



**HAL**  
open science

# Autonomous navigation and teleoperation of unmanned aerial vehicles using monocular vision

Diego Alberto Mercado-Ravell

► **To cite this version:**

Diego Alberto Mercado-Ravell. Autonomous navigation and teleoperation of unmanned aerial vehicles using monocular vision. Other [cs.OH]. Université de Technologie de Compiègne, 2015. English. NNT : 2015COMP2239 . tel-01418220

**HAL Id: tel-01418220**

**<https://theses.hal.science/tel-01418220>**

Submitted on 16 Dec 2016

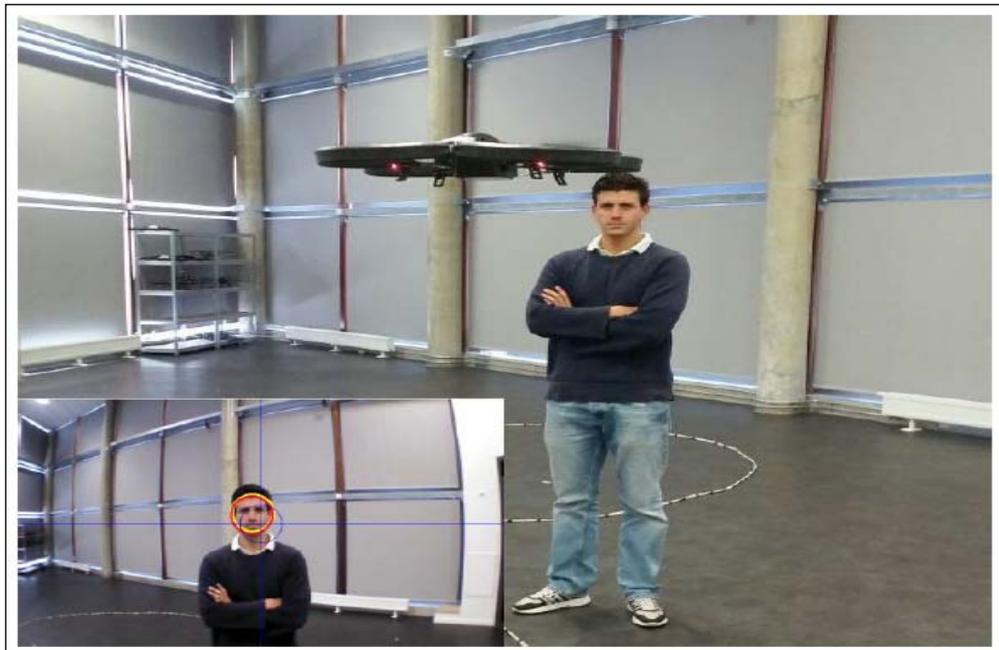
**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

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 **Diego Alberto MERCADO-RAVELL**

*Autonomous navigation and teleoperation of  
unmanned aerial vehicles using monocular vision*

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



Soutenue le 04 décembre 2015

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

D2239



# **Autonomous Navigation and Teleoperation of Unmanned Aerial Vehicles using Monocular Vision.**

by

**Diego Alberto MERCADO-RAVELL**

Submitted to the Department Technologies de l'Information et des  
Systèmes

in partial fulfillment of the requirements for the degree of

Ph.D. in Automation, Embedded Systems and Robotics at

the

Université de Technologie Compiègne

4 December 2015

© Université de Technologie Compiègne 2015. All rights reserved.

Author .....  
Department Technologies de l'Information et des Systèmes  
December 4, 2015

Certified by .....  
Rogelio LOZANO  
Directeur de recherche CNRS  
Thesis Supervisor

Certified by .....  
Pedro CASTILLO  
Chercheur CNRS  
Thesis Supervisor

Certified by .....  
Rafael CASTRO  
Chercheur CINVSTAV  
Thesis Supervisor



# Navigation Autonome et Télé-Opération de Véhicules Aériens en utilisant la Vision Monoculaire.

Diego Alberto MERCADO-RAVELL

---

Thèse soutenue le 04 décembre devant le jury composé de :

**Président:**

*Pascal VASSEUR*  
Professeur  
Univ. de Rouen

**Rapporteurs:**

<i>Nicolas MARCHAND</i> Directeur de Recherche CNRS GIBSA-lab	<i>Pascal VASSEUR</i> Professeur Univ. de Rouen
---	---

**Examineurs:**

<i>Vincent FREMONT</i> Maître de Conférences Univ. de Technologie de Compiègne	<i>Anand SANCHEZ</i> Chercheur CINVESTAV, Mexico	<i>Juan ESCAREÑO</i> Enseignant-chercheur IPSA
--	--	--

**Directeurs de Thèse:**

<i>Rogelio LOZANO</i> Directeur CNRS Univ. de Technologie de Compiègne	<i>Pedro CASTILLO</i> Chercheur CNRS Univ. de Technologie de Compiègne	<i>Rafael CASTRO</i> Chercheur CINVESTAV-IPN
--	--	--

---

Université de Technologie de Compiègne

Laboratoire Heudiasyc UMR CNRS 7253

4 December 2015



heudiasyc



utc  
Recherche







# Autonomous Navigation and Teleoperation of Unmanned Aerial Vehicles using Monocular Vision.

by

Diego Alberto MERCADO-RAVELL

Submitted to the Department Technologies de l'Information et des Systèmes  
on December 4, 2015, in partial fulfillment of the  
requirements for the degree of  
Ph.D. in Automation, Embedded Systems and Robotics

## Abstract

The present document addresses, theoretically and experimentally, the most relevant topics for Unmanned Aerial Vehicles (UAVs) in autonomous and semi-autonomous navigation. According with the multidisciplinary nature of the studied problems, a wide range of techniques and theories are covered in the fields of robotics, automatic control, computer science, computer vision and embedded systems, among others.

As part of this thesis, two different experimental platforms were developed in order to explore and evaluate various theories and techniques of interest for autonomous navigation. The first prototype is a quadrotor specially designed for outdoor applications and was fully developed in our lab. The second testbed is composed by a non expensive commercial quadrotor kind *AR. Drone*, wireless connected to a ground station equipped with the Robot Operating System (ROS), and specially intended to test computer vision algorithms and automatic control strategies in an easy, fast and safe way.

In addition, this work provides a study of data fusion techniques looking to enhance the UAVs pose estimation provided by commonly used sensors. Two strategies are evaluated in particular, an Extended Kalman Filter (EKF) and a Particle Filter (PF). Both estimators are adapted for the system under consideration, taking into account noisy measurements of the UAV position, velocity and orientation. Simulations show the performance of the developed algorithms while adding noise from real GPS (Global Positioning System) measurements.

Safe and accurate navigation for either autonomous trajectory tracking or haptic teleoperation of quadrotors is presented as well. A second order Sliding Mode (2-SM) control algorithm is used to track trajectories while avoiding frontal collisions in autonomous flight. The time-scale separation of the translational and rotational dynamics allows us to design position controllers by giving desired references in the roll and pitch angles, which

is suitable for quadrotors equipped with an internal attitude controller. The 2-SM control allows adding robustness to the closed-loop system. A Lyapunov based analysis probes the system stability. Vision algorithms are employed to estimate the pose of the vehicle using only a monocular SLAM (Simultaneous Localization and Mapping) fused with inertial measurements. Distance to potential obstacles is detected and computed using the sparse depth map from the vision algorithm. For teleoperation tests, a haptic device is employed to feedback information to the pilot about possible collisions, by exerting opposite forces. The proposed strategies are successfully tested in real-time experiments, using a low-cost commercial quadrotor.

Also, conception and development of a Micro Aerial Vehicle (MAV) able to safely interact with human users by following them autonomously, is achieved in the present work. Once a face is detected by means of a Haar cascade classifier, it is tracked applying a Kalman Filter (KF), and an estimation of the relative position with respect to the face is obtained at a high rate. A linear Proportional Derivative (PD) controller regulates the UAV's position in order to keep a constant distance to the face, employing as well the extra available information from the embedded UAV's sensors. Several experiments were carried out through different conditions, showing good performance even under disadvantageous scenarios like outdoor flight, being robust against illumination changes, wind perturbations, image noise and the presence of several faces on the same image.

Finally, this thesis deals with the problem of implementing a safe and fast transportation system using an UAV kind quadrotor with a cable suspended load. The objective consists in transporting the load from one place to another, in a fast way and with minimum swing in the cable. An Interconnection and Damping Assignment-Passivity Based Control (IDA-PBC) for the system of interest is successfully implemented and tested in real time experiments. The use of this control technique avoids the use of an extra sensor for the swing angle measurement. An experimental platform composed by a quadrotor equipped with a rigid rod with two degrees of freedom in the joint, is developed. UAV navigation is achieved via monocular vision.

# Résumé

Ce travail porte, de façon théorique et pratique, sur les sujets plus pertinents autour des drones en navigation autonome et semi-autonome. Conformément à la nature multidisciplinaire des problèmes étudiés, une grande diversité des techniques et théories ont été couverts dans les domaines de la robotique, l'automatique, l'informatique, la vision par ordinateur et les systèmes embarqués, parmi autres.

Dans le cadre de cette thèse, deux plates-formes expérimentales ont été développées afin de valider la théorie proposée pour la navigation autonome d'un drone. Le premier prototype, développé au laboratoire, est un quadrirotor spécialement conçu pour les applications extérieures. La deuxième plate-forme est composée d'un quadrirotor à bas coût du type *AR.Drone* fabrique par *Parrot*. Le véhicule est connecté sans fil à une station au sol équipé d'un système d'exploitation pour robots (ROS) et dédié à tester, d'une façon facile, rapide et sécurisé, les algorithmes de vision et les stratégies de commande proposés.

Les premiers travaux développés ont été basés sur la fusion de données pour estimer la position du drone en utilisant des capteurs inertiels et le GPS. Deux stratégies ont été étudiées et appliquées, le Filtre de Kalman Etendu (EKF) et le filtre à Particules (PF). Les deux approches prennent en compte les mesures bruitées de la position de l'UAV, de sa vitesse et de son orientation. On a réalisé une validation numérique pour tester la performance des algorithmes.

Une tâche dans le cahier de cette thèse a été de concevoir d'algorithmes de commande pour le suivi de trajectoires ou bien pour la télé-opération. Pour ce faire, on a proposé une loi de commande basée sur l'approche de Mode Glissants à deuxième ordre. Cette technique de commande permet de suivre au quadrirotor de trajectoires désirées et de réaliser l'évitement des collisions frontales si nécessaire. Etant donné que la plate-forme *A.R.Drone* est équipée d'un auto-pilote d'attitude, nous avons utilisé les angles désirés de roulis et de tangage comme entrées de commande. L'algorithme de commande proposé donne de la robustesse au système en boucle fermée.

De plus, une nouvelle technique de vision monoculaire par ordinateur a été utilisée pour la localisation d'un drone. Les informations visuelles sont fusionnées avec les mesures inertiels du drone pour avoir une bonne estimation de sa position. Cette technique utilise l'algorithme *PTAM* (localisation parallèle et mapping), qui s'agit d'obtenir un nuage de points caractéristiques dans l'image par rapport à une scène qui servira comme repère. Cet algorithme n'utilise pas de cibles, de marqueurs ou de scènes bien définies. La contribution dans cette méthodologie a été de pouvoir utiliser le nuage de points dispersé pour détecter possibles obstacles en face du véhicule. Avec cette information nous avons proposé un algorithme de commande pour réaliser l'évitement d'obstacles. Cette loi de commande

utilise les champs de potentiel pour calculer une force de répulsion qui sera appliquée au drone. Des expériences en temps réel ont montré la bonne performance du système proposé.

Les résultats antérieurs ont motivé la conception et développement d'un drone capable de réaliser en sécurité l'interaction avec les hommes et les suivre de façon autonome. Un classificateur en cascade du type Haar a été utilisé pour détecter le visage d'une personne. Une fois le visage est détecté, on utilise un filtre de Kalman (KF) pour améliorer la détection et un algorithme pour estimer la position relative du visage. Pour réguler la position du drone et la maintenir à une distance désirée du visage, on a utilisé une loi de commande linéaire. Plusieurs testes pratiques ont été réalisés sur différents conditions de luminosité (intérieur et extérieur). Ils ont montré la bonne performance du système même dans de scénarios désavantageux comme le vol à l'extérieur et même en présence de changements importants de luminosité, de vent, de bruit dans l'image, et la présence de plusieurs visages dans la même image.

Le dernier travail dans cette thèse est lié au problème de transporter un objet connecte par câble à un quadrirotor. L'objectif était de transporter rapidement la charge avec un minimum d'oscillations du câble. La commande IDA-PBC basé sur la passivité a été avec succès mise en œuvre dans la plate-forme et validée en temps-réel avec des résultats très satisfaisants. L'utilisation de cette technique permet d'éviter l'inclusion d'un capteur supplémentaire pour la mesure de l'angle du câble. De plus pour réaliser les tests, nous avons modifié notre plate-forme avec une articulation rigide avec deux degrés de liberté. La localisation du drone a été réalisée par vision monoculaire.

Thesis Supervisor: Rogelio LOZANO  
Title: Directeur de recherche CNRS

Thesis Supervisor: Pedro CASTILLO  
Title: Chercheur CNRS

Thesis Supervisor: Rafael CASTRO  
Title: Chercheur CINVESTAV

## Acknowledgments

To God, for giving me the strength of will to do what I must do, and the intelligence to know what it is.

To my parents J. Trinidad MERCADO and Ma. Elsa RAVELL, because without their love and education I would not be who I am.

To my brothers Jaime, Abraham and Fernando, to share a life by my side.

To all my family for its dearness and comprehension.

To my directors Pedro CASTILLO and Rogelio LOZANO for their valuable contributions to this thesis.

To all the members of the HEUDIASYC lab, for all the provided help.

To the *Minister de l'Enseignement Superieur et la Recherche*, for the given grant.

To all the beautiful people that I have had the honor to meet in Compiègne, for all the unforgettable memories we share.

To all the people who has believed in me at some point, because their trust motivated me to keep going.



# Publications

## International Journals

- [J1]** D. Mercado, P. Castillo and R. Lozano. “Human-Robot Interaction: the Face Follower UAV”, (in progress).
- [J2]** D. Mercado, P. Castillo and R. Lozano. “Sliding Mode Collision-Free Navigation and Haptic Teleoperation for Quadrotors Using Monocular Vision”, (in progress).
- [J3]** E. Guerrero, D. Mercado, R. Lozano and C. García. “Swing-Free Transportation using a Quadrotor UAV with a Cable-Suspended Payload”, in International Journal of Robust and Nonlinear Control (accepted with major revisions).

## International Conferences

- [C1]** E. Guerrero, D. Mercado, R. Lozano and C. García. “Passivity Based Control for a Quadrotor UAV Transporting a Cable-Suspended Payload with Minimum Swing”, in IEEE Conference on Decision and Control (CDC), Osaka, Japan, 2015.
- [C2]** D. Mercado, P. Castillo and R. Lozano. “Quadrotor’s Trajectory Tracking Control using Monocular Vision Navigation”, in International Conference on Unmanned Aircraft Systems (ICUAS), Denver, USA, 2015.
- [C3]** E. Guerrero, D. Mercado, R. Lozano and C. García. “IDA-PBC Methodology for a Quadrotor UAV Transporting a Cable-Suspended Payload”, in IEEE International Conference on Unmanned Aircraft Systems (ICUAS), Denver, USA, 2015.

- [C4]** D. Mercado, P. Castillo, R. Castro and R. Lozano. “2-Sliding Mode Trajectory Tracking Control and EKF Estimation for Quadrotors”, in the 19th IFAC World Conference, Cape Town, South Africa, 2014.
- [C5]** D. Mercado, P. Castillo and R. Lozano. “Quadrotors Data Fusion using a Particle Filter”, in International Conference on Unmanned Aircraft Systems (ICUAS), Orlando, USA, 2014.
- [C6]** D. Mercado, R. Castro and R. Lozano. “Quadrotors Flight Formation Control Using a Leader-Follower Approach”, in European Control Conference (ECC), Zurich, Switzerland, 2013.
- [C7]** D. Mercado, R. Castro and R. Lozano. “Quadrotors Trajectory Tracking Control with Inaccurate GPS Measurements”, in 2nd IFAC Workshop on Research, Education and Development of Unmanned Aerial Systems (RED UAS), Compiègne, France, 2013.
- [C8]** D. Mercado, G. Flores, P. Castillo, J. Escareno and R. Lozano. “GPS/INS/Optic Flow Data Fusion for Positioning and Velocity Estimation”, in International Conference on Unmanned Aircraft Systems (ICUAS), Atlanta, USA, 2013.
- [C9]** D. Mercado, R. Castro, R. Lozano. “Control de Quadri-rotores en un Esquema de Formación Líder-Seguidor”, in XV Congreso Latinoamericano de Control Automático, Lima, Perú, 2012.

