



**HAL**  
open science

# Design, modeling and control of a convertible mini airplane having four tilting rotors

Gerardo Ramón Flores Colunga

► **To cite this version:**

Gerardo Ramón Flores Colunga. Design, modeling and control of a convertible mini airplane having four tilting rotors. Other. Université de Technologie de Compiègne, 2014. English. NNT: 2014COMP2095 . tel-01297721

**HAL Id: tel-01297721**

**<https://theses.hal.science/tel-01297721>**

Submitted on 4 Apr 2016

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Par **Gerardo Ramón FLORES COLUNGA**

*Design, modeling and control of a convertible mini airplane having four tilting rotors*

Thèse présentée  
pour l'obtention du grade  
de Docteur de l'UTC



Soutenue le 31 octobre 2014

**Spécialité** : Technologies de l'Information et des Systèmes

D2095

# Design, modeling and control of a convertible mini airplane having four tilting rotors

---

T H È S E

pour obtenir le grade de docteur délivré par

**l'Université de Technologie de Compiègne**

Spécialité «Technologies de l'information et des systèmes »

présentée et soutenue publiquement par

**Gerardo Ramón FLORES COLUNGA**

le 31 octobre 2014

Directeur de thèse: **Rogelio LOZANO**

Co-encadrement de thèse: **Juan ESCAREÑO**

## **Jury**

<b>M. Pascual CAMPOY</b> , Professeur, U.P.M, Spain	Rapporteur
<b>M. Stéphane VIOLLET</b> , Directeur de Recherche, ISM	Rapporteur
<b>M. Rogelio LOZANO</b> , Directeur de Recherche, UTC	Examineur
<b>M. Juan ESCAREÑO</b> , <i>Associate Professor</i> , IPSA	Examineur
<b>M. Claude PEGARD</b> , Professeur, UPJV	Examineur
<b>M. Philippe BONNIFAIT</b> , Professeur, UTC	Examineur

## Assessment committee:

**Prof. Rogelio Lozano**

CNRS Research Director, HEUDIASYC Laboratory, Université de Technologie de Compiègne, Compiègne, France

**Prof. Juan Escareño**

Associate Professor, Institut Polytechnique des Sciences Avancées (IPSA), Ivry sur Seine, France

**Prof. Pascual Campoy**

Chair Professor, Computer Visión Group, Universidad Politécnica de Madrid U.P.M, Spain

**Prof. Stéphane Viollet**

CNRS Research Director, Responsable équipe Biorobotique, Institut des Sciences du Mouvement, Marseille, France

**Prof. Claude Pegard**

Professor, MIS Laboratory, Université de Picardie Jules Verne, Amiens, France

**Prof. Philippe Bonnifait**

Professor, HEUDIASYC Laboratory, Université de Technologie de Compiègne, Compiègne, France

Copyright © 2014 by Gerardo Flores (gflorescolunga@gmail.com)

ISBN xxxxx



---

## **Abstract**

This thesis studies some of the most relevant problems in the sense of guidance, navigation and control presented in a particular class of mini aerial vehicles (MAV): the convertible MAV with fixed wings and tilting rotors. This aircraft is able to change its flight configuration from hover to level flight and vice-versa by means of a transition maneuver.

Motivated by civilian applications, we theoretically and experimentally study Lyapunov-based control laws for dynamics presented in the convertible MAV. Results of asymptotic convergence are obtained over the complete flight envelope of the vehicle: from low-speed vertical flight through high-speed forward flight. We have divided this thesis in four main parts: the study of 1) the fixed-wing aircraft; 2) the quadrotor; 3) the convertible aircraft and 4) vision applications by using the convertible aircraft. In a first part, a Lyapunov-based control law is developed to steer a fixedwing mini aerial vehicle along a desired path. Furthermore a path generator is proposed. The resulting control strategy yields global convergence of the current path of the MAV to the desired path. In a second part, a Lyapunov-based control using singular perturbation theory is proposed and applied on dynamics of the MAV. Furthermore, in this part we address the problem of fault detection and diagnosis (FDD) for a quad-rotor. In the third part a new control strategy for the transition between airplane and helicopter mode, and viceversa, in convertible planes is presented. The analysis is carried out for the longitudinal model of the PVHAT (Planar Vertical Helicopter-Airplane Transition) aircraft, which is an airplane having tilting rotors in order to achieve the transition maneuver. The resulting closed loop control algorithm is proved to be globally asymptotically stable. Finally in the fourth part of this thesis the problem of estimation and tracking of a road using a vision embedded system in the PVHAT aircraft is solved. The global exponential stability of the position subsystem together with the switching controller is demonstrated.

Illustrative simulations and experimental results obtained on several experimental platforms developed in this thesis, assess the implementability of the proposed control laws and highlight the merits of the approach.

## **Keywords**

Fixed-wing aircraft; quadrotor; modeling; stability; vision systems; convertible aircraft; tilt-rotor; Lyapunov-based control; transition maneuver; nonlinear systems; Planar Vertical Helicopter-Airplane Transition (PVHAT).

---



# ACKNOWLEDGEMENTS

---

It is a pleasure for me to express my gratitude to the many people who have helped me or have been an important part of my life the last four years.

First of all, I am forever thankful to my parents Ramón Flores and Eva de Flores and my sister Evita Flores for all of your love, support and specially for all your prayers. *Dios los bendiga siempre.*

I am grateful to my advisor in Compiègne France, Rogelio Lozano, for supporting me all the way from my Master thesis to the PhD; I have learned a lot of things from you. Although circumstances hindered an explicit supervision during most of the PhD I am incredibly thankful for the time and responsibility you trusted me with all this time. I have always been free to pursue the projects I had in mind yet still been guided in the right direction, and I am grateful for that.

I am equally thankful to my supervisor and colleague at Paris, Juan Escareño, for all the advice you gave me and for the encouragement to follow my own ideas.

I also thank Pascual Campoy and Stéphane Viollet, who have honored me by agreeing to be the reporters of this thesis. Also thank you to Claude Pegard and Philippe Bonnifait for participating in my thesis jury.

I am particularly grateful to all of my co-authors: José E. Gomez (GIPSA-lab), José A. Guerrero (INSA), L. R. Garcia (Texas A&M University), Hipólito Aguilar (CINVESTAV), Diego Mercado (UTC), Hugo Romero (CINVESTAV) and Israel Lugo (UTC). Thank you for our fruitful collaboration.

A warm thanks goes to all my students/colleagues from whom I have learned a lot: Shuting Zhou (UTC), Adolfo González (CIC-IPN), Jesus Gonzalez (Universidad de Sevilla), Héctor Escamilla (UTC-CINVESTAV), Hilario Salazar (UTC-BUAP), Mariano Larios (UTC-BUAP) and Eric Bazan (UTC-CINVESTAV).

I am thankful to the previous and current group members at UTC, Alexandru Brezoescu and Corentin Chauffaut, for the good coffee breaks and discussions we have had. Specially I am thankful to Jose Luis Rullan who helped me out when I arrived in France.

Finally, I am thankful to the people of Mexico represented by the National Council of Science and Technology (CONACYT) for giving me the financial support to accomplish this thesis.

*Gerardo Flores*

*Compiègne, France, September 2014*

# LIST OF PUBLICATIONS

---

## Journal Papers (JCR)

- J1 Gerardo Flores, J. Escareño, R. Lozano and S. Salazar  
*Quad-Tilting Rotor Convertible MAV: Modeling and Real-Time Hover Flight Control*  
Journal of Intelligent and Robotic Systems, vol. 65, no. 1-4, p. 457-471, (2012).
- J2 J. Balderas, Gerardo Flores, L. Garcia and R. Lozano  
*Tracking a Ground Moving Target with a Quadrotor Using Switching Control*  
Journal of Intelligent and Robotic Systems, vol. 70, no. 1-4, p. 65-78, (2012).
- J3 L. Garcia, Gerardo Flores, G. Sanahuja and R. Lozano  
*Quad Rotorcraft Switching Control : An Application for the Task of Path Following*  
IEEE Transactions on Control Systems Technology, vol. 22, no. 4, p. 1255-1267, (2014).
- J4 Gerardo Flores, S. Zhuo, R. Lozano and P. Castillo  
*A Vision and GPS-Based Real-Time Trajectory Planning for a MAV in Unknown and Low-Sunlight Environments*  
Journal of Intelligent and Robotic Systems, Volume 74, Issue 1-2, pp 59-67, (2014).
- J5 H. Aguilar, Gerardo Flores, S. Salazar and R. Lozano  
*Fault Estimation for a Quad-Rotor MAV Using a Polynomial Observer*  
Journal of Intelligent and Robotic Systems, Volume 73, Issue 1-4, pp 455-468. DOI 10.1007/s10846-013-9924-5, (2014).
- J6 Gerardo Flores and R. Lozano  
*Nonlinear control design of transition maneuvers for convertible aerial vehicles: application to PVHAT aircraft* Automatica, to be submitted.
- J7 Gerardo Flores, H. Salazar and R. Lozano *Switching Control for a Highway Estimation and Tracking* Control Engineering Practice, to be submitted.
- J8 Gerardo Flores, I. Lugo and R. Lozano *Nonlinear Path-Following Strategy for Unmanned Aerial Vehicles* Control Engineering Practice, to be submitted.
- J9 J. Escareño, Gerardo Flores and R. Lozano *Task-based Control of a Rotorcraft MAV Having an Onboard Manipulator* Robotica, to be submitted.

- J10 Gerardo Flores and R. Lozano *Control in the 6-DOF of the Quad Tilt-Rotor MAV* IEEE Transactions on Control Systems Technology, to be submitted.
- J11 Gerardo Flores and R. Lozano *Lyapunov-based Controller using Singular Perturbation Theory for a Convertible MAV* Automatica, to be submitted.
- 

## Book chapters

- Ch1 Gerardo Flores, J. Guerrero, J. Escareño and R. Lozano  
*Modeling and Control of mini UAV*  
in Flight formation control, pp 99-134, edited by J. Guerrero and R. Lozano, John Wiley and Sons, 2012.
- Ch2 Gerardo Flores, J. Escareño, R. Lozano and S. Salazar  
*Quad-Tilting Rotor Convertible MAV : Modeling and Real-time Hover Flight Control*  
in Recent Developments in Unmanned Aircraft Systems, pp 457-472, edited by Kimon P. Valavanis, John Wiley and Sons, 2012, VI, 638p., ISBN 978-94-007-3032-8.
- Ch3 Gerardo Flores, H. Aguilar-Sierra, R. Lozano and S. Salazar  
*Fault Estimation and Control for a Quad-Rotor MAV Using a Polynomial Observer. Part I : Fault Detection*  
in Advances in Intelligent Systems and Computing Volume 252, 2014, pp 151-170, edited by Manuel A. Armada, Alberto Sanfeliu and Manuel Ferre, Springer International Publishing, 2014.
- 

## Conference papers

- C1 Gerardo Flores, J. Escareño, S. Salazar and R. Lozano  
*Quad-Tilting Rotor Convertible MAV : Modeling and Real-time Hover Flight Control*  
International Conference on Unmanned Aircraft Systems (ICUAS'11), May 24-27, 2011, Denver Colorado, USA.
- C2 J. Balderas, Gerardo Flores, L. Garcia and R. Lozano  
*Tracking a Ground Moving Target with a Quadrotor Using Switching Control*  
International Conference on Unmanned Aircraft Systems (ICUAS'12), Jun 12-15, 2012, Philadelphia, PA., USA.

- 
- C3 L. García, Gerardo Flores, G. Sanahuja and R. Lozano  
*Quad-Rotor Switching Control : An Application for the Task of Path Following*  
IEEE American Control Conference (ACC'12), Montréal, Canada, Jun. 2012,  
pp. 4637-4642.
- C4 Gerardo Flores, L. García, G. Sanahuja and R. Lozano  
*PID Switching Control for a Highway Estimation and Tracking Applied on a Convertible Mini-UAV*  
in Proc. IEEE Conference on Decision and Control (CDC'12), Maui, HI, USA, Dec. 2012, pp. 3110-3115.
- C5 Gerardo Flores and Rogelio Lozano  
*Lyapunov-based Controller using Singular Perturbation Theory : An Application on a mini-UAV*  
IEEE American Control Conference (ACC'13), Washington, DC, Jun. 2013,  
pp 1596 - 1601.
- C6 Gerardo Flores, S. Zhuo, R. Lozano and P. Castillo  
*A Vision and GPS-Based Real-Time Trajectory Planning for MAV in Unknown Urban Environments*  
The IEEE 2013 International Conference on Unmanned Aircraft Systems (ICUAS'13), Atlanta, Georgia, USA, May 28-31, 2013, pp 1150 - 1155,  
ICUAS'13 Invited Session : Vision and sensing for localization of UAVs and their targets.
- C7 Gerardo Flores, G. Sanahuja and R. Lozano  
*Lyapunov-Based Switching Control for a Road Estimation and Tracking Applied on a Convertible MAV*  
AIAA Guidance, Navigation, and Control and Co-located Conferences and AIAA Infotech@Aerospace 2013 AIAA GNC, Boston, Massachusetts, USA, 19 - 22 August 2013, DOI : 10.2514/6.2013-4940.
- C8 J. Escareño, M. Rakotondrabe, Gerardo Flores and R. Lozano  
*Rotorcraft MAV Having an Onboard Manipulator : Longitudinal Modeling and Robust Control*  
2013 European Control Conference (ECC'13), Zürich, Switzerland, July 17-19, 2013, pp. 3258-3263.
- C9 D. Mercado, Gerardo Flores, J. Escareño, R. Lozano and P. Castillo  
*GPS/INS/optic flow data fusion for position and Velocity estimation*  
The IEEE 2013 International Conference on Unmanned Aircraft Systems (ICUAS'13), Atlanta, Georgia, USA, May 28-31, 2013, pp. 486-491.

- C10 H. Aguilar, Gerardo Flores, S. Salazar and R. Lozano  
*Fault Estimation For a Quad-rotor MAV Using a Polynomial Observer*  
The IEEE 2013 International Conference on Unmanned Aircraft Systems (ICUAS'13), Atlanta, Georgia, USA, May 28-31, 2013, pp. 717-724.
- C11 Gerardo Flores and Rogelio Lozano  
*Transition Flight Control of the Quad-Tilting Rotor Convertible MAV*  
The IEEE 2013 International Conference on Unmanned Aircraft Systems (ICUAS'13), Atlanta, Georgia, USA, May 28-31, 2013, pp. 789-794.
- C12 Gerardo Flores, I. Lugo-Cardenas and R. Lozano  
*A Nonlinear Path-Following Strategy for a Fixed-Wing MAV*  
The IEEE 2013 International Conference on Unmanned Aircraft Systems (ICUAS'13), Atlanta, Georgia, USA, May 28-31, 2013, pp. 1014-1021.
- C13 S. Zhuo, Gerardo Flores, R. Lozano and P. Castillo  
*Vision-Based Window Estimation for MAV in Unknown Urban Environments*  
2nd IFAC Workshop on Research, Education and Development of Unmanned Aerial Systems, RED-UAS'13, Compiègne France, November 20-22, 2013, pp. 107-111.
- C14 Gerardo Flores, H. Aguilar-Sierra, R. Lozano, and S. Salazar  
*Fault Estimation and Control for a Quad-rotor MAV using a Polynomial Observer. Part I : Fault Detection*  
ROBOT 2013 First Iberian Robotics Conference, Madrid, Spain, November 28-29, 2013.
- C15 Gerardo Flores and Rogelio Lozano  
*Design and Lyapunov Stability Analysis of a Controller for Autonomous Road Following*  
ROBOT 2013 First Iberian Robotics Conference, Madrid, Spain, November 28-29, 2013.
- C16 Gerardo Flores and Rogelio Lozano  
*Nonlinear Control Strategy for the Transition Maneuver of the Quad Tilt-rotor Convertible MAV*  
The 19th World Congress of the International Federation of Automatic Control (IFAC'14), Cape Town, South Africa, August 24-29, 2014, pp. 11055-11059.
- C17 Israel Lugo Cárdenas, Gerardo Flores and Rogelio Lozano  
*The MNAV3DSim : A Simulation Platform for Research, Education and Development of UAV Controllers*  
The 19th World Congress of the International Federation of Automatic

- Control (IFAC'14), Cape Town, South Africa, August 24-29, 2014, pp. 713-717.
- C18 Juan Escareno, Gerardo Flores, Micky Rakotondrabe and R. Lozano  
*Task-based Control of a Multirotor Miniature Aerial Vehicle Having an Onboard Manipulator*  
The IEEE 2014 International Conference on Unmanned Aircraft Systems (ICUAS'14), Orlando, FL, USA, May 27-30, pp. 857-863.
- C19 Israel Lugo Cárdenas, Gerardo Flores, Sergio Salazar and R. Lozano  
*Dubins Path Generation for a Fixed Wing UAV*  
The IEEE 2014 International Conference on Unmanned Aircraft Systems (ICUAS'14), Orlando, FL, USA, May 27-30, pp. 339 - 346.
- C20 Gerardo Flores, Israel Lugo Cárdenas and R. Lozano  
*6-DOF Hovering Controller Design of the Quad Tiltrotor Aircraft: Simulations and Experiments*  
The 53rd IEEE Conference on Decision and Control (CDC'14), Dec 15-17, 2014, Los Angeles, CA, USA, to appear.
- C21 S. Zhuo, Gerardo Flores, R. Lozano and P. Castillo  
*Navigation d'un quadrirotor pour accéder à l'intérieur d'un bâtiment, Journée Caméra 3D (type Kinect) : De la modélisation à l'application*  
Université d'Evry-Val-d'Essonne, Paris, France, 14 juin 2013.

Platforms developed during PhD:

- *ElGuapo V1* Quadrotor experimental platform shown in Fig. 1. Further, Fig. 1 shows the *ElGuapo V2* experimental platform which is under development.



(a) ElGuapo V1 flying.



(b) ElGuapo V2 on the right.

**Figure 1:** Quad-rotor experimental platforms ElGuapo V1 and ElGuapo V2.

- *ElCerdo* Fixed-wing experimental platform shown in Fig. 2.



**Figure 2:** ElCerdo experimental platform.

- *Quad-plane* Convertible MAV experimental platform depicted in Fig. 3.



**Figure 3:** Quad-plane experimental platform.

- *PVHAT* Tilt-rotor convertible aircraft shown in Fig. 4.



**Figure 4:** The PVHAT aircraft experimental platform.

For more details about platforms please refer to the website:

<https://sites.google.com/site/gerardoflorescolunga/research>



# CONTENTS

---

<b>Acknowledgements</b>	<b>iii</b>
<b>1 Introduction</b>	<b>1</b>
<b>I The Fixed-wing aircraft</b>	<b>7</b>
<b>2 A Nonlinear Path-Following Strategy for a Fixed-Wing MAV</b>	<b>9</b>
2.1 Introduction . . . . .	10
2.2 Modeling . . . . .	11
2.3 Problem Statement . . . . .	12
2.3.1 Error dynamics for the path-following controller . . . . .	12
2.4 Control Strategy . . . . .	14
2.4.1 Kinematic Controller Design . . . . .	14
2.4.2 Roll angle control . . . . .	16
2.5 Simulation results and Experimental platform . . . . .	17
2.5.1 Experimental Platform . . . . .	20
2.6 Experiments . . . . .	21
<b>3 Dubins Path Generation for a Fixed Wing MAV</b>	<b>29</b>
3.1 Introduction . . . . .	30
3.1.1 Coordinates Transformation . . . . .	30
3.2 Path Generation . . . . .	33
3.2.1 Dubins path <i>RSR</i> . . . . .	34
3.2.2 Dubins path <i>RSL</i> . . . . .	36
3.2.3 Dubins path <i>LSL</i> . . . . .	38
3.2.4 Dubins path <i>LSR</i> . . . . .	39
3.3 Simulation Example . . . . .	41
3.3.1 MAV3DSim Simulation Platform . . . . .	41
3.3.2 Simulation Scenario . . . . .	43
<b>II The Quadrotor</b>	<b>45</b>
<b>4 Lyapunov-based Controller for a mini-UAV using Singular Perturbation Theory</b>	<b>47</b>
4.1 Introduction . . . . .	48
4.2 Problem Description . . . . .	49
4.3 System Description . . . . .	50
4.3.1 Modeling for the Singular Perturbation Problem . . . . .	51

4.4	Controller Design . . . . .	51
4.4.1	Real Input Controls . . . . .	57
4.5	Simulations and Experimental Results . . . . .	58
4.5.1	Numerical Results . . . . .	58
4.5.2	Experimental Results . . . . .	59
<b>5</b>	<b>Fault Estimation and Control for a Quad-rotor MAV using a Polynomial Observer</b>	<b>63</b>
5.1	Introduction . . . . .	65
5.2	Fault Diagnosis Problem . . . . .	66
5.2.1	Observability and Diagnosability Condition . . . . .	67
5.3	Polynomial Observer . . . . .	68
5.3.1	Observer design . . . . .	68
5.3.2	Convergence Analysis . . . . .	69
5.4	Application to Quad-rotor MAV . . . . .	72
5.4.1	Modeling . . . . .	72
5.4.2	Control Strategy . . . . .	73
5.4.3	Diagnosability analysis . . . . .	74
5.4.4	Polynomial Observer . . . . .	75
5.5	Evaluation Function . . . . .	76
5.5.1	Fault detection . . . . .	76
5.5.2	Experimental results by using the evaluation function . . . . .	76
5.6	Fault reconstruction results . . . . .	78
<b>III</b>	<b>The Convertible Aircraft</b>	<b>85</b>
<b>6</b>	<b>Control in the 6-DOF of the PVHAT aircraft.</b>	
	<b>Hover Control</b>	<b>87</b>
6.1	Introduction . . . . .	88
6.2	System Description . . . . .	89
6.2.1	Modeling of the PVHAT Aircraft . . . . .	89
6.2.2	Dynamic Extension . . . . .	91
6.3	Lyapunov-Based Tracking Control Strategy . . . . .	92
6.4	Simulations and Experimental Results . . . . .	97
6.4.1	Simulations Results . . . . .	97
6.4.2	Experimental Results . . . . .	97
6.4.3	Experimental Setup . . . . .	99
<b>7</b>	<b>Nonlinear control design of transition maneuvers for convertible aerial vehicles</b>	<b>103</b>
7.1	Introduction . . . . .	104

7.2	System Description . . . . .	105
7.2.1	Flight envelope description . . . . .	105
7.3	Control Algorithm Design . . . . .	107
7.4	Numerical Results . . . . .	110
7.4.1	Hover to airplane mode simulation . . . . .	110
7.4.2	Airplane to hover mode simulation . . . . .	111

**IV Vision applications 113**

**8 Lyapunov-Based Switching Control for a Road Estimation and Tracking 115**

8.1	Introduction . . . . .	117
8.2	Problem Statement . . . . .	118
8.3	States estimation using a vision sensor . . . . .	119
8.3.1	Computing the heading angle . . . . .	121
8.3.2	Computing the relative lateral position . . . . .	122
8.3.3	Translational velocities . . . . .	122
8.3.4	Computing the heading and the lateral position . . . . .	123
8.4	Control Strategy . . . . .	125
8.4.1	Operating modes for the road following mission . . . . .	126
8.4.2	Control Laws for each Operating Mode . . . . .	126
8.4.3	Stability Analysis of the Lateral Position Control . . . . .	130
8.5	Experimental platform . . . . .	132
8.5.1	The convertible MAV experimental vehicle . . . . .	132
8.5.2	Embedded vision system . . . . .	133
8.5.3	Ground station . . . . .	134
8.6	Simulations and Real Time Experiments . . . . .	134
8.6.1	Numerical Simulations . . . . .	134
8.6.2	Real Time Experiments . . . . .	136

**9 Conclusions and Outlook 145**

9.1	Part I: The Fixed-wing aircraft . . . . .	146
9.2	Part II: The Quadrotor . . . . .	146
9.3	Part III: The Convertible Aircraft . . . . .	147
9.4	Part IV: Vision applications . . . . .	148

**Bibliography 149**



# PREFACE

---

This thesis was submitted to the *École doctorale* of the University of Technology of Compiègne, as a partial fulfillment of the requirements to obtain the PhD degree. The work presented was carried out in the years 2010-2014 in the UMR CNRS 7253: Heuristics and Diagnostics for Complex Systems laboratory at the Department of Computer Engineering of the University of Technology of Compiègne.

The thesis is a direct continuation of my Master's thesis titled "*Modeling, control and implementation of convertible UAVs*" which was carried out at CINVESTAV, Mexico and defended in 2010. The paper resulting from my Master's work, published in JINT in 2011, is highly relevant in this thesis.

## THESIS OBJECTIVES

The objectives of the work reported in this thesis are primarily 1) to investigate the aerodynamics involved on the main classes of unmanned aerial vehicles: the fixed-wing, the quadrotor and the convertible aircraft; 2) design new control algorithms to stabilize the convertible aircraft over the complete flight envelope, i.e. in hover, fixed-wing and in transition mode; 3) build and develop several classes of experimental platforms in order to demonstrate the efficiency of the proposed algorithms; 4) propose solutions using drones to solve real-world problems.

## THESIS OUTLINE

This thesis is in the form of a synopsis. Part I of the thesis provides the study of the fixed-wing aircraft. It presents modeling, control and introduces an experimental platform developed at laboratory. Part II presents the quadrotor. Two main subjects are covered in this part of the thesis: fault estimation and control for the quadrotor and the study of the time-scale separation between the attitude and translational dynamics of the quadrotor. Part III constitutes the research of the convertible aircraft, it includes a control strategy for the transition from hover to fixed-wing and viceversa. Finally in Part IV the estimation and tracking of a road using the new tilt-rotor convertible MAV is presented. In the four parts of the thesis, simulations and experimental tests demonstrate the validity of results.

*Dedicado a Evita.*



# 1

## INTRODUCTION

---

In the last few years Micro Aerial Vehicles (MAVs) have been successfully employed to address a large variety of applications in the area of surveillance [1], environmental awareness [2], search and rescue operations [3], aerial refueling [4], aerial robotics and many others [5]. The key feature of MAVs is to provide a mobile extension of human perceptions allowing not only the security of the user but also gathering information such as images or video, locations coordinates, weather conditions, etc., for either online or offline analysis. However, there are missions whose scope is beyond the capabilities of conventional MAVs designs since they require not only longer flight endurance but also hovering/VTOL capabilities. Missions like the surveillance of both fast-moving and static targets, identification of cracks in pipelines or bridges, medical supplies (blood samples, saliva samples, medications), exchange between hospitals and clinics located in remote areas, are missions that can be carried out in a more efficient manner with hybrid aerial vehicles than standard airplanes or helicopters. Besides these commonly used aerial vehicles the *Convertible Aircraft*, which is an aerial vehicle capable of combining the advantages of horizontal and vertical flight, has been recently gaining popularity [6]. The distinguishing feature of such a class of systems is the ability of operating both in the stable hover flight and in the fast and efficient level flight. During level flight, the aircraft configuration appears similar to the one of a fixed-wing aerial vehicle, in which the force of gravity is compensated through the lift obtained by means of suitable aerodynamic surfaces installed on the vehicle, such as wings or canards. On the other side, at hover, the aircraft configuration is more similar to the one of a helicopter, in which the gravity force is compensated only by the force produced by the propeller. In this case the additional maneuverability typical of helicopters is paid back with a larger amount of energy consumed to sustain the flight. In this setting, the transition maneuver is a particular trajectory of the system in which the flight configuration of the vehicle is changed from hover to level flight or vice versa [7].

While the convertible aircraft concept is very promising, it also comes with significant challenges. Indeed it is necessary to design controllers working over

the complete flight envelope of the vehicle: from low-speed vertical flight through high-speed forward flight. The main change at this respect, is investigate the large variation in the vehicle dynamics between these two different flight regimes.

This thesis reports current work on the modeling, control and development of an experimental prototype of a new convertible aircraft: the *PVHAT* (Planar Vertical Helicopter-Airplane Transition), which is capable of flying in horizontal and vertical modes. This mini aerial vehicle is one of the first of its kind among tilt-rotor vehicles on that scale range. The vehicle is driven by four rotors and has a conventional airplane-like structure, which constitutes a highly nonlinear plant and thus the control design should take into account this aspect.

In order to investigate the tilt-rotor convertible aircraft and because of the presence of fixed-wing and quadrotor dynamics, we have planned to investigate several features of the fixed-wing aircraft and the quadrotor each individually. With these two kinds of aerial vehicles studied, we proceed to investigate the dynamics and behaviour of the *PVHAT* aircraft. Furthermore, some vision techniques have been applied to estimate the states of the MAV. This thesis is then divided in four parts, pictured in Figure 1.1 and described as follows.

**Part I: The Fixed-wing aircraft** This part focuses on the fixed-wing aircraft. Simulations and experimental results are presented by using the Dubins model. In chapter 2, a Lyapunov-based control law is developed to steer a fixed-wing MAV along a desired path. The proposed controller overcomes stringent initial condition constraints that are present in several path-following strategies in the literature. The key idea behind the proposed strategy, is to minimize the error of the path-following trajectory by using a virtual particle, which should be tracked along the path. For this purpose, the particle speed is controlled, providing an extra degree of freedom. Controller design is stated by using Lyapunov techniques. The resulting control strategy yields asymptotic convergence of the current path of the MAV to the desired path. Simulations are presented using the simulator *MAV3DSim*, in order to demonstrate the effectiveness of the control law. Furthermore an experimental platform called *ElCerdo* is introduced.

In chapter 3 a path generator is proposed for the fixed-wing MAV. The Dubins paths serve as a strategy to find the shortest route for the non-holonomic model of the aerial vehicle. The Dubins path generation is combined with a nonlinear Lyapunov-based path-following control. We present a complete simulation environment in which the path generator and path following strategy are validated. As an example of application we propose the scenario in which a missing person is located in some known area and we use the path generator

## AIRCRAFT ADDRESSED IN THIS THESIS

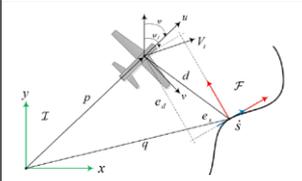
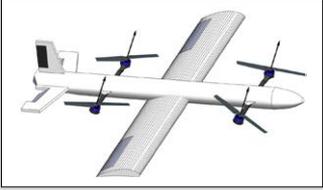
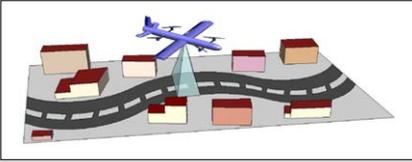
THE FIXED-WING AIRCRAFT (Part I)	THE QUADROTOR (Part II)
<p>A Nonlinear Path-Following Strategy (Ch. 2) and Dubins Path Generation (Ch. 3)</p> <ul style="list-style-type: none"> <li>• Modeling of Dubins airplane</li> <li>• Dubins path</li> <li>• Simulations and experimental results</li> </ul> 	<p>Lyapunov-based Controller for a mini-UAV using Singular Perturbation Theory (Ch. 4)</p> <ul style="list-style-type: none"> <li>• Study of the hierarchical control scheme (inner-outer loop)</li> <li>• Case of study: The convertible aircraft in hover mode</li> <li>• Simulations and experimental results</li> </ul> <p>Fault Estimation and Control for a Quad-rotor MAV using a Polynomial Observer (Ch. 5)</p> <ul style="list-style-type: none"> <li>• Fault detection and diagnosis problem, case of study: The quadrotor</li> <li>• Simulations and preliminary experimental results</li> </ul>
THE CONVERTIBLE AIRCRAFT (Part III)	VISION APPLICATIONS (Part IV)
<p>Hover control in the 6-DOF of the PVHAT aircraft. (Ch. 6) and nonlinear control design of transition maneuvers (Ch. 7)</p> <ul style="list-style-type: none"> <li>• 6DOF modeling</li> <li>• 2DOF modeling</li> <li>• Simulations and experimental results</li> </ul> 	<p>Lyapunov-Based Switching Control for a Road Estimation and Tracking Applied on a Convertible MAV (Ch. 8)</p> <ul style="list-style-type: none"> <li>• Road-following mission</li> <li>• States estimation using vision sensors</li> <li>• Simulations and experimental results</li> </ul> 

Figure 1.1: The problem addressed in this thesis.

along with this path-following strategy applied to the fixed wing MAV to search and find this person.

**Part II: The Quadrotor** In chapter 4, a Lyapunov-based control using singular perturbation theory is proposed and applied on dynamics of a miniature unmanned aerial vehicle. Such controller is designed taking into account the presence of the small parameter  $\varepsilon$  on vehicle dynamics, causing a time-scale separation between the attitude and translational dynamics of the MAV. The stability analysis is demonstrated by presenting a scenario in which the time-scale property arises on the the MAV dynamics. In addition, the values of the parameter  $\varepsilon$  for which the control law is validated, are given. Simulations are derived and presented to demonstrate the effectiveness of the control law. The proposed controller has been applied to the *Quad-plane* MAV experimental

platform, in order to validate the performance and to show the time-scale property.

Chapter 5 addresses the problem of fault detection and diagnosis (FDD) for a quad-rotor. Actuator faults are considered on this chapter. The basic idea behind the proposed method is to estimate the faults signals using the extended state observers theory. To estimate the faults, a polynomial observer is presented by using the available measurements and the known inputs of the system. To investigate the observability and diagnosability properties of the system, a differential algebra approach is proposed. Furthermore, an evaluation function depending on the system states is developed, in order to be used in a controller responsible to compensate the failures. The effectiveness of the methodology is illustrated by means of numerical simulations and some preliminary experimental tests.

**Part III: The Convertible Aircraft** Chapter 6 presents a particular class of a convertible mini aerial vehicle with fixed wings, the so-called PVHAT aircraft. This aircraft is able to change its flight configuration from hover to level flight and vice-versa by means of a transition maneuver. In this first part of the research, the hover dynamics of the PVHAT aircraft is investigated. Dynamical model and nonlinear control based on Lyapunov design are studied. The presented approach focuses on the problem of finding a control law capable of stabilizing the aircraft's position. Some simulations results are given, which demonstrate the effectiveness of the controller. Further, some experimental results are presented and tested on the PVHAT aircraft experimental platform.

Chapter 7 presents a new control strategy for the transition between airplane and helicopter mode, and vice-versa, in convertible planes. The analysis is carried out for the longitudinal model of the PVHAT aircraft, which is an airplane having tilting rotors in order to achieve the transition. The control strategy is smooth in the sense that it does not involve commutation between two different controllers. The resulting closed-loop control algorithm is proved to be globally asymptotically stable. The altitude and the longitudinal speed are proved to converge to desired values. The proposed controller has been successfully tested in numerical simulations.

**Part IV: Vision applications** This part addresses the problem of estimation and tracking of a road using the Quad-plane experimental platform. For that objective, we consider the following scenario: (i) no previous knowledge of the road, i.e. shape, dimension and color, (ii) loss of information by the sensors is considered and (iii) nonlinear dynamics of the MAV is taken into

consideration. Aiming at this goal, two operational regions are defined: road detected and road not detected by the sensors. A switching between the measurements of imaging and inertial sensors enables estimation of the required states in both operational regions. For dealing with both aforementioned cases, a Lyapunov-based switching control for stabilizing the vehicle's position is proposed. Unmodeled dynamics such as friction forces are estimated by means of the proposed controller. The global exponential stability of the position subsystem together with the switching controller is demonstrated exploiting the fact that the individual closed-loop systems are globally exponentially stable and the switching is sufficiently slow, so as to allow the transient effects to dissipate after each switch. The control law is validated on the Quad-plane experimental platform, showing the expected behavior during autonomous navigation.



# Part I

THE FIXED-WING AIR-  
CRAFT



# 2

## A NONLINEAR PATH-FOLLOWING STRATEGY FOR A FIXED-WING MAV

---

*In this chapter, a Lyapunov-based control law is developed to steer a fixed-wing mini aerial vehicle (MAV) along a desired path. The proposed controller overcomes stringent initial condition constraints that are present in several path-following strategies in the literature. The key idea behind the proposed strategy, is to minimize the error of the path-following trajectory by using a virtual particle, which should be tracked along the path. For this purpose, the particle speed is controlled, providing an extra degree of freedom. Controller design is stated by using Lyapunov techniques. The resulting control strategy yields global convergence of the current path of the MAV to the desired path. Simulations are presented using the simulator MAV3DSim, in order to demonstrate the effectiveness of the control law. Furthermore an experimental platform called ElCerdo is introduced.*

### Contents

---

<b>2.1</b>	<b>Introduction</b>	<b>10</b>
<b>2.2</b>	<b>Modeling</b>	<b>11</b>
<b>2.3</b>	<b>Problem Statement</b>	<b>12</b>
2.3.1	Error dynamics for the path-following controller	12
<b>2.4</b>	<b>Control Strategy</b>	<b>14</b>
2.4.1	Kinematic Controller Design	14
2.4.2	Roll angle control	16
<b>2.5</b>	<b>Simulation results and Experimental platform</b>	<b>17</b>
2.5.1	Experimental Platform	20
<b>2.6</b>	<b>Experiments</b>	<b>21</b>

---

### 2.1 Introduction

---

In the last few years there has been a considerable growth in the development of micro and mini air vehicles (MAV). Increasing capabilities and falling costs have increase the popularity of such vehicles, not only in military missions but also in civilian tasks. As a result, interest in development of flight control systems has raised considerably.

Originally, such controllers have been designed based on a linear version of the aircraft dynamics. As a consequence the linear model is no longer valid in some flight conditions, yielding a poor performance of the controller. Gain scheduling techniques [8] have been applied in order to overcome this difficulty, implying all the disadvantages inherent in this kind of techniques, like the necessity to compute different controllers for different operating points and estimating aircraft stability derivatives for the whole flight envelope.

Several nonlinear path planning controllers have been investigated and implemented mainly on ground robots. Some of these techniques have been taken from the ground vehicles and inherited to the MAV.

Among the methods used in path-planning, we can mention the nonlinear lateral track control law proposed in [9] or the method based on vector fields, which are used to generate desired course inputs, which is presented in [10]. In [11] the problem of constrained nonlinear trajectory tracking control for unmanned air vehicles is investigated. A method for UAV path-following using vector fields to direct the vehicle onto the desired path is presented in [12]. Several methods based on potential field functions have been investigated [13], [14] however the primitive forms of potential field functions present some difficulties when choosing an appropriate potential function, and the algorithm may be stuck at some local minimum [15]. Path planning techniques based on optimization methods like Model Predictive Control approaches and linear programming have been investigated in [16], however, the complex computations demanded by this kind of control and similar approaches make the implementation unfeasible for low-cost MAV.

The chapter is organized as follows. A nonlinear path-following strategy for a fixed-wing MAV is considered, taking into account a simplified version of the lateral dynamics of an airplane. This model is defined in Section 2.2, where we also present the flight conditions considered for the controller. Section 2.3 presents the problem statement of path-following mission for a MAV. The strategy developed in this work, is based on the idea of a *virtual particle* moving along a desired path. A velocity controller for such particle is presented in

Section 6.3. With this approach we can avoid the problems that arise when the virtual particle is defined only as a projection of the vehicle onto the path. The performance of the proposed controller is showed in Section 2.5 where simulation results are presented.

## 2.2 Modeling

The term *Dubins Aircraft* was introduced in [17], where the time-optimal path problem was examined in order to achieve different altitudes. The system model of the Dubins aeroplane is described by the subsequent relations

$$\begin{aligned}\dot{x} &= V_t \cos \psi \\ \dot{y} &= V_t \sin \psi \\ \dot{\psi} &= \omega\end{aligned}\tag{2.1}$$

where  $x$  and  $y$  denotes the inertial position of the aircraft,  $\psi$  is the heading angle,  $\omega$  is the heading rate,  $\phi$  is the roll angle,  $V_t$  is the airspeed, i.e. the speed of an aircraft relative to the air. For this work, we have chosen the ( $X$ - $Y$ - $Z$ ) inertial reference frame, but the analysis can be performed considering a different reference frame. The model (2.1) is a simplified kinematic version of the lateral dynamics of an airplane. The aircraft is considered to be moving with a constant velocity  $V_t$  at a constant altitude  $h_d$ . In the subsequent analysis, we assume no sideslip at a banked-turn maneuver. Also, we consider a boundedness in the roll angle given by

$$|\phi| \leq \phi_{max}\tag{2.2}$$

Assuming a coordinated turn, and given the boundedness of the roll angle  $\phi$  the minimum turn radius  $\rho$  that the aircraft can fly is given by

$$\rho = \frac{V_t^2}{g \tan(\phi_{max})}\tag{2.3}$$

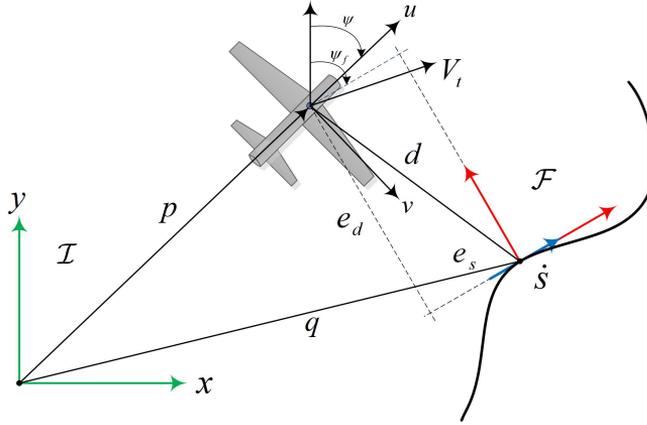
Moreover, the heading rate  $\omega$  is induced by the roll angle as

$$\omega = \frac{g}{V_t} \tan \phi\tag{2.4}$$

where  $g$  is the gravity acceleration.

## 2.3 Problem Statement

In this section, the problem statement is introduced and a dynamic system suitable for control purposes is formulated. Considering Fig. 2.1, the key idea



**Figure 2.1:** Path following control problem schema.

behind the path-following controller relies on reducing two expressions to zero: the first one is the distance between the aircraft's center of mass  $p$  and the point  $q$  on the path, the second one is the angle between the airspeed vector and the tangent to the path at  $q$ . To accomplish these objectives, we introduce a virtual particle moving along the geometric path at a velocity  $\dot{s}$ . Consider a frame attached to such particle, this frame plays the role of a body axis of the virtual particle, and is the so-called Serret-Frenet frame denoted by  $\mathcal{F}$  [18]. It is worth noting that the particle velocity evolves according to a conveniently defined control law  $\dot{s}$ , yielding an extra controller design parameter. With this set-up in mind, the aforementioned angle and distance will become the coordinates of the error space, where the control problem is stated and solved.

### 2.3.1 Error dynamics for the path-following controller

Consider that the 2-D geometric path is represented by smooth functions parameterized by  $t$ , i.e.  $x_s(t)$  and  $y_s(t)$ . Thus,  $(x_s(t), y_s(t))$  represent the virtual particle coordinates. The inertial position of the aircraft is defined by  $p = [x \ y]^T$  in the inertial reference frame  $\mathcal{I}$ . For the purpose of following

the given path, we define the inertial vector error  $d^I = p - q(s)$  expressed in  $\mathcal{F}$ , which will be minimized in order to track the path. Such error vector  $d^I$  has been decomposed into its components  $e_s$  and  $e_d$ , corresponding to the error in the  $x$ -axis of the frame  $\mathcal{F}$  and the error in the  $y$ -axis of the frame  $\mathcal{F}$ , respectively as it is shown in Fig. 2.1. From the Fig. 2.1, we can see that the tangent vector to the path at  $q(s)$  is parallel to  $x$ -axis of the frame  $\mathcal{F}$ . The angle  $\psi_f$  is measured from the inertial frame to the tangent vector of  $q(s)$ .

Considering an arbitrary point  $q$  on the path, and let

$$R = \begin{pmatrix} \cos(\psi_f) & -\sin(\psi_f) \\ \sin(\psi_f) & \cos(\psi_f) \end{pmatrix} \quad (2.5)$$

the rotation matrix from  $\mathcal{F}$  to  $\mathcal{I}$ , parameterized locally by  $\psi_f$ . Thus, the error  $d^I$  expressed in the Serret-Frenet frame is given by

$$d^{SF} = (R^T)(d^I) = R^T(p - q(s)) \quad (2.6)$$

Furthermore, we define the yaw angle error as

$$\tilde{\psi} = \psi - \psi_f \quad (2.7)$$

The angle  $\psi_f$  can be computed by using the information provided by the geometric path and its first derivative with respect to the parameter  $t$ , as follows

$$\psi_f = \arctan \frac{y'_s}{x'_s} \quad (2.8)$$

where  $x'_s = \frac{dx_s}{dt}$ ,  $y'_s = \frac{dy_s}{dt}$ .

