)*, and controller* (4.13)*. Under the assumption that a matrix S_1 exists such that*

$$L_1^*S_1 = S_1'L_1^{*\prime} > 0 \tag{4.15}$$

then, the adaptive laws for UAV 1

$$\begin{aligned} \dot{\Theta}'_{D_1} &= -S_1B_m'Pe_1(-K_pq_1 - K_v\dot{q}_1 + r)'\xi'_{D_1} \\ \dot{\Theta}'_{C_1} &= -S_1B_m'Pe_1\dot{q}_1'\xi'_{C_1} \\ \dot{\Theta}'_{g_1} &= -S_1B_m'Pe_1\xi'_{g_1} \end{aligned} \tag{4.16}$$

---

[1] From here till the end of the chapter we assume for simplicity the inertia matrix to be constant. This is done to make the presentation consistent with the well-known methods of multivariable adaptive control [75]. Most EL systems in body coordinates have a constant inertia matrix (cf. Sect. 4.4). For coordinate-dependent inertia matrices, the inverse of the estimated matrix should be included in the regressors, cf. [70, Sect. 3.2].

where $P = P' > 0$ *is such that*

$$PA_m + A'_m P = -Q, \quad Q > 0 \tag{4.17}$$

guarantee synchronization of the leader dynamics (4.8) *to the reference model* (4.2)*, i.e.* $e_1 \to 0$.

With reference to Figure 4.2, the control law of UAV 2 is similar as UAV 1.

Proof. To analytically show the asymptotic convergence of the synchronization error between the leader and the reference model, let us introduce the following Lyapunov function:

$$\begin{aligned} V_1(e_1,\tilde{\Theta}_{D_1},\tilde{\Theta}_{C_1},\tilde{\Theta}_{g_1}) = {} & e_1' P e_1 + \mathrm{tr}(\tilde{\Theta}'_{\mathrm{D}_1} \mathrm{S}_1^{-1} \mathrm{L}_1^{*-1} \tilde{\Theta}_{\mathrm{D}_1}) \\ & + \mathrm{tr}(\tilde{\Theta}'_{\mathrm{C}_1} \mathrm{S}_1^{-1} \mathrm{L}_1^{*-1} \tilde{\Theta}_{\mathrm{C}_1}) + \mathrm{tr}(\tilde{\Theta}'_{\mathrm{g}_1} \mathrm{S}_1^{-1} \mathrm{L}_1^{*-1} \tilde{\Theta}_{\mathrm{g}_1}). \end{aligned} \tag{4.18}$$

Then it is possible to verify that

$$\begin{aligned} \dot{V}_1(e_1,\tilde{\Theta}_{D_1},\tilde{\Theta}_{C_1},\tilde{\Theta}_{g_1}) = {} & e_1'(PA_m + A'_m P)e_1 + 2e_1' PB_1(\tilde{\Theta}'_{D_1} \xi_{D_1}(-K_p q_1 - K_v \dot{q}_1 + r) \\ & + \tilde{\Theta}'_{C_1} \xi_{C_1} \dot{q}_1 + \tilde{\Theta}'_{g_1} \xi_{g_1}) + 2\mathrm{tr}(\tilde{\Theta}'_{\mathrm{D}_1} \mathrm{S}_1^{-1} \mathrm{L}_1^{*-1} \dot{\tilde{\Theta}}_{\mathrm{D}_1}) \\ & + 2\mathrm{tr}(\tilde{\Theta}'_{\mathrm{C}_1} \mathrm{S}_1^{-1} \mathrm{L}_1^{*-1} \dot{\tilde{\Theta}}_{\mathrm{C}_1}) + 2\mathrm{tr}(\tilde{\Theta}'_{\mathrm{g}_1} \mathrm{S}_1^{-1} \mathrm{L}_1^{*-1} \tilde{\Theta}_{\mathrm{g}_1}) \\ = {} & -e_1' Q e_1 + 2\mathrm{tr}(\tilde{\Theta}'_{\mathrm{D}_1} \mathrm{L}_1^{*-1} (\mathrm{B}'_{\mathrm{m}} \mathrm{P} \mathrm{e}_1 (-\mathrm{K_p} \mathrm{q}_1 - \mathrm{K_v} \dot{\mathrm{q}}_1 + \mathrm{r})' \xi'_{\mathrm{D}_1} + \mathrm{S}_1^{-1} \dot{\tilde{\Theta}}_{\mathrm{D}_1})) \\ & + 2\mathrm{tr}(\tilde{\Theta}'_{\mathrm{C}_1} \mathrm{L}_1^{*-1} (\mathrm{B}'_{\mathrm{m}} \mathrm{P} \mathrm{e}_1 \dot{\mathrm{q}}'_1 \xi'_{\mathrm{C}_1} + \mathrm{S}_1^{-1} \dot{\tilde{\Theta}}_{\mathrm{C}_1})) \\ & + 2\mathrm{tr}(\tilde{\Theta}'_{\mathrm{g}_1} \mathrm{L}_1^{*-1} (\mathrm{B}'_{\mathrm{m}} \mathrm{P} \mathrm{e}_1 \xi'_{\mathrm{g}_1} + \mathrm{S}_1^{-1} \dot{\tilde{\Theta}}_{\mathrm{g}_1})) \\ = {} & -e_1' Q e_1. \end{aligned} \tag{4.19}$$

Here we used the property $a'b = \mathrm{tr}(\mathrm{b}'\mathrm{a})$. From (4.19), we deduce that V_1 has a finite limit, so $e_1, \tilde{\Theta}_{D_1}, \tilde{\Theta}_{C_1}, \tilde{\Theta}_{g_1} \in \mathcal{L}_\infty$. Because $e_1 = x_1 - x_m \in \mathcal{L}_\infty$ and $x_m \in \mathcal{L}_\infty$, we have $x_1 \in \mathcal{L}_\infty$. This implies that $x_1, \tilde{\Theta}_{D_1}, \tilde{\Theta}_{C_1}, \tilde{\Theta}_{g_1} \in \mathcal{L}_\infty$. Consequently, we can deduce that $\tau_1 \in \mathcal{L}_\infty$. Therefore, all signals in the closed-loop systems are bounded. From (4.19), we can establish that V_1 has a bounded integral, so that we have $e_1 \in \mathcal{L}_2$. Then by using $\tilde{\Theta}_{D_1}, \tilde{\Theta}_{C_1}, \tilde{\Theta}_{g_1}, e_1 \in \mathcal{L}_\infty$, we have $\dot{e}_1 \in \mathcal{L}_\infty$. This concludes the proof of the boundedness of all closed-loop signal and convergence $e_1 \to 0$ as $t \to \infty$.

Remark 4.3.1 *Condition* (4.15) *is inspired by the well-known condition of multivariable MRAC [75]: even though such condition might sound restrictive because it involves a possibly unknown matrix* L_1^**, it can be easily satisfied in most EL systems of practical interests. In fact, in most EL systems like robotic manipulators and mobile robots, the matrix* D_i *is symmetric in view of some symmetrical geometry of the robot: this implies that* L_i^**, even if unknown, is symmetric. Therefore,* (4.15) *is satisfied by simply selecting* $S_i = \gamma I$*, for any positive scalar* γ*.*

With reference to Figure 4.2, the dynamics of any follower (UAV 3 and 4) in the form (2.10) can be written in the state-space form

$$\begin{bmatrix} \dot{q}_3 \\ \ddot{q}_3 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & \mathbb{1} \\ 0 & -D_3^{-1} C_3 \end{bmatrix}}_{A_3} \underbrace{\begin{bmatrix} q_3 \\ \dot{q}_3 \end{bmatrix}}_{x_3} + \begin{bmatrix} 0 \\ -D_3^{-1} g_3 \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ D_3^{-1} \end{bmatrix}}_{B_3} \tau_3. \tag{4.20}$$

We aim to find a matching controller for agent 3 and 4: however, since the reference model signals are not available to this agent, we assume the dynamics of the neighboring agent 1 and 2 to act as a reference model. Let us propose the following controller for UAV 3 to match the follower dynamics based on (4.20) to the dynamics of leader UAV 1 based on (4.8)

$$\tau_3^* = \underbrace{\begin{bmatrix} \bar{K}_{31}^{*\prime} & \bar{\bar{K}}_{31}^{*\prime} \end{bmatrix}}_{K_{31}^{*\prime}} \begin{bmatrix} q_1 \\ \dot{q}_1 \end{bmatrix} + \underbrace{\begin{bmatrix} \bar{K}_3^{*\prime} & \bar{\bar{K}}_3^{*\prime} \end{bmatrix}}_{K_3^{*\prime}} \underbrace{\begin{bmatrix} q_3 - q_1 \\ \dot{q}_3 - \dot{q}_1 \end{bmatrix}}_{e_{31}} + G_3^{*\prime} + L_{31}^{*\prime}\tau_1. \tag{4.21}$$

The following proposition explains how to find the matching control gains in (4.21).

Proposition 4.3.2 *There exists an ideal control law in the form* (4.21) *that matches the follower dynamics* (4.20) *to the leader dynamics* (4.8)*, and whose gains* $\bar{K}_3^*, \bar{\bar{K}}_3^*, \bar{K}_{31}^*, \bar{\bar{K}}_{31}^*, L_{31}^*$*, and* G_3^* *are*

$$\begin{aligned} \bar{K}_3^{*\prime} &= -D_3 K_p & \bar{K}_{31}^{*\prime} &= 0 & G_3^{*\prime} &= g_3 \\ \bar{\bar{K}}_3^{*\prime} &= -D_3 K_v + C_3 & \bar{\bar{K}}_{31}^{*\prime} &= C_3 - D_3 D_1^{-1} C_1 & L_{31}^{*\prime} &= D_3 D_1^{-1}. \end{aligned} \tag{4.22}$$

Proof. By direct substitution of (4.21) into (4.20), we have the leader closed-loop dynamics

$$\begin{aligned} \begin{bmatrix} \dot{q}_3 \\ \ddot{q}_3 \end{bmatrix} = &\begin{bmatrix} 0 & \mathbb{1} \\ D_3^{-1}\bar{K}_3^{*\prime} & D_3^{-1}(\bar{\bar{K}}_3^{*\prime} - C_3) \end{bmatrix} \begin{bmatrix} q_3 \\ \dot{q}_3 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ D_3^{-1}(\bar{K}_{31}^{*\prime} - \bar{K}_3^{*\prime}) & D_3^{-1}(\bar{\bar{K}}_{31}^{*\prime} - \bar{\bar{K}}_3^{*\prime}) \end{bmatrix} \begin{bmatrix} q_1 \\ \dot{q}_1 \end{bmatrix} \\ &+ \begin{bmatrix} 0 \\ -D_3^{-1}(-g_3 + G_3^{*\prime}) \end{bmatrix} + \begin{bmatrix} 0 \\ D_3^{-1} L_{31}^{*\prime} \end{bmatrix} \tau_1 \end{aligned} \tag{4.23}$$

from which we see that matching is achieved for the ideal control law

$$\begin{aligned} \tau_3^* &= C_3 \dot{q}_1 - D_3 D_1^{-1} C_1 \dot{q}_1 - D_3 K_p \bar{e}_{31} - D_3 K_v \bar{\bar{e}}_{31} + C_3 \bar{\bar{e}}_{31} + g_3 + D_3 D_1^{-1}\tau_1 \\ &= C_3 \dot{q}_3 + D_3 D_1^{-1}\tau_1 - D_3 D_1^{-1} C_1 \dot{q}_1 - D_3 (K_p \bar{e}_{31} + K_v \bar{\bar{e}}_{31}) + g_3 \end{aligned} \tag{4.24}$$

where we have defined $\bar{e}_{31} = q_3 - q_1$, $\bar{\bar{e}}_{31} = \dot{q}_3 - \dot{q}_1$. From (4.24) we find the control gains (4.22). This concludes the proof.

Remark 4.3.2 *Differently from Proposition 4.3.1, which gives us matching conditions between an agent and the reference model dynamics, Proposition 4.3.2 gives us matching conditions among neighboring agent. In fact, it is easy to show how* (4.24) *implies the existence of coupling gains* $\bar{K}_{31}^{*\prime}, \bar{\bar{K}}_{31}^{*\prime}, L_{31}^{*\prime}$ *satisfying*

$$\begin{aligned} \bar{K}_{31}^{*\prime} &= \bar{K}_3^{*\prime} - L_{31}^{*\prime}\bar{K}_1^{*\prime} \\ \bar{\bar{K}}_{31}^{*\prime} &= \bar{\bar{K}}_3^{*\prime} - L_{31}^{\prime}\bar{\bar{K}}_1^{*\prime} \\ L_{31}^{*\prime} &= L_3^{*\prime}(L_1^{*\prime})^{-1} \end{aligned} \tag{4.25}$$

where $L_3^* = D_3$*. Therefore, Proposition 4.3.2 can be interpreted as a distributed matching condition among neighboring agents.*

As the system matrices in (4.20) are unknown, the control (4.24) cannot be implemented, and the synchronization task has to be achieved adaptively. Then, inspired by the ideal controller (4.24), we propose the controller

$$\tau_3 = -\underbrace{\Theta'_{D_3}\xi_{D_3}}_{\hat{D}_3}(K_p\bar{e}_{31} + K_v\bar{\bar{e}}_{31}) + \underbrace{\Theta'_{C_3}\xi_{C_3}}_{\hat{C}_3}\dot{q}_3 + \underbrace{\Theta'_{D_3D_1}\xi_{D_3D_1}}_{\widehat{D_3D_1}}\tau_1 - \underbrace{\Theta'_{D_3D_1C_1}\xi_{D_3D_1C_1}}_{\widehat{D_3D_1C_1}}\dot{q}_1 + \underbrace{\Theta'_{g_3}\xi_{g_3}}_{\hat{g}_3} \tag{4.26}$$

where the estimates $\hat{D}_3$, $\hat{C}_3$, $\widehat{D_3D_1}$, $\widehat{D_3D_1C_1}$, $\hat{g}_3$ of the ideal matrices have been split in a linear-in-the-parameter form, in view of Assumption 4.2.3. In fact, Assumption 4.2.3 guarantees $D_3 = \Theta^{*\prime}_{D_3}\xi_{D_3}$, $C_3 = \Theta^{*\prime}_{C_3}\xi_{C_3}$, $g_3 = \Theta^{*\prime}_{g_3}\xi_{g_3}$, $D_3D_1 = \Theta^{*\prime}_{D_3D_1}\xi_{D_3D_1}$ and $D_3D_1C_1 = \Theta^{*\prime}_{D_3D_1C_1}\xi_{D_3D_1C_1}$: again, a specific form of regressand Θ and regressor ξ will be revealed in Sect. 4.4 for the example of UAV. Let us define the error $e_{31} = x_3 - x_1$, whose dynamics are

$$\begin{aligned}
\dot{e}_{31} &= A_m e_{31} + B_3(\tilde{K_3}' e_{31} + \tilde{K_{31}}' x_1 + \tilde{L}'_{31}\tau_1 + \tilde{G}'_3) \\
&= A_m e_{31} + B_3(\tilde{\bar{\bar{K}}}'_3\bar{e}_{31} + \tilde{\bar{\bar{\bar{K}}}}'_3\bar{\bar{e}}_{31} + \tilde{\bar{\bar{K}}}'_{31}q_1 + \tilde{\bar{\bar{\bar{K}}}}'_{31}\dot{q}_1 + \tilde{L}'_{31}\tau_1 + \tilde{G}'_3) \\
&= A_m e_{31} + B_3(\tilde{\Theta}'_{C_3}\xi_{C_3}\dot{q}_3 + \tilde{\Theta}'_{D_3D_1}\xi_{D_3D_1}\tau_1 - \tilde{\Theta}'_{D_3D_1C_1}\xi_{D_3D_1C_1}\dot{q}_1 \\
&\quad - \tilde{\Theta}'_{D_3}\xi_{D_3}(K_p\bar{e}_{31} + K_v\bar{\bar{e}}_{31}) + \tilde{\Theta}'_{g_3}\xi_{g_3})
\end{aligned} \tag{4.27}$$

where $\tilde{K}_3 = K_3 - K^*_3$, $\tilde{K}_{31} = K_{31} - K^*_{31}$, $\tilde{L}_{31} = L_{31} - L^*_{31}$, $\tilde{\Theta}_{D_3} = \Theta_{D_3} - \Theta^*_{D_3}$, $\tilde{\Theta}_{C_3} = \Theta_{C_3} - \Theta^*_{C_3}$, $\tilde{\Theta}_{g_3} = \Theta_{g_3} - \Theta^*_{g_3}$, $\tilde{\Theta}_{D_3D_1} = \Theta_{D_3D_1} - \Theta^*_{D_3D_1}$ and $\tilde{\Theta}_{D_3D_1C_1} = \Theta_{D_3D_1C_1} - \Theta^*_{D_3D_1C_1}$. The following theorem provides the follower-leader synchronization.

Theorem 4.3.2 *Consider the reference model (4.2), the unknown leader dynamics (4.8), the unknown follower dynamics (4.20), and controller (4.26). Provided that there exists a matrix S_3 such that*

$$L^*_3 S_3 = S'_3 L^{*\prime}_3 > 0 \tag{4.28}$$

then, the adaptive laws

$$\begin{aligned}
\dot{\Theta}'_{C_3} &= -S_3B'_mPe_{31}\dot{q}'_3\xi'_{C_3} & \dot{\Theta}'_{D_3} &= S_3B'_mPe_{31}(K_p\bar{e}_{31} + K_v\bar{\bar{e}}_{31})'\xi'_{D_3} \\
\dot{\Theta}'_{D_3D_1} &= -S_3B'_mPe_{31}\tau'_1\xi'_{D_3D_1} & \dot{\Theta}'_{g_3} &= -S_3B'_mPe_{31}\xi'_{g_3} \\
\dot{\Theta}'_{D_3D_1C_1} &= S_3B'_mPe_{31}\dot{q}'_1\xi'_{D_3D_1C_1}
\end{aligned} \tag{4.29}$$

where $P = P' > 0$ is such that (4.17) holds, guarantee synchronization of the follower dynamics (4.20) to the leader dynamics (4.8), i.e. $e_{31} \to 0$.

With reference to Figure 4.2, the control law of UAV 4 is similar as UAV 3.

Proof. To analytically show the asymptotic convergence of the synchronization error between the follower and the leader, let us introduce the following Lyapunov function

$$\begin{aligned}
&V_3(e_{31}, \tilde{\Theta}_{C_3}, \tilde{\Theta}_{D_3D_1}, \tilde{\Theta}_{D_3D_1C_1}, \tilde{\Theta}_{D_3}, \tilde{\Theta}_{g_3}) = \\
&\quad e'_{31}Pe_{31} + \mathrm{tc}(\tilde{\Theta}'_{\mathrm{C_3}}\mathrm{S}_3^{-1}\mathrm{L}_3^{*-1}\tilde{\Theta}_{\mathrm{C_3}}) + \mathrm{tr}(\tilde{\Theta}'_{\mathrm{D_3D_1}}\mathrm{S}_3^{-1}\mathrm{L}_3^{*-1}\tilde{\Theta}_{\mathrm{D_3D_1}}) + \\
&\quad \mathrm{tr}(\tilde{\Theta}'_{\mathrm{D_3D_1C_1}}\mathrm{S}_3^{-1}\mathrm{L}_3^{*-1}\tilde{\Theta}_{\mathrm{D_3D_1C_1}}) + \mathrm{tr}(\tilde{\Theta}'_{\mathrm{D_3}}\mathrm{S}_3^{-1}\mathrm{L}_3^{*-1}\tilde{\Theta}_{\mathrm{D_3}}) + \\
&\quad \mathrm{tr}(\tilde{\Theta}'_{\mathrm{g_3}}\mathrm{S}_3^{-1}\mathrm{L}_3^{*-1}\tilde{\Theta}_{\mathrm{g_3}}).
\end{aligned} \tag{4.30}$$

Then it is possible to verify that

$$\begin{aligned}
\dot{V}_3 = &-e_{31}' Q e_{31} + 3e_{31}' P B_3 (\tilde{\Theta}_{C_3}' \xi_{C_3} \dot{q}_3 + \tilde{\Theta}_{D_3 D_1}' \xi_{D_3 D_1} \tau_1 - \tilde{\Theta}_{D_3 D_1 C_1}' \xi_{D_3 D_1 C_1} \dot{q}_1 \\
&- \tilde{\Theta}_{D_3}' \xi_{D_3} (K_p \bar{e}_{31} + K_v \bar{\bar{e}}_{31}) + \tilde{\Theta}_{g_3}' \xi_{g_3}) + 3\mathrm{tr}(\tilde{\Theta}_{\mathrm{C_3}}' \mathrm{S}_3^{-1} \mathrm{L}_3^{*-1} \dot{\tilde{\Theta}}_{\mathrm{C_3}}) \\
&+ 2\mathrm{tr}(\tilde{\Theta}_{\mathrm{D_3 D_1}}' \mathrm{S}_3^{-1} \mathrm{L}_3^{*-1} \dot{\tilde{\Theta}}_{\mathrm{D_3 D_1}}) + 2\mathrm{tr}(\tilde{\Theta}_{\mathrm{D_3 D_1 C_1}}' \mathrm{S}_3^{-1} \mathrm{L}_3^{*-1} \dot{\tilde{\Theta}}_{\mathrm{D_3 D_1 C_1}}) \\
&+ 2\mathrm{tr}(\tilde{\Theta}_{\mathrm{D_3}}' \mathrm{S}_3^{-1} \mathrm{L}_3^{*-1} \dot{\tilde{\Theta}}_{\mathrm{D_3}}) + 2\mathrm{tr}(\tilde{\Theta}_{\mathrm{g_3}}' \mathrm{S}_3^{-1} \mathrm{L}_3^{*-1} \dot{\tilde{\Theta}}_{\mathrm{g_3}}) \\
= &-e_{31}' Q e_{31} + 2\mathrm{tr}(\tilde{\Theta}_{\mathrm{C_3}}' \mathrm{L}_3^{*-1} (\mathrm{B_m' P e_{31}} \dot{\mathrm{q}}_3' \xi_{\mathrm{C_3}}' + \mathrm{S}_3^{-1} \dot{\tilde{\Theta}}_{\mathrm{C_3}})) \\
&+ 2\mathrm{tr}(\tilde{\Theta}_{\mathrm{D_3 D_1}}' \mathrm{L}_3^{*-1} (\mathrm{B_m' P e_{31}} \tau_1' \xi_{\mathrm{D_3 D_1}}' + \mathrm{S}_3^{-1} \dot{\tilde{\Theta}}_{\mathrm{D_3 D_1}})) \\
&- 2\mathrm{tr}(\tilde{\Theta}_{\mathrm{D_3 D_1 C_1}}' \mathrm{L}_3^{*-1} (\mathrm{B_m' P e_{31}} \dot{\mathrm{q}}_1' \xi_{\mathrm{D_3 D_1 C_1}}' + \mathrm{S}_3^{-1} \dot{\tilde{\Theta}}_{\mathrm{D_3 D_1 C_1}})) \\
&- 2\mathrm{tr}(\tilde{\Theta}_{\mathrm{D_3}}' \mathrm{L}_3^{*-1} (\mathrm{B_m' P e_{31}} (\mathrm{K_p} \bar{\mathrm{e}}_{31} + \mathrm{K_v} \bar{\bar{\mathrm{e}}}_{31})' \xi_{\mathrm{D_3}}' + \mathrm{S}_3^{-1} \dot{\tilde{\Theta}}_{\mathrm{D_3}})) \\
&+ 2\mathrm{tr}(\tilde{\Theta}_{\mathrm{g_3}}' \mathrm{L}_3^{*-1} (\mathrm{B_m' P e_{31}} \xi_{\mathrm{g_3}}' + \mathrm{S}_3^{-1} \dot{\tilde{\Theta}}_{\mathrm{g_3}})) \\
= &-e_{31}' Q e_{31}.
\end{aligned} \tag{4.31}$$

Following similar steps as in the proof of Theorem 4.3.1, from (4.31) we deduce that V_3 has a finite limit, so e_3, $\tilde{\Theta}_{C_3}$, $\tilde{\Theta}_{D_3 D_1}$, $\tilde{\Theta}_{D_3 D_1 C_1}$, $\tilde{\Theta}_{D_3}$, $\tilde{\Theta}_{g_3} \in L_\infty$. Because $e_{31} = x_3 - x_1 \in L_\infty$ and $x_1 \in L_\infty$, we have $x_3 \in L_\infty$. This implies that x_3, $\tilde{\Theta}_{C_3}$, $\tilde{\Theta}_{D_3 D_1}$, $\tilde{\Theta}_{D_3 D_1 C_1}$, $\tilde{\Theta}_{D_3}$, $\tilde{\Theta}_{g_3} \in L_\infty$. Consequently, we can deduce $\tau_3 \in L_\infty$. Therefore, all signals in the closed-loop systems are bounded. From (4.31), we can establish that V_3 has a bounded integral, so that we have $e_{31} \in L_3$. Then by using $\tilde{\Theta}_{C_3}$, $\tilde{\Theta}_{D_3 D_1}$, $\tilde{\Theta}_{D_3 D_1 C_1}$, $\tilde{\Theta}_{D_3}$, $\tilde{\Theta}_{g_3}, e_{31} \in L_\infty$, we have $\dot{e}_{31} \in L_\infty$. This concludes the proof of the boundedness of all closed-loop signal and convergence $e_{31} \to 0$ as $t \to \infty$.