# Contents

<b>1</b>	<b>Introduction</b>	<b>23</b>
1.1	State of the art . . . . .	24
1.2	Quadrotor Dynamic Model . . . . .	27
1.3	Objectives . . . . .	30
1.3.1	Practical objectives . . . . .	31
1.4	Contributions . . . . .	32
1.4.1	Data fusion . . . . .	
<b>2</b>	<b>Experimental Platforms</b>	<b>35</b>
2.1	Conception and Development of a Four Rotor Rotor-Craft for Autonomous Outdoor Navigation . . . . .	36
2.1.1	UAV Embedded Components . . . . .	37
2.1.2	Ground Station . . . . .	39
2.2	Integration of Available Technologies for a non Expensive Navigation Testbed on ROS . . . . .	41
2.2.1	Commercial Quadrotor . . . . .	42
2.2.2	Ground Station . . . . .	43
2.2.3	Haptic Teleoperation . . . . .	44
2.2.4	Simulator . . . . .	45
2.3	Motion Capture System for Validation . . . . .	46
<b>3</b>	<b>UAV Data Fusion for Localization</b>	<b>47</b>
3.1	Introduction . . . . .	47

3.2	Extended Kalman Filter Data Fusion . . . . .	49
3.2.1	Model Adaptation . . . . .	50
3.2.2	Simulations . . . . .	52
3.3	Particle filter . . . . .	57
3.3.1	Simulations . . . . .	58
<b>4</b>	<b>Collision-Free Navigation and Haptic Teleoperation using Monocular Vision</b>	<b>63</b>
4.1	Introduction . . . . .	64
4.2	Monocular vision localization . . . . .	67
4.3	Collision-free autonomous tracking . . . . .	68
4.3.1	2-Sliding Mode Trajectory Tracking Control . . . . .	69
4.3.2	Collision Avoidance . . . . .	72
4.4	Haptic teleoperation . . . . .	73
4.5	Experimental results . . . . .	75
4.5.1	Trajectory tracking . . . . .	75
4.5.2	Collision-free teleoperation . . . . .	77
<b>5</b>	<b>Human-Robot Interaction: the Face Follower UAV</b>	<b>83</b>
5.1	Introduction . . . . .	84
5.2	Overall System Description . . . . .	88
5.3	Face Detection and Tracking . . . . .	90
5.3.1	Computer Vision Algorithm for Face Detection . . . . .	90
5.3.2	Kalman Filter for Object Tracking . . . . .	90
5.3.3	Relative Position Estimation: From Image to Real World . . . . .	93
5.4	Relative Position Control . . . . .	94
5.4.1	PD Control . . . . .	95
5.4.2	Altitude speed estimation . . . . .	97
5.5	Experimental Results . . . . .	100
5.5.1	Face Detection and Tracking . . . . .	101
5.5.2	Relative Position Control Performance . . . . .	103

<b>6</b>	<b>Implementation of a Fast and Safe Transportation System using a Quadrotor with a Cable-Suspended Payload</b>	<b>105</b>
6.1	Introduction . . . . .	106
6.2	Control Strategy . . . . .	108
6.3	Real-Time Implementation . . . . .	111
6.4	Experimental results . . . . .	111
6.5	Acknowledgements . . . . .	115
<b>7</b>	<b>Conclusions and Future Work</b>	<b>117</b>
7.1	Experimental Platforms . . . . .	118
7.2	UAV Data Fusion for Localization . . . . .	118
7.3	Collision-Free Navigation and Haptic Teleoperation . . . . .	119
7.4	The Face Follower UAV . . . . .	120
7.5	Fast and Safe Transportation System . . . . .	121



# List of Figures

1-1	Quadrotor in an inertial reference frame. . . . .	28
1-2	Forces and torques produced by the rotors. . . . .	29
2-1	Quadrotor in flight. . . . .	36
2-2	Quadrotor main components. . . . .	37
2-3	Real time parameter tuning. . . . .	40
2-4	Real time plots. . . . .	40
2-5	GPS localization with <i>Google Maps</i> . . . . .	41
2-6	Navigation testbed on ROS. . . . .	42
2-7	Quadrotor simulator for ROS. . . . .	45
3-1	GPS measurement error. . . . .	51
3-2	Measured velocity. . . . .	52
3-3	External perturbation $w$ . . . . .	53
3-4	Estimation error . . . . .	54
3-5	Orientation. . . . .	54
3-6	Position. . . . .	55
3-7	Tracking error. . . . .	55
3-8	XY plane. . . . .	56
3-9	3D. . . . .	56
3-10	Position. . . . .	59
3-11	PF estimation error. . . . .	59
3-12	XY plane. . . . .	60
3-13	3D. . . . .	60

4-1	<i>AR.Drone</i> following a circular trajectory. . . . .	65
4-2	UAV localization w.r.t. the sparse depth map (top left). Characteristic features on the image (top right). Horizontal projection of the sparse depth map obtained by PTAM (bottom). . . . .	67
4-3	3D Collision-free lemniscate trajectory tracking close to a wall. . . . .	76
4-4	Collision-free trajectory tracking. . . . .	77
4-5	x position response for a lemniscate trajectory. . . . .	78
4-6	y position response for a lemniscate trajectory. . . . .	79
4-7	Tracking error. . . . .	79
4-8	Trajectory tracking control inputs. . . . .	80
4-9	y coordinate. . . . .	80
4-10	Haptic feedback forces. . . . .	81
4-11	UAV control inputs. . . . .	81
5-1	Quadrotor following a face. . . . .	84
5-2	Overall system description. . . . .	87
5-3	Flowchart of the vision algorithm. . . . .	89
5-4	Face detection and tracking with a flying quadrotor. . . . .	91
5-5	From real world to image coordinates transformation. . . . .	93
5-6	Altitude speed estimation comparison. . . . .	99
5-7	3D trajectory. . . . .	100
5-8	Kalman filter tracking. . . . .	101
5-9	Relative position estimated by the computer vision algorithm vs ground truth. . . . .	102
5-10	Computer vision estimation error. . . . .	102
5-11	Relative position control performance. . . . .	103
5-12	Quadrotor position relative to the face. . . . .	104
6-1	Experimental platform for a quadrotor with a cable suspended load. . . . .	106
6-2	Quadrotor with a cable suspended payload. . . . .	108
6-3	Quadrotor with a cable suspended payload, on the longitudinal plane. . . . .	109
6-4	3D view. . . . .	112

6-5	Quadrotor position. . . . .	112
6-6	Quadrotor position error. . . . .	113
6-7	Quadrotor attitude. . . . .	113
6-8	Swing angles. . . . .	114
6-9	Control inputs. . . . .	114



# List of Tables

1.1	Problem statement. . . . .	25
2.1	DSP main features. . . . .	38
4.1	Parameters . . . . .	75



# Chapter 1

## Introduction

*“Once you have tasted flight, you will forever walk the earth with your eyes turned skyward, for there you have been, and there you will always long to return.”*

*Leonardo da Vinci*



This chapter offers a brief introduction about the problematic and motivation of this thesis work, and is structured as follows: in section 1.1 a small state of the art with a review of the most impressive recent results for UAVs is given. It is followed by a general introduction of the dynamic model of a quadrotor in section 1.2. Then, in section 1.3 the general and specific objectives are established. Finally, in section 1.4 the main obtained contributions are listed.

## 1.1 State of the art

From the beginning of the times, humans have lived admiring the ski, dreaming about conquering it imitating the gracious movements of birds. Even before that the development of the science and technology allowed to build artifacts able to fly carrying a man, a special type of flying machines was conceived, the Unmanned Aerial Vehicles. At their first steps, nobody imagined all their huge potential in all kind of application, but later on, with the fast growing of robotics, computer science, electronics, aerodynamics, automatic control, etc., those flying machines became more and more autonomous, until evolve to the sophisticated flying robots that can be found today. Specially in the last decades, UAVs attracted great attention from research centers and enterprises all over the world, to be employed in missions of exploration, remote surveillance, patrolling, search and rescue, transportation, among many others. Despite the big effort and large progress achieved with this kind of systems, their study continues to be an exiting research field where new open problems emerge where others are solved, and there is still a huge set of possibilities to be explored. This have motivated the present work.

The design of new aerial machines is amazingly evolving to complex bio-inspired flying robots with ultralight weight, improved aerodynamic qualities and extreme agility. It is enough to look at the bionic gamma of new developed technologies by *Festo*, including the *SmartBird* [4] and the *BionicOpter* [1], two outstanding UAVs inspired in the herring gull and the dragonfly, respectively, which prove the huge potential still to be explored in the field of UAVs.

However, in this work we are more interested in four rotor rotorcrafts, also known as

quadrotors, since their rotor's configuration produces cancellation of the reactive torques, considerably simplifying their analysis and control. Also, they are suitable for vertical take off and landing, as well as hovering, making them a good choice for maneuvering in small spaces or performing high precision tasks.

More in detail, this work deals with the general problem of autonomous navigation for UAVs type quadrotor in outdoor missions with corrupted GPS data or no GPS at all. Then two main challenges arise as summarized in Table 1.1, the state estimation, specially for localization in GPS denied scenarios, and the robustness of the control laws against perturbations.

For the first problem, three different approaches are considered according to the application and particular conditions, first of all two data fusion techniques are studied for the case of corrupted GPS measurements. The idea is to take advantage of the information from commonly used embedded sensors, such as optic flow sensors for translational speed estimation, inertial sensors for orientation and angular speed, and ultrasounds and air pressure sensors for altitude measurement, to improve the localization despite measurement noise and uncertainties. The second approach consists in the use of computer vision techniques for pose estimation with respect to a fixed scene in GPS denied environments, in special, the so-called PTAM algorithm is considered. The main constraints are to be limited to a small and fixed scene and the expensive required computations. The third proposed solution stands for special missions where the real interest is to locate the UAV with respect to a specific object of interest, for example a moving target, then only a relative position is

<b>Autonomous Navigation</b>		
State Estimation (Localization <del>GPS</del> )	* Data Fusion (IMU / OF / GPS / altitude) * Monocular Vision ( <b>PTAM</b> )  * With respect to a target	- Uncertainties, measurement noise - Fixed scene, expensive computations - Relative position
Robustness against Perturbations	* Wind * Payload	- Unknown - Oscillations
Haptique Tele-Operation		

Table 1.1: Problem statement.

obtained.

As for the second problem of robustness on the control strategies against external perturbations, two main cases are considered. The first deals with perturbations produced by external forces, produced by the wind for example, that are unknown to the system and hard to predict. The second deals with perturbations steaming from the system itself, as for the case of transporting an extra payload attached by a cable, introducing undesired oscillations.

Then, four applications are developed combining the different studied solutions according to the particular needs. First, collision-free navigation is accomplished using monocular vision localization with respect to a fixed scene along with a second order sliding mode trajectory tracking control robust against perturbations, and potential fields for collision avoidance. Due to the complexity of a complete solution for fully autonomous navigation, an alternative operation mode is proposed for semi-autonomous tele-operation assisted by a haptic device, able to return forces to the human pilot, again the monocular vision localization is employed. The third application consists in detect and track a mobile object using relative localization by means of a camera. Last but not least, a transportation system composed by a quadrotor and a cable suspended payload is proposed, applying passivity based techniques to deal with the cable swing.

The recent research effort focused on quadrotors has produced quite impressive results, including aggressive flight [53, 52], collaborative constructions [10, 50], dancing drones [11], swarms [47], fault recovery [56], or even ball throwing and catching [63]. However, these applications are constrained to the use of an expensive external motion capture system for the vehicle localization.

The challenge to overcome this hard constrain, consists in localizing the vehicle only from its own embedded sensors (usually limited to inertial information, altitude sensors and a camera) and robustly navigate on unknown environments under disadvantageous conditions, with possible external perturbations and sensor failures. This becomes even harder using inexpensive hardware with inaccurate and noisy sensors, delayed signals and limited processors. Therefore, navigation in unstructured and GPS denied environments continues to be an open research problem of great interest, however, excellent progress can be

found on the literature about this issues based on solving the Simultaneous Localization and Mapping (SLAM) problem, or using robust state estimators, or a combination of both. A particularly interesting technique arises from the Parallel Tracking and Mapping (PTAM) algorithm [46], originally conceived for Augmented Reality (AR) applications and successfully extended to UAVs using either an expensive full embedded platform [7], or a low-cost commercial quadrotor and an external computer [28, 29].

Finally, other prominent applications consist in extending the UAVs capabilities to interact and manipulate their environment. In order to do so, it is required to introduce additional actuators like manipulator arms to handle objects [41], or to perch-and-stare, just like birds, considerably saving energy [27]. Moreover, it is even possible to equip the drone with two leg mechanisms in order to climb walls [26].

## 1.2 Quadrotor Dynamic Model

The dynamic model of a quadrotor have been widely studied and can be obtained from different methods, being the most important the Euler-Lagrange formalism [20] and Newton's second law [40], [44]. Furthermore, there exist several variants for discrete-time [38], or applying a quaternion representation to avoid singularities [74]. For more details about the model of this and other flying machines, it is recommended to review some of the classic books in aerodynamics [30], [65].

For this work, we consider the dynamic model for a quadrotor by means of the Newton-Euler formalism, for this, the quadrotor can be represented as a rigid body in space with mass  $m$  and inertia matrix  $J$ , subject to gravitational and aerodynamic forces. Let us consider an inertial coordinate frame  $I = \{X Y Z\}$ , fixed to the ground and a body fixed coordinate frame,  $B = \{e_1, e_2, e_3\}$ , see Fig. 1-1. Then consider the vectors

$$\xi = [x y z]^T \quad (1.1)$$

$$\Phi = [\varphi \ \theta \ \psi]^T \quad (1.2)$$

which stand for the position of the center of gravity, with respect to the inertial frame  $I$ , and

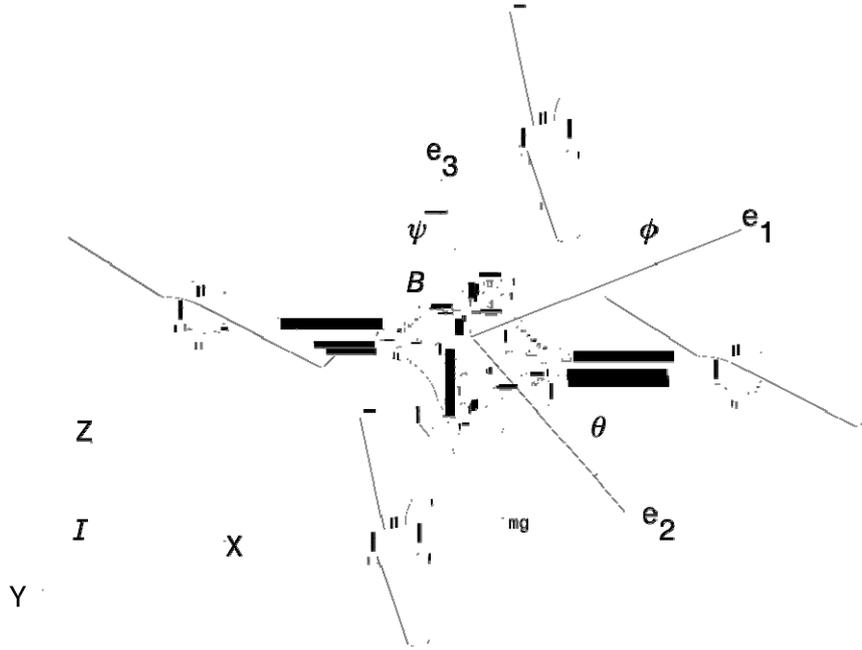


Figure 1-1: Quadrotor in an inertial reference frame.

the Euler angles roll, pitch and yaw, respectively. The motion equations are given by the Newton-Euler equations in the inertial frame  $I$  [44]

$$m\ddot{\xi} = TRe_3 - mge_3 \quad (1.3)$$

$$J\dot{\Omega} = -\Omega_x J \Omega + \Gamma \quad (1.4)$$

where  $T \in \mathbb{R}^+$  is the total thrust of the motors,  $g$  defines the gravitational constant and  $\Gamma \in \mathbb{R}^3$  describes the control torque defined in the body fixed frame  $B$ .  $e_3 = [0 \ 0 \ 1]^T$  represents the third axis in the body frame  $B$ .  $R \in SO(3) : B \rightarrow I$  introduces the rotational matrix from the body frame to the inertial frame, and using the short notation:  $s\alpha = \sin \alpha$ ,  $c\alpha = \cos \alpha$ ,  $t\alpha = \tan \alpha$ , it is given by

$$R = \begin{bmatrix} c\psi c\theta & -s\psi c\theta + c\psi s\theta s\phi & s\psi s\theta + c\psi s\theta c\phi \\ s\psi c\theta & c\psi c\theta + s\psi s\theta s\phi & -c\psi s\theta + s\psi s\theta c\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix} \quad (1.5)$$

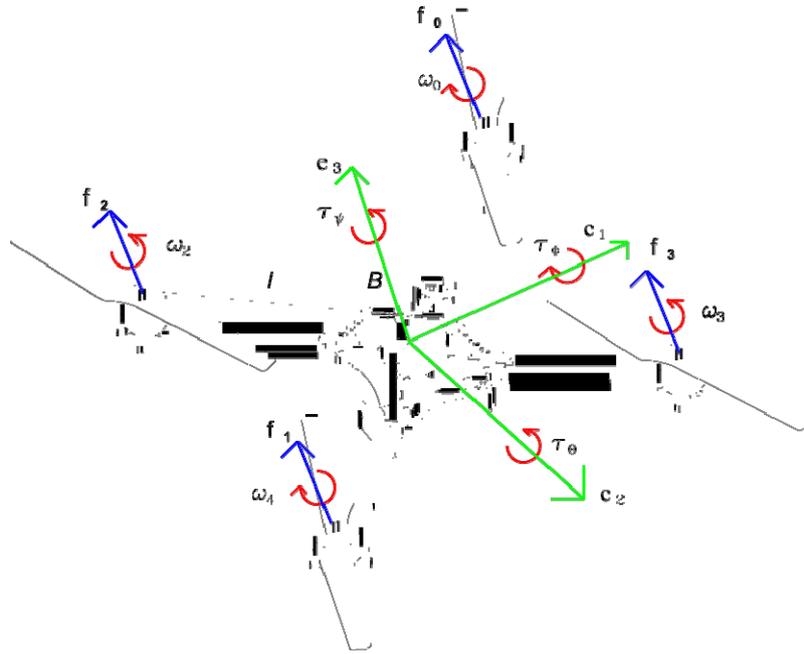


Figure 1-2: Forces and torques produced by the rotors.

$\Omega = [p \ q \ r]^T$  represents the angular velocity in the body frame  $B$ .  $\Omega_x$  stands for the skew symmetric matrix such that  $\Omega_x u = \Omega \times u$  is the vector cross product. The kinematic relation between the generalized velocities  $\dot{\Phi} = (\dot{\varphi}, \dot{\theta}, \dot{\psi})$  and the angular velocity  $\Omega$  is given by [65]

$$\Omega = \begin{bmatrix} 1 & 0 & -s\theta \\ 0 & c\varphi & c\theta s\varphi \\ 0 & -s\varphi & c\theta c\varphi \end{bmatrix} \dot{\Phi} \quad (1.6)$$

For the particular case of the quadrotor, each pair of face-to-face rotors spins in a different sense, neutralizing the reactive torques, as illustrated in Figure 1-2. Then, rotation along each axis is produced by increasing one motor's speed while decreasing the opposite one. The thrust  $f_i$  generated by the rotor  $i = 0, 1, 2, 3$  is

$$f_i \approx \omega_i^2 k_m \quad (1.7)$$

with  $\omega_i$  as the rotational speed of the motor  $i$  and the constant  $k_m$  dependent of the propellers shape and the air density, among others. The force  $T$  and the torque vector  $\Gamma$  are the inputs of the system and are produced by the total sum of the thrusts and the differential thrust produced by the rotors, respectively, such as

$$\begin{matrix}
 \square & \square & \square & & & \square & \square & \square \\
 \square & \tau_\phi & \square & = & \square & l & -l & l & -l & \square & \square & f_0 \\
 \square & \tau_\theta & \square & & \square & l & -l & -l & l & \square & \square & f_1 \\
 \square & \tau_\psi & \square & & \square & b & b & -b & -b & \square & \square & f_2 \\
 \square & T & \square & & \square & 1 & 1 & 1 & 1 & \square & \square & f_3
 \end{matrix} \tag{1.8}$$

with  $l$  and  $b$  being constants that represent respectively the distance from a motor to the gravity center and the relation between the rotational speed and the torque from each rotor.

## 1.3 Objectives

### General objectives

In general, UAVs autonomous navigation continues to be an open and exiting problem, hence, the general aim of this thesis is to deal with the ultimate and most important theoretical and practical challenges arising from UAVs autonomous navigation in hazardous situations.

Other important related objectives may be:

- Study and fully understand the dynamic model and operating principles of UAVs, in particular quadrotors.
- Conceive and develop new useful and interesting applications for UAVs.
- Ameliorate the UAVs perception and capabilities, to increase the level of autonomy.
- Enhance the human users experience by providing them important information and increasing the level of interaction with the flying robot, for the case where autonomous navigation is infeasible.

- Ensure the system's robustness and security.

## **Particular objectives**

Inside the general objective, two main problems in special are to be confronted, through various approaches. On the one hand, localization in GPS denied environments. On the other hand, to add robustness against uncertainties and perturbations presents in outdoor flight. Then, the particular objectives according to each problematic are as follows:

- Face the localization problem in GPS denied environment. Look through different alternatives and find the best solution according to the desired application.
  - Fusion information from commonly used sensors to obtain a good pose estimation.
  - Exploit the great potential of computer vision techniques to enhance the drones perception and capabilities.
- Accomplish outdoor flight, considering measurements errors and the weather conditions, such as wind.
  - Navigate in unknown unstructured scenarios, detecting and avoiding possible obstacles.
  - Design and implement new control strategies, robust against uncertainties and external perturbations.

### **1.3.1 Practical objectives**

Besides, the practical objectives are:

- Implement and validate the proposed algorithms in real-time experiments.
- Design and construct a quadrotor conditioned for outdoor flight.