To obtain the error state dynamic equations suitable for control purposes, we must compute the time derivative of (2.6) and (2.7). By differentiating (2.6), it follows that

$$\begin{aligned} \dot{d}^{SF} &= R^T(\dot{p} - \dot{q}(s)) + \dot{R}^T(p - q(s)) \\ &= R^T(\dot{p} - \dot{q}(s)) + S^T(\dot{\psi})R^T(p - q(s)) \end{aligned} \quad (2.9)$$

where  $S(\dot{\psi})$  is given by

$$S(\psi) = \begin{pmatrix} 0 & -\dot{\psi}_f \\ \dot{\psi}_f & 0 \end{pmatrix} \quad (2.10)$$

From (2.1), the time derivative of  $p$  and  $q(s)$  can be represented as follows

$$\dot{p} = R(\psi) \begin{pmatrix} V \\ 0 \end{pmatrix} \quad ; \quad \dot{q} = R \begin{pmatrix} \dot{s} \\ 0 \end{pmatrix} \quad (2.11)$$

## 14 2. A Nonlinear Path-Following Strategy for a Fixed-Wing MAV

---

The time derivative of (2.7) results in

$$\dot{\tilde{\psi}} = \omega - \dot{\psi}_f \quad (2.12)$$

with

$$\dot{\psi}_f = \mathcal{C}_C(s)\dot{s} \quad (2.13)$$

where  $\frac{d\psi_f}{dt} = \mathcal{C}_C(s)$  is the path curvature. The path curvature is expressed as a function of the path coordinates  $(x_s(t), y_s(t))$  and its first and second derivatives with respect to the parameter  $t$ , i.e.  $x'_s = \frac{dx_s}{dt}$ ,  $y'_s = \frac{dy_s}{dt}$ . Thus, the path curvature  $\frac{d\psi_f}{dt} = \mathcal{C}_C(s)$  is given by

$$\mathcal{C}_C = \frac{|y''_s x'_s - y'_s x''_s|}{(x'^2_s + y'^2_s)^{3/2}} \quad (2.14)$$

Finally, by substituting (2.11) in (2.9) and using (2.12) we obtain the error kinematic model suitable for the control purposes as

$$\begin{aligned} \dot{e}_s &= V_t \cos \tilde{\psi} - (1 - \mathcal{C}_C(s)e_d)\dot{s} \\ \dot{e}_d &= V_t \sin \tilde{\psi} - \mathcal{C}_C(s)e_s \dot{s} \\ \dot{\tilde{\psi}} &= \omega - \mathcal{C}_C(s)\dot{s} \end{aligned} \quad (2.15)$$

## 2.4 Control Strategy

---

In this section we present a nonlinear path following control strategy. Such control strategy is done in two steps. The first step yields a kinematic controller by adopting the yaw rate  $\omega$  from 2.1 as a virtual control input. The second step addresses the vehicle dynamics in order to obtain the control law for the input variable  $\phi$ . Such control law relies on the kinematic controller previously derived.

### 2.4.1 Kinematic Controller Design

---

Following a similar approach as in [19], we introduce a desired approach angle parameterized by  $k_\delta > 0$  as

$$\delta(e_d) = -\psi_a \frac{e^{2k_\delta e_d} - 1}{e^{2k_\delta e_d} + 1} \quad (2.16)$$

where  $0 < \psi_a < \pi/2$ . The sigmoid function (2.16) is bounded and differentiable with respect to the error  $e_d$ . It provides the desired relative course transition of the fixed-wing MAV to the path as a function of  $e_d$ . Moreover, (2.16) satisfies the condition  $e_d \delta(e_d) \leq 0 \forall e_d$ . Such condition guides the MAV to the correct direction, i.e., turn left when the MAV is on the right side of the path, and turn right in the opposite situation.

With the purpose of investigating the control law for the system (2.1), we propose a Lyapunov function candidate given by

$$V(e_d, e_s, \tilde{\psi}) = \frac{1}{2}e_d^2 + \frac{1}{2}(\tilde{\psi} - \delta(e_d))^2 + \frac{1}{2}e_s^2 \quad (2.17)$$

The time derivative of (2.17) along the trajectory of (2.1) is computed as follows

$$\begin{aligned} \dot{V}(e_d, e_s, \tilde{\psi}) &= (\tilde{\psi} - \delta(e_d))(\omega + \beta) + (e_d)(V_t \sin(\delta(e_d))) \\ &\quad + (e_s)(V_t \cos \tilde{\psi} - \dot{s}) \end{aligned} \quad (2.18)$$

where

$$\beta = -\mathcal{C}_c(s)\dot{s} - \dot{\delta}(e_d)(V_t \sin \tilde{\psi} - \mathcal{C}_c(s)e_s\dot{s}) + (V_t e_d) \left( \frac{\sin \tilde{\psi} - \sin(\delta(e_d))}{\tilde{\psi} - \delta(e_d)} \right)$$

where the derivative with respect to  $e_d$  of (2.16) is

$$\dot{\delta}(e_d) = -\frac{4\psi_a k_\delta e^{2k_\delta e_d}}{(e^{2k_\delta e_d} + 1)^2} \quad (2.19)$$

Substituting the following kinematic control law

$$\begin{aligned} \dot{s} &= V_t \cos \tilde{\psi} + k_s e_s \\ \omega &= -\beta - k_{\omega_1}(\tilde{\psi} - \delta(e_d)) \end{aligned} \quad (2.20)$$

where  $k_s, k_{\omega_1}$  are positive real numbers, in (2.18), yields

$$\begin{aligned} \dot{V}(e_d, e_s, \tilde{\psi}) &= -k_s e_s^2 - k_{\omega_1}(\tilde{\psi} - \delta(e_d))^2 \\ &\quad + (V_t e_d)(\sin(\delta(e_d))) \leq 0 \end{aligned}$$

To conclude convergence of the states  $(e_s, e_d, \tilde{\psi})$  to zero, we state de LaSalle's Invariance Principle.

**Theorem 1.** LaSalle's Theorem *Let  $\mathcal{O}$  be a positively invariant set of system (2.17). Let  $\Omega \subset \mathcal{O}$  a set in which every solution starting in  $\mathcal{O}$  converges to  $\Omega$ . Furthermore, let  $\mathcal{M}$  be the largest invariant set contained in  $\Omega$ . Then, as  $t \rightarrow \infty$ , every bounded solution starting in  $\mathcal{O}$  converges to  $\mathcal{M}$ .*

## 16 2. A Nonlinear Path-Following Strategy for a Fixed-Wing MAV

*Proof.* Convergence of the states  $(e_s, e_d, \tilde{\psi})$  to zero. The proof relies on Theorem 1. Consider the system (16) and the radially unbounded Lyapunov function candidate (2.17). Let us define the compact set  $\mathcal{O}$  as  $\mathcal{O} = \{V(e_d, e_s, \tilde{\psi}) \leq a\}$ , where  $a \in \mathbb{R}^+$ . Define the set  $\Omega$  as

$$\Omega = \{[e_d \ e_s \ \tilde{\psi}]^T \in \mathcal{O} \ : \ \dot{V}(e_d, e_s, \tilde{\psi}) = 0\} \quad (2.21)$$

Equivalently, the expression  $\dot{V}(e_d, e_s, \tilde{\psi}) = 0$  means that  $e_s = e_d = 0$  and  $\tilde{\psi} = \delta$ . Since  $\delta$  is a function of the error  $e_d$ , it is easy to verify that any point starting from  $\Omega$  is an invariant set. Hence, by LaSalle Theorem, every trajectory starting in  $\mathcal{O}$  converges to 0 as  $t \rightarrow \infty$ , i.e.  $\lim_{t \rightarrow \infty} e_s = 0$ ,  $\lim_{t \rightarrow \infty} e_d = 0$  and therefore  $\lim_{t \rightarrow \infty} \tilde{\psi} = \delta(e_d) = 0$ .  $\square$

### 2.4.2 Roll angle control

In this section we compute the roll control from the heading rate command of the kinematic controller previously obtained in Section 2.4.1. For this purpose, we have adopted an inner and outer feedback-loop control approach, where the outer-loop provides the desired value to the inner-loop.

It's the aileron's mission to set up the bank that causes the turn, i.e. the lateral accelerations on an airplane are produced by aileron displacement as we can see in (2.1) and (2.4). To get consistency with the mechanical limitations of the airplane, referring in terms of maximum turn rate, the lateral acceleration  $g \tan \phi$  stated in (2.4) is transformed into heading commands suitable to guarantee the control maneuvers of the aircraft.

We introduce an auxiliary control input  $u$  for the roll angle by augmenting the error model (2.15) with  $\dot{\tilde{\phi}} = u$ . Thus, the augmented system is given as

$$\begin{aligned} \dot{e}_s &= V_t \cos \tilde{\psi} - (1 - \mathcal{C}_C(s)e_d)\dot{s} \\ \dot{e}_d &= V_t \sin \tilde{\psi} - \mathcal{C}_C(s)e_s\dot{s} \\ \dot{\tilde{\psi}} &= \frac{g}{V} \tan \phi - \mathcal{C}_C(s)\dot{s} \\ \dot{\tilde{\phi}} &= p \\ \dot{p} &= u \end{aligned} \quad (2.22)$$

where  $p$  is the roll rate. The controller obtained in the Section 2.4.1 will take the role of the desired heading rate  $\omega_d$  for the control  $u$ .

Using (2.4), we proceed by defining the error state for the roll rate as follows

$$\tilde{\phi} = \phi - \phi_d = \arctan \left( \frac{V_t \omega}{g} \right) - \phi_d \quad (2.23)$$

Taking the time derivative of (2.23) as

$$\dot{\phi} = \dot{\phi} - \dot{\phi}_d = \frac{\gamma\dot{\omega}}{1 + \gamma\omega} - \dot{\phi}_d \quad (2.24)$$

where  $\gamma = \frac{V_t}{g}$  and  $\dot{\phi}_d = \frac{\gamma\dot{\omega}_d}{1 + \gamma\omega_d}$  with  $\omega_d$  given by the second equation of (2.20).

In order to obtain the control  $u$ , we propose the total candidate Lyapunov function given by

$$\begin{aligned} W(\tilde{\phi}, \dot{\tilde{\phi}}, e_d, e_s, \tilde{\psi}) &= \frac{\lambda}{2}(\tilde{\phi})^2 + \frac{\dot{\tilde{\phi}}}{\lambda} + \frac{q}{2\lambda}\dot{\tilde{\phi}}^2 + \frac{1}{2}e_s^2 \\ &+ \frac{1}{2}e_d^2 + \frac{1}{2}(\tilde{\psi} - \delta(e_d))^2 \end{aligned} \quad (2.25)$$

Where  $\lambda > 0$  and  $q > 1$  are free parameters to be chosen. Using the controllers (2.20), the time derivative of (2.26) along the trajectory of (2.22) is given by

$$\begin{aligned} \dot{W}(\tilde{\phi}, \dot{\tilde{\phi}}, e_d, e_s, \tilde{\psi}) &= u \left( \frac{q}{\lambda}\dot{\tilde{\phi}} + \tilde{\phi} \right) + \lambda\tilde{\phi}\dot{\tilde{\phi}} + \dot{\tilde{\phi}}^2 - k_{\omega_1}(\tilde{\psi} - \delta(e_d))^2 \\ &- k_s e_s^2 + (V_t e_d)(\sin(\delta(e_d))) \end{aligned} \quad (2.26)$$

Consider the control input

$$u = -k_p \tilde{\phi} - k_d \dot{\tilde{\phi}} \quad (2.27)$$

where  $k_p$  and  $k_d$  are positive real numbers. Then, substituting (2.27) in (2.26) it leads to

$$\begin{aligned} \dot{W}(\tilde{\phi}, \dot{\tilde{\phi}}, e_d, e_s, \tilde{\psi}) &= -k_p \tilde{\phi}^2 - \left( \frac{k_d q}{\lambda} - 1 \right) \dot{\tilde{\phi}}^2 - k_{\omega_1}(\tilde{\psi} - \delta(e_d))^2 \\ &- k_s e_s^2 + (V_t e_d)(\sin(\delta(e_d))) \end{aligned} \quad (2.28)$$

Using the same procedure of the previous section, it leads to  $\dot{W}(\tilde{\phi}, \dot{\tilde{\phi}}, e_d, e_s, \tilde{\psi}) < 0$  provided that  $\frac{k_p q}{\lambda} + k_d - \lambda = 0$ ,  $k_d > \frac{\lambda}{q}$ . Therefore, the control law (2.27) makes the convergence of the states to  $e_d \rightarrow 0$ ,  $e_s \rightarrow 0$ ,  $\tilde{\psi} \rightarrow 0$  and  $\omega \rightarrow \omega_d$ .

## 2.5 Simulation results and Experimental platform

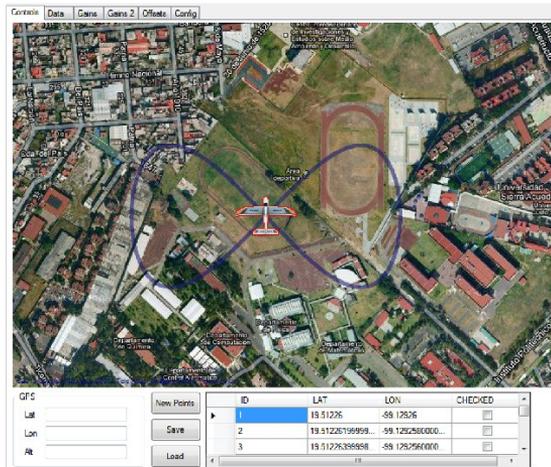
The simulation platform MAV3DSim is based on the open-source simulator CRRCsim [20], which was created based on a simulator (BASIC) developed by the National Aeronautics and Space Administration (NASA). Our simulator MAV3DSim implements the complete nonlinear model in six degrees of freedom

## 18 2. A Nonlinear Path-Following Strategy for a Fixed-Wing MAV

(6DoF). In addition, we have include aerodynamic forces generated by the aircraft control surfaces in order to to incorporate the 6DoF kinematic model. MAV3DSim has the capability to simulate the behavior of a specific model plane, additionally the user can change the aircraft aerodynamic coefficients. It also has a 3D representation to visualize the position and orientation of the plane.

Regarding communication capacities, one of the main advantages is that the user can establish a communication link with another computer via UDP protocol, and thereby send and receive information under a standard package of data used by the MNAV100CA Robotics Sensor Suites [21]. It means that MAV3DSim can simulate the sending data in a similar manner as does the aircraft. Such data includes the information provided by the IMU (attitude and angular rate), GPS (global position and velocities) and pitot tube (airspeed). Furthermore, it can receive control commands for the thrust and the control surfaces such as aileron, elevator and rudder.

The MAV3DSim is able to load maps directly from Google Maps or any other map provider (Fig. 2.2) and it can set the start point ant any location on Earth, with the latitude and longitude coordinates. On the map it can see the path that generates the plane in the tangential plane to the Earth. We can also produce a desired path generated by a parametric curve in a desired interval. The user is able to chose between several display options in order to



**Figure 2.2:** The MAV3DSim is able to load maps directly from Google Maps or any other map provider.

visualize the variables. Such display options include plots showing variables vs time and avionics instruments as those used in commercial aircrafts. Such avionics instruments includes: an altimeter which indicates the altitude relative

to a reference level at which the aircraft is flying. An attitude indicator, which shows the position of the longitudinal and transversal aircraft axes with respect to the natural horizon, this is obtained by reading the roll and pitch angles. A heading indicator, which displays the aircraft's heading with respect to magnetic north. And finally it includes an airspeed indicator, which gives the aircraft's speed relative to the surrounding air. Those four avionics instruments are depicted in Fig. 2.3. The MAV3DSim was created using object-oriented

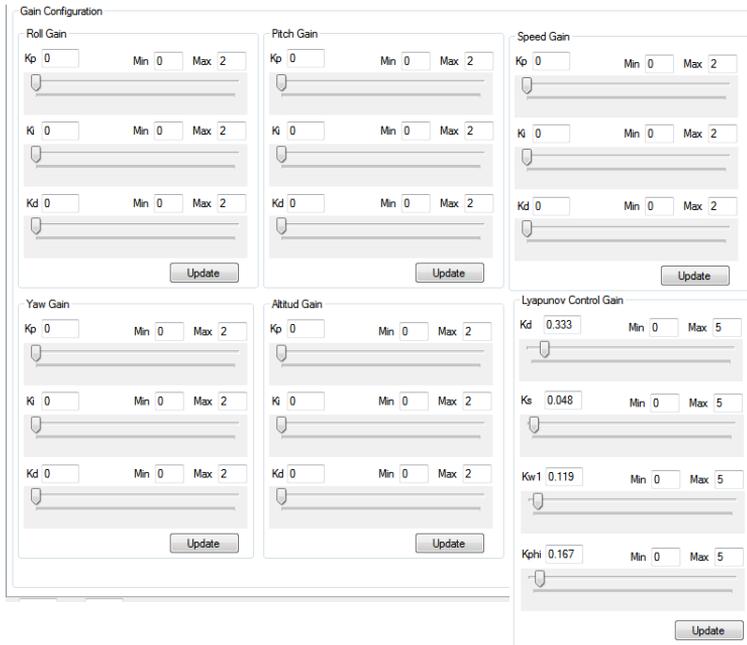


**Figure 2.3:** Avionics instruments of the MAV3DSim.

programming. The user can use many controllers as they need, for example when using multiple PID controllers for different aircraft states we only need to create a PID class and then make all instances we need. When tuning the controllers, it is important to choose gains online. As shown in Fig. 2.4, we have a graphic interface to tune all the gains involved in the controllers. To manage the simulator in a more natural way, a control interface for a Xbox 360<sup>TM</sup> controller was developed. Nevertheless the control interface can be plugged with in any commercial joystick. Furthermore, such interface allows the user to switch between manual flight and controlled flight, thus allowing the analysis of a series of controllers previously programmed.

The simulator presented in this section is intended to provide a development platform for the various controllers used on the aircraft. It is important to notice that with MAV3DSim we have a good approach to what we would have in the real world, so once that the proposed controller operates under the simulation platform we could move to a real platform without the risk of control failure. Another advantage of the MAV3DSim is the similar way in which it communicates with a simulator. So, it could communicate with a real platform and obtain a similar behavior as that obtained in the simulation platform.

## 20 2. A Nonlinear Path-Following Strategy for a Fixed-Wing MAV



**Figure 2.4:** Graphic interface of the MAV3DSim.

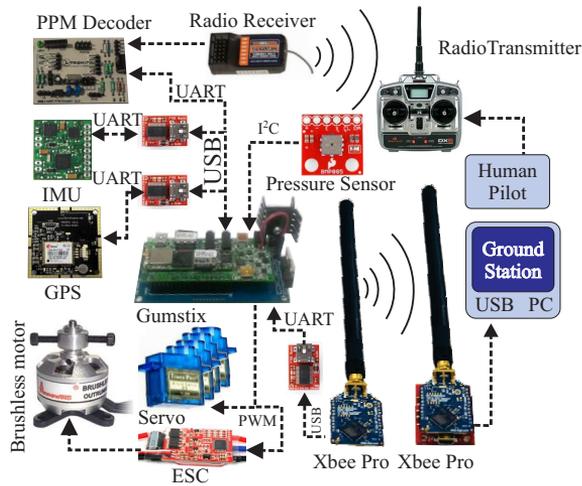
### 2.5.1 Experimental Platform

This section presents some characteristics of the experimental platform called *ElCerdo*. The airframe selected for this system is the commercial airplane TWINSTAR II which can be seen in Fig. 2.5. This vehicle is powered by a 1400kv brushless outrunner which is using a 8x4 propeller and a LiPo Battery of 2200 mAh which provides us a 820 g of thrust according to the test, this engine is in a pusher configuration and it has four servo motors used to actuate the control surfaces. Such configuration is very useful when testing control laws in the plane due to rear position of the propeller and in the case of an accident the motor has more chance to stay away from the ground and remain intact. The experimental platform components are: an on-board computer Gumstix<sup>TM</sup>Overo Fire equipped with an expansion board, an inertial measurement unit CH-Robotix CHR-6d, a GPS radio u-blox LEA-6S, a barometric pressure sensor BMP085 from BOSCH, which is used as an altitude sensor, an air speed sensor Freescale MPXV7002, and a communication device Maxstream XBee Pro 900 RPSMA. The Gumstix<sup>TM</sup> computer processes all the data from sensors, computes the control law and then sends the signal to the servo motors and the Brushless speed controller. The user is encouraged to see

for more details about the ElCerdo experimental platform in [22].



**Figure 2.5:** ElCerdo experimental platform.



**Figure 2.6:** Hardware configuration of the platform ElCerdo.

## 2.6 Experiments

This section illustrates the performance of the path-following strategy derived in the previous section. The simulation is performed with the MAV3DSim Simulator. In order to show the controller performance, we have chosen the following scenario: The trajectory has been chosen as

$$\begin{aligned} x_s &= \rho s \\ y_s &= \rho \sin(s) \end{aligned} \quad (2.29)$$

**Table 2.1:** Airplane Parameters.

Parameter	Value
Wingspan	1.6 m
Length	1.04 m
Weight	0.95 kg

**Table 2.2:** Control gains for the position control.

Gain	Value
$k_\delta$	0.143
$k_{\omega_1}$	0.405
$k_s$	1.5
$\psi_a$	0.048

**Table 2.3:** Control gains for the roll control.

Gain	Value
$k_p$	0.868
$k_d$	0.041

with  $\rho = \frac{1}{1000}$ . Table (2.1) summarizes the ElCerdo parameters. Such parameters can be set on the MAV3DSim simulator. A desired airspeed of  $V_t = 17$  m/s is achieved by means of a PD controller. An altitude controller is implemented in order to stabilize the airplane altitude to a desired value. Details of both controllers can be seen in [22]. From (2.2), the maximum permissible value for the roll angle is  $\phi_{max} = 17$  deg. The gains for the controller (2.20) are shown in Table 2.2. Table 2.3 shows the gains for the roll controller (2.27).

The reference and actual airplane paths are depicted in (2.7). Figures 2.8-2.13 shows the results of the simulations. Note the convergence of the errors in Fig. 2.10. Fig. 2.14 shows the airplane on flight and the trajectory in the MAV3DSim simulator.

Fig. 2.15 shows some experimental results of the proposed control obtained with the fixed-wing platform. Such figure shows the path-following of a circle. Furthermore, such experiments can be seen at the website:

<https://sites.google.com/site/gerardoflorescolunga/research/fixed-wing-uav>

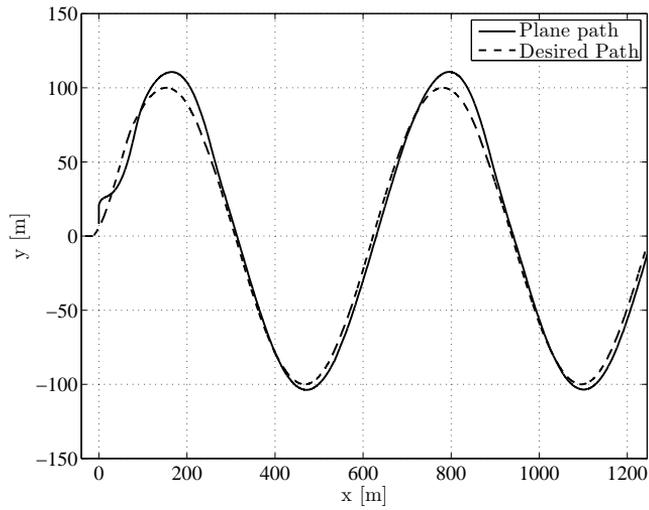


Figure 2.7: Fixed-wing MAV trajectory.

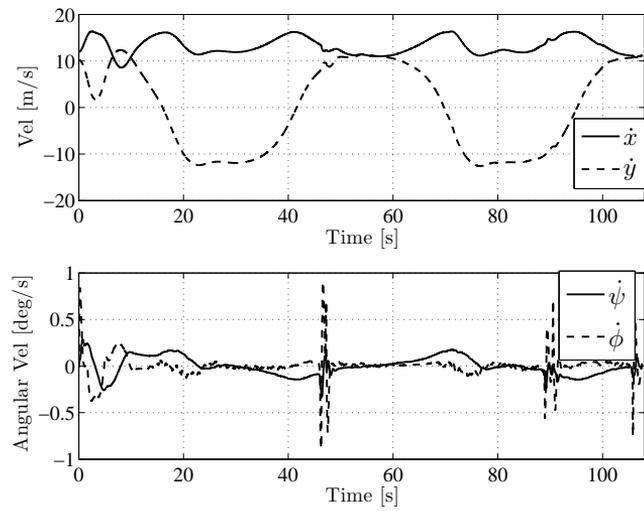


Figure 2.8: Translational and angular velocities.

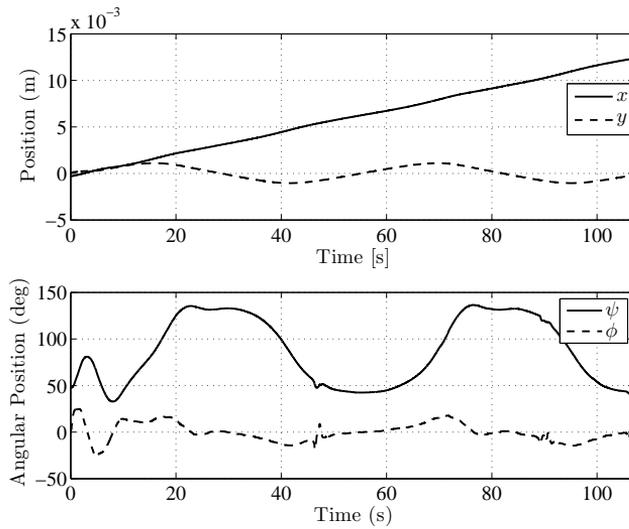


Figure 2.9: Position and angular position.

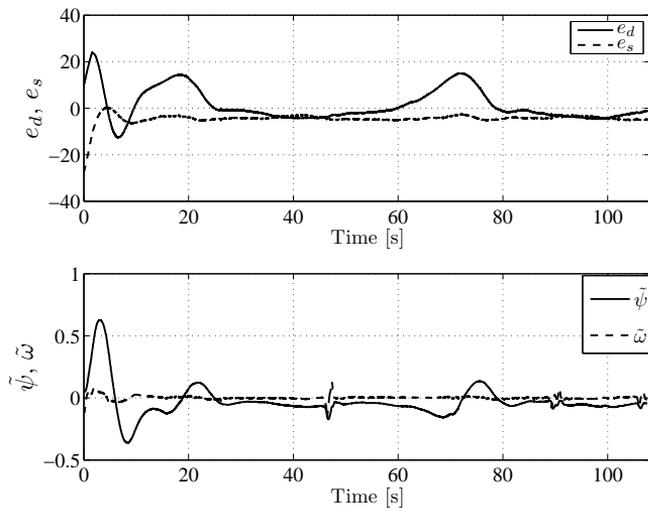
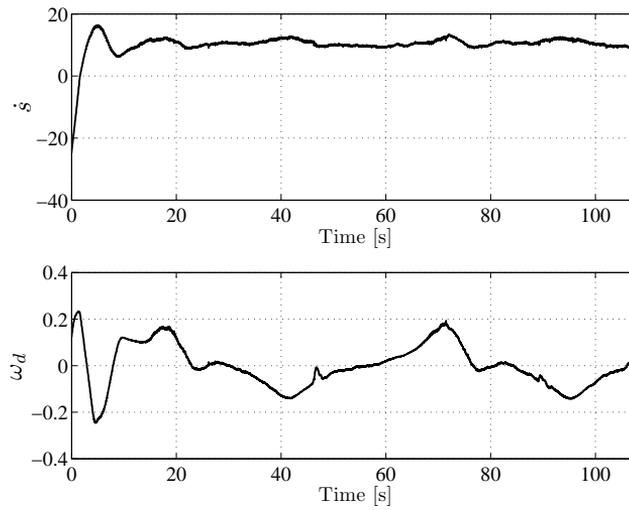
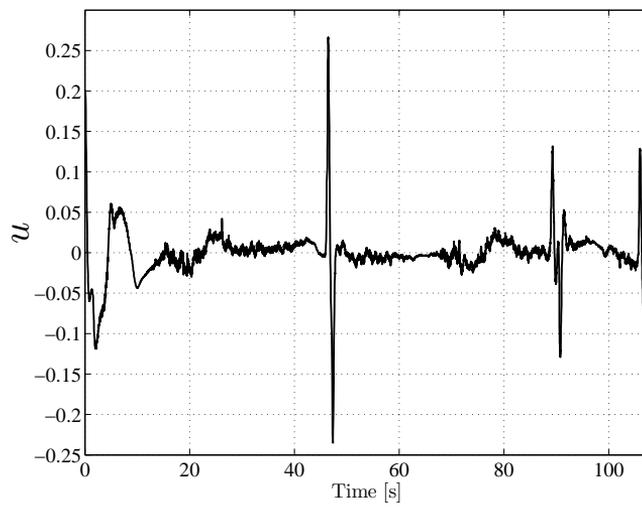


Figure 2.10: Errors.

**Figure 2.11:** Controllers.**Figure 2.12:** Control  $u$ .

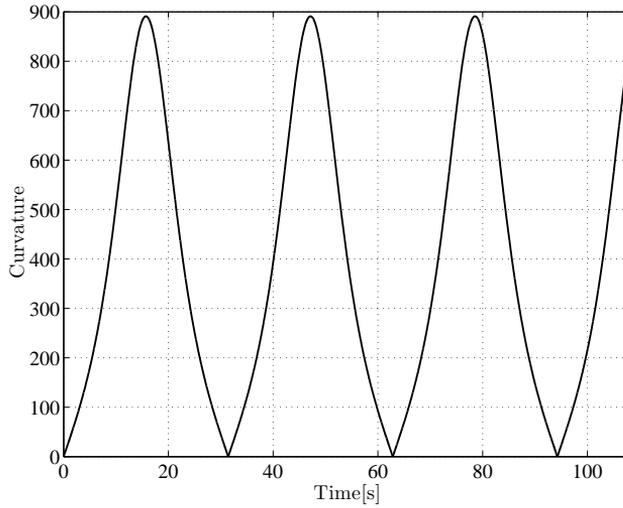
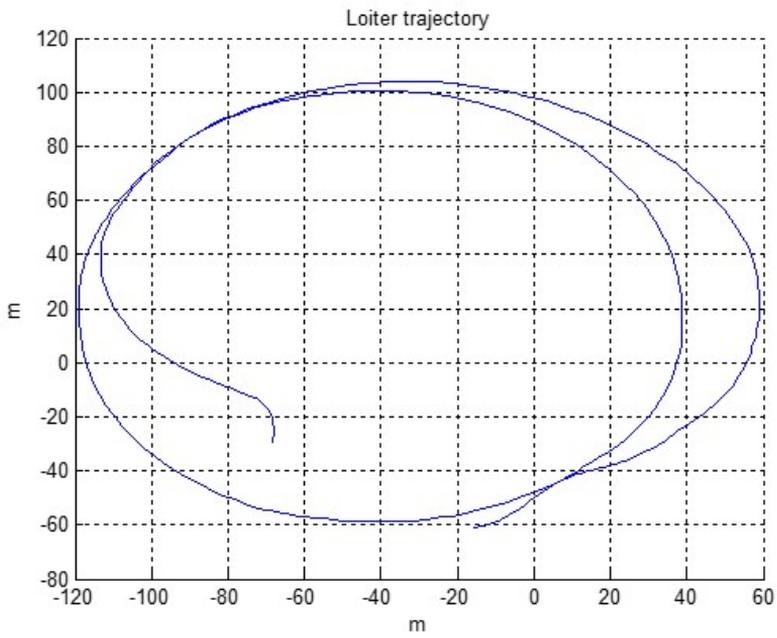


Figure 2.13: Curvature for the trajectory generated by (2.29).



Figure 2.14: Airplane in the simulator tracking a path.



**Figure 2.15:** Experimental results following a circle.



# 3

## DUBINS PATH GENERATION FOR A FIXED WING MAV

---

*A path generator is proposed for a fixed-wing Micro Aerial Vehicle (MAV). The Dubins paths serve as a strategy to find the shortest path for the non-holonomic model of the MAV. Dubins paths consist of three path segments which are based on straight lines or arcs of circle of a given radius. The Dubins path generation is combined with a nonlinear Lyapunov-based path-following control. We present a complete simulation environment in which the path generator and path following strategy are validated. As an example of application we propose the scenario in which a missing person is located in some known area and we use the path generator along with this path-following strategy applied to the fixed wing MAV to search and find this person.*

### Contents

---

<b>3.1</b>	<b>Introduction</b>	<b>30</b>
3.1.1	Coordinates Transformation	30
<b>3.2</b>	<b>Path Generation</b>	<b>33</b>
3.2.1	Dubins path <i>RSR</i>	34
3.2.2	Dubins path <i>RSL</i>	36
3.2.3	Dubins path <i>LSL</i>	38
3.2.4	Dubins path <i>LSR</i>	39
<b>3.3</b>	<b>Simulation Example</b>	<b>41</b>
3.3.1	MAV3DSim Simulation Platform	41
3.3.2	Simulation Scenario	43

---

## 3.1 Introduction

---

Dubins [23] showed that a car-like robot with initial prescribe heading can arrive to its final configuration, position and heading, with exactly three paths segments which are either arcs of circles with a minimal radius or straight lines segments. Reeds and Sheep [24] solve a similar problem in which the vehicle can move forward as well as backward. Kavaraki and Svestka [25] use the Probabilistic Road Map (PRM) method which explore all the possible paths within the space surrounding the vehicle and finally select the lowest cost route. Other planning techniques used by Kuwata and Richards [26] are based on optimizations methods, such as Mixed Integer Linear Programming or Model Predictive Control techniques. Mehta and Egerstedt [27] used optimal control for constructing control programs from a given collection of motion primitives.

In this chapter we present a path generator for a fixed-wing MAV using a reduced kinematic version of the lateral dynamics of an airplane, with constant altitude and velocity. This path generator uses the Dubins paths to generate the new path from the current position and direction of the plane to the desired position and direction. We use the nonlinear Lyapunov-based path-following strategy from our previous work [28] in order to follow the generated path. We combine the path generator with the path-following and validate it in a complete 6DOF simulation environment. A test scenario was developed in which a person(point of interest) is lost in a known area, we use the path generator to define a path to sweep this known area to search and find this point of interest.

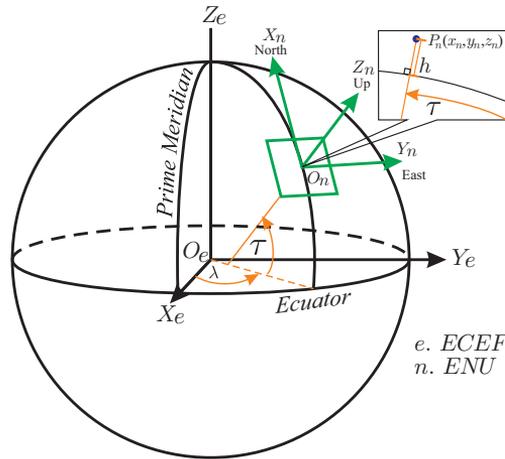
The chapter is organized as follows. In Section 3.1.1 we present the reference frame transformation needed to transform the geodetic coordinates (*Latitude, Longitude, Height*) to a local tangent reference frame ( $x, y, z$ ). Section 3.2 addresses the path generation for the Dubins paths. The simulation carried out for validate the path generator in combination with the path-following strategy are presented in Section 3.3.

### 3.1.1 Coordinates Transformation

---

The geodetic coordinate system is used in many fields, such as: navigation, surveying and cartography, in order to define the position of an objet on the

Earth's surface we use a set of three values called *geodetic coordinates* [29]. However, the geodetic coordinates lack of an intuitive understanding of distance, unlike other coordinate systems as the local East, North, Up (ENU) Cartesian coordinate system. The local ENU coordinates are formed from a plane tangent to the Earth's surface fixed to a specific location and it is known as a Local Tangent Plane (LTP). By convention the east axis is labeled  $x$ , the north  $y$  and the up  $z$ . The three different coordinate systems are represented in the Fig. 3.1.



**Figure 3.1:** Geodetic, ECEF and ENU coordinates frames.

### 3.1.1.1 Geodetic to ECEF coordinates

Here we introduce the equations to convert geodetic coordinates measurements to Local Tangent Plane coordinates. The method used passes through the Earth-Centered, Earth-Fixed (ECEF) rectangular coordinate system on the way to the Local Tangent Plane. Geodetic coordinates (latitude  $\tau$ , longitude  $\lambda$ , height  $h$ ) can be converted into ECEF coordinates using the following relationships:

$$\begin{aligned}
 X &= (N(\tau) + h) \cos \tau \cos \lambda \\
 Y &= (N(\tau) + h) \cos \tau \sin \lambda \\
 Z &= (N(\tau) (1 - e^2) + h) \sin \tau
 \end{aligned} \tag{3.1}$$

where

$$N(\tau) = \frac{a}{\sqrt{1 - e^2 \sin^2 \tau}}$$

The semi-major axis and the first numerical eccentricity of the ellipsoid are represented by  $a$  and  $e$ , respectively, the numeric value of this constants can be found in the definition of the World Geodetic System 1984 [30].  $N(\tau)$  is the distance from the surface to the to the Z-axis along the ellipsoid normal.

### 3.1.1.2 ECEF to Local Tangent coordinates

A local reference point is needed to perform a coordinate transformation from ECEF to the local ENU coordinates. The launching site position will serve as the local reference point. If the launching site is at  $(\lambda_0, \tau_0, h_0)$  in geodetic coordinates, then using the previous coordinate transformation we obtain  $(X_0, Y_0, Z_0)$ , the launching site expressed in ECEF coordinates. The aircraft location is defined as  $(\lambda, \tau, h)$ ; we use the same coordinate transformation to obtain  $(X, Y, Z)$ , the aircraft position expressed in ECEF coordinates. The vector pointing from the launching site to the aircraft in the ENU coordinate system is computed as follows

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \mathbf{R} \begin{bmatrix} X - X_0 \\ Y - Y_0 \\ Z - Z_0 \end{bmatrix} \quad (3.2)$$

where

$$\mathbf{R} = \begin{bmatrix} -\sin \lambda_0 & \cos \lambda_0 & 0 \\ -\sin \tau_0 \cos \lambda_0 & -\sin \tau_0 \sin \lambda_0 & \cos \tau_0 \\ \cos \tau_0 \cos \lambda_0 & \cos \tau_0 \sin \lambda_0 & \sin \tau_0 \end{bmatrix}$$

The World Geodetic System of 1984 (WGS84) [30] comprises a standard coordinate system and is one of the most used coordinate system used on GPS devices and we will use the coordinate transformations defined in this section to express the position of the airplane in the local ENU tangent plane which is suitable for the mathematical model and control purposes.

The Dubins Airplane model described in chapter 2 section 2.2 is used in the following sections. In such model, the position of the airplane can be represented by  $p(x, y, \psi)$  with  $\psi$  measured from the  $y$  axis and  $(x, y)$  measured in the local ENU reference frame.

## 3.2 Path Generation

In this section, the classical result of Dubins [23] is used as a basis for path generation. Dubins showed that the shortest path consist of exactly three path segments which are either *a)* arcs of a minimal radius or *b)* straight lines. The four different configurations for the Dubins paths which are composed by two curved segments and a straight line segment are arranged as shown in Fig. 3.2. The four cases of Dubins paths are *LSL*, *LSR*, *RSR*, *RSL*; in which *L* stands for Left, *R* stands for Right and *S* for Straight.

The first step in determining the Dubins paths is to choose what type of path must be used. We have the initial and final configuration of the airplane, this is the initial position  $p_i$ , the initial heading  $\psi_i$  the final position  $p_f$  and the final heading  $\psi_f$  and with every initial-final configuration we can generate the 4 types of Dubins paths,i.e. from the starting point it can turn to the right or the left and arrive to the final point from the right or the left. We choose the shortest path by comparing the distance between the center of the circles, see Fig. 3.3. The smallest distance between the center of the circles gives us the shortest Dubin path according to the Table 3.1.

Based on the initial and final configuration  $(p_i, \psi_i)$  and  $(p_f, \psi_f)$ , respectively, and the minimal turn radius  $\rho$  from (2.3), the center of each circle is computed

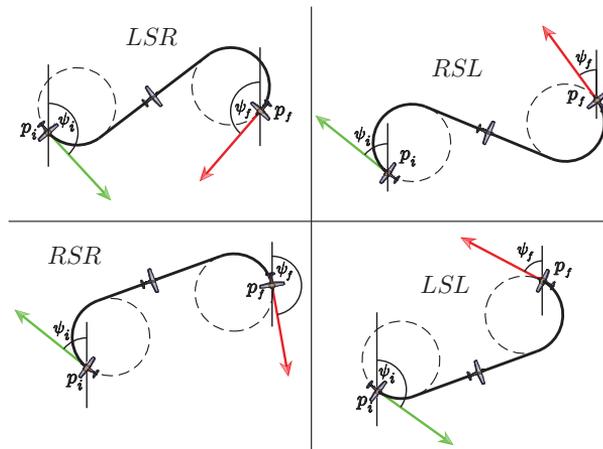


Figure 3.2: Dubins shortest paths.

Table 3.1: Dubins path selection.

Shortest distance	Dubins Path
$CR_i CR_f$	$RSR$
$CR_i CL_f$	$RSL$
$CL_i CL_f$	$LSL$
$CL_i CR_f$	$LSR$

as follows

$$CR_i = (x_{Ri}, y_{Ri}) = (x_i + \rho \cos \psi_i, y_i - \rho \sin \psi_i)$$

$$CL_i = (x_{Li}, y_{Li}) = (x_i - \rho \cos \psi_i, y_i + \rho \sin \psi_i)$$

$$CR_f = (x_{Rf}, y_{Rf}) = (x_f + \rho \cos \psi_f, y_f - \rho \sin \psi_f)$$

$$CL_f = (x_{Lf}, y_{Lf}) = (x_f - \rho \cos \psi_f, y_f + \rho \sin \psi_f)$$

### 3.2.1 Dubins path $RSR$

The initial and final configuration  $(p_i, \psi_i)$  and  $(p_f, \psi_f)$ , respectively, are given w.r.t. an inertial frame (Loca ENU frame). The  $RSR$  is generated by a clockwise rotation from the initial position describing an arc of radius  $\rho$  and

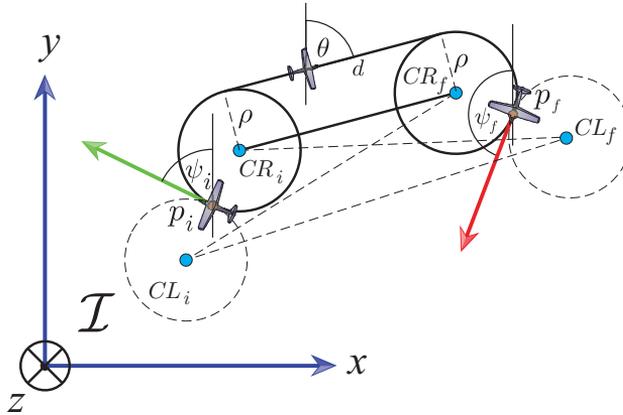
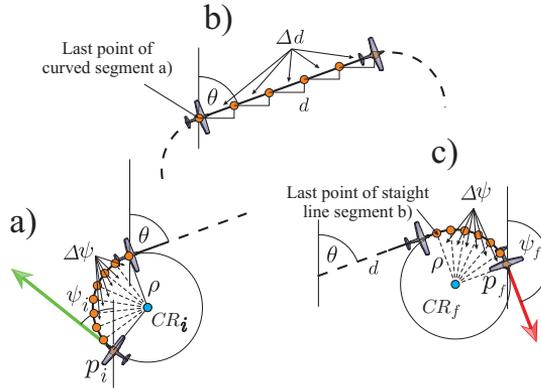


Figure 3.3: The Dubins paths are chosen by comparing the distance between the center of the circles segments.



**Figure 3.4:** The path generator algorithm produce an array of points  $p_n$

center  $CR_i$  with coordinates  $(x_{Ri}, y_{Ri})$  until the aircraft heading achieves an angle of  $\theta$  degrees. Then it follows a straight line segment  $d$ , finally it continues with a turn to the right describing an arc of radius  $\rho$  and center in  $CR_f$  with coordinates  $(x_{Rf}, y_{Rf})$  until the plane complete the final heading. In Fig 3.3 The angle  $\theta$  is the angle of the straight line segment  $d$  which is measured from the vertical  $y$  axis and computed as follows

$$\theta = \frac{\pi}{2} - \tan^{-1} \left( \frac{y_{Rf} - y_{Ri}}{x_{Rf} - x_{Ri}} \right) \quad (3.3)$$

The length  $\bar{d}$  of the straight line segment  $d$  equals the distance  $\overline{CR_i CR_f}$  between the center of the circles  $CR_i$  and  $CR_f$  and is computed as

$$d = \sqrt{(x_{Rf} - x_{Ri})^2 + (y_{Rf} - y_{Ri})^2} \quad (3.4)$$

The path generator algorithm produce an array of points  $p_n$  which starts in  $p_0 = p_i$  and ends in  $p_n = p_f$ .