Remark 4.3.3 *In line with Assumptions 4.2.3, the distributed gains in (4.23) should be written in the linear-in-the parameter form: this in general requires some reparametrization or overparameterization (e.g. collecting two or more parameters in a new parameter to be estimated), as shown in the UAV case of Sect. 4.4.*

Remark 4.3.4 *The benefit of the adaptive law is to allow all UAVs to homogenize to the same dynamics, by adapting the control action to compensate for different mass and inertia. In fact, it is well known in formation control literature that homogeneous dynamics are a crucial feature in order to achieve proper coordinated motion [76, 77].*

The proposed algorithm has been illustrated for the formation in Figure 4.2, but it can also be implemented with a different number of leaders and followers: Figure 4.3 shows the communication graph for the Y formation. For such Y formation (3 leaders and 1 follower), the following controller can be proposed:

$$\begin{aligned}
\tau_1 &= \hat{D}_1(-K_p q_1 - K_v \dot{q}_1 + r) + \hat{C}_1 \dot{q}_1 + \hat{g}_1 \\
\tau_2 &= \hat{D}_2(-K_p q_2 - K_v \dot{q}_2 + r) + \hat{C}_2 \dot{q}_2 + \hat{g}_2 \\
\tau_3 &= \hat{D}_3(-K_p q_3 - K_v \dot{q}_3 + r) + \hat{C}_3 \dot{q}_3 + \hat{g}_3 \\
\tau_4 &= -\hat{D}_4[K_p(q_4 - q_3) + K_v(\dot{q}_4 - \dot{q}_3)] + \hat{C}_4 \dot{q}_4 + \widehat{D_4 D_3} \tau_3 - \widehat{D_4 D_3 C_3} \dot{q}_3 + \hat{g}_4
\end{aligned} \tag{4.32}$$

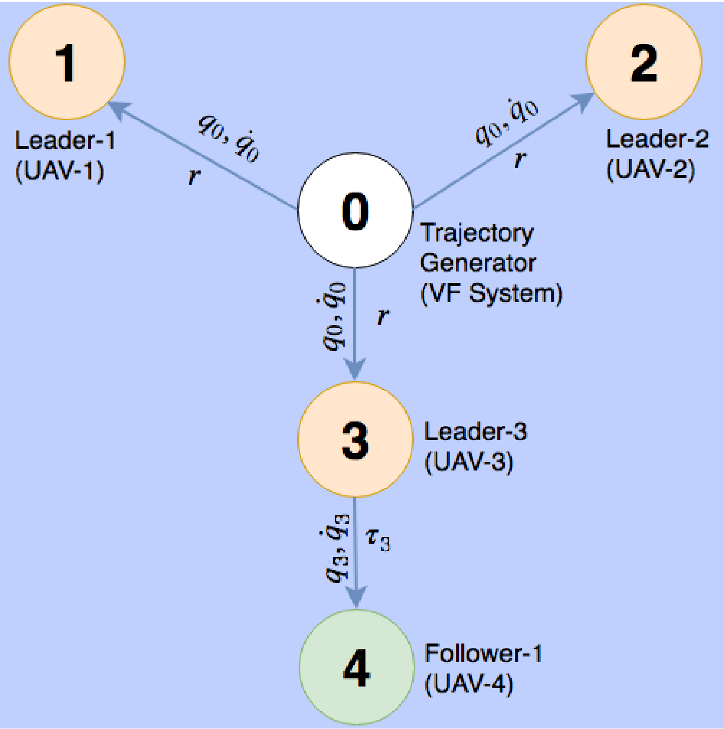


Figure 4.3: Communication graph with Y formation.

with adaptive laws

$$
\begin{aligned}
\dot{\Theta}'_{C1} &= -\Gamma B'_m P\ e_1\ \dot{q}_1{}'\xi'_{C1} \\
\dot{\Theta}'_{g1} &= -\Gamma B'_m P\ e_1\ \xi'_{g1} \\
\dot{\Theta}'_{D1} &= -\Gamma B'_m\ P\ e_1\ (-K_p q_1 - K_v \dot{q}_1 + r)'\xi'_{D1} \\
\dot{\Theta}'_{C2} &= -\Gamma B'_m P\ e_2\ \dot{q}_2{}'\xi'_{C2} \\
\dot{\Theta}'_{g2} &= -\Gamma B'_m P\ e_2\ \xi'_{g2} \\
\dot{\Theta}'_{D2} &= -\Gamma B'_m P\ e_2\ (-K_p q_2 - K_v \dot{q}_2 + r)'\xi'_{D2} \\
\dot{\Theta}'_{C3} &= -\Gamma B'_m P\ e_3\ \dot{q}_3{}'\xi'_{C3} \\
\dot{\Theta}'_{g3} &= -\Gamma B'_m P\ e_3\ \xi'_{g3} \\
\dot{\Theta}'_{D3} &= -\Gamma B'_m P\ e_3\ (-K_p q_3 - K_v \dot{q}_3 + r)'\xi'_{D3} \\
\dot{\Theta}'_{D_4 D_3} &= -\Gamma B'_m\ P\ e_{34}\ \tau'_3\ \xi'_{D_4 D_3} \\
\dot{\Theta}'_{D_4 D_3 C_3} &= -\Gamma B'_m\ P\ e_{34}\ \dot{q}'_3\ \xi'_{D_4 D_3 C_3} \\
\dot{\Theta}'_{C_4} &= -\Gamma B'_m\ P\ e_{34}\ \dot{q}'_4\ \xi'_{C_4} \\
\dot{\Theta}'_{g_4} &= -\Gamma B'_m\ P\ e_{34}\ \xi'_{g_4} \\
\dot{\Theta}'_{D4} &= -\Gamma\ B'_m\ P\ e_{34}\ \left[K_p(q_4 - q_3) + K_v(\dot{q}_4 - \dot{q}_3)'\xi'_{D_4 D_3}\right].
\end{aligned}
\tag{4.33}
$$

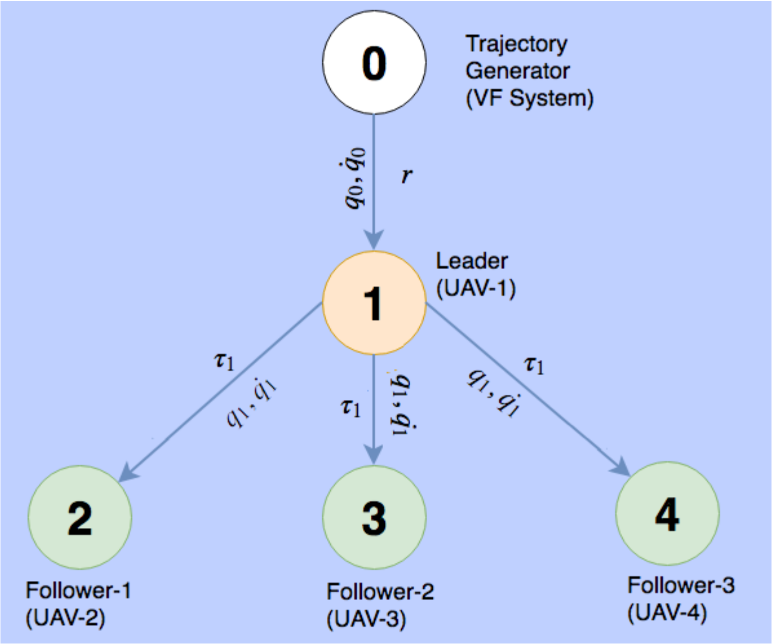


Figure 4.4: Communication graph with T formation.

In other words, the structure of the controller is suitable for many formations: clearly, because each UAVs has different neighbors according to the formation, the signals used to implement the control action will be different.

Another typical formation shape is the inverted T formation as in Figure 4.4 (1 leader and 3 followers). For such T formation, the following controller can be proposed:

$$\begin{aligned}
\tau_1 &= \hat{D}_1(-K_p q_1 - K_v \dot{q}_1 + r) + \hat{C}_1 \dot{q}_1 + \hat{g}_1 \\
\tau_2 &= -\hat{D}_2[K_p(q_2 - q_1) + K_v(\dot{q}_2 - \dot{q}_1)] + \hat{C}_2 \dot{q}_2 + \widehat{D_2 D_1}\tau_1 - \widehat{D_2 D_1 C_1}\dot{q}_1 + \hat{g}_2 \\
\tau_3 &= -\hat{D}_3[K_p(q_3 - q_1) + K_v(\dot{q}_3 - \dot{q}_1)] + \hat{C}_3 \dot{q}_3 + \widehat{D_3 D_1}\tau_1 - \widehat{D_3 D_1 C_1}\dot{q}_1 + \hat{g}_3 \\
\tau_4 &= -\hat{D}_4[K_p(q_4 - q_1) + K_v(\dot{q}_4 - \dot{q}_1)] + \hat{C}_4 \dot{q}_4 + \widehat{D_4 D_1}\tau_1 - \widehat{D_4 D_1 C_1}\dot{q}_1 + \hat{g}_4
\end{aligned} \tag{4.34}$$

with adaptive laws

$$\begin{aligned}
\dot{\Theta}'_{C1} &= -\Gamma B'_m P\ e_1\ \dot{q}_1{}'\xi'_{C1} \\
\dot{\Theta}'_{g1} &= -\Gamma B'_m P\ e_1\ \xi'_{g1} \\
\dot{\Theta}'_{D1} &= -\Gamma B'_m\ P\ e_1\ (-K_p q_1 - K_v \dot{q}_1 + r)'\xi'_{D1} \\
\dot{\Theta}'_{D_2D_1} &= -\Gamma B'_m\ P\ e_{12}\ \tau'_1\ \xi'_{D_2D_1} \\
\dot{\Theta}'_{D_2D_1C_1} &= -\Gamma B'_m\ P\ e_{12}\ \dot{q}'_1\ \xi'_{D_2D_1C_1} \\
\dot{\Theta}'_{C_2} &= -\Gamma B'_m\ P\ e_{12}\ \dot{q}'_2\ \xi'_{C_2} \\
\dot{\Theta}'_{g_2} &= -\Gamma B'_m\ P\ e_{12}\ \xi'_{g_2} \\
\dot{\Theta}'_{D2} &= -\Gamma\ B'_m\ P\ e_{12}\ \left[K_p(q_2 - q_1) + K_v(\dot{q}_2 - \dot{q}_1)'\xi'_{D_2D_1}\right] \qquad (4.35)\\
\dot{\Theta}'_{D_3D_1} &= -\Gamma B'_m\ P\ e_{13}\ \tau'_1\ \xi'_{D_3D_1} \\
\dot{\Theta}'_{D_3D_1C_1} &= -\Gamma B'_m\ P\ e_{13}\ \dot{q}'_1\ \xi'_{D_3D_1C_1} \\
\dot{\Theta}'_{C_3} &= -\Gamma B'_m\ P\ e_{13}\ \dot{q}'_3\ \xi'_{C_3} \\
\dot{\Theta}'_{g_3} &= -\Gamma B'_m\ P\ e_{13}\ \xi'_{g_3} \\
\dot{\Theta}'_{D3} &= -\Gamma\ B'_m\ P\ e_{13}\ \left[K_p(q_3 - q_1) + K_v(\dot{q}_3 - \dot{q}_1)'\xi'_{D_3D_1}\right] \qquad (4.36)\\
\dot{\Theta}'_{D_4D_1} &= -\Gamma B'_m\ P\ e_{14}\ \tau'_1\ \xi'_{D_4D_1} \\
\dot{\Theta}'_{D_4D_1C_1} &= -\Gamma B'_m\ P\ e_{14}\ \dot{q}'_1\ \xi'_{D_4D_1C_1} \\
\dot{\Theta}'_{C_4} &= -\Gamma B'_m\ P\ e_{14}\ \dot{q}'_4\ \xi'_{C_4} \\
\dot{\Theta}'_{g_4} &= -\Gamma B'_m\ P\ e_{14}\ \xi'_{g_4} \\
\dot{\Theta}'_{D4} &= -\Gamma\ B'_m\ P\ e_{14}\ \left[K_p(q_4 - q_1) + K_v(\dot{q}_4 - \dot{q}_1)'\xi'_{D_4D_1}\right]. \qquad (4.37)
\end{aligned}$$

Remark 4.3.5 *The proposed methodology has been presented for specific formations, but it can be extended to any formation with a communication graph that contains a directed acyclic graph, i.e. with a topological ordering. Notice that the presented V, Y and T formations are all directed acyclic graphs. It is not needed that every follower is directly connected to the leader, but it is needed that there exists a possibly multi-hop path that connects each follower to the leader. It is acknowledged that, the proposed method does not directly address issues such as collision avoidance and string stability, which are of practical concern but are often studied using different methods than standard formation control methods. For example, studying collision avoidance requires to include potential fields in the control law, whereas studying string stability requires to study the propagation of disturbances along the formation. These topics provide a motivation for future work.*

4.4. SIMULATION RESULTS

In order to implement the control laws more explicitly, let us derive the control law in the form (4.13) for a UAV indicated by subscript i, and with dynamics as in (2.9): it is possible to show that the linear-in-the-parameter forms for D_i, C_i, g_i are

$$
\begin{gathered}
\Theta^{*}_{D_i} = \begin{bmatrix} m_i & 0 & 0 & 0 & 0 & 0 \\ 0 & m_i & 0 & 0 & 0 & 0 \\ 0 & 0 & m_i & 0 & 0 & 0 \\ 0 & 0 & 0 & I_{x_i} & 0 & -I_{xz_i} \\ 0 & 0 & 0 & 0 & I_{y_i} & 0 \\ 0 & 0 & 0 & -I_{xz_i} & 0 & I_{z_i} \end{bmatrix} \qquad \xi_{D_i} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \\
\Theta^{*\prime}_{C_i} = \begin{bmatrix} m_i & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & & & & & \\ 0 & m_i & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & & & & & \\ 0 & 0 & m_i & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & & & & & \\ 0 & 0 & 0 & I_{x_i} & 0 & 0 & I_{y_i} & 0 & 0 & I_{z_i} \\ 0 & 0 & I_{xz_i} & 0 & 0 & & & & & \\ 0 & 0 & 0 & 0 & I_{x_i} & 0 & 0 & I_{y_i} & 0 & 0 \\ I_{z_i} & 0 & 0 & I_{xz_i} & 0 & & & & & \\ 0 & 0 & 0 & 0 & 0 & I_{x_i} & 0 & 0 & I_{y_i} & 0 \\ 0 & I_{z_i} & 0 & 0 & I_{xz_i} & & & & & \end{bmatrix} \\
\xi^{\prime}_{C_i} = \begin{bmatrix} 0 & \bar{r}_i & -\bar{q}_i & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & & & & & \\ -\bar{r}_i & 0 & \bar{p}_i & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & & & & & \\ \bar{q}_i & -\bar{p}_i & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & & & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \bar{q}_i & 0 \\ -\bar{r}_i & 0 & 0 & \bar{p}_i & 0 & & & & & \\ 0 & 0 & 0 & 0 & 0 & -\bar{p}_i & 0 & 0 & 0 & \bar{r}_i \\ 0 & 0 & -\bar{p}_i & 0 & \bar{r}_i & & & & & \\ 0 & 0 & 0 & 0 & \bar{p}_i & 0 & -\bar{q}_i & 0 & 0 & 0 \\ 0 & 0 & 0 & -\bar{r}_i & 0 & & & & & \end{bmatrix} \\
\Theta^{*\prime}_{g_i} = \begin{bmatrix} 0 \\ 0 \\ m_i \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad \xi_{gi} = g.
\end{gathered} \tag{4.38}
$$

Table 4.2: Fixed-wing UAVs parameters

	Mass (kg)	Moment of Inertia (kgm^2)
UAV-0 (Pinner)	10	$I_x = 0.02,\ I_y = 0.026$ $I_z = 0.053,\ I_{xz} = 0.01$
UAV-1 (Leader 1)	20	$I_x = 0.1,\ I_y = 0.05$ $I_z = 0.1,\ I_{xz} = 0.01$
UAV-2 (Follower 1)	30	$I_x = 0.2,\ I_y = 0.1$ $I_z = 0.2,\ I_{xz} = 0.02$
UAV-3 (Leader 2)	40	$I_x = 0.4,\ I_y = 0.02$ $I_z = 0.4,\ I_{xz} = 0.04$
UAV-4 (Follower 2)	50	$I_x = 0.8,\ I_y = 0.04$ $I_z = 0.08,\ I_{xz} = 0.08$

Next, we derive the control law in the form (4.26), for two neighboring UAVs, indicated by the subscripts i and j: it is not difficult to show that the linear-in-the-parameter forms of D_jD_i and $D_jD_iC_i$ are

$$\Theta^*_{D_jD_i} = \begin{bmatrix} \frac{m_j}{m_i} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{m_j}{m_i} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{m_j}{m_i} & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{I_{z_i} I_{x_j} I_{xz_j}}{I_{x_i} I_{z_i} - I_{xz_i} I_{xz_i}} & 0 & \frac{I_{xz_i} I_{x_j} I_{xz_j}}{I_{x_i} I_{z_i} - I_{xz_i} I_{xz_i}} \\ 0 & 0 & 0 & 0 & \frac{I_{y_j}}{I_{y_i}} & 0 \\ I_{x_i} 0 & 0 & 0 & \frac{I_{xz_i} I_{xz_j} I_{z_j}}{I_{x_i} I_{z_i} - I_{xz_i} I_{xz_i}} & 0 & -\frac{I_{x_i} I_{xz_j} I_{z_j}}{I_{x_i} I_{z_i} - I I_{xz_i} I_{xz_i}} \end{bmatrix}$$

$$\xi_{D_jD_i} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad (4.39)$$

$$
\Theta^{*}_{D_jD_iC_i} = \begin{bmatrix}
m_j & 0 & 0 & 0 & 0 & 0 \\
0 & m_j & 0 & 0 & 0 & 0 \\
0 & 0 & m_j & 0 & 0 & 0 \\
0 & 0 & 0 & \Gamma_1 & 0 & 0 \\
0 & 0 & 0 & 0 & \Gamma_1 & 0 \\
0 & 0 & 0 & 0 & 0 & \Gamma_1 \\
0 & 0 & 0 & \Gamma_2 & 0 & 0 \\
0 & 0 & 0 & 0 & \Gamma_2 & 0 \\
0 & 0 & 0 & 0 & 0 & \Gamma_2 \\
0 & 0 & 0 & \Gamma_3 & 0 & 0 \\
0 & 0 & 0 & 0 & \Gamma_3 & 0 \\
0 & 0 & 0 & 0 & 0 & \Gamma_3 \\
0 & 0 & 0 & \Gamma_4 & 0 & 0 \\
0 & 0 & 0 & 0 & \Gamma_4 & 0 \\
0 & 0 & 0 & 0 & 0 & \Gamma_4 \\
0 & 0 & 0 & \Gamma_5 & 0 & 0 \\
0 & 0 & 0 & 0 & \Gamma_5 & 0 \\
0 & 0 & 0 & 0 & 0 & \Gamma_5 \\
0 & 0 & 0 & \Gamma_6 & 0 & 0 \\
0 & 0 & 0 & 0 & \Gamma_6 & 0 \\
0 & 0 & 0 & 0 & 0 & \Gamma_6 \\
0 & 0 & 0 & \Gamma_7 & 0 & 0 \\
0 & 0 & 0 & 0 & \Gamma_7 & 0 \\
0 & 0 & 0 & 0 & 0 & \Gamma_7 \\
0 & 0 & 0 & \Gamma_8 & 0 & 0 \\
0 & 0 & 0 & 0 & \Gamma_8 & 0 \\
0 & 0 & 0 & 0 & 0 & \Gamma_8 \\
0 & 0 & 0 & \Gamma_9 & 0 & 0 \\
0 & 0 & 0 & 0 & \Gamma_9 & 0 \\
0 & 0 & 0 & 0 & 0 & \Gamma_9 \\
0 & 0 & 0 & \Gamma_{10} & 0 & 0 \\
0 & 0 & 0 & 0 & \Gamma_{10} & 0 \\
0 & 0 & 0 & 0 & 0 & \Gamma_{10} \\
0 & 0 & 0 & \Gamma_{11} & 0 & 0 \\
0 & 0 & 0 & 0 & \Gamma_{11} & 0 \\
0 & 0 & 0 & 0 & 0 & \Gamma_{11} \\
0 & 0 & 0 & \Gamma_{12} & 0 & 0 \\
0 & 0 & 0 & 0 & \Gamma_{12} & 0 \\
0 & 0 & 0 & 0 & 0 & \Gamma_{12}
\end{bmatrix}
\xi_{D_jD_iC_i} = \begin{bmatrix}
0 & -\bar{r}_i & \bar{q}_i & 0 & 0 & 0 \\
\bar{r}_i & 0 & -\bar{p}_i & 0 & 0 & 0 \\
-\bar{q}_i & \bar{p}_i & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -\bar{q}_i & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -\bar{r}_i & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \bar{p}_i & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \bar{q}_i & 0 & 0 \\
0 & 0 & 0 & 0 & \bar{r}_i & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -\bar{p}_i & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \bar{r}_i & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -\bar{p}_i & 0 \\
0 & 0 & 0 & 0 & 0 & -\bar{q}_i \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -\bar{r}_i \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \bar{p}_i \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \bar{q}_i
\end{bmatrix}
\tag{4.40}
$$

where

$$D_jD_i^{-1} = \begin{bmatrix} \frac{m_j}{m_i} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{m_j}{m_i} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{m_j}{m_i} & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{I_{z_i}I_{x_j}I_{xz_j}}{I_{x_i}I_{z_i}-I_{xz_i}I_{xz_i}} & 0 & \frac{I_{xz_i}I_{x_j}I_{xz_j}}{I_{x_i}I_{z_i}-I_{xz_i}I_{xz_i}} \\ 0 & 0 & 0 & 0 & \frac{I_{y_j}}{I_{y_i}} & 0 \\ 0 & 0 & 0 & \frac{I_{xz_i}I_{xz_j}I_{z_j}}{I_{x_i}I_{z_i}-I_{xz_i}I_{xz_i}} & 0 & -\frac{I_{x_i}I_{xz_j}I_{z_j}}{I_{x_i}I_{z_i}-I_{xz_i}I_{xz_i}} \end{bmatrix} \tag{4.41}$$

$$D_jD_i^{-1}C_i = \begin{bmatrix} 0 & -m_j\bar{r}_i & m_j\bar{q}_i & 0 & 0 & 0 \\ m_j\bar{r}_i & 0 & -m_j\bar{p}_i & 0 & 0 & 0 \\ -m_j\bar{q}_i & m_j\bar{p}_i & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \bar{q}_i\Gamma_1 & \bar{r}_i\Gamma_5 - \bar{p}_i\Gamma_6 & -\bar{q}_i\Gamma_9 \\ 0 & 0 & 0 & -\bar{r}_i\Gamma_2 + \bar{p}_i\Gamma_3 & 0 & -\bar{r}_i\Gamma_{10} + \bar{p}_i\Gamma_{11} \\ 0 & 0 & 0 & \bar{q}_i\Gamma_4 & \bar{r}_i\Gamma_7 - \bar{p}_i\Gamma_8 & \bar{q}_i\Gamma_{12} \end{bmatrix} \tag{4.42}$$

$$\begin{aligned}
\Gamma_1 &= \frac{I_{y_i}(I_{xz_j}I_{x_i} - I_{xz_i}I_{x_j})}{I_{xz_i}^2 - I_{x_i}I_{z_i}} & \Gamma_7 &= \frac{(I_{xz_j}I_{xz_i} - I_{x_i}I_{z_j})I_{xz_i} - (I_{z_j}I_{xz_i} - I_{z_i}I_{xz_j})I_{z_i}}{I_{xz_i}^2 - I_{x_i}I_{z_i}} \\
\Gamma_2 &= \frac{I_{y_j}I_{z_i}}{I_{y_i}} & \Gamma_8 &= \frac{(I_{xz_j}I_{z_i} - I_{x_i}I_{z_j})I_{x_i} - (I_{z_j}I_{xz_i} - I_{z_i}I_{xz_j})I_{xz_i}}{I_{xz_i}^2 - I_{x_i}I_{z_i}} \\
\Gamma_3 &= \frac{I_{y_j}I_{xz_i}}{I_{y_i}} & \Gamma_9 &= \frac{I_{y_i}(I_{xz_j}I_{xz_i} - I_{z_i}I_{x_j})}{I_{xz_i}^2 - I_{x_i}I_{z_i}} \\
\Gamma_4 &= \frac{I_{y_i}(I_{xz_j}I_{xz_i} - I_{x_i}I_{z_j})}{I_{xz_i}^2 - I_{x_i}I_{z_i}} & \Gamma_{10} &= \frac{I_{y_j}I_{xz_i}}{I_{y_i}} \\
\Gamma_5 &= \frac{(I_{xz_j}I_{x_i} - I_{xz_i}I_{x_j})I_{xz_i} + (I_{xz_j}I_{xz_i} - I_{z_i}I_{x_j})I_{z_i}}{I_{xz_i}^2 - I_{x_i}I_{z_i}} & \Gamma_{11} &= \frac{I_{y_j}I_{x_i}}{I_{y_i}} \\
\Gamma_6 &= \frac{(I_{xz_j}I_{x_i} - I_{xz_i}I_{x_j})I_{x_i} + (I_{xz_j}I_{xz_i} - I_{z_i}I_{x_j})I_{xz_i}}{I_{xz_i}^2 - I_{x_i}I_{z_i}} & \Gamma_{12} &= \frac{I_{y_i}(I_{z_j}I_{xz_i} - I_{z_i}I_{xz_j})}{I_{xz_i}^2 - I_{x_i}I_{z_i}}.
\end{aligned} \tag{4.43}$$

In the simulations we will consider the following parameters: constant airspeed $V_a = 15$ m/s, constant altitude $h_m = 50\ m$. The control parameters of the vector field approach are $\kappa_{sl} = \kappa_o = \frac{\pi}{2}$, $k_{sl} = k_o = 0.1$, $\epsilon_{sl} = \epsilon_o = 1$, while the control parameters of the adaptive formation algorithm are

$$Q = 100\mathbb{1}, \quad K_p = 50, \quad K_v = 50, \quad S_i = 100 \tag{4.44}$$

In line with most UAV path generation approaches, the path is composed of straight-lines and orbits. For these simulations we take a path consisting of a straight-line followed by an orbit.

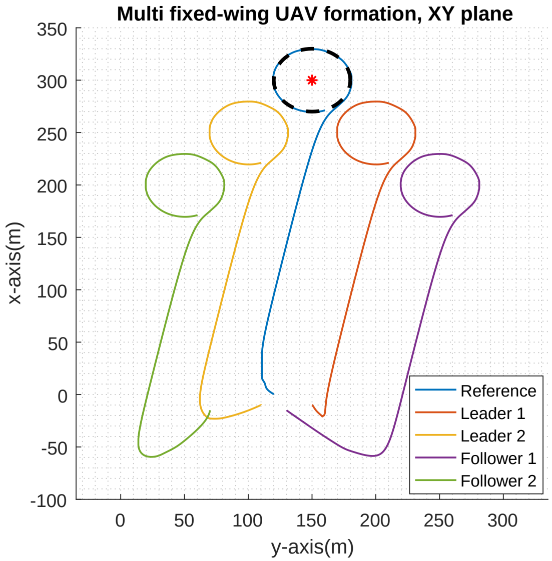


Figure 4.5: Path following with V formation. The UAVs in the formation follow a line and then orbit around a point.

Figure 4.5 shows the result of the simulations for an inverted V formation as in Figure 4.2. It can be noted that the formation control task is achieved despite uncertainty, which demonstrates the effectiveness of the proposed formation control method.

Figure 4.6 shows the result of the simulations for a Y formation (3 leaders and 1 follower), while Figure 4.7 shows the result of the simulations for an inverted T formation (1 leader and 3 followers) with control law.

Finally, we would like to highlight the relevance of embedding adaptation in formation control by showing what happens in the absence of such adaptation. To this purpose, we set up another simulation with an inverted V formation in which two UAVs (Leader 2 and Follower 2) adopt the adaptive algorithm, whereas the other two (Leader 1 and Follower 1) do not employ adaptation. This means that their control gains are kept fixed without adapting to different masses/inertias. Figure 4.8 shows the result of such simulation: it can be seen that the two UAVs not employing adaptation cannot close the gap with respect to their predecessor and they eventually leave the formation. It can be noted from Table 4.2 that the masses of the UAVs vary of a factor 5, whereas the inertias vary with a factor 10: it is remarkable that a unique algorithm can adapt to such heterogeneity. In the absence of such adaptation, it might be difficult to find a formation control strategy that can work for any inertia and any mass. Therefore, the proposed simulations show the capability of achieving different UAV formations while handling uncertain masses and inertias.

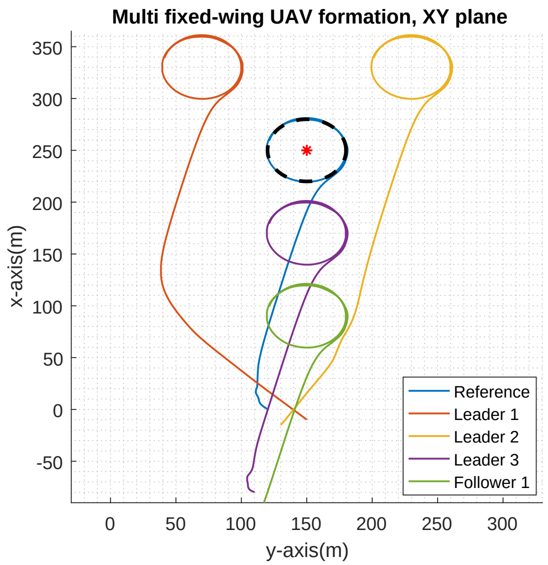


Figure 4.6: Path following with Y formation. The UAVs in the formation follow a line and then orbit around a point.

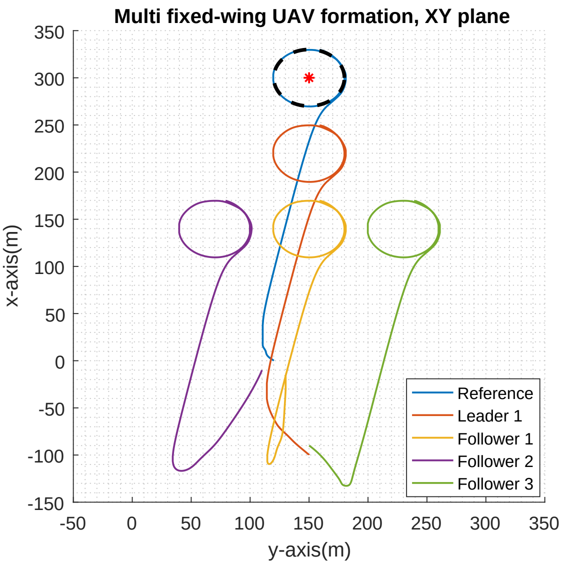


Figure 4.7: Path following with T formation. The UAVs in the formation follow a line and then orbit around a point.

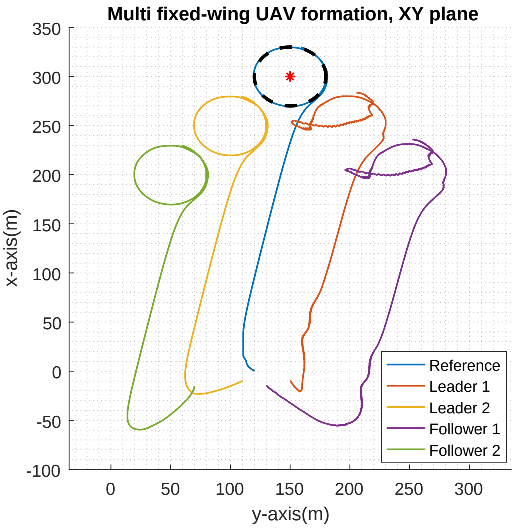


Figure 4.8: Unsuccessful path following in the absence of adaptation. Leader 2 and Follower 2, employing the adaptive algorithm, manage to achieve their part of the formation, while Leader 1 and Follower 1, which do not employ adaptation, leave the formation.