## 1.4 Contributions

This thesis covered a wide variety of problems involved in different UAVs applications, giving place to some interesting practical and theoretical contributions. The most relevant of those contributions are classified according to the respective application and listed bellow:

### **Collision free navigation and teleoperation:**

- Different powerful tools were assembled and adapted to obtain a working solution for either autonomous trajectory tracking or semi-autonomous teleoperation. This includes extending the use of the PTAM algorithm to detect and avoid collisions, without extra expensive calculations or sensors.
- A high order sliding mode, robust against perturbations and uncertainties, was proposed and successfully validated in real time experiments, for quadrotors autonomous trajectory tracking.
- A simple but effective strategy for avoiding frontal collisions was developed. This allows to improve the safety of the system, without extra expensive computations, using already available information.

### **Face follower UAV:**

- An innovative application was conceived and successfully implemented: the face follower UAV. It integrates several useful techniques and tools to offer a solution to all the involved problems.
- An algorithm to estimate the distance between the face and the camera is developed from the size of the detected face on the image.
- A complete computer vision algorithm was developed to detect and track a face despite the presence of noise and uncertainties.

- A real time relative position control algorithm is proposed for an UAV following a moving object with unknown dynamics, including the observation of the missing state.

#### **1.4.1 Data fusion**

- Two data fusion algorithms, for localization under bad GPS conditions, are proposed and studied through simulations, the EKF and the PF.

#### **Swing-free transportation system:**

- A practical implementation is accomplished for a transportation system using a quadrotor with a cable suspended payload, using passivity based techniques to minimize the swing angle oscillations without a direct measurement from it. The proposed solution utilize monocular vision techniques for precise localization and is not constrained to any external motion capture system.

#### **Practical contribution:**

- Two different experimental platforms were developed for this thesis. The first was completely developed in our lab and was designed for outdoor flights. The second continues and extends the work in [28] to employ a commercial quadrotor connected by *Wi-Fi* to a ground computer with *ROS*, in order to test in a fast and safe way new control laws and vision based algorithms.



# Chapter 2

## Experimental Platforms

The huge variety of applications for UAVs and the growing market around them, have produced an emerging industry specialized in drones. All kind of products related to UAVs are easily found on the market trough a wide range of prices and requirements, from actuators, sensors and micro-processors particularly adapted to the needs of flying robots, to powerful software packages commonly released as open source, until complete solutions for a specific application. This offers a lot of different options to accomplish certain task, either by constructing your own drone or using an already available one. From the research point of view, on the one hand, building your own UAV is an excellent way to really understand how does such systems work, not to mention all the technique skills and experience that will be earned. But this comes at the cost of time and big effort to reproduce something already existing. On the other hand, technologies involved in UAVs advance at a vertiginous speed, and is already hard to keep pace of it and using an available commercial drone allows to focus on the most interesting new research topics on the field.

As part of this work, two different experimental platforms were developed, in order to explore and evaluate the different theories and techniques of interest for autonomous navigation. The first prototype is a quadrotor specially designed for outdoor applications and was fully developed in our lab. The second testbed is composed by a non expensive commercial quadrotor type *AR. Drone*, connected to a ground station via wireless communication and particularly intended to test computer vision algorithms and automatic control strategies in an easy, fast and safe way.

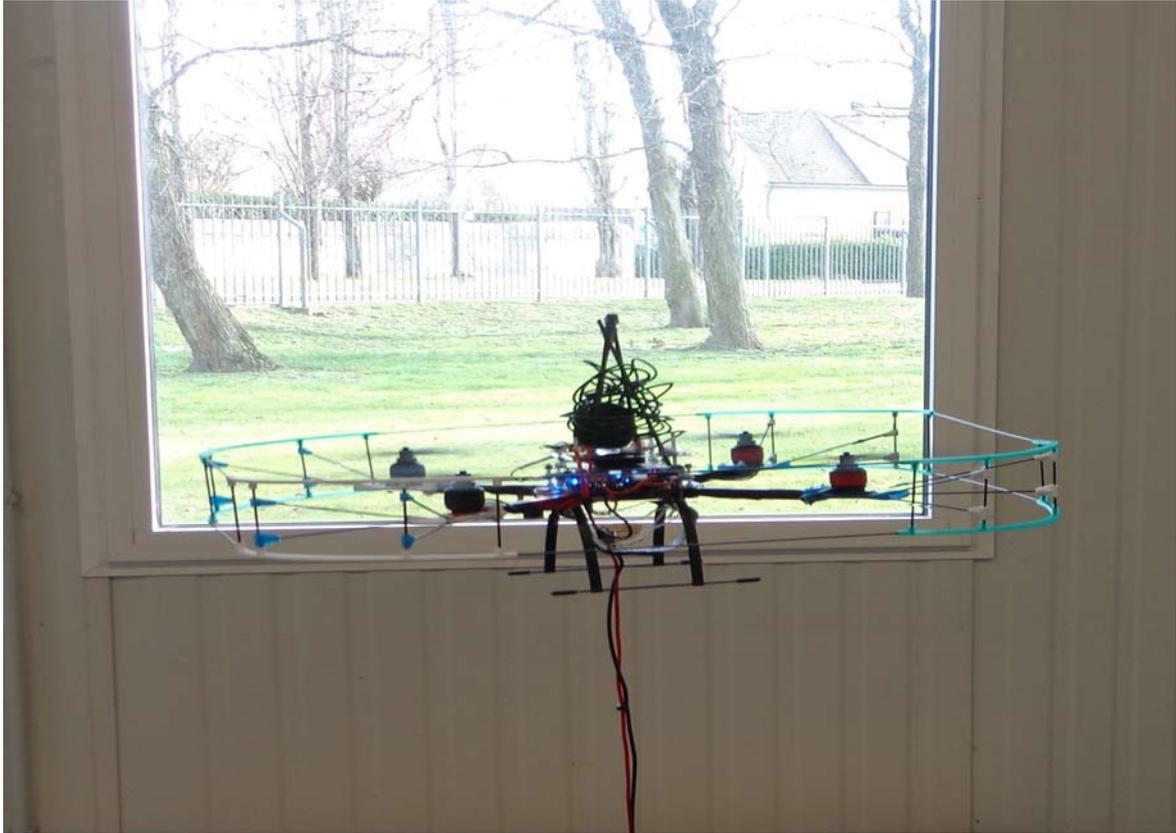


Figure 2-1: Quadrotor in flight.

## 2.1 Conception and Development of a Four Rotor Rotor-Craft for Autonomous Outdoor Navigation

Figure 2-1 shows the quadrotor conceived and constructed for this thesis work. It was specially designed for autonomous outdoor applications and therefore, it was equipped with the necessary sensors, actuators, communication devices and processor for fully embedded control. Such a platform includes the development of a ground station for real time monitoring and online parameter tuning. The main components of this experimental setup are illustrated in Fig. 2-2 and will be briefly presented in the following.



Figure 2-2: Quadrotor main components.

### 2.1.1 UAV Embedded Components

The aluminum and carbon fiber structure of a *QuadroKopter* from *MikroKopter* is chosen along with four brushless electric motors kind *robbe ROXXY BL 2827-35*. They are controlled by four drivers using Power Width Modulation (PWM). This configuration has a total payload of approximately 2Kg, and is suitable in size and weight for outdoor missions where wind perturbations demand bigger drones for robustness, and environment uncertainties require heavier equipment. This structure is augmented with a protection, designed and constructed in our lab using carbon fiber sticks and small plastic pieces made with the help of a 3D printer. This is important to guarantee safe operation close to humans and add robustness against crashes.

For processing the sensors signals, computing the control laws and the actuators management, a Digital Signal Processor (DSP) kind *tsm320f2812* from *Texas Instruments* is employed [5]. Its main characteristics are summarized in Table 2.1, where its computational speed of 150MHz and small size of 25.4mm × 25.4mm are appropriated for this kind

Frequency	150KHz
CPU	32-bit
Input voltage	3.3V
RAM	36KB
Flash	256KB
PWM	16
CAP/QEP	6/2
ADC	1 16-Ch 12-Bit
SCI	2
SPI	1
CAN	2
Timers	3 32-Bit
GPIO	56

Table 2.1: DSP main features.

of applications. It is programmed in C language for automatic attitude stabilization using a linear Proportional Derivative (PD) controller.

A battery *Pro-Tronik* of 6000mAh powers the flying vehicle and allows a time flight of about 15 min. The overall system weights about 1.38Kg including the battery of 0.46Kg, and has a size of 61cm × 61cm, considering the protection structure.

The Inertial Measurement Unit (IMU) is certainly the most important sensor, since it provides estimation of the attitude and angular velocities, which are vital for the UAV stabilization. It is normally conformed by three axis accelerometers and gyroscopes to measure linear accelerations and angular speeds, respectively, as well as a magnetometer. From those measurements it estimates the orientation and angular speed, usually applying an Extended Kalman Filter. In this case the *MicroStrain 3DS-GX3-45* IMU is used due to its high performance. It provides the required information to the DSP via serial communication at a rate up to 150Hz. Furthermore, it includes a GPS fused to the inertial information, with an EKF in a loosely coupled scheme. However, this comes at a very expensive price, hence it is advised to have a separate GPS and IMU sensors for future implementation. Alternative solutions, cheaper but still adequate, can be easily found on the market and should be used to reduce the economic cost.

For the altitude measurement, a *Maxbotix LV-EZ4* ultrasonic range finder pointing to the ground is used. It generates analog signals proportional to the distance to the ground in a

range from 15.24cm to 645cm and a resolution within 2.54cm. Note that this is only useful for mostly flat ground surfaces. An Analog to Digital Converter (ADC) is used in the DSP to read the altitude signal, nevertheless, it results in a very noisy measurement, reason why a low-pass recursive filter type Chevishev I was implemented. As for the altitude speed, it is estimated by means of a Luenberger observer. Similar techniques are used for the other experimental platform and will be explained in detail in section 5.4.2. Both altitude measurement and estimated altitude speed are used for automatic altitude regulation using a PD control implemented in the DSP, along with automatic take-off and landing routines.

Other sensors are envisioned to be added to this system in order to extend and improve its characteristics. For example, a pressure sensor can be used to estimate the altitude in a wider range. Also, visual feedback with a video camera is always desired, and can be used to implement computer vision algorithms. Finally, an optic flow sensor can provide the UAV with translational speed estimations to complement the GPS information and improve the system localization.

Communication to the ground station is achieved with two wireless communication modules *XBee pro*, one embedded on the quadrotor and the other connected to a ground computer using serial communication. The UAV sends important information of its status to the ground station at a rate of 100Hz, such as its attitude, position and other sensor measurements for ground monitoring. Meanwhile, the ground station computer sends high order commands on demand, for taking-off and landing, changing the operation mode or switching through different control laws, as well as modifying online the control parameters and references, or even more, for manual piloting the drone using a joystick.

### **2.1.2 Ground Station**

Finally, a Graphical User Interface (GUI) was particularly designed and implemented for this platform, using the Qt libraries in C++ language on a ground computer running a linux based Operative System (OS). It is composed by three main tabs depicted in Figures 2-3 to 2-5. The first one on Fig. 2-3 is intended for tuning online the control parameters and references. This powerful tool aids enormously the control design and calibration process.

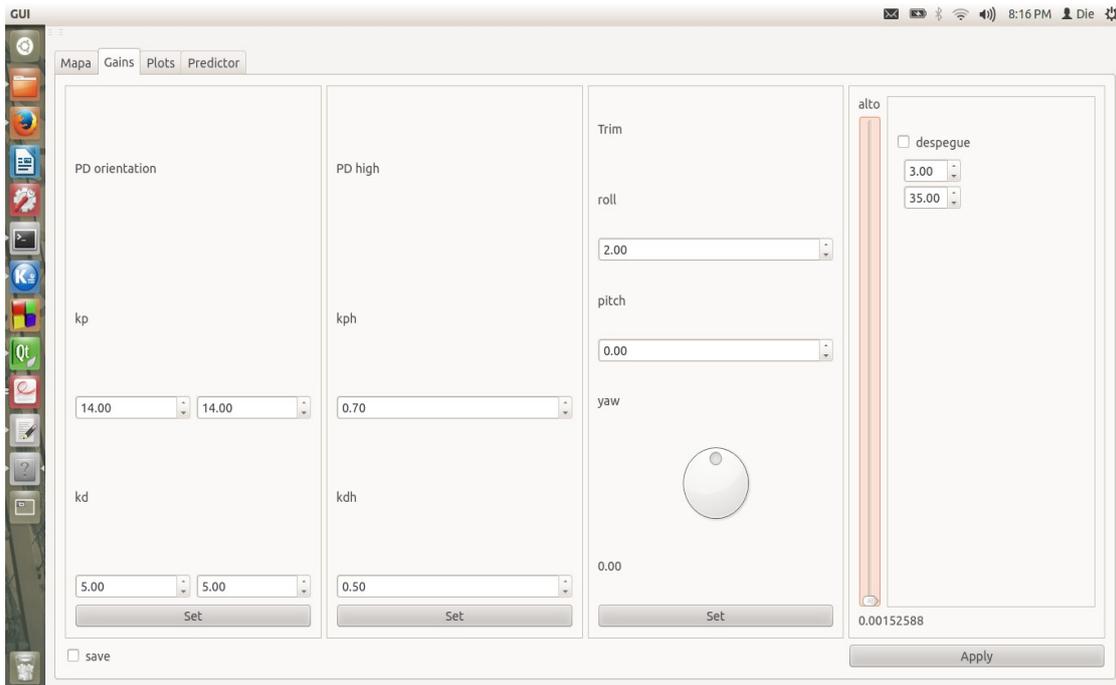


Figure 2-3: Real time parameter tuning.



Figure 2-4: Real time plots.

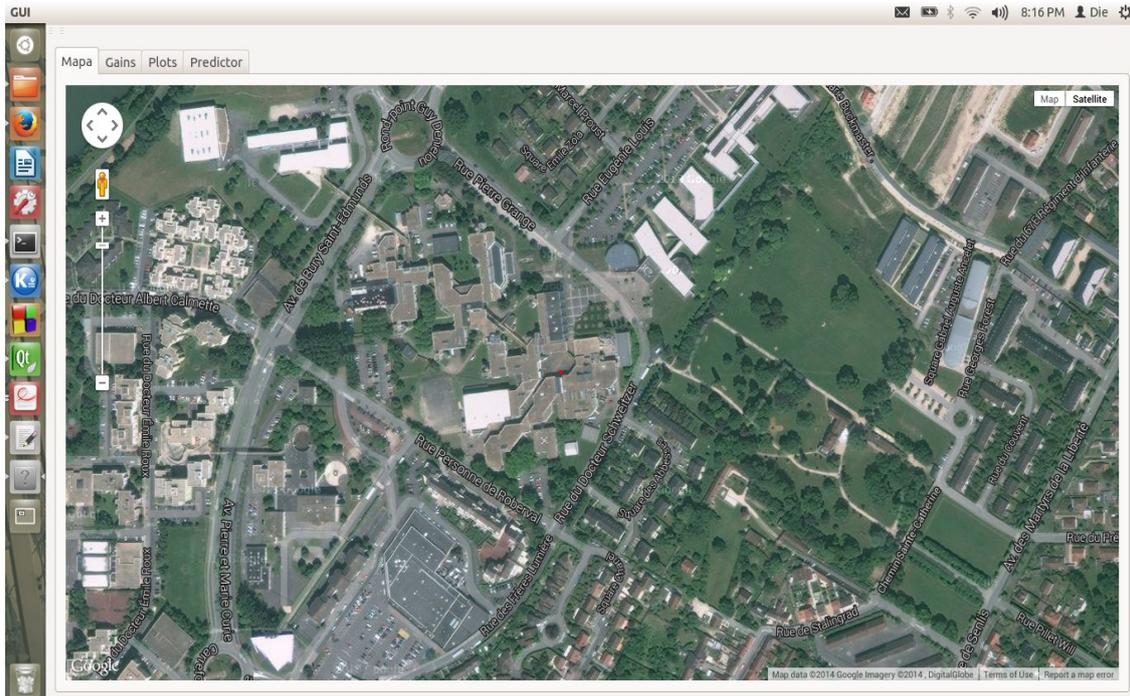


Figure 2-5: GPS localization with *Google Maps*.

In addition, high level commands can be given from this tab, such as automatic take-off and landing, besides the option for save the flight data to a file compatible with *MATLAB*. The next tab (Fig. 2-4), has as purpose to display in real-time plots of the information of interest for monitoring the performance of the system. The third tab (Fig. 2-5) offers an useful graphic application to localize the drone in real-time with *Google Maps*, using the GPS information. Such GUI provides an easy and friendly way to control, monitor and adjust the behavior of the system.

## 2.2 Integration of Available Technologies for a non Expensive Navigation Testbed on ROS

Fixing a “homemade” quadrotor is a time consuming and expensive task, because of the high cost and fragility of the sensors and other electronic components embedded on the UAV. In addition, evaluating new algorithms results to be a risky mission where the ex-

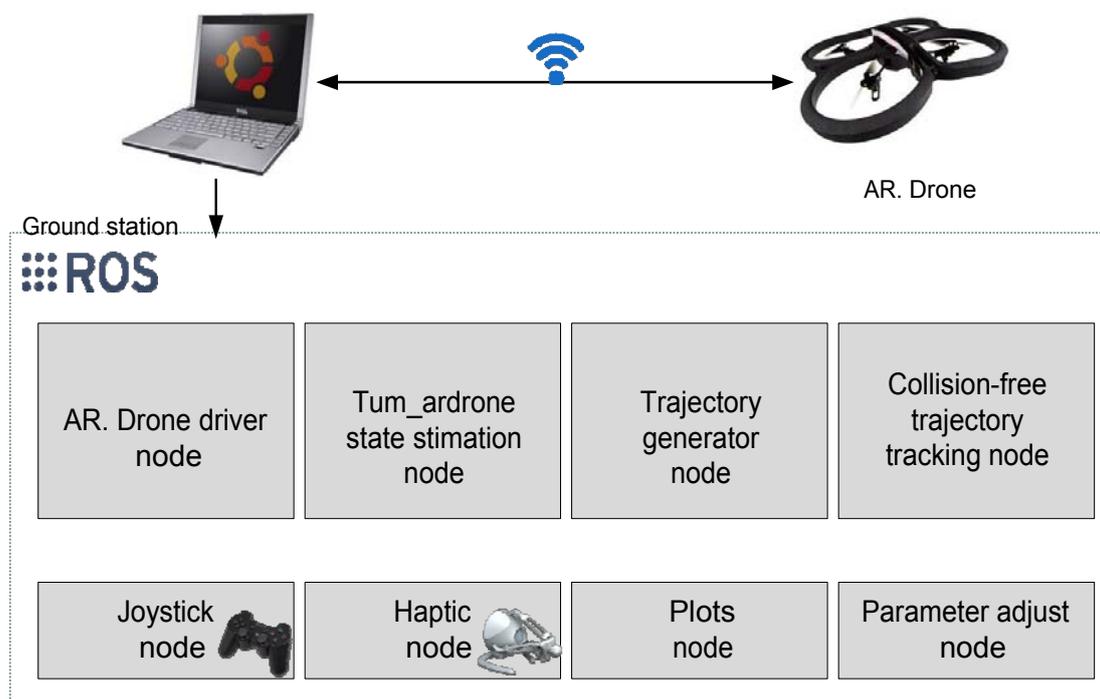


Figure 2-6: Navigation testbed on ROS.

perimental platform is continuously in danger of falling and crashing due to bugs in the implementation or to a bad parameter tuning. Henceforth, a second experimental platform was built to reduce the implementation time and economic cost. It is conformed by an inexpensive commercial quadrotor type *AR.Drone*, wireless connected to a ground computer where all the extra computations are executed using the middleware *ROS*, as illustrated in Figure 2-6. Such a testbed follows and extends the ideas presented in [28], and offers an excellent alternative for fast and safe validation of new proposed control strategies, state observers and computer vision algorithms.