The coordinates of the  $n - th$  point  $p_n$  of the arc segments are obtained by rotating the initial point  $p_i$  clockwise around  $CR_i$  as a center

$$p_n = \begin{bmatrix} x_n \\ y_n \end{bmatrix} = \begin{bmatrix} x_{Ri} + \rho \sin(\psi_n) \\ y_{Ri} + \rho \cos(\psi_n) \end{bmatrix} \quad (3.5)$$

where  $\psi_n$  starts at  $\psi_i$  and is incremented by given  $\Delta\psi$  each time. These procedure is repeated until  $\psi_n = \theta$ , see Figure 3.4a. Each point in the straight line segment is computed by incrementing the previous point  $p_{n-1}$  in a given

$\Delta d$  in direction of the angle  $\theta$  as follows

$$p_n = \begin{bmatrix} x_n \\ y_n \end{bmatrix} = \begin{bmatrix} x_{n-1} + \Delta d \sin(\theta) \\ y_{n-1} + \Delta d \cos(\theta) \end{bmatrix} \quad (3.6)$$

The elements  $p_n$  of the final segment are computed by rotating the final point of the straight line clockwise around  $CR_f$  as a center; see Fig. 3.4c.

$$p_n = \begin{bmatrix} x_n \\ y_n \end{bmatrix} = \begin{bmatrix} x_{Rf} + \rho \sin(\psi_n) \\ y_{Rf} + \rho \cos(\psi_n) \end{bmatrix} \quad (3.7)$$

where  $\psi_n$  starts in  $\theta$  and each time is incremented by  $\Delta\psi$ . This procedure is repeated until  $\psi_n = \psi_f$ ; see Figure 3.4c.

The complete path generation is summarized in algorithm 1

### 3.2.2 Dubins path RSL

This is the case where the closest circles are  $CR_i$  and  $CL_f$ , see Table 3.1. From the initial and final configuration,  $(p_i, \psi_i)$   $(p_f, \psi_f)$  the *RSL* path is generated with a clockwise rotation from the initial position  $p_i$  describing an arc of circle of radius  $\rho$  with center  $CR_i$  with coordinates  $(x_{Ri}, y_{Ri})$  until the heading aircraft achieves the angle  $\theta$ . Then it follows a straight line segment  $d$ , finally it will turn to the left describing an arc of radius  $\rho$  and center in  $CL_f$  with coordinates  $(x_{Lf}, y_{Lf})$  until the aircraft reaches the final heading. See Figure

---

**Algorithm 1** Generate Dubin path *RSR*

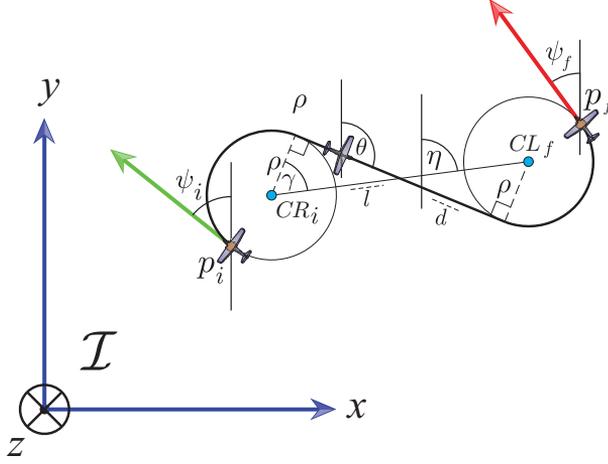
---

```

n = 1; p0 = pi
ψn = 0
while ψn ≤ θ do
    pn.x = xRi + ρ sin(ψn); pn.y = yRi + ρ cos(ψn)
    ψn = ψn + Δψ; n = n + 1
end while
dsum = 0
while dsum ≤  $\bar{d}$  do
    pn.x = pn-1.x + Δd sin θ; pn.y = pn-1.y + Δd cos(θ)
    dsum = dsum + Δd; n = n + 1
end while
while ψn ≤ ψf do
    pn.x = xRi + ρ sin(ψn); pn.y = yRi + ρ cos(ψn)
    ψn = ψn + Δψ; n = n + 1
end while

```

---



**Figure 3.5:** Right-Straight-Left (*RSL*) Dubins path.

3.5. In this case the angle  $\theta$  is computed is computed aided by the triangle formed by the center of the circle  $CR_i$  the midpoint of the segment  $d$  and the point of the circle tangent to the straight line  $d$  using the following formula

$$\theta = \eta - \gamma + \frac{\pi}{2} \quad (3.8)$$

where  $\eta$  is the angle of the segment  $\overline{CR_i CL_f}$  measured from the  $y$  axis as in Fig. 3.6 and is computed as follows

$$\eta = \frac{\pi}{2} - \tan^{-1} \left( \frac{y_{Lf} - y_{Ri}}{x_{Lf} - x_{Ri}} \right) \quad (3.9)$$

$\gamma$  is the angle between the segment  $\overline{CR_i CL_f}$  and the normal to the tangent point of circle  $CR_i$  and the segment  $d$ .  $\gamma$  is computed as follows

$$\gamma = \tan^{-1} \left( \frac{2\rho}{d} \right) \quad (3.10)$$

The length of the straight line segment  $d$  is computed with the distance  $l$  from the segment  $\overline{CR_i CL_f}$  and the radius  $\rho$  as

$$d = \sqrt{l^2 - 4\rho^2} \quad (3.11)$$

The coordinates of the  $n - th$  point  $p_n$  of the arc segments are obtained by rotating the initial point  $p_i$  clockwise around  $CR_i$  as a center using (3.5), see Figure 3.4a. Each point in the straight line segment is computed by incrementing the previous point  $p_{n-1}$  in a given  $\Delta d$  in direction of the angle

---

**Algorithm 2** Generate Dubin path *RSR*


---

```

n = 1; p0 = pi
ψn = 0
while ψn ≤ θ do
  pn.x = xRi + ρ sin(ψn); pn.y = yLi + ρ cos(ψn)
  ψn = ψn + Δψ; n = n + 1
end while
dsum = 0
while dsum ≤  $\bar{d}$  do
  pn.x = pn-1.x + Δd sin θ; pn.y = pn-1.y + Δd cos(θ)
  dsum = dsum + Δd; n = n + 1
end while
while ψn ≤ ψf do
  pn.x = xLi + ρ sin(ψn); pn.y = yLi + ρ cos(ψn)
  ψn = ψn + Δψ; n = n + 1
end while

```

---

$\theta$  as in (3.6) The elements  $p_n$  of the final segment are computed by rotating the final point of the straight line clockwise around  $CL_f$  as a center; see Fig. 3.4c.

$$p_n = \begin{bmatrix} x_n \\ y_n \end{bmatrix} = \begin{bmatrix} x_{Lf} + \rho \sin(\psi_n) \\ y_{Lf} + \rho \cos(\psi_n) \end{bmatrix}$$

where  $\psi_n$  starts in  $\theta$  and each time is incremented by  $\Delta\psi$ . This procedure is repeated until  $\psi_n = \psi_f$ ; see Figure 3.4c. The complete path generation is summarized in algorithm 2

### 3.2.3 Dubins path LSL

---

The *LSL* case is very similar to the *RSR* case but with the turns to the left instead of right and it occurs when the smallest distance between the circles (see Fig. 3.3) is  $\overline{CL_iCL_f}$ . The *LSL* path is generated with a counterclockwise rotation from the initial position  $p_i$  describing an arc of a circle of radius  $\rho$  and center in  $CL_i$  with coordinates  $(x_{Li}, y_{Li})$  until the aircraft heading achieves an angle of  $\theta$  degrees. The it follows a straight line segment  $d$  and finally it continues with the a turn to the left describing an arc of radius  $\rho$  and center in  $CL_f$  with coordinates  $(x_{Lf}, y_{Lf})$  until the airplane achieves the final heading  $\psi_f$ , as depicted in Fig. 3.6. The angle  $\theta$  measured from the vertical  $y$  axis is

$$\theta = \frac{\pi}{2} - \tan^{-1} \left( \frac{y_{Lf} - y_{Li}}{x_{Lf} - x_{Li}} \right) \quad (3.12)$$

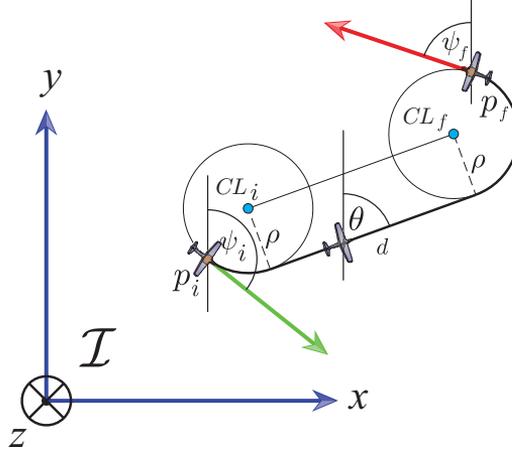


Figure 3.6: Left-Straight-Left (*LSL*) Dubins path.

The length of the segment  $d$  is the same as the distance between the center of the circles  $CL_i$  and  $CL_f$  and it is computed as

$$d = \sqrt{(x_{Rf} - x_{Ri})^2 + (y_{Rf} - y_{Ri})^2} \quad (3.13)$$

The coordinates of the  $n$ -th point  $p_n$  of the arc segments are obtained by rotating the initial point  $p_i$  counterclockwise around the  $CL_i$  as a center, as follows

$$p_n = \begin{bmatrix} x_n \\ y_n \end{bmatrix} = \begin{bmatrix} x_{Li} + \rho \sin(\psi_n) \\ y_{Li} + \rho \cos(\psi_n) \end{bmatrix} \quad (3.14)$$

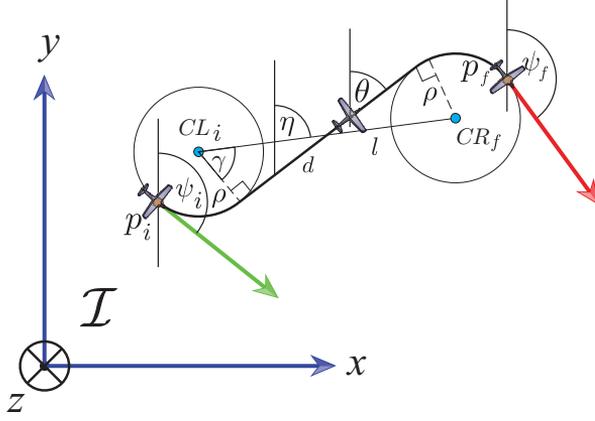
where  $\psi$  starts in zero and it is incremented each time in  $\Delta\psi$  until it reach the angle  $\delta$ . The straight line path is computed by the increment of  $\Delta s$  in the same direction as  $\delta$ . The last curved segment is a turn to the left and the segment coordinates are computed as follows

$$\begin{aligned} x_p &= x_{Lf} + \rho \sin(\psi_i + \psi) \\ y_p &= y_{Lf} + \rho \cos(\psi_i + \psi) \end{aligned} \quad (3.15)$$

### 3.2.4 Dubins path LSR

According to Table 3.1 the Dubins path *LSR* is when the shortest distance is the one between the circles  $CL_i$  and  $CR_f$ . The first segment of this path is a left turn which generated with a counter-clockwise rotation from the initial position  $p_i$  describing an arc of radius  $\rho$  with center in  $CL_i = (x_{Li}, y_{Li})$  until

the airplane reach the heading  $\delta$ , then it follows a straight line segment of length  $d$  and it finish with a right turn described by the arc of the circle of radius  $\rho$  with center in  $CR_f = (x_{Rf}, y_{Rf})$  and it will turn until it achieve the angle  $\psi_f$  as depicted in Figure 3.7. The computation of the angle  $\delta$  is aided by



**Figure 3.7:** Left-Straight-Left (*LSR*) Dubins path.

the triangle formed by the center of the circle  $CL_i$  the midpoint of the segment  $d$  and the point of the circle tangent to the segment  $d$  using the following equation

$$\delta = \eta + \gamma - \frac{\pi}{2} \quad (3.16)$$

where

$$\eta = \frac{\pi}{2} + \tan^{-1} \left( \frac{y_{Rf} - y_{Li}}{x_{Rf} - x_{Li}} \right)$$

and

$$\gamma = \cos^{-1} \left( \frac{2\rho}{d} \right)$$

The length of the straight line segment  $d$  is computed with the following equation

$$d = \sqrt{l^2 - 4\rho^2} \quad (3.17)$$

The coordinates of the first arc segment are computed by revolving the initial point  $p_i$  counterclockwise around  $CL_i$  as the center of the arc circle described by the following equations

$$\begin{aligned} x_p &= x_{Li} + \rho \sin(\psi_i + \psi) \\ y_p &= y_{Li} + \rho \cos(\psi_i + \psi) \end{aligned} \quad (3.18)$$

where  $\psi$  starts in zero and it is incremented in  $\Delta\psi$  each time until it reach the desired angle  $\delta$ . The straight line path segment is computed using an increment

of  $\Delta s$  in direction of the angle  $\delta$  as follows

$$\begin{aligned}x_p &= x_{p-1} + \Delta s \sin(\delta) \\y_p &= y_{p-1} + \Delta s \cos(\delta)\end{aligned}$$

### 3.3 Simulation Example

---

Simulations were done on a complete simulation platform, the MAV3DSim(Multi-Aerial Vehicle 3D Simulator) provide a complete 6 degrees of freedom (DoF) computer model of fixed wing aircraft. The MAV3DSim software layers are described briefly in this section. The application scenario is in the use of the path generation and path-following algorithms to command a desired path to the fixed wing MAV. The results from the simulation are presented at the end of this section.

#### 3.3.1 MAV3DSim Simulation Platform

---

The MAV3DSim is a custom C# .Net based application and implements a complete 6DoF nonlinear model. It has a 3D representation to visualize the position and orientation of the plane, also, it has the capability to load maps directly from Google Maps servers and set the launching site on any location on Earth. The trajectory generated by the plane can be seen on the map, this map is the tangential plane to the Earth.

The data generated by the simulator is coded in the same manner as the common sensors, i.e. it send data emulating an inertial measurement unit(IMU) sending inertial gyroscope, accelerometer and magnetometer, a GPS radio in the latitude/longitude format, altitude and airspeed. It can receive commands to move the control surfaces aileron elevators, rudder, and the thrust of the fixed-wing MAV. The position provided by the simulator is in a standard geodetic WGS84 Latitude( $\lambda$ ), Longitude( $\tau$ ) and Height( $h$ ), and we will use the transformation to the local tangent ENU described in Section 3.1.1.

The software layers, depicted in the Fig 3.8, are briefly described as follows

**3.3.1.1** Path Generator

---

This layer is in charge of the generation of paths using the Dubins path generation described in Section 3.2. It can generate new paths and maintain the old ones for later use. It is possible to interact online with the path generation and change the course of action of the aircraft in any time either by an autonomous action or by a human interaction. Once the path is fully generated, it is transmitted to the path-following strategy.

**3.3.1.2** Path-Following Strategy

---

The path-following control described in chapter 2 section 6.3 is implemented in this layer. The curvature calculation is given for any path described by the parametric curve  $\alpha(s)$ , but in the Dubins paths there are only curves of a circle of radius  $\rho$  and straight lines. The curvature of a straight line is zero and using 2.14 we can calculate the curvature of the circle of radius  $\rho$  as follows

$$\mathcal{C}_c = \frac{\rho^2 \sin^2 \psi + \rho^2 \cos^2 \psi}{(\rho^2 \sin^2 \psi + \rho^2 \cos^2 \psi)^{3/2}} = \frac{1}{\rho}$$

The path is stored in an array of  $n$  points of the form  $(x_m, y_m)$  starting with  $m = 0$  then the path following strategy computes the errors  $e_s, e_d, \tilde{\psi}$  from (2.15) and (2.7), with this information it computes the control input  $\omega$  and  $\dot{s}$  from (2.20). The control  $\omega$  is a desired heading rate and is induced into the aircraft dynamics through the roll angle using (2.4). The computed roll angle  $\phi$  will be used by the low level autopilot

**3.3.1.3** Low Level Autopilot

---

The role of low-level autopilot is to stabilize the aircraft in roll and pitch angles, maintain a constant altitude and airspeed by implementing a PD controller for each dynamic (roll, pitch, altitude and airspeed). The altitude and airspeed setpoints are manually introduced by a graphic user interface, the altitude controller outputs the pitch setpoint and the roll setpoint is obtained from the path following controller.



**Figure 3.8:** Communication scheme between the MAV3DSim and the CRRCSim.

#### 3.3.1.4 Aircraft Dynamics

This layer integrates the set of differential equations representing the aircraft dynamics. The input of this layer are the inputs of the low level autopilot layer and the outputs are the data from the simulated sensors: GPS position, aircraft attitude, airspeed. The aircraft dynamics layer sends the outputs to the upper layers.

#### 3.3.2 Simulation Scenario

We use the MAV3DSim simulation platform along with the Dubins path generator and the path-following strategy previously described to present a simulation scenario. The description of the scenario is as follows: A person is missing and is located somewhere in a known area. The main task of the MAV is to find this person, so it will sweep this area in order to find the missing person. First we need to define the search area as a rectangle with the aid of a user interface, then using the proposed path generator algorithm define the a path for sweeping the rectangle area. The starting point of the path will be one of the corners of the rectangle and it selects the closest to the current position of the MAV as depicted in Fig. 3.10. The MAV will travel along the path until it is sufficiently close to the lost person (red dot in Fig.3.10). When the missing person is found a circular path is generated to surround the missing person. The simulation can be seen at

[https://www.youtube.com/watch?v=\\_AUW8\\_g-jb0](https://www.youtube.com/watch?v=_AUW8_g-jb0)



Figure 3.9: The path generated to weep the search area.



Figure 3.10: The path generated to weep the search area.

# Part II

THE

QUADROTOR



# 4

## LYAPUNOV-BASED CONTROLLER FOR A MINI-UAV USING SINGULAR PERTURBATION THEORY

---

*In this chapter, a Lyapunov-based control using singular perturbation theory is proposed and applied on dynamics of a miniature unmanned aerial vehicle (MAV). Such controller is designed taking into account the presence of the small parameter  $\varepsilon$  on vehicle dynamics, which causes a time-scale separation between the attitude and translational dynamics of the MAV. The stability analysis is demonstrated by presenting a scenario in which the time-scale property arises on the the MAV dynamics. Simulations are derived and presented to demonstrate the effectiveness of the control law. The proposed controller has been applied to a Quad-plane MAV experimental platform, in order to validate the performance and to show the time-scale property.*

### Contents

---

<b>4.1</b>	<b>Introduction</b>	<b>48</b>
<b>4.2</b>	<b>Problem Description</b>	<b>49</b>
<b>4.3</b>	<b>System Description</b>	<b>50</b>
4.3.1	Modeling for the Singular Perturbation Problem	51
<b>4.4</b>	<b>Controller Design</b>	<b>51</b>
4.4.1	Real Input Controls	57
<b>4.5</b>	<b>Simulations and Experimental Results</b>	<b>58</b>
4.5.1	Numerical Results	58
4.5.2	Experimental Results	59

---

## 4.1 Introduction

The growing development of technologies like microcomputers, vision systems, IMU's and other sensor devices, has increased the interest and research on MAVs. As a consequence, the control and robotics community have been interested on developing controllers that can deal with the complexity of the MAV dynamics. It is well known that the sub-actuated dynamics of the MAV has a fast dynamics formed by the orientation subsystem, and a slow dynamics formed by the translational subsystem [31], [32]. Thus, the stabilization and tracking trajectory problem applied on a MAV can be addressed by using the fact that there exists a time scale separation between the translational and rotational dynamics, leading to a hierarchical control. The hierarchical control scheme presents two or more separate controllers that can be designed separately to successively stabilize the dynamics of the vehicle, and can be used in order to design position and orientation controllers leading to simplify the problem.

Some researchers have exploited the aforementioned hierarchical structure [31], [33], [34], [35], [6], [36], but they use the time scale separation just to justify the implementation of two different controllers: a controller for the attitude subsystem and a controller for the translational subsystem. However, the vast majority of the works only make mention of this feature, and do not address the problem in a theoretical point of view, using tools like the singular perturbation theory. This is because in general, the presence of such property makes the problem hard from the numerical solution point of view [37]. However, in a few works [38], [39] such phenomena is pointed out justifying the time-scale separation property.

In this chapter, we present a stabilization analysis of the MAV dynamics in the six degrees of freedom, by using the singular perturbation theory. The considered system presents a not pure strict-feedback structure, and the control vectors formed by the force and trust have a different relative degree w.r.t. MAV position. In order to overcome this difficulty we propose to use a dynamic extension on the force control vector. Furthermore, an analysis of the system is considered by introducing the parameter  $\varepsilon$  on the MAV dynamics, in order to illustrate the time-scale separation between the attitude and translational dynamics of the MAV. The proposed control law should be tested at simulation level, and be implemented on an experimental platform.

The chapter is organized as follows. In Section 4.2 the problem description and the singular perturbed problem are introduced. The MAV model is presented

as a singular perturbation structure in Section 4.3. In section 4.4 the controller is developed and the corresponding stability proof is derived. Simulations are given in Section 4.5, in order to prove the proper operation of the control law. In addition, experimental results tested on the *Quad-plane* MAV platform [34], are shown.

## 4.2 Problem Description

Singular perturbation and hence, the time-scale character is often associated with a small parameter  $\varepsilon$ , multiplying some of the state variables of the considered system. One difficult is that such parameter does not appear in the desired form or it may not be identifiable at all. Frequently, only by past experience and physical insights, one can know that a particular system has fast and slow modes. In the cases where it is impossible to identify the parameter  $\varepsilon$ , one can artificially introduce  $\varepsilon$  to be associate with the fast dynamics.

There exist three different approaches for the selection of time scales [37]: linearization of the state equations, transformation of the state equations and direct identification of small parameters. However, in many aerospace problems, no singular parameter appears explicitly on the dynamic equations. In such cases, the parameter  $\varepsilon$  may be artificially inserted for presenting a singular perturbation structure, which is shown below

$$\begin{aligned}\dot{x} &= f(x, z, u) \\ \varepsilon \dot{z} &= g(x, z, u)\end{aligned}\tag{4.1}$$

where  $x \in \mathbb{R}^n$ ,  $z \in \mathbb{R}^m$ ,  $f$  and  $g$  are smooth functions,  $u \in \mathbb{R}^r$  is a control input and  $\varepsilon \in (0, 1]$ .

The first goal of the chapter is to present the MAV mathematical model as a singular perturbation structure, as in (4.1). Then, we proceed to design a Lyapunov-based control law, which stabilizes the vehicle on a predefined three-dimensional position. The second goal is to investigate the  $\varepsilon$  values in which the proposed control can be implemented.

### 4.3 System Description

In this section, an idealized mathematical model of the MAV is described. Such model will be suitable for developing the purposed control law. We will consider the nonlinear rigid body dynamics in terms of rotational and translational dynamics [40] given by

$$\begin{aligned}
 \dot{\xi} &= v \\
 \dot{v} &= \frac{RF}{m} - ge_{z_I} \\
 \dot{R} &= RS(\Omega) \\
 \dot{\Omega} &= -J^{-1}\Omega \times J\Omega + J^{-1}\tau
 \end{aligned} \tag{4.2}$$

where  $\xi = (x, y, z)^T$  and  $v = (v_x, v_y, v_z)^T$  are respectively, position and velocity of the MAV relative to the inertial frame  $\mathcal{I} = (e_{x_I}, e_{y_I}, e_{z_I})$ .  $R \in SO(3)$  is the rotational matrix representing MAV orientation in body coordinate frame  $\mathcal{B} = (e_{x_B}, e_{y_B}, e_{z_B})$  w.r.t.  $\mathcal{I}$ ,  $\Omega \in \mathfrak{R}^3$  is the body angular velocity vector.  $F \in \mathfrak{R}^3$  and  $\tau \in \mathfrak{R}^3$  are the force and torque, respectively applied at the center of mass of the MAV and specified w.r.t.  $\mathcal{B}$ .  $J \in \mathfrak{R}^3$  is the inertia matrix,  $m$  is the mass of the body,  $ge_{z_I}$  is the gravitational force and  $e_{z_I} = (0, 0, 1)$  is a unit vector. In (4.2),  $SO(3)$  denotes the special orthogonal group of  $\mathfrak{R}^{3 \times 3}$ , and  $so(3)$  is the group of antisymmetric matrices of  $\mathfrak{R}^{3 \times 3}$ . Also, we define by  $S(v)$  the operator from  $\mathfrak{R}^3 \rightarrow so(3)$  such that

$$\forall v \in \mathfrak{R}^3, S(v) = \begin{pmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{pmatrix} \tag{4.3}$$

where  $v_i$  denotes the  $i$ th component of vector  $v$ . Thus,  $S(v)\Omega = v \times \Omega$ . It is important to note that the dynamic model (4.2) is not in pure strict-feedback structure, and control vectors  $F$  and  $\tau$  have different relative degree w.r.t. the position  $\xi$ . For this reason we take the dynamic extension of control  $F$  as

$$\ddot{F} = \tilde{F} \tag{4.4}$$

In this way, the actual control  $F$  and its first time derivative  $\dot{F}$ , become internal variables of a dynamic controller [41]. Thus, it is possible to represent (4.2) as

$$\begin{aligned}
 \dot{\xi} &= v \\
 \dot{v} &= X \\
 \dot{X} &= Y \\
 \dot{Y} &= \frac{R}{m} (\tilde{F} - S(F)\tilde{\tau} + 2S(\Omega)\dot{F} + S(\Omega)S(\Omega)F)
 \end{aligned} \tag{4.5}$$

where new states  $X$ ,  $Y$  and control input  $\tilde{\tau}$  are defined as

$$\begin{aligned} X &:= \frac{RF}{m} - ge_{z_I} \\ Y &:= \dot{X} = \frac{R}{m}(S(\Omega)F + \dot{F}) \\ \tilde{\tau} &:= \dot{\Omega} \end{aligned} \quad (4.6)$$

Thereby, new inputs  $\tilde{F}$ ,  $\tilde{\tau}$  have a relative degree equal to four w.r.t. the state  $\xi$ . Thus, they can be assigned at the same stage, eliminating the problem of the presence of an aggressive control, which may lead to extreme ill-conditioning of the remaining closed-loop system [42].

### 4.3.1 Modeling for the Singular Perturbation Problem

In many MAV problems or more generally, in more aerospace problems, no singular perturbation parameter appears explicitly on the mathematical model. In such cases, a singular perturbation parameter may be artificially inserted to define a rapid response of a certain dynamic w.r.t. other. In other cases, this parameter can be inserted to suppress the variables in the equations that are expected to have relatively negligible effects. The slow-fast time scale character is often associated with a small parameter multiplying some of the state variables of the state equations describing a physical system. However, often that parameter may not be identifiable at all and only by physical insight and past experiences does one know that the system has fast and slow dynamics. Experience indicates that among the state variables, the position and velocity are slow relative to the dynamic of the Euler angles. It is this separation of the states velocities, that motivates to formulate a singular perturbation structure as follows

$$\begin{aligned} \dot{\xi} &= v \\ \dot{v} &= X \\ \varepsilon \dot{X} &= Y \\ \varepsilon \dot{Y} &= \frac{R}{m}(\tilde{F} - S(F)\tilde{\tau} + 2S(\Omega)\dot{F} + S(\Omega)S(\Omega)F) \end{aligned} \quad (4.7)$$

## 4.4 Controller Design

In this section a control strategy for stabilization of (4.7) is proposed. The controller will be successively designed as presented below.

From the last equation of (4.7) we can write

$$\frac{R}{m} (\tilde{F} - S(F)\tilde{\tau} + 2S(\Omega)\dot{F} + S(\Omega)S(\Omega)F) = u \quad (4.8)$$

where  $u$  will be taken as control input. By adding  $X + Y$  in both sides of (4.8), this equation stays in balance, then we get a feedback connection in (4.7) as follows

$$\dot{\xi} = v \quad (4.9)$$

$$\dot{v} = X \quad (4.10)$$

$$\varepsilon \dot{X} = Y \quad (4.11)$$

$$\varepsilon \dot{Y} = Y + X + u \quad (4.12)$$

To be consistent with the notation used in (4.1), vectors  $x$  and  $z$  are given by  $x = [\xi \ v]^T$ ,  $z = [X \ Y]^T$  and  $f(x, z, u) = [v \ X]^T$ ,  $g(x, z, u) = [Y \ Y + X + u]^T$ .

The goal is to design a feedback control law which stabilizes the system (4.9)-(4.12) at the equilibrium point  $\xi = 0$ ,  $v = 0$ ,  $X = 0$ ,  $Y = 0$  and prove the asymptotic stability of the closed-loop system. For achieve this goal, we need to investigate a candidate Lyapunov function for such system. The key idea is to analyze the system separately, beginning with the slow subsystem (4.9)-(4.10) and continuing with the fast subsystem (4.11)-(4.12). Then, find a control for each subsystem and investigate their corresponding candidate Lyapunov functions. Finally, combining both candidate Lyapunov functions in a proper way, we find the candidate Lyapunov function for the entire system (4.9)-(4.12).

We begin by analyzing the slow system (4.9)-(4.10). Let us assume that the open-loop system (4.9)-(4.10) is a standard singularly perturbed system for every  $u \in B_u \subset \mathbb{R}^3$ , that is to say, the equations

$$\begin{aligned} 0 &= Y \\ 0 &= Y + X + u \end{aligned} \quad (4.13)$$

have a unique root  $z = h(x, u)$ . Such control  $u$  will be composed of the sum of slow and fast controls

$$u = u_s + u_f \quad (4.14)$$

where

$$u_s = \Xi_s(\xi, v) \quad (4.15)$$

is a feedback function of the states that compose the fast system dynamics

(4.11)-(4.12), and

$$u_f = \Xi_f(\xi, v, X, Y) \quad (4.16)$$

is a feedback function depending on the states  $(\xi, v, X, Y)$ . In order to find the control (4.15), we see from (4.13) that

$$X = -u_s \quad (4.17)$$

and then, we propose the slow controller  $u_s$  given by

$$u_s = k_{P_s}\xi + k_{D_s}v \quad (4.18)$$

Thus, with the control (4.18) the closed-loop reduced system (4.9)-(4.10) results in

$$\begin{aligned} \dot{\xi} &= v \\ \dot{v} &= -k_{P_s}\xi - k_{D_s}v \end{aligned} \quad (4.19)$$

Using the candidate Lyapunov function

$$V(\xi, v) = \frac{\lambda}{2}\xi^T\xi + \frac{q}{2\lambda}v^T v + \xi^T v \quad (4.20)$$

with parameters  $\lambda > 0$ ,  $q > 0$  properly chosen, the derivative of (4.20) w.r.t. (4.19) is given by

$$\dot{V}(\xi, v) = -k_{P_s}\xi^T\xi - \left(\frac{q}{\lambda}k_{D_s} - 1\right)v^T v \quad (4.21)$$

We need to investigate a scalar function  $\psi(\cdot)$  of vector arguments which vanish only when its arguments are zero, such that

$$\dot{V}(\xi, v) \leq -\alpha_1\psi^2(\xi, v) \quad (4.22)$$

where  $\alpha_1 > 0$ . The inequality (4.22) holds with the scalar function

$$\psi(\xi, v) = \left\| \begin{pmatrix} |\xi| \\ \rho|v| \end{pmatrix} \right\| \quad (4.23)$$

where  $\|\cdot\|$  is the Euclidean norm of a vector and  $\rho$  is an arbitrary positive number to be chosen.

The boundary layer model [43] of the closed-loop system (4.9)-(4.12) is defined as

$$\begin{aligned} \frac{dX}{d\tau_t} &= Y \\ \frac{dY}{d\tau_t} &= X + Y + k_{P_s}\xi + k_{D_s}v + u_f \end{aligned} \quad (4.24)$$

where

$$\frac{d\tau_t}{dt} = \frac{1}{\varepsilon}$$

We proceed to design a fast control law (4.16). One inspection of (4.24) suggests to chose the fast control as

$$u_f = -3(X + Y + k_{P_s}\xi + k_{D_s}v) \quad (4.25)$$

The control (4.25) needs to fulfill certain requirements for system (4.9)-(4.12) to remain a standard singularly perturbed system. The first one is that  $u_f = \Xi_f(\xi, v, X, Y)$  be inactive for  $z = h(x, u_s)$ , i.e.  $\Xi_f(x, h(x, \Xi_f(x))) = 0$ , then

$$\Xi_f(x, h(x, \Xi_f(x))) = -3(X + Y + k_{P_s}\xi + k_{D_s}v) = 0 \quad (4.26)$$

holds with  $X = -k_{P_s}\xi - k_{D_s}v$  and  $Y = 0$ . The requirement (4.26) guarantees that  $z = h(x, \Xi_s(x))$  is a root of

$$\begin{aligned} 0 &= Y \\ 0 &= Y + X + u_s + u_f \end{aligned} \quad (4.27)$$

In addition, (4.27) should have a unique root  $z = h(x, \Xi_s(x))$  in a certain domain of interest  $B_x \times B_z$ , which is easy to verify from (4.18), (4.25). We proceed to investigate a candidate Lyapunov function  $W$  such that

$$\frac{\partial W}{\partial z} g(x, z, \Xi_s(x) + \Xi_f(x, z)) \leq -\alpha_2 \phi^2(z - h(x, \Xi_s(x))) \quad (4.28)$$

$\forall (x, z) \in B_x \times B_z$ , where  $\alpha_2 > 0$  and  $\phi(\cdot)$  is a scalar function of vector arguments which vanish only when its arguments are zero. Using the candidate Lyapunov function

$$\begin{aligned} W &= \frac{\lambda_w}{2} Y^T Y + \frac{q_w}{2\lambda_w} (X + k_{P_s}\xi + k_{D_s}v)^T (X + k_{P_s}\xi + k_{D_s}v) \\ &\quad + (Y)^T (X + k_{P_s}\xi + k_{D_s}v) \end{aligned} \quad (4.29)$$

the time derivative of (4.29) can be calculated as

$$\begin{aligned} \dot{W} &= (-2\lambda_w + 1)Y^T Y + (-2\lambda_w + \frac{q_w}{\lambda_w} - 2)(Y^T)(X + k_{P_s}\xi + k_{D_s}v) \\ &\quad - 2(X + k_{P_s}\xi + k_{D_s}v)^T (X + k_{P_s}\xi + k_{D_s}v) \end{aligned} \quad (4.30)$$

where  $\lambda_w > \frac{1}{2}$  and  $-2\lambda_w + \frac{q_w}{\lambda_w} - 2 < 0$ . The inequality

$$\dot{W} \leq -\alpha_2 \phi^2(z - h(x)) \quad (4.31)$$

holds with  $\alpha_2 > 0$  and with the function

$$\phi(\xi, v, X, Y) = \left\| \begin{pmatrix} |X + k_{P_s}\xi + k_{D_s}v| \\ \rho_w |Y| \end{pmatrix} \right\| \quad (4.32)$$

where  $\rho_w$  is an arbitrary positive number, and  $\alpha_2$  can take the value  $\alpha_2 = 1$ . In order to complete a Lyapunov function for the entire system (4.9)-(4.12), the aforementioned Lyapunov function candidates, (4.20) for the slow system and (4.29) for the fast system, should verify the next interconnection conditions [43]

$$\begin{aligned} \frac{\partial W}{\partial x} f(x, z, \Xi_s(x) + \Xi_f(x, z)) &\leq \gamma \phi^2(z - h(x, \Xi_s(x))) \\ &\quad + \beta_2 \psi(x) \phi(z - h(x, \Xi_s(x))) \end{aligned} \quad (4.33)$$

$$\begin{aligned} \frac{\partial V}{\partial x} [f(x, z, \Xi_s(x) + \Xi_f(x, z)) - f(x, h(x, \Xi_s(x)), \Xi_s(x))] &\leq \\ \beta_1 \psi(x) \phi(z - h(x, \Xi_s(x))) \end{aligned} \quad (4.34)$$

We proceed to verify inequalities (4.33) and (4.34). Developing inequality (4.33) by using (4.23), (4.32) it leads to

$$\begin{aligned} [k_{P_s}v + k_{D_s}X]^T \left[ \frac{q_w}{\lambda_w} (X + k_{P_s}\xi + k_{D_s}v) + Y \right] &\leq \gamma |X + k_{P_s}\xi + k_{D_s}v|^2 \\ + \gamma \rho_w^2 |Y|^2 + \beta_2 \left( \sqrt{|\xi|^2 + \rho^2 |v|^2} \right) &\left( \sqrt{|X + k_{P_s}\xi + k_{D_s}v|^2 + \rho_w^2 |Y|^2} \right) \end{aligned} \quad (4.35)$$

adding and subtracting the term  $k_{D_s}[k_{P_s}\xi + k_{D_s}v]^T \left[ \frac{q_w}{\lambda_w} (X + k_{P_s}\xi + k_{D_s}v) + Y \right]$  to the LHS of (4.35) and by using the triangle inequality, it leads to

$$\begin{aligned} \frac{\partial W}{\partial x} f(x, z, \Xi_s(x) + \Xi_f(x, z)) &\leq \\ |k_{P_s}v + k_{D_s}(-k_{P_s}\xi - k_{D_s}v)| &\left| \frac{q_w}{\lambda_w} (X + k_{P_s}\xi + k_{D_s}v) + Y \right| \\ + k_{D_s} |X + k_{P_s}\xi + k_{D_s}v| &\left| \frac{q_w}{\lambda_w} (X + k_{P_s}\xi + k_{D_s}v) + Y \right| \end{aligned} \quad (4.36)$$

Thus, if we can verify that

$$\begin{aligned} k_{D_s} |X + k_{P_s}\xi + k_{D_s}v| &\left| \frac{q_w}{\lambda_w} (X + k_{P_s}\xi + k_{D_s}v) + Y \right| \leq \\ \gamma |X + k_{P_s}\xi + k_{D_s}v|^2 &+ \gamma \rho_w^2 |Y|^2 \end{aligned} \quad (4.37)$$

and

$$\begin{aligned} |k_{P_s}v + k_{D_s}(-k_{P_s}\xi - k_{D_s}v)| &\left| \frac{q_w}{\lambda_w} (X + k_{P_s}\xi + k_{D_s}v) + Y \right| \\ \leq \beta_2 \left( \sqrt{|\xi|^2 + \rho^2 |v|^2} \right) &\left( \sqrt{|X + k_{P_s}\xi + k_{D_s}v|^2 + \rho_w^2 |Y|^2} \right) \end{aligned} \quad (4.38)$$

hold, then the inequality (4.33) holds.

We proceed to prove (4.37). A simple calculation leads to  $\frac{\gamma}{k_{D_s}} \geq 1.21$  and  $\frac{q_w}{\lambda_w} \leq \rho_w^2$  which verify (4.37). For the inequality (4.38) is satisfied, we need to verify that

$$|k_{P_s}v + k_{D_s}(-k_{P_s}\xi - k_{D_s}v)| \leq \beta_{21} \left( \sqrt{|\xi|^2 + \rho^2|v|^2} \right) \quad (4.39)$$

and

$$\left| \frac{q_w}{\lambda_w} (X + k_{P_s}\xi + k_{D_s}v) + Y \right| \leq \beta_{22} \left( \sqrt{|X + k_{P_s}\xi + k_{D_s}v|^2 + \rho_w^2|Y|^2} \right) \quad (4.40)$$

hold, where  $\beta_2 = \beta_{21}\beta_{22}$ . Inequality (4.39) verifies with  $\beta_{21} \geq (k_{P_s}k_{D_s})(k_{P_s} + k_{D_s}^2)$ , and (4.40) verifies with  $\beta_{22} \geq \frac{q_w}{\lambda_w}$ .

We proceed to develop (4.34) as

$$\left| \frac{q}{\lambda}v + \xi \right| |X + k_{P_s}\xi + k_{D_s}v| \leq \beta_1 \left( \sqrt{|\xi|^2 + \rho^2|v|^2} \right) \left( \sqrt{|X + k_{P_s}\xi + k_{D_s}v|^2 + \rho_w^2|Y|^2} \right)$$

which can be broken up into

$$\left| \frac{q}{\lambda}v + \xi \right| \leq \beta_{11} \left( \sqrt{|\xi|^2 + \rho^2|v|^2} \right) \quad (4.41)$$

and

$$|X + k_{P_s}\xi + k_{D_s}v| \leq \beta_{12} \left( \sqrt{|X + k_{P_s}\xi + k_{D_s}v|^2 + \rho_w^2|Y|^2} \right) \quad (4.42)$$

where  $\beta_1 = \beta_{11}\beta_{12}$ . Inequality (4.41) is satisfied with  $\frac{q}{\lambda} \leq \rho^2$  and  $\beta_{11} \geq 2$ . Inequality (4.42) verifies with  $\beta_{12} \geq 1$ .

In order to prove stability of the system (4.9)-(4.12) together with controller (4.14) composed by (4.18) and (4.25), we choose a candidate Lyapunov function given by

$$\nu(\xi, v, X, Y) = (1 - d)V(\xi, v) + dW(X, Y) \quad (4.43)$$

where  $d \in (0, 1)$ . The derivative of (4.43) along the closed-loop system (4.9)-(4.12)-(4.14) is given by

$$\dot{\nu} \leq (1 - d) \frac{\partial V}{\partial x} f(x, z) + \frac{d}{\varepsilon} \frac{\partial W}{\partial z} g(x, z) + d \frac{\partial W}{\partial x} f(x, z) \quad (4.44)$$

such Lyapunov function (4.44) can be represented as

$$\begin{aligned} \dot{v} \leq & (1-d) \frac{\partial V}{\partial x} f(x, h(x)) + (1-d) \frac{\partial V}{\partial x} [f(x, z) - f(x, h(x))] \\ & + \frac{d}{\varepsilon} \frac{\partial W}{\partial z} g(x, z) + d \frac{\partial W}{\partial x} f(x, z) \end{aligned} \quad (4.45)$$

with inequalities (4.21), (4.28) and (4.33), (4.34) we get

$$\begin{aligned} \dot{v} \leq & -(1-d)\alpha_1\psi^2(x) + (1-d)\beta_1\psi(x)\phi(z-h(x)) - \frac{d}{\varepsilon}\alpha_2\phi^2(z-h(x)) \\ & + d\gamma\phi^2(z-h(x)) + d\beta_2\psi(x)\phi(z-h(x)) \\ \leq & -\Phi^T A \Phi \end{aligned} \quad (4.46)$$

where  $\Phi = [\psi(x) \quad \phi(z-h(x))]^T$  and  $A$  is given by

$$A = \begin{pmatrix} (1-d)\alpha_1 & \frac{1}{2}(1-d)\beta_1 - \frac{1}{2}d\beta_2 \\ \frac{1}{2}(1-d)\beta_1 - \frac{1}{2}d\beta_2 & d\left(\frac{\alpha_2}{\varepsilon} - \gamma\right) \end{pmatrix} \quad (4.47)$$

The quadratic form given in (4.46) is negative-definite when

$$\varepsilon < \frac{\alpha_1\alpha_2}{\alpha_1\gamma + \frac{((1-d)\beta_1 + d\beta_2)^2}{4d(1-d)}} := \varepsilon_d \quad (4.48)$$

Thus, the system (4.9)-(4.12) with the controller (4.14) is asymptotically stable  $\forall \varepsilon < \varepsilon_d$ .

#### 4.4.1 Real Input Controls

In order to express the control inputs  $\tilde{\tau}$  and  $\tilde{F}$ , let us consider the notation

$$\varepsilon \dot{Y} = Z \quad (4.49)$$

From (4.7) and (4.12) it follows that

$$\tilde{F} - S(F)\tilde{\tau} = mR^T(u + X + Y) - S(\Omega)F - mR^TY(S(\Omega) + 1) - mR^TX := \tilde{u} \quad (4.50)$$

where the auxiliary variable  $Z$  and the vector  $\tilde{u} \in \mathfrak{R}^3$  are functions of known signals. Thus, using (4.4), (4.6) and (4.50), the original inputs  $F$  and  $\tau$  can be recovered by simple calculations.

## 4.5 Simulations and Experimental Results

In this section we describe the numerical simulations and the experiments that demonstrate the effectiveness of the controller presented in section 4.4.

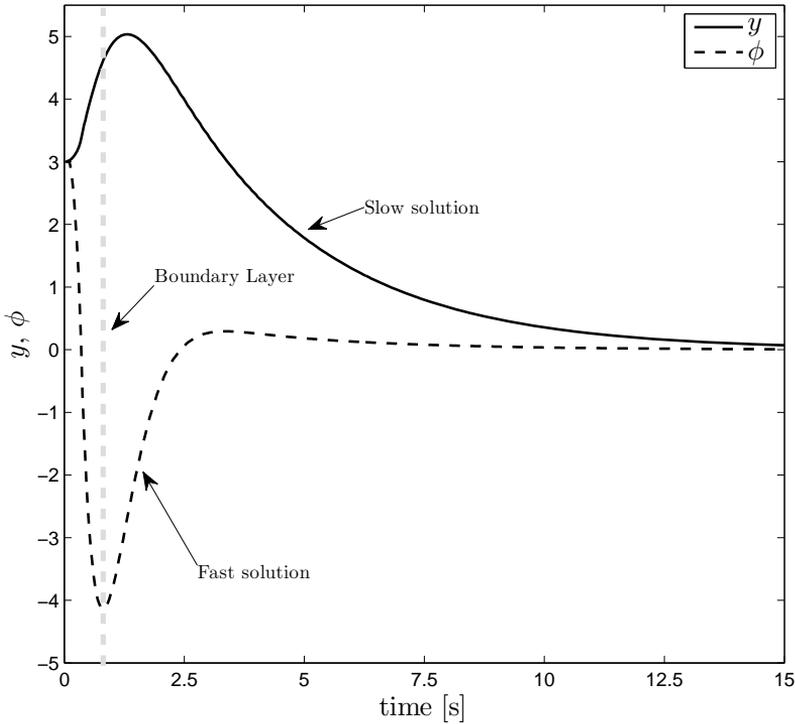
The entire fast system is comprised by the rotational dynamics composed by the states:  $(\theta, \dot{\theta}, \phi, \dot{\phi}, \psi, \dot{\psi})$ . The slow system is composed by the translational dynamics which is comprised by the states:  $(x, \dot{x}, y, \dot{y}, z, \dot{z})$ . For simplicity, we present simulations and experimental results only for two states representing the entire fast system: the roll dynamics, i.e.  $(\dot{\phi}, \phi)$  and for two states representing the slow system: the lateral position dynamics, i.e.  $(\dot{y}, y)$ .