4.5. SUMMARY

The chapter has discussed the possibility to synchronize fixed-wing UAVs formation with Euler-Lagrange dynamics, in the presence of parametric uncertainties represented by uncertain mass and inertia. The formation control problem was defined as a synchronization problem where a distributed model reference adaptive control was used to synchronize the Euler-Lagrange systems. The idea behind the proposed adaptive algorithm was to make each agent converge to the model defined by its neighbors. The presence of uncertainty was handled first by showing that distributed nonlinear matching gains exist between neighboring agents, and then by developing adaptive laws to estimate these gains. The stability of the proposed controlled was derived analytically by introducing an appropriately defined distributed Lyapunov function. The effectiveness of the proposed methodology has been verified via simulation of fixed-wing UAVs in MATLAB environments.

5

A FIXED-WING UAV FORMATION ALGORITHM BASED ON VECTOR FIELD GUIDANCE

5.1. INTRODUCTION

The problem of UAV guidance has been examined in the literature from different points of view and recent surveys include [6, 78, 79]. However, despite the progress in guidance for fixed-wing UAVs, up to now there are still limited or no established guidance tools for achieving formation tasks. To support this observation, notice that despite the several commercial or open-source autopilot suites developed for single UAVs (ArduPilot, PX4, DJI, NAVIO2, AscTec Trinity, etc.) none of them yet provides formation control functionalities. For a single fixed-wing UAV, the vector field method has become a well-established method for guidance towards a desired path. As discussed in Chapters 2 and 3, the vector field idea is based on generating a field of desired course inputs that become the reference input to the inner-loop attitude control laws. Extensions to the vector field idea include tracking targets [31], tracking general curved paths [80], tackling unmodeled course angle dynamics [48], or removing singularities in the vector field that prevent to achieve global convergence [81].

Motivated by these and other advances, the vector field method has gained popularity and studies have been made to extend it towards formation tasks: examples include circular formations with constant speed [82] or, when the velocity can be controlled, a non-uniform vector field whose vectors have different directions and magnitudes [83]. In this sense, the vector field method offers an alternative to formation-keeping methods based on PID control [84], inverse optimal control [85], Nash equilibrium [86, 87], model predictive control [88, 89], or consensus-based formation control [90–93]. It is worth mentioning that, while the vector field method has been originally developed for fixed-wing UAVs, not all the alternative methods are directly applicable to fixed-wing UAVs: consensus- and Nash equilibrium-based methods have been studied mostly for quadrotors [94] or for vertical takeoff and landing UAVs [95]; methods based on inverse optimal control or model predictive control require a different architecture than the established open-source architectures of many autopilot suites (e.g. ArduPilot, PX4, NAVIO2).

Vector-field-based formation control offers the appealing possibility of being integrated with autopilot suites for fixed-wing UAVs, which may be not straightforward with other approaches. For example, formation control laws based on the consensus approach (including the one presented in Chapter 4 of this thesis) are designed to deliver forces and torques (to affect velocities and angular rates). This is due to the fact that the dynamics of the UAV are derived from Newton's law as double integrators or similar fully-actuated dynamics (this approach is used for quadrotors [96–98] and sometimes also for fixed-wing UAVs [99]). However, the typical control architecture of most of the aforementioned autopilot software suites is the successive loop closure, where inner loops are used to provide thrust and actuator deflections, while the guidance law (outer loop) is in charge of providing desired the velocity and the desired course [16, 100]. In some cases, the guidance law provides the velocity and the angular rate [101].

The approach that we take in this chapter is to consider that the autopilot is synthesized according to the successive loop closure architecture: this is in line with standard literature [102, 103] and standard books [16]. We propose a non-uniform vector field that changes in both magnitude and direction: the main contributions of this Chapter as compared to related work are:

- Recent literature has proposed the use of backstepping control to implement the vector field for formation tasks [102, 103]: however, with this method an assumption on the absence of wind is required (see next item). In this chapter, in place of backstepping, we adopt a philosophy that is closer to the originally proposed vector field approach, which relies on a sliding mode control method;

- In related literature, the absence of wind was assumed so that the UAV ground speed coincides with its air speed and the UAV course angle coincides with its heading angle. This setting is restrictive and does not hold in practice. By adopting a philosophy closer to the originally proposed vector field approach, we do not require zero wind speed: sliding-mode-based control commands are designed under the scenario that the air speed can be different from the ground speed, and the heading angle can be different from the course angle.

- Besides analyzing the stability of the proposed method, comparative experiments are provided in both software-in-the-loop and hardware-in-the-loop environments, using PX4 as software and hardware.

Comparisons with standard formation algorithms show that the proposed formation method is effective in in different path scenarios, even with high-latency communication (2Hz).

The rest of the chapter is organized as follows: The control problem is formulated in Sect. 5.2. In Sect. 5.3, we introduce the error dynamics of the proposed method in detail, and the stability analysis based on Lyapunov theory. Software-in-the-loop and hardware-in-the-loop experiments are presented in Sect. 5.4: the method is implemented using PX4 as autopilot suite. Concluding remarks are in Sect. 5.5.

5.2. PROBLEM FORMULATION

This section explains the formation control problem based on the vector field method. To this purpose, consider the scenario in Figure 5.1 with a leader and a follower UAV (indexed with the subscripts "l" and "f"). Their positions in the inertial frame are $(x_\mathrm{l}, y_\mathrm{l})$ and $(x_\mathrm{f}, y_\mathrm{f})$. The vectors V_gl and V_gf are their ground speeds. Note from Figure 5.1 that the position to be tracked by the follower UAV is shifted by a vector that represents the desired formation gap. We refer to Table 5.1 for a list of symbols used in this work.

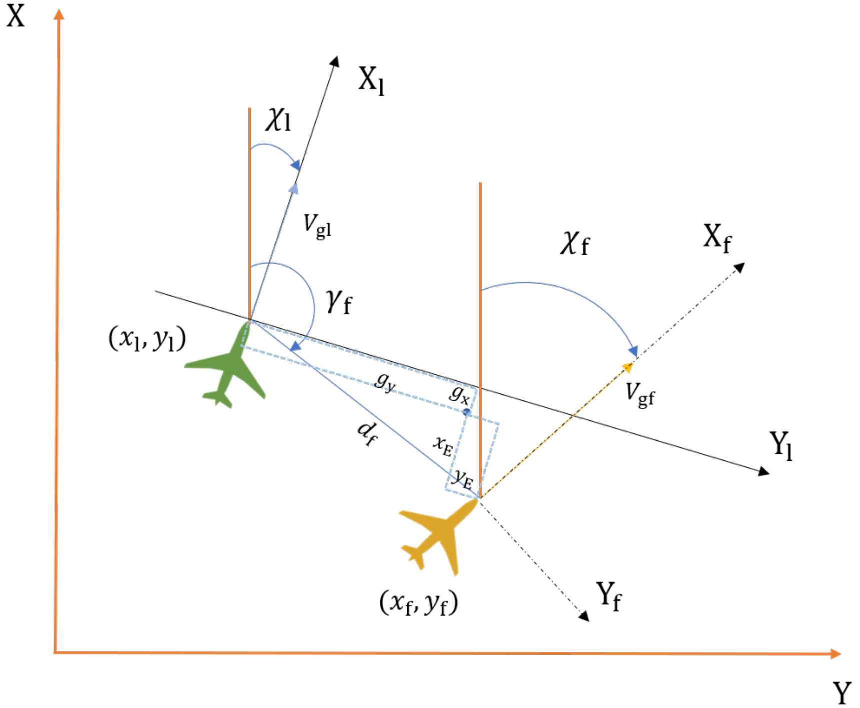


Figure 5.1: Coordinates for formation error.

According to Sect. 2.4, the dynamics of the leader are represented as:

$$\begin{aligned} \dot{x}_\mathrm{l} &= V_\mathrm{gl}\cos(\chi_\mathrm{l}) \\ \dot{y}_\mathrm{l} &= V_\mathrm{gl}\sin(\chi_\mathrm{l}) \\ \dot{\chi}_\mathrm{l} &= \alpha_l(\chi_\mathrm{l}^\mathrm{c} - \chi_\mathrm{l}) \end{aligned} \tag{5.1}$$

with $\chi_\mathrm{l}^\mathrm{c}$ resulting from the standard vector field (2.24)

$$\chi^c = \chi - \chi_\infty \frac{2}{\pi} \frac{\beta_s V_g}{\alpha} \sin(\chi) - \frac{\kappa}{\alpha} \mathrm{sat}\left(\frac{\tilde{\chi}}{\varepsilon}\right)$$

Table 5.1: Notation

Symbol	Description	Unit
$(x_\mathrm{l}, y_\mathrm{l})$, $(x_\mathrm{f}, y_\mathrm{f})$	Coordinates of UAV$_\mathrm{l}$ and UAV$_\mathrm{l}$ in the inertial frame	m
V_gl, V_gf	Ground speed	m/s
χ_l, χ_f	Course angle	rad
V_gl^d, V_gf^d	Desired speed	m/s
$\chi_\mathrm{l}^\mathrm{d}$, $\chi_\mathrm{f}^\mathrm{d}$	Desired course angle	rad
V_gl^c, V_gf^c	Speed command	m/s
χ_l^c, χ_f^c	Course angle command	rad
α_l, α_f	Inverse time constant of course-hold loop	s^{-1}
β_l, β_f	Inverse time constant of velocity-hold loop	s^{-1}
(g_x, g_y)	Desired gap in the leader's body frame	m
d_f	Distance of UAV$_\mathrm{f}$ from UAV$_\mathrm{l}$	m
γ_f	Angle between North and UAV$_\mathrm{f}$ with respect to the leader position	rad
$(x_\mathrm{E}, y_\mathrm{E})$	Formation error in the leader's body frame	m

Accordingly, the dynamics of the follower UAV are

$$\begin{aligned} \dot{x}_\mathrm{f} &= V_\mathrm{gf}\cos(\chi_\mathrm{f}) \\ \dot{y}_\mathrm{f} &= V_\mathrm{gf}\sin(\chi_\mathrm{f}) \\ \dot{\chi}_\mathrm{f} &= \alpha_\mathrm{f}(\chi_\mathrm{f}^\mathrm{c} - \chi_\mathrm{f}) \\ \dot{V}_\mathrm{gf} &= \beta_\mathrm{f}(V_\mathrm{gf}^\mathrm{c} - V_\mathrm{gf}). \end{aligned} \tag{5.2}$$

The main difference between (5.2) and (5.1) is given by the last equation, which represents a velocity-hold loop, where β_f is the inverse of the time constant of the velocity-hold loop, and V_gf^c the speed command. In other words, we allow the follower to change its velocity V_gf in such a way as to achieve the formation. In several open-source autopilot suites (e.g. ArduPilot, PX4, NAVIO2) commanding the velocity is possible thanks to the Total Energy Control System (TECS), which controls velocity and altitude. The interested reader is referred to standard literature [6, 7, 16, 35, 36] describing how dynamics (5.2) can be obtained.

Instead of defining the formation error in the inertial frame, it is more convenient to express it in the leader's frame. Therefore, let (g_x, g_y) be the desired formation gap expressed in the leader's frame. In other words, the leader's frame represents the reference frame for the follower, which calculates the formation error in this frame as:

$$\begin{aligned} x_\mathrm{E} &= g_x + d_f \sin\left(\gamma_\mathrm{f} - \frac{\pi}{2} - \chi_\mathrm{l}\right) \\ y_\mathrm{E} &= g_y + d_f \cos\left(\gamma_\mathrm{f} - \frac{\pi}{2} - \chi_\mathrm{l}\right) \end{aligned} \tag{5.3}$$

where d_f is the distance of the follower UAV from the leader UAV, and γ_f is the angle between the North and the follower UAV position with respect to the leader position. The formation control problem can be formulated as:

Control Problem: Given the leader dynamics (5.1) and the follower dynamics (5.2), design the follower commands $\chi_{\mathrm{f}}^{\mathrm{c}}$, $V_{\mathrm{gf}}^{\mathrm{c}}$ such that the errors x_{E}, y_{E}, $\chi_{\mathrm{f}} - \chi_{\mathrm{f}}^{\mathrm{d}}$ and $V_{\mathrm{gf}} - V_{\mathrm{gf}}^{d}$ asymptotically converge to zero. Here, $\chi_{\mathrm{f}}^{\mathrm{d}}$, and $V_{\mathrm{gf}}^{\mathrm{d}}$ are vector fields to be designed appropriately.

5.3. VECTOR FIELD FOR FORMATION CONTROL

5.3.1. ERROR DYNAMICS

Before designing the formation guidance control law, we need to derive the error dynamics of the leader-follower system. These dynamics are useful to prove stability of the proposed approach. To describe the position of the follower with respect to the leader we use

$$\begin{aligned} x_{\mathrm{f}} &= x_{\mathrm{l}} + d_{\mathrm{f}}\cos(\gamma_{\mathrm{f}}) \\ y_{\mathrm{f}} &= y_{\mathrm{l}} + d_{\mathrm{f}}\sin(\gamma_{\mathrm{f}}) \end{aligned} \tag{5.4}$$

where d_{f} is the distance of the follower UAV from the leader UAV, and γ_{f} is the angle between the north and the follower UAV position with respect to the leader position.

We can now obtain the dynamics for γ_{f} and d_{f} in these coordinates by taking the derivative of (5.4) and using the dynamics of the follower in (5.2):

$$\begin{aligned} V_{\mathrm{gf}}\cos(\chi_{\mathrm{f}}) &= \dot{x}_{\mathrm{l}} + \dot{d}_{\mathrm{f}}\cos(\gamma_{\mathrm{f}}) - d_{\mathrm{f}}\sin(\gamma_{\mathrm{f}})\dot{\gamma}_{\mathrm{f}} \\ V_{\mathrm{gf}}\sin(\chi_{\mathrm{f}}) &= \dot{y}_{\mathrm{l}} + \dot{d}_{\mathrm{f}}\sin(\gamma_{\mathrm{f}}) + d_{\mathrm{f}}\cos(\gamma_{\mathrm{f}})\dot{\gamma}_{\mathrm{f}}. \end{aligned} \tag{5.5}$$

To obtain the dynamics $\dot{d}_{\mathrm{f}}$, we multiply the first equation in (5.5) by $\cos(\gamma_{\mathrm{f}})$ and the second equation by $\sin(\gamma_{\mathrm{f}})$. Next, we sum the two resulting equations:

$$\begin{aligned} V_{\mathrm{gf}}\cos(\chi_{\mathrm{f}} - \gamma_{\mathrm{f}}) &= \dot{x}_{\mathrm{l}}\cos(\gamma_{\mathrm{f}}) + \dot{y}_{\mathrm{l}}\sin(\gamma_{\mathrm{f}}) + \dot{d}_{\mathrm{f}} \\ \dot{d}_{\mathrm{f}} &= V_{\mathrm{gf}}\cos(\chi_{\mathrm{f}} - \gamma_{\mathrm{f}}) - \dot{x}_{\mathrm{l}}\cos(\gamma_{\mathrm{f}}) - \dot{y}_{\mathrm{l}}\sin(\gamma_{\mathrm{f}}). \end{aligned} \tag{5.6}$$

Substituting $\dot{x}_{\mathrm{l}}$ and $\dot{y}_{\mathrm{l}}$ in (5.6) with the leader dynamics (5.1) gives:

$$\begin{aligned} \dot{d}_{\mathrm{f}} &= V_{\mathrm{gf}}\cos(\chi_{\mathrm{f}} - \gamma_{\mathrm{f}}) - V_{\mathrm{gl}}\cos(\chi_{\mathrm{l}})\cos(\gamma_{\mathrm{f}}) - V_{\mathrm{gl}}\sin(\chi_{\mathrm{l}})\sin(\gamma_{\mathrm{f}}) \\ &= V_{\mathrm{gf}}\cos(\chi_{\mathrm{f}} - \gamma_{\mathrm{f}}) - V_{\mathrm{gl}}\cos(\chi_{\mathrm{l}} - \gamma_{\mathrm{f}}). \end{aligned} \tag{5.7}$$

In order to obtain $\dot{\gamma}_{\mathrm{f}}$ we take (5.5) and multiply the first equation by $\sin(\gamma_{\mathrm{f}})$ and the second equation by $\cos(\gamma_{\mathrm{f}})$. Then, we subtract the first equation from the second so as to obtain

$$\begin{aligned} V_{\mathrm{gf}}\sin(\chi_{\mathrm{f}} - \gamma_{\mathrm{f}}) &= -\dot{x}_{\mathrm{l}}\sin(\gamma_{\mathrm{f}}) + \dot{y}_{\mathrm{l}}\cos(\gamma_{\mathrm{f}}) + d_{\mathrm{f}}\dot{\gamma}_{\mathrm{f}} \\ \dot{\gamma}_{\mathrm{f}} &= \frac{V_{\mathrm{gf}}}{d_{\mathrm{f}}}\sin(\chi_{\mathrm{f}} - \gamma_{\mathrm{f}}) + \frac{\dot{x}_{\mathrm{l}}}{d_{\mathrm{f}}}\sin(\gamma_{\mathrm{f}}) - \frac{\dot{y}_{\mathrm{l}}}{d_{\mathrm{f}}}\cos(\gamma_{\mathrm{f}}) \\ &= \frac{V_{\mathrm{gf}}}{d_{\mathrm{f}}}\sin(\chi_{\mathrm{f}} - \gamma_{\mathrm{f}}) - \frac{V_{\mathrm{gl}}}{d_{\mathrm{f}}}\sin(\chi_{\mathrm{l}} - \gamma_{\mathrm{f}}). \end{aligned} \tag{5.8}$$

Using (5.3) we finally derive the error dynamics

$$\begin{aligned}\dot{x}_{\mathrm{E}} &= \dot{d}_{\mathrm{f}}\sin(\gamma_{\mathrm{f}}-\frac{\pi}{2}-\chi_{\mathrm{l}})+d_{\mathrm{f}}\cos(\gamma_{\mathrm{f}}-\frac{\pi}{2}-\chi_{\mathrm{l}})\dot{\gamma}_{\mathrm{f}}\\ &= V_{\mathrm{gf}}\sin(\chi_{\mathrm{f}}-\frac{\pi}{2}-\chi_{\mathrm{l}})+V_{\mathrm{gl}}\\ \dot{y}_{\mathrm{E}} &= \dot{d}_{\mathrm{f}}\cos(\gamma_{\mathrm{f}}-\frac{\pi}{2}-\chi_{\mathrm{l}})-d_{\mathrm{f}}\sin(\gamma_{\mathrm{f}}-\frac{\pi}{2}-\chi_{\mathrm{l}})\\ &= V_{\mathrm{gf}}\cos(\chi_{\mathrm{f}}-\frac{\pi}{2}-\chi_{\mathrm{l}}).\end{aligned} \tag{5.9}$$

The error dynamics depend on the difference between the course of the leader and the follower, the ground speed of the follower, and the ground speed of the leader, as one could intuitively expect. An alternative way to arrive at (5.9) is by using polar coordinates: however, this would require to introduce extra notation that might decrease readability. The error dynamics (5.9) will play a role in analyzing the stability of the formation control law.

5.3.2. GUIDANCE LAW FOR FOLLOWER

The vector field for the follower is defined in the leader frame and can be written as follows:

$$\chi_{\mathrm{f}}^{\mathrm{d}} = \chi_{\mathrm{l}}+\chi_{\infty}\frac{2}{\pi}\tan^{-1}(k_y y_{\mathrm{E}}) \tag{5.10}$$

$$V_{\mathrm{gf}}^{\mathrm{d}} = V_{\mathrm{gl}}+V_{\infty}\frac{2}{\pi}\tan^{-1}(k_x x_{\mathrm{E}}) \tag{5.11}$$

where V_{∞} is the maximum ground speed correction of the follower when the tracking error is very large (V_{∞} should be selected considering the the engine limits and stall limitations, since the maximum ground speed would be $V_{\mathrm{gl}}+V_{\infty}$ and the minimum one would be $V_{\mathrm{gl}}-V_{\infty}$); and k_x is a gain that defines how aggressive the feedback action is. The parameters in (5.10) have similar meanings as in the guidance law for a straight-line in (2.22).

Remark 5.3.1 *The proposed control law can be interpreted as a double vector field. One vector field is* (5.10)*, which depends only on* y_{E}*: this has the structure of a standard vector field for a line that allows the follower to approach the line with the same course as the leader course and passing through the desired course point. The second vector field is for the velocity* (5.11) *and it depends only on* x_{E}*: it allows the follower to accelerate or decelerate in order to reach the desired point.*

A convenient way to visualize the proposed double vector field is to use a vector field similar to the straight-line vector field, but where the arrows of the vector field have different length: each arrow in the space has a length proportional to the desired ground speed.

The guidance law for the follower solving the formation problem is

$$\chi_{\mathrm{f}}^{c} = \chi_{\mathrm{f}}+\frac{1}{\alpha}\dot{\chi}_{\mathrm{f}}^{d}-\frac{\kappa}{\alpha}\mathrm{sat}\left(\frac{\tilde{\chi}_{\mathrm{f}}}{\epsilon}\right) \tag{5.12}$$

$$V_{\mathrm{gf}}^{c} = V_{\mathrm{gf}}+\frac{1}{\beta}\dot{V}_{\mathrm{gf}}^{d}+\frac{1}{\rho\beta}x_{\mathrm{E}}-\frac{\kappa}{\beta}\mathrm{sat}\left(\frac{\tilde{V}_{\mathrm{gf}}}{\epsilon}\right) \tag{5.13}$$

where the derivatives in (5.12), (5.13) can be calculated as

$$\dot{V}_{\mathrm{gf}}^{d} = -V_{\infty}\frac{2}{\pi}\frac{k}{1+(k_x x_{\mathrm{E}})^2}\dot{x}_{\mathrm{E}} \tag{5.14}$$

$$\dot{\chi}_{\mathrm{f}}^{d} = -\chi_{\infty}\frac{2}{\pi}\frac{k}{1+(k_x d_{\mathrm{f}})^2}\dot{y}_{\mathrm{E}} - \dot{\chi}_{\infty}\frac{2}{\pi}\tan^{-1}(k_y y_{\mathrm{E}}) \tag{5.15}$$

and $\chi_{\infty} = \gamma_{\mathrm{f}} - \pi - \chi_{\mathrm{l}}$. The following theorem holds:

Theorem 5.3.1 *The closed-loop system given by the leader (5.1), the follower (5.2), the commands (5.12), (5.13) with vector field (5.10), (5.11) is asymptotically stable if*

$$\min\left(\frac{\rho\kappa}{V_{\mathrm{gf}}}, \frac{\overline{y_{\mathrm{E}}}}{2}\right) > \frac{\pi\epsilon\mu}{4\chi_{\infty}k} \tag{5.16}$$

where $\overline{y_{\mathrm{E}}}$ *and* μ *are an upper bound on the lateral error and a gain specified in the following stability analysis.*

STABILITY ANALYSIS

We consider the following candidate Lyapunov function:

$$W = \frac{1}{2}x_{\mathrm{E}}^2 + \frac{1}{2}\rho\tilde{V}_{\mathrm{gf}}^2 + \frac{1}{2}y_{\mathrm{E}}^2 + \frac{1}{2}\rho\tilde{\chi}_{\mathrm{f}}^2 \tag{5.17}$$

where $\tilde{V}_{\mathrm{gf}} = V_{\mathrm{gf}} - V_{\mathrm{gf}}^{\mathrm{d}}$ is the difference between the ground speed of the follower and its desired ground speed, while $\tilde{\chi}_{\mathrm{f}} = \chi_{\mathrm{f}} - \chi_{\mathrm{f}}^{\mathrm{d}}$ is the difference between the follower course and its desired course. In the following, we will analyze the proposed approach by splitting the Lyapunov function into two parts:

$$W_y = \frac{1}{2}y_{\mathrm{E}}^2 + \frac{1}{2}\rho\tilde{\chi}_{\mathrm{f}}^2, \quad W_x = \frac{1}{2}x_{\mathrm{E}}^2 + \frac{1}{2}\rho\tilde{V}_{\mathrm{gf}}^2 \tag{5.18}$$

where the first part refers to lateral dynamics, and the second part refers to longitudinal dynamics. The reason why longitudinal and lateral dynamics are decoupled is because the stability analysis becomes more tractable. A coupled longitudinal/lateral stability analysis would be more challenging and could be the subject of future work. The stability analysis is done inside the boundary layer of the saturation function. The analysis outside the boundary layer, can be easily done along similar lines as [16].

Lateral Error and Course

We start by analyzing W_{y}: for this part, an approach similar to the analysis of the standard vector field can be adopted [16]. First, starting from (5.18), we calculate the time derivative

$$\begin{aligned}
\dot{W}_y &= y_{\mathrm{E}}\dot{y}_{\mathrm{E}} + \rho\tilde{\chi}_{\mathrm{f}}\dot{\tilde{\chi}}_f \\
&= V_{\mathrm{gf}}y_{\mathrm{E}}\sin(\tilde{\chi}_{\mathrm{f}} + \chi_{\mathrm{f}}^{\mathrm{d}} - \chi_{\mathrm{l}}) - \rho\frac{k}{\epsilon}\tilde{\chi}_{\mathrm{f}}^2 \\
&\leq -\rho\frac{\kappa}{\epsilon}\tilde{\chi}_{\mathrm{f}}^2 + V_{\mathrm{gf}}y_{\mathrm{E}}\sin(\hat{\chi}_{\mathrm{f}}^{\mathrm{d}}) + V_{\mathrm{gf}}y_{\mathrm{E}}\sin(\hat{\chi}_{\mathrm{f}}^{\mathrm{d}} + \tilde{\chi}_{\mathrm{f}}) - \sin(\hat{\chi}_{\mathrm{f}}^{\mathrm{d}})
\end{aligned} \tag{5.19}$$

where we have defined $\hat{\chi}_{\mathrm{f}}^{\mathrm{d}} = \chi_{\mathrm{f}}^{\mathrm{d}} - \chi_{\mathrm{l}}$ for compactness. The following inequalities hold:

$$\begin{aligned}
&|\sin(\hat{\chi}_{\mathrm{f}}^{\mathrm{d}} + \tilde{\chi}_{\mathrm{f}}) - \sin(\hat{\chi}_{\mathrm{f}}^{\mathrm{d}})| \\
&\quad = |\sin(\hat{\chi}_{\mathrm{f}}^{\mathrm{d}})\cos(\tilde{\chi}_{\mathrm{f}}) + \cos(\hat{\chi}_{\mathrm{f}}^{\mathrm{d}})\sin(\tilde{\chi}_{\mathrm{f}}) - \sin(\hat{\chi}_{\mathrm{f}}^{\mathrm{d}})| \\
&\quad = |\sin(\hat{\chi}_{\mathrm{f}}^{\mathrm{d}})(\cos(\tilde{\chi}_{\mathrm{f}}) - 1) + \cos(\hat{\chi}_{\mathrm{f}}^{\mathrm{d}})\sin(\tilde{\chi}_{\mathrm{f}})| \\
&\quad \le |\cos(\tilde{\chi}_{\mathrm{f}} - 1) + \sin(\tilde{\chi}_{\mathrm{f}})| \le 2|\tilde{\chi}_{\mathrm{f}}|.
\end{aligned}$$

The time derivative of W_y becomes

$$\begin{aligned}
\dot{W}_y &\le -\rho\frac{\kappa}{\epsilon}\tilde{\chi}_{\mathrm{f}}^2 + 2V_{\mathrm{gf}}\left|y_{\mathrm{E}}\right|\left|\tilde{\chi}_{\mathrm{f}}\right| + V_{\mathrm{gf}}y_{\mathrm{E}}\sin(\chi_{\mathrm{f}}^{\mathrm{d}} - \chi_{\mathrm{l}}) \\
&= -\rho\frac{\kappa}{\epsilon}\tilde{\chi}_{\mathrm{f}}^2 + 2V_{\mathrm{gf}}\left|y_{\mathrm{E}}\right|\left|\tilde{\chi}_{\mathrm{f}}\right| - V_{\mathrm{gf}}y_{\mathrm{E}}\sin(\chi_\infty\frac{2}{\pi}\tan^{-1}(ky_{\mathrm{E}}) - \frac{\pi}{2}).
\end{aligned} \tag{5.20}$$

In order to have $\dot{W}_y$ negative, we can distinguish two cases. The two cases are obtained by defining the function

$$\phi(y_{\mathrm{E}}) = y_{\mathrm{E}}\sin\left(\frac{2\chi_\infty}{\pi}\tan^{-1}\left(ky_{\mathrm{E}}\right) - \frac{\pi}{2}\right). \tag{5.21}$$

We note that $\phi(y_{\mathrm{E}}) \to (2\chi_\infty k/\pi)y_{\mathrm{E}}^2$ for $ky_{\mathrm{E}} \to 0$ and $\phi(y_{\mathrm{E}}) \to (\sin\chi_\infty)y_{\mathrm{E}}$ for large value of ky_{E}. Consider a new function defined as:

$$\varphi(y_{\mathrm{E}}) = \begin{cases} \dfrac{2\chi_\infty k}{\mu\pi}y_{\mathrm{E}}^2, & \text{if } |y_{\mathrm{E}}| \le \overline{y_{\mathrm{E}}} \\ \dfrac{2\chi_\infty k\overline{y_{\mathrm{E}}}}{\mu\pi}(2|y_{\mathrm{E}}| - \overline{y_{\mathrm{E}}}), & \text{otherwise.} \end{cases} \tag{5.22}$$

Then we have to find μ such that $0 < \varphi(y_{\mathrm{E}}) \le \phi(y_{\mathrm{E}})$. Since the functions are symmetric, we will restrict our attention to $y_{\mathrm{E}} \ge 0$ and aim to show that $\varphi(y_{\mathrm{E}}) \le \phi(y_{\mathrm{E}})$. Given this reasoning, the following two cases are defined:

- Case 1: $0 \le y_{\mathrm{E}} \le \overline{y_{\mathrm{E}}}$
- Case 2: $y_{\mathrm{E}} > \overline{y_{\mathrm{E}}}$

which are analyzed in the following.