### 2.2.1 Commercial Quadrotor

The *AR.Drone* is a well-known non expensive commercial quadrotor (price $\approx$  \$300), which can be used safely close to people and is robust to crashes. It measures 53 $\times$ 52cm and weights 0.42Kg. It is equipped with three-axis gyroscopes and accelerometers, an ultrasound altimeter, an air pressure sensor and a magnetic compass. Also, it provides video

streams from two cameras, the first one is looking downwards with a resolution of  $320 \times 240$  pixels at a rate of 60fps, and is used to estimate the horizontal velocities with an optic flow algorithm. The second camera is looking forward, with a resolution of  $640 \times 360$  at 30fps, and is used for the monocular vision algorithms.

However, neither the software nor the hardware can be modified easily from the *AR.Drone*. It includes an internal on-board autopilot to control roll, pitch, altitude velocity and yaw rotational speed ( $\varphi$ ,  $\theta$ ,  $\dot{z}$  and  $\dot{\psi}$ ), according to external references. These references are considered as control inputs and are computed and sent at a frequency of 100 Hz. All sensor measurements are sent to a ground station at a frequency of 200 Hz, where vision localization and state estimation algorithms are calculated in real-time on ROS.

### 2.2.2 Ground Station

A ground computer is responsible for recovering the drone's information, real-time monitoring the system state, online parameter tuning, compute the control laws, state observers and computer vision algorithms, as well as giving high level commands and switch between the different operation modes, among others. In order to do so, it is used the *Robot Operative System*, better known as ROS. ROS is a set of open source libraries and tools that help to build robot applications, from drivers to the last developed algorithms, with powerful developer tools. It serves as a middleware to assist the different programs, so-called nodes, to interact and communicate among themselves and the operative system, by means of messages and services. As can be appreciated in Figure 2-6, the main nodes utilized in this work are:

- **AR.Drone driver node:** This node is in charge of communicating with the *AR.Drone* via *Wi-Fi*. Here, information from the quadrotor embedded sensors is recovered at a rate of 200Hz, as well as the video streams from the two attached cameras. Also, this node controls the drone by sending the desired references to the internal autopilot, along with take-off and landing commands.
- **Tum\_ardrone state estimation node [28]:** It applies the PTAM algorithm to estimate the quadrotor pose with respect to a visual scene (further details are given in

chapter 3).

- **Trajectory generator node:** This is used to provide time varying trajectories to be used as desired position references.
- **Trajectory tracking node:** In this node are computed the trajectory tracking control strategies. This node is of particular interest for this work, since from here it is possible to easily modify the control laws or implement new ones.
- **Joystick node:** Manual piloting the UAV is also possible through this node, for security reasons, in case of a system failure or simply to give high level commands like take-off and landing.
- **Haptic node:** It allows an alternative operation mode for assisted teleoperation with force feedback.
- **Plots node:** The *rgt\_plot* node provided with *ROS* facilitates the monitoring task, vital for most mobile robot applications. In it, it is possible to easily plot in real time any available variable.
- **Parameters tuning node:** Online parameter tuning is possible thanks to the *rgt\_reconfigure* node included with *ROS*.

Those nodes represent the basic operation of the testbed. However, other nodes can be programmed and easily added to the system to expand or even substitute some of the described nodes, depending on the application and desired behavior of the system. For example, for the face follower application presented in chapter 5, different nodes were developed according to the problem.

### 2.2.3 Haptic Teleoperation

A different operation mode is allowed for haptic assisted teleoperation. It consists in a semiautonomous navigation mode where the user gives angular references to the internal autopilot on the quadrotor through an haptic device. Such a device is able to assist the

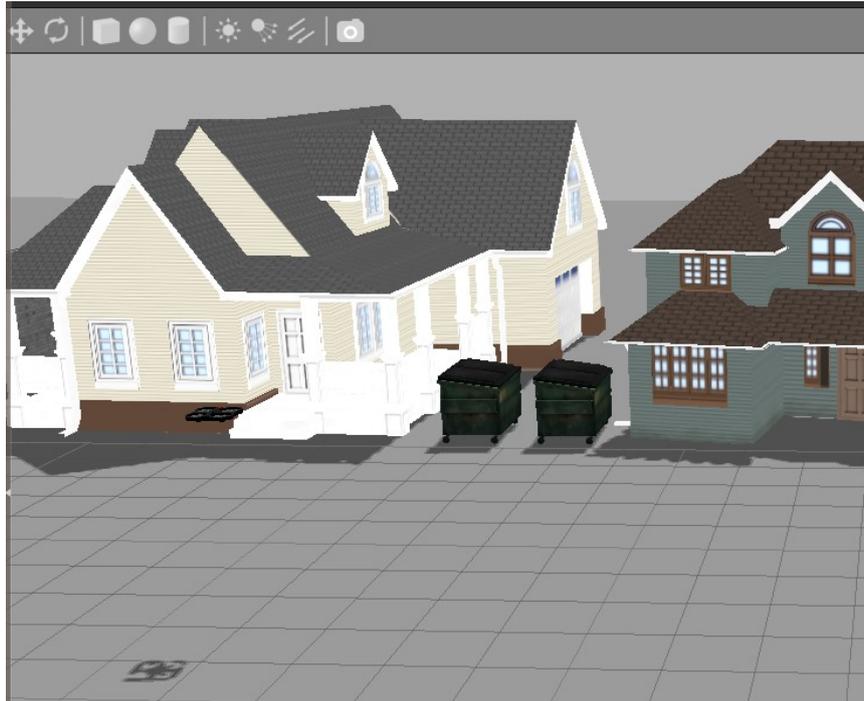


Figure 2-7: Quadrotor simulator for ROS.

human operator by feeding back essential information for safe operation. In this case, feedback force is used to prevent about possible frontal collisions, as it will be discussed in chapter 4.3.2. For this matter, the low-cost Novint Falcon haptic device is employed. It consists of a three degrees of freedom haptic interface in a delta configuration, able to feedback forces up to  $8.9\text{ N}$  to a human operator. Its touch workspace is about  $10 \times 10 \times 10\text{ cm}$ , with a resolution bigger than  $400\text{ dpi}$ . An algorithm to connect this haptic device with the vision algorithm was developed in order to improve the flying experience.

#### 2.2.4 Simulator

A *Gazebo* based simulator for the *AR. Drone* as showed in Figure 2-7, also available as open source software for *ROS*, was employed to safely test the proposed algorithms in a realistic way, before using the real quadrotor. Also, it provides a decent approximation of the optimal values for the parameters involved, helping the final tuning in the real platform.

## 2.3 Motion Capture System for Validation

An *Optitrack/TrackingTools* motion capture system serves for validation in some of the realized experiments, providing accurate information of the localization and orientation of a rigid object which is used as “ground truth”. It is composed by 12 infrared cameras *Prime 13* with a resolution of 1.3 *MP* ( $1280 \times 1024$ ) and a frame rate of 240 *FPS*, to detect the infrared light reflected by special spherical markers attached to the body of interest. The *TrackingTools* software is able to determine the pose of the object with a precision within millimeters, if at least three cameras detect the same three markers. The obtained data is sent at 120 *Hz* using the *VRPN* (Virtual Reality Peripheral Network) protocol. It is important to remark that this motion capture system is used only to compare our results and validate them, but in any case our proposed experimental setup depends or is constrained to it, hence, we are able to easily reproduce our experiments in a wide range of different environments, indoor and outdoor.

# Chapter 3

## UAV Data Fusion for Localization

This chapter presents a study of data fusion techniques, in the search of enhancing the UAVs pose estimation provided by commonly used sensors. Two strategies are evaluated in particular, the Extended Kalman filter and the Particle filter. Both estimators are adapted for the quadrotor system, taking into account noisy measurements of the UAV position, velocity and orientation. Simulations show the performance of the developed algorithms while adding error from real GPS measurements.

### 3.1 Introduction

Quadrotors have gained great relevance in the last years thanks to their huge potential in many civilian and military applications, in a cheap way and without risking human lives in dangerous situations. Some of these applications are exploration, surveillance, search and rescue, package deployment and video recording, among others. Several stabilization and trajectory tracking control laws have been proposed for this kind of systems and validated through simulations and experiments in indoor applications, relying on a good measurement of the position and velocity. To cite some examples, [20] proposes an attitude stabilization control strategy for hover flight using nested saturations, while in [80] a sliding mode approach is used to accomplish position control of a quadrotor. In [38] a tra-

jectory tracking control by means of a discrete time feedback linearization control scheme is proposed.

However, big effort is still required to achieve autonomous flight for outdoor applications, due to presence of external perturbations, especially the wind, and the lack of a good measurement for the position and velocity. Inexpensive GPS sensors can provide an estimation of the position and velocity, however, the errors, of 2m at best, and their low measurement rate of about 5Hz, are not suitable for precise applications and can interfere with the system stability, even more, GPS can easily lost its signal leaving the system without a position measurement. Another alternative widely studied are the optical flow sensors which use computer vision algorithms for estimating the motion velocity of a system, but they are noisy and sensibles for lighting changes. Data fusion algorithms are an interesting solution to this problem, they take information from multiple sensors, especially GPS, cameras and IMU, to improve the estimation of the position and velocity. The Kalman filter and its variations are the most popular approaches, for example, in [57] an observer-control scheme for quadrotors using the EKF is proposed and tested in real time indoor experiments, using data from an optic flow algorithm and an IMU. In [39], a Robust Adaptive Kalman Filter is proposed for estimation of UAV dynamics in the presence of sensor faults, see also [33], [25] and [76]. However, these techniques assume that the process and measurement noise are Gaussian, which is not the case of a GPS in a loosely coupled scheme.

The great advances in micro controllers processing capacity allow to explore other alternatives. The PF presents an interesting option to these problems, at the cost of computational expense, because it is not restricted to Gaussian noise. In [64] a vision based algorithm using a particle filter tracking for estimating the position of a quadrotor is presented.

In this chapter two different data fusion techniques are studied aiming to solve the localization problem for UAVs performing outside missions. On the one hand an EKF and on the other hand a PF. Both techniques are explored trough simulations, considering a quadrotor equipped with multiple commonly used sensors. Particularly, an optic flow sensor can estimate the traslational speeds, while an IMU provides the orientation and

angular rate, and finally a GPS is considered for position measurement. In addition, real GPS data was collected and the obtained error was added to the simulated position.

The chapter is composed as: section 3.2 presents a review of the EKF theory, as well as the model adaptation. Performance is tested through simulations. Section 3.3 states the PF algorithm for the considered system and it is evaluated in simulations.

## 3.2 Extended Kalman Filter Data Fusion

This section deals with the issue of estimating the position and velocity of the quadrotor by means of a data fusion algorithm using an EKF, taking information from multiple sensors whose measurements are inaccurate and noisy.

The EKF is a well known, powerful and widely used estimation algorithm for the case of nonlinear systems of the form [35]

$$\chi_{k+1} = f(\chi_k, u_k) + \omega_k \quad (3.1)$$

$$Z_k = g(\chi_k) + v_k \quad (3.2)$$

where  $\chi$  is the state vector and  $Z$  is the measurement vector. The process and measurement noise  $\omega$ ,  $v$  are supposed to be Gaussian with known covariance matrices  $Q$  and  $\Theta$ , i.e.

$$\omega \sim N(0, Q) \quad (3.3)$$

$$v \sim N(0, \Theta) \quad (3.4)$$

Then, the a priori states estimate are given by

$$\hat{\chi}_{k+1|k} = f(\hat{\chi}_{k|k}, u_{k|k}) \quad (3.5)$$

and the estimated outputs is

$$Y_{k+1|k} = g(\chi_{k+1|k}) \quad (3.6)$$

The a priori error covariance can be calculated as

$$P_{k+1|k} = AP_{k|k}A^T + Q_k \quad (3.7)$$

where  $A = [\frac{\partial f}{\partial X}(\hat{X}, u)]$ . Now, the Kalman gain matrix can be determined

$$K_{k+1|k} = P_{k+1|k}G^T(GP_{k+1|k}G^T + \Theta_{k+1})^{-1} \quad (3.8)$$

with  $G = [\frac{\partial g}{\partial X}(\hat{X})]$ . After measuring the process, the a posteriori state estimate  $\hat{X}$  and error covariance  $P_k$  are respectively

$$\hat{X}_{k+1|k+1} = \hat{X}_{k+1|k} + K_{k+1}(Z_{k+1} - Y_{k+1|k}) \quad (3.9)$$

$$P_{k+1|k+1} = (I - K_{k+1|k}G_{k+1})P_{k+1|k} \quad (3.10)$$

### 3.2.1 Model Adaptation

Let us retake the model (1.3), (1.4). Also, consider the state vector

$$X = [\xi \ \dot{\xi} \ \Phi \ \dot{\Phi}]^T \quad (3.11)$$

and the measurement vector

$$Z = \chi + v = [\xi_{GPS} \ \dot{\xi}_{OF} \ \Phi_{IMU} \ \dot{\Phi}_{IMU}]^T \quad (3.12)$$

it is, the position measured by a GPS, the velocity obtained by a vision system and an optic flow algorithm (an extra altitude sensor is required for the z coordinate) and the orientation and angular speed given by an IMU.

The discrete time process dynamics, considering a sample time  $T_s$  small enough is given

by

$$\begin{aligned}
 \chi_{k+1} = & \begin{bmatrix} \dot{\xi}_k T_s + \xi_k \\ (\frac{1}{m} T_k R_k e_3 - g e_3) T_s + \dot{\xi}_k \\ \dot{\phi}_k T_s + \phi_k \\ \bar{\Gamma}_k T_s + \dot{\phi}_k \end{bmatrix} + \omega_k
 \end{aligned} \tag{3.13}$$

where the  $\bar{\Gamma} = [\bar{\tau}_\phi \ \bar{\tau}_\theta \ \bar{\tau}_\psi]^T$  is a simplified control torque which includes the rotational dynamics such that

$$\ddot{\phi} = \bar{\Gamma} \tag{3.14}$$

and together with the total thrust  $T$  conforms the system inputs.

Since the GPS sample frequency is slow (5Hz) with respect to the others sensors, and in order to deal with the GPS loss signal, the measured position is updated in the following form

$$\xi_{GPSk} = \xi_{GPSk-n} + n T_s \dot{\xi}_k \tag{3.15}$$

where  $n$  is the number of sample times  $T_s$  passed since the last valid GPS data.

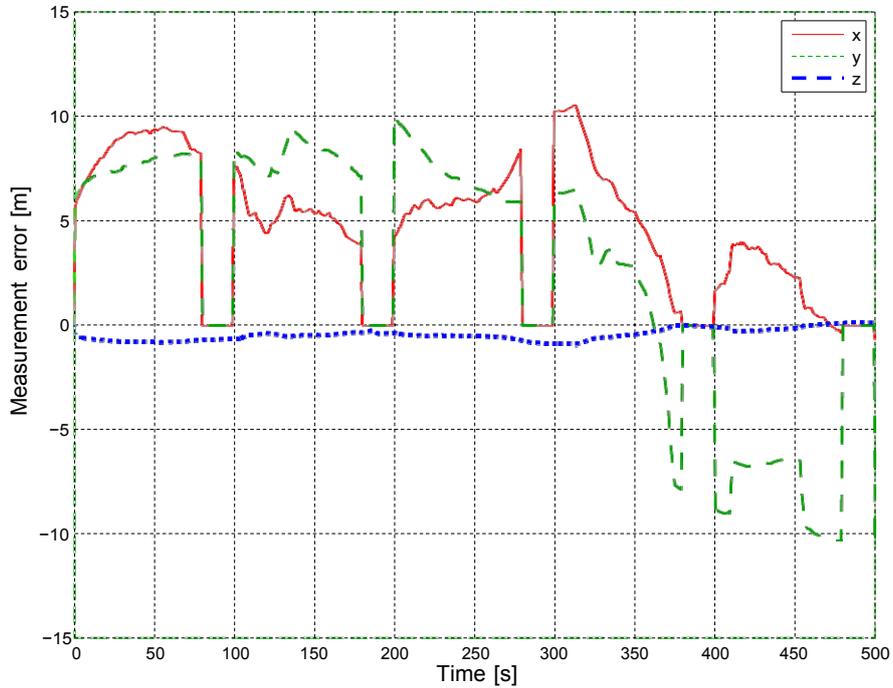


Figure 3-1: GPS measurement error.

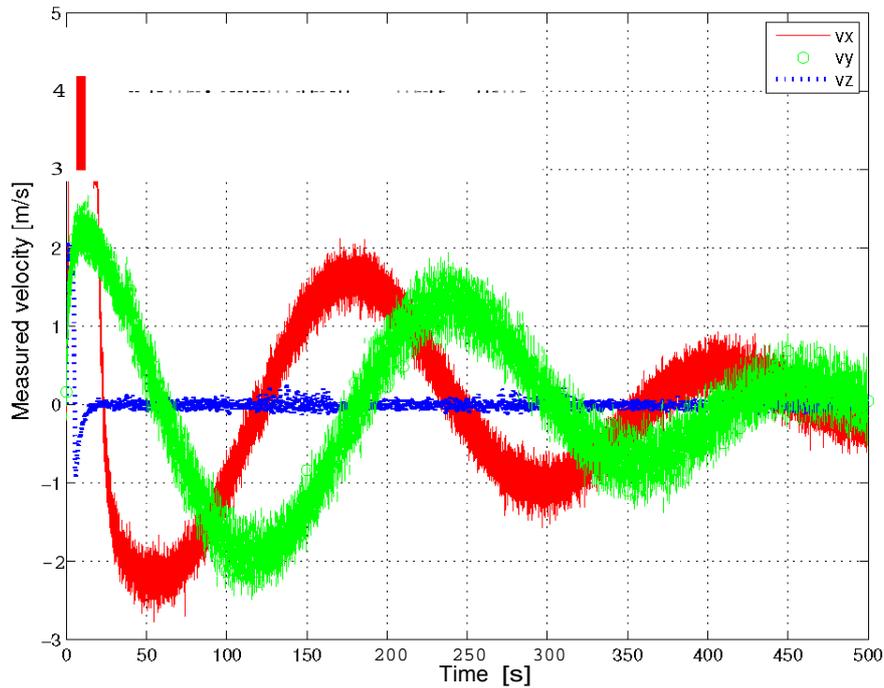


Figure 3-2: Measured velocity.

### 3.2.2 Simulations

Simulations were carried out to observe the behavior of the proposed observer scheme with external perturbations and inaccurate measurements from a GPS sensor. The considered scenario is a quadrotor following autonomously a spiral trajectory in the  $XY$  plane, at a constant altitude of  $3m$ , while the embedded sensors read corrupted measurements. In order to introduce the errors from the GPS in a more realistic way, real GPS data was previously collected offline under bad conditions for the GPS receiver, it is in an urban environment surrounded by buildings, then this real data was added to the simulated actual position of the quadrotor. Fig. 3-1 shows such GPS measurement error obtained from offline experiments where a  $u-blox$  GPS was read at  $5 Hz$  while staying static at the same position in an urban environment, this allows us to observe the GPS drift, furthermore, the zero values every 100 seconds were added to represent signal losses. A second sensor was considered to measure the velocity, with some white noise (Fig. 3-2 shows the noisy measured velocity), this can represent for example the measure from an optic flow sensor.