### 4.5.1 Numerical Results

In order to emphasize the time-scale separation property, and to show the faster convergence on the fast dynamics, we chose the same initial conditions for all states as  $\phi(0) = \dot{\phi}(0) = y(0) = \dot{y}(0) = 3$ . We have simulated the proposed controller for three different values of  $\varepsilon$ :  $\varepsilon = 0.5$ ,  $\varepsilon = 0.25$  and  $\varepsilon = 0.1$ . In Fig. 4.2, the first row shows results for  $\varepsilon = 0.5$ , the second row for  $\varepsilon = 0.25$  and the third row for  $\varepsilon = 0.1$ . The simulation have been performed with the same control parameters.

The case when  $\varepsilon = 0.5$  is depicted in Fig. 4.1, and its corresponding controller in Fig. 4.3. The region of rapid response named boundary layer, exists near the initial point and is shown in Fig. 4.1. The state  $\phi$  converges faster than the state  $y$ , and its convergence depends on the the  $\varepsilon$  value. From the simulations shown in Fig. 4.2, the rotational dynamics converge faster than the translational dynamics, according to the parameter  $\varepsilon$ . In addition, the boundary layer or region of rapid transition occurs near to  $t = 0$ .

The fast and slow control, (4.25) and (4.18) respectively, are shown in Fig. 4.3. Where the fast control presents a faster response regarding the slow control. Such controller corresponds to  $\varepsilon = 0.5$ .

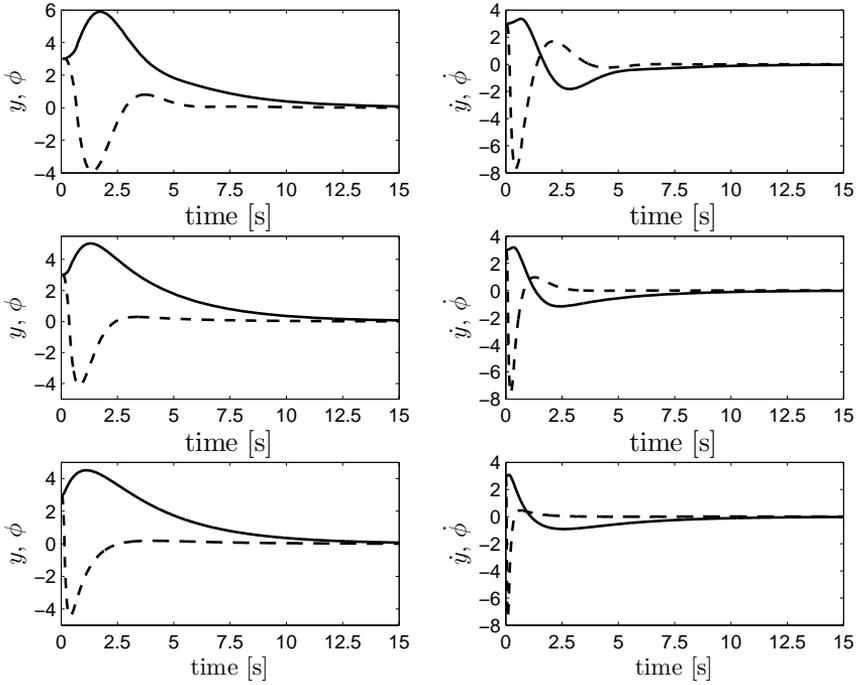


**Figure 4.1:** The state  $\phi$  (the fast dynamics), converges faster than the state  $y$  (the slow dynamics). The boundary layer exists near the initial condition.

#### 4.5.2 Experimental Results

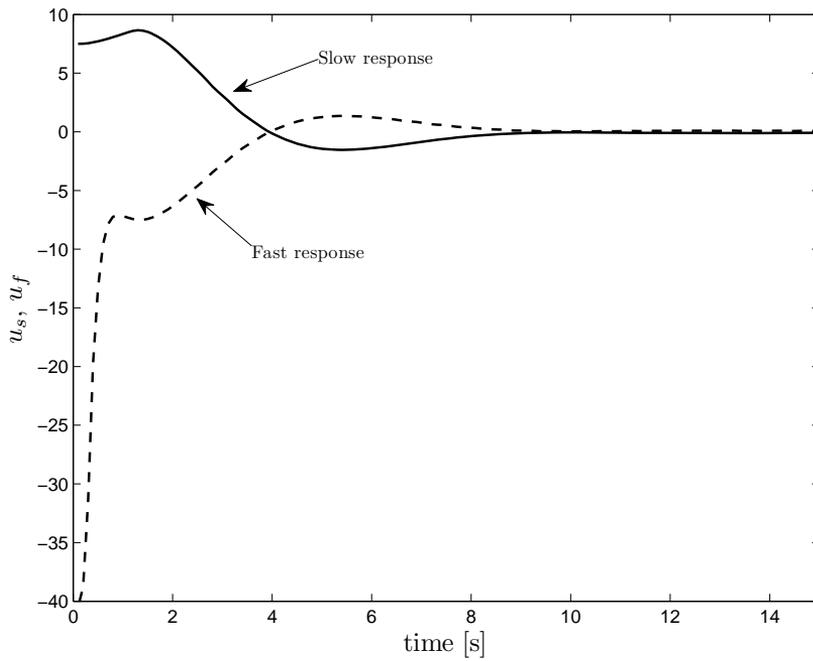
The proposed controller has been tested on the *Quad-Plane* experimental platform illustrated in Fig. 4.4. With the Quad-plane at  $y = 0$ ,  $\dot{y} = 0$ , the desired position is set to  $y = 0$ , during 40 seconds.

The experiment consist in an autonomous take-off, then the MAV goes forward on  $x$  direction following a line on the floor using its embedded visual system, while maintaining a  $y$  relative position equal to zero w.r.t. the estimated line. Thus, the desired position and orientation in the roll dynamics and  $y$ -position is zero. Fig. 4.5 illustrates the behavior of the proposed controller, in which the vehicle is stabilized on the desired position. The Quad plane is disturbed on the roll angle at  $t = 32$  sec. as it is shown in Fig. 4.5. The system has been disturbed in order to show the rapid response on the fast dynamics represented



**Figure 4.2:** The states corresponding to the fast dynamics ( $(\dot{\phi}, \phi)$ : dashed line) converge faster than the states corresponding to the slow dynamics ( $(\dot{y}, y)$ : solid line). Depending on the value of  $\varepsilon$ , the convergence is faster. In addition, the size of the boundary layer, shown in Fig. 4.1, is proportional to  $\varepsilon$ .

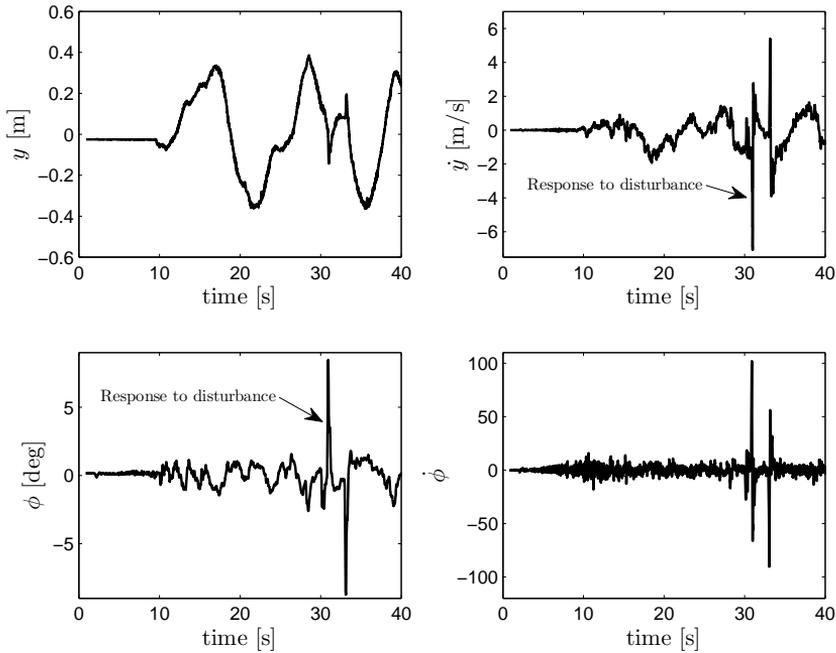
by the pitch dynamics  $(\phi, \dot{\phi})$  in comparison with the slow dynamics represented by the the position dynamics  $(y, \dot{y})$ . As we see in Fig. 4.5, the performance of the controller is satisfactory.



**Figure 4.3:** Slow control  $u_s$  is represented by the solid line while the fast control  $u_f$  is represented by the dashed line. The time scale separation property is also presented on both controllers.



**Figure 4.4:** Quad-plane experimental platform, developed at HEUDIASYC laboratory.



**Figure 4.5:** Experimental results showing the effectiveness of the controller. In the first row the translational dynamics is shown while in the second row the rotational dynamics is illustrated. Beyond the inherent noise, the response in translational dynamics behaves slower than the rotational dynamics. We have disturbed the vehicle on the roll angle at  $t = 32$ .

# 5

## FAULT ESTIMATION AND CONTROL FOR A QUAD-ROTOR MAV USING A POLYNOMIAL OBSERVER

---

*This chapter addresses the problem of fault detection and diagnosis (FDD) for a quad-rotor mini aerial vehicle (MAV). Actuator faults are considered on this chapter. The basic idea behind the proposed method is to estimate the faults signals using the extended state observers theory. To estimate the faults, a polynomial observer is presented by using the available measurements and know inputs of the system. In order to investigate the observability and diagnosability properties of the system, a differential algebra approach is proposed. Furthermore, an evaluation function depending on the system states is developed, in order to be used in a controller, which will compensate the failures. The effectiveness of the methodology is illustrated by means of numerical simulations and some experimental tests.*

<b>5.1</b>	<b>Introduction</b>	<b>65</b>
<b>5.2</b>	<b>Fault Diagnosis Problem</b>	<b>66</b>
5.2.1	Observability and Diagnosability Condition	67
<b>5.3</b>	<b>Polynomial Observer</b>	<b>68</b>
5.3.1	Observer design	68
5.3.2	Convergence Analysis	69
<b>5.4</b>	<b>Application to Quad-rotor MAV</b>	<b>72</b>
5.4.1	Modeling	72
5.4.2	Control Strategy	73
5.4.3	Diagnosability analysis	74
5.4.4	Polynomial Observer	75
<b>5.5</b>	<b>Evaluation Function</b>	<b>76</b>
5.5.1	Fault detection	76
5.5.2	Experimental results by using the evaluation function	76
<b>5.6</b>	<b>Fault reconstruction results</b>	<b>78</b>

---

---

## 5.1 Introduction

---

The growing development in research on MAVs and the consequent improvement of technologies like microcomputers, vision systems and other sensor devices, have increased the performance requirements of such kind of systems. Problems related to trajectory tracking, flight-formation, vision-based localization and lately MAV equipped with manipulators, have been widely researched in the last few years. Therefore, a good performance in the inner-loop of such flight envelopes is needed. A wide range of nonlinear control techniques like backstepping [6], [31], singular perturbation techniques [44], sliding modes and switching control [39], [34], have been treated to deal with the complex dynamics of the quad-rotor. Due to the high cost of the MAV equipment, it is imperative to provide such systems with a fault-control loop, responsible for the identification of possible faults presented at any time of the flight envelope.

Motivated by the *fault diagnosis problem*, which is the problem of observing fault signals, and the necessity of developing sufficiently robust controllers to cope the presence of likely faults, this research work deals not only with the MAV stabilization problem, but also with the identification of actuator faults. Few works dealing with the fault diagnosis problem applied on quad-rotors are presented in the literature [45], [46], [47], [48], [49].

Taking the attitude, position, angular and translational velocities of the quad-rotor MAV as available measurements, we develop a solution for the fault diagnosis problem by means of the differential algebraic approach. With this approach, it is possible to construct a bank of observers in order to implement a scheme of residual generation for fault diagnosis [50], or implement a control law based on state estimation [51]. Thus, it is possible to combine different schemes of nonlinear observers. In [52], the authors present a reduced order and a sliding mode observer, to reconstruct faults in an experimental task, for the case when only one output is available. A reduced order observer and an algebraic observer is presented in [53]. The approach given in [54] is used for fault detection and fault estimation of a wound-rotor induction motor (WRIM). In [55] a polynomial observer, a reduced order observer and a sliding mode observer are used in order to estimate and reconstruct the system states and faults for the case of multiple available outputs. In [56], the polynomial observer is used for the synchronization of chaotic systems.

The chapter is organized as follows. The fault diagnosis problem is formulated in Section 5.2. In order to estimate not only the system states but also the faults dynamics, an extended Luenberger observer called polynomial observer is

developed in Section 5.3. Next, in Section 5.4, the results previously obtained is applied to the Quad-rotor MAV. Section 5.6 presents some simulation results for the fault reconstruction problem.

## 5.2 Fault Diagnosis Problem

---

The Fault Detection and Diagnosis (FDD) task has the goal of detecting the presence of a fault and construct an estimate of the unknown fault dynamics. Such faults can affect directly the performance of the system components. Therefore, a FDD scheme provides all the necessary information about faults, such as presence (time), type (actuator/sensor) and dynamics (magnitude and form). Thus, based on this information, it is possible to design a system reconfiguration to minimize the fault effects. We begin by defining the terms *fault* and *failure* as follows:

**Fault:** An undesired change in a system parameter or variable that reduces the performance/magnitude of one component of its nominal value. In summary, a fault is an *unacceptable tolerable malfunction*.

**Failure:** A complete breakdown of the system, caused by a catastrophic malfunction of one or more components of the system. In summary, a failure is an *intolerable malfunction*.

Throughout this chapter, we describe a class of nonlinear systems with faults as follows

$$\begin{aligned} \dot{x}(t) &= g(x, u, f) \\ y(t) &= h(x, u) \end{aligned} \tag{5.1}$$

where

$x \in \mathbb{R}^n$  is the state vector

$u \in \mathbb{R}^m$  is the vector of known inputs

$f \in \mathbb{R}^\mu$  is the faults vector (unknown inputs)

$y \in \mathbb{R}^p$  is the outputs vector

In this chapter, we consider only the case of faults in the actuators. So, we introduce the concept of observability and diagnosability in the field of the differential algebra.

### 5.2.1 Observability and Diagnosability Condition

The observability and diagnosability notion of a system, linear or nonlinear in the differential algebra approach need a basic definition. Further details can be found in [52].

**Definition 1.** For the system described by (5.1) a state  $x$  is said to be observable if it is possible to estimate the state by means of the available measurements of the system, so we say that  $x$  is observable if it is algebraically observable, i.e., the state  $x$  satisfies a polynomial equation in terms of  $u$  and  $y$  and some of their time derivatives:

$$P\left(x, y, \dot{y}, \ddot{y}, \dots, y^{(n)}, u, \dot{u}, \ddot{u}, \dots, u^{(n)}\right) = 0 \quad (5.2)$$

**Definition 2.** A fault  $f$  is said to be diagnosable if it is possible to estimate the fault from the available measurements of the system, i.e.,  $f$  is diagnosable if it is algebraically observable and it satisfies a polynomial equation in terms of  $u$  and  $y$  and some of their time derivatives:

$$P\left(f, y, \dot{y}, \ddot{y}, \dots, y^{(n)}, u, \dot{u}, \ddot{u}, \dots, u^{(n)}\right) = 0 \quad (5.3)$$

**Remark 2.** The diagnosability condition is independent of the observability of the system.

Referring to system (5.1), the vector  $f$  contains the unknown inputs. In order to estimate its uncertain dynamics, the state vector is extended to deal with the fault vector. So, we can rewrite the system in an extended form as follows

$$\begin{aligned} \dot{x}(t) &= g(x, u, f) \\ \dot{f}_k(t) &= \Omega_k(x, u, f) \quad 1 \leq k \leq \mu \\ y(t) &= h(x, u) \end{aligned} \quad (5.4)$$

The following results from the theory of differential algebra are a useful tool to determine whether a fault can be reconstructed from the known inputs and available outputs.

**Theorem 3.** Assume that the system (5.1) is diagnosable, then the number of faults is less or equal to the number of available measurements (outputs), i.e.

$$\mu \leq p$$

The proof of Theorem 3 can be seen in [57].

### 5.3 Polynomial Observer

---

The polynomial observer, is a scheme that combines two kinds of observers. The first one is like an extended Luenberger observer which is used to reconstruct and estimate the system states. While the second one is a free mode observer, which has the function of reconstruct an estimate of the faults dynamics. The polynomial observer can be seen as a Taylor series, where the first-order term is the observed state, thus improving the performance and speed of convergence including terms of high-order correction in the structure. It is worth mentioning that this scheme is considered for the case of multiple outputs available, where the terms of higher order correction are odd powers and are a linear combination of the observation errors of each output available and the order of the polynomial compensations is a determining factor for the parameter  $q$  which will be defined in this section.

Consider the system with presence of faults, given in (5.4), the observation problem for the unknown vector of faults can be estimated using a polynomial observer. Therefore the system (5.4) can be rewritten as

$$\begin{aligned}\dot{\hat{x}}(t) &= Ax + \Psi(x, \bar{u}) \\ \dot{\hat{f}}_k(t) &= \Omega_k(x, \bar{u}) \quad 1 \leq k \leq \mu \\ y(t) &= Cx\end{aligned}\tag{5.5}$$

where  $\|\Omega_k(x, \bar{u})\| \leq N$ ,  $N \in \mathbb{R}^+$  and  $\Psi(x, \bar{u})$  is a nonlinear function that satisfies the Lipschitz condition, with  $\bar{u} = (u, f)$  uniformly bounded.

$$\|\Psi(x, \bar{u}) - \Psi(\hat{x}, \bar{u})\| \leq L\|x - \hat{x}\|\tag{5.6}$$

#### 5.3.1 Observer design

---

Now, consider the system with faults (5.5), the following lemma describes the construction of the polynomial observer.

**Lemma 4.** *Let the system (5.5) be algebraically diagnosable, then, the following nonlinear system is a full order state observer for the given system*

$$\begin{aligned}\dot{\hat{x}}(t) &= A\hat{x} + \Psi(\hat{x}, u, \hat{f}) + \sum_{i=1}^p \sum_{j=1}^q K_{ij}(y_i - C_i\hat{x})^{2j-1} \\ \dot{\hat{f}}_k(t) &= \sum_{l=1}^q \bar{K}_{kl}(f_k - \hat{f}_k)^{2l-1}\end{aligned}\tag{5.7}$$

where

$$A \in \mathbb{R}^{n \times n}$$

$\hat{x} \in \mathbb{R}^{n \times 1}$  is the estimate of the state  $x$

$\hat{f}_k \in \mathbb{R}^\mu$  is the estimate of faults vector  $f$

$$q \in \mathbb{R}^+$$

$$\Psi(\hat{x}, u, \hat{k}) \in \mathbb{R}^{n \times 1}$$

$[K_{ij}]_{\substack{1 \leq i \leq p \\ 1 \leq j \leq q}}, [\bar{K}_{kl}]_{\substack{1 \leq k \leq \mu \\ 1 \leq l \leq q}}$  are positive gains

where  $\hat{x}_0 = \hat{x}(t_0)$  and  $\hat{f}_{k0} = \hat{f}_k(t_0)$  are arbitrary initial conditions, the parameter  $q$  determines the order of the polynomial compensation. To ensure the observer convergence, the following assumptions are considered:

- **A1:**  $f_k(t)$  is algebraically observable.
- **A2:** The gains  $[K_{il}]_{1 \leq i \leq p}$  can be chosen such that the following algebraic Riccati equation has a symmetric and positive definite solution  $P$  for some  $\varepsilon > 0$

$$(A - \sum_{i=1}^p K_{i1} C_i)^T P + P(A - \sum_{i=1}^p K_{i1} C_i) + L^2 P P + I + \varepsilon I = 0$$

- **A3:** The gains  $[K_{ij}]_{\substack{1 \leq i \leq p \\ 2 \leq j \leq q}}$  are chosen such that

$$\lambda_{\min}((PK_{ij} C_i)^T + (PK_{ij} C_i)) \geq 0$$

We define the estimation error vector as  $e = [e_x, e_k]^T$ , whit  $e_x = x - \hat{x}$  and  $e_k = f_k - \hat{f}_k$ . So from the systems (5.5) and (5.7), we determine the dynamics for the corresponding error estimation

$$\begin{aligned} \dot{e}_x &= (A - \sum_{i=1}^p K_{i1} C_i) e_x - \sum_{i=1}^p \sum_{j=2}^q K_{ij} (C_i e_x)^{2j-1} \\ &\quad + [\Psi(x, \bar{u}) - \Psi(\hat{x}, \bar{u})] \end{aligned} \quad (5.8)$$

$$\dot{e}_k = \Omega_k - \bar{K}_{k1} e_k - \sum_{j=2}^q \bar{K}_{kj} (e_k)^{2j-1} \quad (5.9)$$

### 5.3.2 Convergence Analysis

In order to ensure the convergence to zero of the estimation error, we establish the following theorem.

**Theorem 5.** *For the system (5.5), suppose that  $x(t) \exists \forall t \geq 0$ , the function  $\Psi(x, \bar{u})$  satisfies the Lipschitz condition given in (5.6), and  $x(t), f(t)$  are algebraically observable. Thus, if there exists a positive definite matrix  $P$  and positive observer gains  $K_{ij}, \bar{K}_{kl}$  such that the system (5.7) is an observer for system (5.5), then the estimation error converges to zero asymptotically.*

*Proof.* Consider the following Lyapunov function candidate

$$\begin{aligned} V &= V_1 + V_2 \\ V_1 &= e_x^T P e_x; V_2 = \frac{1}{2} e_k^2 \end{aligned} \quad (5.10)$$

where the matrix  $P$  satisfies the assumption A2.

The proof of theorem 5, is developed in two parts as follows:

i) The time derivative of  $V_1$  is given as

$$\begin{aligned} \dot{V}_1 &= \dot{e}_x^T P e_x + e_x^T P \dot{e}_x \\ &= e_x^T \left( (A - \sum_{i=1}^p K_{i1} C_i)^T P + P (A - \sum_{i=1}^p K_{i1} C_i) \right) e_x \\ &\quad + 2e_x^T P [\Psi(x, \bar{u}) - \Psi(\hat{x}, \bar{u})] \\ &\quad - 2 \sum_{i=1}^p \sum_{j=2}^q K_{ij} (C_i e_x)^{2j-2} e_x^T ((PK_{ij} C_i)^T + (PK_{ij} C_i)) e_x \end{aligned}$$

From the follow inequality based on the Lipschitz condition

$$2e_x^T P [\Psi(x, \bar{u}) - \Psi(\hat{x}, \bar{u})] \leq L^2 e_x^T P P e_x + e_x^T e_x \quad (5.11)$$

and using the Rayleigh's inequality together with assumption A3. it follows that

$$-e_x^T P K_{ij} C_i e_x \leq -\lambda_{\min}(PK_{ij} C_i + (PK_{ij} C_i)^T) \|e_x\|^2 \quad (5.12)$$

Therefore, applying inequalities (5.11) and (5.12) we have

$$\begin{aligned} \dot{V}_1 &\leq e_x^T \left[ (A - \sum_{i=1}^p K_{i1} C_i)^T P + P (A - \sum_{i=1}^p K_{i1} C_i) + L^2 P P + I \right] e_x \\ &\quad - 2 \sum_{i=1}^p \sum_{j=2}^q K_{ij} (C_i e_x)^{2j-2} \lambda_{\min}(PK_{ij} C_i + (PK_{ij} C_i)^T) \|e_x\|^2 \\ &\leq e_x^T \left[ (A - \sum_{i=1}^p K_{i1} C_i)^T P + P (A - \sum_{i=1}^p K_{i1} C_i) + L^2 P P + I \right] e_x \\ &= -\varepsilon \|e_x\|^2 \end{aligned}$$

- ii) In the same way, for the second term in the Lyapunov function candidate, we obtain the time derivative of  $V_2$  as

$$\begin{aligned}
\dot{V}_2 &= e_k \dot{e}_k \\
&= e_k \left( \Omega_k - \bar{K}_{k1} e_k - \sum_{l=2}^q \bar{K}_{kl} e_k^{2l-1} \right) \\
&= e_k \Omega_k - \bar{K}_{k1} e_k^2 - \sum_{l=2}^q \bar{K}_{kl} e_k^{2l} \\
&\leq |e_k| |\Omega_k| - \bar{K}_{k1} e_k^2 \\
&\leq |e_k| N - \bar{K}_{k1} |e_k|^2 \\
&= -[\bar{K}_{k1} |e_k| - N] |e_k|
\end{aligned}$$

$\dot{V}_2$  is negative inside the set  $\{|e_k| > N/\bar{K}_{k1}\}$ , i.e., exists  $\bar{\varepsilon} > 0$  such that  $\bar{K}_{k1} |e_k| - N = \bar{\varepsilon} > 0$ .

We prove that  $|e_k|$  is upper bounded. Now let  $\alpha, \beta$  be the upper bounds of  $V_2(e_k)$ . With  $\beta > \frac{N^2}{2\bar{K}_{k1}^2}$ , the solution that initiates in the set  $\{V_2(e_k) \leq \beta\}$  will remain inside that set for all  $t \geq 0$ , because  $\dot{V}_2$  is negative in  $V_2 = \beta$ . Therefore the solution of  $\dot{e}_k$  is uniformly bounded [58]. Furthermore, if  $\frac{N^2}{2\bar{K}_{k1}^2} < \alpha < \beta$ , then  $\dot{V}_2$  will be negative in the set  $\{\alpha \leq V_2 \leq \beta\}$ . In this set  $V_2$  will decrease monotonously until the solution is in the set  $\{V_2 \leq \alpha\}$ . According to [58] the solution is uniformly ultimately bounded with ultimate bound  $|e_k| \leq \sqrt{2\alpha}$ . For example, if we define  $\alpha = \frac{N^2}{2\bar{K}_{k1}^2}$  and  $\beta = \frac{N^2}{\bar{K}_{k1}^2}$ , the ultimate bound is

$$|e_k| \leq \frac{N}{\bar{K}_{k1}}$$

hence

$$\dot{V}_2 \leq -\bar{\varepsilon} |e_k|.$$

Finally, from (i) and (ii), we conclude that

$$\dot{V} \leq -\varepsilon \|e_x\|^2 - \bar{\varepsilon} |e_k| < 0$$

□

## 5.4 Application to Quad-rotor MAV

---

In this section, the polynomial observer approach developed in Section 5.3 is applied to a Quad-rotor MAV. We will state the mathematical model of the MAV and some notations. Then, the diagnosability analysis of such vehicle dynamics is developed.

### 5.4.1 Modeling

---

The Quad-rotor mathematical model using the corresponding coordinate system shown in Figure (5.1) is given as follows

$$\begin{aligned}
 m\ddot{x} &= (u_1 + u_2 + u_3 + u_4)(s_\psi s_\phi + c_\psi s_\theta c_\phi) \\
 m\ddot{y} &= (u_1 + u_2 + u_3 + u_4)(s_\psi s_\theta c_\phi - c_\psi s_\phi) \\
 m\ddot{z} &= mg - (u_1 + u_2 + u_3 + u_4)(c_\phi c_\theta) \\
 \ddot{\theta} &= u_3 - u_1 \\
 \ddot{\phi} &= u_2 - u_4 \\
 \ddot{\psi} &= u_1 - u_2 + u_3 - u_4
 \end{aligned} \tag{5.13}$$

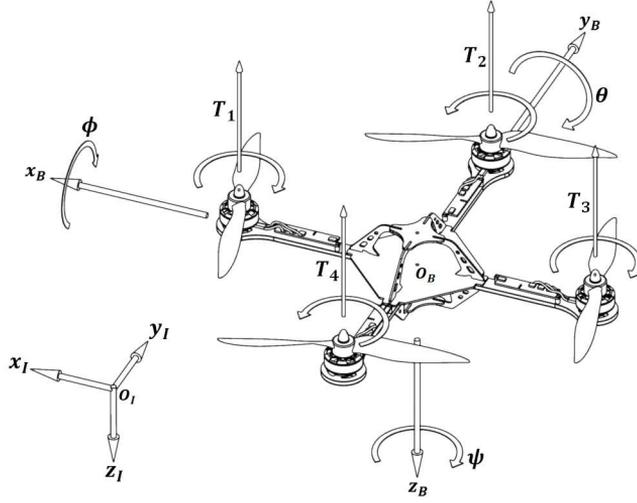
where the notation  $c_a$  and  $s_a$  stands for  $\cos(a)$  and  $\sin(a)$  respectively;  $u_i$  are the control inputs,  $(x, y, z)$  the position states and  $(\theta, \phi, \psi)$  the orientation angles. Using the model (5.13), the following changes of coordinates is made in order to formulate the problem

$$\begin{array}{llllll}
 x_1 = x & x_3 = y & x_5 = z & x_7 = \theta & x_9 = \phi & x_{11} = \psi \\
 x_2 = \dot{x} & x_4 = \dot{y} & x_6 = \dot{z} & x_8 = \dot{\theta} & x_{10} = \dot{\phi} & x_{12} = \dot{\psi}
 \end{array}$$

With this notation, the state vector is given by  $x = [x_1, x_2, \dots, x_{12}]^T$  and the input vector as  $u = [u_1, u_2, u_3, u_4]$ . For a Quad-rotor MAV, we consider a fault as a reduction of the performance of one or two actuator, with the constraint that if the failure affects an actuator, the second failure will affect the opposite actuator by the symmetry of the structure, i.e. the failure effects can only be minimized when these occurs in the following form:

1. The fault effects only one actuator
2. The fault affects the pairs  $(u_1, u_3)$  or  $(u_2, u_4)$ .

Otherwise we would have a catastrophic malfunction and it would be impossible to minimize the fault effects.



**Figure 5.1:** The three-dimensional quad-rotor model.

We now consider the presence of a fault on the engines, so we define the input with together with the fault as  $\bar{u}_k = u_k + f_k$ . Thus, the resulting system is given by:

$$\begin{aligned}
 \dot{x}_1 &= x_2 \\
 \dot{x}_2 &= \frac{1}{m}(\bar{u}_1 + \bar{u}_2 + \bar{u}_3 + \bar{u}_4)(s_{x_{11}}s_{x_9} + c_{x_{11}}s_{x_7}c_{x_9}) \\
 \dot{x}_3 &= x_4 \\
 \dot{x}_4 &= \frac{1}{m}(\bar{u}_1 + \bar{u}_2 + \bar{u}_3 + \bar{u}_4)(s_{x_{11}}s_{x_7}c_{x_9} - c_{x_{11}}s_{x_9}) \\
 \dot{x}_5 &= x_6 \\
 \dot{x}_6 &= g - \frac{1}{m}(\bar{u}_1 + \bar{u}_2 + \bar{u}_3 + \bar{u}_4)(c_{x_9}c_{x_7}) \\
 \dot{x}_7 &= x_8 \\
 \dot{x}_8 &= \bar{u}_3 - \bar{u}_1 \\
 \dot{x}_9 &= x_{10} \\
 \dot{x}_{10} &= \bar{u}_2 - \bar{u}_4 \\
 \dot{x}_{11} &= x_{12} \\
 \dot{x}_{12} &= \bar{u}_1 - \bar{u}_2 + \bar{u}_3 - \bar{u}_4
 \end{aligned} \tag{5.14}$$

where  $f_k$  are additive faults which affect directly the performance of the engines that produce the thrust inputs  $u_k$ .

### 5.4.2 Control Strategy

The proposed control strategy is based on the idea that the global system (5.13) is constituted of two subsystems, the attitude dynamics and the position

dynamics, each one with a time-scale separation between them [43]. From this fact, it is possible to propose a hierarchical control scheme where the position controller provides desired attitude angles  $\phi_d, \theta_d$  which are the angles to be tracked by the orientation controllers. We have implemented a nonlinear control strategy based on this principle. The interested lector can see more details about the control strategy applied on this chapter in [44].

### 5.4.3 Diagnosability analysis

---

From Theorem 3 it is required that the number of faults ( $\mu = 4$ ) be less or equal to available measurements. For this case, we consider the output vector as  $y = [y_1, y_3, y_5, y_7, y_9, y_{11}] = [x_1, x_3, x_5, x_7, x_9, x_{11}]$ . Taking into account the above mentioned considerations, the condition from theorem 3 is hold with  $4 = \mu < p = 6$ . To determine the diagnosability of the system (5.14), we evaluate the algebraic diagnosability condition given in definition 2. For the considered faults, inputs and outputs, the system (5.14) results in

$$\begin{aligned}
 \ddot{y}_2 &= \frac{1}{\eta} (\bar{u}_1 + \bar{u}_2 + \bar{u}_3 + \bar{u}_4) (s_{y_{11}} s_{y_9} + c_{y_{11}} s_{y_7} c_{y_9}) \\
 \ddot{y}_3 &= \frac{1}{m} (\bar{u}_1 + \bar{u}_2 + \bar{u}_3 + \bar{u}_4) (s_{y_{11}} s_{y_7} c_{y_9} - c_{y_{11}} s_{y_9}) \\
 \ddot{y}_5 &= g - \frac{1}{m} (\bar{u}_1 + \bar{u}_2 + \bar{u}_3 + \bar{u}_4) (c_{y_9} c_{y_7}) \\
 \ddot{y}_7 &= \bar{u}_3 - \bar{u}_1 \\
 \ddot{y}_9 &= \bar{u}_2 - \bar{u}_4 \\
 \ddot{y}_{11} &= \bar{u}_1 - \bar{u}_2 + \bar{u}_3 - \bar{u}_4
 \end{aligned} \tag{5.15}$$

From system (5.15), we have that

$$\frac{m(g - \ddot{y}_6)}{c_{y_9} c_{y_7}} = u_1 + f_1 + u_2 + f_2 + u_3 + f_3 + u_4 + f_4 \tag{5.16a}$$

$$\ddot{y}_7 = u_3 + f_3 - u_1 - f_1 \tag{5.16b}$$

$$\ddot{y}_9 = u_2 + f_2 - u_4 - f_4 \tag{5.16c}$$

$$\ddot{y}_{11} = u_1 + f_1 - u_2 - f_2 + u_3 + f_3 - u_4 - f_4. \tag{5.16d}$$

Adding (5.16a) and (5.16d)

$$\frac{m(g - \ddot{y}_5)}{c_{y_9} c_{y_7}} + \ddot{y}_{11} = u_1 + f_1 + 2u_3 + 2f_3 \tag{5.17}$$

adding 2(5.16b) and (5.17)

$$f_3 = \frac{m(g - \ddot{y}_5)}{4c_{y_9} c_{y_7}} + \frac{1}{2}\ddot{y}_7 + \frac{1}{4}\ddot{y}_{11} - u_3 \tag{5.18}$$

replacing (5.18) into (5.16b)

$$f_1 = \frac{m(g - \ddot{y}_5)}{4c_{y_9}c_{y_7}} - \frac{1}{2}\ddot{y}_7 + \frac{1}{4}\ddot{y}_{11} - u_1 \quad (5.19)$$

adding, (5.16a) and (5.16c)

$$\frac{m(g - \ddot{y}_5)}{c_{y_9}c_{y_7}} + \ddot{y}_9 = u_1 + f_1 + 2u_2 + 2f_2 + u_3 + f_3 \quad (5.20)$$

replacing (5.19) and (5.18) into (5.20)

$$f_2 = \frac{m(g - \ddot{y}_5)}{4c_{y_9}c_{y_7}} + \frac{1}{2}\ddot{y}_9 + \frac{1}{4}\ddot{y}_{11} - u_2 \quad (5.21)$$

Finally, replacing (5.21) into (5.16c) it follows that

$$f_4 = \frac{m(g - \ddot{y}_5)}{4c_{y_9}c_{y_7}} - \frac{1}{2}\ddot{y}_9 + \frac{1}{4}\ddot{y}_{11} - u_4 \quad (5.22)$$

Therefore, from equations (5.18), (5.19), (5.21) and (5.22) we conclude that the system (5.14) is diagnosable, with the considered inputs and outputs.

#### 5.4.4 Polynomial Observer

The system (5.14), can be expressed in a similar way as in (5.5) with  $A \in \mathbb{R}^{12 \times 12}$ ; where the elements of the matrix are given as follows:  $a_{1,2} = a_{3,4} = a_{5,6} = a_{7,8} = a_{9,10} = a_{11,12} = 1$  and zero otherwise. The nonlinear function  $\Psi(x, \bar{u}) = [\psi_1, \psi_2, \dots, \psi_{12}]$  is given by

$$\begin{aligned} \psi_1 &= \psi_3 = \psi_5 = \psi_7 = \psi_9 = \psi_{11} = 0 \\ \psi_2 &= \frac{1}{m}(\bar{u}_1 + \bar{u}_2 + \bar{u}_3 + \bar{u}_4)(s_{x_{11}}s_{x_9} + c_{x_{11}}s_{x_7}c_{x_9}) \\ \psi_4 &= \frac{1}{m}(\bar{u}_1 + \bar{u}_2 + \bar{u}_3 + \bar{u}_4)(s_{x_{11}}s_{x_7}c_{x_9} - c_{x_{11}}s_{x_9}) \\ \psi_6 &= -\frac{1}{mg}(\bar{u}_1 + \bar{u}_2 + \bar{u}_3 + \bar{u}_4)(c_{x_{11}}c_{x_9}) \\ \psi_8 &= \bar{u}_3 - \bar{u}_1 \\ \psi_{10} &= \bar{u}_2 - \bar{u}_4 \\ \psi_{12} &= \bar{u}_1 - \bar{u}_2 + \bar{u}_3 - \bar{u}_4. \end{aligned}$$

So, the following system is a polynomial observer for the given system

$$\begin{aligned} \dot{\hat{x}}(t) &= A\hat{x} + \Psi(\hat{x}, u, \hat{f}) + \sum_{i=1}^6 \sum_{j=1}^3 K_{ij}(y_i - C_i\hat{x})^{2j-1} \\ \dot{\hat{f}}_k(t) &= \sum_{l=1}^3 \bar{K}_{kl}(f_k - \hat{f}_k)^{2l-1} \end{aligned} \quad (5.23)$$

where we fixed the parameter  $q = 3$  and the  $f_{1 \leq k \leq 4}$  are given by (5.19), (5.21), (5.18), (5.22).

## 5.5 Evaluation Function

---

In this section, an evaluation function is presented in order to use the estimate of the failure with the methodology developed in the previous section. Such evaluation function is presented in a way that could be used in a controller to compensate the failure. Without loss of generality, we present the evaluation function for the roll dynamics, but the same procedure could be developed for the pitch and yaw dynamics.

### 5.5.1 Fault detection

---

In order to detect the fault and at the same time differentiate it from perturbations inherent in the system, we define the evaluation function  $e_f$  as a function of the system states as follows

$$e_f = \frac{1}{(1 + e^{-c_b(\phi-b)})(1 + e^{-c_a(\dot{\phi}-a))}} \quad (5.24)$$

where parameters  $c_a$  and  $c_b$  are positive real numbers which define the fault slope for  $\phi$  and  $\dot{\phi}$  respectively. The parameters  $a$  and  $b$  represent the position and velocity in roll dynamics for which there will be a fault. Accordingly, they are the boundaries between a disturbance and a fault.

So, the basic idea is to estimate the disturbances and then use them in the evaluation function (5.24). In this way, one should predefine the parameter values  $a$ ,  $b$ . So, by means of an observation of the system dynamics, one knows the existence of a fault. This approach allows to determine the existence of a fault without any measurement of engine speed, eliminating the use of additional sensors on the platform. In addition, a control strategy could be implemented by using (5.24), due to such evaluation function is continuously differentiable and it depends on the system states  $(\phi, \dot{\phi})$ .

### 5.5.2 Experimental results by using the evaluation function

---

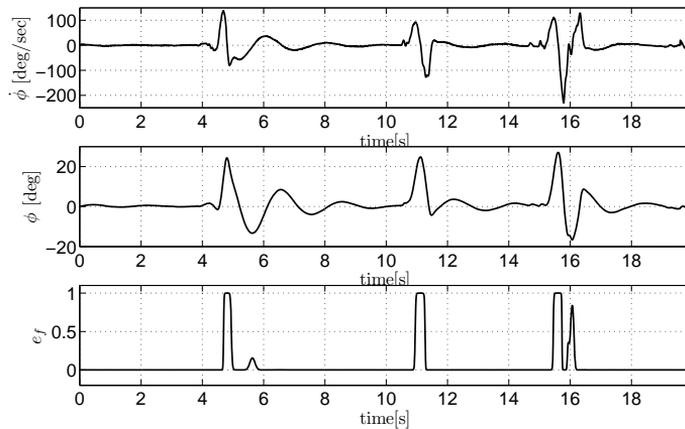
In this subsection, some experimental results are presented to visualize the performance of the evaluation function at real-time experiments. The experimental

**Table 5.1:** The parameters of the Quad-rotor MAV considered for the experimental tests.

Parameter	Value
$c_b$	1
$c_a$	1
$b$	20 deg
$a$	10 deg/sec

results have been tested on the Quad-rotor experimental platform developed at the HEUDIASYC Laboratory. More details about this platform can be seen in [36]. We have simulated the failures by disturbing the Quad-rotor platform in the roll dynamics. For these tests, we have used the parameters shown in table 5.1.

Two motors are involved in the pitch dynamics (Fig. 5.1). Thus, if one of this motors fails, its velocity will be reduced or augmented, causing an aggressive rolling moment due to the difference of thrust between the faulty motor and the operating motor. Therefore, we can evaluate such moment by inspecting the pitching dynamics, i.e. the behavior of the states  $(\theta, \dot{\theta})$ . In order to visualize the behavior of the evaluation function on the real-time experiments, we have perturbed the Quad-rotor platform on its roll axis. Fig. 5.2 shows the performance of the evaluation function (5.24). As we see in this figure, the disturbances lower than 20 deg have been omitted by the evaluation function, while the disturbances greater or equal to 20 deg have been taken as failures. With this approach, we can tune the parameters given in table 5.1, in order to choose a desired performance to tell the system how to distinguish a perturbation of a fault.

**Figure 5.2:** Evaluation function (5.24) and states of the roll dynamics.

## 5.6 Fault reconstruction results

In this section, we present some simulation results of the procedure developed in Section 5.3. The dynamics of the Quad-rotor MAV and the fault dynamics have been simulated using MATLAB Simulink. For the simulation procedure, we have consider the following conditions: The desired values for the position dynamics are  $x_d = y_d = 0\text{m}$  and  $z_d = 0.75\text{m}$  and for the attitude dynamics are  $\theta_d = \phi_d = 0$  and  $\psi_d = 45$  degrees. The objective is that the Quad-rotor take off and reaches the desired height and remain stable in that position, in other words, we want that the desired values for the linear and angular velocities are equal to zero. To simplify the calculations we assume that  $m = g = 1$ . A simulation time of 300s and a step of 0.001s has been chosen.

For all simulation results we have considered that the fault affects the performance of each engine, i.e. the actuators for the Quad-rotor MAV. Four faults were artificially generated as follows

$$\begin{aligned}
 f_1 &= 0.226(1 + \sin(0.5t \exp(-0.1t)))\mathcal{U}(t - 50) \\
 f_2 &= 0.045(1 + \sin(0.076 \exp(t - 0.3)))\mathcal{U}(t - 20) \\
 f_3 &= 0.055 \exp(-0.01(t - 0.3))\mathcal{U}(t - 10) \\
 &\quad + 0.068 \exp(-0.005(t - 1))\mathcal{U}(t - 80) \\
 &\quad + 0.159 \exp(-0.07(t - 1.3))\mathcal{U}(t - 140) \\
 f_4 &= 0.718 \exp(-0.01(t - 2))\mathcal{U}(t - 30)
 \end{aligned}$$

where  $\mathcal{U}(t)$  is the unit step function. The magnitude of the faults were selected very close to the magnitude of the generated thrusts inputs for the case without faults, to obtain better results.

The results of the implementation of the polynomial observer proposed in (5.23) for the fault diagnosis task, for the considered available outputs and inputs. As shown in the Figure 5.3, for the fault  $f_1$ . The gain values for the proposed observer were  $K_{11} = 2.5$ ,  $K_{12} = 34$  and  $K_{13} = 66$ . In the same way in figures 5.4, 5.5 and 5.6, we show the estimation result for the faults  $f_2$ ,  $f_3$  and  $f_4$ , where the gain values for each observer are  $K_{21} = 5.05$ ,  $K_{22} = 2$ ,  $K_{23} = 1.6$ ,  $K_{31} = 1.5$ ,  $K_{32} = 27$ ,  $K_{33} = 56$  and  $K_{41} = 4.23$ ,  $K_{42} = 7$ ,  $K_{43} = 3$  respectively.