Case 1: For $0 \le y_{\mathrm{E}} \le \overline{y_{\mathrm{E}}}$, we have

$$\begin{aligned}
\phi'(y_{\mathrm{E}}) &= \sin\left(\frac{2\chi_\infty}{\pi}\tan^{-1}\left(ky_{\mathrm{E}}\right) - \frac{\pi}{2}\right) + \frac{2k\chi_\infty}{\pi}y_{\mathrm{E}}\left[\frac{\cos(\frac{2\chi_\infty\tan^{-1}(ky_{\mathrm{E}})}{\pi})}{1 + (ky_{\mathrm{E}})^2}\right] \\
&\ge \frac{4\chi_\infty k}{\pi}y_{\mathrm{E}}\left[\frac{1}{2}\frac{\cos(\frac{2\chi_\infty\tan^{-1}(ky_{\mathrm{E}})}{\pi})}{1 + (ky_{\mathrm{E}})^2}\right] \\
&\ge \frac{4\chi_\infty k}{\pi}y_{\mathrm{E}}\left[\frac{1}{2}\frac{\cos(\frac{2\chi_\infty\tan^{-1}(k\overline{y_{\mathrm{E}}})}{\pi})}{1 + (k\overline{y_{\mathrm{E}}})^2}\right] \\
&\ge \frac{4\chi_\infty k}{\pi}y_{\mathrm{E}} = \varphi'(y_{\mathrm{E}})
\end{aligned} \tag{5.23}$$

so if we select μ such that

$$\mu \geq \frac{2(1+(k\overline{y_{\mathrm{E}}})^2)}{\cos\left(\frac{2\chi_\infty}{\pi}\tan^{-1}\left(k\overline{y_{\mathrm{E}}}\right)\right)} \tag{5.24}$$

then $\varphi'(y_{\mathrm{E}}) \leq \phi'(y_{\mathrm{E}})$.

Case 2: On the other side, for $y_{\mathrm{E}} > \overline{y_{\mathrm{E}}}$, we have

$$\begin{aligned} \phi(y_{\mathrm{E}}) &= y_{\mathrm{E}} \sin(\frac{2\chi_\infty}{\pi}\tan^{-1}(k y_{\mathrm{E}})) \\ &\geq y_{\mathrm{E}} \sin(\frac{2\chi_\infty}{\pi}\tan^{-1}(k\overline{y_{\mathrm{E}}})) \end{aligned} \tag{5.25}$$

This implies that $\phi(y_{\mathrm{E}}) \geq \varphi(y_{\mathrm{E}})$ if:

$$\mu \geq \frac{4\chi_\infty k\overline{y_{\mathrm{E}}}}{\pi\sin(\frac{2\chi_\infty}{\pi}\tan^{-1}(k\overline{y_{\mathrm{E}}}))}. \tag{5.26}$$

So we obtain that $\varphi(y_{\mathrm{E}}) \leq \phi(y_{\mathrm{E}})$ for both cases.

Case 1: For $|y_{\mathrm{E}}| \leq \overline{y_{\mathrm{E}}}$, the time derivative is

$$\dot{W}_y \leq -V_{\mathrm{gf}}\begin{pmatrix}|\tilde{\chi}_{\mathrm{f}}| & |y_{\mathrm{E}}|\end{pmatrix}\begin{pmatrix}\frac{\rho k}{\epsilon V_{\mathrm{gf}}} & -1 \\ -1 & \frac{2\chi_\infty k}{\mu\pi}\end{pmatrix}\begin{pmatrix}|\tilde{\chi}| \\ |y_{\mathrm{E}}|\end{pmatrix} \tag{5.27}$$

which is negative definite for system stability if the following holds:

$$\frac{\rho k 2\chi_\infty \kappa}{\epsilon V_{\mathrm{gf}}\mu\pi} > 1 \tag{5.28}$$

Case 2: If $|y_{\mathrm{E}}| \geq \overline{y_{\mathrm{E}}}$, the fact that $\varphi(y_{\mathrm{E}}) \leq \phi(y_{\mathrm{E}})$ implies:

$$\dot{W}_y \leq -\frac{\rho\kappa}{\epsilon}\tilde{\chi}_{\mathrm{f}}^2 + 2V_{\mathrm{gf}}|y_{\mathrm{E}}||\tilde{\chi}_{\mathrm{f}}| - V_{\mathrm{gf}}\varphi(y_{\mathrm{E}}) \tag{5.29}$$

and

$$\dot{W}_y \leq 2V_{\mathrm{gf}}y_{\mathrm{E}}\left(\epsilon - \frac{\chi_\infty k\overline{y_{\mathrm{E}}}|y_{\mathrm{E}}|}{\mu\pi}\right) \tag{5.30}$$

which is negative definite for system stability if the following holds:

$$\frac{\chi_\infty k\overline{y_{\mathrm{E}}}}{\mu\epsilon\pi} > 1 \tag{5.31}$$

The inequalities (5.28) and (5.31) (resulting in (5.16)) can both be satisfied with ϵ small enough, i.e. selecting the boundary layer small enough. For considerations on how to satisfy the inequality (5.16) in practice, the interested reader is referred to [16], where a similar condition also appears.

Longitudinal error and ground speed

The rest of the analysis can be performed using $y_{\mathrm{E}},\ \tilde{\chi}_{\mathrm{f}} \to 0$ for the design described previously. Define the difference between the follower ground speed and the desired one:

$$\tilde{V}_{\mathrm{gf}} = V_{\mathrm{gf}} - V_{\mathrm{gf}}^{\mathrm{d}} = V_{\mathrm{gf}} - V_{\mathrm{gl}} - V_{\infty}\frac{2}{\pi}\tan^{-1}(k x_{\mathrm{E}}). \tag{5.32}$$

This yields the Lyapunov time derivative

$$\begin{aligned}\dot{W}_x &= x_{\mathrm{E}}\dot{x}_{\mathrm{E}} + \rho\tilde{V}_{\mathrm{gf}}\dot{\tilde{V}}_{\mathrm{gf}} \\ &= x_{\mathrm{E}}(V_{\mathrm{gf}}\sin(\chi_{\mathrm{f}} - \chi_{\mathrm{l}} - \frac{\pi}{2}) + V_{\mathrm{gl}}) + \rho\tilde{V}_{\mathrm{gf}}(\dot{V}_{\mathrm{gf}} - \dot{V}_{\mathrm{gf}}^{\mathrm{d}}).\end{aligned} \tag{5.33}$$

Using the convergence property of $\tilde{\chi}_{\mathrm{f}}$, we obtain

$$\dot{W}_x = x_{\mathrm{E}}(V_{\mathrm{gl}} - V_{\mathrm{gf}} + \delta_x) + \rho\tilde{V}_{\mathrm{gf}}(\dot{V}_{\mathrm{gf}} - \dot{V}_{\mathrm{gf}}^{\mathrm{d}}) \tag{5.34}$$

where $\delta_x = V_{\mathrm{gf}} + V_{\mathrm{gf}}\sin(\chi_{\mathrm{f}} - \chi_{\mathrm{l}} - \frac{\pi}{2}) \to 0$ thanks to convergence of $\tilde{\chi}_{\mathrm{f}}$, y_{E} and therefore it can be neglected. We continue the analysis of the time derivative by substituting the dynamics of V_{gf} in (5.2) and expressions of $V_{\mathrm{gf}}^{\mathrm{c}}$ in (5.13)

$$\begin{aligned}\dot{W}_x &= x_{\mathrm{E}}(V_{\mathrm{gl}} - V_{\mathrm{gf}}) + \rho\tilde{V}_{gf}(\beta(V_{\mathrm{gf}}^{\mathrm{c}} - V_{\mathrm{gf}}) - \dot{\tilde{V}}_{\mathrm{gf}}^{\mathrm{d}}) \\ &= x_{\mathrm{E}}(V_{\mathrm{gl}} - V_{\mathrm{gf}}) + \rho\tilde{V}_{gf}(\beta(\frac{1}{\rho\beta}x_{\mathrm{E}} - \frac{\kappa}{\beta\epsilon}\tilde{V}_{\mathrm{gf}})) \\ &= -x_{\mathrm{E}}\tilde{V}_{\mathrm{gf}} + x_{\mathrm{E}}(V_{\mathrm{gl}} - V_{\mathrm{gf}}^{\mathrm{d}}) + \rho\beta\frac{1}{\rho\beta}x_{\mathrm{E}} - \rho\frac{\kappa}{\epsilon}\tilde{V}_{\mathrm{gf}}^{2} \\ &= -V_{\infty}x_{\mathrm{E}}\frac{2}{\pi}\tan^{-1}(k_x x_{\mathrm{E}}) - \rho\frac{\kappa}{\epsilon}\tilde{V}_{\mathrm{gf}}^{2}.\end{aligned} \tag{5.35}$$

By noticing that x_{E} and $\tan^{-1}(k_x x_{\mathrm{E}})$ are both functions in the 1st and 3rd quadrant, it is straightforward to conclude that the time derivative is negative, resulting in asymptotic stability. This concludes the analysis.

The flowchart of the proposed algorithm is given in Figure 5.2. Communication between the leader and the follower is established to calculate the errors used by the controller. The formation vector field controlling the follower's velocity and course can be easily integrated with the velocity-hold loop and course-hold loop on top of the low-level controllers (attitude controllers). In other words, the proposed architecture is compatible with most open-source architectures such as ArduPilot, PX4, NAVIO2. The pseudo-code is shown in Algorithm 1: since the algorithm involves simple additions, multiplications, and static nonlinearities, its computational complexity is $O(1)$.

5.4. ALGORITHM EVALUATION

The algorithm is programmed in ROS and validated in software-in-the-loop (SITL) and hardware-in-the-loop (HITL) environments, using the PX4 autopilot suite. PX4 interacts with a companion PC (for SITL) or with Raspberry pi 3B+ (for HITL) by using the MAVROS communication package, while the 3D physical simulators of the UAV dynamics are Gazebo (for SITL) and X-plane (for HITL) [104–106]. The controllers run at 20Hz in SITL and 2Hz in HITL. The communication frequency between leader and followers is 2Hz in all tests. For comparison purposes, we consider two methods:

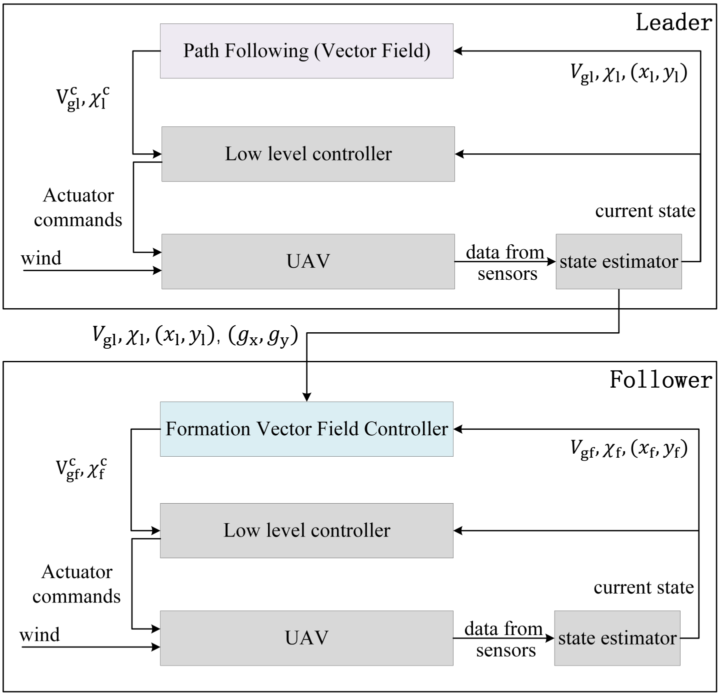


Figure 5.2: The flowchart of the algorithm.

Algorithm 1 Vector-field-based Formation Control

```
Input: $V_{\mathrm{gf}}, V_{\mathrm{gl}}, \chi_{\mathrm{f}}, \chi_{\mathrm{l}}, (g_x, g_y), (x_{\mathrm{l}}, y_{\mathrm{l}}), (x_{\mathrm{f}}, y_{\mathrm{f}})$
$\gamma_{\mathrm{f}} \leftarrow \mathrm{arctan}((x_{\mathrm{f}}, y_{\mathrm{f}}) - (x_{\mathrm{l}}, y_{\mathrm{l}}))$
$d_{\mathrm{f}} \leftarrow ||(x_{\mathrm{f}}, y_{\mathrm{f}}) - (x_{\mathrm{l}}, y_{\mathrm{l}})||$
$x_{\mathrm{E}} \leftarrow g_x + d_f \sin(\gamma_{\mathrm{f}} - \frac{\pi}{2} - \chi_{\mathrm{l}})$
$y_{\mathrm{E}} \leftarrow g_y + d_f \cos(\gamma_{\mathrm{f}} - \frac{\pi}{2} - \chi_{\mathrm{l}})$
$\chi^{\mathrm{d}}_{\mathrm{f}} \leftarrow \chi_{\mathrm{l}} + \chi_{\infty} \frac{2}{\pi} \tan^{-1}(k_y y_{\mathrm{E}})$, follower's desired course
$V^{\mathrm{d}}_{\mathrm{gf}} \leftarrow V_{\mathrm{gl}} + V_{\infty} \frac{2}{\pi} \tan^{-1}(k_x x_{\mathrm{E}})$, follower's desired ground speed
$\chi^{c}_{\mathrm{f}} \leftarrow \chi_{\mathrm{f}} + \frac{1}{\alpha} \dot{\chi}^{d}_{\mathrm{f}} - \frac{\kappa}{\alpha} \mathrm{sat}\left(\frac{\tilde{\chi}_{\mathrm{f}}}{\epsilon}\right)$, where $\tilde{\chi}_{\mathrm{f}} \leftarrow \chi_{\mathrm{f}} - \chi^{\mathrm{d}}_{\mathrm{f}}$
$V^{c}_{\mathrm{gf}} \leftarrow V_{\mathrm{gf}} + \frac{1}{\beta} \dot{V}^{d}_{\mathrm{gf}} + \frac{1}{\rho\beta} x_{\mathrm{E}} - \frac{\kappa}{\beta} \mathrm{sat}\left(\frac{\tilde{V}_{\mathrm{gf}}}{\epsilon}\right)$, where $\tilde{V}_{\mathrm{gf}} \leftarrow V_{\mathrm{gf}} - V^{d}_{\mathrm{gf}}$
Output: $\chi^{c}_{\mathrm{f}}, V^{c}_{\mathrm{gf}}$
```

- The state-of-the-art method proposed in [103] under the assumption of absence of wind. This method, abbreviated as "wind-absence method", assumes that the UAV air speed coincides with its ground speed, and that the UAV heading angle coincides with its course angle;

- The state-of-the-art method based on unicycle dynamics [107]. The method, abbreviated as "unicycle method" exploits the fact that the guidance dynamics for UAVs are analogous to the kinematic model of a unicycle mobile robot, cf. the survey [6].

5.4.1. Software-in-the-loop experiments

To assess the difference without and with wind, the wind plugin of Gazebo is used to create a wind scenario, using: WindForceMean = 2.5, WindForceMax= 3.5, WindGuestMean = 7, WindGuestMax = 10, WindDirectionMean = 45.

Line and orbit paths

The experiments are first performed with two UAVs for a line and an orbit path (without and with wind). The desired gap between leader and follower is (-2, -2)m. In the straight-line path scenario, the leader flies at 18 m/s along a straight-line path, while the follower starts from a perpendicular position with respect to the path. In the orbit path scenario, the orbit has a radius of 400m. The follower starts from the bottom left region.

The results without wind are reported in Figures 5.3 and 5.4. The results in the presence of wind are reported in Figures 5.5 and 5.6. The formation errors below each path show that the presence of wind leads to larger errors. In particular, the wind makes the orbit scenario more challenging for two reasons: first, depending on the course, the wind affects the UAV with different intensity along the orbit; second, in order to keep the gap, the follower should perform an orbit of smaller radius and at smaller velocity, which makes the flight harder.

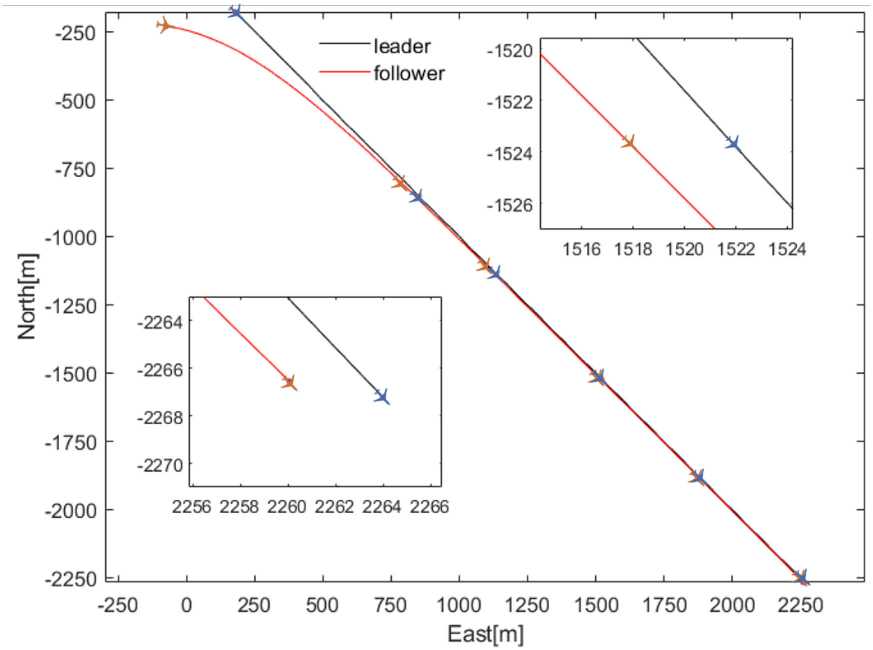
Mixed path

The mixed path is a combination of lines and orbits so that an 8-like shape is formed. We consider 1 leader with 4 followers, with desired gaps between leader and followers being (-2, -2), (-4, -4), (-6,-6), (-8, -8)m respectively. In order to evaluate the transient performance, we let the 4 followers start from different regions.

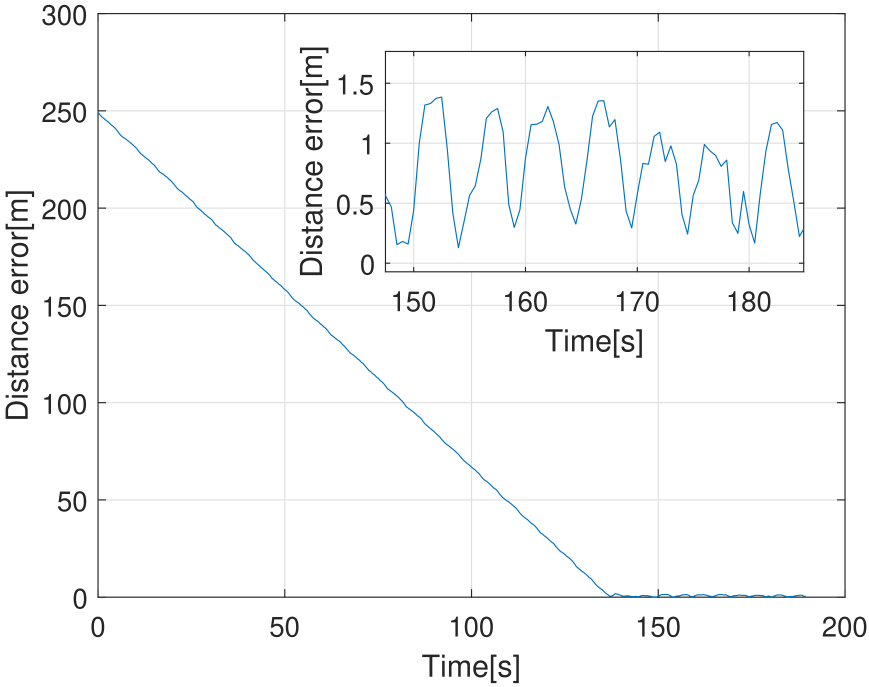
To conclude the SITL part, we report the results (in the presence of wind) of the wind-absence method [103] for a straight-line (Figure 5.9), an orbit (Figure 5.10), and a mixed path (Figure 5.11), and of the unicycle method [107] for a straight-line (Figure 5.12), an orbit (Figure 5.13) and a mixed path (Figure 5.14). Comparisons show that the proposed method performs more than 36% better than the method of [103] and more than 120% better than the method of [107] (see also Table 5.2). The proposed method performs better than the wind-absence method because the wind-absence method cannot not properly handle the effect of the wind on UAV air speed and heading angle: the fact that these quantities are different from the ground speed and the course angle creates some bias in the formation error. The proposed method performs better than the unicycle method because with the unicycle method the follower UAV tries to imitate the course of the leader independently from its gap. On the other hand, with the proposed method, the follower UAV tries first to reduce the gap when the gap is large and then it will aim at aligning its course to the leader when the gap is smaller.

5.4.2. Hardware-in-the-loop experiments

In the HITL environment, the code runs on PX4 hardware interfaced to the rest of the system as shown in Figure 2.15. X-plane provides the fixed-wing dynamics, and connects with the Pixhawk board by a serial port. The proposed method runs in ROS in a Raspberry pi 3B+. The WiFi module in the Raspberry pi 3B+ is used for communication between the leader

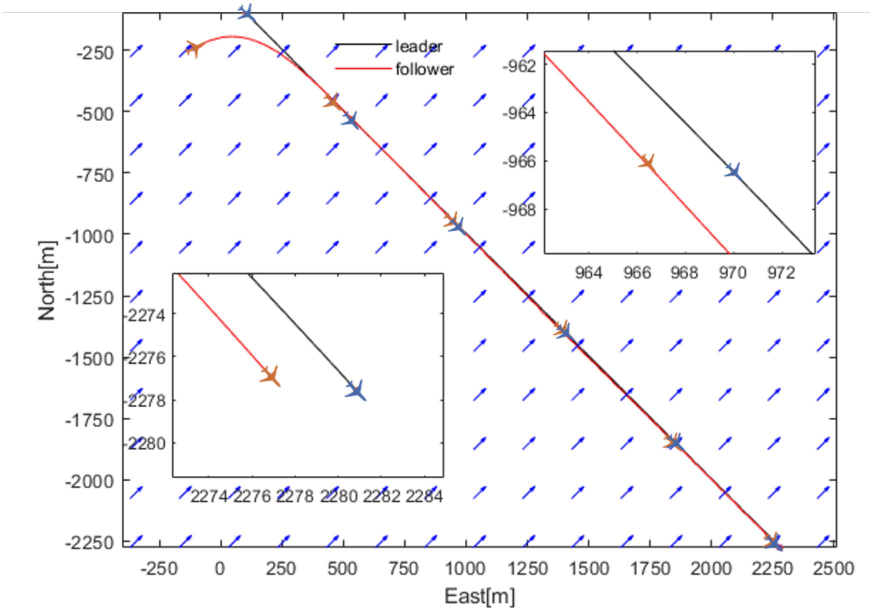


(a) Formation along straight-line path.

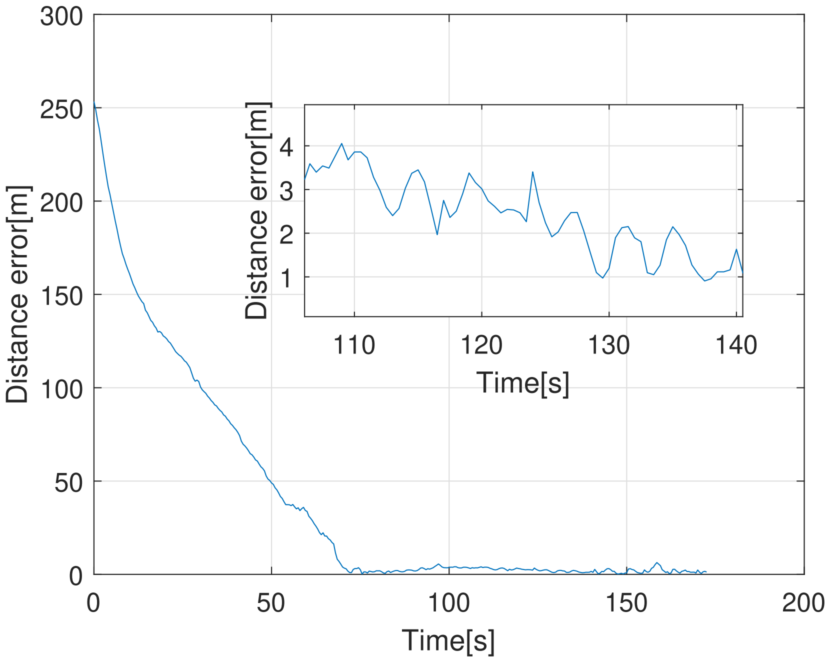


(b) Formation error, straight-line path.

Figure 5.3: Proposed method: SITL experiment without wind, straight-line path.

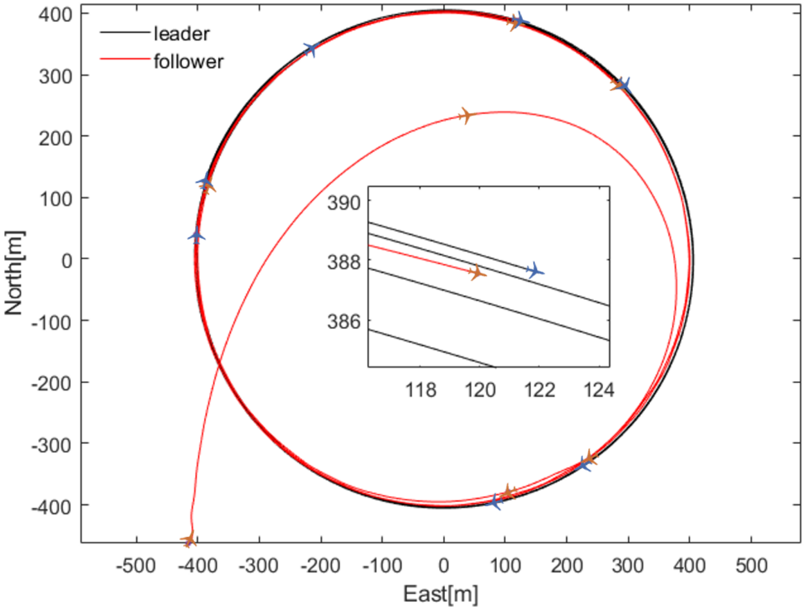


(a) Formation along straight-line path.

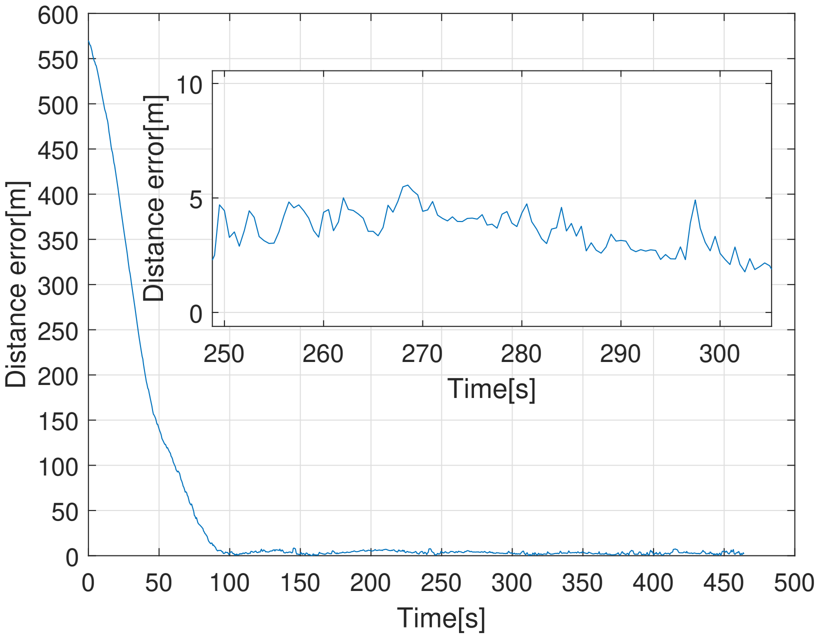


(b) Formation error, straight-line path.