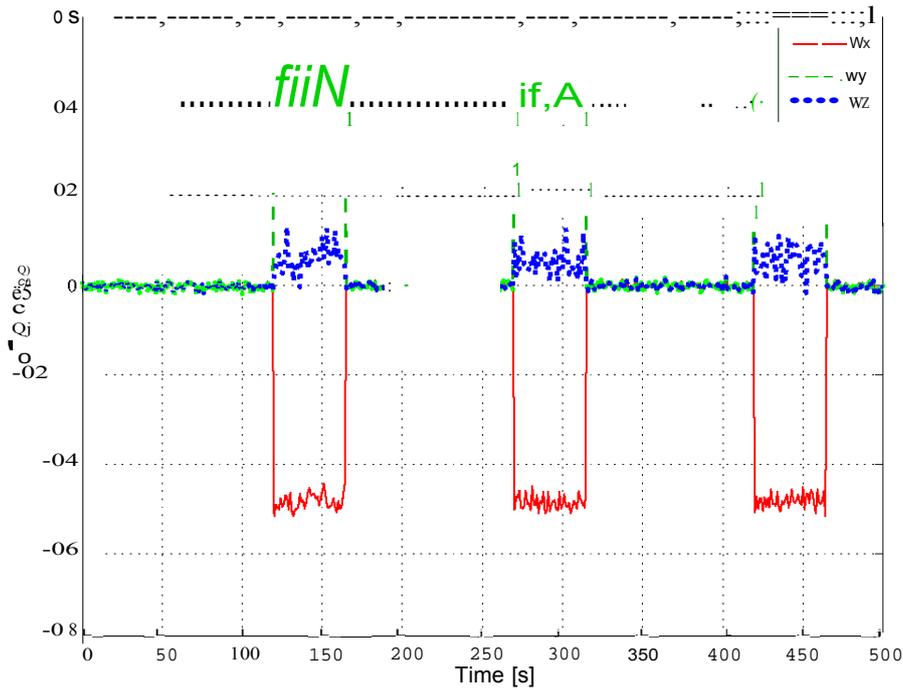


Figure 3-3: External perturbation  $w$ .

Also, an external perturbation  $w$  as the one showed in Figure 3-3 is added to the system. Even more, the estimated state vector given by the EKF is used to close the loop in the trajectory tracking control law, in this case a high order sliding mode [54].

The performance of the estimation algorithm can be studied through Fig. 3-4, where the estimation error is presented. It can be noticed that the estimator improves considerably the measured data and perfectly handles with the missing data while GPS signal is lost. Fig. 3-5 shows the orientation of the helicopter. Position and tracking errors are presented in Figs. 3-6 and 3-7, showing good tracking results despite the presence of the measurement error and external perturbations. Finally, Figs. 3-8 and 3-9 contain the real and desired position in the XY plane and in the three dimensional space. The steady state error is attributed to the measurement error as can be observed in this two last figures, where the estimated position is very close to the desired one.

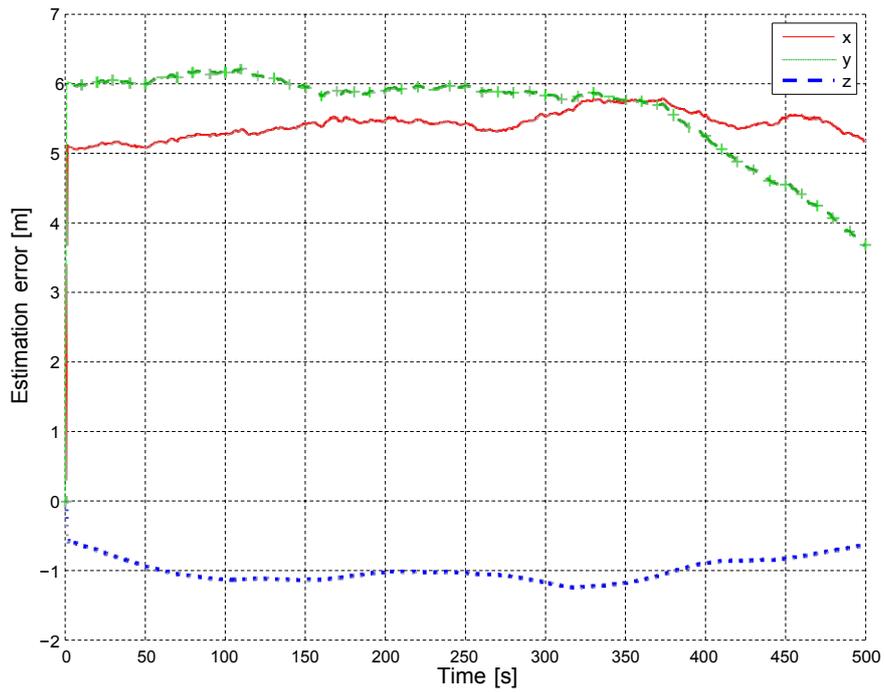


Figure 3-4: Estimation error.

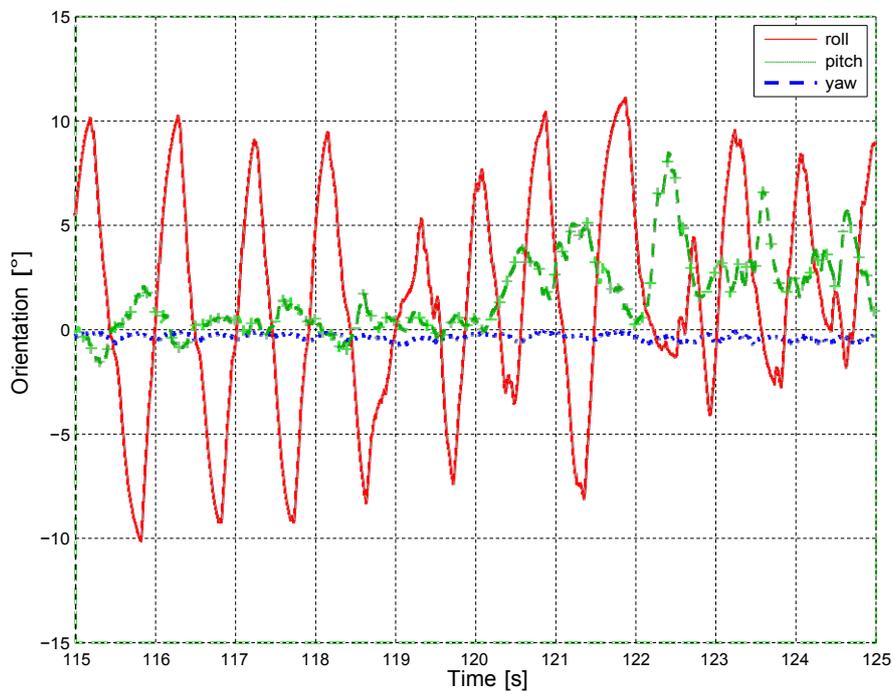


Figure 3-5: Orientation.

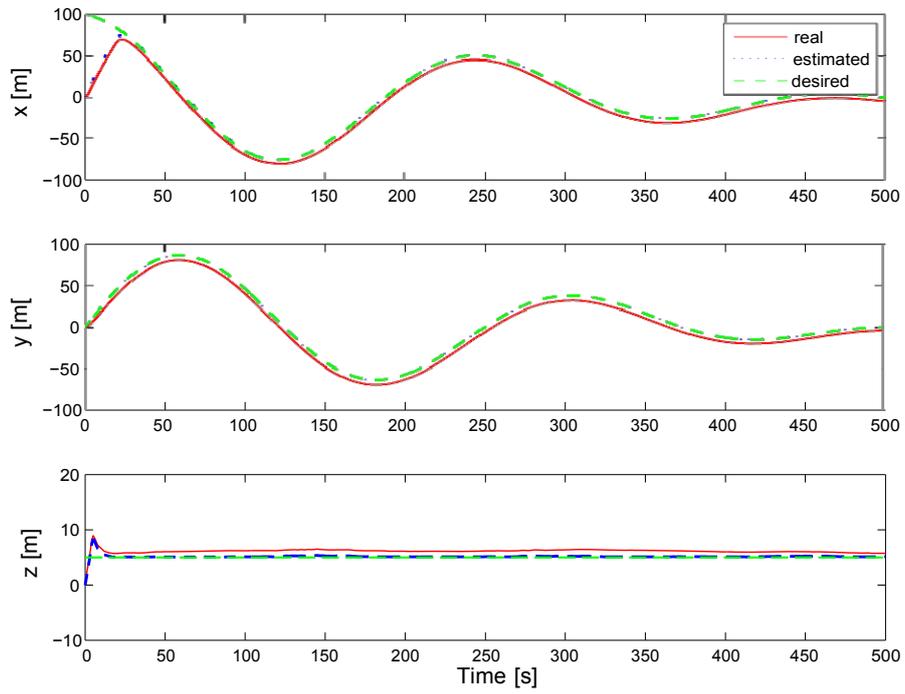


Figure 3-6: Position.

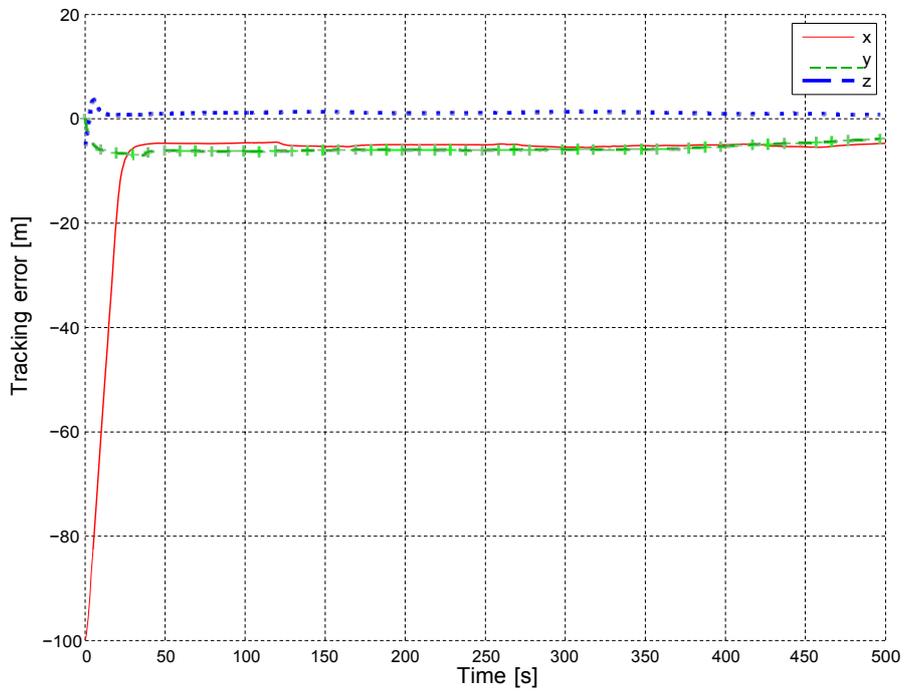


Figure 3-7: Tracking error.

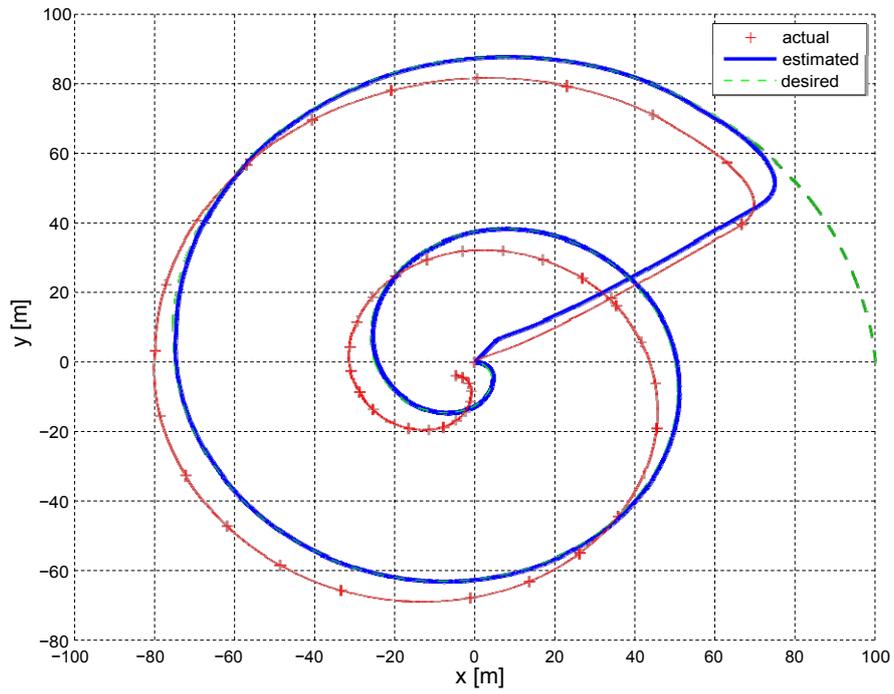


Figure 3-8: XY plane.

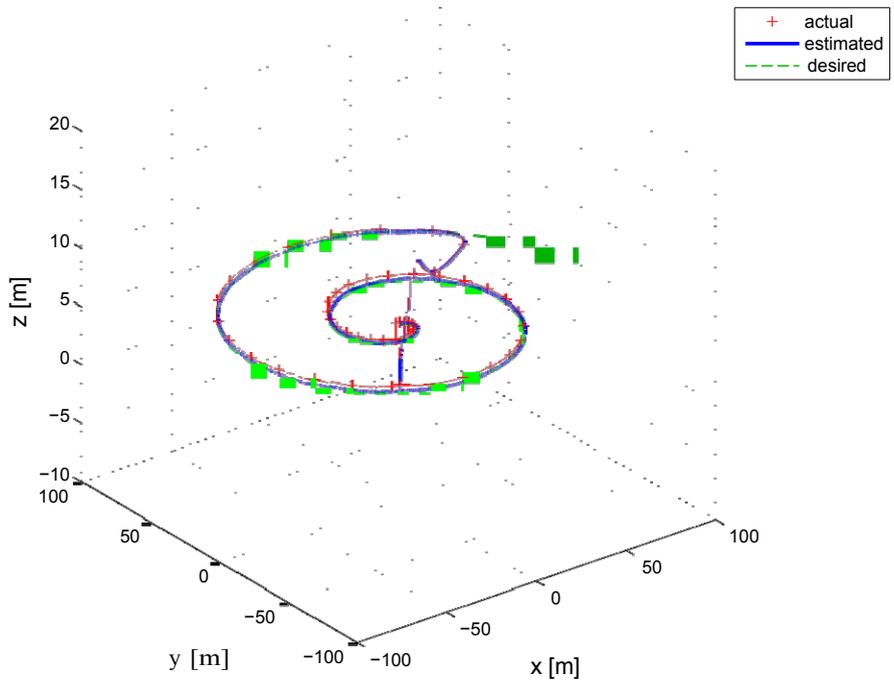


Figure 3-9: 3D.

### 3.3 Particle filter

Consider again the dynamic model at equations (1.3) and (1.4), and the state vector (3.11). Then, with a sampling time  $T_s$  small enough, the discrete time evolution model is given by

$$\chi_k = \begin{bmatrix} \xi_{k-1}T_s + \xi_{k-1} \\ (\frac{1}{m}T_{k-1}R_{k-1}e_3 - ge_3)T_s + \dot{\xi}_{k-1} \\ \Phi_{k-1}T_s + \Phi_{k-1} \\ \bar{\Gamma}_{k-1}T_s + \dot{\Phi}_{k-1} \end{bmatrix} + \varpi_k \quad (3.16)$$

where  $\varpi$  stands for the model noise with a probability density  $p_\varpi$ .

From the measurement vector (3.12), the more general observation model is

$$Z_k = \chi_k + \vartheta_k \quad (3.17)$$

with  $\vartheta$  representing the measurement noise with a probability density  $p_\vartheta$ .

The particle filter (PF) or sequential Monte Carlo technique is a kind of recursive Bayesian filter for state estimation [21]. It consists in generating  $N_p$  particles according to the a priori probability, with an importance weight  $W$  defined as

$$W_k = \frac{p(Z_{1:k}|\chi_{0:k})p(\chi_{0:k})}{q(\chi_{0:k}|Z_{1:k})} \quad (3.18)$$

where  $p(Z|\chi)$  is the probability density function of  $Z$  given  $\chi$ , and  $q(\chi_k|Z_k)$  is the so-called proposal distribution.

At each iteration, the outputs are measured by the sensors and the weights can be updated recursively, using the bootstrap method, by the following expression

$$W_k^i = W_{k-1}^i p(Z_k|\chi_k^i) = W_{k-1}^i p_\vartheta(Z_k - \chi_k^i) \quad (3.19)$$

$$i = 1, \dots, N_p$$

for a normal probability distribution in the measurement, with covariance matrix  $Q$

$$W_k^i = W_{k-1}^i e^{(-\frac{1}{2}(Z_k - \chi_k^i)Q^{-1}(Z_k - \chi_k^i))} \quad (3.20)$$

normalizing

$$w_k^i = \frac{W_k^i}{\sum_{i=1}^{N_p} W_k^i} \quad (3.21)$$

Then, the set of particles evolves following the evolution model (3.16), with  $\varpi_k^i$  randomly generated according to  $p_{\varpi}$ .

In order to avoid the weight degeneracy problem, a resampling step is required. In order to do so, the effective sample size  $N_{eff}$  is employed as a measure of the number of active particles

$$N_{eff} = \frac{1}{\sum_{i=1}^{N_p} (W_k^i)^2} \quad (3.22)$$

it can be noticed that this value is maximum ( $N_{eff} = N_p$ ) when all the particles have the same weight, and minimum ( $N_{eff} = 1$ ) when only one weight is different from zero. If the effective sample size is lower than a certain threshold  $N_{eff} < N_{th}$ , the resampling step is performed by means of the Kitagawa method [45].

The estimated state is given by the particles barycenter, it is

$$\bar{\chi}_k = \sum_{i=1}^{N_p} W_k^i \chi_k^i \quad (3.23)$$

Once more we employ the measurement position update at (3.15).

### 3.3.1 Simulations

Simulations were performed to test the proposed estimation algorithm for a quadrotor with an inaccurate position measurement, which is the case for a not expensive GPS. Similar to the previous case, GPS errors showed in Figure 3-1 were obtained from real experiments and included in the simulations. Optical flow sensors can provide a noisy velocity measurement, hence some white noise was added to the velocity signal.

The performance of the particle filter for position estimation while the UAV evolves in a spiral trajectory can be seen in Fig. 3-10, using  $N_p = 100$  particles and a resampling threshold of  $N_{th} = 10$ . It can be noticed that the PF performs an excellent position estimation despite the poor position measurement, this is clearer from Fig. 3-11 where the estimation error is displayed, which can be compared against the measurement error in Fig.