In order to evaluate the effectiveness of the proposed polynomial observer, we use initial conditions different from zero, to see how long it takes to converge to the actual value of the fault. The initial conditions were  $f_{1,c,i} = 0.087$ ,  $f_{2,c,i} = 0.065$ ,  $f_{3,c,i} = 0.055$  and  $f_{4,c,i} = 0.073$ .

For all faults we obtained good estimation results. As can be seen, the proposed observer converge quickly to the actual values of the faults, and although the approach only considers the case of fault with differentiable dynamics, it is noted that the approach has the capacity to reconstruct abrupt faults as shown in figures 5.5 and 5.6.

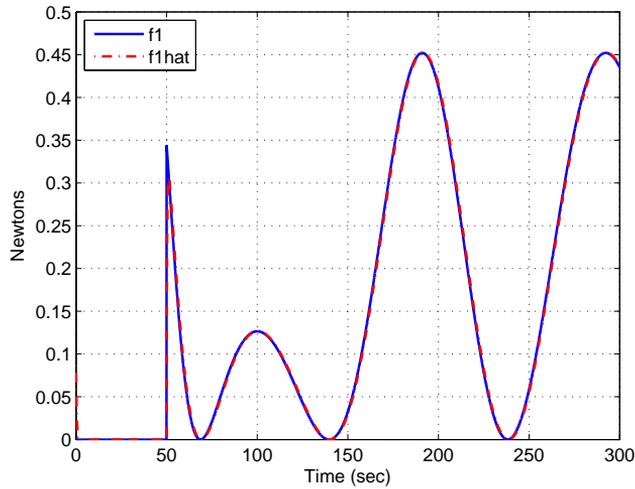


Figure 5.3: Estimation result for the fault  $f_1$

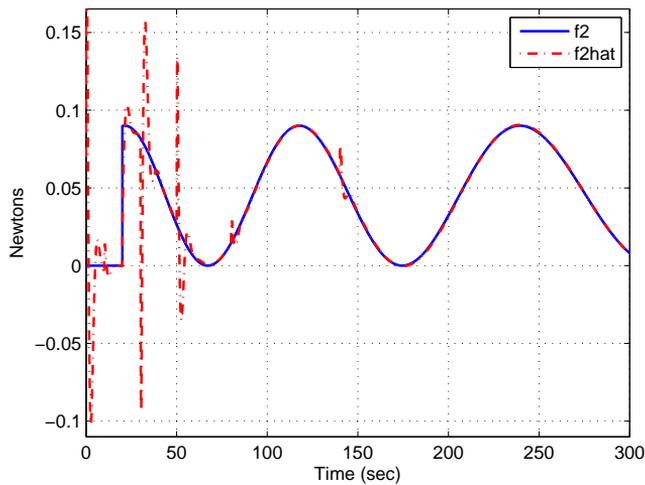


Figure 5.4: Estimation result for the fault  $f_2$

The attitude dynamic under the effect of the faults is shown in figure 5.7. The direct consequence of the presence of faults on actuators, is that, the controller tries to stabilize the system and bring the dynamics of roll, pitch and yaw to the desired values. However due to thrust limitation, the objective is not fully accomplished.

Figure 5.8, shows the corresponding angular velocities in roll, pitch and yaw for the case of presence of faults. As can be seen, between the first 100 seconds the angular velocities change abruptly, because in this time interval, all the

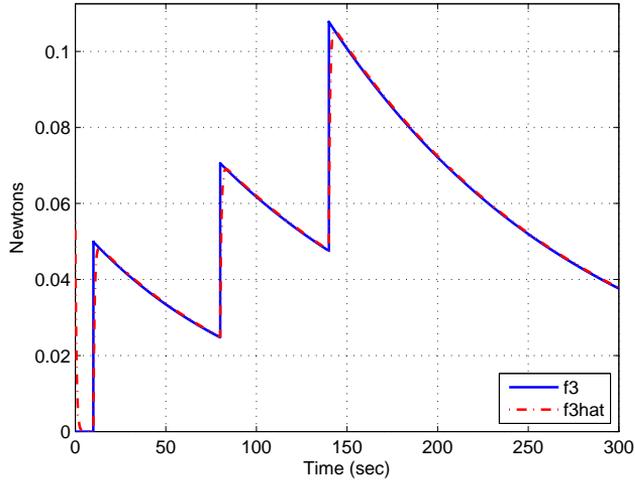


Figure 5.5: Estimation result for the fault  $f_3$

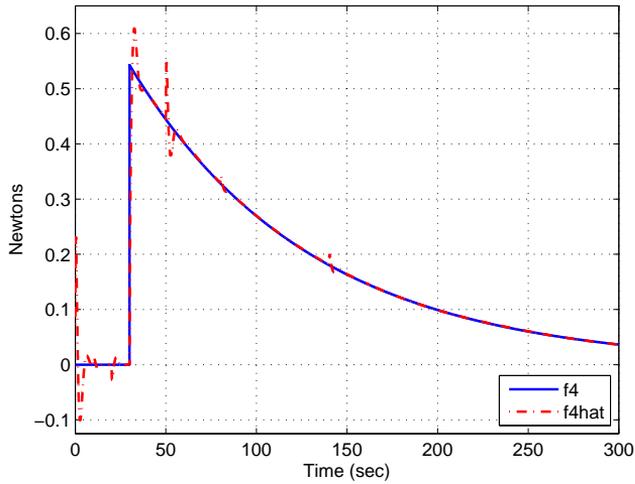
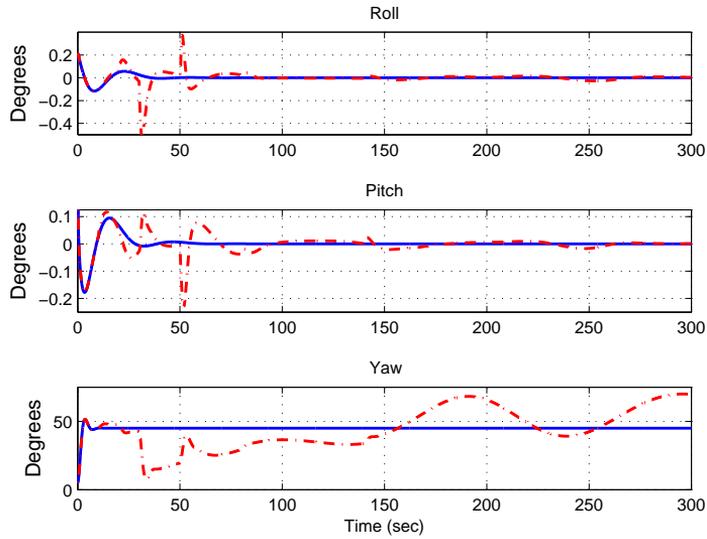


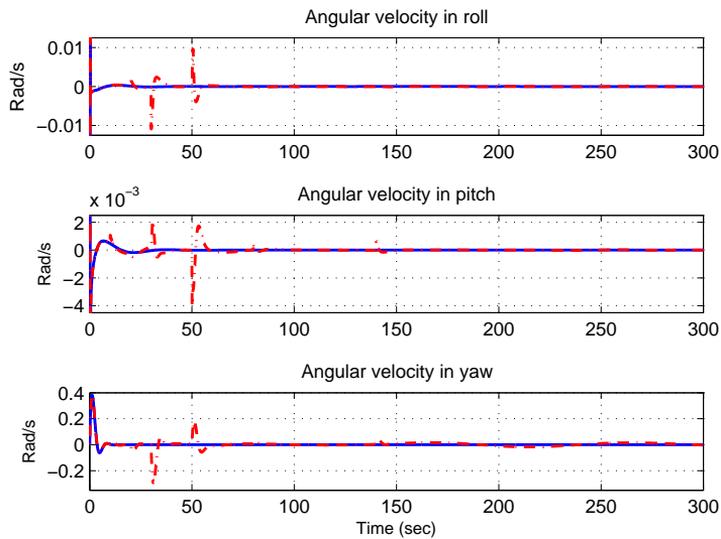
Figure 5.6: Estimation result for the fault  $f_4$

faults appear.

The position dynamics affected for the faults are shown in Figure 5.9. Note that the difference between the dynamic without faults and with fault is very significant, especially for the dynamics in the  $y$ -axis, which is very large. The Position controller generates large inputs to try reach the desired position values. However due to the faults the errors grow and the controller is unable to compensate such errors.

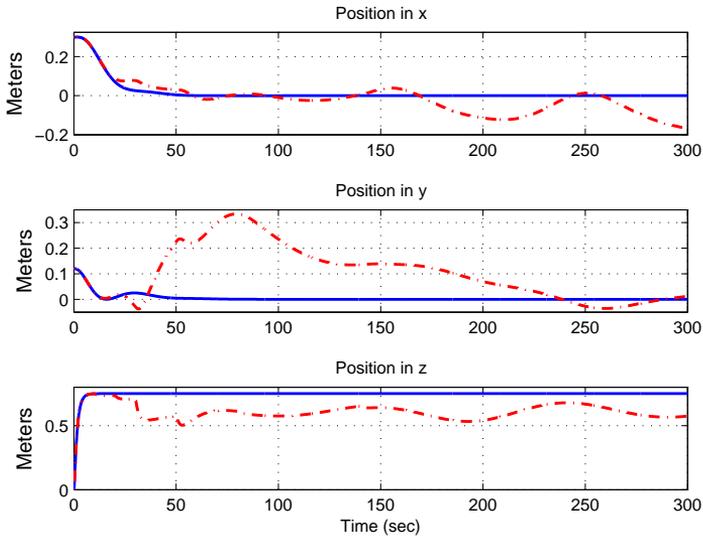


**Figure 5.7:** Comparison for the attitude dynamics for the case without faults (blue line) and with faults (red dash-dot line)

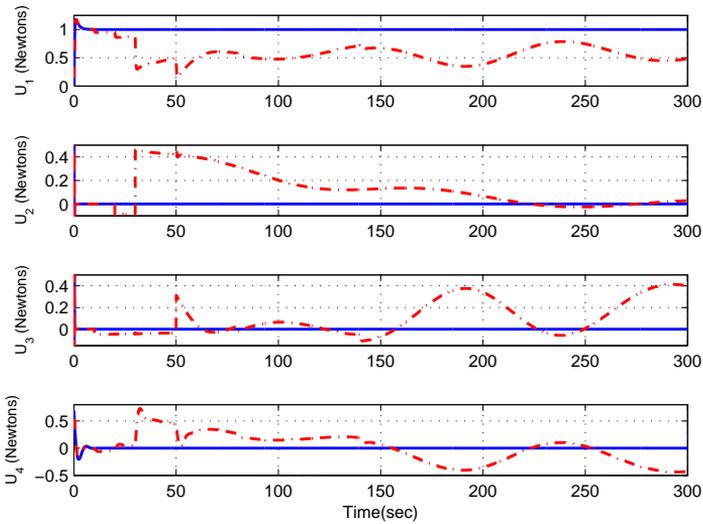


**Figure 5.8:** Comparison for the angular velocity dynamics for the case without faults (blue line) and with faults (red dash-dot line)

Finally figures 5.10 and 5.11 show the control inputs and corresponding thrusts generated by the control strategy. As can be seen in Figure 5.10 the difference

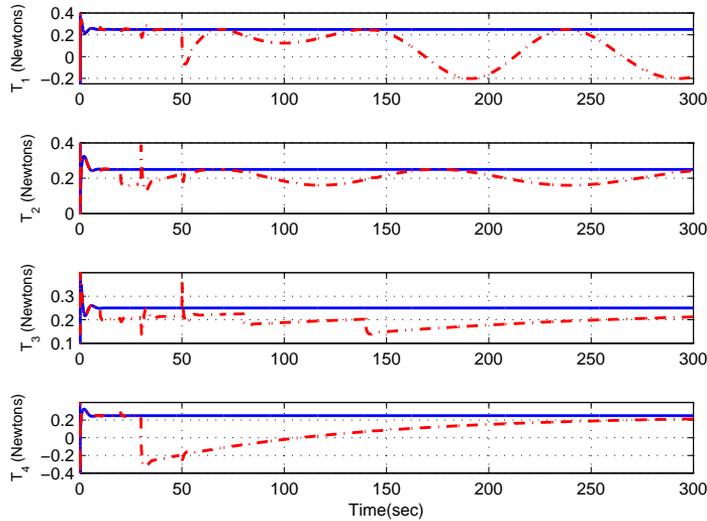


**Figure 5.9:** Comparison for the position dynamics for the case without faults (blue line) and with faults (red dash-dot line)



**Figure 5.10:** Comparison for the inputs generated by the control strategy for the case without faults (blue line) and with faults (red dash-dot line)

between both cases is very significant, because for the case without faults, the control inputs are constant while for the opposite case, the thrusts are



**Figure 5.11:** Comparison for the corresponding thrust inputs for the case without faults (blue line) and with faults (red dash-dot line)

nonconstant and larger. The controller tries to compensate the error generated by the faults. Figure 5.11 shows that the corresponding thrusts (inputs) are very similar to the dynamics of the faults with opposite sign. Notice that the faults cause the controller does not work properly, as we observed for the thrusts 1 and 4 in Figure 5.11 and the control inputs 2, 3 and 4 in Figure 5.10. There is an instant of time when it becomes negative which is infeasible, since it would mean that the thrust force is opposite, i.e., when the thrust force becomes negative, the engine does not have the ability to change the direction of rotation, and therefore stops completely (turned off). In the latter case we are not dealing with a fault, but with a failure.



# Part III

THE CONVERTIBLE  
AIRCRAFT



# 6

## CONTROL IN THE 6-DOF OF THE PVHAT AIRCRAFT. HOVER CONTROL

---

*This chapter presents a particular class of a convertible mini aerial vehicle (MAV) with fixed wings, the so-called PVHAT aircraft. This aircraft is able to change its flight configuration from hover to level flight and vice versa by means of a transition maneuver. In this first part of the research, the hover dynamics of the PVHAT aircraft is investigated. Dynamical model and nonlinear control based on Lyapunov design are studied. The presented approach focuses on the problem of finding a control law capable of stabilizing the aircraft's position. Some simulations results are given, which demonstrate the effectiveness of the controller. Further, some experimental results are presented and tested on the PVHAT aircraft experimental platform.*

### Contents

---

<b>6.1</b>	<b>Introduction</b>	<b>88</b>
<b>6.2</b>	<b>System Description</b>	<b>89</b>
6.2.1	Modeling of the PVHAT Aircraft	89
6.2.2	Dynamic Extension	91
<b>6.3</b>	<b>Lyapunov-Based Tracking Control Strategy</b>	<b>92</b>
<b>6.4</b>	<b>Simulations and Experimental Results</b>	<b>97</b>
6.4.1	Simulations Results	97
6.4.2	Experimental Results	97
6.4.3	Experimental Setup	99

---

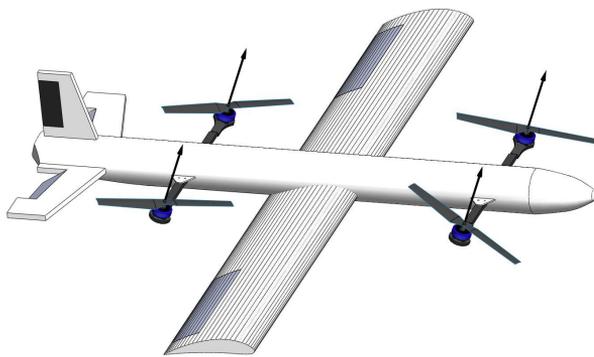
## 6.1 Introduction

Helicopters and fixed-wing aircrafts have their own advantages and shortcomings. Helicopters can take off and land vertically, but they cannot fly forward at high speed. On the other hand, fixed-wing aircrafts can fly forward at high speed, however, they cannot take-off and land vertically. Some advantages of both configurations can be combined in a single aerial vehicle, the so-called *convertible aircraft*.

When a convertible aircraft hovers, takes off or lands, it can be controlled as a classical helicopter. Also, during high speed flight, the convertible's configuration is similar to the one of a conventional airplane, in which the aerodynamic surfaces generate the necessary lift force to compensate the gravity.

While the convertible aircraft is a very promising concept, it also comes with significant challenges. Indeed it is necessary to design controllers which will work over the complete flight envelope of the vehicle: from low-speed vertical flight through high-speed forward flight. The main challenge is to deal with the large variation in the vehicle dynamics between these two different flight regimes.

In this first part of our research, we focus on hover position control of the convertible tilt-rotor aircraft, see Fig 6.1. This vehicle combines the high-speed cruise capabilities of a conventional airplane with the hovering capabilities of a helicopter by tilting its four rotors. Dynamics involved with the tilting mechanism are investigated. This tilting mechanism provides an additional input to the system, which serves not only as position actuator in hover mode but also as an input to performing the transition maneuver. Since the present research is focus on the hover control, the new input, namely the tilting angle, must be limited to a value less than 90 degrees, in order to control the forward position.



**Figure 6.1:** The PVHAT aircraft design.

There exist some tilt-rotor vehicles with a quad-rotor similar structure as the Boeing's V44 [59], [60] and the QTW UAV [61]. In [62] some preliminary results are presented for the vertical flight of a two-rotor MAV as well as a low-cost embedded flight control system.

The main contribution of this chapter is to develop a control strategy suitable for handling the hover dynamics of the convertible aircraft. The key idea is that the dynamics of the aircraft lend themselves well to individual control strategies for hover, transition and forward flight from a continuous point of view avoiding switching between both dynamics involved. In this first part of the work, we present an analysis of the convertible aircraft's mathematical model in which it appears that the control inputs for hover and airplane mode are different, and then both dynamics can be controlled individually. The control inputs in the hovering mode are given by the thrusts provided by the motors, while the control inputs in airplane mode are the forward thrust and the torques generated by the controlling surfaces.

The remainder of this chapter is organized as follows. Section 6.2 presents dynamics of the convertible aircraft in hover mode; also a model suitable for control is developed. The main result is stated in Section 6.3, where a Lyapunov-based control design is performed using the dynamical model obtained in Section 6.2. Simulations and experimental results are given in Section 6.4 where the effectiveness of the presented approach is tested.

## 6.2 System Description

In this section, an idealized mathematical model for the dynamics of the Quad Tilt-Rotor MAV is described.

### 6.2.1 Modeling of the PVHAT Aircraft

We will consider the nonlinear rigid body dynamics in terms of rotational and translational dynamics given by [40]. The derived model will lead to a model suitable for the purpose of control. Such model is given as follows

$$\begin{aligned}
 \dot{\xi} &= v \\
 \dot{v} &= \frac{RF}{m} - g e_{z\mathcal{I}} \\
 \dot{R} &= R\mathcal{S}(\Omega) \\
 \dot{\Omega} &= -J^{-1}\Omega \times J\Omega + J^{-1}\tau
 \end{aligned} \tag{6.1}$$

where  $\xi = (x, y, z)^T$  and  $v = (v_x, v_y, v_z)^T$  are the position and the velocity of the MAV relative to the inertial frame  $\mathcal{I} = (e_{x\mathcal{I}}, e_{y\mathcal{I}}, e_{z\mathcal{I}})$ .  $R \in SO(3)$  is the rotational matrix representing the orientation of the MAV in the body

coordinate frame  $\mathcal{B} = (e_{x\mathcal{B}}, e_{y\mathcal{B}}, e_{z\mathcal{B}})$  with respect to  $I$ ; the matrix  $R$  is a function of the Euler angles  $\phi$ ,  $\theta$  and  $\psi$ , i.e. roll, pitch and yaw, respectively.  $\Omega \in \mathfrak{R}^3$  is the body angular velocity vector.  $F \in \mathfrak{R}^3$  and  $\tau \in \mathfrak{R}^3$  are the force and torque, respectively applied at the center of mass of the MAV and specified with respect to  $\mathcal{B}$ .  $J \in \mathfrak{R}^3$  is the inertia matrix;  $m$  is the mass of the body; and  $ge_{z\mathcal{I}}$  is the gravitational force, where  $e_{z\mathcal{I}} = (0, 0, 1)$  is a unit vector. In (6.1)  $SO(3)$  denotes the special orthogonal group of  $\mathfrak{R}^{3 \times 3}$ . Further, we define by  $S(v)$  the operator from  $\mathfrak{R}^3 \rightarrow SO(3)$  such that

$$\forall v \in \mathfrak{R}^3, S(v) = \begin{pmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{pmatrix}$$

where  $v_i$  denotes the  $i^{th}$  component of the vector  $v$ .  $S(v)$  is the group of antisymmetric matrices of  $\mathfrak{R}^{3 \times 3}$ . Therefore,  $S(v)\Omega = v \times \Omega$ . We use four additional coordinate frames:  $A_i = (A_{ix}, A_{iy}, A_{iz})$ , for  $i = 1, \dots, 4$  which are associated with the four rotors (see Fig. 6.1). Thus, the orientation of each rotor w.r.t the body frame can be defined by the rotation matrix

$$\mathcal{R}^\gamma = \begin{pmatrix} \cos \gamma & 0 & \sin \gamma \\ 0 & 1 & 0 \\ -\sin \gamma & 0 & \cos \gamma \end{pmatrix} \quad (6.2)$$

The force  $F$  generated by the rotorcraft expressed in the body frame is given by

$$F = \begin{pmatrix} T_T \sin \gamma \\ 0 \\ T_T \cos \gamma \end{pmatrix} \quad (6.3)$$

where  $T_T = \sum_{i=1}^4 T_i$  and the thrust  $T_i$  can be modeled as  $T_i = C_l \omega_i^2$ , where  $C_l$  is the lift coefficient and  $\omega_i$  is the velocity of the  $i$ -th rotor.

We take into consideration the following assumption:

- a) Since in hover mode an angle greater than  $20^\circ$  causes an increase in the lift enough to achieve a transition from hover to fixed-wing mode [63], the tilting angle  $\gamma$  is imposed to be  $\gamma \leq 20^\circ$ .

The inputs are defined as

$$u_x = T_T \sin \gamma \quad (6.4)$$

$$u_z = T_T \cos \gamma. \quad (6.5)$$

Since it is possible to independently modify the speed of each propeller, the moments applied on the vehicle's body along an axis are given by

$$\tau = \begin{pmatrix} \tau_\phi \\ \tau_\theta \cos \gamma \\ \tau_\psi \end{pmatrix} \quad (6.6)$$

where  $\tau_\phi = lC_l((\omega_1^2 + \omega_3^2) - (\omega_2^2 + \omega_4^2))$ ;  $\tau_\theta = lC_l((\omega_1^2 + \omega_2^2) - (\omega_3^2 + \omega_4^2))$ ;  $\tau_\psi = d(\omega_1^2 - \omega_2^2 + \omega_4^2 - \omega_3^2)$ ;  $l$  is the lever;  $C_l$  is the thrust factor previously

defined;  $d$  is the drag factor and  $\omega_i$ , for  $i = 1, \dots, 4$ , is the rotor speed.

### 6.2.2 Dynamic Extension

It is important to note that the system (6.1) is not in pure strict-feedback structure and control vectors  $F$  and  $\tau$  have different relative degree with respect to the output  $\xi$ . Thus, we take the dynamic extension of the control  $F$  following the next procedure [64]. Let us consider the dynamic feedback law

$$\ddot{F} = \tilde{F} \quad (6.7)$$

where  $F$  and its first time derivative  $\dot{F}$  are regarded as components of the state of the augmented system (6.1), (6.7), i.e. they are becoming internal variables of the dynamic controller  $\tilde{F}$ .

In order to obtain the model for control purposes, let us consider system (6.1) with outputs  $\xi = (x, y, z)$ , which represent the vehicle's position, under the state coordinate transformation

$$\begin{aligned} X &:= \frac{RF}{m} - g e_{z\mathcal{I}} \\ Y &:= \dot{X} = \frac{\tilde{R}}{m} (S(\Omega)F + \dot{F}) \\ \tilde{\tau} &:= \tilde{\Omega} \end{aligned} \quad (6.8)$$

where  $X, Y$  are new states and  $\tilde{\tau}$  is the control input. By defining these new variables it is possible to represent (6.1) as

$$\dot{\xi} = v \quad (6.9a)$$

$$\dot{v} = X \quad (6.9b)$$

$$\dot{X} = Y \quad (6.9c)$$

$$\dot{Y} = \frac{R}{m} (\tilde{F} - S(F)\tilde{\tau} + 2S(\Omega)\dot{F} + S(\Omega)S(\Omega)F) \quad (6.9d)$$

**Remark 6.** Using the model (6.9), the new input vectors

$$(\tilde{F}, \tilde{\tau}) = (\ddot{u}_x, \ddot{u}_z, \tilde{\tau}_\phi, \tilde{\tau}_\theta, \tilde{\tau}_\psi) \quad (6.10)$$

have a relative degree equal to four with respect to the output  $\xi$ , so they can be assigned at the same stage eliminating the problem of the presence of an aggressive control which may lead to extreme ill-conditioning of the remaining closed-loop system [42]. In other words,  $(\tilde{F}, \tilde{\tau})$  will perform a non-aggressive control for the translation dynamics compared to the control of the rotational dynamics.

## 6.3

 Lyapunov-Based Tracking Control Strategy

In this section a control strategy for stabilization of (6.7), (6.9) is proposed. The main result is summarized in the following theorem.

**Theorem 7.** *Let  $(\hat{\xi}, \hat{\psi}) = (x_d(t), y_d(t), z_d(t), \psi_d(t)) \in \mathbb{R}^4$  be smooth trajectories. Let  $k_1$  and  $k_2$  denote strictly positive constants, and apply the control law*

$$\begin{aligned}
 \ddot{u}_x &= \ddot{Z}_1 - k_1 u_x - k_2 \dot{u}_x \\
 \ddot{u}_z &= \ddot{Z}_3 + \frac{u_x}{u_z} (k_1 u_x + k_2 \dot{u}_x) \\
 \ddot{\tau}_\phi &= \frac{u_x \ddot{\tau}_\psi - \ddot{Z}_2}{u_z} \\
 \ddot{\tau}_\theta &= \frac{1}{u_z} (k_1 u_x + k_2 \dot{u}_x) \\
 \ddot{\tau}_\psi &= \frac{\cos \theta}{\cos \phi} \ddot{L} - \frac{\tan \phi}{u_z} (k_1 u_x + k_2 \dot{u}_x)
 \end{aligned} \tag{6.11}$$

to the system (6.9), with  $\tilde{Z} = mR^T Z - 2S(\Omega)\dot{F} - S(\Omega)S(\Omega)F = (\tilde{Z}_1, \tilde{Z}_2, \tilde{Z}_3)$  and  $\tilde{L} = \frac{\cos \phi}{\cos \theta} \tilde{\tau}_\psi + \frac{\sin \phi}{\cos \theta} \tilde{\tau}_\theta$ . Then the tracking error  $e := (\xi(t) - \xi_d(t), \psi(t) - \psi_d(t))$  is asymptotically stable.

*Proof.* The proof is based on the backstepping procedure as will be explained below. We define the error  $e_1$  as

$$e_1 = \xi(t) - \xi_d(t) \tag{6.12}$$

where the time derivative of (6.12) is  $\dot{e}_1 = v - \dot{\xi}_d$ . By choosing the storage function

$$S_1 = \frac{1}{2} |e_1|^2 \tag{6.13}$$

the time derivative of (6.13) results in  $\dot{S}_1 = e_1^T \dot{e}_1$  and regarding  $v$  as a virtual controller, the control law

$$v_d = -e_1 + \dot{\xi}_d = -(\xi - \xi_d) + \dot{\xi}_d \tag{6.14}$$

stabilizes  $e_1$  to zero. We define the error

$$e_2 = v - v_d \tag{6.15}$$

and with (6.15) the time derivative of (6.13) becomes

$$\dot{S}_1 = -|e_1|^2 + e_1^T e_2 \tag{6.16}$$

Taking the time derivative of (6.15) we get

$$\dot{e}_2 = X - \dot{v}_d \tag{6.17}$$

and augmenting the storage function as

$$S_2 = S_1 + \frac{1}{2}|e_2|^2 \quad (6.18)$$

and deriving (6.18) yields

$$\dot{S}_2 = -|e_1|^2 + e_1^T e_2 + e_2^T (X - \dot{v}_d) \quad (6.19)$$

In order to get  $\dot{S}_2$  negative definite, we take  $X$  as a virtual control input as

$$X_d = \dot{v}_d - e_1 - e_2 \quad (6.20)$$

Taking the next error as

$$e_3 = X - X_d \quad (6.21)$$

the time derivative of (6.18) can be written as

$$\dot{S}_2 = -|e_1|^2 - |e_2|^2 + e_2^T e_3 \quad (6.22)$$

In order to stabilize the error related to the  $\psi$  angle, the error  $\delta_3 = \psi - \psi_d$  is introduced. Then, proceeding in the same way, consider the storage function

$$S_3 = S_2 + \frac{1}{2}|e_3|^2 + \frac{1}{2}\delta_3^2 \quad (6.23)$$

whose time derivative is given by

$$\dot{S}_3 = \dot{S}_2 + e_3^T (Y - \dot{X}_d) + \delta_3 (\dot{\psi} - \dot{\psi}_d) \quad (6.24)$$

We consider that  $Y$  and  $\dot{\psi}$  are the virtual control inputs, and then we choose

$$Y_d = \dot{X}_d - e_2 - e_3 \quad (6.25)$$

$$\dot{\psi}_d = \dot{\psi} - \dot{\delta}_3 \quad (6.26)$$

with the error terms

$$e_4 = Y - Y_d \quad (6.27)$$

$$\delta_4 = \dot{\psi} - \dot{\psi}_d \quad (6.28)$$

Thus, the derivative of (6.23) may be written as

$$\dot{S}_3 = -|e_1|^2 - |e_2|^2 - |e_3|^2 + e_3^T e_4 - \delta_3^2 + \delta_3 \delta_4 \quad (6.29)$$

Finally, we chose the storage function

$$S_4 = S_3 + \frac{1}{2}|e_4|^2 + \frac{1}{2}\delta_4^2. \quad (6.30)$$

Differentiating (6.30) we obtain

$$\dot{S}_4 = \dot{S}_3 + e_4^T(\dot{Y} - \dot{Y}_d) + \delta_4(\ddot{\psi} - \ddot{\psi}_d) \quad (6.31)$$

In order to express the control inputs  $\tilde{\tau}$  and  $\tilde{F}$ , let us consider the following notation

$$\dot{Y} = Z \quad (6.32)$$

$$\ddot{\psi} = L. \quad (6.33)$$

Now, we choose

$$Z = \dot{Y}_d - e_3 - e_4 \quad (6.34)$$

$$L = \ddot{\psi}_d - \delta_3 - \delta_4. \quad (6.35)$$

$\dot{S}_4$  can be represented as

$$\dot{S}_4 = -|e_1|^2 - |e_2|^2 - |e_3|^2 - |e_4|^2 - \delta_3^2 - \delta_4^2. \quad (6.36)$$

With the aforementioned analysis  $Z$  and  $L$  can be expressed as follows

$$\begin{aligned} Z &= -k_\xi(\xi - \xi_d) - k_v(\dot{\xi} - \dot{\xi}_d) - k_X(X - \ddot{\xi}_d) \\ &\quad - k_Y(Y - \xi_d^{(3)}) + \xi_d^{(4)} \\ L &= -k_\psi(\psi - \psi_d) - k_{\dot{\psi}}(\dot{\psi} - \dot{\psi}_d) + \ddot{\psi}_d \end{aligned}$$

where  $k_\xi, k_v, k_X, k_Y, k_\psi, k_{\dot{\psi}}$  are positive real numbers. From (6.9) and (6.32) we can write

$$\frac{R}{m}(\tilde{F} - S(F)\tilde{\tau} + 2S(\Omega)\dot{F} + S(\Omega)S(\Omega)F) = Z \quad (6.37)$$

then

$$\tilde{F} - S(F)\tilde{\tau} = mR^T Z - 2S(\Omega)\dot{F} - S(\Omega)S(\Omega)F := \tilde{Z} \quad (6.38)$$

where the auxiliary variable  $\tilde{Z} \in \mathfrak{R}^3$  is a function of known signals.

In order to obtain the  $\psi$  control, it is important to provide the desired control  $L$ . To achieve this objective, let us recall the kinematic relationship between the angular velocity  $\Omega$  and the generalized velocities  $\dot{\eta} = (\dot{\phi}, \dot{\theta}, \dot{\psi})$ , which is given as [40]

$$\dot{\eta} = W^{-1}(\eta)\Omega \quad (6.39)$$

Taking the time derivative of (6.39) we get

$$\ddot{\eta} = \dot{W}^{-1}\Omega + W^{-1}\dot{\Omega} = \dot{W}^{-1}\Omega + W^{-1}\tilde{\tau} \quad (6.40)$$

From (6.40)  $\ddot{\psi}$  can be deduced as

$$\ddot{\psi} = e_{z\tau}^T \ddot{\eta} \quad (6.41)$$

Computing (6.41) we obtain

$$\frac{\cos \phi}{\cos \theta} \tilde{\tau}_\psi + \frac{\sin \phi}{\cos \theta} \tilde{\tau}_\theta = L - e_{z\mathbf{x}}^T \dot{W}^{-1} \Omega := \tilde{L} \quad (6.42)$$

where the variable  $\tilde{L}$  is a function of known parameters.

Rewriting (6.38) and (6.42) one obtains

$$\begin{aligned} \tilde{Z}_1 &= \ddot{u}_x + u_z \tilde{\tau}_\theta \\ \tilde{Z}_2 &= u_x \tilde{\tau}_\psi - u_z \tilde{\tau}_\phi \\ \tilde{Z}_3 &= \ddot{u}_z - u_x \tilde{\tau}_\theta \\ \tilde{L} &= \frac{\cos \phi}{\cos \theta} \tilde{\tau}_\psi + \frac{\sin \phi}{\cos \theta} \tilde{\tau}_\theta. \end{aligned} \quad (6.43)$$

The system (6.43) needs to be solved in order to find the inputs. Let us choose

$$\tilde{\tau}_\theta = \frac{1}{u_z} (k_1 u_x + k_2 \dot{u}_x) \quad (6.44)$$

where  $k_1, k_2$  are positive real numbers. We consider that  $u_z \neq 0$ , from a discussion that will be shown later in the proof. Then, from (6.43) and (6.44), it follows that

$$\ddot{u}_x = \tilde{Z}_1 - k_1 u_x - k_2 \dot{u}_x. \quad (6.45)$$

From (6.43), we deduce that

$$\ddot{u}_z = \tilde{Z}_3 + \frac{u_x}{u_z} (k_1 u_x + k_2 \dot{u}_x). \quad (6.46)$$

The expression of  $\tilde{\tau}_\psi$  is obtained from (6.43) by

$$\tilde{\tau}_\psi = \frac{\cos \theta}{\cos \phi} \tilde{L} - \frac{\tan \phi}{u_z} (k_1 u_x + k_2 \dot{u}_x). \quad (6.47)$$

The control input for the roll dynamics  $\tilde{\tau}_\phi$  is deduced from (6.43) as

$$\tilde{\tau}_\phi = \frac{u_x \tilde{\tau}_\psi - \tilde{Z}_2}{u_z} \quad (6.48)$$

By construction, consider the Lyapunov function

$$V = S_1 + S_2 + S_3 + S_4. \quad (6.49)$$

The time derivative of (6.49) along the trajectories of (6.9) is given as

$$\dot{V} = -|e_1|^2 - |e_2|^2 - |e_3|^2 - |e_4|^2 - \delta_3^2 - \delta_4^2 \leq 0. \quad (6.50)$$

Therefore the system (6.9) together with the control law (6.11) is asymptotically stable.

From (6.9b) and (6.17) is clear that  $e_2 \equiv 0$  implies that

$$\dot{v}_d = \frac{RF}{m} - g. \quad (6.51)$$

Therefore one can define a locally unique thrust direction solution to the last equation in the domain where the Euler angles are well defined [65], i.e.  $(\theta, \phi) \in (-\pi/2, \pi/2)$ .

Moreover, Lyapunov's method can be used to find an estimate of the region of attraction [58]. The simplest such estimate i.e. a *conservative region of attraction* is given by the set

$$\Omega_c = \{\chi \in \mathbb{R}^6 | V(\chi) \leq c\}. \quad (6.52)$$

Rewriting (6.49) as  $V = \chi^T P \chi$ , where  $\chi = [e_1 \ e_2 \ e_3 \ e_4 \ \delta_3 \ \delta_4]^T$  and

$$P = \begin{pmatrix} 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3/2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/2 \end{pmatrix}$$

one can find such estimate  $c$  as

$$c = \min_{|b^T \chi| = g} \frac{g^2}{b^T P^{-1} b} \quad (6.53)$$

where  $\Omega_c$  is a subset of the domain  $D = \{|b^T \chi| < g\}$ ; where  $b^T = [1 \ 1 \ 1 \ 1 \ 1 \ 1]$ . We conclude that the elements of  $\chi$  are bounded as follows for all time  $t > t_0$ :  $|e_1| \leq g/6$ ,  $|e_2| \leq g/6$ ,  $|e_3| \leq g/6$ ,  $|e_4| \leq g/6$ ,  $|\delta_4| \leq g/6$ ,  $|\delta_3| \leq g/6$  and

$$|e_1 + e_2 + e_3 + e_4 + \delta_3 + \delta_4| \leq g. \quad (6.54)$$

With these values, the estimate of the region of attraction is computed as  $\Omega_c = \{\chi \in \mathbb{R}^6 | V(\chi) \leq 13.4283\}$ .

In order to prove that the total thrust  $T_T$  is bounded away from zero let us assume for simplicity that the desired trajectory is

$$(\hat{\xi}, \hat{\psi}) = (x_d(t), y_d(t), z_d(t), \psi_d(t)) = (0, 0, 0, 0).$$

Since  $|e_2| \leq g/6$  it follows that  $|\ddot{z}| \leq g/6$ . Note that applying Newton second law to the vertical axis we have

$$m\ddot{z} = T_T - mg$$

or

$$T_T = m(\ddot{z} + g)$$

Thus

$$|T_T| = m |g + \ddot{z}| > m(g - |\ddot{z}|) > m(g - g/6) = 5mg/6$$

which proves that the total thrust is never equal to zero provided that the state start in the proposed region of attraction. Other less conservative domains can certainly be proposed but the above proves that the thrust can be bounded away from zero.

□

**Remark 8.** *By integration, using (6.7) and (6.8) the original control inputs  $u_x$ ,  $u_z$  and  $\tau$  can be recovered.*

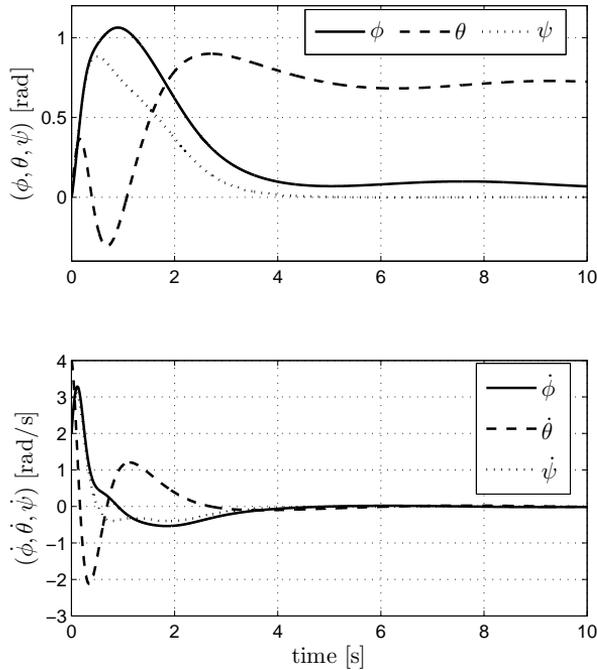
## 6.4 Simulations and Experimental Results

### 6.4.1 Simulations Results

To demonstrate the stability property of the closed-loop system, some simulations are carried out in this section. The parameters used for simulation purposes are as follows:  $mg = 1$ ,  $J = I_3$ ,  $k_\xi = 7$ ,  $k_v = 6$ ,  $k_X = 4$ ,  $k_Y = 2$ ,  $k_\psi = 10$ ,  $k_{\dot{\psi}} = 5$ ,  $k_1 = 10$ ,  $k_2 = 20$ . The desired trajectory is given by  $\hat{\xi}(t) = (\cos t, \sin t, \cos t)$  and  $\psi_d(t) = 0$ , with initial conditions:  $\xi(t) = (3, 1, 0)$ ;  $v(t) = (0, 0, 0)$ ;  $(\phi, \theta, \psi) = (0, 0, 0)$ ;  $(\dot{\phi}, \dot{\theta}, \dot{\psi}) = (2, 4, 3)$ ; and  $u_z = 1$ ,  $u_x = 0$ . The system response can be observed in Fig. 6.2; while the control inputs are visualized in Fig. 6.4. The convergence of the tracking error to zero is depicted in Fig. 6.3.

### 6.4.2 Experimental Results

The ultimate test for the proposed control strategy consists of verifying its proper operation when used during real-time experiments. The experimental procedure can be described as follows. The PVHAT aircraft platform starts its mission on the ground, its initial position is given by the optical flow sensor and the altitude sensor. Next, an autonomous takeoff is performed, achieving a desired altitude provided by the user by means of the radio controller; this set-point can be changed in any time. Once at this height, the vehicle is then required to navigate forward at a given desired speed, while regulating its heading angle  $\psi$  to a fixed value given by the user. To achieve a smooth transition from a forward velocity to a desired forward velocity, a ramp velocity



**Figure 6.2:** Attitude dynamics behavior.

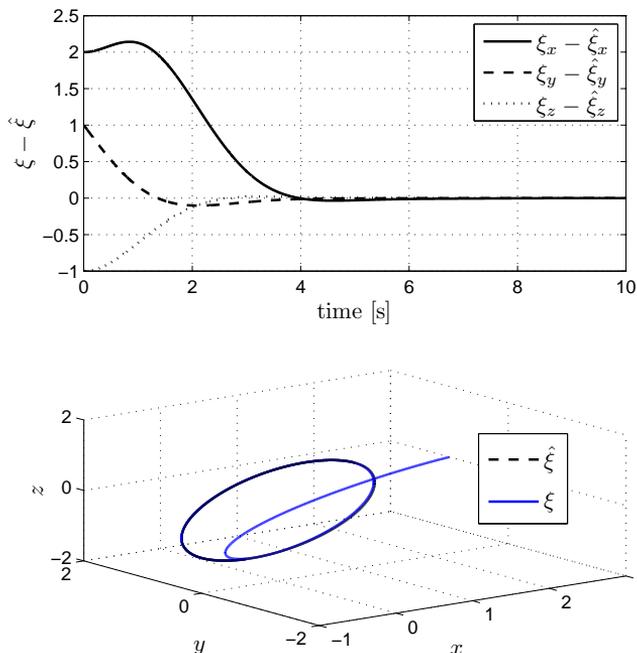
profile is implemented, in such a way that the velocity slowly increases after the takeoff task.

A 3-D reconstruction of the path navigated by the PVHAT aircraft during the real-time experiment is shown in Fig. 6.6. In this figure, the PVHAT aircraft trajectory is represented with respect to an inertial coordinate frame  $\mathcal{I}$ , whose origin is located at the MAV's initial position. It can be observed that the vehicle performs the tracking mission inside a rectangular area of  $4\text{m} \times 6\text{m}$ . It is worth mentioning that the forward displacement (relative to the body fixed frame) is obtained by integrating the optical flow.

Euler angles during experiment are shown in Fig. 6.5; it is shown that angular velocities varies around the zero value. Fig. 6.7 shows the input controllers given by the torques  $\tilde{\tau}$ , thrust  $T$  and tilting angle  $\gamma$ . It is important to note that the MAV takes off around  $t = 15$  seconds, and the thrust never achieves zero after that time.