Figure 5.4: Proposed method: SITL experiment with wind, straight-line path.

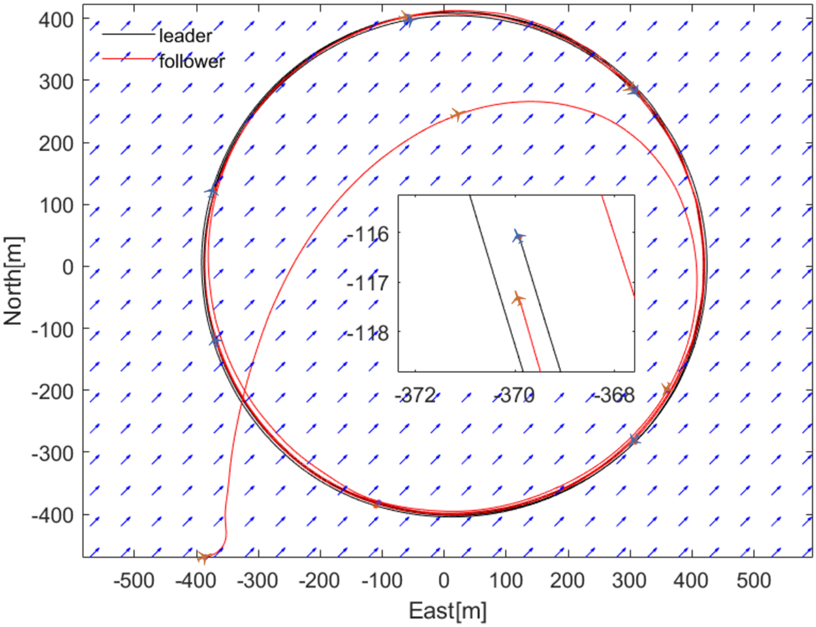


(a) Formation along orbit path.

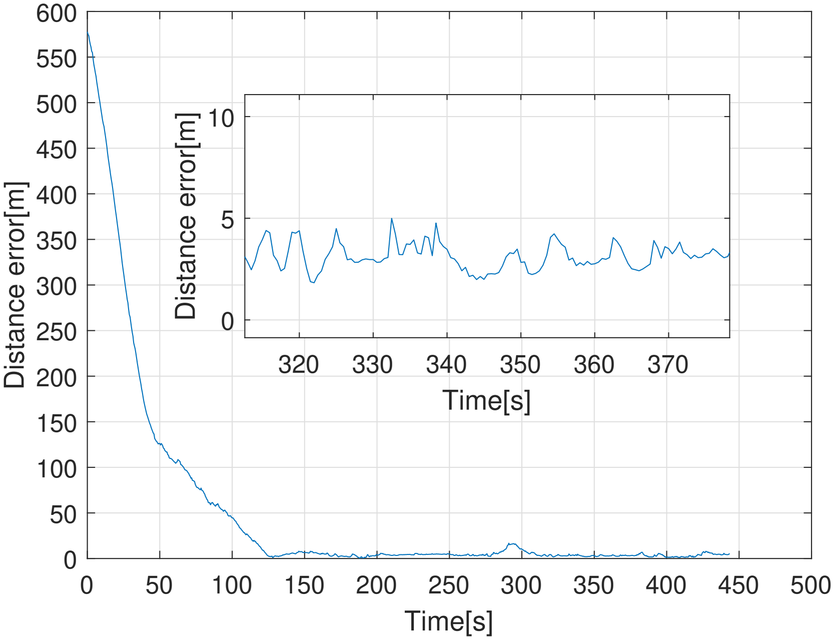


(b) Formation error, orbit path.

Figure 5.5: Proposed method: SITL experiment without wind, orbit path.

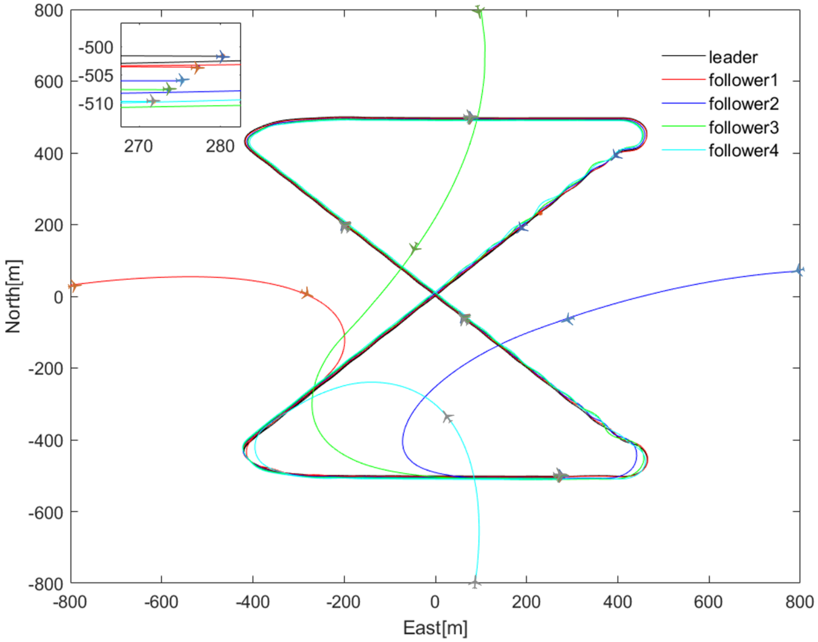


(a) Formation along orbit path.

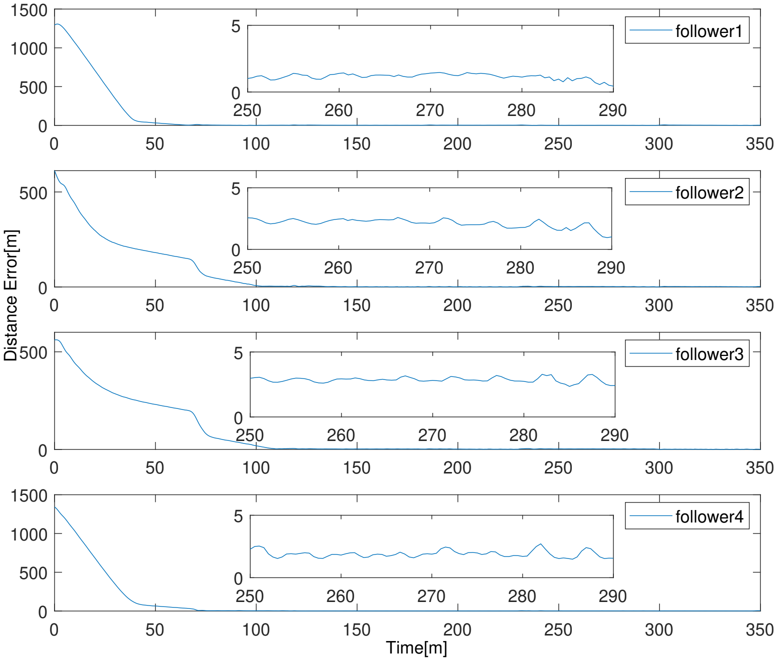


(b) Formation error, orbit path.

Figure 5.6: Proposed method: SITL experiment with wind, orbit path.

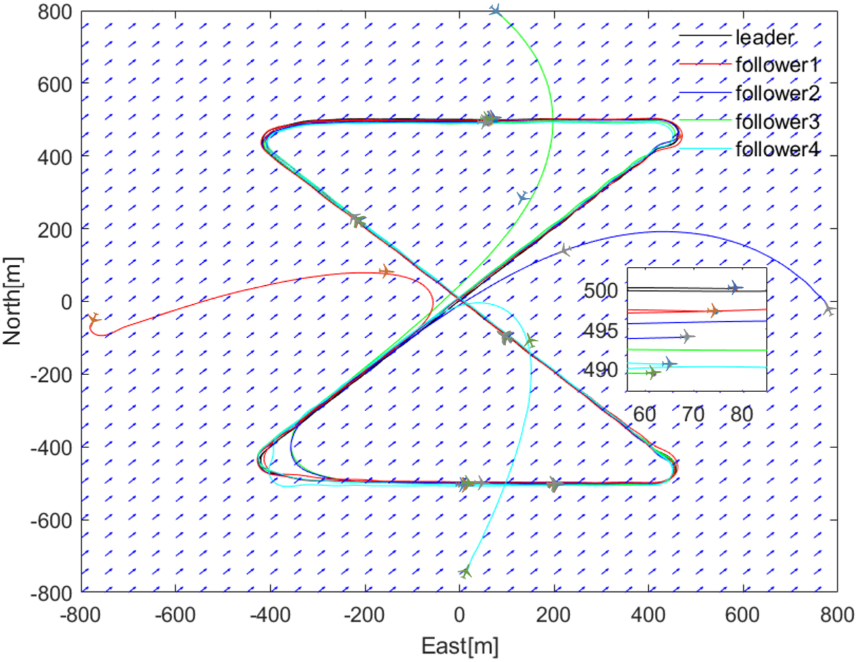


(a) Formation along mixed path.

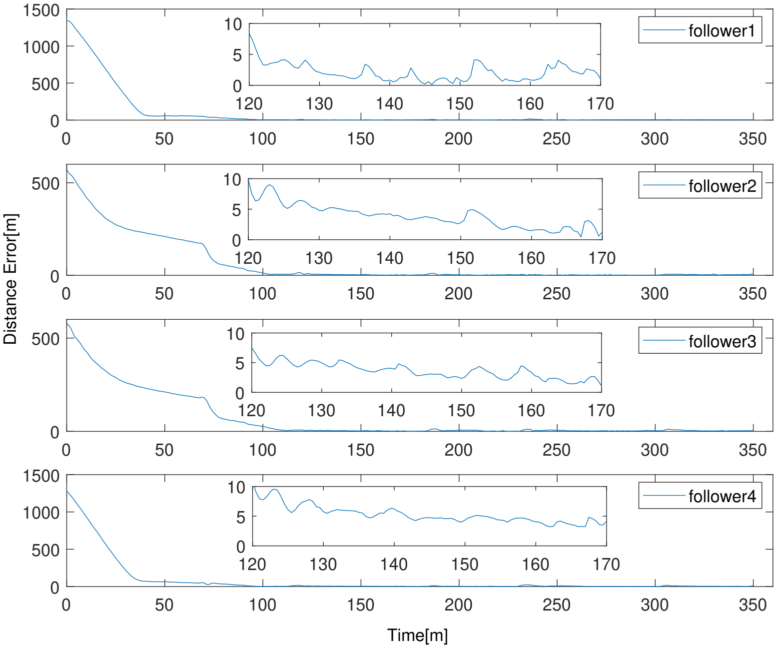


(b) Formation error, mixed path.

Figure 5.7: Proposed method: SITL experiment without wind, mixed path.

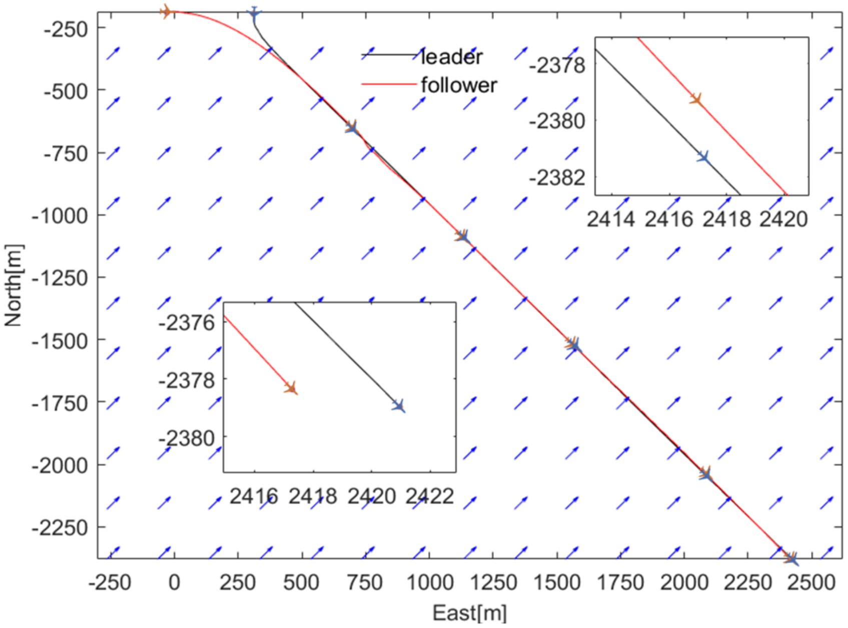


(a) Formation along mixed path.

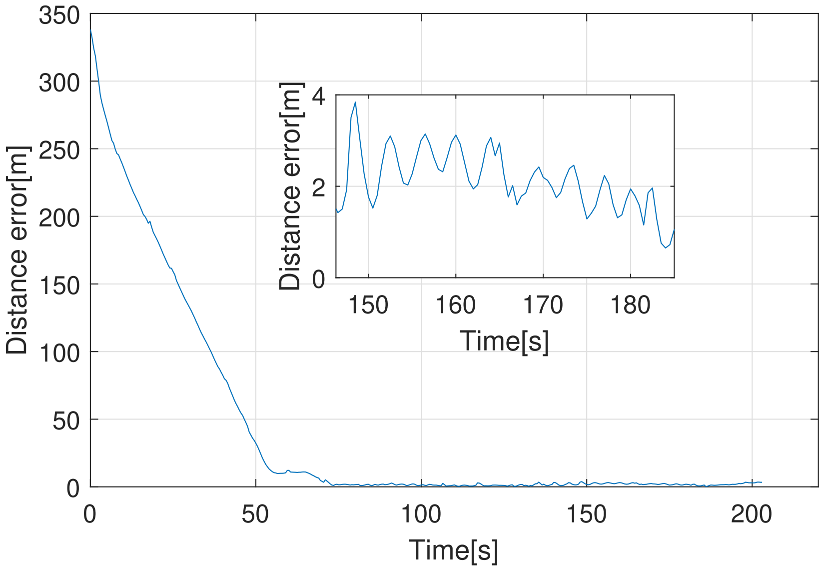


(b) Formation error, mixed path.

Figure 5.8: Proposed method: SITL experiment with wind, mixed path.

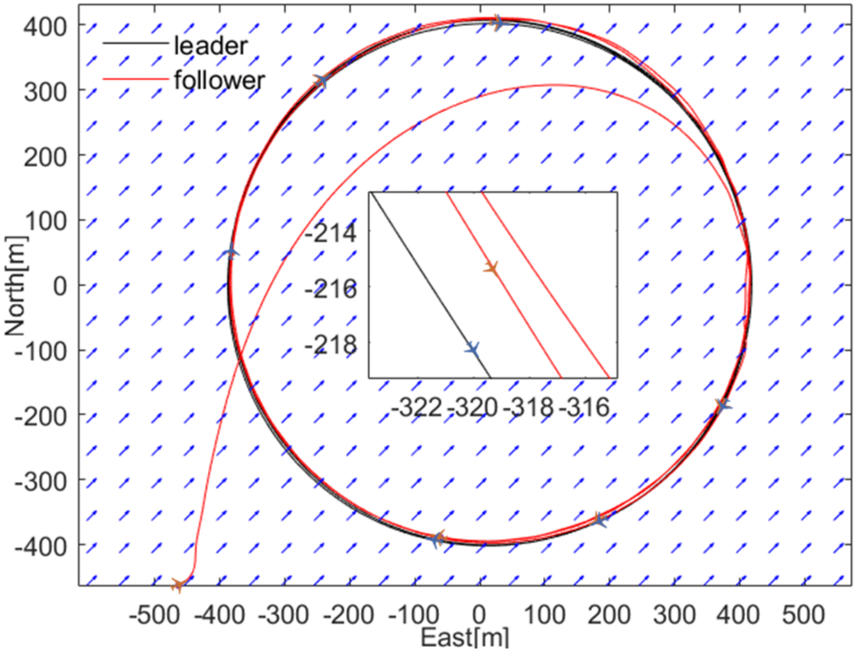


(a) Formation along straight-line path.

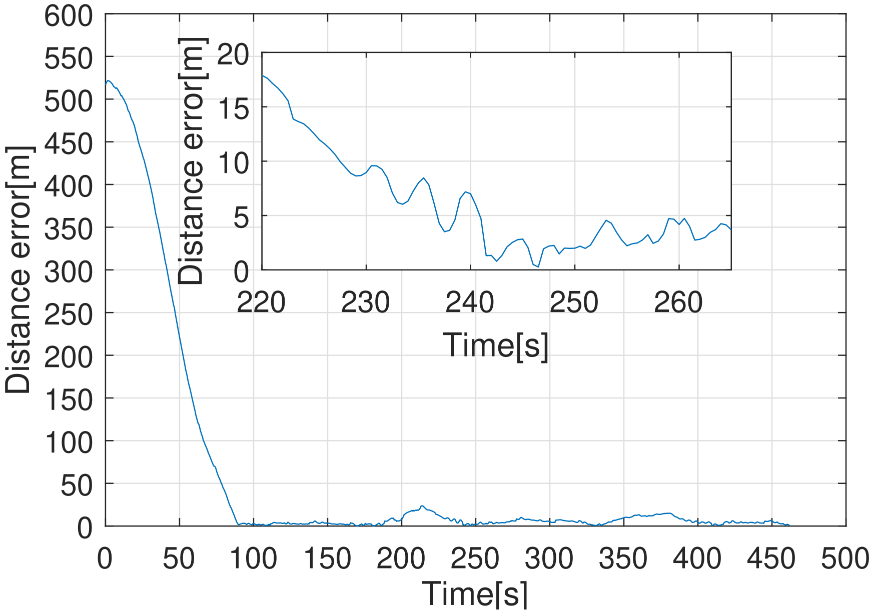


(b) Formation error, straight-line path.

Figure 5.9: State-of-the-art method assuming absence of wind: experiments with wind, straight-line path.

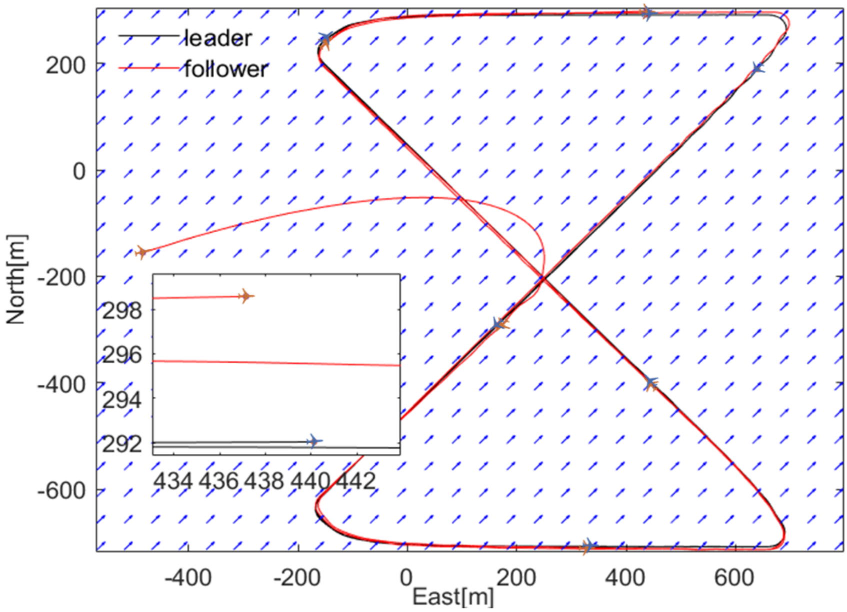


(a) Formation along orbit path.

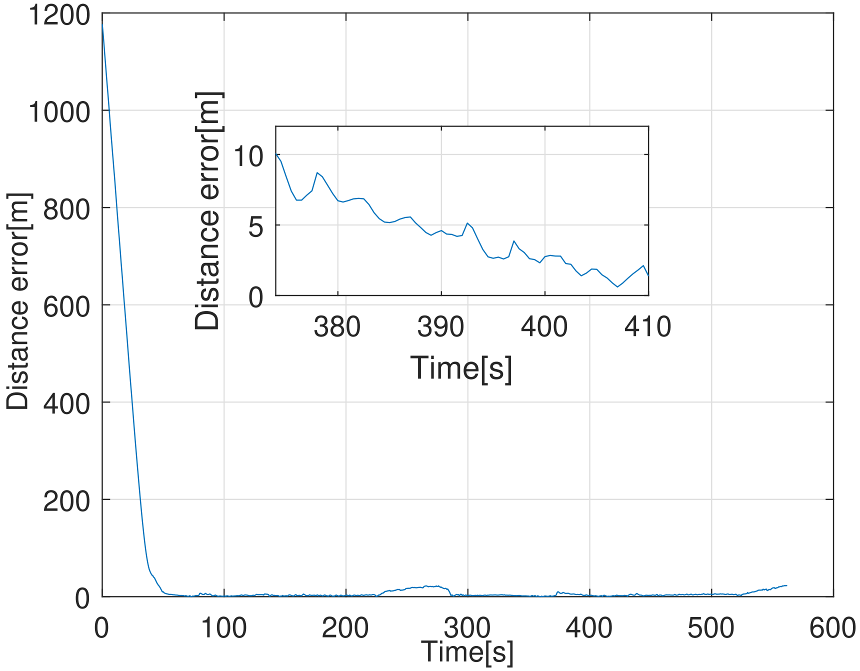


(b) Formation error, orbit path.

Figure 5.10: State-of-the-art method assuming absence of wind: experiments with wind, orbit path.

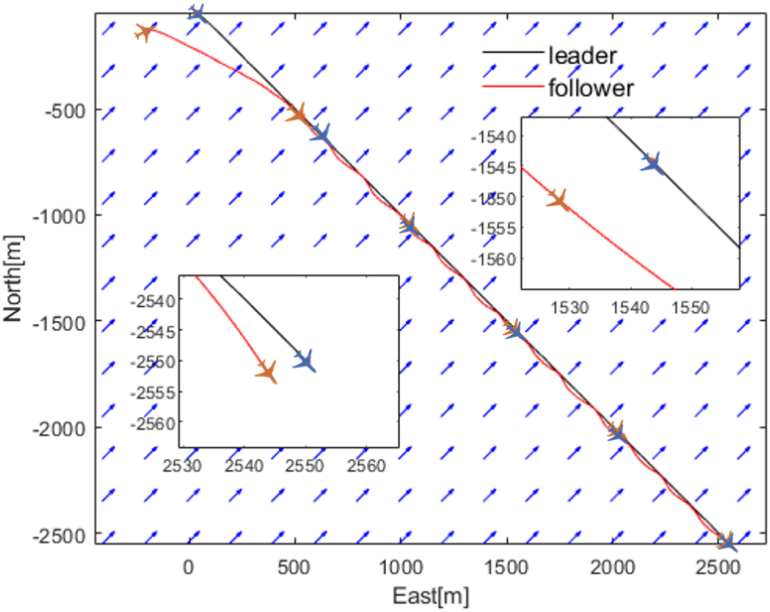


(a) Formation along mixed path.

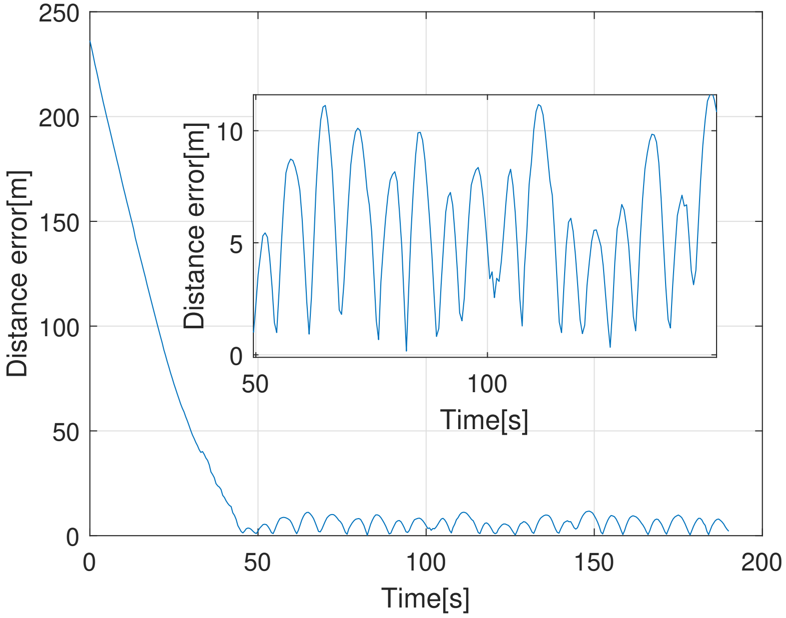


(b) Formation error, mixed path.

Figure 5.11: State-of-the-art method assuming absence of wind: experiments with wind, mixed path.

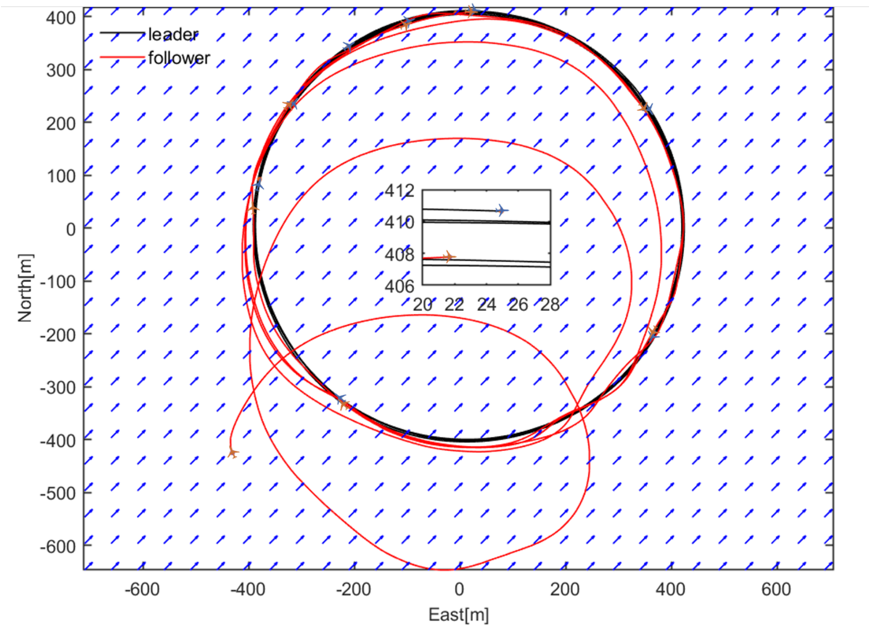


(a) Formation along straight-line path.

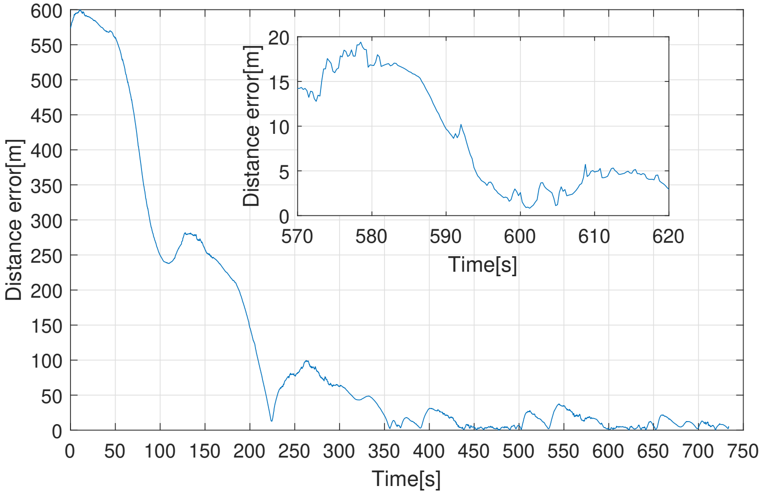


(b) Formation error, straight-line path.

Figure 5.12: State-of-the-art method assuming unicycle dynamics: experiments with wind, straight-line path.