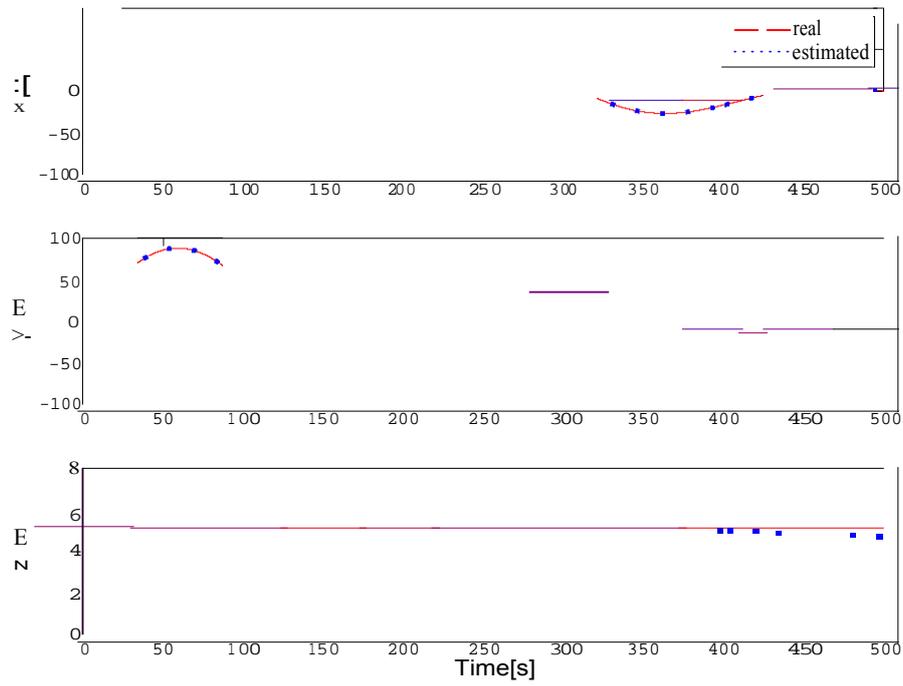


Figure 3-10: Position.

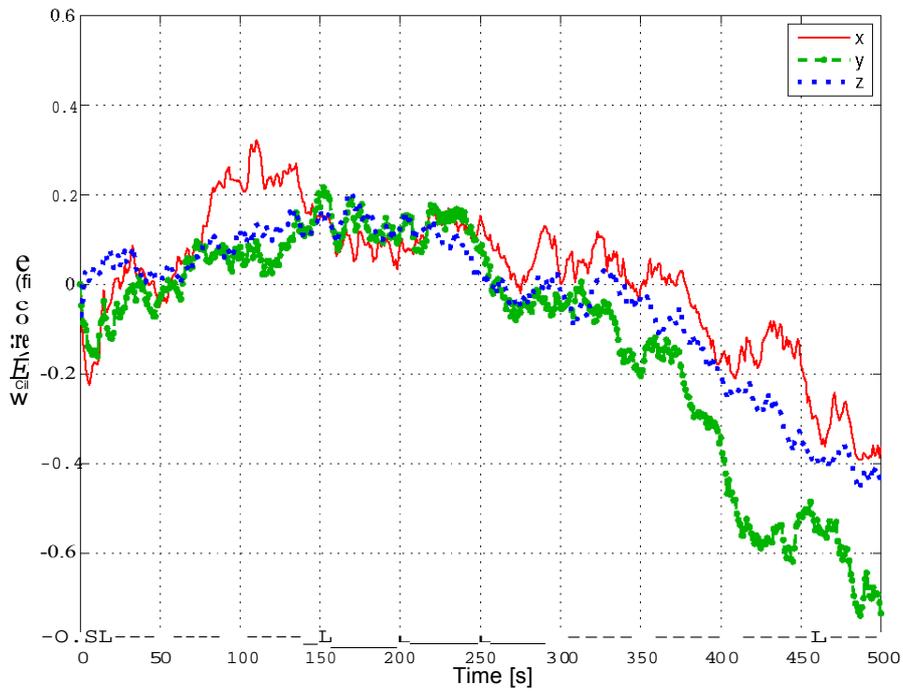


Figure 3-11: PF estimation error.

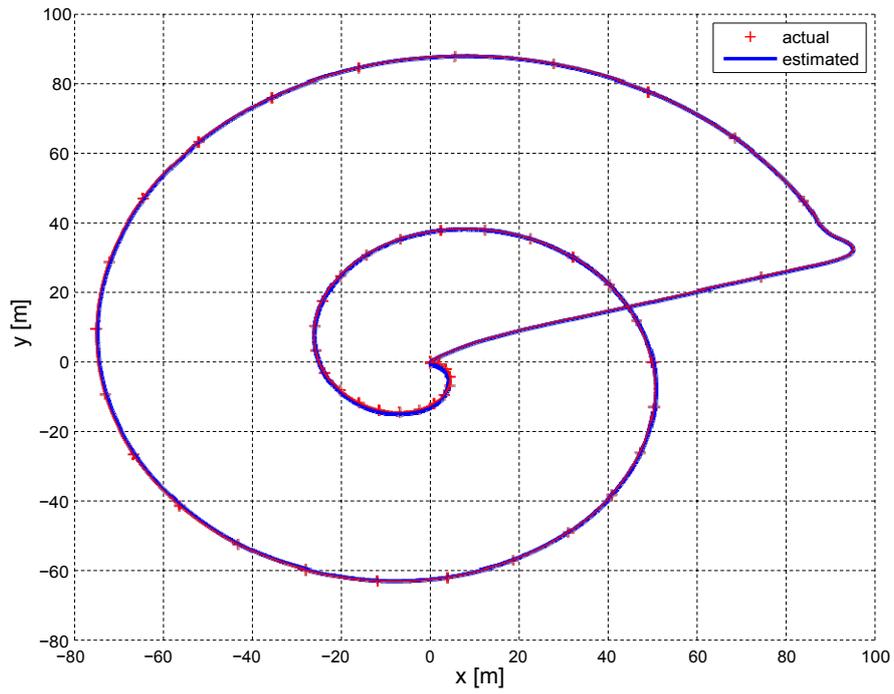


Figure 3-12: XY plane.

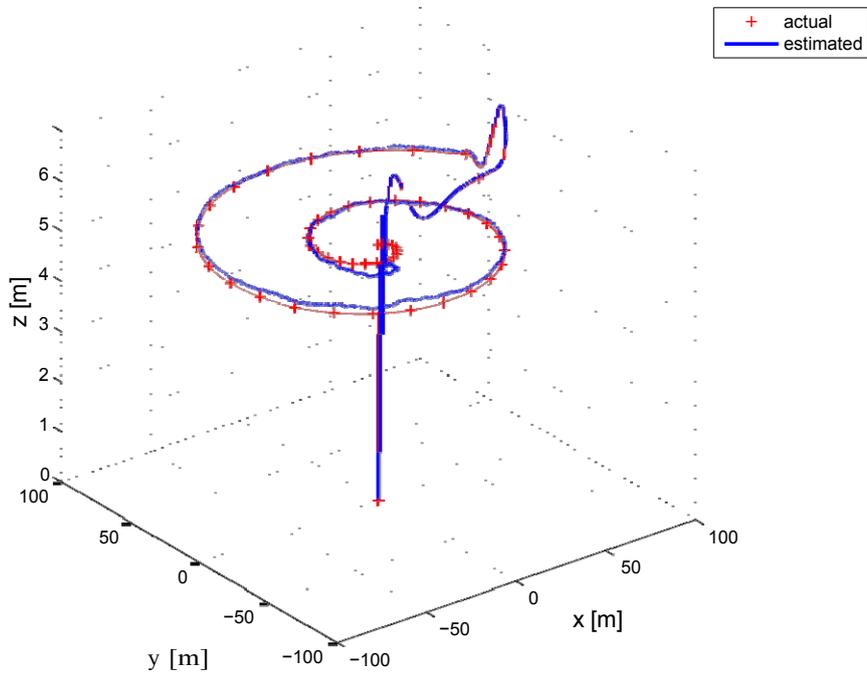


Figure 3-13: 3D.

3-1. It is clear that the position estimation greatly improved the measured one and perfectly handled the signal losses and from measurement errors of until 20 meters are obtained errors smaller than one meter, making it suitable for precision tasks. Finally, Figs. 3-12 and 3-13 contain the XY plane view and the three dimensional one with the real and estimated positions.

Now we are able to do a direct comparison of the two proposed methods by looking into the estimation error for the EKF at Figure 3-4 against the one for the PF at 3-11. Observe here that the errors are much higher for the EKF case. This can be explained due to the fact that the EKF assumes that measurement noise is Gaussian, which is not the case for a GPS in a loosely coupled scheme, see Fig. 3-1. However, the computational cost, and therefore the speed, for a PF with 100 particles is considerably higher than the corresponding one for the EKF, which is undesired for real time embedded implementation.

With all this in mind, we can conclude that, under the formulations proposed here, none of the tested strategies offers a fully satisfactory solution for the localization problem in UAVs. However, both techniques have interesting properties and advantages that can be exploited under certain conditions, depending on the application. This reason motivated the exploration of the alternative techniques presented in the following chapters, but it would be equally interesting to study other data fusion formulations in future work, such as expanding the state vector with slow moving biases to compensate the GPS error in a loosely coupled scheme, or explore the GPS data fusion in a tightly coupled integration instead.



## Chapter 4

# Collision-Free Navigation and Haptic Teleoperation using Monocular Vision

Safe and accurate navigation for either autonomous trajectory tracking or haptic teleoperation of quadrotors is presented in this chapter. A second order Sliding Mode (2-SM) control algorithm is used to track desired trajectories while avoiding frontal collisions in autonomous flight. The time-scale separation of the translational and rotational dynamics allows to design position controllers by giving desired reference in roll and pitch angles, which is suitable for quadrotors equipped with an internal attitude controller. The 2-SM control adds robustness to the closed-loop system. Vision algorithms are employed to estimate the pose of the vehicle using only a monocular SLAM fused with inertial measurements. Distance to potential collisions is detected and computed using the sparse depth map from the vision algorithm. For teleoperation tests, a haptic device is employed to feedback information to the pilot about possible collisions, by exerting opposite forces. The proposed strategies are successfully tested in real-time experiments, using a low-cost commercial quadrotor.

## 4.1 Introduction

Unmanned Aerial Vehicles are being used or envisioned for use in operations where they serve primarily as a platform for a sensor payload. Regardless of their size and mission, all UAVs share the need for sensor or sensor systems which provide an estimate of the vehicle's full state vector (position, orientation, translational velocity and angular rate). These states must be available to the controller at high fidelity and high bandwidth in order to guide and control the aerial vehicle. For precision applications, accurate position, control and guidance systems are always required. In simple terms, the guidance system generates instructions on what state trajectory the UAV should follow in accomplishing its mission. Navigation without reference to some global coordinate system, without a GPS for example, is a pivotal problem in the field of UAVs applications. In many situations, GPS is unavailable or corrupted due to jamming or environmental considerations and then other estimation techniques are necessary to locate the vehicle.

Small UAVs, however, present a unique set of challenges ultimately stemming from the limited payload capacity of the vehicle. Sensing is greatly limited, often including only an IMU and a monocular camera. Furthermore, since small UAVs are typically intended to be inexpensive, the use of high-quality IMUs is precluded, making a purely inertial navigation solution unfeasible.

In the recent years, some interesting works to pose estimate using monocular cameras have appeared in the literature. They are based on the PTAM algorithm to pose estimation presented in [46]. It was originally designed to track a hand-held camera in a small AR workspace. Their resulting system produces detailed maps with thousands of landmarks which can be tracked at frame-rate, with a good accuracy and robustness. In addition, the PTAM algorithm has been successfully applied to UAVs where the absolute scale problem is solved by fusing visual information with inertial and altitude measurements. In [7]-[77] the authors introduce implementations of an onboard vision-based UAV controller for navigation on unknown environments without any external assistance. The drawback in these works is to consider expensive aerial prototypes to get a fully embedded system.

Other teams, contrary to the above cited works, have obtained similar results using



Figure 4-1: *AR.Drone* following a circular trajectory.

only a low-cost commercial UAV coupled with a ground-station to compute externally the algorithms [28], [29]. In these research works the system is composed of a monocular SLAM system, an EKF and a linear control. The contribution was to give a closed-form solution to estimate the absolute scale of the generated visual map from inertial and altitude measurements. Furthermore, they provide their complete implementation as open-source code.

On the other hand, obstacle detection and avoidance are key components for the success of autonomous operation of UAVs. Several teams have explored this matter using different approaches, for example, in [58] the authors proposed a complete system with a multimodal sensor setup for omnidirectional obstacle perception consisting of a 3D laser scanner, two stereo camera pairs, and ultrasonic distance sensors. Detected obstacles were aggregated in egocentric local multiresolution grid maps. Planning of collision-free trajectories and reactive obstacle avoidance were tested in real-time experiments. However, this approach requires a huge payload for the multiple sensors involved.

Monocular cameras arise again as an excellent alternative to detect and avoid obstacles for small UAVs. First works were based on optical flow algorithms [15] or perspective cues [16], however these methods can not handle frontal obstacles well. In [55], the problem of detecting frontal obstacles for an UAV is examined through a method to detect relative size changes in images patches, with highly confident results in experiments. Similarly, in [8] the authors presented an approach that allows a quadrotor with a monocular camera to locally generate collision-free waypoints. In that paper a dense depth map is computed from a small set of images, then a 2D scan is rendered and a suitable waypoint is obtained, but extensive extra computations are required. They also employed the navigation algorithms proposed in [28], [29].

In our work, we use the sparse depth map from the special features on the image, provided by the PTAM algorithm, to extend its use to detect obstacles. For a wide range of applications, we consider that no extra computation is needed for an extra dense map generation for collision avoidance, despite the fact that the sparse map is noisy and weak for low-texture regions. Consequently, the proposed technique is based on the monocular pose estimation proposed by [28], [29] to locate the vehicle, and extended to detect possible collisions. The prototype is a commercial quadcopter Parrot *AR.Drone*. The frontal collision avoidance is validated in two scenarios; collision-free autonomous navigation for trajectory tracking and haptic teleoperation in semi-autonomous mode (only the orientation is in autonomous mode).

High order sliding mode techniques are well-known for their inherent robustness properties and can be used to attenuate the undesired chattering effect typical in sliding mode strategies. Some applications of the sliding mode theory to UAVs can be found in the literature, for example, in [24] a 2-SM is used to achieve attitude control, while in [80] a sliding mode is used for position control. For autonomous navigation, a second order sliding mode controller was designed in this work, to track desired trajectories in a robust way with respect to uncertainties and perturbations, making the closed-loop system suitable for outdoors applications.

The outline of this chapter is the following: a brief description of the monocular vision algorithms to pose estimation and the experimental platform are presented in section 4.2,

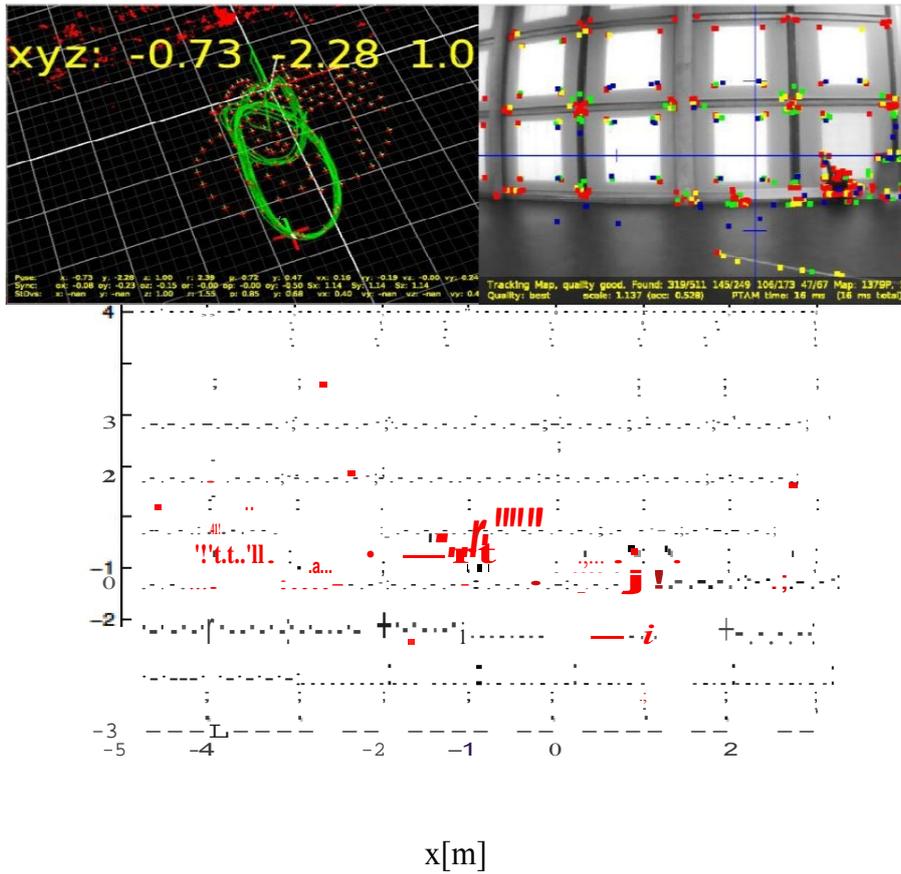


Figure 4-2: UAV localization w.r.t. the sparse depth map (top left). Characteristic features on the image (top right). Horizontal projection of the sparse depth map obtained by PTAM (bottom).

the proposed 2-SM strategy for robust trajectory tracking with frontal collision avoidance is described in section 4.3. Teleoperation with reactive collision avoidance is addressed in section 4.4. Main graphs from experimental results are introduced in section 4.5.

## 4.2 Monocular vision localization

The vision-based localization algorithm consists of a monocular SLAM and an EKF algorithms. The solution for monocular SLAM is based on Parallel Tracking and Mapping (PTAM) [46], which is a useful method for estimating camera pose in an unknown scene, originally conceived for AR applications and then successfully extended for MAVs localization [78], [28]. The tracking and mapping are split into two different tasks and are processed in parallel. It also constructs a sparse depth map (see Figure 4-2) which is used in this work to estimate distance to frontal objects.

The drawback of the PTAM algorithm is that it is constrained to mostly static, small

scenes, and it does not provide the absolute scale for the generated solution map, which is not suitable for MAV's applications. However, this has been overcome in [28], [29] by fusing the visual information with inertial and altitude measurements, by means of an EKF and a scale estimator. This vision algorithm is available online as open-source software in ROS, and can be directly applicable to a low-cost commercial quadrotor, the Parrot *AR.Drone*. The piece of code provided by [28] for the control algorithm was replaced for a new one in order to easily implement and validate different control strategies and help to tune the required gains. Furthermore a trajectory generator was also included for tracking autonomously different kinds of time-varying trajectories, rather than just way-points. Finally, the localization algorithm was modified to recover the pointcloud of the depth map generated by the PTAM algorithm, and send it to another node to estimate the distance to potential collisions, as explained in subsection 4.3.2. In this manner, the operator can select online between the different programmed trajectories, control laws and operation modes, as well as modify in real time any parameter for tuning.

### 4.3 Collision-free autonomous tracking

Let us consider a simplified version of the well-known dynamic model of a quadrotor [20]:

$$m\ddot{\xi} = TRe_3 - mge_3 \quad (4.1)$$

$$\ddot{\phi} \approx \Gamma \quad (4.2)$$

with mass  $m$  and the gravity constant  $g$ . Also,  $\xi = [x \ y \ z]^T$  is the position of the quadrotor with respect to an inertial frame,  $\Phi = [\varphi \ \theta \ \psi]^T$  stands for the Euler angles roll, pitch and yaw.  $T \in \mathbb{R}^+$  defines the total thrust produced by the motors and  $\Gamma \in \mathbb{R}^3$  is the control torque produced by the differential of velocities for the rotors. Finally,  $R \in SO(3)$  is the rotation matrix from the body fixed frame to the inertial one, and  $e_3 = [0 \ 0 \ 1]^T$ .

### 4.3.1 2-Sliding Mode Trajectory Tracking Control

Assuming that the closed loop dynamics of rotation are faster than the translational ones, it is possible to separate the model in two independent subsystems [14]. The strategy consists on designing a controller for the translational dynamics such that it guarantees the trajectory tracking by using the desired orientation as virtual inputs. Note that this is suitable for most commercial available UAVs with internal autopilot.