Fig. 6.8 shows an image of the MAV in the experimental area during the real-time tests. In addition, a video showing the quad-rotor while performing some experiments can be observed at

<https://sites.google.com/site/gerardoflorescolunga/research/convertible-mav>



**Figure 6.3:** Convergence of the system's outputs  $\xi$  to the desired commands  $\hat{\xi}$ .

### 6.4.3 Experimental Setup

The experimental platform shown in Fig. 6.8 has been used to evaluate the performance of the proposed controller. Such experimental platform has been developed by the authors; it is based on the fuselage of a commercial airplane and a H-form quad-rotor. The structure of the H-form is built from balsa wood and carbon fiber. The tilt mechanism is controlled by two individual servomotors which can tilt the rotors from 0 up to 90 degrees. The flight control unit (FCU) is based on the PX4 Autopilot system. The PX4 Autopilot system consists of two boards: the PX4FMU and the PX4IO. The PX4FMU is a high-performance autopilot-on-module suitable for a wide variety of aircrafts, such as airplanes and quad-rotors. The PX4IO is the expansion module to handle the input/output signals of the system. The key features of the PX4FMU is the 168 MHz ARM-CortexM4F which contains a hardware floating-point-unit, 192 KB of SRAM and 1024 KB of flash memory. The PX4 autopilot system is equipped with a set of sensor units like the MPU-6000, a 3-axis accelerometer and gyroscope; the L3GD20, a 3-axis gyroscope; the HMC5883L, a 3-axis magnetometer; and the MS5611, a barometric pressure sensor. They are used to provide an estimation of the attitude using an extended Kalman

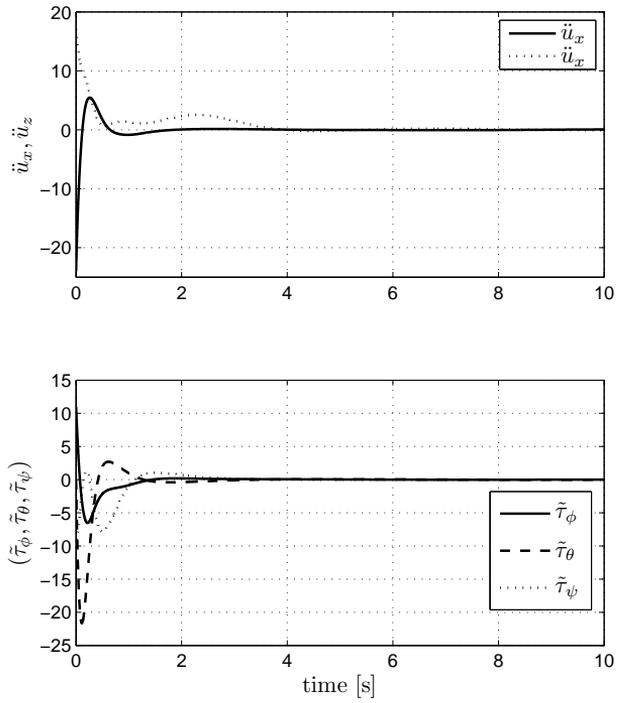


Figure 6.4: Control inputs  $\tilde{\tau}$ ,  $\ddot{u}_x$  and  $\ddot{u}_z$ .

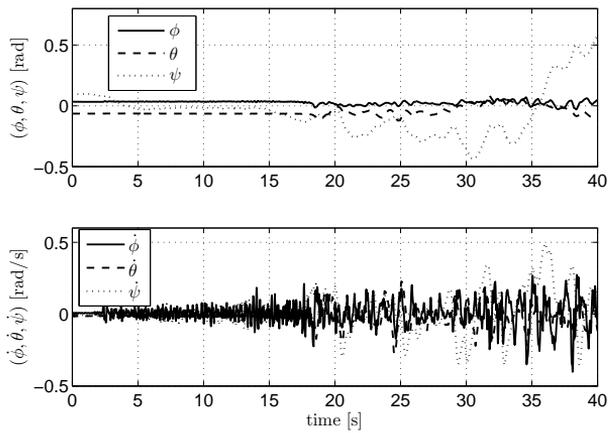
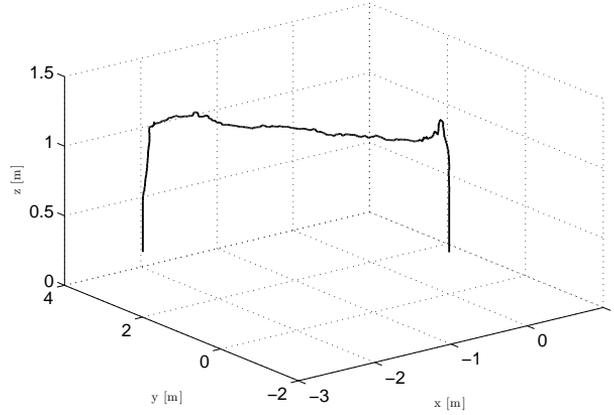
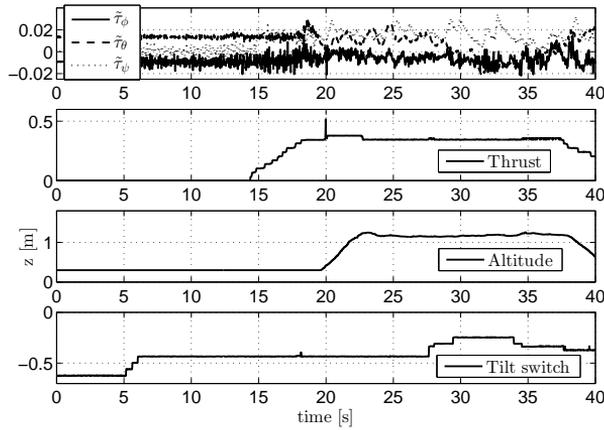


Figure 6.5: Attitude dynamics behavior in real experiments.

filter (EKF).



**Figure 6.6:** Convergence of the system's outputs  $\xi$  to the desired commands  $\hat{\xi}$  in real experiments.



**Figure 6.7:** Control inputs  $\tilde{\tau}$ ,  $T$ , altitude  $z$  and control  $\gamma$  in real experiments.

The QGroundControl is a specially designed ground station that allows the user to communicate with the autopilot. Some of the main features of this ground station is the visual representation of the sensors. The control parameters can be adjusted by means the QGroundControl. The wireless communication link between the PX4 autopilot and the QGroundControl is performed through a wireless serial protocol using two XBee's modems.



(a) Hover mode.



(b) Airplane mode.

**Figure 6.8:** The PVHAT aircraft experimental platform: The MAV equipped with imaging, inertial and altitude sensing systems: (a) The MAV in helicopter mode with the tilting angle  $\gamma = 0$ ; (b) The MAV in airplane mode with the tilting angle  $\gamma = 90$ .

# 7

## NONLINEAR CONTROL DESIGN OF TRANSITION MANEUVERS FOR CONVERTIBLE AERIAL VEHICLES

---

*This chapter presents a new control strategy for the transition between airplane and helicopter mode, and viceversa, in convertible planes. The analysis is carried out for the longitudinal model of the PVHAT (Planar Vertical Helicopter-Airplane Transition) aircraft, which is an airplane having tilting rotors in order to achieve the transition maneuver. The control strategy is smooth in the sense that it does not involve commutation between two different controllers. The resulting closed loop control algorithm is proved to be globally asymptotically stable. The altitude and the longitudinal speed are proved to converge to desired values. The proposed controller has been successfully tested in numerical simulations.*

### Contents

---

<b>7.1</b>	<b>Introduction</b>	<b>104</b>
<b>7.2</b>	<b>System Description</b>	<b>105</b>
7.2.1	Flight envelope description	105
<b>7.3</b>	<b>Control Algorithm Design</b>	<b>107</b>
<b>7.4</b>	<b>Numerical Results</b>	<b>110</b>
7.4.1	Hover to airplane mode simulation	110
7.4.2	Airplane to hover mode simulation	111

---

## 7.1 Introduction

---

Among the vast variety of aerial vehicle configurations, the *convertible aircraft* has recently gained attention in the control community. This is not only because of its capability of combining both the flight efficiency of an airplane and the maneuverability of a helicopter, but also due to the inherent problem present in this kind of vehicles, namely *the transition maneuver*; where the aircraft behaves as a combination of helicopter and airplane dynamics. Among convertible aircrafts designs, tilt-rotor vehicles constitute an attractive research area due to their stability, energy efficiency and controllability [59]. Furthermore this design is characterized by the tilting mechanism which allows the performance of the transition maneuver keeping a constant attitude throughout the flight envelope, in contrast to *tail-sitter vehicles* where the aircraft's fuselage must be rotated in order to accomplish the transition maneuver. This advantage permits additional stability due to the mechanical configuration.

Some airplanes with four tilting rotors are in current development, such as the Bell Boeing Quad Tilt-Rotor (QTR) derived from the V-22 Osprey tilt-rotor developed jointly by Bell Helicopter and Boeing [66]; the QTW of Chiba University [67]; the SUAVI UAV [68]; and the QUX-02 of Japan Aerospace Exploration Agency [69]. In [70] the authors described the development of a two-rotor tail-sitter. The control architecture features a switching logic of classical linear controllers to deal with the vertical, transition and forward flight. In [71] the authors present a classical airplane configuration to perform both operational modes. The hover flight is autonomously controlled by an onboard control flight system while the transition and cruise flight is controlled manually. A standard PD controller is employed during hover flight to command the rudder and the elevator. In [72] the authors describe the development of robust, multi-variable  $H_\infty$  control systems for the cruise and hover operating points of an experimental tilting-wing aircraft. The effectiveness of a similar configuration is reported in [73], where the problem of stabilization of a V/STOL (Vertical/Short Take-Off and Landing) aircraft is addressed. However, the authors do not present a strategy to control the transition maneuver. In [68] the authors present a new unmanned aerial vehicle called SUAVI, which is capable of Vertical Take-Off and Landing (VTOL) like a helicopter and long duration horizontal flight like an airplane. A hierarchical control system is designed where a high level controller is responsible for generating references for low level controllers for attitude and altitude stabilization. However, the paper does not present the analysis concerning the transition maneuver, which is the most interesting phenomena in this kind of vehicles. In [74] an optimal transition maneuver for the tail-sitter V/STOL aircraft is investigated. The authors present numerical simulations showing the performance in the transition maneuver. More recently [69] and [68] presented the development of convertible experimental platforms. Nevertheless most of the available work concerning convertible aircrafts [70], [71], [73], [75], [68], [76] have not addressed the control problem of the transition maneuver. On the aforementioned works some analysis are presented considering the airplane and the helicopter dynamics independently from each other. Thus, the controllers

for the hover and airplane modes are derived individually using a switching condition, but without developing any analysis between those flying regimes. There exist many works in the literature studying either the control of VTOL aircrafts or either the control of airplanes, see for example [77], [78], [79] for fixed-wing aircrafts and [80], [81], [82], [83] for helicopter or V/TOL aircrafts.

The key problem in the transition maneuver comes from the fact that the lift force in the airplane mode strongly depends on the velocity. A simple control strategy can be obtained by commuting between the airplane and the helicopter modes. Nevertheless such a control strategy may lead to a poor performance and a reduced stability domain. It is not straightforward to obtain a smooth and stable transition strategy using existing control algorithms. Therefore the transition maneuver requires the synthesis of a new control algorithm. For the purpose of illustration, it is particularly useful to consider a simple model of the convertible aircraft that has a minimum number of states and inputs but retains many of the features that must be considered when designing control laws for a real convertible aircraft. Fig. 6.1 shows our prototype PVHAT (Planar Vertical Helicopter-Airplane Transition) aircraft. The aerial vehicle's states are simply the altitude  $z$  of the aircraft center of mass, and the velocities  $\dot{x}$ ,  $\dot{z}$  of the aircraft. The control inputs,  $T$ ,  $\gamma$ , are the thrust and the rotor's tilting angle. The main contribution of this chapter is the proposition of a control strategy suitable for handling the transition maneuver for the so-called convertible aircrafts using smooth continuous control algorithms and avoiding switching between helicopter and airplane controllers. A Lyapunov stability analysis is presented and the performance of the control strategy is tested in numerical simulations.

This chapter is organized as follows. Section 7.2 describes the dynamics of the convertible aircraft. The problem statement and main result are given in Section 7.3. Some numerical results obtained with the presented approach are described in Section 7.4.

## 7.2 System Description

---

This section presents the longitudinal equations of motion as well as the aerodynamics of the PVHAT aircraft shown in Fig. 6.1.

### 7.2.1 Flight envelope description

---

The flight envelope of the vehicle encompasses three different flight conditions described as follows:

1. Hover flight mode. The vehicle is driven by four rotors as a classical quadrotor. The controller for this regime disregards the aerodynamic terms for low translational speed.
2. Airplane mode. In this mode, the aircraft has gained enough speed to generate aerodynamic forces to lift and control the aircraft using the control surfaces.
3. Transition flight mode. This mode is an intermediate operation between hover and airplane modes. The purpose of this chapter is to develop a control strategy for this regime.

In the case of the PVHAT aircraft, the rotors can be simultaneously and gradually tilted from  $0^\circ$  to  $90^\circ$  to change the flight configuration from hover to level flight and viceversa. The tilt angular position of such mechanism is represented by  $\gamma$  (see Fig. 7.1).

Using the 2nd Newton's law, the longitudinal aerodynamical model of the convertible vehicle can be written as

$$m\ddot{x} = T \sin(\gamma) - d\dot{x}^2 \quad (7.1)$$

$$m\ddot{z} = T \cos(\gamma) + l\dot{x}^2 - mg \quad (7.2)$$

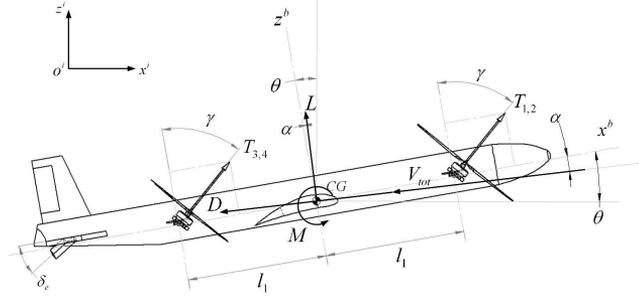
where  $x$  is the longitudinal position,  $z$  is the vertical position,  $m$  is the total mass of the aircraft,  $g$  is the gravitational acceleration,  $T$  and  $\gamma$  are considered as the control inputs,  $T$  is the total thrust generated by the four propellers and  $\gamma$  is the tilt angle. The lift and drag aerodynamic forces have been approximated respectively as [84]:  $L(\cdot) \approx l\dot{x}^2$  and  $D(\cdot) \approx d\dot{x}^2$  where  $l = 0.5C_l\rho S$ ,  $d = 0.5C_d\rho S$ ,  $\rho$  is the air density,  $S$  the airfoil's surface and  $(C_l, C_d)$  are constant coefficients depending on the airfoil characteristics. Small angles of attack  $\alpha$  will be considered, i.e.  $|\alpha| \leq \bar{\alpha}$ , where  $\bar{\alpha} \approx 0.2$  rad, and the values for  $C_l$  and  $C_d$  will be approximated as  $C_l \approx C_{l_\alpha}\alpha$  and  $C_d \approx C_{d_0} + C_{d_\alpha}\alpha^2$ , where the aerodynamic parameters  $C_{l_\alpha}$ ,  $C_{d_0}$ ,  $C_{d_\alpha}$  depend on the airfoil characteristics [84].

The mathematical model is completed by considering the following attitude dynamics

$$J\ddot{\theta} = \tau \quad (7.3)$$

where  $J$  represents the system's inertia,  $\theta$  is the pitch angle and  $\tau$  is the torque input due to a difference in the rotors thrust in the hover mode and an elevator displacement in the airplane mode.

**Remark 9.** *The attitude dynamics for  $\theta$  is fully-actuated; this can be seen from (7.3). In fact sub-system (7.3) represents the fastest dynamics of the entire aircraft system (7.1), (7.2) and (7.3). The inner control loop provides high gain stabilization of the vehicle's angular velocity based on the Inertial Measurement Unit (IMU) [65]. Then, sub-system (7.3) can be controlled independently of (7.1) and (7.2). One controller for this sub-system can be found in our previous work [6]. Therefore, in the sequel we focus on the control of system (7.1)-(7.2).*



**Figure 7.1:** Free-body scheme showing the coordinate systems, forces and variables involved on the PVHAT.

## 7.3 Control Algorithm Design

Let us consider the airplane mode, i.e. when  $\gamma = \pi/2$  and  $T = mg$ . From (7.1) it is clear that the maximum translational velocity is achieved when  $\ddot{x} = 0$ , i.e. when  $\dot{x}_{\max} = \sqrt{\frac{mg}{d}}$ . On the other hand the stall velocity is the minimum speed required so that the lift force compensates the weight of the aircraft. Therefore from (7.2) the stall velocity is  $\dot{x}_{\text{stall}} = \sqrt{\frac{mg}{l}}$ . It is important to notice that the inequality  $0 < \sqrt{\frac{mg}{l}} < \sqrt{\frac{mg}{d}}$  holds, since  $d < l$  is always fulfilled in any real aircraft design.

Since we are interested in controlling the aircraft linear velocity  $\dot{x}$  and altitude  $z$ , we will consider the variables  $(\dot{x}, z, \dot{z})$  as the state of system (7.1)-(7.2). For practical reasons we will assume that the total thrust  $T$  is large enough to compensate the aircraft weight and an additional amount for control purposes. Therefore  $T \leq mg + w_c$ .

It is important to notice that the control approach should take into account the fact that in real applications the thrust is by nature nonnegative and is generated by actuators having saturation limits. This motivates us to define the controllers in terms of linear saturation functions, whose definition is given next.

**Definition 3.** Given two positive constants  $L, M$  with  $L \leq M$ , a function  $\sigma : \mathbb{R} \rightarrow \mathbb{R}$  is said to be a linear saturation for  $(L, M)$  if it is a continuous, nondecreasing function satisfying:

- $\sigma(s) > 0$  for all  $s \neq 0$ ;
- $\sigma(s) = s$  when  $|s| \leq L$ ;
- $|\sigma(s)| \leq M$  for all  $s \in \mathbb{R}$ .

Taking into account the aforementioned discussion and Definition 3, the main result is stated in the following theorem.

**Theorem 10.** Let  $(L_{eps}, M_{eps})$ ,  $(L_{mg}, M_{mg})$ ,  $(L_{int}, M_{int})$  and  $(L_{ext}, M_{ext})$  be positive constants, where  $M_{int} < \frac{1}{2}L_{ext}$  and  $M_{eps} < \frac{1}{2}L_{mg}$ . Consider linear saturation functions  $\sigma_{mg}$ ,  $\sigma_\varepsilon$ ,  $\sigma_{ext}$  and  $\sigma_{int}$ . Then, assume that the control inputs  $T, \gamma$  satisfy the following:

$$T \sin \gamma = -\sigma_{mg}(-d\dot{x}^2 + \sigma_\varepsilon(\dot{x} - \dot{x}_d)) \quad (7.4)$$

$$T \cos \gamma = mg - l\dot{x}^2 + \mu \quad (7.5)$$

$$\mu = -\sigma_{ext}(\dot{z} + \sigma_{int}((z - z_d) + \dot{z})) \quad (7.6)$$

where  $\dot{x}_d \leq \dot{x}_{\max}$  is the desired translational speed and  $z_d$  is the desired altitude. The closed loop system obtained from (7.1)-(7.2) and the above control law is globally asymptotically stable.

*Proof of Theorem 10.* To simplify the notation we will consider in the sequel  $m = 1$ . From system (7.2) and (7.5) we obtain

$$\ddot{z} = \mu \quad (7.7)$$

Define

$$z_1 = z$$

$$z_2 = \dot{z}$$

and consider the transformation

$$y_1 = z_1 + z_2$$

$$y_2 = z_2$$

Combining the above we get

$$\dot{y}_1 = y_2 + \mu \quad (7.8)$$

$$\dot{y}_2 = \mu \quad (7.9)$$

and

$$\mu = -\sigma_{ext}(y_2 + \sigma_{int}((y_1 - y_{1d}))) \quad (7.10)$$

where  $y_{1d} = z_d$ , assuming  $\dot{z}_d = 0$ . Following the ideas in [85] and [86], the analysis starts by considering the evolution of the state  $y_2$ . Let

$$V_y = \frac{1}{2}y_2^2$$

be a candidate Lyapunov function for system (7.8). The time derivative of  $V_y$  along trajectories of the closed loop system (7.5)-(7.6)-(7.8) is given by

$$\begin{aligned} \dot{V}_y &= y_2 \dot{y}_2 \\ &= y_2 (-\sigma_{ext}(y_2 + \sigma_{int}((y_1 - y_{1d})))) \end{aligned}$$

Given that  $M_{int} < \frac{1}{2}L_{ext}$ , it follows that  $\dot{V}_y < 0$ ,  $\forall y_2 \notin Q_{y2}$  where  $Q_{y2} = \{y_2 :$

$|y_2| \leq \frac{1}{2}L_{ext}\}$ . As a consequence,  $y_2$  enters in the set  $Q_{y_2}$  in a finite time and remains in  $Q_{y_2}$  thereafter. Regarding state  $y_1$ , observe that once  $y_2$  has entered in the set  $Q_{y_2}$ , the argument of  $\sigma_{ext}$  is bounded as  $|y_2 + \sigma_{int}((y_1 - y_{1d}))| \leq L_{ext} + M_{int} \leq L_{ext}$ . Consequently,  $\sigma_{ext}$  operates in its linear region after  $y_2$  has entered  $Q_{y_2}$ . Therefore the evolution of  $y_1$  is given by  $\dot{y}_1 = -\sigma_{int}(y_1 - y_{1d})$ . Following the same procedure as for  $y_2$  it is possible to show that  $y_1$  enters an analogous set  $Q_{y_1}$  in finite time, and remains in  $Q_{y_1}$  thereafter. After a finite time, the closed loop system has the form

$$\dot{y}_1 = -y_1 + y_{1d} \quad (7.11)$$

$$\dot{y}_2 = -(y_1 - y_{1d}) - y_2 \quad (7.12)$$

and then the closed loop system (7.11) is locally exponentially stable in the domain where  $y_2 \in Q_{y_2}$ .

Since the structure of sub-system (7.1) and sub-system (7.2) is basically the same, a similar procedure can be followed to prove global asymptotic stability of the system having output  $(\dot{x} - \dot{x}_d)$ .  $\square$

### 7.3.0.1

Computation of the control inputs  $T$  and  $\gamma$

Notice from (7.1)-(7.2) that  $T \sin \gamma$  and  $T \cos \gamma$  depend on each other. In fact the real control inputs are the thrust  $T$  and the tilting angle  $\gamma$ . We will first show that the resulting thrust  $T$  is bounded by the aircraft weight i.e.  $T \leq mg + w_c$  as required. Consider the system in (7.1) and (7.2) and the control strategy in (7.4), (7.5) and (7.6). Using the trigonometric identity  $\sin^2 \gamma + \cos^2 \gamma = 1$  it follows that once the saturation functions operate in their linear region we have:

$$\begin{aligned} T^2 &= (mg - l\dot{x}^2 + \mu)^2 + (d\dot{x}^2 + \bar{\varepsilon})^2 \\ &= (mg - l\dot{x}^2 + \mu + d\dot{x}^2 + \bar{\varepsilon})^2 \\ &\quad - 2(mg - l\dot{x}^2 + \mu)(d\dot{x}^2 + \bar{\varepsilon}) \\ &= (mg - (l - d)\dot{x}^2 + \mu + \bar{\varepsilon})^2 \\ &\quad - 2(mg - l\dot{x}^2 + \mu)(d\dot{x}^2 + \bar{\varepsilon}) \\ &\leq (mg + \mu + \bar{\varepsilon})^2 \end{aligned}$$

where  $\bar{\varepsilon} \leq M_{mg}$ . Then the real inputs are given by

$$\begin{aligned} T &= \sqrt{(-\sigma_{mg}(-d\dot{x}^2 + \sigma_\varepsilon(\dot{x} - \dot{x}_d)))^2 + (mg - l\dot{x}^2 + \mu)^2} \\ \gamma &= \arcsin\left(\frac{|-\sigma_{mg}(-d\dot{x}^2 + \sigma_\varepsilon(\dot{x} - \dot{x}_d))|}{T}\right) \end{aligned} \quad (7.13)$$

It is important to note that when the aircraft achieves the transition from hover to level flight, the system (7.1)-(7.2) changes to the following dynamics

$$\begin{aligned}\ddot{x} &= T - d\dot{x}^2 \\ \ddot{z} &= l\dot{x}^2 - mg\end{aligned}$$

since  $\gamma \rightarrow \pi/2$ . The above is the dynamics of an airplane, and therefore any controller for an aircraft can be applied. The same applies to the transition from level to hover mode, but in this case  $\gamma \rightarrow 0$  and thus, any controller for VTOL aircraft can be implemented.

## 7.4 Numerical Results

---

To illustrate and validate the above approach, we carried out some simulations considering both the transition from hover to level flight and the transition from level to hover flight mode. The simulation results have been obtained by using the real parameters of the Quad-tilt rotor [6] sketched in 6.8. To get the simulation results, we have used the following linear saturation functions:

$$\sigma_{ij}(s) = \begin{cases} \frac{\arctan(a[s-L_{ij}])}{a} + L_{ij} & \text{if } s > L_{ij} \\ s & \text{if } |s| \leq L_{ij} \\ \frac{\arctan(a[s+L_{ij}])}{a} - L_{ij} & \text{if } s < -L_{ij}. \end{cases}$$

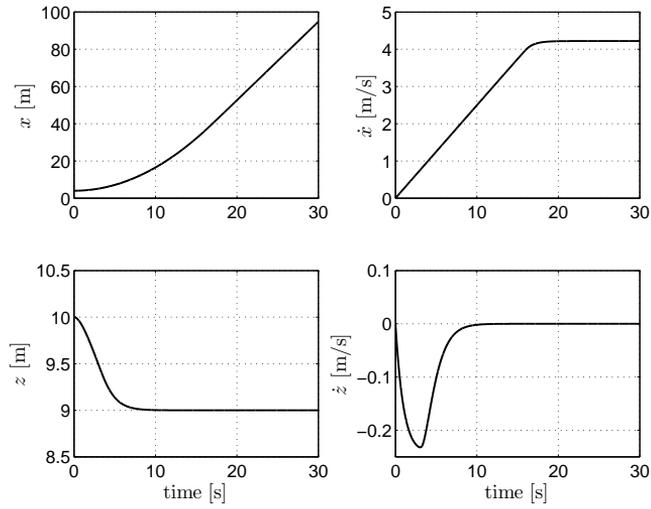
where  $a := \frac{\pi}{2(M_{ij}-L_{ij})}$ , with  $M_{ij} > L_{ij} > 0$ . Details about these saturation function such as its first and second time derivatives can be seen in [87]. For both simulation cases, the following parameters are chosen:  $L_{eps} = 0.2$ ,  $M_{eps} = 0.25$ ;  $L_{mg} = 0.1mg$ ,  $M_{mg} = 0.15mg$ ;  $L_{int} = 0.2$ ,  $M_{int} = 0.25$ ;  $L_{ext} = 1$ ,  $M_{ext} = 1.5$ ;  $m = 2$ ,  $g = 9.8$ ; and  $d = 0.01$ ,  $l = 1.1$ .

### 7.4.1 Hover to airplane mode simulation

---

Let us start with the transition maneuver from helicopter to airplane mode considering the following initial conditions:  $\dot{x}(0) = 0$ ,  $x(0) = 4$ ,  $\dot{z}(0) = 0$ ,  $z(0) = 10$ . The desired values are chosen to be:  $z_d = 9$  and  $\dot{x}_d = 4.221$ .

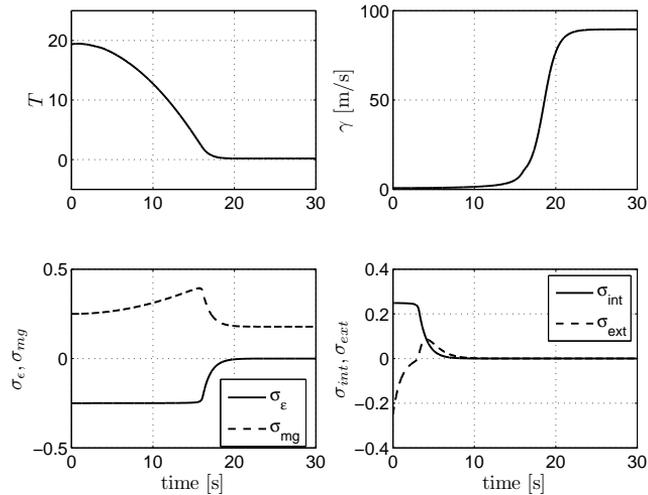
From Fig. 7.3 one can see that the function  $\sigma_{int}$  is saturated during the first 4 seconds of the simulation. As a consequence the states converge slowly. In fact, the rate of convergence is diminished proportionally to the parameters  $L_{eps}$ ,  $M_{eps}$ ,  $L_{mg}$ ,  $M_{mg}$ ,  $L_{int}$ ,  $M_{int}$ ,  $L_{ext}$ ,  $M_{ext}$ . However, the performance of the proposed controller is satisfactory as can be observed in Fig. 7.2.



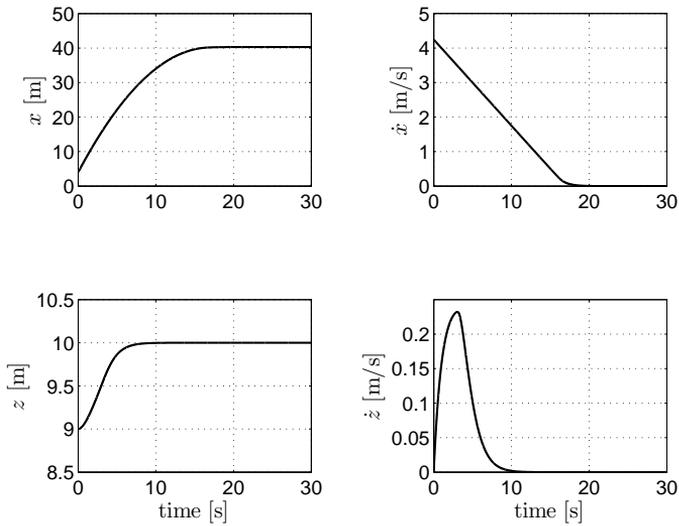
**Figure 7.2:** Transition from hover to level flight. Stabilization of  $\dot{x}$  and  $z$ .

### 7.4.2 Airplane to hover mode simulation

Now consider the transition maneuver from helicopter to airplane mode with the next initial conditions:  $\dot{x}(0) = 4.248$ ,  $x(0) = 4$ ,  $\dot{z}(0) = 0$ ,  $z(0) = 9$ . The desired values are chosen to be:  $z_d = 10$  and  $\dot{x}_d = 0$ . The performance of the

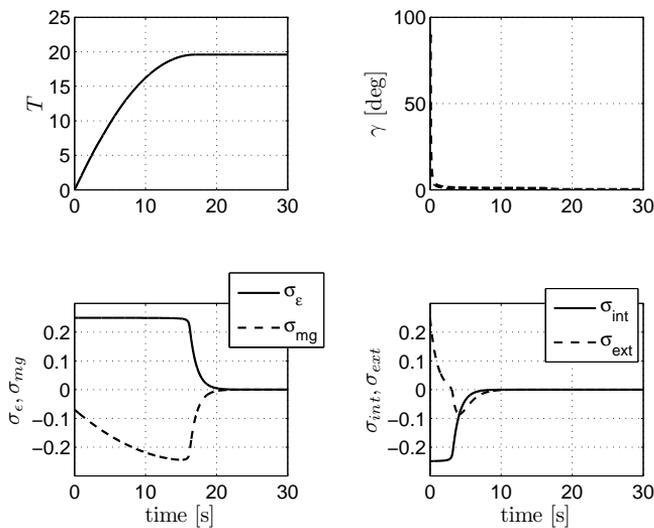


**Figure 7.3:** Control inputs ( $T, \gamma$ ) and saturation functions for helicopter to airplane transition.



**Figure 7.4:** Transition from level to hover flight. Stabilization of  $\dot{x}$  and  $\dot{z}$ .

controller is shown in Fig. 7.5. As in the transition from hover to airplane mode, the rate of convergence is affected by the magnitude of the control parameters  $L_{eps}$ ,  $M_{eps}$ ,  $L_{mg}$ ,  $M_{mg}$ ,  $L_{int}$ ,  $M_{int}$ ,  $L_{ext}$ ,  $M_{ext}$ . Finally, the convergence of the system is demonstrated as shown in Fig. 7.4.



**Figure 7.5:** Control inputs ( $T, \gamma$ ) and saturation functions for airplane to helicopter transition. The functions  $\sigma_\epsilon$  and  $\sigma_{int}$  are saturated for 16 and 3 seconds respectively, and thus the systems states converge slowly.

# Part IV

VISION

APPLICATIONS



# 8

## LYAPUNOV-BASED SWITCHING CONTROL FOR A ROAD ESTIMATION AND TRACKING

---

*This chapter addresses the problem of estimation and tracking of a road using the Quad-plane Convertible MAV. For that objective, we consider the following scenario: (i) no previous knowledge of the road, i.e. shape, dimension and color, (ii) loss of information by the sensors is considered and (iii) nonlinear dynamics of the MAV is taken into consideration. Aiming at this goal, two operational regions are defined: road detected and road not detected by the sensors. A switching between the measurements of imaging and inertial sensors enables estimation of the required states in both operational regions. For dealing with both aforementioned cases, a Lyapunov-based switching control for stabilizing the vehicle's position is proposed. Unmodeled dynamics such as friction forces are estimated by means of the proposed controller. The global exponential stability of the position subsystem together with the switching controller is demonstrated exploiting the fact that the individual closed-loop systems are globally exponentially stable and the switching is sufficiently slow, so as to allow the transient effects to dissipate after each switch. The control law is validated on the Quad-plane Convertible MAV experimental platform, showing the expected behavior during autonomous navigation.*

## Contents

---

<b>8.1</b>	<b>Introduction</b>	<b>117</b>
<b>8.2</b>	<b>Problem Statement</b>	<b>118</b>
<b>8.3</b>	<b>States estimation using a vision sensor</b>	<b>119</b>
8.3.1	Computing the heading angle	121
8.3.2	Computing the relative lateral position	122
8.3.3	Translational velocities	122
8.3.4	Computing the heading and the lateral position	123
<b>8.4</b>	<b>Control Strategy</b>	<b>125</b>
8.4.1	Operating modes for the road following mission	126
8.4.2	Control Laws for each Operating Mode	126
8.4.3	Stability Analysis of the Lateral Position Control	130
<b>8.5</b>	<b>Experimental platform</b>	<b>132</b>
8.5.1	The convertible MAV experimental vehicle	132
8.5.2	Embedded vision system	133
8.5.3	Ground station	134
<b>8.6</b>	<b>Simulations and Real Time Experiments</b>	<b>134</b>
8.6.1	Numerical Simulations	134
8.6.2	Real Time Experiments	136

---

## 8.1 Introduction

This work is motivated by the theory of switching systems and the need to develop effective micro aerial vehicle (MAV) controllers and state estimators, not only for attitude and position stabilization, but also for successfully executing a more complex predefined mission. Enabling a MAV to perform a task in a completely autonomous fashion is perhaps one of the most important and challenging control problems. Furthermore, estimating the vehicle position with sufficient accuracy is one of the important problems related with this subject.

The task addressed in this chapter consists of performing an autonomous navigation mission. The objective is to enable a MAV to estimate and track a road with no prior knowledge of the path that must be followed. In addition, the mission must be successfully performed in the presence of external disturbances, which are present not only in the attitude angles, but also in the vehicle translational dynamics. To further improve the autonomy of the vehicle, the mission is complemented with an autonomous take-off, as well as with a landing at some specific point near the end of the road. With the purpose of simplifying the inherent complexity of the whole aforementioned task, the complete mission is divided into five modes of operation: *take-off*,  *$\psi$  alignment*, *lateral position ( $y$ -state) alignment*, *road tracking*, and *landing*.

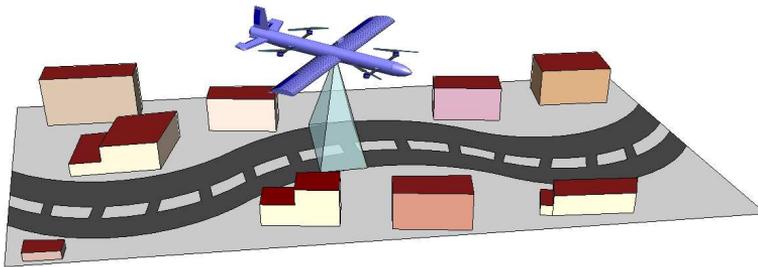
The problem of stabilizing a quad rotorcraft using visual feedback has been addressed by several research groups, see for example [88], [89], [90] and the references therein. From previous experiences performed, it has been found that proportional derivative (PD) controllers work efficiently, in practice, for stabilizing the MAV attitude dynamics [91]. In addition, these kind of controllers have proved to be robust enough for applications having objectives similar to the one being addressed in the present research [92]. In both previously mentioned approaches, the imaging system was used for estimating the vehicle translational dynamics. However, the strategies proposed there are not robust for dealing with external disturbances which may cause the imaging system to temporarily lose the artificial landmarks used for obtaining the vehicle states. Switching system ideas have been previously implemented to solve MAV-related problems. The implementation of different controllers with different gains is addressed in [93], this approach is adopted by the fact that the sole use of a PD controller in the position dynamics is not enough for attenuating the disturbances caused by, for example, wind gusts. In [94], the authors propose a switching strategy for estimating the states of a MAV equipped with imaging, inertial and air data sensing systems, capable of dealing with instants of time when the visual detection of the landmark fails. Following a similar reasoning, in [41] the authors present the analysis and control of a vertical take-off and landing (VTOL) aircraft at simulation level. By using a common Lyapunov function, the stability of the complete system, divided into three modes, is proved.

In this chapter, we aim at estimating and controlling the vehicle relative

position, orientation, and velocity with respect to a road that must be followed. An onboard camera allows estimation of the vehicle heading angle w.r.t. the longitudinal orientation of the road. Similarly, the imaging sensor is used for stabilizing the lateral distance of the vehicle in order to navigate exactly over the road. With the objective of developing a switching control strategy for estimation and tracking purposes, two operational regions are defined: one region for the situation when the road is in the camera field of view (FOV), and another region for when this is not the case. The system stability is shown not only in the two operational regions, but also in the switching boundaries between them. The performance of the switching control was tested in real time experiments, successfully validating the effectiveness of the proposed approach.

The chapter is organized as follows. Section 8.2 presents the problem statement. The methods proposed to estimate the vehicle states are described in Section 8.3. The switching control strategy and a discussion of the system stability are presented in Section 8.4. Section 8.5 presents the tilt-rotor convertible MAV experimental platform. Numerical simulations, as well as the performance of the MAV during real-time experiments are shown in Section 8.6.

## 8.2 Problem Statement



**Figure 8.1:** Road following setup: The tilt-rotor convertible MAV under consideration is equipped with inertial sensors, an ultrasonic range finder, and a calibrated camera. The objective is to enable the MAV to estimate and track the road, with no prior knowledge of the path that must be followed.

In order to perform the road-following task, the convertible MAV must be capable of measuring its angular dynamics, as well as its relative 3-dimensional position and heading w.r.t. the road. An image describing the scenario considered in the present research is shown in Figure 8.1. The convertible MAV platform is equipped with inertial sensors, an ultrasonic range finder, and a calibrated camera. The inertial system provides the vehicle angular dynamics, allowing the development of a control strategy for attitude stabilization. The ultrasonic range finder points downwards, directly measuring the altitude of the aerial vehicle during flight. The camera is installed on-board pointing

downwards, in a manner that the plane formed by the helicopter ( $B_x, B_y$ ) plane and the camera ( $C_x, C_y$ ) image plane are parallel and have the same orientation, see Figure 8.2. The camera and the aerial vehicle move together as a rigid body, therefore, by using the image provided by the camera, vision-based strategies could be implemented for estimating the vehicle states required during the navigation mission.

The present study addresses the problem of stabilizing the convertible MAV in 6 degrees of freedom (DOF) during a road-following task. Such a procedure can be detailed as follows. In the first stage, the vehicle performs an autonomous take-off, reaching the desired altitude  $z_d$  over the road. At this point, the heading of the vehicle, expressed by  $\psi$ , is driven to yield a parallel positioning between the helicopter  $x$ -axis (represented by  $B_x$ ) and the longitudinal direction of the road (expressed by  $R_x$ ). The helicopter forward speed, expressed by  $\dot{x}$ , is kept to a constant value while the distance between  $B_x$  and  $R_x$ , expressed by  ${}^B e_y$  (see Figure 8.2), is regulated and kept to a minimum value, achieving a flight path well aligned and centered w.r.t. the road.

The objective of this chapter is to design a road following strategy based on computer vision and switching controllers, with the purpose of stabilizing the vehicle altitude, heading angle, and lateral position ( $z$ ,  $\psi$ , and  $y$  states, respectively) w.r.t. the road, while traveling at constant forward speed ( $\dot{x} = c$ ). Overcoming external disturbances while performing the navigation task is also a subject of interest. In general, the traveling speed may be time-varying. However, for simplicity we will only consider here the case of constant speed.

### 8.3

## States estimation using a vision sensor

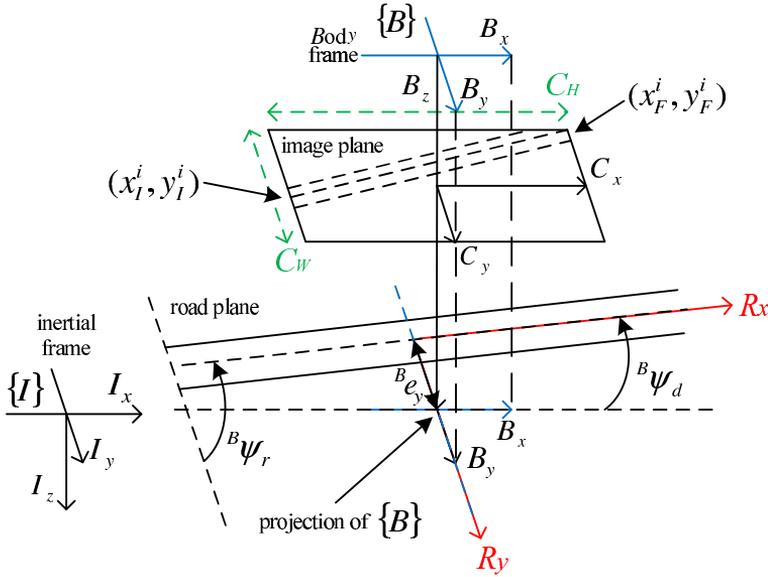
Suppose the camera-equipped convertible MAV is flying over a road composed by straight sections and smooth curves. If the distance between the road and aerial vehicle (altitude state) is appropriate, the road will be projected in the camera image plane as a group of straight lines, see Figure 8.2. A straight line in the image plane can be seen as a segment of infinite length, whose center of gravity belongs to the straight line [95]. A scheme of such a representation can be seen in Figure 8.3. By implementing the Hough transform method for line detection, a straight line can be represented as [96]:

$$\rho = x \cos \theta + y \sin \theta. \quad (8.1)$$

The center of gravity ( $x_g, y_g$ ) of each straight line detected can be computed as

$$x_g^i = \cos(\theta)\rho \quad ; \quad y_g^i = \sin(\theta)\rho \quad (8.2)$$

where the super-index  $i$  stands for line  $i$ . Notice that it is possible to assign initial and final bounds to the line. Let us define  $(x_I^i, y_I^i)$  as the initial point of the line, located in a place below the image lower margin. Similarly, let  $(x_F^i, y_F^i)$



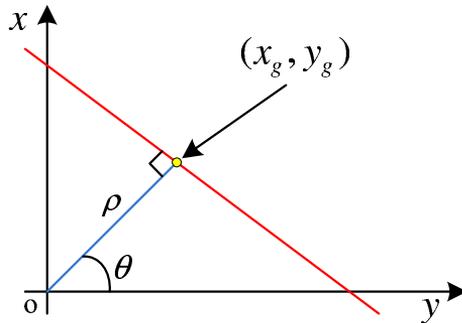
**Figure 8.2:** Scheme of the road following mission: while flying at an appropriate altitude, the road is projected onto the image plane as a group of straight lines.

be the final point of the line, located in a place above the image upper margin. If the line has a parameter  $\rho > 0$ , the coordinates  $(x_I^i, y_I^i)$  and  $(x_F^i, y_F^i)$  will be defined as

$$x_I^i = x_g + \Gamma(-\sin(\theta)) \quad ; \quad y_I^i = y_g + \Gamma \cos(\theta) \quad (8.3)$$

$$x_F^i = x_g - \Gamma(-\sin(\theta)) \quad ; \quad y_F^i = y_g - \Gamma \cos(\theta) \quad (8.4)$$

where  $\Gamma$  is a known constant defining the bounds of the line. In the case where  $\rho < 0$ , the point  $(x_F^i, y_F^i)$  will be computed as in equation (8.3), while the point  $(x_I^i, y_I^i)$  will be computed as in equation (8.4). The set of lines obtained from the projection of the road on the image are grouped together



**Figure 8.3:** Scheme of the Hough transform method for line detection.

with the purpose of obtaining just an average line. This average line will uniquely represent the road in the image with a single pair of initial and final coordinates

$$x_I = \frac{1}{n} \sum_{i=1}^n x_I^i \quad ; \quad y_I = \frac{1}{n} \sum_{i=1}^n y_I^i \quad (8.5)$$

$$x_F = \frac{1}{n} \sum_{i=1}^n x_F^i \quad ; \quad y_F = \frac{1}{n} \sum_{i=1}^n y_F^i \quad (8.6)$$

where  $n$  is the number of lines grouped together,  $(x_I, y_I)$  represents the initial (lowermost) road coordinate, and  $(x_F, y_F)$  represents the final (uppermost) road coordinate.

Wrong line detections may occur due to unexpected situations (e.g., changes in the scene illumination), therefore, a method for identifying false detections is required. In the developed algorithm, every line  $i$  detected must satisfy a safety condition in order to be considered as a *member* of the road. To achieve this, the final and initial coordinates of line  $i$  are compared with respect to the average initial and final coordinates computed in equation (8.5) and (8.6). If the coordinates of line  $i$  are far from the average values, line  $i$  is considered as a false detection and it is removed from further computations.