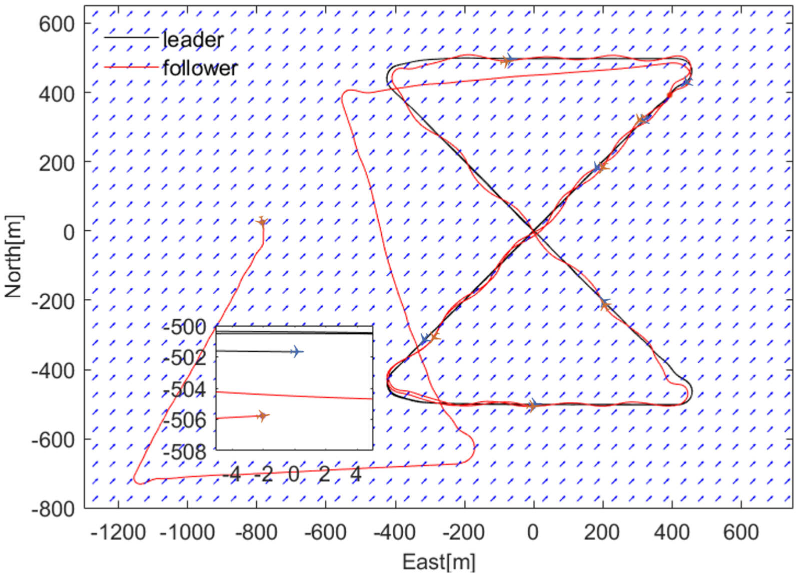


(a) Formation along orbit path.

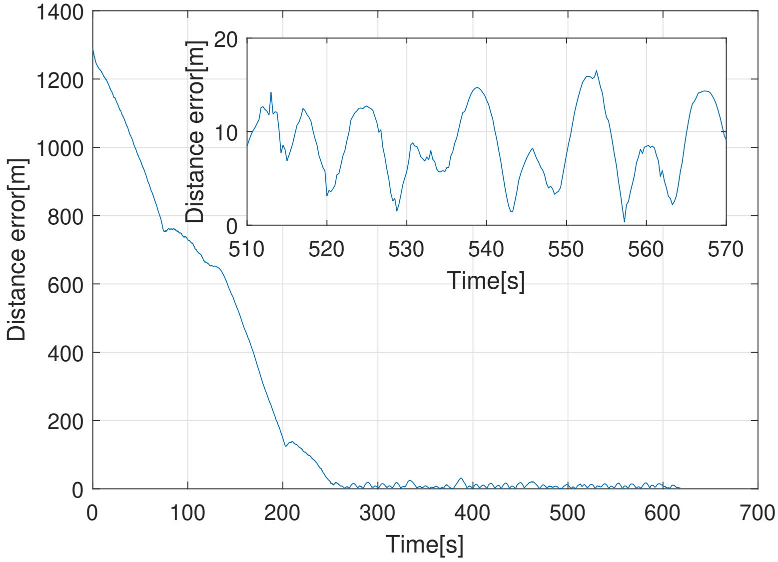


(b) Formation error, orbit path

Figure 5.13: State-of-the-art method assuming unicycle dynamics: SITL experiments with wind, orbit path.



(a) Formation along mixed path.



(b) Formation error, mixed path.

Figure 5.14: State-of-the-art method assuming unicycle dynamics: experiments with wind, mixed path.

and the followers. The HITL setting allows to check the compatibility of the implementation with the true hardware. Most importantly, HITL allows to test the algorithm in the presence of realistic communication time delays and packet losses, which will inevitable influence the algorithm results. Because X-plane only supports 1 UAV for each PC, we tested 2 UAVs at same time as shown in Figure 5.15.

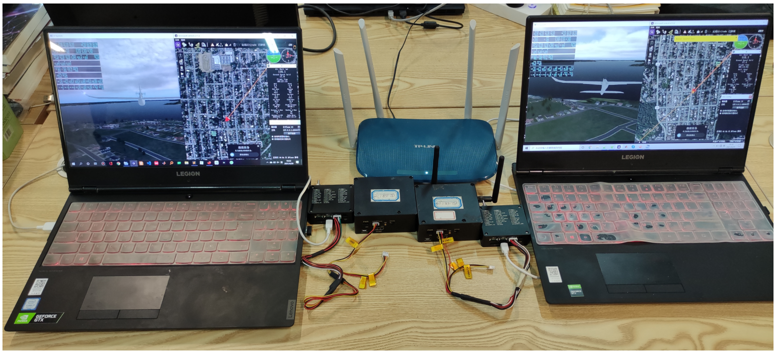
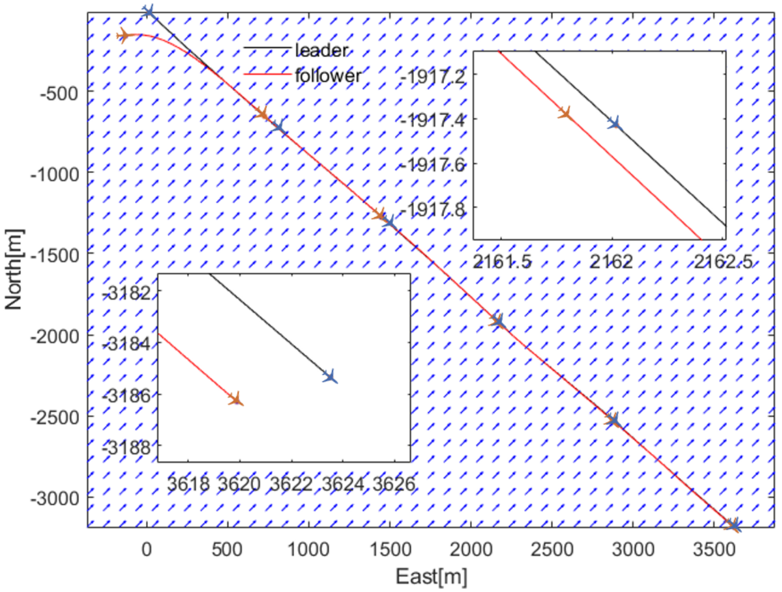


Figure 5.15: Setup for Hardware-In-The-Loop experiments, with PX4 as autopilot suite.

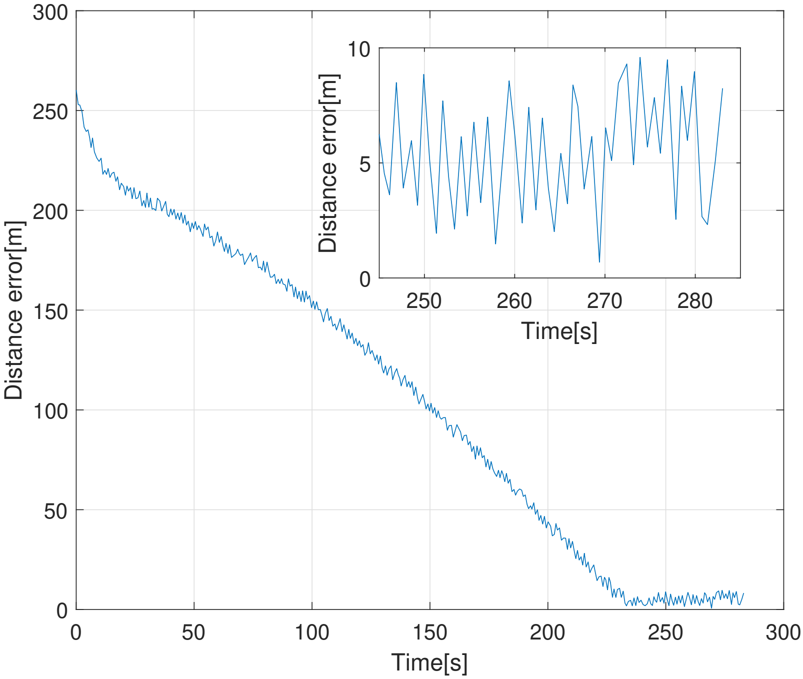
Similarly to the SITL scenario, we test a straight-line path, an orbit path, and a mixed path, all of them in the presence of wind. The results for the straight-line are in Figure5.16; the results for the orbit are in Figure 5.17; the results for the mixed path are in Figure 5.18. All the results are summarized in Table 5.2: notice that all the results are steady-state formation errors, i.e. after removing the transient. The performance of the HITL degrades as compared to the corresponding SITL, which can be explained in at least two ways: first, the control frequency in HITL is 2Hz (in SITL it is 20Hz); second, time delays and packet losses become non-negligible in HITL. In fact, the USB, serial port, WiFi, and ROS node give a cumulative time delay which we have estimated to be between 20ms and 300ms. This is in line with results reported in the literature about ROS node operation [108, 109]. According to the leader air speed, these delays can result in a distance error of 3-5m [42,110]. Therefore, the presented results provide a realistic validation of the proposed method. As a matter of fact, Table 5.2 shows that the errors of the proposed algorithm with the HITL implementation are much smaller than the unicycle method with SITL implementation: this shows good performance even with high-latency communication.

5.5. SUMMARY

This chapter has proposed a vector field method to formation control of fixed-wing UAVs, by means of a non-uniform vector field (the field can change in both magnitude and direction). It was shown that the proposed formation control law can work even with low frequency. Differently from related literature in the field that work under the assumption of absence of wind, the proposed method does not need such assumption, which increases its practical value. This implies that one can handle a UAV air speed different from its ground speed, and a UAV heading angle different from its course angle (due to the effect of the wind triangle), and can work with a low communication frequency. The stability of the proposed method

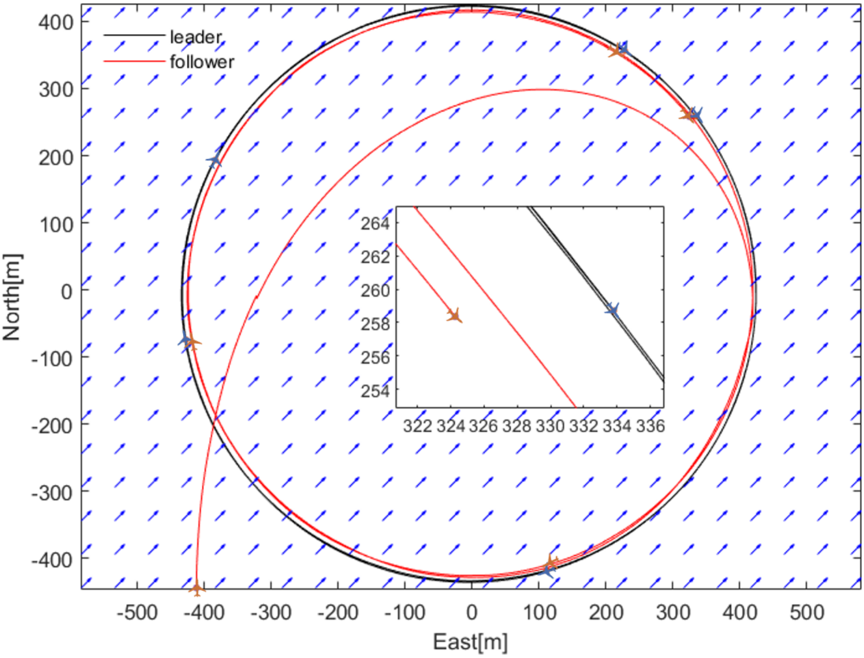


(a) Formation along straight-line path.

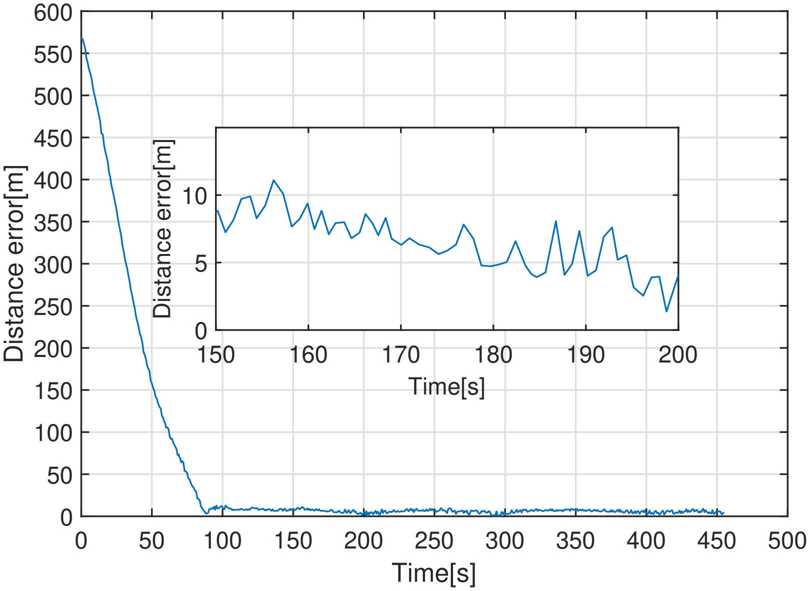


(b) Formation error, straight-line path.

Figure 5.16: HITL experiment with wind, straight-line path.

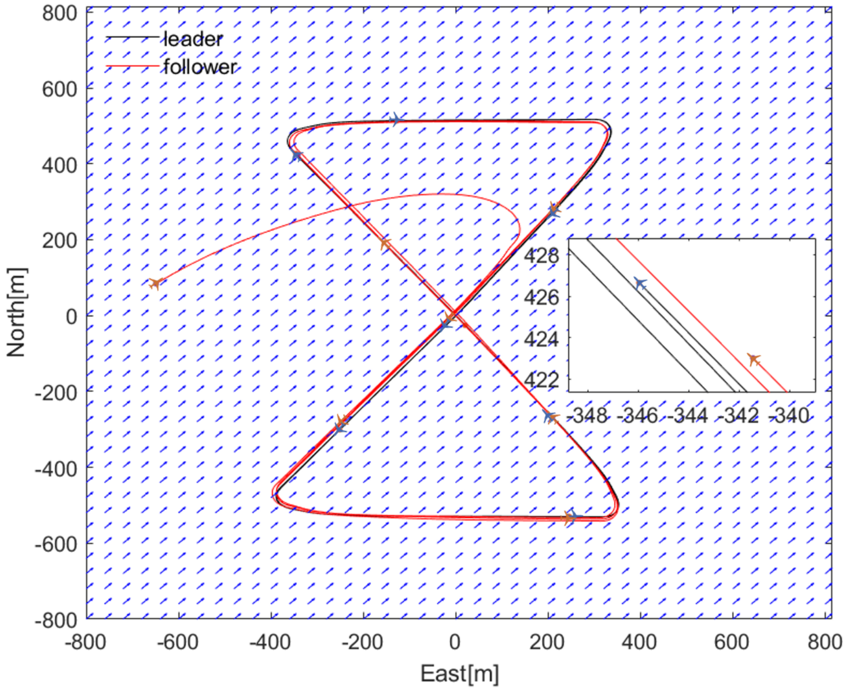


(a) Formation along orbit path.

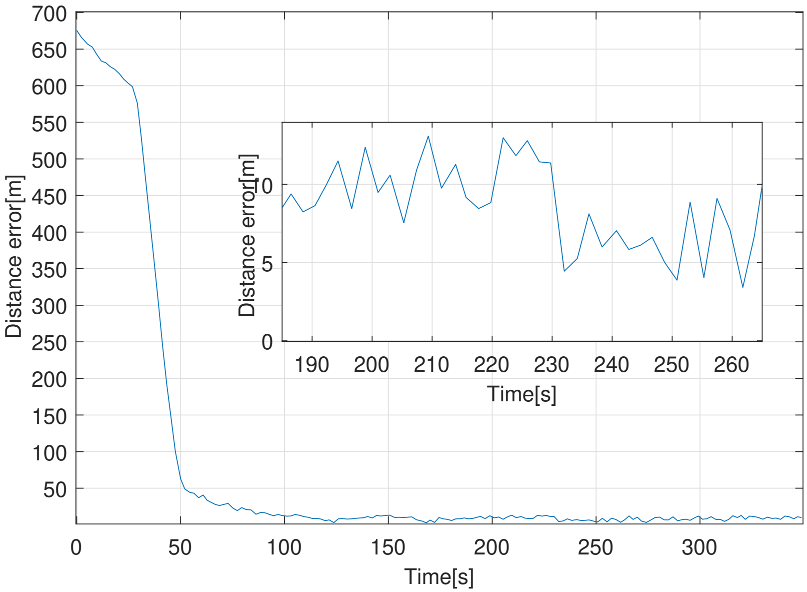


(b) Formation error, orbit path.

Figure 5.17: HITL experiment with wind, orbit path.



(a) Formation along mixed path.



(b) Formation error, mixed path.

Figure 5.18: HITL experiment with wind, mixed path.

Table 5.2: RMS formation error at steady-state (after convergence) under different scenarios. In the last line, the performance of three algorithms (proposed, wind-absence, and unicycle) in SITL with wind are compared.

	Line path RMS error	Orbit path RMS error	Mixed path RMS error
SITL: no wind (proposed)	0.826m	3.295m	1.804m, 1.906m, 2.132m, 2.168m
SITL: wind (proposed)	1.889m	5.228m	4.973m, 4.833m, 5.750m, 6.701m
HITL: wind (proposed)	5.468m	6.191m	7.782m
SITL: wind (wind absence)	2.562m **(+36%)**	8.411m **(+61%)**	7.884m **(+63%)**
SITL: wind (unicycle)	6.278m **(+232%)**	13.996m **(+168%)**	11.009m **(+121%)**

was analyzed. Validations have been carried out in software-in-the-loop and hardware-in-the-loop comparative experiments using PX4 autopilot.

6

REAL FLIGHT EXPERIMENTS FOR FIXED-WING UAV FORMATION

6.1. INTRODUCTION

Despite the increasing interest in fixed-wing UAVs formation, few flight experiment can be found in the literature, and even fewer explanations can be found about how to deal with uncertainties, such as unknown wind speed and direction, unreliable communication, and unmodeled dynamics in real formation flight. In this chapter, we consider a real flight test with a formation of UAVs, with the aim to achieve the transfer of the algorithms proposed in this Ph.D. thesis from simulations to the real world (also referred to in the literature as sim-to-real transfer). To deal with the uncertainty in the communication and in the wind environment, we augment the vector field formation algorithm discussed in Chapter 5 with an appropriately designed distributed estimator. We will present the resulting performance for a real flight of 5 fixed-wing UAVs.

The rest of this chapter is organized as follows. Sect. 6.2 explains some control problems typically met in the real formation flight. The proposed estimation method is presented in Sect. 6.3. The flight experiments and comparison analysis are illustrated in Sect. 6.4. Finally, Sect. 6.5 concludes the chapter.

6.2. OPEN PROBLEMS

The first bottleneck in real formation flight is coming from communication. As we know from Chapters 4 and 5, each UAV in the formation inevitably requires some reference information (e.g. the leader's position and status) shared via communication equipment. Hence, data transmission delay is unavoidable during network communication, which can span from a few of milliseconds to several seconds, depending on the channel quality, on the number of UAVs in the formation, and on the distance between the master and slave communication nodes. Data transmission delay may result in instability and poor performance of the formation: for example, it may create some mismatch between the expected formation gap and the actual one [111]. In several UAV formation papers, it is very common to ignore the data transmission delay or to assume that the communication is working in an ideal environment. In [112], a multi-hop communication structure is designed for formation control, where the information is transferred via the centralized ground control station. However, it is inevitable in the centralized ground control station scenario that the communication quality will decrease within the increase of UAVs. In [113], an ad-hoc communication protocol is proposed based on ROS bridge. The protocol publishes the UAV position (with frequency 10Hz) and status (with frequency 2Hz) by using an 802.11n wireless radio: however, this wireless communication will inevitably limit the fixed-wing UAVs working distance. The communication delay is controlled in a fixed time in [114], which means that the error between the real leader's position and the follower received leader's position is assumed to be a constant error. However, in real flight, such error is time-varying and influences the formation shape. The most evident problem is that the formation will be destroyed if the communication delay is large or the communication is disconnected in general. In fact, the follower will keep following the last received position/status of the leader. Therefore, how to tackle the time-varying radio communication uncertainty is still an open problem for a real flight test with a formation of UAVs.

6.3. Optimization of formation vector-field algorithm

6.3.1. Error analysis

There are multiple sources of uncertainty that influence UAV formation results during real world flight testing. To explain some of these factors, let us recall the ideal formation error in Figure 5.1, where the distance d_f is designed as the distance between the leader and the follower, and the follower can obtain it without any time delay. However, in real flight testing with a time-varying time-delay t_Δ, we need to adopt a more realistic way to describe the formation error, which is shown in Figure 6.1.

Here the variables $(x_\mathrm{l}, y_\mathrm{l})$ and V_gl are the follower received leader's position and the ground speed at time t_i. However, the real position and ground speed of the leader UAV at time t_i are $(x_1^{'}, y_1^{'})$ and $V_\mathrm{gl}^{'}$, due to the delay. Accordingly, $d_\Delta^{'}$ is the moving distance covered by the leader in the communication delay t_Δ. In the follower frame, the variables $x_{E_\Delta}^{'}$ and $y_{E_\Delta}^{'}$ are the error caused by $d_\Delta^{'}$. Let us now define $\hat{d}_\Delta^{'}$ as an estimate of $d_\Delta^{'}$, and let us define by $\psi_\Delta^{'}$ a wind disturbance factor resulting in the difference between V_gl and $V_\mathrm{gl}^{'}$. The main idea adopted in this chapter will be the following: a distributed local estimator will be used to reduce some of the uncertainties affecting the UAV cooperation network.

6.3.2. The improved formation vector algorithm

Based on the error analysis, the guidance law in Sect. 5.3.2 will be updated to account for the real flight.

Because all UAVs are equipped with navigation equipment, such as GPS, it is easy to get an accurate estimate of the communication time delay t_Δ by adding a time stamp in the communication message. Hence, we can design a distributed estimator for the follower UAVs by using the dead reckoning position method to estimate is the leader's position at time t_i.

Figure 6.2 illustrates a standard dead reckoning position method for the follower UAV.

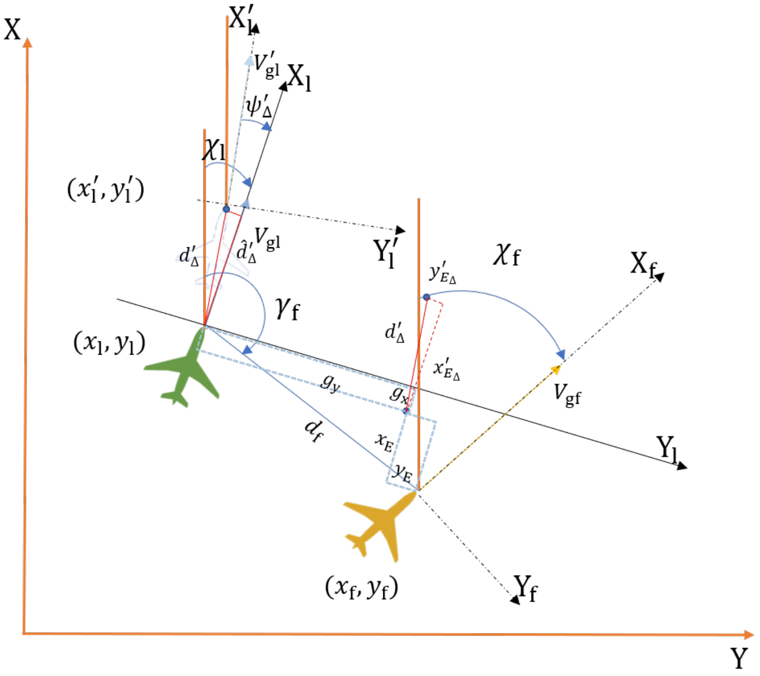


Figure 6.1: The coordinates for the formation error with uncertainties

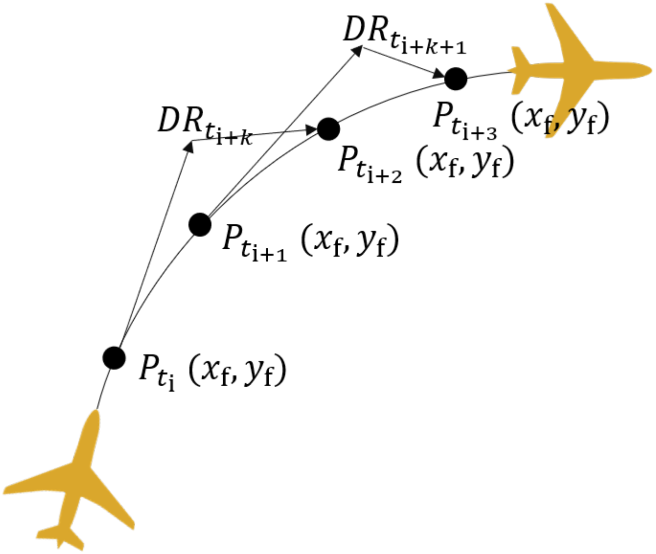


Figure 6.2: The dead reckoning position of follower UAV

$P_{(t_i)}(x_{\mathrm{f}}, y_{\mathrm{f}})$ is the position of the follower at time t_i, which is used for estimating the position of the follower $DR_{(t_{i+k})}(x_{\mathrm{dr}}, y_{\mathrm{dr}})$ at time t_{i+k} by the following dead reckoning position method (6.1):

$$\begin{aligned} x_{\mathrm{dr}(t_{i+k})} &= x_{\mathrm{f}(t_i)} + V^c_{\mathrm{gf}_{(t_i)}} t_k \cos(\chi^c_{\mathrm{gf}_{(t_i)}}) \\ y_{\mathrm{dr}(t_{i+k})} &= y_{\mathrm{f}(t_i)} + V^c_{\mathrm{gf}_{(t_i)}} t_k \sin(\chi^c_{\mathrm{gf}_{(t_i)}}) \end{aligned} \tag{6.1}$$

where t_i and t_{i+k} are the time indexes, the subscript k indicates the steps ahead. The variables χ^c_{gf} and V^c_{gf} are groundspeed and heading control signals of follower at time t_i, which can be calculated by (5.12) and (5.13). t_k is the value of the step size.

According to the wind triangle scheme of Figure 2.6 and the disturbance analysis of in Sect. 3.2.2, the shift error between the estimated position and the actual position of the follower at time t_{i+k} (referred to as $||P_{(t_{i+k})}(x_{\mathrm{f}}, y_{\mathrm{f}}) - DR_{(t_{i+k})}(x_{\mathrm{dr}}, y_{\mathrm{dr}})||$) is mainly caused by the wind disturbance. Because the UAVs fly in a formation with relatively small gaps, the following is a reasonable assumption.

Assumption 6.3.1 *The UAVs are close to each other (e.g. less than 100m range). Hence, the effect of the wind on the formation is the same for all fixed-wing UAVs, that is, the effect of* ψ'_{Δ} *is the same.*

Based on the assumption, the distributed estimator for calculating the estimated leader's position can be calculated according to the following proposition.

Proposition 6.3.1 *The shift error* $||P_{t_{i+k}}(x_{\mathrm{f}}, y_{\mathrm{f}}) - DR_{t_{i+k}}(x_{\mathrm{dr}}, y_{\mathrm{dr}})||$ *expressed in the follower frame is equal to the error between* d'_{Δ} *and* $\hat{d}'_{\Delta}$ *in Figure 6.1 in the leader frame.*

Hence, the distance error $(x'_{\mathrm{E}},\ y'_{\mathrm{E}})$ *used to replace* $(x_{\mathrm{E}},\ y_{\mathrm{E}})$ *of Sect. 5.3.2 starting from time* $t_{(n+k)}$ *can be estimated by:*

$$\begin{aligned} x'_{\mathrm{E}} &= x_E + \hat{d}'_{\Delta}\cos(\chi_{\mathrm{l}}) + \frac{\sum\limits_{i=1}^{n} ||P_{t_{i+k}}(x_{\mathrm{f}}, y_{\mathrm{f}}) - DR_{t_{i+k}}(x_{\mathrm{dr}}, y_{\mathrm{dr}})|| t_{\Delta}}{n t_k} \cos(\chi_{\mathrm{f}}) \\ y'_{\mathrm{E}} &= y_E + \hat{d}'_{\Delta}\sin(\chi_{\mathrm{l}}) + \frac{\sum\limits_{i=1}^{n} ||P_{t_{i+k}}(x_{\mathrm{f}}, y_{\mathrm{f}}) - DR_{t_{i+k}}(x_{\mathrm{dr}}, y_{\mathrm{dr}})|| t_{\Delta}}{n t_k} \sin(\chi_{\mathrm{f}}) \end{aligned} \tag{6.2}$$

where $\hat{d}'_{\Delta} = V_{\mathrm{gl}} t_{\Delta}$ *and* n *is designed to reduce the wind noisy effect of the estimated position: if the wind is strong,* n *should be sufficiently large.*

Because the loop time n is a small integer, the computational complexity of the optimization guidance law is $O(1)$.

6.4. FLIGHT EXPERIMENTS

6.4.1. FORMATION FLIGHT EXPERIMENT

For testing the methods, we prepared 5 fixed-wing UAVs as shown in Figure 6.3.

Figure 6.3: The crosswind UAVs

The algorithm has been implemented in the Raspberry pi 3B+ using the ROS bridge to connect with the Ardupilot open-source flight controller (cf. the hardware in Figure 5.15). The ground control station is QGC (cf. Figure 6.5), which supports to control multiple UAVs in one single ground station.