Due to its robustness property, a second order sliding mode control have been chosen to deal with uncertainties and perturbations in the translational dynamics, while tracking a desired position [69].

Define the desired position  $\xi_d$  and the position error as  $\bar{\xi} = \xi - \xi_d$ , then substituting into (4.1) leads to

$$m\ddot{\bar{\xi}} = (TRe_3)_d - mge_3 - m\ddot{\xi}_d + w \quad (4.3)$$

where  $w \in \mathcal{R}^3$  is an external and unknown disturbance vector.

Let us consider the so called switching function with relative degree 2

$$\sigma = \bar{\xi} + k_1 \dot{\bar{\xi}} \quad (4.4)$$

where  $k_1 \in \mathcal{R}^{3 \times 3}$  is a diagonal positive definite gain matrix. Then calculating the second time derivative

$$\ddot{\sigma} = \ddot{\bar{\xi}} + k_1 \dot{\bar{\xi}} = \frac{1}{m}(TRe_3)_d - ge_3 - \ddot{\xi}_d + k_1 \dot{\bar{\xi}} + \frac{w}{m} \quad (4.5)$$

Consider now  $u = (TRe_3)_d$  to be the control input. Then using the Twisting Algorithm [69], the following discontinuous controller is proposed

$$u = m(ge_3 + \ddot{\xi}_d - k_1 \dot{\bar{\xi}} - r_1 \text{Sgn}(\sigma) - r_2 \text{Sgn}(\dot{\sigma}) - k_2 \dot{\sigma}) \quad (4.6)$$

where the first three terms will annihilate the continuous dynamics of the closed loop system, while the next two expressions introduce the discontinuous control. Note the inclusion of the final term  $(-k_2 \dot{\sigma})$ , which induces a proportional feedback of the position error, it is aimed to improve the controller behavior and facilitate the parameters tuning.

$r_1, r_2, k_2 \in \mathbb{R}^+$  are constant control parameters.

Here, a vectorial sign function is employed, then for a vector  $Y = [u_1 \ u_2 \ u_3]^T$  we will have

$$\text{Sgn}(Y) = \begin{bmatrix} \text{sgn}(u_1) \\ \text{sgn}(u_2) \\ \text{sgn}(u_3) \end{bmatrix}; \quad \text{sgn}(u_i) = \begin{cases} 1 & u_i > 0 \\ -1 & u_i < 0 \end{cases} \quad (4.7)$$

The closed-loop dynamic is obtained introducing (4.6) into (4.5), it is

$$\ddot{\sigma} = -r_1 \text{Sgn}(\sigma) - r_2 \text{Sgn}(\dot{\sigma}) - k_2 \dot{\sigma} + \frac{1}{m} w \quad (4.8)$$

In order to analyze the stability at the discontinuous point  $(\dot{\sigma}, \sigma)$ , let us consider the Lyapunov candidate function

$$V = r_1 \|\sigma\|_1 + \frac{1}{2} \dot{\sigma}^T \dot{\sigma} \quad (4.9)$$

which is Lipschitz continuous.

Differentiating (4.9) with respect to time leads to

$$\dot{V} = r_1 \text{Sgn}(\sigma)^T \dot{\sigma} + \dot{\sigma}^T \dot{\sigma} \quad (4.10)$$

everywhere but on  $\sigma = 0$  where  $V$  is not differentiable.

Introducing (4.8) into the above equation

$$\dot{V} = r_1 \text{Sgn}(\sigma)^T \dot{\sigma} + (-r_1 \text{Sgn}(\sigma) - r_2 \text{Sgn}(\dot{\sigma}) - k_2 \dot{\sigma} + \frac{1}{m} w)^T \dot{\sigma} \quad (4.11)$$

or, equivalently,

$$\dot{V} = -r_2 \|\dot{\sigma}\|_1 - k_2 \|\dot{\sigma}\|^2 + \frac{1}{m} w^T \dot{\sigma} \quad (4.12)$$

and thus

$$\dot{V} \leq \|\dot{\sigma}\|_1 (-r_2 + \frac{1}{m} \|w\|) - k_2 \|\dot{\sigma}\|^2 \quad (4.13)$$

if the external disturbance is bounded  $\|w\| \leq ma$ , for some  $a > 0$ , and choosing  $r_2 > a$

leads to

$$\dot{V} \leq 0 \quad (4.14)$$

The set  $S$  where  $\dot{V} = 0$  is given by

$$S = \{(\sigma, \dot{\sigma}) | \dot{\sigma} = 0\} \quad (4.15)$$

on this set, the trajectories of the system take the form

$$\ddot{\sigma} = -r_1 \text{Sgn}(\sigma) - r_2 \text{Sgn}(\dot{\sigma}) + \frac{1}{m} w \quad (4.16)$$

where each element of  $\text{Sgn}(\dot{\sigma}) \in \mathbb{R}^3$  are of the form  $\text{sgn}(\dot{\sigma}) \in [-1, 1]$ . An invariant set on  $S$  implies  $\ddot{\sigma} = 0$ , hence

$$0 = -r_1 \text{Sgn}(\sigma) - r_2 \text{Sgn}(\dot{\sigma}) + \frac{1}{m} w \quad (4.17)$$

selecting  $r_1 > r_2 + a$  is enough to guarantee that the largest invariant set on  $S$  contains only the origin  $\sigma = \dot{\sigma} = 0$ . Using the extended version of the LaSalle invariance principle, all the trajectories of the system converge to the origin, and the system is stable [68]. A formal proof for globally equi-uniformly finite time stability for this kind of systems with the proposed gains conditions can be found in [59].

Therefore the sliding mode control law (4.6) solves the trajectory tracking problem despite uncertainties and perturbations. It is important to notice that

$$R_d e_3 = \begin{bmatrix} R_{dx} \\ R_{dy} \\ R_{dz} \end{bmatrix} = \frac{(TRe_3)_d}{T_a} \quad (4.18)$$

with  $T_d = \|(TRe_3)_d\|$ , and using the short notation  $s\alpha = \sin(\alpha)$ ,  $c\alpha = \cos(\alpha)$ , we have

$$\begin{bmatrix} R_{dx} \\ R_{dy} \\ R_{dz} \end{bmatrix} = \begin{bmatrix} s\psi_d s\varphi_d + c\psi_d s\theta_d c\varphi_d \\ -c\psi_d s\varphi_d + s\psi_d s\theta_d c\varphi_d \\ c\theta_d c\varphi_d \end{bmatrix} \quad (4.19)$$

then, proposing a constant  $\psi_d$ , it is possible to find  $\varphi_d$  and  $\theta_d$  explicitly as

$$\varphi_d = \arcsin R_{dx}s\psi_d - R_{dy}c\psi_d \quad (4.20)$$

$$\theta_d = \arcsin \frac{R_{dx}c\psi_d + R_{dy}s\psi_d}{c\varphi_d} \quad (4.21)$$

where  $\varphi_d$  and  $\theta_d$  are the desired roll and pitch angles provided to the *AR.Drone*.

### 4.3.2 Collision Avoidance

One of the main challenges in autonomous navigation for mobile robots such as UAVs, is the perception of unknown environments, with the extra difficulty of the limited payload. In the present work, we are interested in detecting possible collisions using the already available information from the sensors of an inexpensive commercial quadrotor. Particularly we are interested in using monocular vision with the frontal camera to estimate the distance to obstacles by taking advantage of the sparse depth map generated by the localization algorithm.

Then, distance to frontal collision is estimated from the horizontal projection of the sparse depth map computed by the PTAM algorithm as the one showed in Figure 4-2. It consists of a set  $P$  of  $n$  points  $p_i(x_i, y_i, z_i)$ ,  $i = 1, \dots, n$  obtained from the characteristic features of the image stream provided by the frontal camera of the quadrotor. PTAM uses these points as a map to estimate the pose of the UAV. An horizontal projection of these points cloud as depth map is used in this work, i.e., all the obstacles are considered having the same height (this stands for walls, columns, etc.). However, it results in a very noisy depth map and should be given special care for obstacles presenting low-texture surfaces for the vision algorithm.

Therefore, we define the estimated distance to frontal obstacles  $d$  as

$$d = y - \frac{1}{n} \sum_{i \in \Omega} y_i \quad (4.22)$$

$$\Omega = \{p_i(x_i, y_i, z_i) \in P \mid x_i \in [x - \varepsilon, x + \varepsilon]\}$$

i. e., the average depth, w.r.t. the position of the quadrotor along  $y$ , of the  $\eta$  points inside certain lateral range  $\varepsilon$  from the lateral position of the quadrotor  $x$ .

In order to avoid collision, we apply a potential field, such that if distance  $d$  falls below certain safe distance  $d_s$ , then, a repulsive force  $F_{rep}$  will be exerted, given by [70]

$$F_{rep} = \begin{cases} 0 & d > d_s \\ -k_{rep} \left( \frac{1}{d} - \frac{1}{d_s} \right) \left( \frac{1}{d^2} \right) & d \leq d_s \end{cases} \quad (4.23)$$

note that  $F_{rep} \rightarrow \infty$  as  $d \rightarrow 0$ , hence for the practical implementation  $F_{rep}$  is saturated at a maximum value.

Finally, retaking the 2-SM trajectory tracking control (4.20), (4.21), the full collision-free trajectory tracking strategy becomes

$$\varphi_d = \arcsin R_{dx}s\psi_d - R_{dy}c\psi_d + F_{rep} \quad (4.24)$$

$$\theta_d = \arcsin \frac{R_{dx}c\psi_d + R_{dy}s\psi_d}{c\varphi_d} \quad (4.25)$$

The same can be applied for the lateral motions, however, obstacle detection in this directions becomes much more challenging and requires more sophisticated techniques.

## 4.4 Haptic teleoperation

Although huge advances have been accomplished in the research for fully autonomous UAVs, human tele-operated UAVs remain as a good alternative due to the still unsolved issues and security constrains related to the fully autonomous ones. Still, it is important to aid the operator with his tasks as much as possible, by providing him useful information and making the operation more intuitive. Even more, it is necessary to minimize the risk of human mistakes.

Haptic devices appear as a good option to assist humans in the teleoperation of systems such as UAVs. Haptic is tactile feedback technology which recreates the sense of touch by applying forces, vibrations, or motions to the user. In this sense, haptic technologies can be applied to UAVs teleoperation in order to improve the piloting experience by providing

him feedback of forces. For example, he would be able to feel a gust of wind or any other external perturbation, affecting the vehicle. It can also be used to prevent the user from executing forbidden or risky actions such as driving the UAV to unsafe areas or out of the range of communication.

In this work, we are interested in investigating the use of haptic devices for safe UAV teleoperation by preventing the user to frontal collide the quadrotor. The most common techniques for haptic feedback to avoid collisions in UAVs are force feedback. For example artificial force field, and stiffness feedback using a virtual spring, a comparative analysis of these techniques can be found in [48] and [18].

In this work we successfully applied the Novint Falcon haptic device to assist in teleoperation and prevent the user from crashing the quadrotor against frontal obstacles. To do so, the position of the final effector of the haptic device  $(x_h, y_h, z_h)$  provides in a linear relation the desired roll and pitch angles  $(\varphi_d, \theta_d)$  and the desired altitude velocity  $(\dot{z}_d)$ , respectively. It is

$$\begin{aligned}\varphi_d &= k_{hy}(y_h - off_y) \\ \theta_d &= k_{hx}(x_h - off_x) \\ \dot{z}_d &= k_{hz}(z_h - off_z)\end{aligned}\tag{4.26}$$

where  $k_{hx}, k_{hy}, k_{hz}, off_x, off_y, off_z$  are suitable gains and offsets. This strategy is useful for quadrotors with an internal orientation and altitude controllers. Observe from (4.26) that to keep the quadrotor hovering in a desired position, the user should keep the haptic device at the center of its workspace. In order to assist the user in this task, and simplify the manual control of the UAV, a proportional derivative controller is applied to regulate the haptic's final effector position at the origin if no force from the operator is exerted. Similarly, a repulsive force is applied to the haptic device once the quadrotor approaches a frontal obstacle. Then, the feedback forces applied to the haptic device are defined as

$$\begin{aligned}F_x &= -k_{px}x - k_{dx}\dot{x} \\ F_y &= \begin{cases} -k_{py}y - k_{dy}\dot{y} & d > d_s \\ -h_{rep}(\frac{1}{d} - \frac{1}{d_s})(\frac{1}{d^2}) & d \leq d_s \end{cases} \\ F_z &= -k_{pz}z - k_{dz}\dot{z}\end{aligned}\tag{4.27}$$

with the control gains  $k_{px}, k_{py}, k_{pz}, k_{dx}, k_{dy}, k_{dz}, h_{rep} \in \mathbb{R}^+$ .

Once again, this can be extended for the lateral motion if a better estimation of the obstacles is available.

## 4.5 Experimental results

Several flight tests were carried out to validate the experimental setup and the proposed strategies. The practical validations were developed in two ways; autonomous trajectory tracking and haptic teleoperation. The parameters used for both cases were tuned by trial and error and are presented in Table 4.1.

### 4.5.1 Trajectory tracking

In this test, the mission was to follow autonomously a lemniscate trajectory with a length of 3m. The challenge in this mission is to introduce a wall close to the trajectory, such that the vehicle, using the monocular camera detects the wall and activates the repulsive algorithm for collision avoidance. It is important to notice that, for the practical implementation, the sign function 4.7 is approximated by a high gain saturation such that

$$\text{sgn}(u) \approx \frac{\text{sat}(u)}{\varepsilon} \quad (4.28)$$

with  $\varepsilon \approx 0$  as a small constant.

Figure 4-3 illustrates the experiment in a 3D view, where we can appreciate how the vehicle modifies its trajectory when it is close to the wall, see also Figure 4-4. Notice here that the repulsive force  $F_{rep}$  becomes bigger than the force generated by the tracking control

Table 4.1: Parameters

$k_{1x}, k_{1y}$	$k_{2x}, k_{2y}$	$\varepsilon[m]$	$k_{rep}$	$d_s[m]$	
2	25	0.5	4	5	
$k_{hx}$	$k_{hy}$	$k_{hz}$	$k_{px, py, pz}$	$k_{dx, dy, dz}$	$h_{rep}$
12.7	14.3	18.2	100	500	2

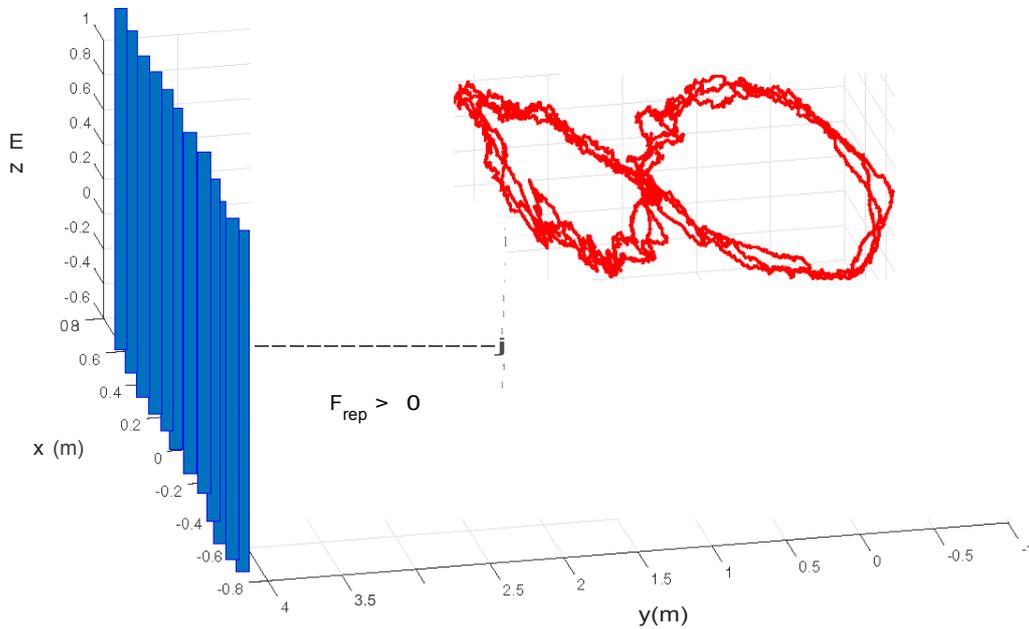


Figure 4-3: 3D Collision-free lemniscate trajectory tracking close to a wall.

*Fy*. The aerial view of this results are illustrated in Figure 4-4.

Figures 4-5 and 4-6 show the  $x$  and  $y$  states of the quadrotor when following the trajectory close to a wall. Observe in these figures that the quadrotor carries out very well the path tracking in the  $y$  coordinate, except when the quadrotor approaches the wall and the repulsive force acts on the control scheme, see Fig. 4-6. Meanwhile in the  $x$  position, see Fig. 4-5, some minor error can be observed due to the more aggressive nature of the lemniscate trajectory on this axis.

In Figure 4-7 the tracking error is displayed. These errors always remain bounded in small values except for the time when the UAV get close to the wall. The control inputs for the internal autopilot are depicted in Figure 4-8. It is important to point out that small gains have been chosen in the discontinuous terms, for the real time implementation to avoid aggressive commands. A good alternative would be to use the Super Twisting Algorithm [69] in order to avoid the chattering phenomena.

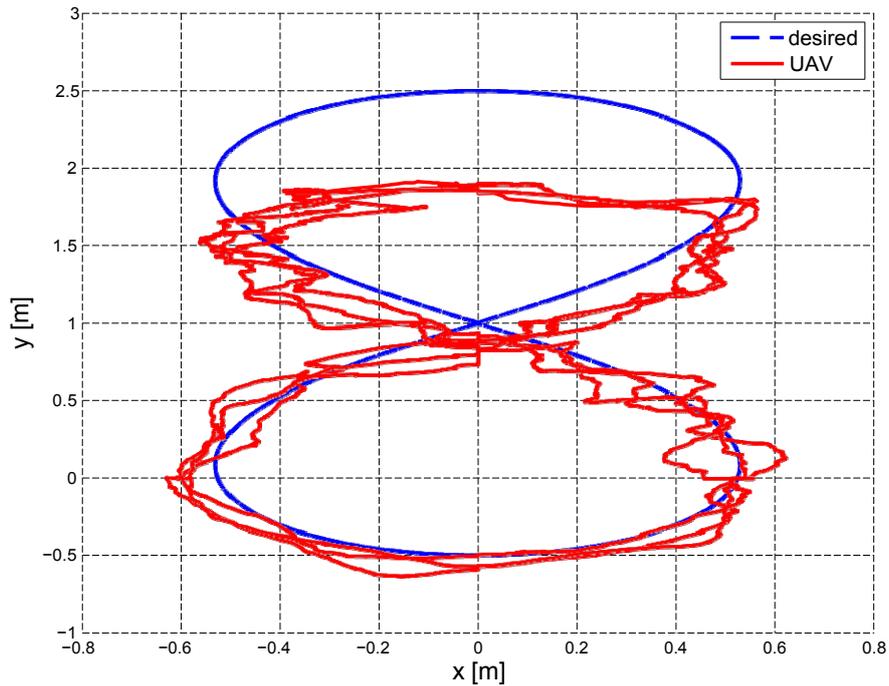


Figure 4-4: Collision-free trajectory tracking.

#### 4.5.2 Collision-free teleoperation

For the teleoperation scenario, a human pilot controls the quadrotor through a haptic device. The practical goal here is to use the haptic device to feedback information from the vehicle to the pilot and prevent him from colliding, via opposite forces, when the quadrotor goes out of a safety zone.