### 8.3.1 Computing the heading angle

The angle between the camera  $C_y$  axis and the line going from  $(x_I, y_I)$  to  $(x_F, y_F)$  can be computed using the coordinates  $(x_I, y_I)$  and  $(x_F, y_F)$  in the two argument arctangent function as follows

$${}^B\psi_r = \text{arctangent}(y_F - y_I, x_F - x_I). \quad (8.7)$$

The angle  ${}^B\psi_r$  is used for obtaining the desired heading angle  ${}^B\psi_d$  which will align the vehicle  $x$ -axis ( $B_x$ ) with the road longitudinal axis ( $R_x$ ), see Figure 8.2. The desired heading angle  ${}^B\psi_d$  can be expressed as

$${}^B\psi_d = {}^B\psi_r - \frac{\pi}{2}. \quad (8.8)$$

In equation (8.8) the term  $\frac{\pi}{2}$  is subtracted because the angle  ${}^B\psi_r$  calculated using equation (8.7) has its origin in the position corresponding to 3:00 hours on the clock. If  $\frac{\pi}{2}$  is not subtracted, the heading measurement obtained when the heading angle of the vehicle is aligned with the direction of the road would be equal to  $+90^\circ$ . With this subtraction we adjust such measurement to a value of  $0^\circ$ .

**8.3.2** Computing the relative lateral position

Consider an image-based distance  ${}^C e_y$  located between the road center of gravity projection  $(x_g, y_g)$  and the vehicle center of gravity projection  $(x_0, y_0) = (C_W/2, C_H/2)$ , where  $C_W$  and  $C_H$  represent the image width and height respectively, in pixel coordinates (see Figure 8.2). For the case, when  $x_I > x_F$ , one has

$${}^C e_y = \left( \frac{x_I - x_F}{2} + x_F \right) - \frac{C_W}{2}. \tag{8.9}$$

On the other hand, for the case when  $x_I < x_F$ ,  $x_I$  must be replaced by  $x_F$  and vice-versa. The lateral position of the aerial vehicle w.r.t. the road can be estimated from  ${}^C e_y$  as

$$e_y = z \frac{{}^C e_y}{\alpha_y} \tag{8.10}$$

where  $z$  represents the altitude of the rotorcraft w.r.t. the road, and  $\alpha_y$  represents the camera focal length, in terms of pixel dimensions, in the direction of  $C_y$ .

**8.3.3** Translational velocities

The vehicle translational velocities in the forward and lateral directions can be estimated by implementing an optical flow algorithm in the camera image. In our work, the algorithm of Shi and Tomasi [97] is used to identify the mesh of points allowing an efficient feature tracking, i.e., the best textured points. After this step, the selected features are used by the pyramidal implementation of the Lucas-Kanade algorithm [98] to estimate the optical flow.

Consider the camera-vehicle arrangement moving in a 3-dimensional space w.r.t. a rigid scene. The optical flow computed at an image point  $(x_i, y_i)$  comprises translational and rotational parts expressed as

$$\begin{bmatrix} OF_{xi} \\ OF_{yi} \end{bmatrix} = T_{OF}^i + R_{OF}^i \tag{8.11}$$

where the translational part is

$$T_{OF}^i = \frac{1}{z} \begin{bmatrix} -\alpha_x & 0 & x_i \\ 0 & -\alpha_y & y_i \end{bmatrix} \begin{bmatrix} \dot{x}^c \\ \dot{y}^c \\ \dot{z}^c \end{bmatrix} \tag{8.12}$$

and the rotational part is

$$R_{OF}^i = \begin{bmatrix} \frac{x_i y_i}{\alpha_x} & -(\alpha_x + \frac{(x_i)^2}{\alpha_x}) & y_i \\ (\alpha_y + \frac{(y_i)^2}{\alpha_y}) & -\frac{x_i y_i}{\alpha_y} & -x_i \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}. \quad (8.13)$$

The terms  $OF_{x_i}$  and  $OF_{y_i}$  are the optical flow components in the  $x$  and  $y$  coordinates, respectively, of the  $(x_i, y_i)$  feature. The terms  $(\dot{x}^c, \dot{y}^c, \dot{z}^c)$  and  $(w_x, w_y, w_z)$  represent the camera translational velocities and rotation rates, respectively, while  $\alpha_x$  and  $\alpha_y$  express the camera focal lengths.

During the optical flow computation process the  $(x_i, y_i)$  features share the same movement (rigid scene assumption). By using all the tracked features, the mean values for the optical flow in the forward and lateral direction can be expressed as

$$\overline{OF}_x = \overline{V}_{OF_x} + K_x \overline{V}_{OF_z} + \overline{R}_{OF_x} \quad (8.14)$$

$$\overline{OF}_y = \overline{V}_{OF_y} + K_y \overline{V}_{OF_z} + \overline{R}_{OF_y} \quad (8.15)$$

where  $\overline{OF}_x$  and  $\overline{OF}_y$  are the mean values of the optical flow (sensed in the image coordinate system),  $\overline{V}_{OF_z}$  represents the relative depth, and  $K_x$  and  $K_y$  are known scale factors depending on the intrinsic parameters of the camera. Applying a similar approach to the one presented in [92], the rotational optical flow terms  $\overline{R}_{OF_x}$  and  $\overline{R}_{OF_y}$  are compensated and the pseudo-speeds ( $\overline{V}_{OF_x}, \overline{V}_{OF_y}, \overline{V}_{OF_z}$ ) are deduced. This procedure yields

$$-z \frac{\overline{V}_{OF_x}}{\alpha_x} = \dot{x} \quad (8.16)$$

$$-z \frac{\overline{V}_{OF_y}}{\alpha_y} = \dot{y} \quad (8.17)$$

$$z \overline{V}_{OF_z} = \dot{z} \quad (8.18)$$

where the fact that the camera and the vehicle share the same movement has been used, i.e., the velocity of the camera is equal to the velocity of the vehicle, or  $(\dot{x}^c, \dot{y}^c, \dot{z}^c) = (\dot{x}, \dot{y}, \dot{z})$ .

### 8.3.4 Computing the heading and the lateral position

The computation of the desired heading angle and lateral position of the vehicle are compromised by the sensitivity of the Hough transform method for line detection. In addition to this, it is not possible to estimate such parameters when the vehicle is flying over a region where the road is out of the camera FOV. Then, a realistic scenario suggests that the image of the road can be temporarily lost by the onboard camera. For this reason, it is crucial to extend the results presented in Sections 8.3.1 and 8.3.2 with the main objective of

dealing with time events where the road is out of sight. Aiming at this goal, let us define a binary signal  $s : [0, \infty) \rightarrow \{0, 1\}$  as

$$s(t) := \begin{cases} 0 & \text{no line detection at time } t \\ 1 & \text{camera detects line at time } t. \end{cases} \quad (8.19)$$

In equation (8.19),  $s(t)$  allows switching between two different methods for computing the vehicle states. Furthermore, for a given binary signal  $s$  and  $t > \tau > 0$ , denote by  $T_s(\tau, t)$  the amount of time in the interval  $(\tau, t)$  for which  $s = 0$ . Formally,  $T_s(\tau, t) := \int_{\tau}^t (1 - s(l)) dl$ .

Let us consider first the desired heading angle  $\psi_d$ . Using equation (8.19) it is possible to define

$$\psi_s(t) = s(t)\psi_d(t) + (1 - s(t))\psi_{\text{IMU}}(T_s(\tau)) \quad (8.20)$$

where  $\psi_d(t)$  is obtained from equation (8.8),  $\psi_{\text{IMU}}(T_s(\tau))$  is the heading angle measured by the IMU at the time when the binary signal  $s$  changes from 0 to 1, and  $\psi_s(t)$  represents the desired heading angle that will be used in the helicopter control algorithm. Given the lack of visual information when  $s = 0$ , we decided to keep the yaw angle at a fixed value as long as the road is not seen. Equation (8.20) allows switching between: (i) a desired heading provided by the imaging algorithm, and (ii) a desired heading temporarily determined by the IMU heading measured at the time when the binary signal  $s$  changes from 1 to 0.

Consider now the computation of the lateral position of the vehicle when the road is out of sight. Equation (8.19) allows us to compute

$$e_{ys}(t) = s(t)e_y(t) + (1 - s(t)) \left( e_y(T_s(\tau)) + \int_{\tau}^t \dot{y}(t) dt \right) \quad (8.21)$$

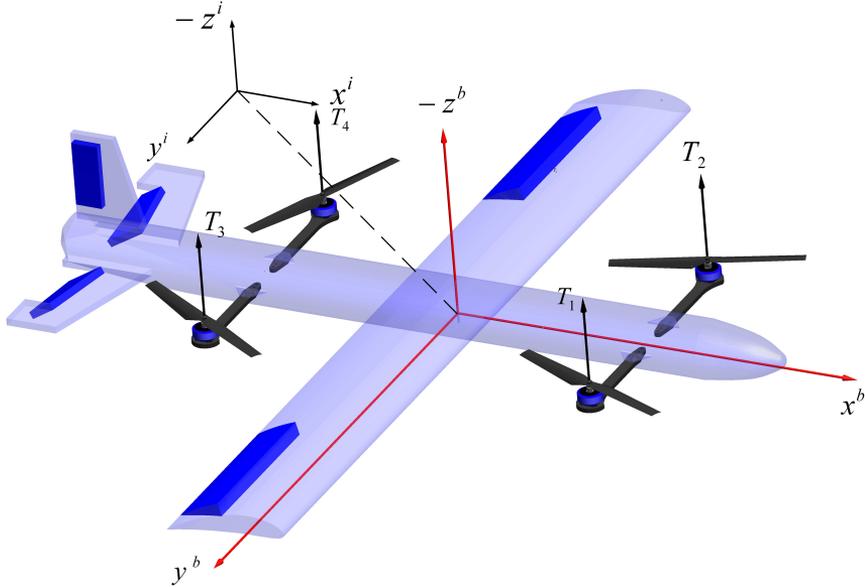
where  $e_y(t)$  is obtained as in equation (8.10),  $e_y(T_s(\tau))$  is the lateral position measurement at the time when the binary signal  $s$  changes from 1 to 0, and  $\dot{y}(t)$  represents the vehicle lateral velocity obtained from equation (8.17). Notice that equation (8.21) allows switching between two different approaches for estimating the vehicle lateral position w.r.t. the road. In the first approach (when  $s = 1$ ) the lateral position is directly obtained from the detection of the road in the camera image (equation (8.10)). The second approach (when  $s = 0$ ) makes use of the lateral position measurement at the time when the binary signal  $s$  changes from 1 to 0, in combination with the integral of the translational velocity  $\dot{y}$  during the time the road is not being detected by the vision algorithm, i.e., the amount of time in the interval  $(\tau, t)$  that  $s = 0$ .

## 8.4 Control Strategy

In this chapter, we will focus on the hover mode of the Quad-plane MAV. In this mode, our vehicle behaves like a conventional Quad-rotor, then we will use the Quad-rotor mathematical model [31].

$$\begin{aligned}
 m\ddot{x} &= u_1(\sin\psi\sin\phi + \cos\psi\sin\theta\cos\phi) - \frac{1}{2}C_xA_c\zeta\dot{x}|\dot{x}| \\
 m\ddot{y} &= u_1(\sin\psi\sin\theta\cos\phi - \cos\psi\sin\phi) - \frac{1}{2}C_yA_c\zeta\dot{y}|\dot{y}| \\
 m\ddot{z} &= mg - u_1(\cos\theta\cos\phi) \\
 \ddot{\theta} &= u_2 \\
 \ddot{\phi} &= u_3 \\
 \ddot{\psi} &= u_4
 \end{aligned} \tag{8.22}$$

where  $u_i$  are the control inputs,  $(x, y, z)$  are the position states and  $(\theta, \phi, \psi)$  are the Euler angles.  $C_x$ ,  $C_y$ ,  $A_c$  and  $\zeta$  are unknown parameters related to the friction force [99]. The corresponding coordinate system is represented in Figure 8.4. The following set of state variables are used:



**Figure 8.4:** CAD design of the Convertible MAV experimental platform.

$$\begin{array}{cccccc}
 x_1 = x & y_1 = y & z_1 = z & \theta_1 = \theta & \phi_1 = \phi & \psi_1 = \psi \\
 x_2 = \dot{x} & y_2 = \dot{y} & z_2 = \dot{z} & \theta_2 = \dot{\theta} & \phi_2 = \dot{\phi} & \psi_2 = \dot{\psi}
 \end{array}$$

**8.4.1** Operating modes for the road following mission

---

We divide the road following mission into four different stages which are described next:

- **Take-off mode.** The objective is to achieve the desired altitude  $z_d$ , while avoiding drift in the  $x$ - $y$  plane.
- **$\psi$ -alignment mode ( $\psi$  AL).** In this mode the vehicle has already achieved the desired altitude. The task to be accomplished here is to align the vehicle heading angle w.r.t. the road direction.
- **$y$ -alignment mode ( $y$  AL).** The vehicle is required to maintain a flying path well aligned w.r.t. the center of the road.
- **Tracking mode.** In this mode, the forward displacement of the vehicle is stabilized in order to navigate at constant velocity.
- **Landing mode.** When the vehicle has reached a previously defined position near to the end of the road, the altitude is controlled in order to perform an autonomous landing. In the current implementation of the algorithm, the end of the road is not detected autonomously, i.e., a signal for activating the landing mode is manually generated.

**8.4.2** Control Laws for each Operating Mode

---

The control strategy proposed in all the operating modes is based on the idea that the global system (8.22) is constituted of two subsystems, the attitude dynamics and the position dynamics, each one with a time-scale separation between them [44]. From this fact, it is possible to propose a hierarchical control scheme where the position controller outputs desired attitude angles  $\phi_d$ ,  $\theta_d$ ,  $\psi_d$  ( $\psi_d$  can be independently controlled) which are the angles to be tracked by the attitude controllers.

**8.4.2.1** Attitude Control

---

The integral sliding mode control is used for stabilizing the attitude dynamics of the Convertible MAV. The approach is explained for the roll dynamics, but the same procedure must be followed for generating the pitch and yaw dynamics. The error equation for the roll sub-system is defined as  $\tilde{\phi}_1 = \phi_1 - \phi_{1d}$ , and its

time derivative as  $\dot{\tilde{\phi}}_1 = \dot{\phi}_1 - \dot{\phi}_{1_d} = \tilde{\phi}_2$ . Let us choose the switching function defined in [100] as

$$\varphi(\phi, t) = \tilde{\phi}_1 + 2\lambda\tilde{\phi}_1 + \lambda^2 \int_0^t \tilde{\phi}_1(\tau) d\tau. \quad (8.23)$$

In (8.23) the parameter  $\lambda$  is the slope of the sliding line, which should accomplish  $\lambda > 0$  to ensure the asymptotic stability of the sliding mode. The time derivative of (8.23) can be calculated as  $\dot{\varphi} = u_3 + 2\lambda\tilde{\phi}_2 + \lambda^2\tilde{\phi}_1$ , and from the sliding mode condition  $\dot{\varphi} = 0$ , we find the equivalent control

$$u_3 = -2\lambda\tilde{\phi}_2 - \lambda^2\tilde{\phi}_1. \quad (8.24)$$

For obtaining a control law such that  $\tilde{\phi}_1$  remains on the sliding surface  $\varphi(\phi, t) = 0$ ,  $\forall t > 0$ , a candidate Lyapunov function  $v(\varphi) = \frac{1}{2}\varphi^2$  is proposed. The condition for the stability of the roll sub-system is satisfied if we can ensure that the condition  $\dot{v}(\varphi) = \frac{1}{2}\frac{d}{dt}\varphi^2 \leq \eta|\varphi|$  holds for  $\eta \geq 0$ . Thus, the system remains on the sliding surface and the states go to the origin. Then  $\varphi\dot{\varphi} \leq -\eta|\varphi|$  and the controller must be chosen in a way that  $\phi_1 = u_3 - K \text{sign}(\varphi)$  where  $K > 0$ .

#### 8.4.2.2 Position Control

The lateral position stabilization makes use of two PID controllers, one for the case when the road is being detected and the other for the case when it is not. Although both PID controllers are similar, they are designed having different parameters. When the road is inside the FOV, the gains of the control algorithm are tuned to behave as a PD controller since the vehicle needs to regulate its  $y$  coordinate at a minimum value ( $y \approx 0$ ) as fast as possible [93]. On the other hand, when the camera loses sight of the road, a switching strategy allows using a different method for measuring the vehicle  $\psi$  angle and  $y$  position. Additionally, the parameters of the positioning controller switch to alternative values emulating a PI controller. In both cases the control objective is to regulate the  $y_1$  state to the origin, i.e.  $y_{1d} = 0$ . Thus, we proceed to develop a control law for the  $y$ -position.

In (8.22) we can note that the motion along  $y$ -axis is related to the  $\phi$  angle, thus one can design a PID controller to manipulate the  $\phi$  angle in order to control  $y$  motions. Setting  $\theta = \psi = 0$  and  $u_1 = 1$  from (8.22) the term  $\ddot{y}$  results in

$$\ddot{y} = -u_1 \sin \phi - p\dot{y}^2 = -k_{L3}y_1 - k_{L4}y_2 - k_{LI}\xi \quad (8.25)$$

where  $p\dot{y}^2 = \frac{1}{2}C_y A_c \zeta |\dot{y}|$  and  $p$  is a unknown parameter to be estimated;  $k_{L3}$ ,  $k_{L4}$  and  $k_{LI}$  are positive real numbers. Here we have introduced the additional state  $\xi$ , where  $\xi$  dynamics are given by  $\dot{\xi} = y_{1d} - y_1 = -y_1$ .

From (8.25) the desired angle  $\phi_d$  can be written as

$$\phi_d = \arcsin(k_{L3}y_1 + k_{L4}y_2 + k_{LI}\xi - p\dot{y}^2). \quad (8.26)$$

By taking the time derivative of (8.26) one obtains

$$\dot{\phi}_d = \frac{k_{L3}\dot{y} + k_{L4}\ddot{y} + k_{LI}\dot{\xi} - 2p\dot{y}\ddot{y}}{\sqrt{1-v}} \quad (8.27)$$

where  $v$  is a function of  $(p, y, \dot{y})$ . Proceeding in the same way one can obtain  $\theta_d$  and its time derivative  $\dot{\theta}_d$  required for the controller  $u_2$ , which can be generated with the procedure shown in section 8.4.2.1.

The altitude, can be controlled by the following controller

$$u_1 = \frac{g + k_{pz}(z_1 - z_{1d}) + k_{vz}(z_2 - z_{2d})}{\cos \theta \cos \phi} \quad (8.28)$$

where  $k_{pz}$  and  $k_{vz}$  are positive real numbers.

---

**8.4.2.3** Discontinuous Adaptation for  $y$  Dynamics

---

In order to apply the virtual control law (8.26) on the dynamics of  $y$ , we need to estimate the unknown parameter  $p$ . To achieve this goal, the key idea is to split the set of parameters into small subsets and design a controller for each of them. Then, after a dwell-time period one needs to check if the derivative of the Lyapunov function does not satisfy a certain inequality, if this is the case, one needs to switch to another controller and exclude the previous one from the indexed controls. As long as the inequality is satisfied, the current controller is maintained in the loop. The procedure is explained as follows.

We assume that the parameter  $p$  belongs to a known, relatively large, compact set  $\mathcal{P}$ , which is partitioned into smaller subsets. Defining the compact set  $\mathcal{P}$  as

$$p \in \mathcal{P} = \bigcup_{i=1}^{21} p^{(i)} = \{-1, -0.9, -0.8, \dots, 0.8, 0.9, 1\}. \quad (8.29)$$

With the controller previously designed (8.26), the closed-loop system (8.25), (8.26) leads to

$$\begin{aligned} \dot{y}_1 &= y_2 \\ \dot{y}_2 &= -k_{L3}y_1 - k_{L4}y_2 - k_{LI}\xi \\ \dot{\xi} &= -y_1. \end{aligned} \quad (8.30)$$

Considering  $k_{LI} = 0$ , or equivalently, the case where we use a PD controller in (8.25), the Lyapunov function for the system (8.30), can be taken as

$$V(y_1, y_2) = \sqrt{k_{L3}} \left( 1 + \frac{k_{L4}}{2\sqrt{k_{L3}}} \right) y_1^2 + y_1 y_2 + \frac{y_2^2}{\sqrt{k_{L3}}} \quad (8.31)$$

whose derivative with respect to (8.25) is given by

$$\dot{V}(y_1, y_2) = -k_{L3}y_1^2 - \left( \frac{2k_{L4}}{\sqrt{k_{L3}}} - 1 \right) y_2^2 = -W(y_1, y_2) \quad (8.32)$$

where  $\frac{2k_{L4}}{\sqrt{k_{L3}}} > 1$ .

To choose the correct candidate controller or equivalently, the next suitable index  $i \in \mathcal{I}$ , we take such index as

$$i = \arg \min_{j \in \mathcal{I}} \left\{ J(p^{(j)}) \right\} \quad (8.33)$$

where the performance index can be taken as

$$J(p^{(j)}) = \left| \dot{y}_2 + \sin \phi + p^{(j)} y_2^2 \right| \quad (8.34)$$

For each set  $\mathcal{P}^j$  we choose a nominal parameter (for example 1) and assuming that the sets  $\mathcal{P}^j$  are small, it is reasonable to expect  $|J(p) - J(p^{(j)})|$  to be small  $\forall p \in \mathcal{P}^j$ , therefore we use  $J(p^{(j)})$  as an index for the set  $\mathcal{P}^j$ . Thus we proceed with  $i = 1$ , and we put the first candidate virtual controller (8.26) as follows

$$\phi_d = \text{Sat} \left( \arcsin(k_{L3}y_1 + k_{L4}y_2 - p^{(i)}y_2^2) \right) \quad (8.35)$$

where  $\text{Sat}(\cdot)$  is a smooth saturation function which switch to another one as soon as the dwell-time period  $\tau$  is over and the inequality

$$\frac{\partial V(y_1, y_2)}{\partial y_1} y_2 + \frac{\partial V(y_1, y_2)}{\partial y_2} \dot{y}_2 \leq -W(y_1, y_2) \quad (8.36)$$

fails.

#### 8.4.2.4 Switching control for $y$ -dynamics

In this section we prove the stability in each region as well as in switching boundaries, i.e., when the gains of (8.25) switch to alternative values. The closed-loop system of the  $y$  dynamics (8.30) can be represented by  $\dot{e}_y = A_L e_y$  where  $e_y = (y_1, y_2, \xi)^T$  and

$$A_L = \begin{pmatrix} 0 & 1 & 0 \\ -k_{L3} & -k_{L4} & k_{LI} \\ -1 & 0 & 0 \end{pmatrix} \quad (8.37)$$

The control schema when the line is not detected proposes the same structure given by (8.30), with the difference of having a set of alternative gains. Thus,

the closed-loop system is represented by  $\dot{e}_y = A_{NL}e_y$ , where

$$A_{NL} = \begin{pmatrix} 0 & 1 & 0 \\ -k_{NL3} & -k_{NL4} & k_{NLI} \\ -1 & 0 & 0 \end{pmatrix} \quad (8.38)$$

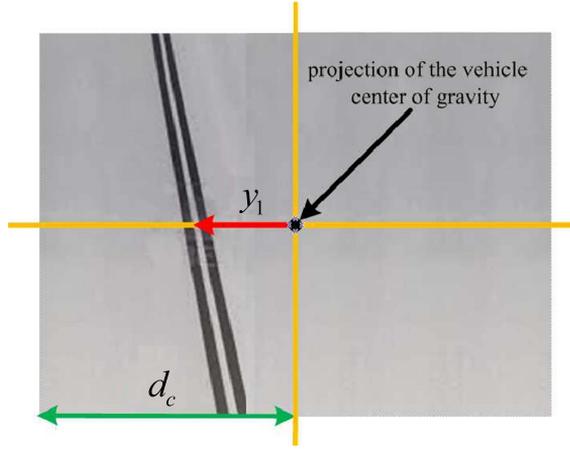
**Remark 11.** *The controller parameters of both operational regions were adjusted independently following an heuristic approach consisting of two main steps which are described next. During the first step, special care is taken to stabilize the vehicle fast dynamics, which corresponds to the angular behavior. The parameters associated with the angular rate are adjusted first, until an appropriate angular rate response is obtained. Following a similar method, the next parameter to be adjusted corresponds to the attitude control parameter. After both parameters have been adjusted properly, an autonomous hover flight experiment is performed to verify the effectiveness of the tuning procedure. The second part of the adjustment procedure is devoted to stabilizing the vehicle translational dynamics, which involves the implementation of visual feedback in the control strategy. First, the parameter corresponding to the translational velocity is adjusted until the vehicle translational drift is almost eliminated. Next, the parameter associated with the translational position is tuned to obtain an appropriate behavior of the controller for stabilizing the 3-dimensional position of the vehicle over road area. The main objective behind the procedure just described is to minimize the tracking error, in order to obtain the most appropriate angular and translational behavior. For the first case (road detected), the gains were selected as  $k_{L3} = 1800$ ,  $k_{L4} = 650$  and  $k_{LI} = 1$ , while in the second case (no road detected) as  $k_{NL3} = 1800$ ,  $k_{NL4} = 1$  and  $k_{NLI} = 20$ . In this way, the system presents a switched-system characteristic.*

### 8.4.3 Stability Analysis of the Lateral Position Control

We now present a study concerning the stability of the system across switching boundaries. In fact, it is possible to find a common Lyapunov function for the closed-loop system of the two controllers proposed for the lateral dynamics [101]. Following such approach, similar pole locations have to be chosen for both cases, i.e., when the road is detected and when it is not. However, this is not the present situation since different gain values are being applied.

Let  $d_c$  be defined as the distance measured from the vehicle center of gravity projection (in the road plane) to the point where the camera loses the image of the road (see Figure 8.5). From this, a state-dependent switched linear system can be defined, which will be given by the closed-loop system (8.37)-(8.38), together with the following switching conditions

$$\dot{e}_y = \begin{cases} A_{NL}e_y & \text{if } y_1 \leq -d_c \\ A_L e_y & \text{if } -d_c < y_1 < d_c \\ A_{NL}e_y & \text{if } y_1 \geq d_c \end{cases} \quad (8.39)$$



**Figure 8.5:** The distance measured from the vehicle center of gravity projection in the image plane to the point where the camera loses the image of the road is defined as  $d_c$ . The distance between the vehicle center of gravity projection and the road is defined as  $y_1$ .

A switching event can occur in system (8.39) every time the trajectory crosses a *switching surface*. For the case under consideration, the switching surfaces are encountered when the system changes from  $\dot{e}_y = A_L e_y$  to  $\dot{e}_y = A_{NL} e_y$ , and vice-versa. In general, there is no need to associate each subsystem in (8.39) with a global Lyapunov function. In fact, it is enough to require that each candidate Lyapunov function  $V_i$  (where  $i = \{L, NL\}$ ) decreases along solutions of the  $i$ -th subsystem in the region where this system is active.

The stability of the overall system (8.39) is accomplished when: (i) individual subsystems are stable, and (ii) the overall system remains stable when a switching surface is crossed. Indeed, it is possible to verify that individual subsystems of the switching system (8.39) are globally exponentially stable (GES). Furthermore, if the switching between individual subsystems is sufficiently slow, the entire system (8.39) will be globally exponentially stable [102]. Aiming at this goal, let us define a dwell time  $\tau_d$  satisfying  $t_{n+1} - t_n \geq \tau_d$ , where  $t_n$  (for  $n = 1, 2, 3, \dots$ ) represents a switching time. Next, it is required to investigate the lower bound on  $\tau_d$  from the exponential decay bounds of the individual subsystems. This procedure is explained as follows.

As previously mentioned, individual subsystems are GES, then there exist Lyapunov functions  $V_L, V_{NL}$  which, for some positive constants  $a_i, b_i$ , and  $c_i$  satisfy [58]

$$a_i \|e_y\|^2 \leq V_i(e_y) \leq b_i \|e_y\|^2 \quad (8.40)$$

and

$$\frac{\partial V_i}{\partial e_y} A_i e_y \leq -c_i \|e_y\|^2 \quad (8.41)$$

for  $i = \{L, NL\}$ . Combining (8.40) and (8.41) we have

$$\frac{\partial V_i}{\partial e_y} A_i e_y \leq -\frac{c_i}{b_i} V_i(e_y). \quad (8.42)$$

This implies that

$$V_i(e_y(t_0 + \tau_d)) \leq e^{-\frac{c_i}{b_i} \tau_d} V_i(e_y(t_0)). \quad (8.43)$$

Let us consider two switching times  $\{t_1, t_2\}$ , and let us also assume that subsystem  $\dot{e}_y = A_L e_y$  is active on the interval  $t \in [t_0, t_1)$ , while subsystem  $\dot{e}_y = A_{NL} e_y$  is active on  $t \in [t_1, t_2)$ . From inequalities (8.40)-(8.43) it follows that

$$V_L(t_2) \leq \frac{b_L}{a_{NL}} V_{NL}(t_2) \leq \frac{b_L}{a_{NL}} e^{-\frac{c_{NL}}{b_{NL}} \tau_d} V_{NL}(t_1) \quad (8.44)$$

and also

$$V_L(t_2) \leq \frac{b_{NL} b_L}{a_L a_{NL}} e^{-\left(\frac{c_L}{b_L} + \frac{c_{NL}}{b_{NL}}\right) \tau_d} V_L(t_0). \quad (8.45)$$

In order to find the lower bound on  $\tau_d$  it is sufficient to ensure that (see Theorem 1 of [103])

$$V_L(t_2) - V_L(t_0) \leq -\sigma \|e_y(t_0)\|^2 \quad (8.46)$$

where  $\sigma > 0$ . It is now straightforward to compute an explicit lower bound on  $\tau_d$ , which guarantees that the switched system (8.39) is GES [101]

$$\tau_d > \frac{b_L + b_{NL}}{c_L + c_{NL}} \log \frac{b_L b_{NL}}{a_L a_{NL}}. \quad (8.47)$$

The lower bound on  $\tau_d$  can be satisfied by adjusting  $a_i$ ,  $b_i$ , and  $c_i$ , for  $i = \{L, NL\}$ , which depend on the controller gains and the Lyapunov function.

## 8.5 Experimental platform

### 8.5.1 The convertible MAV experimental vehicle

The convertible MAV used during the real time application is shown in Figure 8.6. It has been built using a group of commercially available components. The vehicle's fuselage and the H-form structure are built of carbon fiber and balsa wood. The wing was built with depron. Four counter rotating and tilting brushless motors provide the thrust, while the tilting mechanism of the engines is controlled through two analog servomotors. Motors are BL-Outrunner from Robbe ROXXY, which are driven by BiCtrl I2C electronic speed controllers.



**Figure 8.6:** Experimental platform: The Quad-plane Convertible MAV equipped with imaging, inertial and altitude sensing systems.

The weight of the rotorcraft is 1.1 kg. It has a 11.1V - 6000 mAh LiPo battery, allowing an autonomy of about 15 minutes. The onboard electronics are based on a IGEPv2 card, equipped with a Texas Instruments DM3730 System On Chip (SOC). The SOC benefits from having an ARM CortexA8 core running at 1 GHz, and a C64x+ DSP core running at 800 MHz. The ARM core allows execution of Linux, as well as its real-time extension Xenomai. The control law is executed in real-time at a 100 Hz frequency. The convertible MAV sensor suit consists of the next group of components. Inertial measurements are provided at 100 Hz by means of a 3DMGX3-25 IMU from Microstrain<sup>®</sup>. A SRF10 ultrasonic range finder provides the vehicle altitude at 50 Hz in a range between 0 m and 2 m. All the previously mentioned components are fully embedded onboard the vehicle.

### 8.5.2 Embedded vision system

The vehicle is equipped with a PlayStation<sup>®</sup> Eye camera whose focal length is  $(f_x, f_y) = (270.87, 271.76)$  pixels. The camera is capable of providing 120 images per second, at a resolution of  $320 \times 240$  pixels. The camera points downwards which allows observing the scene below the vehicle. The images provided by the camera are processed by computer vision algorithms in order to estimate the helicopter translational velocity in the  $x - y$  plane, the heading angle, as well as the lateral position w.r.t. the road direction. The translational velocity in the  $x - y$  plane is obtained from an optical flow algorithm, which is based on the pyramidal Lucas-Kanade method. For this purpose, the algorithm uses two pyramid levels, searching for 64 characteristic features in the image. A Harris affine region detector was implemented to perform the characteristic features detection.

The computation of the heading angle and lateral position of the vehicle is based on a Hough transform technique for line detection. With the objective of executing the Hough transform computation in the on-board DSP, the Hough's accumulator must fit in the DSP cache, which is possible if the resolution of the  $\rho$  and  $\theta$  parameters from equation (8.1) are properly selected. Based on the chosen resolution, a lookup table containing the different values of sines and cosines is generated and stored in the DSP cache. The image pixels are read in groups of 8 (1 octet) with the purpose of effectively filling the accumulator. It is worth mentioning that reading 1 octet or a group of 8 octets requires the same number of cycles for the DSP. Then, the computations of the Hough transform are optimized for a proper execution in the DSP, which is capable of performing several additions, subtractions, multiplications and scalar products at the same time. The embedded DSP allows computing the optical flow (translational velocities) and Hough transform (heading angle and lateral position) on-board at a 100 Hz frequency. A raw image obtained from the onboard camera while the vehicle is flying over the road is shown in Figure 8.7 (a). Gray scale and black and white images obtained after preprocessing steps are shown in Figure 8.7 (b) and Figure 8.7 (c), respectively. Specifically, the grayscale image is used by the optical flow algorithm, while the black and white image is used by the algorithm for line detection. Finally, the post-processed image where the road detected has been highlighted is shown in 8.7 (d).

---

### 8.5.3 Ground station

The rotorcraft is wirelessly linked to a ground station PC, where a graphical user interface (GUI) based on the QT application allows monitoring and controlling the vehicle. The user can visualize, in real time, graphics representing the measurements from the on-board sensors, as well as graphics representing the control law computation. It is also possible to modify (online) the parameters of the control laws, as well as the implemented filters. The rotorcraft can also be manually controlled by a human pilot using a PlayStation<sup>®</sup>3 joystick, which is linked to the ground station by means of a Bluetooth<sup>™</sup> link.

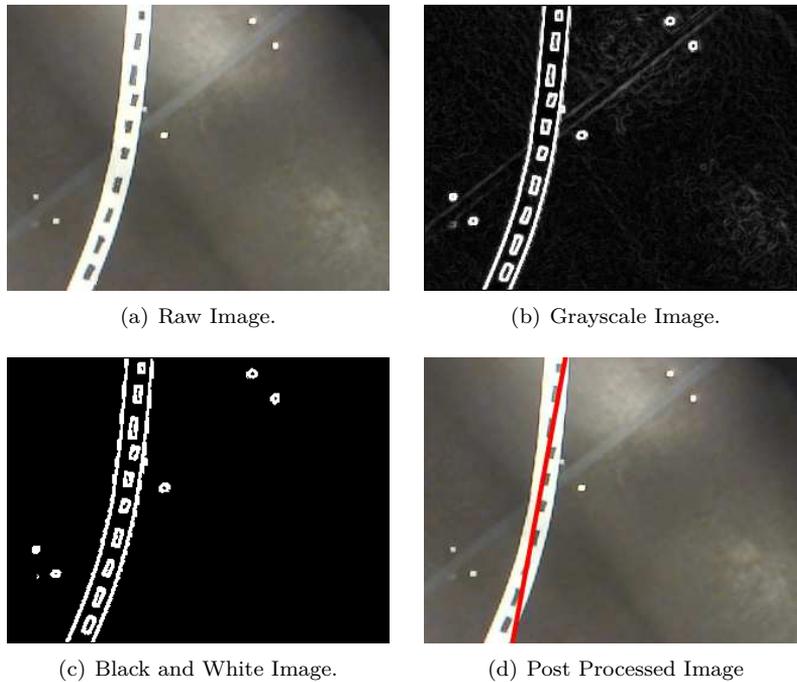
---

## 8.6 Simulations and Real Time Experiments

---

### 8.6.1 Numerical Simulations

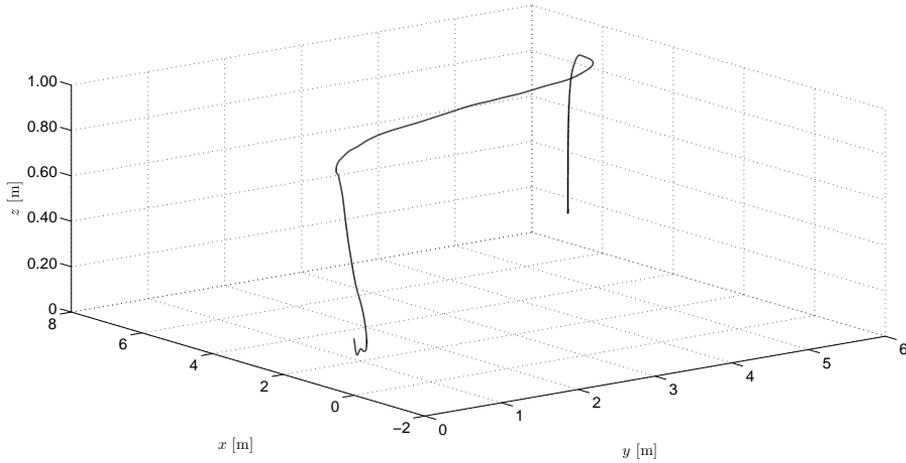
The control strategy developed in Section 8.4 has been designed for the convertible MAV model in equation (8.22). In order to show the effectiveness of the



**Figure 8.7:** Image obtained from the onboard camera while the MAV is flying above the road: (a) the raw image provided by the camera; (b) gray scale image used for optical flow computation; (c) black and white images used in Hough transform algorithm; (d) the post-processed image where the detection of the road has been highlighted.

obtained controller, we have performed a set of numerical simulations. The simulation scenario follows the procedure previously explained in Subsection 8.4.1. A synthetic road having two smooth turns has been designed, and was used as the desired reference that the vehicle has to follow during navigation. The desired altitude is set at 0.70 m over the road. A ramp profile was used for modifying the desired velocity from  $\dot{x} = 0.0$  m/s to  $\dot{x} = 0.3$  m/s during the first stages of the navigation, and also for modifying the desired velocity from  $\dot{x} = 0.3$  m/s to  $\dot{x} = 0.0$  m/s before landing.

As can be seen in the set of set of Figures 8.8-8.12, the controller performs well in spite of the non-linearities of the plant. Figure 8.10 shows a disturbance in the  $y$  state reference signal (blue line) at  $t = 18$  seconds. Notice how the proposed controller maintains stability of the system even in the presence of these unexpected behaviors. From Figure 8.12 it can be observed that the pitch and roll angles are always kept at very small values, and also that the induced disturbance produces only small effects on them.



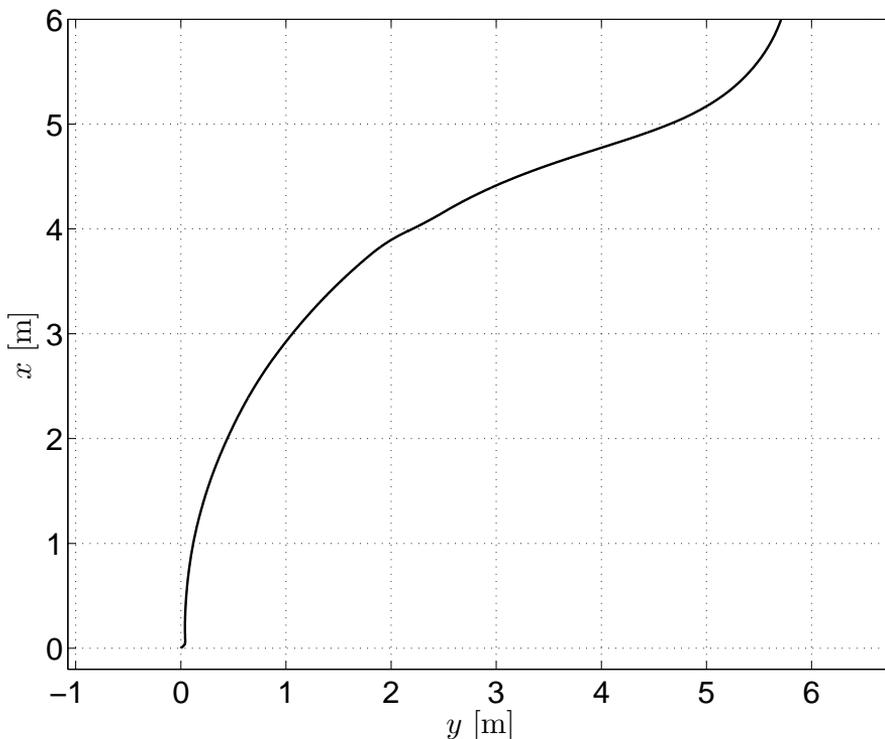
**Figure 8.8:** Simulation of the road-following mission: The 3-dimensional trajectory flown by the vehicle. The black line represents the vehicle displacement.

**8.6.2** Real Time Experiments

The ultimate test for the proposed estimation and control strategies consists on verifying its proper operation when used during real-time experiments. Aiming at this goal, a road model having two smooth curves was built and installed in our experimental area. The experimental procedure can be described as follows.

The rotorcraft starts its mission on the ground, exactly positioned over the road. Next, an autonomous take-off is performed, achieving a desired altitude of 0.70 m over the road. Once at this height, the embedded camera detects the road. The imaging information allows heading and lateral position stabilization for achieving a flight path well aligned w.r.t. the road. The vehicle is then required to navigate forward at a desired speed of  $\dot{x} = 0.3$  m/s, while regulating its heading angle  $\psi$  and lateral position  $y$  w.r.t. the road. In order to achieve a smooth transition from a forward velocity of  $\dot{x} = 0.0$  m/s to a forward velocity of  $\dot{x} = 0.3$  m/s, a ramp velocity profile is implemented, in such a way that the velocity slowly increases after the take-off task. Once the vehicle is near to the end of the road segment, the landing is performed autonomously, and the rotorcraft descends slowly to the ground. Again, a ramp velocity profile is implemented, in such a way that the forward velocity slowly decreases from  $\dot{x} = 0.3$  m/s to  $\dot{x} = 0.0$  m/s before the landing task is engaged.

During the execution of the autonomous navigation experiment, an external force (disturbance) was applied in the lateral dynamics of the vehicle. The disturbance coerces the convertible MAV to displace arbitrarily in the  $y$  direction, making the imaging system unable to detect the road for several instants of time. Specifically, the external perturbation has the purpose of demonstrating

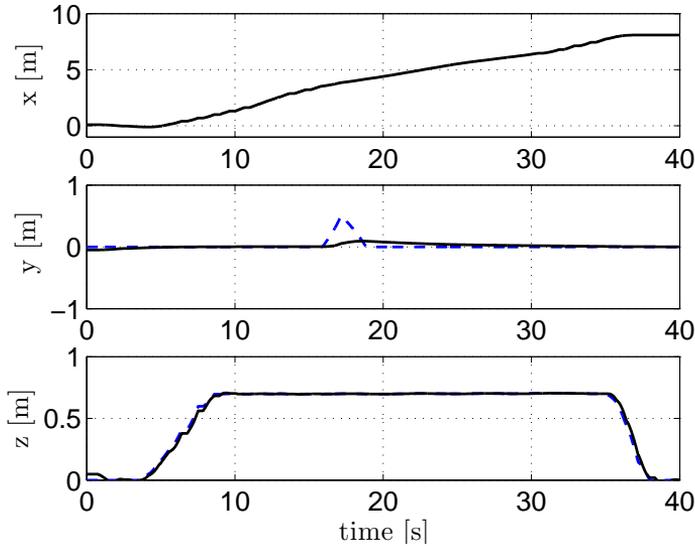


**Figure 8.9:** A projection on the  $(x, y)$  plane of the trajectory flown by the rotorcraft.

the effectiveness of the estimation and control strategies when the switching procedure occurs. It is worth mentioning that, during the experimental application, the maximum time that the road stayed out of sight was around 3 seconds.

A 3-dimensional reconstruction of the path navigated by the rotorcraft during the real-time experiment is shown in Figure 8.13. In this Figure, the rotorcraft trajectory is represented w.r.t. an inertial coordinate frame  $\mathcal{I}$ , whose origin is located at the convertible MAV initial position. It can be seen that the vehicle performs the road following mission inside a rectangular area of  $6 \times 6$  meters. The projection of the rotorcraft displacement in the  $(x, y)$  plane can be seen in Figure 8.14. It is worth mentioning that the forward displacement (relative to the body fixed frame) is obtained by integrating the optical flow measured in the forward direction. On the other hand, the rotorcraft lateral position is obtained from two methods. The first of them directly applies equation (8.10), which provides the lateral position of the vehicle relative to the road. The second method corresponds to the integration of the optical flow, in the lateral direction, during the instants of time when the road is not detected by the vision system.

The inertial coordinates shown in Figure 8.13 and Figure 8.14 are obtained after rotating both the  $(x, y)$  states around the  $z$ -axis using a gymbal like



**Figure 8.10:** The  $(x, y, z)$  states during the simulation test. The solid black line represents the behavior of the rotorcraft. Notice how the system remains stable despite a disturbance in the  $y$  state reference signal (dashed blue line) at  $t = 18$  seconds. This demonstrates that the proposed controller is effective for dealing with these unexpected behaviors.