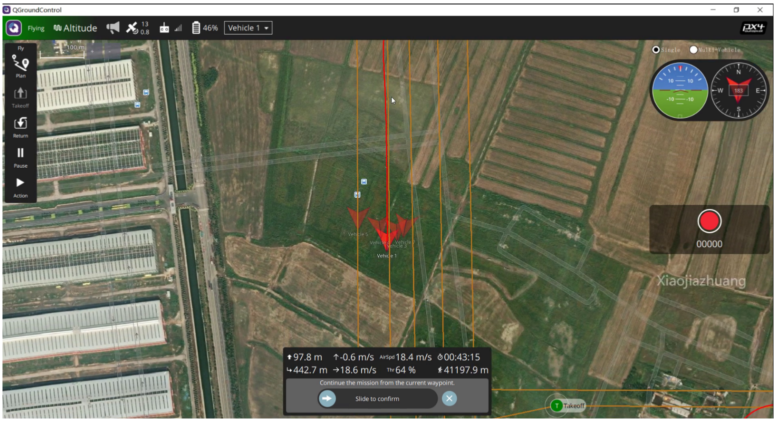
Microhard pDDL900 (as shown in Figure 6.4) is adopted as communication equipment with frequency band is 900MHz. We use a "point to multi-points" mode, which allows the leader UAV node to share its position and status with all followers. The communication data is designed as below. The message is shared by the leader on a 2 Hz basis:

Figure 6.4: The Microhard pDDL900 datalink

```
struct UAV_status{
  uint16_t message_id;//message ID
  uint16_t uav_id;//UAV ID
```

```
    uint16_t plane_state;// UAV character: leader or follower
    uint16_t group_id;//
    int32_t longitude_x; // position in earth frame
    int32_t latitude_y;
    float altitude_z;
    float heading;// the current course angle
    float velocity; // the current ground speed
    float command_course;// the command course angle
    uint64_t Time;// current GPS time(microseconds)
    int parity; // parity bit
};
```

Based on the SITL and HITL simulations in Sect. 5.4, the formation of real flight experiments could be achieved after tuning the parameters according to the platform we used. The designed groundspeed of the leader is 18m/s. The wind speed during the test was 4-6 m/s and the wind direction was northwest at the beginning of the test, but it changed to west during the test. The testing path is a quadrangle with perimeter of around 5.5km. The formation shape is "V" (1 leader and 4 followers) and the gaps between leader and followers are (-20, -20), (20, -20), (-40, -40), and (40, -40) respectively. The time step size k is defined as 1 and the loop times n is 6. The flight paths and distance error analysis results are shown in Figures 6.6 to 6.10.



97.8 m ↑-0.6 m/s AirSpd 18.4 m/s 00:43:15
442.7 m →18.6 m/s Thr 64 % 41197.9 m

Continue the mission from the current waypoint.

Figure 6.5: The screenshot of the formation flight with 5 UAVs

6.4.2. IMPORTANCE OF THE ESTIMATION ALGORITHM

To assess the impact of the proposed improvement, we used the standard vector field formation control algorithm (cf. Sect. 5.3.2) as a basic scenario. We tested a formation with 2 UAVs and the gap was set as (20, -20). Figure 6.11 shows the results of the flight testing.

Because of the communication delay, the steady-state error of the standard vector field formation control algorithm could not be reduced to less than 10m and the RMS of the error is 33.79m. The main reason is that the calculated desired path always lags behind the

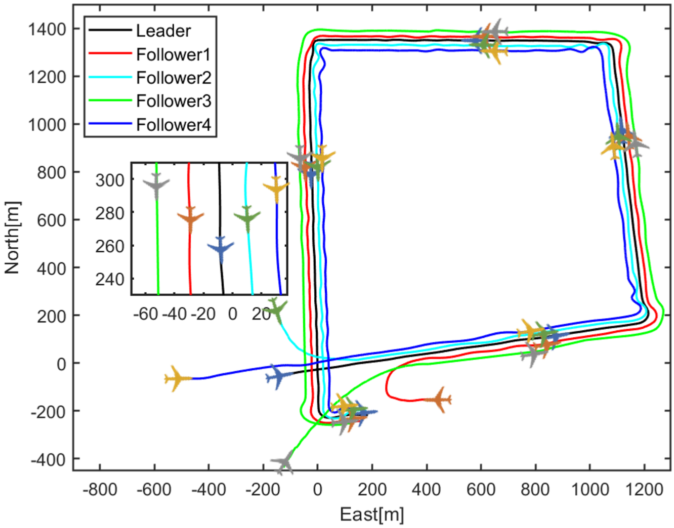


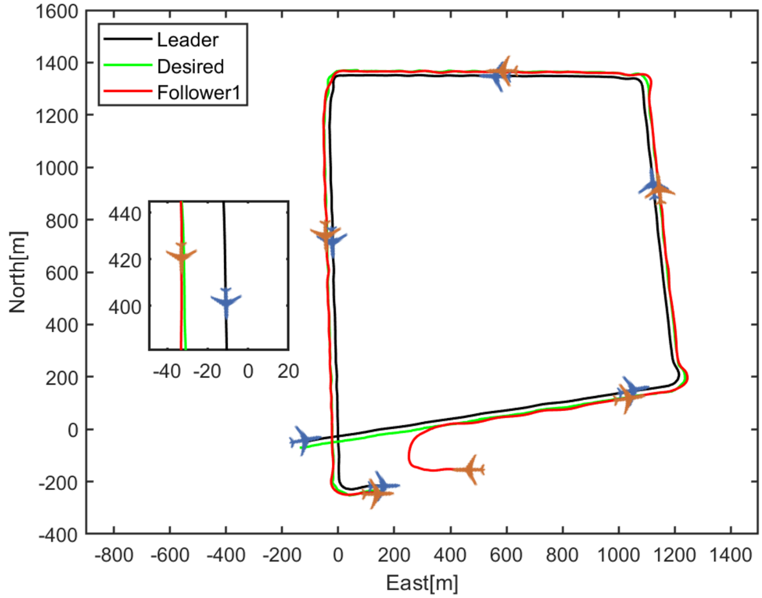
Figure 6.6: "V" formation flight along the quadrangle mixed path.

real desired position due to the communication delay. Another reason is that the desired path cannot be updated regularly under the time-varying time delay condition. Hence, it may even happen that the follower crosses the desired position and starts orbiting around it. However, we can see from the figure that the error reduces to around 10m after 280s. That is because the communication becomes better since the UAVs are close to the ground station. Therefore, this test proves that the standard formation control algorithm has a good tracking ability in an ideal environment but cannot handle the communication uncertainties.

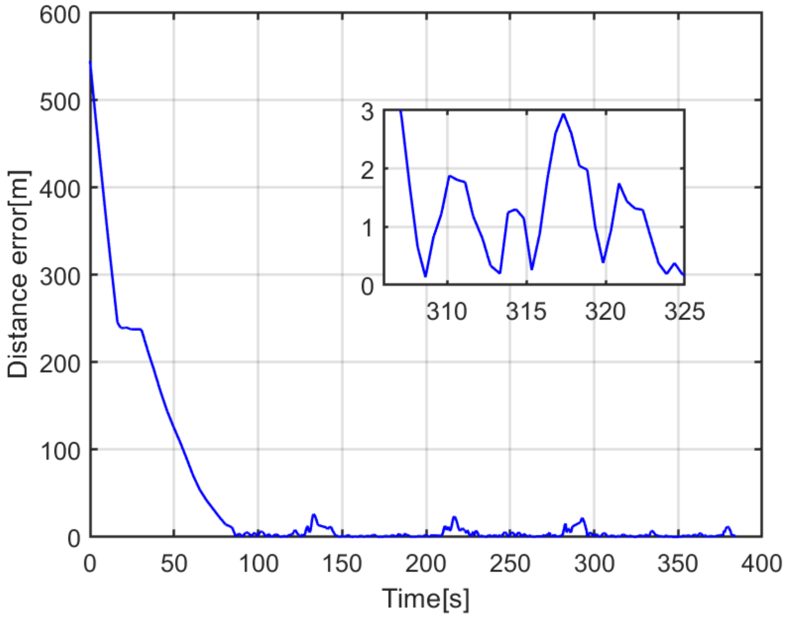
6.4.3. Comparison

For comparison purposes, we tested the state-of-the-art method based on unicycle dynamics [107], which has also been considered in Sect. 5.4. Because the follower with the unicycle method tries to imitate the course of the leader, the follower typically need a long flying path before converging. Hence, in this section, we only test a formation with 2 UAVs. The leader's path is the same as before and the gap is (-20, -20)m. The flight path and error analysis are shown in Figure 6.12.

The distance error results are summarized in Table 6.1. We can find that the error of the proposed vector field method is from 3 to 5m, which is much better than the state of the art method based on unicycle dynamics. The performance of the formation vector field method without the dead reckoning estimation is better than that of the state-of-the-art method: however, it can still happen that the follower orbits around some point due to communication loss, or oscillates along the expected path if the time delay is large. One may observe that the state-of-the-art method based on unicycle dynamics does not seem to be influenced by the communication time delay. But this is only because the follower's position is very far from the leader's position, the follower cannot reach the expected position even

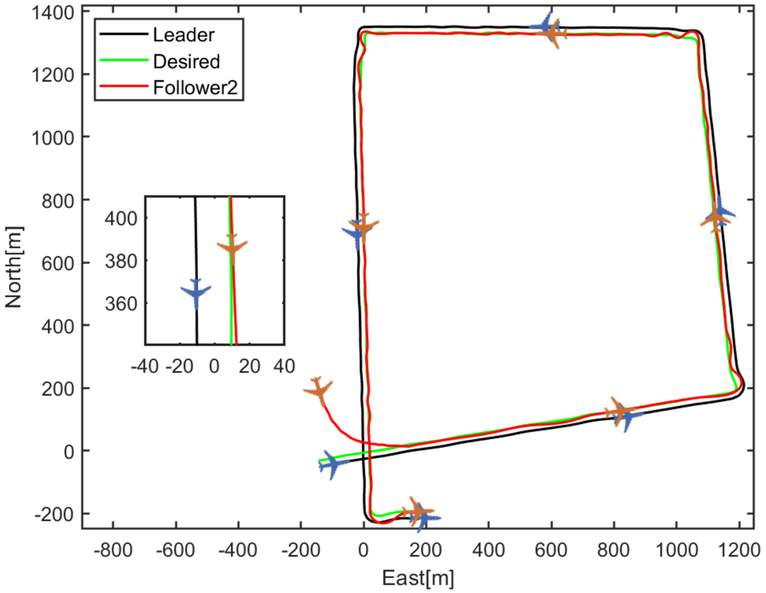


(a) The flight path of follower 1.

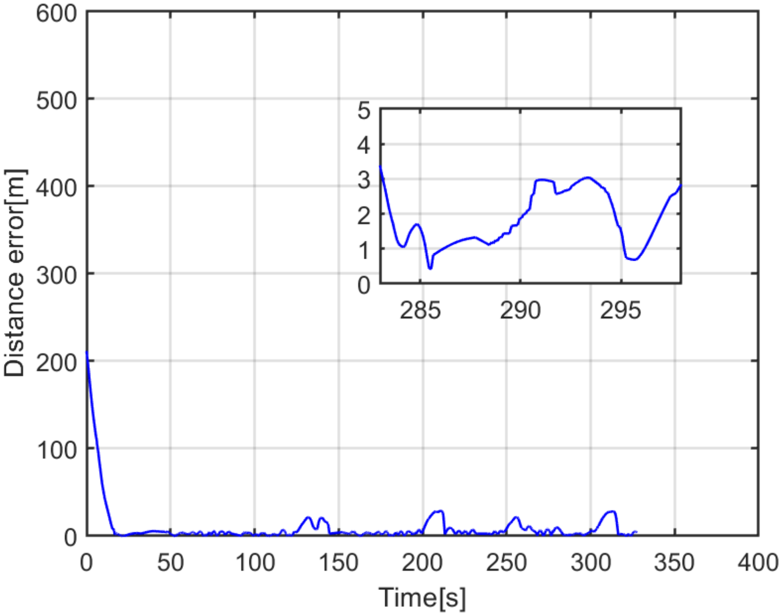


(b) Follower 1 formation error.

Figure 6.7: Performance of follower 1 pin the formation flight

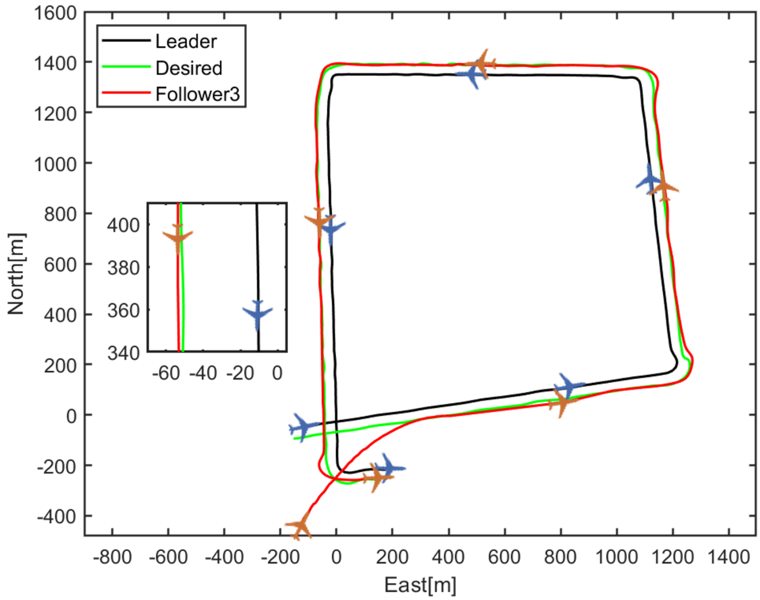


(a) The flight path of follower 2.

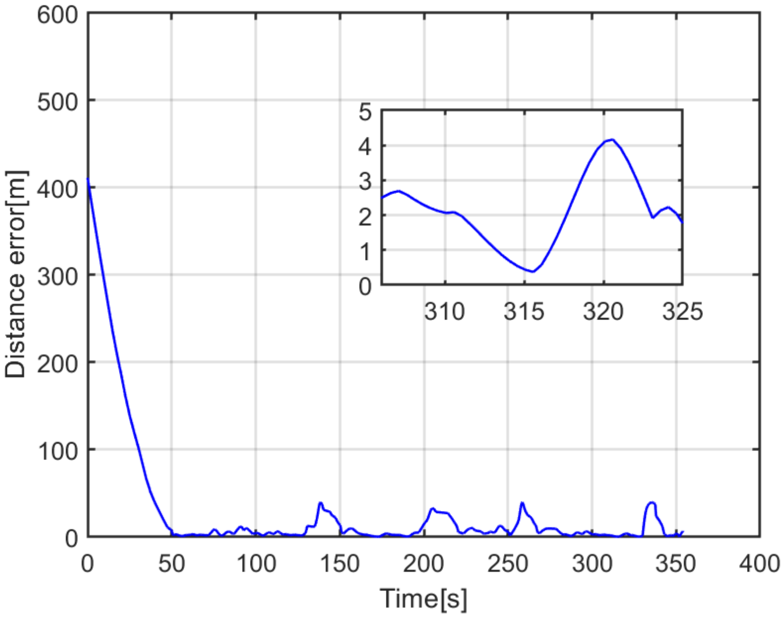


(b) Follower 2 formation error.

Figure 6.8: Performance of follower 2 in the formation flight

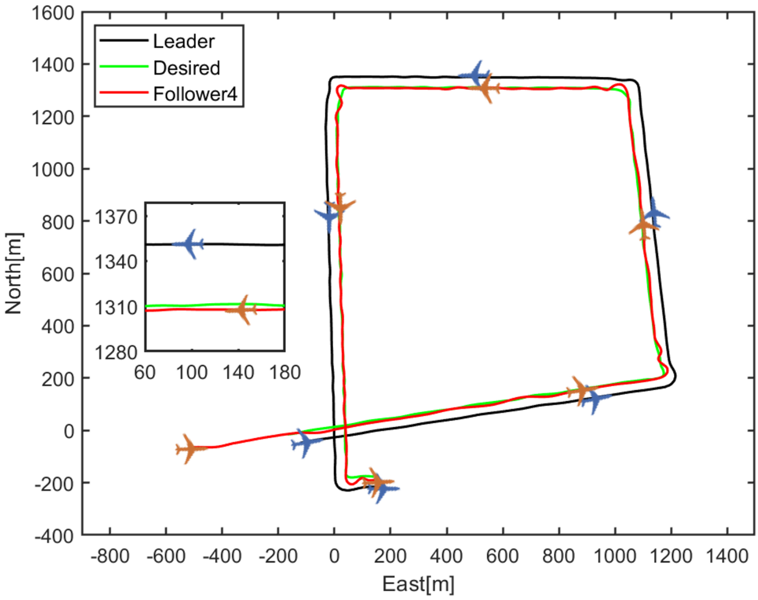


(a) The flight path of follower 3.

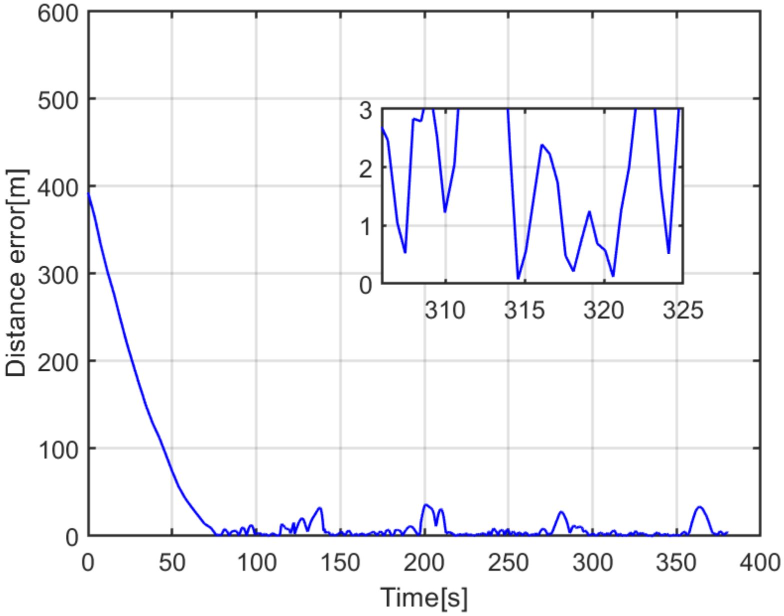


(b) Follower 3 formation error.

Figure 6.9: Performance of follower 3 in the formation flight

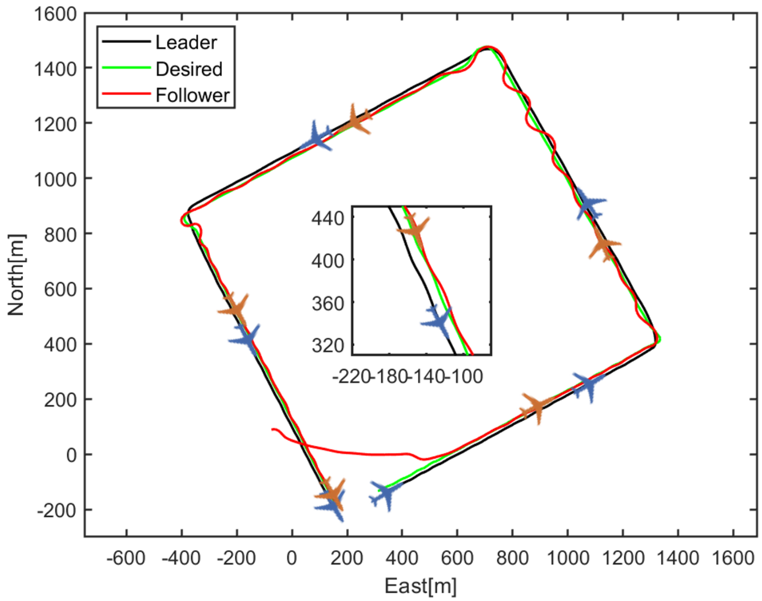


(a) The flight path of follower 4.

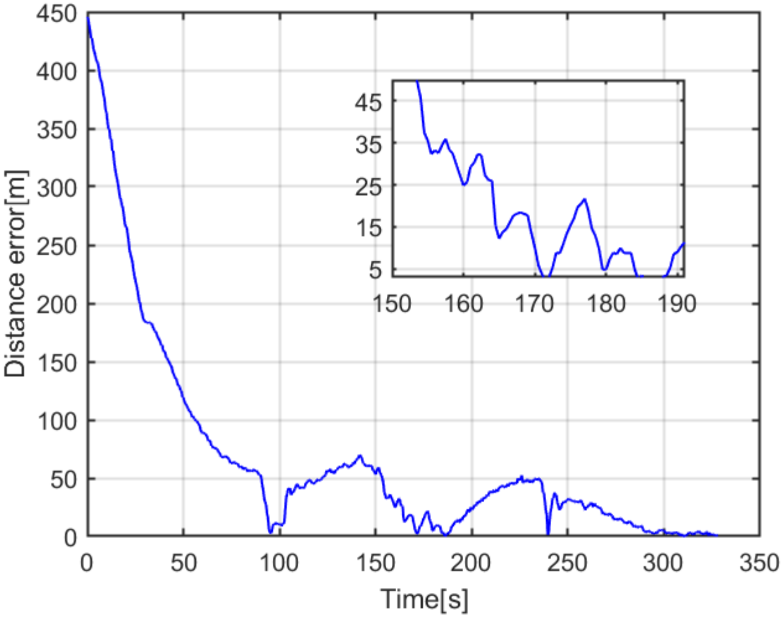


(b) Follower 4 formation error.

Figure 6.10: Performance of follower 4 performance in the formation flight

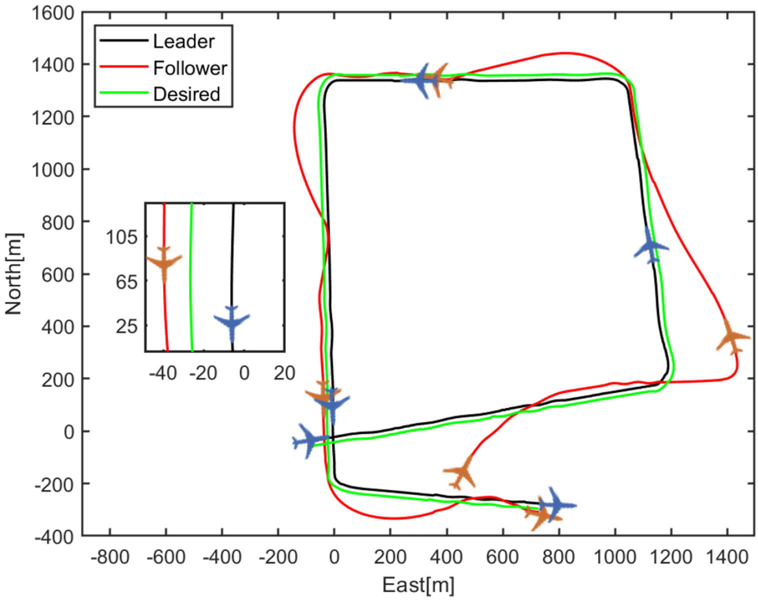


(a) The flight path of standard formation vector field.

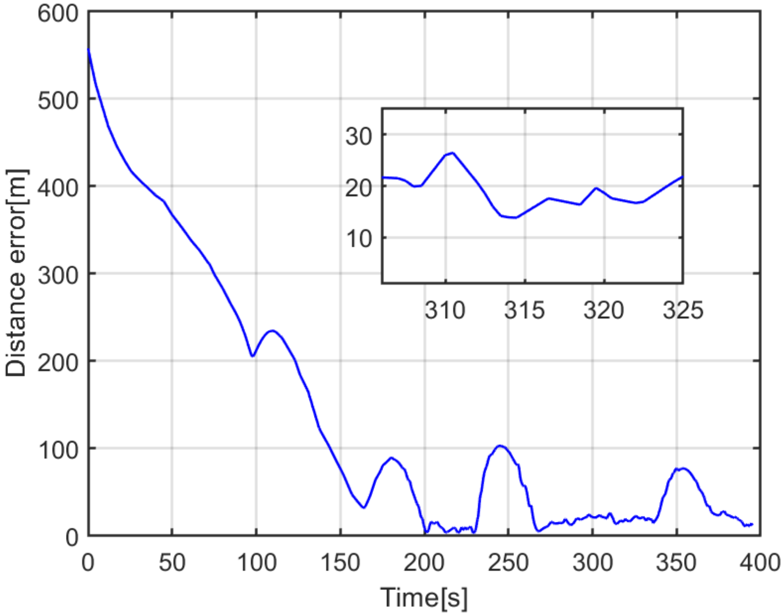


(b) Formation error.

Figure 6.11: The performance of the standard formation vector field



(a) The flight path of unicycle method's formation.



(b) Formation error.

Figure 6.12: The performance of the state-of-the-art method assuming unicycle dynamic

with 2-3s of communication loss.

Table 6.1: RMS formation error at steady-state (after convergence)

	Mixed path RMS error
Vector Field formation with DR estimation	5.51m, 9.29m, 12.14m, 10.82m
Vector Field formation without DR estimation	33.72m(**+264%**)
Unicycle method	43.49m (**+369%**)

6.5. SUMMARY

In this chapter, we have introduced an improvement in the vector field formation control algorithm, with the aim to counteract communication delay problems in real world testing. We have tested the proposed method in a formation flight with 5 UAVs experiment. To achieve sim-to-real transfer, a distributed estimator was presented for dealing with the communication delay uncertainties. As the computational complexity of the estimation mechanism is $O(1)$, the algorithm can be implemented in the embedded system easily. The testing results have proven that the improved vector-field formation control law has higher performance than that of original formation vector field method and unicycle method in terms of short-distance UAVs formation error.

7

CONCLUSIONS AND RECOMMENDATION

In this chapter, the main results of this thesis are summarized and some recommendations for future research are mentioned.

7.1. CONCLUSIONS

This thesis has presented several adaptive guidance control approaches for single and multi fixed-wing UAVs. The conclusions of this research can be summarized as follows:

- **Adaptive mechanisms can improve the autonomy of vector-field guidance laws for fixed-wing UAVs**

  A good guidance law is expected to maintain the path following accuracy of the fixed-wing UAV under possibly time-varying uncertainties. To meet this expectation, Chapter 3 has introduced two adaptive guidance methods that guarantee path following in the presence of unknown wind vector and unmodelled course angle dynamics. In particular, we have presented an estimation mechanism to handle the unknown wind vector. Based on it, we could achieve a vector-field path following approach without a priori knowledge of the wind amplitude/direction. We further improved this guidance method by adopting an uncertainty framework that the similar to the one of adaptive sliding mode. Based on this framework, we could achieve a vector field path following approach without structural knowledge of the unmodelled course angle dynamics coupling effects and without a priori knowledge of the course time constant and of the wind amplitude/direction.

- **Distributed model reference adaptive control can be used for formation control of fixed-wing UAVs**

  We have proposed an adaptive formation algorithm in Chapter 4 for networks of uncertain Euler–Lagrange systems and we have considered the fixed-wing UAVs as a test case. This formation control method is based on a distributed version of model reference adaptive control. This distributed algorithm is added to synchronize uncertain heterogeneous agents with Euler–Lagrange dynamics. The idea behind the proposed distributed model reference adaptive control approach was to make each agent converge to the model defined by its neighbors. The presence of uncertainty, such as different masses and inertia, is compensated by adapting the control action. The effectiveness of the proposed methodology was verified via numerical simulations of several formation shapes of UAVs.

- **The vector field guidance approach for a single UAV can be extended into a vector field formation control approach for multiple UAVs**

  Although the vector-field-based guidance method is quite established for single UAVs, its use in a formation setting is far from established. We have extended the vector field method for formation control, allowing us to integrate and test this tool in established autopilot suites (such as ArduPilot and PX4). The stability of the proposed control law was derived analytically by introducing an appropriately defined Lyapunov function.

- **The proposed adaptive control tools can be easily integrated in established open-source autopilot suites, and tested in the real world to achieve Sim-to-Real transfer**

  All proposed methods have been validated in software-in-the-loop and hardware-in-the-loop comparative experiments. Also, with the aim for real flight testing, we have analyzed the uncertainties in the flight testing and provided the methods for handling them. Real-world testing has been achieved via formation flight testing for 5 fixed-wing UAVs. Comparisons with a state of the art method have proven the effectiveness of the proposed methods.

7.2. RECOMMENDATIONS FOR FUTURE RESEARCH

This research leaves the door open for several improvements. Thus, some recommendations for the possible directions of future research are given as follows:

- This research has shown the connections between adaptive vector field guidance and the framework of adaptive sliding mode control: further connecting the vector field idea to higher-order (adaptive) sliding mode control is an interesting topic for further study. The advantage of this idea could be to overcome classical problems associated to sliding mode control (e.g. the presence of chattering). The challenge is that higher-order (adaptive) sliding mode requires a different control design and stability analysis.

- It is intuitive to expect that when the autopilot layer is poorly tuned, any guidance algorithm can do little to cope with this situation. However, it is possible that, in such a poorly tuned scenario, an adaptive guidance algorithm can do better than a non-adaptive guidance algorithm. The challenge of this direction is that it is difficult to quantify in a mathematical way the "poorly tuned scenario", especially since the guidance problem is intrinsically nonlinear: therefore, investigating this point in an analytic or numerical way could be an interesting future work.

- Even in a formation control setting, the research described in this thesis has been mostly concerned with primitive path tracking missions. Extending the missions, such as handling more general curved paths, or tracking targets while achieving formation, are all interesting points for future research. The advantage of this idea could be to generalize or unify the vector field idea in a coherent control framework. The challenge is to design appropriate vector fields for such a coherent framework. Accordingly, this would require a different control design and stability analysis.

- Given the recent trend in artificial intelligence and machine learning, the design of estimation/compensation mechanisms in formation control based on reinforcement learning methods, e.g. trained during the formation flight, is another interesting point for future research. The challenge of this direction is that UAVs are safety-critical applications, i.e., if they are trained during the flight, one cannot allow a UAV to crash on the ground. Therefore, one should propose learning methods that preserve safety during the operation of the UAVs.

- Especially when considering large teams of UAVs, the risk of collisions among two or more vehicles should be reduced or counteracted. Thus, it is of interest to further improve the algorithm for UAV formations with collision avoidance capabilities.

- Automatic and cooperative path planing via multi-sensor fusion from multi heterogeneous unmanned vehicles is an interesting topic in the control field and also in the artificial intelligence field. The heterogeneous dynamics, the motion characteristics, and the risk of collisions should all be considered in such a path planning problem. It is expected that, as compared to traditional path planning, the cooperative path planning can improve position synchronization, attitude synchronization, and task-based coordination.

- The cyber security of multi-agent systems has been an increasingly popular research direction in research years. Clearly, it concerns all types of unmanned vehicles including UAVs. How to guarantee cyber-secure communication among UAVs, or how to detect (and counteract) cyber security threats are all relevant and largely open questions. The challenge is especially clear in UAVs, since they have limits in their on-board payload and in their on-board computational capabilities (such limits are less evident, e.g. in ground unmanned vehicles).

- How to guarantee safe coordination as the network of UAVs becomes larger and larger is also another open question. One can think about birds that can safely coordinate no matter how many members in the group. Whereas similar coordination capabilities are yet not possible for UAVs. A typical method to ensure coordination could be network partition. A large scale network will can split into several sub-networks. How to control the combined network consistence by regardless the edges connecting should be further developed. However, it is not a trivial task how to define such networks and how to dynamically change their shape, if needed. These are all interesting questions deserving future work.

BIBLIOGRAPHY

[1] W. Zhang, M. S. Branicky, and S. M. Phillips, “Stability of networked control systems,” *IEEE Control Systems Magazine*, vol. 21, no. 1, pp. 84–99, Feb 2001.

[2] L. Zhang, H. Gao, and O. Kaynak, “Network-induced constraints in networked control systems—a survey,” *IEEE Transactions on Industrial Informatics*, vol. 9, no. 1, pp. 403–416, 2012.

[3] A. Liu, W.-a. Zhang, L. Yu, S. Liu, and M. Z. Chen, “New results on stabilization of networked control systems with packet disordering,” *Automatica*, vol. 52, pp. 255–259, 2015.

[4] P. A. Ioannou and J. Sun, *Robust Adaptive Control*. PTR Prentice-Hall Upper Saddle River, NJ, 1996.

[5] M. Bodson and J. E. Groszkiewicz, “Multivariable adaptive algorithms for reconfigurable flight control,” *IEEE Transactions on Control Systems Technology*, vol. 5, no. 2, pp. 217–229, 1997.

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

[7] 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.

[8] F. Gavilan, R. Vazquez, and S. Esteban, “Trajectory tracking for fixed-wing UAV using model predictive control and adaptive backstepping,” *1st IFAC Workshop on Advanced Control and Navigation for Autonomous Aerospace Vehicles (ACNAAV’15)*, pp. 132–137, 2015.

[9] A. P. Aguiar, J. P. Hespanha, and P. V. Kokotović, “Performance limitations in reference tracking and path following for nonlinear systems,” *Automatica*, vol. 44, no. 3, pp. 598–610, 2008.

[10] D. Invernizzi and M. Lovera, “Trajectory tracking control of thrust-vectoring UAVs,” *Automatica*, vol. 95, pp. 180–186, 2018.

[11] S. Ghapani, J. Mei, W. Ren, and Y. Song, “Fully distributed flocking with a moving leader for Lagrange networks with parametric uncertainties,” *Automatica*, vol. 67, pp. 67–76, 2016.

[12] I. A. Azzollini, S. Baldi, and E. B. Kosmatopoulos, “Adaptive synchronization in networks with heterogeneous uncertain Kuramoto-like units,” in *2018 European Control Conference (ECC)*, 2018, pp. 2417–2422.

[13] S. Baldi, I. A. Azzollini, and E. B. Kosmatopoulos, “A distributed disagreement-based protocol for synchronization of uncertain heterogeneous agents,” in *2018 European Control Conference (ECC)*, 2018, pp. 2411–2416.

[14] S. Baldi and P. Frasca, “Leaderless synchronization of heterogeneous oscillators by adaptively learning the group model,” *IEEE Transactions on Automatic Control*, 2019.

[15] E. Nuno, R. Ortega, L. Basanez, and D. Hill, “Synchronization of networks of nonidentical Euler-Lagrange systems with uncertain parameters and communication delays,” *IEEE Transactions on Automatic control*, vol. 56, no. 4, pp. 935–941, 2011.

[16] R. W. Beard and T. W. McLain, *Small Unmanned Aircraft: Theory and Practice*. Princeton University Press, 2012.

[17] R. F. Stengel, *Flight Dynamics*. Princeton University Press, 2015.

[18] T. R. Kane and D. A. Levinson, *Dynamics, Theory and Applications*. McGraw Hill, 1985.

[19] P. H. Zipfel, *Modeling and Simulation of Aerospace Vehicle Dynamics (2nd Edition)*. American Institute of Aeronautics and Astronautics, 2007.

[20] T. Yamasaki, H. Takano, and Y. Baba, “Robust path-following for UAV using pure pursuit guidance,” in *Aerial Vehicles*. IntechOpen, 2009.

[21] G. Ambrosino, M. Ariola, U. Ciniglio, F. Corraro, E. De Lellis, and A. Pironti, “Path generation and tracking in 3-d for UAVs,” *IEEE Transactions on Control Systems Technology*, vol. 17, no. 4, pp. 980–988, 2009.

[22] N. Cho and Y. Kim, “Optimality of augmented ideal proportional navigation for maneuvering target interception,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 52, no. 2, pp. 948–954, 2016.

[23] M. Kothari, I. Postlethwaite, and D.-W. Gu, “UAV path following in windy urban environments,” *Journal of Intelligent & Robotic Systems*, vol. 74, no. 3-4, pp. 1013–1028, 2014.

[24] N. Cho and Y. Kim, “Three-dimensional nonlinear differential geometric path-following guidance law,” *Journal of Guidance, Control, and Dynamics*, vol. 38, no. 12, pp. 948–954, 2015.

[25] H. Chao, Y. Cao, and Y. Chen, “Autopilots for small unmanned aerial vehicles: a survey,” *International Journal of Control, Automation and Systems*, vol. 8, no. 1, pp. 36–44, 2010.

[26] J. Chang, J. Cieslak, J. Davila, A. Zolghadri, and J. Zhou, “Analysis and design of second-order sliding-mode algorithms for quadrotor roll and pitch estimation,” *ISA Transactions*, vol. 71, pp. 495–512, 2017.

[27] G. Casadei, L. Furieri, N. Mimmo, R. Naldi, and L. Marconi, “Internal model-based control for loitering maneuvers of UAVs,” in *2016 European Control Conference (ECC)*, 2016, pp. 672–677.

[28] D. Invernizzi, P. Panizza, F. Riccardi, S. Formentin, and M. Lovera, “Data-driven attitude control law of a variable-pitch quadrotor: a comparison study,” *20th IFAC Symposium on Automatic Control in Aerospace (ACA 2016)*, vol. 49, no. 17, pp. 236 – 241, 2016.

[29] V. M. Goncalves, L. C. A. Pimenta, C. A. Maia, B. C. O. Dutra, and G. A. S. Pereira, “Vector fields for robot navigation along time-varying curves in n-dimensions,” *IEEE Transactions on Robotics*, vol. 26, no. 4, pp. 647–659, 2010.

[30] D. V. Dimarogonas, “Sufficient conditions for decentralized potential functions based controllers using canonical vector fields,” *IEEE Transactions on Automatic Control*, vol. 57, no. 10, pp. 2621–2626, 2012.

[31] H. Chen, K. Chang, and C. S. Agate, “UAV path planning with tangent-plus-Lyapunov vector field guidance and obstacle avoidance,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 49, no. 2, pp. 840–856, 2013.

[32] W. Jiang, D. Wang, Y. Wang, and Z. A. Ali, “UAV rendezvous based on time-varying vector fields,” *Electronics Letters*, vol. 53, no. 10, pp. 653–655, 2017.

[33] J. L. G. Olavo, G. D. Thums, T. A. Jesus, L. C. de Araújo Pimenta, L. A. B. Torres, and R. M. Palhares, “Robust guidance strategy for target circulation by controlled UAV,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 54, no. 3, pp. 1415–1431, 2018.

[34] J. W. Langelaan, N. Alley, and J. Neidhoefer, “Wind field estimation for small unmanned aerial vehicles,” *Journal of Guidance Control and Dynamics*, vol. 34, no. 4, p. 1016, 2011.

[35] J. Gutmann, E. Eade, P. Fong, and M. E. Munich, “Vector field SLAM—localization by learning the spatial variation of continuous signals,” *IEEE Transactions on Robotics*, vol. 28, no. 3, pp. 650–667, 2012.

[36] J. Kwon and D. Chwa, “Hierarchical formation control based on a vector field method for wheeled mobile robots,” *IEEE Transactions on Robotics*, vol. 28, no. 6, pp. 1335–1345, 2012.

[37] B. L. Stevens, F. L. Lewis, and E. N. Johnson, *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems, 3rd edition*. John Wiley & Sons, Inc., 2015.

[38] H. K. Khalil, *Nonlinear Systems*, 2002.

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

[40] P. Poksawat, L. Wang, and A. Mohamed, "Gain scheduled attitude control of fixed-wing UAV with automatic controller tuning," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 4, pp. 1192–1203, 2018.

[41] J. Chang, J. Cieslak, J. Dávila, J. Zhou, A. Zolghadri, and Z. Guo, "A two-step approach for an enhanced quadrotor attitude estimation via IMU data," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 3, pp. 1140–1148, 2017.

[42] L. Fusini, T. I. Fossen, and T. A. Johansen, "Nonlinear observers for GNSS- and camera-aided inertial navigation of a fixed-wing UAV," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 5, pp. 1884–1891, 2018.

[43] P. Marantos, Y. Koveos, and K. J. Kyriakopoulos, "UAV state estimation using adaptive complementary filters," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 4, pp. 1214–1226, 2016.

[44] L. Fusini, T. A. Johansen, and T. I. Fossen, "Dead reckoning of a fixed-wing UAV with inertial navigation aided by optical flow," in *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2017, pp. 1250–1259.

[45] "Aerospace block-set Simulink," 2019. [Online]. Available: https://nl.mathworks.com/help/aeroblks/index.html

[46] A. Ul Haque, W. Asrar, A. A. Omar, E. Sulaeman, and J. Ali, "Comparison of digital DATCOM and wind tunnel data of a winged hybrid airship's generic model," in *Applied Mechanics and Materials*, vol. 629, 2014, pp. 36–41.

[47] J. Yang, X. Wang, S. Baldi, S. Singh, and S. Fari, "A software-in-the-loop implementation of adaptive formation control for fixed-wing uavs," *IEEE/CAA Journal of Automatica Sinica*, vol. 6, no. 5, pp. 1230–1239, 2019.

[48] S. Fari, X. Wang, S. Roy, and S. Baldi, "Addressing unmodelled path-following dynamics via adaptive vector field: a UAV test case," *IEEE Transactions on Aerospace and Electronic Systems*, 2019.

[49] X. Wang, S. Roy, S. Farì, and S. Baldi, "The problem of reliable design of vector-field path following in the presence of uncertain course dynamics," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 9399–9404, 2020, 21st IFAC World Congress.

[50] S. Roy, S. B. Roy, and I. N. Kar, "Adaptive-robust control of Euler-Lagrange systems with linearly parametrizable uncertainty bound," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 5, pp. 1842–1850, 2018.

[51] MathWorks, "Dryden wind turbulence model (discrete)," https://nl.mathworks.com/help/aeroblks/drydenwindturbulencemodeldiscrete.html, accessed July 18, 2022.

[52] X. Wang, S. Roy, S. Farì, and S. Baldi, “Adaptive vector field guidance without a priori knowledge of course dynamics and wind,” *IEEE/ASME Transactions on Mechatronics*, pp. 1–11, 2022.

[53] Y. Shtessel, M. Taleb, and F. Plestan, “A novel adaptive-gain supertwisting sliding mode controller: Methodology and application,” *Automatica*, vol. 48, no. 5, pp. 759–769, 2012.

[54] K. Lu and Y. Xia, “Finite-time attitude control for rigid spacecraft-based on adaptive super-twisting algorithm,” *IET Control Theory & Applications*, vol. 8, no. 15, pp. 1465–1477, 2014.

[55] C. Edwards and Y. B. Shtessel, “Adaptive continuous higher order sliding mode control,” *Automatica*, vol. 65, pp. 183–190, 2016.

[56] H. Obeid, L. M. Fridman, S. Laghrouche, and M. Harmouche, “Barrier function-based adaptive sliding mode control,” *Automatica*, vol. 93, pp. 540–544, 2018.

[57] V. I. Utkin and A. S. Poznyak, “Adaptive sliding mode control with application to super-twist algorithm: Equivalent control method,” *Automatica*, vol. 49, no. 1, pp. 39–47, 2013.

[58] W. Yu, G. Chen, and J. Lu, “On pinning synchronization of complex dynamical networks,” *Automatica*, vol. 45, no. 2, pp. 429 – 435, 2009.

[59] W. Yu, P. DeLellis, G. Chen, M. di Bernardo, and J. Kurths, “Distributed adaptive control of synchronization in complex networks,” *IEEE Transactions on Automatic Control*, vol. 57, no. 8, pp. 2153–2158, 2012.

[60] W. Ren, R. W. Beard, and A. W. Beard, “Decentralized scheme for spacecraft formation flying via the virtual structure approach,” *AIAA Journal of Guidance, Control, and Dynamics*, vol. 27, pp. 73–82, 2003.

[61] V. Lesser, M. Tambe, and C. L. Ortiz, Eds., *Distributed Sensor Networks: A Multiagent Perspective*. Norwell, MA, USA: Kluwer Academic Publishers, 2003.

[62] I. T. Michailidis, T. Schild, R. Sangi, P. Michailidis, C. Korkas, J. Futterer, D. Muller, and E. B. Kosmatopoulos, “Energy-efficient HVAC management using cooperative, self-trained, control agents: A real-life German building case study,” *Applied Energy*, vol. 211, pp. 113 – 125, 2018.

[63] G. Wen, X. Yu, Z. W. Liu, and W. Yu, “Adaptive consensus-based robust strategy for economic dispatch of smart grids subject to communication uncertainties,” *IEEE Transactions on Industrial Informatics*, vol. 14, no. 6, pp. 2484–2496, 2018.

[64] M. Jun and R. D’Andrea, *Path Planning for Unmanned Aerial Vehicles in Uncertain and Adversarial Environments*. Boston, MA: Springer US, 2003, pp. 95–110.

[65] 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.

[66] D. O. Popa, A. C. Sanderson, R. J. Komerska, S. S. Mupparapu, D. R. Blidberg, and S. G. Chappel, “Adaptive sampling algorithms for multiple autonomous underwater vehicles,” in *2004 IEEE/OES Autonomous Underwater Vehicles*, 2004, pp. 108–118.

[67] P. Frasca, R. Carli, F. Fagnani, and S. Zampieri, “Average consensus on networks with quantized communication,” *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal*, vol. 19, no. 16, pp. 1787–1816, 2009.

[68] S. He, G. Yi, and Z. Wu, “Exponential synchronization of dynamical network with distributed delays via intermittent control,” *Asian Journal of Control*, vol. 21, no. 5, pp. 2378–2386, 2019.

[69] Y. Cao, W. Yu, W. Ren, and G. Chen, “An overview of recent progress in the study of distributed multi-agent coordination,” *IEEE Transactions on Industrial informatics*, vol. 9, no. 1, pp. 427–438, 2012.

[70] R. Ortega and M. W. Spong, “Adaptive motion control of rigid robots: A tutorial,” *Automatica*, vol. 25, no. 6, pp. 877–888, 1989.

[71] G. Tao, *Adaptive Control Design and Analysis*. John Wiley & Sons, 2003.

[72] K. Ogata, *Modern Control Engineering*, ser. Instrumentation and controls series. Prentice Hall, 2010.

[73] G. Tao, *Adaptive Control Design and Analysis (Adaptive and Learning Systems for Signal Processing, Communications and Control Series)*. New York, NY, USA: John Wiley & Sons, Inc., 2003.

[74] P. Ioannou and B. Fidan, *Adaptive Control Tutorial (Advances in Design and Control)*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2006.

[75] G. Tao, “Multivariable adaptive control: A survey,” *Automatica*, vol. 50, no. 11, pp. 2737–2764, 2014.

[76] S. Baldi and P. Frasca, “Adaptive synchronization of unknown heterogeneous agents: An adaptive virtual model reference approach,” *Journal of the Franklin Institute*, vol. 356, no. 2, pp. 935–955, 2019, special Issue on Modeling, Analysis and Control of Networked Autonomous Agents.

[77] Y. Abou Harfouch, S. Yuan, and S. Baldi, “An adaptive switched control approach to heterogeneous platooning with inter-vehicle communication losses,” *20th IFAC World Congress, Toulouse, France*, pp. 1382–1387, 2017.

[78] A. Atyabi, S. MahmoudZadeh, and S. Nefti-Meziani, “Current advancements on autonomous mission planning and management systems: An AUV and UAV perspective,” *Annual Reviews in Control*, vol. 46, pp. 196–215, 2018.

[79] T. Konrad, J.-J. Gehrt, J. Lin, R. Zweigel, and D. Abel, “Advanced state estimation for navigation of automated vehicles,” *Annual Reviews in Control*, vol. 46, pp. 181–195, 2018.

[80] S. Zhao, X. Wang, Z. Lin, D. Zhang, and L. Shen, “Integrating vector field approach and input-to-state stability curved path following for unmanned aerial vehicles,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 8, pp. 2897–2904, 2020.

[81] W. Yao, H. G. de Marina, B. Lin, and M. Cao, “Singularity-free guiding vector field for robot navigation,” *IEEE Transactions on Robotics*, vol. 37, no. 4, pp. 1206–1221, 2021.

[82] H. G. de Marina, Z. Sun, M. Bronz, and G. Hattenberger, “Circular formation control of fixed-wing uavs with constant speeds,” in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 5298–5303.

[83] T. Z. Muslimov and R. A. Munasypov, “Consensus-based cooperative circular formation control strategy for multi-UAV system,” in *2019 International Russian Automation Conference (RusAutoCon)*, 2019, pp. 1–8.

[84] R. Singh and B. Bhushan, “Evolving intelligent system for trajectory tracking of unmanned aerial vehicles,” *IEEE Transactions on Automation Science and Engineering*, pp. 1–14, 2021.

[85] J. Wang and M. Xin, “Integrated optimal formation control of multiple unmanned aerial vehicles,” *IEEE Transactions on Control Systems Technology*, vol. 21, no. 5, pp. 1731–1744, 2013.

[86] W. Lin, “Distributed UAV formation control using differential game approach,” *Aerospace Science and Technology*, vol. 35, pp. 54–62, 2014.

[87] O. Thakoor, J. Garg, and R. Nagi, “Multiagent UAV routing: A game theory analysis with tight price of anarchy bounds,” *IEEE Transactions on Automation Science and Engineering*, vol. 17, no. 1, pp. 100–116, 2020.

[88] Y. Liu, J. M. Montenbruck, D. Zelazo, M. Odelga, S. Rajappa, H. H. Bülthoff, F. Allgöwer, and A. Zell, “A distributed control approach to formation balancing and maneuvering of multiple multirotor UAVs,” *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 870–882, 2018.

[89] J. Zhang, J. Yan, and P. Zhang, “Multi-UAV formation control based on a novel back-stepping approach,” *IEEE Transactions on Vehicular Technology*, vol. 69, no. 3, pp. 2437–2448, 2020.

[90] Z. Zhen, G. Tao, Y. Xu, and G. Song, “Multivariable adaptive control based consensus flight control system for UAVs formation,” *Aerospace Science and Technology*, vol. 93, p. 105336, 2019.

[91] D. Yue, S. Baldi, J. Cao, Q. Li, and B. De Schutter, “A directed spanning tree adaptive control solution to time-varying formations,” *IEEE Transactions on Control of Network Systems*, vol. 8, no. 2, pp. 690–701, 2021.

[92] Y. Zou and Z. Meng, “Coordinated trajectory tracking of multiple vertical take-off and landing UAVs,” *Automatica*, vol. 99, pp. 33–40, 2019.

[93] J. Wang, Z. Zhou, C. Wang, and Z. Ding, “Cascade structure predictive observer design for consensus control with applications to UAVs formation flying,” *Automatica*, vol. 121, p. 109200, 2020.

[94] X. Dong, B. Yu, Z. Shi, and Y. Zhong, “Time-varying formation control for unmanned aerial vehicles: Theories and applications,” *IEEE Transactions on Control Systems Technology*, vol. 23, no. 1, pp. 340–348, 2015.

[95] Y. Zou, Z. Zhou, X. Dong, and Z. Meng, “Distributed formation control for multiple vertical takeoff and landing UAVs with switching topologies,” *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 4, pp. 1750–1761, 2018.

[96] J. Wang, L. Han, X. Dong, Q. Li, and Z. Ren, “Distributed sliding mode control for time-varying formation tracking of multi-UAV system with a dynamic leader,” *Aerospace Science and Technology*, vol. 111, p. 106549, 2021.

[97] X. Dong, Y. Hua, Y. Zhou, Z. Ren, and Y. Zhong, “Theory and experiment on formation-containment control of multiple multirotor unmanned aerial vehicle systems,” *IEEE Transactions on Automation Science and Engineering*, vol. 16, no. 1, pp. 229–240, 2019.

[98] S. Rathinam, R. Sengupta, and S. Darbha, “A resource allocation algorithm for multivehicle systems with nonholonomic constraints,” *IEEE Transactions on Automation Science and Engineering*, vol. 4, no. 1, pp. 98–104, 2007.

[99] J. Zhang, J. Yan, and P. Zhang, “Fixed-wing UAV formation control design with collision avoidance based on an improved artificial potential field,” *IEEE Access*, vol. 6, pp. 78 342–78 351, 2018.

[100] F. Liao, R. Teo, J. L. Wang, X. Dong, F. Lin, and K. Peng, “Distributed formation and reconfiguration control of VTOL UAVs,” *IEEE Transactions on Control Systems Technology*, vol. 25, no. 1, pp. 270–277, 2017.

[101] Y. Wang, M. Shan, and D. Wang, “Motion capability analysis for multiple fixed-wing UAV formations with speed and heading rate constraints,” *IEEE Transactions on Control of Network Systems*, vol. 7, no. 2, pp. 977–989, 2020.

[102] T. Z. Muslimov and R. A. Munasypov, “Adaptive decentralized flocking control of multi-UAV circular formations based on vector fields and backstepping,” *ISA Transactions*, vol. 107, pp. 143–159, 2020.

[103] ——, “Consensus-based cooperative control of parallel fixed-wing UAV formations via adaptive backstepping,” *Aerospace Science and Technology*, vol. 109, p. 106416, 2021.

[104] PX4, “Gazebo simulation,” https://docs.px4.io/master/en/simulation/gazebo.html Accessed July, 2022.

[105] ——, "PX4-SITL_gazebo, plane.sdf.jinja," https://github.com/PX4/PX4-SITL_gazebo/blob/e580bbcd1eb6902c658ed3ece3b3b28dfd57eb17/models/plane/plane.sdf.jinja Accessed July, 2022.

[106] Gazebo, "Gazebo: Tutorials," https://gazebosim.org/tutorials Accessed July, 2022.

[107] D. Kostić, S. Adinandra, J. Caarls, N. van de Wouw, and H. Nijmeijer, "Collision-free tracking control of unicycle mobile robots," in *Proceedings of the 48h IEEE Conference on Decision and Control (CDC)*, 2009, pp. 5667–5672.

[108] D. Tardioli, R. Parasuraman, and P. Ögren, "Pound: A multi-master ROS node for reducing delay and jitter in wireless multi-robot networks," *Robotics and Autonomous Systems*, vol. 111, pp. 73–87, 2019.

[109] R. Bottura, D. Babazadeh, K. Zhu, A. Borghetti, L. Nordström, and C. A. Nucci, "SITL and HLA co-simulation platforms: Tools for analysis of the integrated ICT and electric power system," in *Eurocon 2013*, 2013, pp. 918–925.

[110] J. N. Gross, Y. Gu, and M. B. Rhudy, "Robust UAV relative navigation with DGPS, INS, and Peer-to-Peer radio ranging," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 3, pp. 935–944, 2015.

[111] S. Sirouspour and A. Shahdi, "Model predictive control for transparent teleoperation under communication time delay," *IEEE Transactions on Robotics*, vol. 22, no. 6, pp. 1131–1145, 2006.

[112] X. Wang, J. Zhang, D. Zhang, and L. Shen, "UAV formation: From numerical simulation to actual flight," in *2015 IEEE International Conference on Information and Automation*, 2015, pp. 475–480.

[113] T. H. Chung, M. R. Clement, M. A. Day, K. D. Jones, D. Davis, and M. Jones, "Live-fly, large-scale field experimentation for large numbers of fixed-wing uavs," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 1255–1262.

[114] W. Yuan, Q. Chen, Z. Hou, and Y. Li, "Multi-UAVs formation flight control based on leader-follower pattern," in *2017 36th Chinese Control Conference (CCC)*, 2017, pp. 1276–1281.

LIST OF PUBLICATIONS

JOURNAL ARTICLES

1. X. Wang, S. Roy, S. Farì and S. Baldi, "Adaptive Vector Field Guidance Without a Priori Knowledge of Course Dynamics and Wind," in IEEE/ASME Transactions on Mechatronics, scheduled for publication, 2022, doi: 10.1109/TMECH.2022.3160480.
2. X. Wang, S. Baldi, X. Feng, C. Wu, H. Xie and B. De Schutter, "A Fixed-Wing UAV Formation Algorithm Based on Vector Field Guidance," in IEEE Transactions on Automation Science and Engineering, scheduled for publication, 2022, doi: 10.1109/TASE.2022.3144672.
3. J. Yang, X. Wang, S. Baldi, S. Singh and S. Farì, "A software-in-the-loop implementation of adaptive formation control for fixed-wing UAVs," in IEEE/CAA Journal of Automatica Sinica, vol. 6, no. 5, pp. 1230-1239, September 2019, doi: 10.1109/JAS.2019.1911702. (co-first author)
4. S. Farí, X. Wang, S. Roy and S. Baldi, "Addressing Unmodeled Path-Following Dynamics via Adaptive Vector Field: A UAV Test Case," in IEEE Transactions on Aerospace and Electronic Systems, vol. 56, no. 2, pp. 1613-1622, April 2020, doi: 10.1109/TAES.2019.2925487.

CONFERENCE PAPERS

1. X. Wang, S. Roy, S. Farì and S. Baldi, "The problem of reliable design of vector-field path following in the presence of uncertain course dynamics," in IFAC-PapersOnLine, 53 , 9399-9404, 2020, https://doi.org/10.1016/j.ifacol.2020.12.2409.

CURRICULUM VITÆ

Ximan Wang was born in 14th June, 1990, Taiyuan, Shanxi Province, China. He received the B.Sc. degree in Software Engineering from Taiyuan University of Technology, China, in 2014, and the M.Sc. degree in Advanced Control and Systems Engineering from University of Sheffield, UK, in 2016.

In October 2018, he joined the Delft Center for Systems and Controls as Ph.D. candidate and was sponsored by Chinese Scholarship Council (CSC). In his Ph.D. project, he worked on UAV adaptive control and UAVs formation control under the supervision of prof.dr. Simone Baldi and prof.dr.ir. Bart De Schutter.