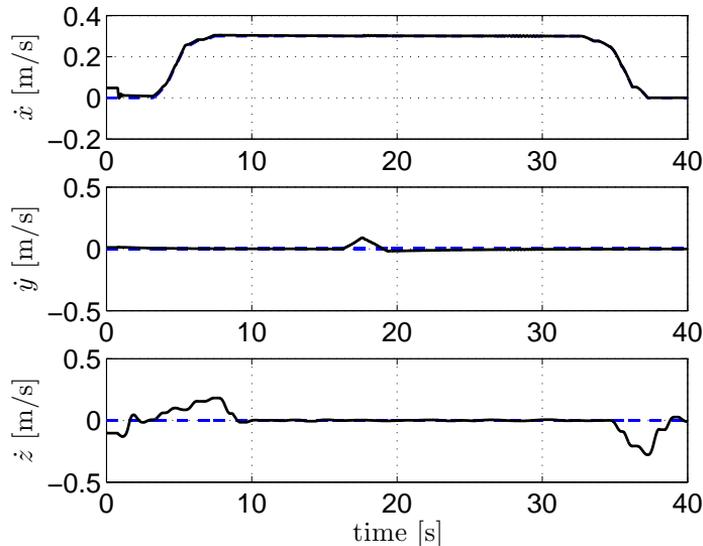
rotation matrix. Specifically, the rotation matrix makes use of the heading angle provided by the onboard IMU. The computation, which is described in equations (8.48) and (8.49), generates the 3-dimensional trajectory of the vehicle with respect to the inertial frame  $\mathcal{I}$ .

$$\mathcal{I}x = \int_0^t \cos(\psi(t))\dot{x}(t)dt - \sin(\psi(t))y(t) \tag{8.48}$$

$$\mathcal{I}y = \int_0^t \sin(\psi(t))\dot{x}(t)dt + \cos(\psi(t))y(t) \tag{8.49}$$

The three translational states are shown individually in Figure 8.15. Here,  $x = 10$  m represents the longitudinal dimension of the road, which is also the total distance flown by the convertible MAV in the forward direction. This measurement was based on the integration of the optical flow sensed by the camera in the  $x$  direction. The  $y$  state represents the lateral position of the convertible MAV w.r.t. the road. An external perturbation in the  $y$  direction can be seen at  $t = 25$  seconds. The rotorcraft translational velocities are shown in Figure 8.16. Notice that, during navigation, the forward speed  $\dot{x}$  is maintained at 0.3 m/s, while the lateral  $\dot{y}$  and altitude  $\dot{z}$  velocities are kept near zero.

The switching between the two different state estimation approaches is illustrated in Figure 8.17. The variable  $s$  (the graphic in the middle) shows the



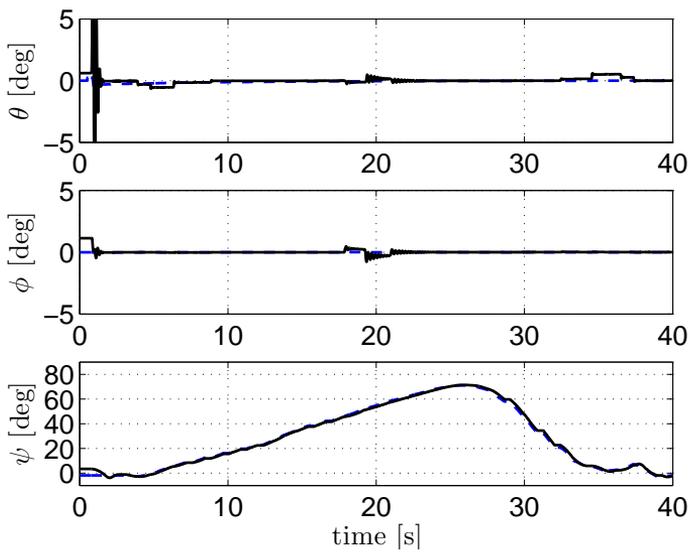
**Figure 8.11:** Translational velocities during the simulation test. The solid black line represents the behavior of the rotorcraft. The forward velocity  $\dot{x}$  is kept at 0.3 m/s, while the lateral velocity  $\dot{y}$  and altitude velocity  $\dot{z}$  are both kept around zero. Ramp profiles were used for increasing-decreasing the  $\dot{x}$  reference value. The variation in  $\dot{y}$  at  $t \approx 18$  s is caused by the induced disturbance.

instants of time when the road is being detected ( $s = 1$ ) and when it is not being detected ( $s = 0$ ). The upper graphic represents the desired heading angle that will align the rotorcraft heading with the road direction, as obtained from equation (8.20). The lower graphic represents the lateral position error as computed from equation (8.21). A zoom to the region where the switching occurs is shown in Figure 8.18.

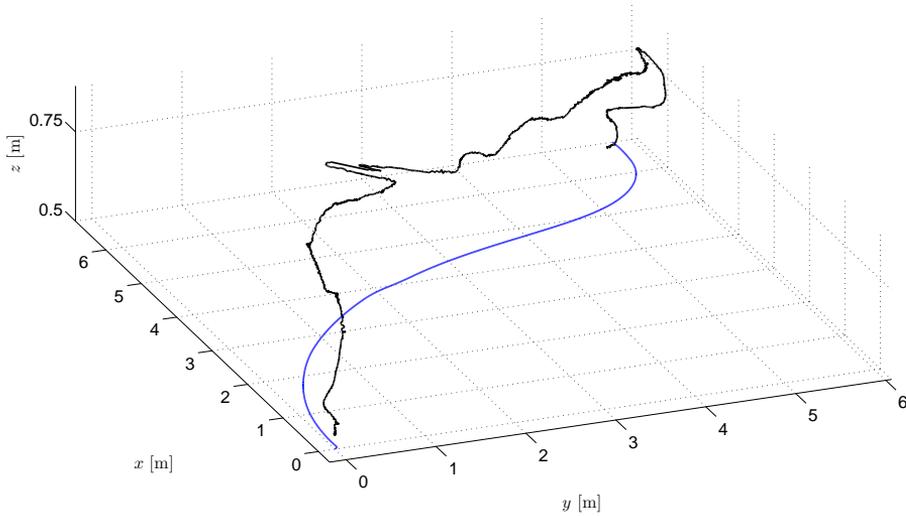
Finally, the rotorcraft Euler angles, as obtained directly from the embedded IMU are shown in Figure 8.19. The pitch angle ( $\theta$ ), shown in the upper graphic, is kept to a small value which ensures forward displacement at the required velocity. The roll angle ( $\phi$ ), shown in the middle graphic, is also kept small. However, when it is required to correct the lateral position error induced at around  $t = 25$  seconds, the roll magnitude increases until reaching again the desired lateral position w.r.t. the road. The lower graphic shows the heading angle ( $\psi$ ) described by the rotorcraft in order to handle the two curves of the road. This heading angle is represented w.r.t. the inertial reference frame whose origin is at the convertible MAV initial position.

Figure 8.20 shows an image of the rotorcraft in the experimental area during the real-time tests. In addition, a video showing the convertible MAV while performing the road following experiment can be seen at

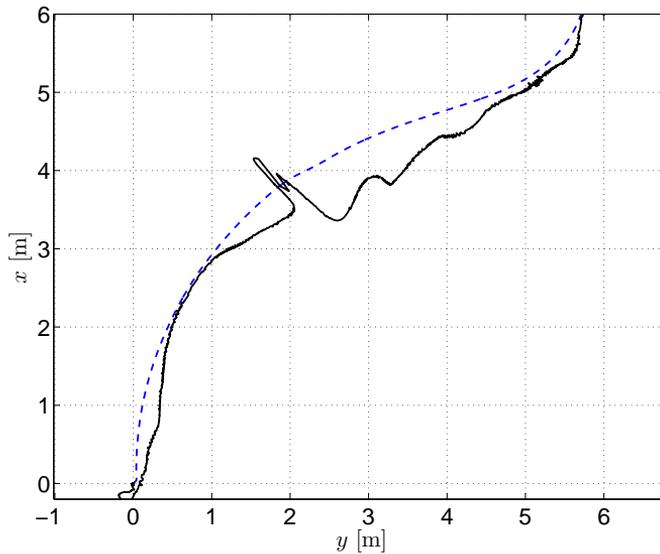
<http://youtu.be/fAQvSXPAG5Q>



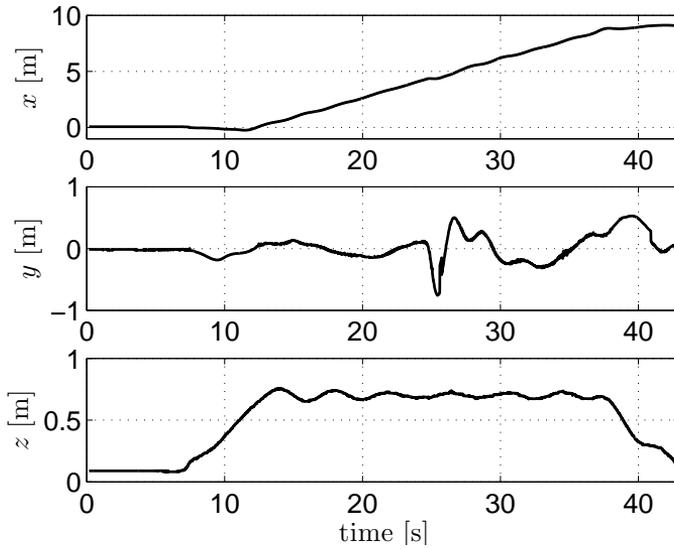
**Figure 8.12:** The rotorcraft Euler angles during the simulation test. The solid black line represents the behavior of the rotorcraft. Notice that the pitch angle ( $\theta$ ) and roll angle ( $\phi$ ) are both kept small. The heading angle ( $\psi$ ) is represented w.r.t. the inertial reference, and describes the heading that the vehicle must track in order to handle the two curves of the road. From the ( $\psi$ ) graphic, notice that the difference between the actual behavior (black line) and expected behavior (dashed blue line) is very small.



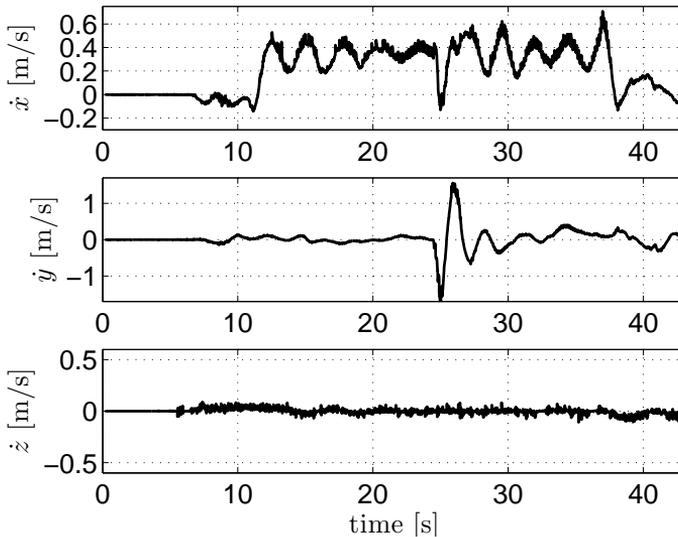
**Figure 8.13:** Road-following mission: 3-dimensional reconstruction of the path navigated during the real-time experiment. The dashed blue line plotted in the  $x$ - $y$  plane represents the position of the road model. The black line represents the actual vehicle displacement.



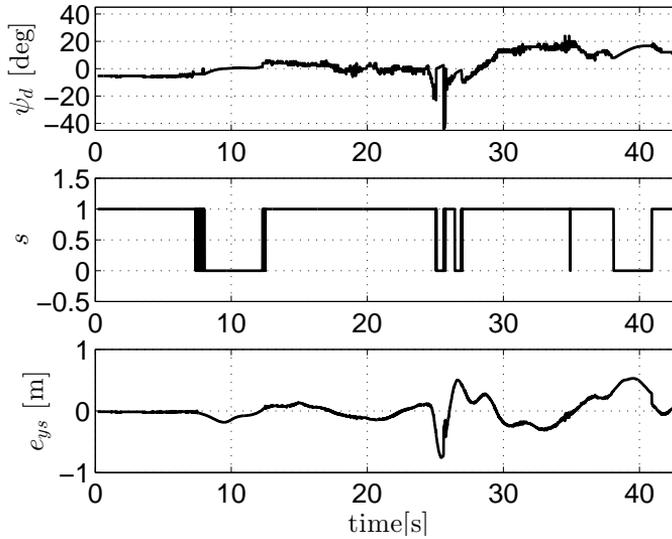
**Figure 8.14:** The trajectory described by the rotorcraft, projected in the  $(x, y)$  plane. The dashed blue line represents the position of the road model. The black line represents the actual vehicle displacement.



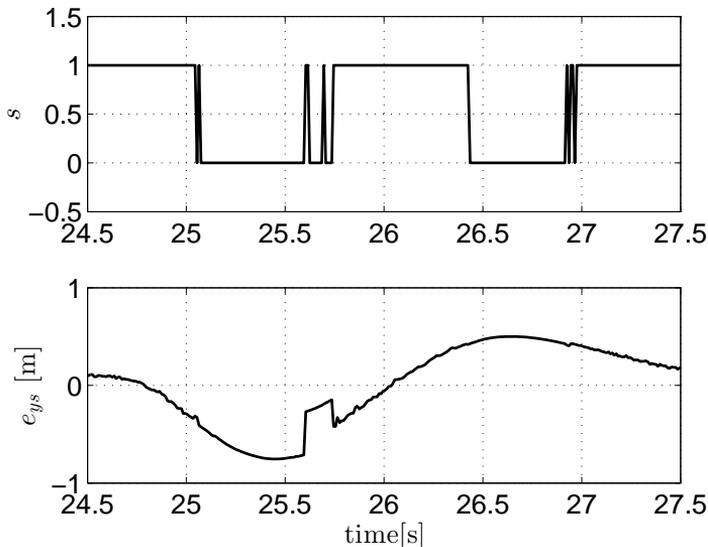
**Figure 8.15:** The  $(x, y, z)$  rotorcraft states: The perturbation in the lateral dynamic ( $y$  state) is shown at  $t = 25$  seconds.



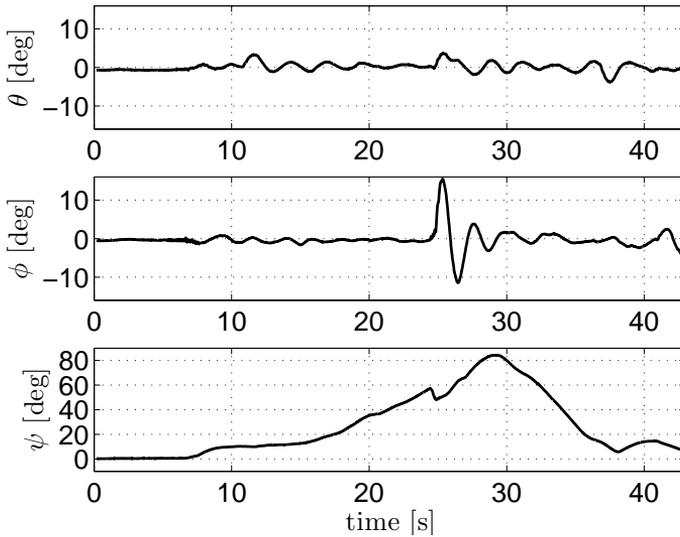
**Figure 8.16:** Translational velocities during the experiment: The forward velocity  $\dot{x}$  is kept at 0.3 m/s, while the lateral velocity  $\dot{y}$  and altitude velocity  $\dot{z}$  are both kept around zero. The variation in  $\dot{y}$  at around 25 s is caused by the external perturbation.



**Figure 8.17:** Behavior of the switching signals  $\psi_d$ ,  $s$  and  $e_{ys}$ . The graphic in the middle represents the instants of time when the line is being detected ( $s = 1$ ) and when the line is not being detected ( $s = 0$ ). During the take-off and landing phases the switching signal  $s$  is kept fixed at a value of  $s = 1$ . Once the desired height is reached, the signal  $s$  varies according to the situation encountered, i.e.,  $s = 1$  when the road is detected, and  $s = 0$  for when the road is not detected.



**Figure 8.18:** A zoom to the region where the disturbance occurs: The upper graphic shows how the  $s$  signal changes between 1 and 0. The lower graphic shows how the switching strategy influences the lateral position measurement.



**Figure 8.19:** The rotorcraft Euler angles as obtained directly from the embedded IMU: The pitch angle ( $\theta$ ) is kept to a value which ensures forward displacement at the required velocity. The roll angle ( $\phi$ ) is kept small. The heading angle ( $\psi$ ) is represented w.r.t. the inertial frame  $\mathcal{I}$ .



**Figure 8.20:** Road-following mission: The rotorcraft flying in the experimental area during real-time tests.

# 9

## CONCLUSIONS AND OUTLOOK

---

*In order to achieve a fully functional convertible MAV, different research areas have been addressed, such as modeling, control, mechanics, computer vision and embedded systems. This chapter concludes and discuss future development of the convertible MAV.*

### Contents

---

9.1	Part I: The Fixed-wing aircraft . . . . .	146
9.2	Part II: The Quadrotor . . . . .	146
9.3	Part III: The Convertible Aircraft . . . . .	147
9.4	Part IV: Vision applications . . . . .	148

---

## 9.1 Part I: The Fixed-wing aircraft

---

**A Nonlinear Path-Following Strategy for a Fixed-Wing MAV** This chapter presented a nonlinear path-following strategy for a fixed-wing MAV based on the Lyapunov theory. The controller has been tested in simulation showing a good performance. Such controller relies on accurate knowledge of vehicle dynamics. It is worth mentioning that although the controller was designed using a reduced model of the airplane, the tests carried out in the simulator, worked with the complete nonlinear aircraft model. This fact leads to the effectiveness of the proposed controller on a real experimental platform. On the other hand, the simulator MAV3DSim proves to be a great candidate for implementation of different controllers. Future work will address the problem of robustness in presence of wind gusts. Also, the proposed controller will be implemented on the ElCerdo experimental platform, in order to test the effectiveness of such algorithm. In the future, other type of aerial vehicles such as quad-rotors or coaxial helicopters, could be implemented using the simulator MAV3DSim.

**Dubins Path Generation for a Fixed Wing MAV** The presented study attempts to propose a framework as a part of a more general project in which more tasks are needed in order to complete a predefined task without human interaction. Some of these tasks are trajectory generation as a function of different variables as time, user commands, visual marks, energy consumption or sensor information, just to mention a few. Dubins paths have been utilized as a tool to estimate the shortest path from the current MAV pose (attitude and position) to a given point provided by the user. Despite the fact that the proposed controller has been designed taking into consideration a simplified aircraft model, the controller performance is proved on the MAV3Dsim simulator by using a full six degree-of-freedom aerodynamic model. Besides these developments, conducting experiments on a platform is indispensable to consolidate the results of the presented study w.r.t claims of modeling simplifications and performance.

## 9.2 Part II: The Quadrotor

---

**Lyapunov-based Controller for a mini-UAV using Singular Perturbation Theory** A controller based on the singular perturbation approach has been proposed so that the closed loop behavior achieves the desired performance. We have associated the translational displacement with the slow dynamics and the rotational displacement with the fast dynamics. For this purpose we have introduced the parameter  $\varepsilon$  on the dynamic equations. The introduction of this parameter leads to a time-scale separation of the MAV system. A Lyapunov

function was proposed for the entire system and stability of the closed loop system was proved for all  $\varepsilon < 1$ .

The Lyapunov-based controller using singular perturbation theory has been tested in numerical simulations. The controller has been also successfully applied to a Quad-plane experimental platform showing good performance.

**Fault Estimation and Control for a Quad-rotor MAV using a Polynomial Observer** This work deals with the problem of fault detection and diagnosis task for a Quad-rotor mini air vehicle (MAV) using the differential algebra approach. This approach consider the unknown faults like an augmented state of the system, the strategy is proposes a bank of observers in order to estimate the fault dynamics, in this case we are only use the available measurements and known inputs. A polynomial observer was proposed to deal with the fault estimation problem for the case of multiple faults. This approach detects and identifies multiple faults of relative small magnitudes. In this work the FDD task for a system stabilized in the closed-loop using a control strategy is presented.

The second part of this chapter concentrates on the study of the controllability of the system with a failure. For that purpose, we have presented an evaluation function depending on system states. By means of this evaluation function, we can develop a control methodology which compensates the effect of the failure under certain considerations.

## 9.3 Part III: The Convertible Aircraft

---

**Control in the 6-DOF of the PVHAT aircraft. Hover Control** The 6-DOF model of the PVHAT aircraft prototype has been developed. The longitudinal tilting of the four rotors provides an additional input in the vehicle's  $x$ -dynamics. The experimental results obtained on the PVHAT aircraft prototype are very promising. In this first part of the research, the hover dynamics has been investigated. A nonlinear controller has been proposed shown satisfactory results. The second part of this research includes development of nonlinear controllers for achieving the transition maneuver, as well as the control in airplane mode.

**Nonlinear control design of transition maneuvers for convertible aerial vehicles** We have proposed a smooth nonlinear control strategy to accomplish the flight transition maneuver between hover and airplane flight for the PVHAT aircraft. The approach has been successfully tested in numerical simulations for the longitudinal model of a PVHAT which is a class of convertible aircraft. This kind of aircraft performs the transition by tilting the four rotors. It has been proved that the proposed control strategy is such that the closed loop system is globally asymptotically stable and that the altitude and the longitudinal speed converge to desired values. The desired speed can be chosen as any positive

value smaller than the maximum speed  $\dot{x}_{\max}$ . Therefore the transition can be accomplished smoothly keeping a desired altitude.

As mentioned above, the presented approach is developed bearing in mind that the controller can be easily implemented on a real platform. To this end, the controller given by (7.4), (7.5) and (7.6) depend only on the position and the airspeed measurements, leading to a feasible controller to implement in practice.

---

## 9.4 Part IV: Vision applications

### **Lyapunov-Based Switching Control for a Road Estimation and Tracking Applied on a Convertible MAV**

The problem of road following using a convertible MAV equipped with a fully embedded imaging and control system was addressed. The goal of this research consists of estimating and tracking a road without a priori knowledge of the path to be tracked, as well as of deriving efficient estimation and control strategies for dealing with situations when the road is not detected by the vision system. Two main operational regions were defined: one for the case when the road is detected, and the other for the case when it is not. A switching between measurements coming from imaging and inertial sensors was used in order to estimate the vehicle parameters required in both regions of operation. In addition to this, a switching control strategy for stabilizing the vehicle lateral position was proposed. The system stability was verified not only in the two operational regions, but also in the switching boundaries between them. The performance of the switching strategies for sensing and control was tested in a numerical simulation and a real time application. The experiment allowed to validate the effectiveness of the proposed methods for autonomously performing the task of road following in the presence of external disturbances and unexpected failures of the imaging system. Future work will concern a hybrid scheme for switching not only different controller gains but also different control strategies.

# BIBLIOGRAPHY

---

- [1] R. W. Beard, T. W. McLain, D. B. Nelson, D. Kingston, and D. Johanson. Decentralized cooperative aerial surveillance using fixed-wing miniature uavs. *Proceedings of the IEEE*, 97(7):1306–1324, 2006.
- [2] L. Merino and A. Ollero. Cooperative fire detection using unmanned aerial vehicles. In *2005 IEEE International Conference on Robotics and Automation (ICRA 2005)*, pages 1884 – 1889, Barcelona, Spain, apr 2005.
- [3] P. Doherty and P. Rudol. Proceedings of the 20th australian joint conference on artificial intelligence, gold coast, australia, december 2-6, 2007. In Mehmet A. Orgun and John Thornton, editors, *AI 2007: Advances in Artificial Intelligence*, Lecture Notes in Computer Science. Springer Berlin Heidelberg, address =.
- [4] Carol Martinez, Thomas Richardson, Peter Thomas, Jonathan Luke du Bois, and Pascual Campoy. A vision-based strategy for autonomous aerial refueling tasks. *Robotics and Autonomous Systems*, 61(8):876–895, 2013.
- [5] E. Feron and E. N. Johnson. Aerial robotics. In Bruno Siciliano and Oussama Khatib, editors, *Springer Handbook of Robotics*. Springer Berlin Heidelberg, address =.
- [6] Gerardo Flores, J. Escareno, R. Lozano, and S. Salazar. Quad-tilting rotor convertible MAV: Modeling and real-time hover flight control. *Journal of Intelligent and Robotic Systems*, 65(1-4):457–471, 2012.
- [7] Roberto Naldi and Lorenzo Marconi. Robust control of transition maneuvers for a class of v/stol aircraft. *Automatica*, 49(6):1693–1704, 2013.
- [8] Nabil Aouf, Benoit Boulet, and Ruxandra Botez. Vector field path following for small unmanned air vehicles. In *Proc. IEEE American Control Conference (ACC2002)*, pages 4439–4442, Anchorage, AK, May 2002.
- [9] Marius Niculescu. Lateral track control law for aerosonde UAV. In *39th AIAA Aerospace Sciences Meeting and Exhibit*, pages AIAA2001–0016, Reno, NV, jan 2001.
- [10] Derek R. Nelson, D. Blake Barber, Timothy W. McLain, and Randal W. Beard. Image moments: a general and useful set of features for visual servoing. *IEEE Trans. on Robotics*, 23(3):512–529, June, 2007.
- [11] Ren Wei and R. W. Beard. Trajectory tracking for unmanned air vehicles with velocity and heading rate constraints. *IEEE Transactions on Control Systems Technology*, 12(5):706–716, Sept. 2004.
- [12] Derek R. Nelson, D. Blake Barber, Timothy W. McLain, and Randal W. Beard. Vector field path following for small unmanned air vehicles. In *Proc. IEEE American Control Conference (ACC'2006)*, pages 5788–5794, Minneapolis, Minnesota, USA, June 2006.
- [13] Oussama Khatib. Real-time obstacle avoidance for manipulators and mobile robots. In *1985 IEEE International Conference on Robotics and Automation ICRA1985*, pages 500–505, St. Louis, Missouri, may 1985.
- [14] K. Sigurd and J. P. How. UAV trajectory design using total field collision avoidance. In *Proc. of the AIAA Guidance, Navigation, and Control Conference*, Austin, TX, August 2003.

- [15] Yoram Koren and Johann Borenstein. Potential field methods and their inherent limitations for mobile robot navigation. In *Proc. IEEE International Conference on Robotics and Automation (ICRA'1991)*, pages 1398–1404, Sacramento, California, April 1991.
- [16] F. Allgower and A. Zheng. *Nonlinear Model Predictive Control*. Birkhauser, 2000.
- [17] H. Chitsaz and S. LaValle. Time-optimal paths for a dubins airplane. In *Proc. IEEE Conference on Decision and Control (CDC'2007)*, pages 2379–2384, New Orleans, LA, USA, December 2007.
- [18] A. Micaelli and C. Samson. Trajectory tracking for unicycle-type and two-steering-wheels mobile robots. Technical Report 2097, INRIA, 1993.
- [19] Lionel Lapierre and Didik Soetanto. Nonlinear path-following control of an auv. *Ocean Engineering*, 11–12(34):1734–1744, 2007.
- [20] Open Source. Crrcsim, 2013.
- [21] Crossbow Technology, Inc. *MNAV100CA User's Manual*, 1996.
- [22] Israel Lugo-Cárdenas. Nonlinear control of a mini aerial vehicle for trajectory tracking. Master's thesis, Centre for Research and Advanced Studies, Av. Instituto Politécnico Nacional 2508, Mexico City, Mexico, 2013.
- [23] L.E.Dubins. On curves of minimal length with a constraint on average curvature and with prescribed initial and terminal positions and tangents. *American Journal of Mathematics*, 79(3):497–516, jul 1957.
- [24] J.A.Reeds and R.A.Sheep. Optimal paths for a car that goes both forward and backward. *Pacific Journal of Mathematics*, 2:367–393, 1990.
- [25] Lydia E. Kavraki, Petr Svestka, Jean-Claude Latombe, and Mark H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE T. Robotics and Automation*, 12(4):566–580, 1996.
- [26] Y. Kuwata, A. Richards, T. Schouwenaars, and J. P. How. Decentralized robust receding horizon control for multi-vehicle guidance. In *American Control Conference (ACC)*, pages 2047–2052, Minneapolis, MN, June 2006.
- [27] T. Mehta and M. Egerstedt. An optimal control approach to mode generation in hybrid systems. *Nonlinear Analysis: Theory, Methods and Applications*, 65(5):963–983, Sept 2006.
- [28] Gerardo Flores, Israel Lugo-Cardenas, and Rogelio Lozano. A nonlinear path-following strategy for a fixed-wing MAV. In *International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 1014–1021, Grand Hyatt Atlanta, Atlanta, GA, May 2013.
- [29] J.A. Farrel and M. Barth. *The Global Positioning System and Inertial Navigation*. John Wiley and Sons, Hoboken, NJ, USA, 2007.
- [30] Anonymous. World geodetic system 1984 its definition and relationships with local geodetic systems. Technical Report NIMA TR 8350.2, Defense Mapping Agency, 2000.
- [31] S. Bouabdallah and R. Siegwart. Backstepping and sliding-mode techniques applied to an indoor micro quadrotor. In *In Proceedings of IEEE Int. Conf. on Robotics and Automation*, pages 2247–2252, Barcelona, Spain, April 2005.

- [32] D. Cabecinhas, R. Cunha, and C. Silvestre. Saturated output feedback control of a quadrotor aircraft. In *Proc. IEEE American Control Conference (ACC'2012)*, pages 4667–4672, Montréal, Canada, June 2012.
- [33] Taeyoung Lee, Melvin Leok, and N. Harris McClamroch. Nonlinear robust tracking control of a quadrotor UAV on SE(3). In *Proc. IEEE American Control Conference (ACC'2012)*, pages 4649–4654, Montréal, Canada, June 2012.
- [34] Gerardo Flores, L. Garcia, G. Sanahuja, and Rogelio Lozano. Pid switching control for a highway estimation and tracking applied on a convertible mini-UAV. In *Proc. 51st IEEE Conference on Decision and Control (CDC'2012)*, Maui, HI, USA, December 2012. 3110 - 3115.
- [35] Erdinç Altuğ, James P. Ostrowski, and Camillo J. Taylor. Control of a quadrotor helicopter using dual cameravisual feedback. *The International Journal of Robotics Research*, 24(5):329–341, 2005.
- [36] L.R. García Carrillo, Gerardo Flores, G. Sanahuja, and R. Lozano. Quad-rotor switching control: An application for the task of path following. In *Proc. IEEE American Control Conference (ACC'2012)*, pages 4637–4642, Montréal, Canada, June 2012.
- [37] D. Subbaram Naidu and Anthony J. Calise. Singular perturbations and time scales in guidance and control of aerospace systems: A survey. *Journal of Guidance, Control and Dynamics*, 24(6):1057–1078, 2001.
- [38] Sergio Esteban, Javier Aracil, and Francisco Gordillo. Lyapunov based asymptotic stability analysis of a three-time scale radio/control helicopter model. In *AIAA Atmospheric Flight Mechanics Conference and Exhibit*, Honolulu, Hawaii, August 2008.
- [39] S. Bertrand, T. Hamel, and H. Piet-Lahanier. Stability analysis of an UAV controller using singular perturbation theory. In *Proceedings of the 17th World Congress The International Federation of Automatic Control*, pages 5706–5711, Seoul, Korea, July 2008.
- [40] B. Etkin and L.D. Reid. *Dynamics of Flight Stability and Control*. John Wiley and Sons, 1996.
- [41] M. Oishi and C. Tomlin. Switching in nonminimum phase systems: Application to VSTOL aircraft. In *Proc. IEEE American Control Conference (ACC'2000)*, pages 838–843, Chicago, IL, June 2000.
- [42] Robert Mahony and Tarek Hamel. Robust trajectory tracking for a scale model autonomous helicopter. *International Journal of Robust and Nonlinear Control*, 14(12):1035–1059, 2004.
- [43] P. Kokotović, H. K. Khalil, and J. O'Reilly. *Singular Perturbation Methods in Control: Analysis and Design*. Siam, London: Academic Press, 1999.
- [44] Gerardo Flores and Rogelio Lozano. Lyapunov-based controller using singular perturbation theory: An application on a mini-UAV. In *Proc. IEEE American Control Conference (ACC'2013)*, pages 1599–1604, Washington, DC, June 2013.
- [45] Hicham Khebbache. Robust control algorithm considering the actuator faults for attitude tracking of an UAV quadrotor aircraft. *International Journal of Control and Automation*, 5(4):55–66, 2012.

- [46] A. Freddi, S. Longhi, and A. Monteriù. A model-based fault diagnosis system for a mini-quadrotor. In *7th Workshop on Advanced Control and Diagnosis (ACD 2009)*, pages –, Zielona Góra, Poland, nov 2009.
- [47] A. Freddi, S. Longhi, and A. Monteriù. Actuator fault detection system for a mini-quadrotor. In *2010 IEEE International Symposium on Industrial Electronics (ISIE)*, pages 2055–2060, Bari, Italy, 2010.
- [48] Alessandro Freddi, Alexander Lanzon, and Sauro Longhi. A feedback linearization approach to fault tolerance in quadrotor vehicles. In *Proceedings of The 2011 IFAC World Congress, Milan, Italy*, 2011.
- [49] M. Ranjbaran and K. Khorasani. Fault recovery of an under-actuated quadrotor aerial vehicle. In *Decision and Control (CDC), 2010 49th IEEE Conference on*, pages 4385–4392, dec. 2010.
- [50] Michel Fliess, Cedric Join, and Hebertt Sira-Ramirez. Robust residual generation for linear fault diagnosis: an algebraic setting with examples. *International Journal of Control*, 77(14):1223–1242, 2004.
- [51] Michel Fliess, Hebertt Sira-Ramirez, et al. Control via state estimations of some nonlinear systems. In *IFAC Symposium on Nonlinear Control Systems (NOLCOS 2004)*, 2004.
- [52] J Juan Rincon-Pasaye, Rafael Martinez-Guerra, and Alberto Soria-Lopez. Fault diagnosis in nonlinear systems: An application to a three-tank system. In *American Control Conference, 2008*, pages 2136–2141. IEEE, 2008.
- [53] Rafael Martinez-Guerra and Alberto Luviano-Juarez. Fault diagnosis of nonlinear systems using reduced-order observers and algebraic observers. In *Decision and Control, 2006 45th IEEE Conference on*, pages 544–549. IEEE, 2006.
- [54] Carlos D. Garcia-Beltran, Alejandro Rodriguez-Palacios, Gerardo V. Guerrero-Ramirez, Flor M. Silva-Carranza, and Felipe Sorcia-Vazquez. Fault diagnosis of an induction motor based on differential algebra reconstructors. In *World Automation Congress (WAC), 2012*, pages 1–6, june 2012.
- [55] H. Aguilar Sierra, R. Martinez-Guerra, and J.L. Mata-Machuca. Fault diagnosis via a polynomial observer. In *Electrical Engineering Computing Science and Automatic Control (CCE), 2011 8th International Conference on*, pages 1–6, oct. 2011.
- [56] JL Mata-Machuca, Rafael Martinez-Guerra, and Ricardo Aguilar-López. An exponential polynomial observer for synchronization of chaotic systems. *Communications in Nonlinear Science and Numerical Simulation*, 15(12):4114–4130, 2010.
- [57] Juan C Cruz-Victoria, Rafael Martínez-Guerra, and José Juan Rincón-Pasaye. On nonlinear systems diagnosis using differential and algebraic methods. *Journal of the Franklin Institute*, 345(2):102–118, 2008.
- [58] H. K. Khalil. *Nonlinear Systems*. Prentice Hall, New York, 2002.
- [59] D. Snyder. The quad tiltrotor: Its beginning and evolution. In *Proceedings of the 56th Annual Forum, American Helicopter Society*, 2000.
- [60] Duc Anh Ta, Isabelle Fantoni, and Rogelio Lozano. Modeling and control of a tilt tri-rotor airplane. In *Proc. IEEE American Control Conference (ACC'2012)*, pages 131–136, Montréal, Canada, June 2012.

- [61] K. Nonami. Prospect and recent research and development for civil use autonomous unmanned aircraft as UAV and MAV. In *Journal of System Design and Dynamics*, volume 1, 2007.
- [62] J. Escareno, S. Salazar, and R. Lozano. Modelling and control of a convertible VTOL aircraft. In *45th IEEE Conference on Decision and Control*, pages 69–74, San Diego, CA, USA, Dec. 2006.
- [63] Gerardo Flores and Rogelio Lozano. A nonlinear control law for hover to level flight for the quad tilt-rotor UAV. In *19th IFAC World Congress*, page To appear, Cape Town, South Africa, Aug. 2014.
- [64] Alberto Isidori. *Nonlinear Control Systems*. Springer, Great Britain, third edition, 1995.
- [65] Minh-Duc Hua, T. Hamel, Pascal Morin, and Claude Samson. A control approach for thrust-propelled underactuated vehicles and its application to VTOL drones. *IEEE Transactions on Automatic Control*, 54(8):1837 – 1853, 2009.
- [66] Marc Sklar. Diversity in design. *Boeing Frontiers*, 5(8):44–45, 2007.
- [67] Kenzo Nonami, Farid Kendoul, Satoshi Suzuki, Wei Wang, and Daisuke Nakazawa. *Autonomous Flying Robots*, chapter Development of Autonomous Quad-Tilt-Wing (QTW) Unmanned Aerial Vehicle: Design, Modeling, and Control. Springer Japan, Japan, 2010.
- [68] E. Cetinsoy, S. Dikyar, C. Hancer, K.T. Oner, E. Sirimoglu, M. Unel, and M.F. Aksit. Design and construction of a novel quad tilt-wing UAV. *Mechatronics*, 22(6):723–745, 2012.
- [69] Masayuki Sato and Koji Muraoka. Flight test verification of flight controller for quad tilt wing unmanned aerial vehicle. In *Proc. of the AIAA Guidance, Navigation, and Control Conference*, Boston, MA, USA, Aug. 2013.
- [70] R. Hugh Stone. Aerodynamic modelling of a wing-in-slipstream tail-sitter UAV. In *AIAA Biennial International Powered Lift Conference and Exhibit*, pages AIAA 2002–5951, Williamsburg, Virginia, USA, Nov. 2002.
- [71] William E. Green and Paul Y. Oh. A hybrid MAV for ingress and egress of urban environments. *IEEE Transactions on Robotics*, 25(2):253–263, 2009.
- [72] David R. Mix, John S. Koenig, Karen M. Linda, Oguzhan Cifdaloz, Valana L. Wells, and Armando A. Rodriguez. Towards gain-scheduled  $\mathcal{H}_\infty$  control design for a tilt-wing aircraft. In *Proc. IEEE Conference on Decision and Control (CDC'2004)*, pages 1222–1222, Atlantis, Paradise Island, Bahamas, Dec 2004.
- [73] Roberto Naldi, Luca Gentili, Lorenzo Marconi, and Andrea Sala. Design and experimental validation of a nonlinear control law for a ducted-fan miniature aerial vehicle. *Control Engineering and Practice*, 18(7):747–760, 2010.
- [74] Roberto Naldi and Lorenzo Marconi. Optimal transition maneuvers for a class of V/STOL aircraft. *Automatica*, 47(5):870–879, 2011.
- [75] W.E Green and P.Y. Oh. Autonomous hovering of a fixed-wing micro air vehicle. In *IEEE International Conference on Robotics and Automation, ICRA 2006.*, pages 2164–2169, Orlando, FL, USA, May 2006.
- [76] J. Escareno, R.H. Stone, A. Sanchez, and R. Lozano. Modeling and control strategy for the transition of a convertible tail-sitter UAV. In *Proceedings of the European Control Conference*, pages 3385–3390, Kos, Greece, Jul. 2007.

- [77] Randal W. Beard and Timothy W. McLain. *Small Unmanned Aircraft: Theory and Practice*. Princeton University Press, United Kingdom, 2012.
- [78] P.B. Sujit, S. Saripalli, and J. Borges Sousa. Unmanned aerial vehicle path following: A survey and analysis of algorithms for fixed-wing unmanned aerial vehicles. *IEEE Control Systems Magazine*, 34(1):42–59, 2014.
- [79] S. Park, J. Deyst, and J. P. How. Performance and Lyapunov stability of a nonlinear path-following guidance method. *Journal of Guidance, Control and Dynamics*, 30(6):1718–1728, 2007.
- [80] John Hauser, Shankar Sastry, and George Meyer. Nonlinear control design for slightly non-minimum phase systems: Application to V/STOL aircraft. *Automatica*, 28(4):665–679, 1992.
- [81] Minh-Duc Hua, T. Hamel, Pascal Morin, and Claude Samson. Introduction to feedback control of underactuated VTOL vehicles: A review of basic control design ideas and principles. *IEEE Control Systems Magazine*, 33(1):61–75, 2013.
- [82] Abdelkader Abdessameud and Abdelhamid Tayebi. Global trajectory tracking control of VTOL-UAVs without linear velocity measurements. *Automatica*, 46(6):1053–1059, 2010.
- [83] Luis Carrillo, Gerardo Flores, Gillaume Sanahuja, and Rogelio Lozano. Quad rotorcraft switching control: An application for the task of path following. *IEEE Transactions on Control Systems Technology*, 22(4):1255–1267, 2014.
- [84] Robert F. Stengel. *Flight Dynamics*. Princeton University Press, 2004.
- [85] Andrew R. Teel. Global stabilization and restricted tracking for multiple integrators with bounded controls. *Systems & Control Letters*, 18(3):165–171, 1992.
- [86] H.J. Sussmann, E.D. Sontag, and Y. Yang. A general result on the stabilization of linear systems using bounded controls. *IEEE Transactions on Automatic Control*, 39(12):2411–2425, 1994.
- [87] Isabelle Fantoni, Arturo Zavala, and Rogelio Lozano. Global stabilization of a PVTOL aircraft with bounded thrust. In *Conference on Decision and Control*, pages 4462–4467, Las Vegas, Nevada, USA, Dec. 2002.
- [88] E. Altug, J.P. Ostrowski, and R. Mahony. Control of a quadrotor helicopter using visual feedback. In *International Conference on Robotics and Automation*, pages 72–77, Washington, DC, May. 2002.
- [89] S. Zingg, D. Scaramuzza, S. Weiss, and R. Siegwart. MAV navigation through indoor corridors using optical flow. In *International Conference on Robotics and Automation*, pages 3361–3368, Anchorage, Alaska, 2010.
- [90] D. Cabecinhas, C. Silvestre, and R. Cunha. Vision-based quadrotor stabilization using a pan and tilt camera. In *2010 49th IEEE Conference on Decision and Control (CDC)*, pages 1644–1649, Atlanta, GA, USA, Dec 2010.
- [91] Luis García-Carrillo, E. Rondon, A. Sanchez, A. Dzul, and R. Lozano. Stabilization and trajectory tracking of a quad rotor using vision. *Journal of Intelligent and Robotic Systems*, 61(1-4):103–118, January, 2011.
- [92] E. Rondon, L. Garcia-Carrillo, and I. Fantoni. Vision-based altitude, position and speed regulation of a quad-rotor rotorcraft. In *International Conference on Intelligent Robots and Systems*, pages 628 – 633, Taipei, Taiwan, October, 2010.

- [93] G.M. Hoffmann, H. Huang, S. L. Waslander, and C.J. Tomlin. Quadrotor helicopter flight dynamics and control: Theory and experiment. In *Proc. of the AIAA Guidance, Navigation, and Control Conference*, Hilton Head, South Carolina, USA, 2007.
- [94] J. Hespanha, O. Yakimenko, I. Kaminer, and A. Pascoal. Linear parametrically varying systems with brief instabilities: An application to integrated vision/imu navigation. *IEEE Transactions on Aerospace and Electronic Systems*, 40(3):889–902, July, 2004.
- [95] F. Chaumette. Image moments: a general and useful set of features for visual servoing. *IEEE Trans. on Robotics*, 20(4):713–723, August, 2004.
- [96] L. Shapiro and G. Stockman. *Computer Vision*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1st ed edition, 2001.
- [97] J. Shi and C. Tomasi. Good features to track. In *1994 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'94)*, pages 593–600, Seattle, WA, USA, jun 1994.
- [98] J. Bouguet. Pyramidal implementation of the lucas-kanade feature tracker. Technical Report Technical Report, Intel Corporation 5, Microprocessor Research Labs, January 1999.
- [99] Samir Bouabdallah. *Design And Control of Quadrotors with Application to Autonomous Flying*. PhD thesis, Ecole Polytechnique Federale De Lausanne, Lausanne, February 2007.
- [100] J. Slotine and W. Li. *Applied Nonlinear Control*. Prentice Hall, 1990.
- [101] D. Liberzon. *Switching in Systems and Control*. Birkhuser, Boston, 2003.
- [102] João Pedro Hespanha and A. Stephen Morse. Stability of switched systems with average dwell-time. In *Conference on Decision and Control*, pages 2655–2660, December 1999.
- [103] P. Peleties and R. DeCarlo. Asymptotic stability of m-switched systems using Lyapunov-like functions. In *IEEE American Control Conference*, pages 1679–1684, Boston, MA, USA, Jun 1991.