Collision-free haptic teleoperation is studied through Figures 4-9 to 4-11. In these flight tests, the user flew the UAV in semi-autonomous mode using a haptic device. Here, only the orientation is in autonomous mode. The goal was to crash deliberately the vehicle against a frontal wall. Several attempts were made to do so, the first ones were realized slowly, but the last was done at high speed.

The position response in the  $y$  state of the UAV through the experiments is depicted in Figure 4-9, observe in this figure at times 6s, 9s and 13.5s, how the reactive collision avoidance algorithm prevented the user from driving the quadrotor to a dangerous area too close to the wall. Furthermore, at time 16s the pilot deliberately tried to crash the UAV

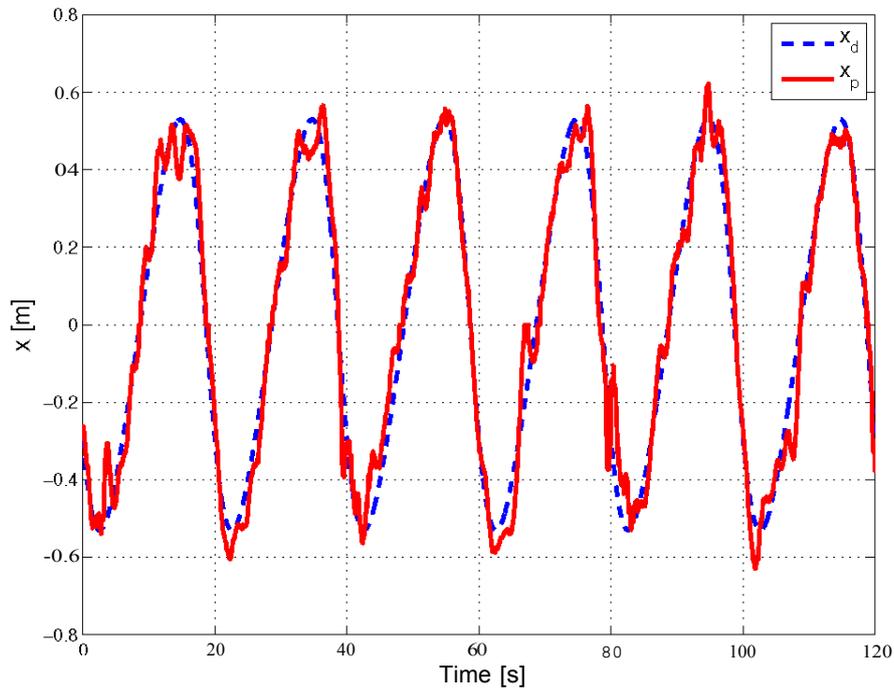


Figure 4-5: x position response for a lemniscate trajectory.

against the wall by quickly moving towards it, even though it gets to touch the wall, it never crashes thanks to the good performance of the proposed algorithm even in this extreme case. Finally, observe in Figures 4-10 and 4-11 the feedback force applied to the haptic device and the control inputs sent to the UAV, respectively. When approaching the wall, the haptic device exerts a repulsive force alerting the human operator of the danger and even forbidding him from crashing the quadrotor, see second 16 on Figures 4-10 and 4-11.

Some of the tests can be found on video at <https://www.youtube.com/watch?v=PuT3-NPj698>.

It is important to point out that the obtained results are quite satisfactory, taking into account that only a monocular camera is used to locate the quadrotor and to detect collisions, instead of using an expensive motion capture system and/or extra range sensors as other teams. In addition, these results can be easily reproduced or extended to outdoor flight tests.

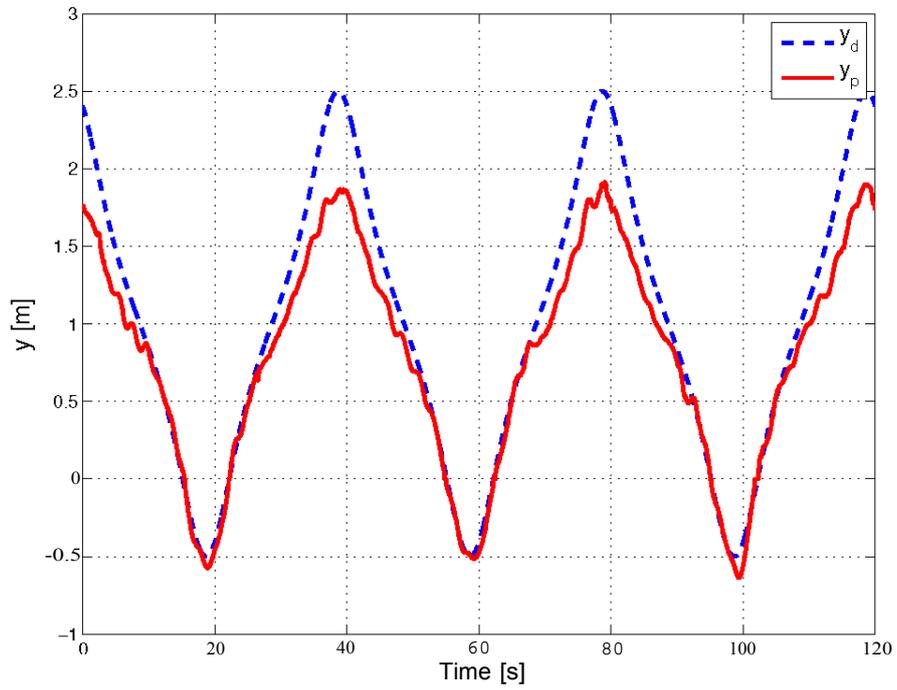


Figure 4-6:  $y$  position response for a lemniscate trajectory.

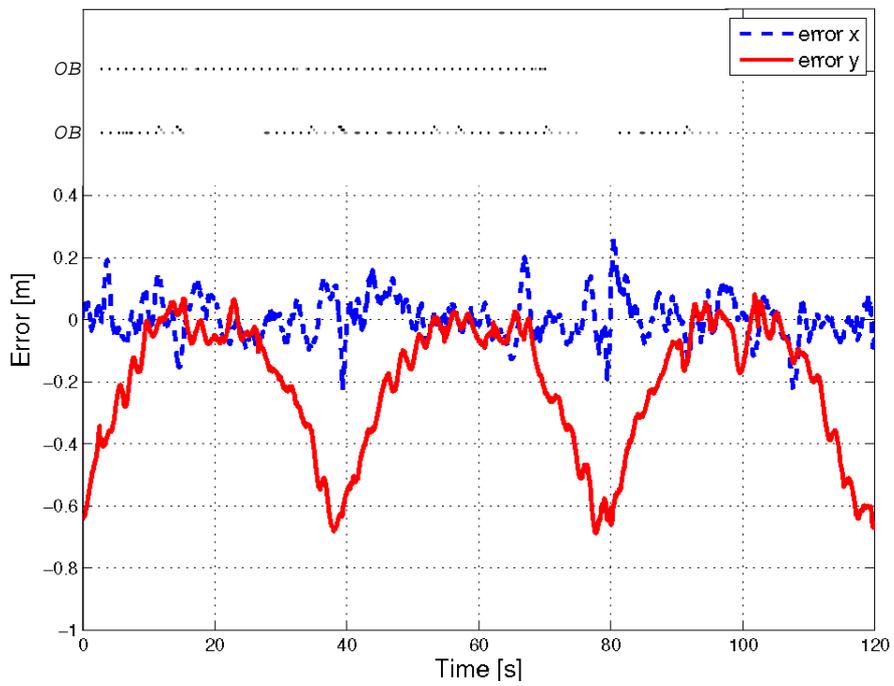


Figure 4-7: Tracking error.

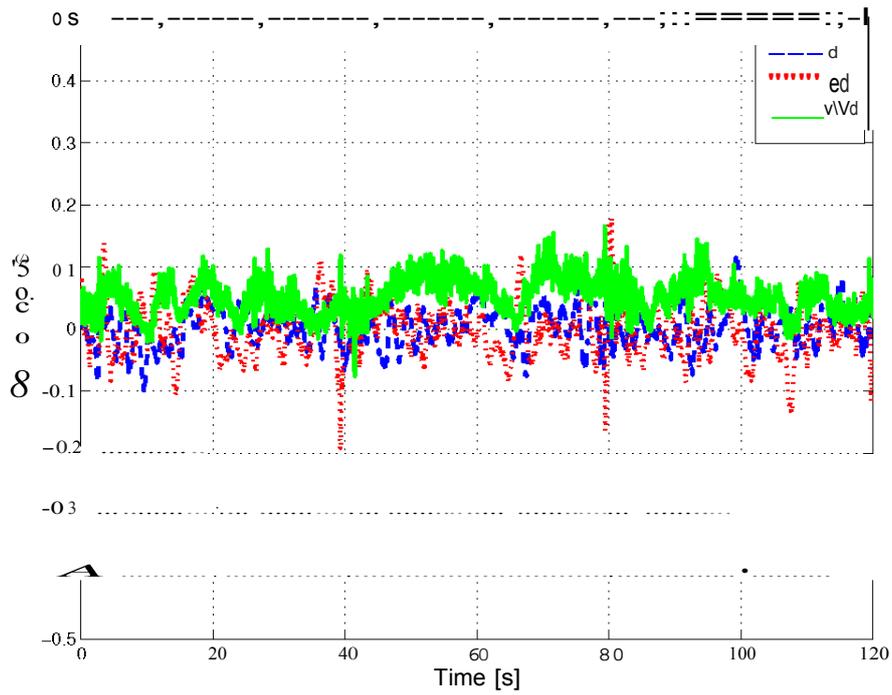


Figure 4-8: Trajectory tracking control inputs.

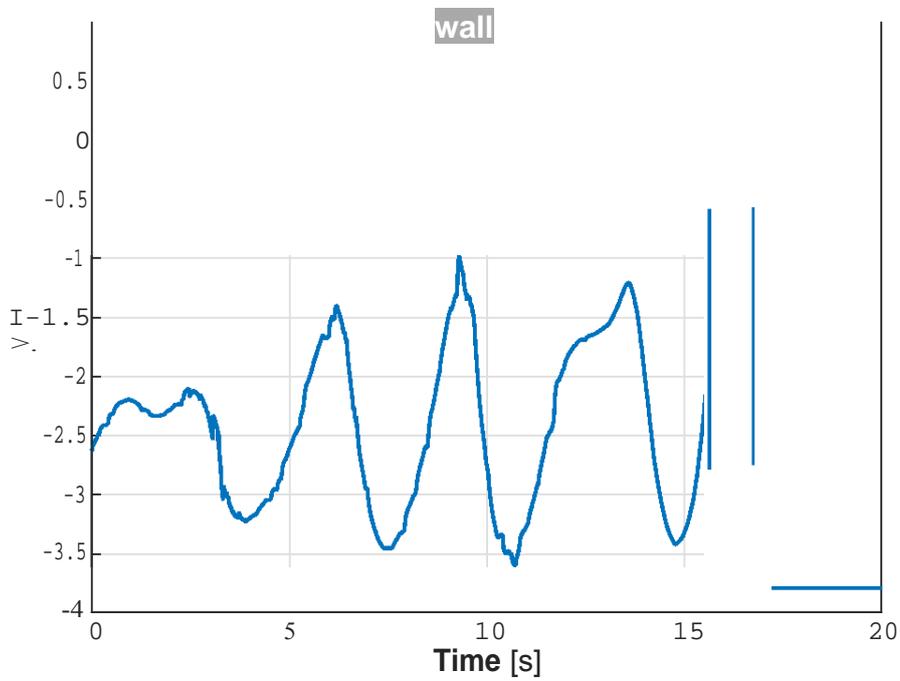


Figure 4-9: y coordinate.

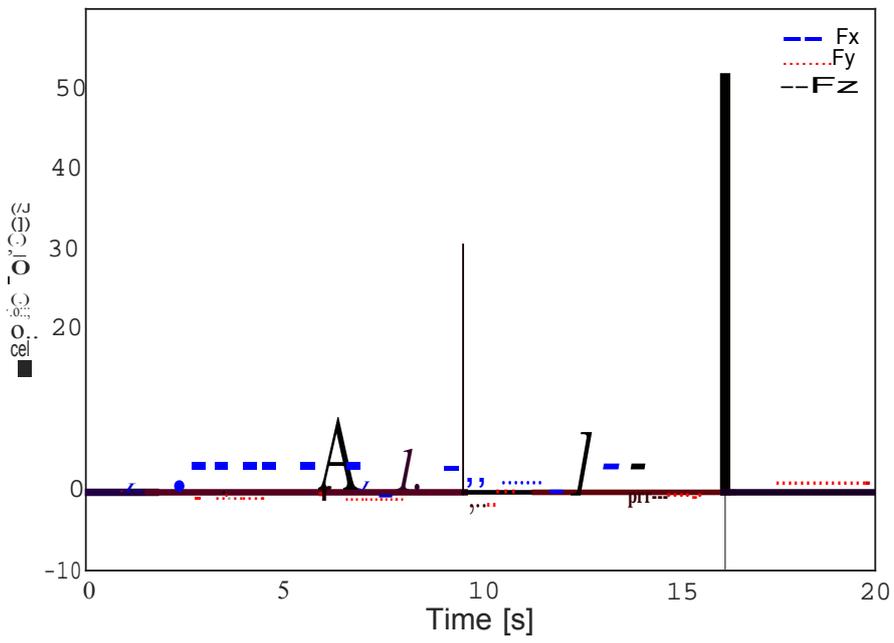


Figure 4-10: Haptic feedback forces.

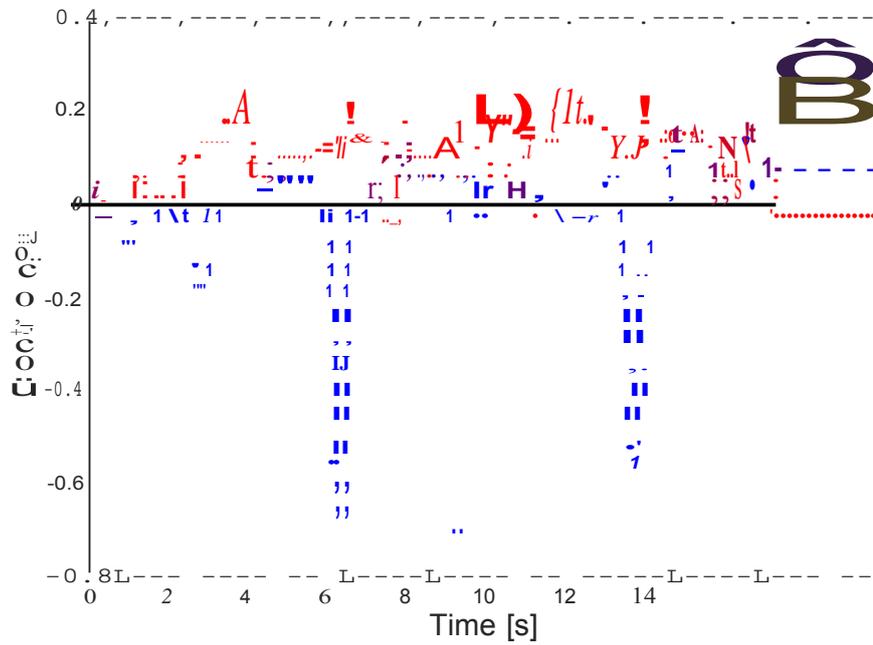


Figure 4-11: UAV control inputs.



## **Chapter 5**

### **Human-Robot Interaction: the Face**

#### **Follower UAV**

Conception and development of an Unmanned Aerial Vehicle able to safely interact with human users by following them autonomously, is presented in this work. Face detection is accomplished by a Haar cascade classifier. Once a face is detected, it is tracked with the help of a KF, and an estimation of the relative position with respect to the face is obtained at a high rate. A linear PD controller regulates the UAV's position in order to keep a constant distance to the face, employing as well the extra available information from the embedded UAV's sensors. The proposed system was extensively tested, using a commercial inexpensive quadrotor connected via wireless communication to a ground station computer with the help of ROS, under different conditions, showing good performance even under disadvantageous conditions as outdoor flight, being robust against illumination changes, image noise and the presence of several faces on the same image.

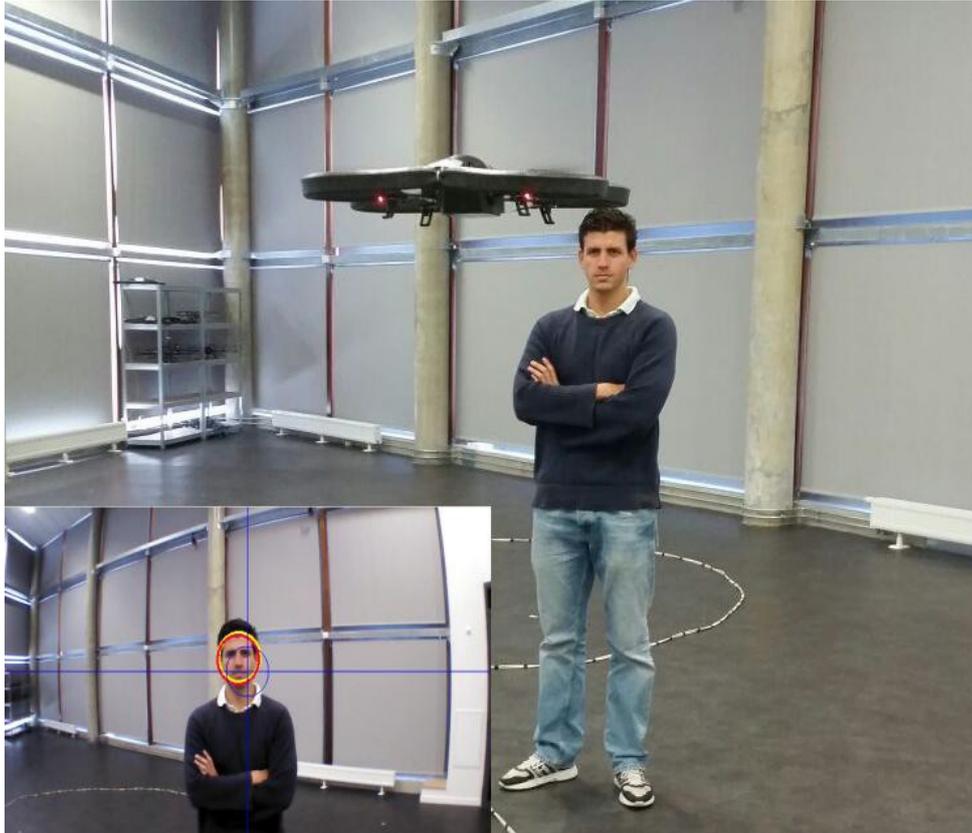


Figure 5-1: Quadrotor following a face.

## 5.1 Introduction

The astonishing fast developments on the fields of computer science, electronics and mechanics, and more particularly on robotics, over the last recent years, allows us to believe that the science fiction vision of a world surrounded by all kind of robots interacting with humans in all sort of tasks, is no longer a fairy dream, but a real possibility in the near future.

Nowadays, the use of *smart* devices in everyday life is getting more and more common. From smart phones, to watches and televisions, a real software revolution is permitting that even small and trivial objects like keys are getting smarter these days. In contrast, things have been moving slower for robotics due to the great danger that some robots could represent while working close to humans or interacting with them. This is a delicate topic and must be taken carefully, hence, before continue speculating about the future, let us

not to forget the three laws of robotics envisioned at [9], which perfectly summarize such security concerns:

1. A robot may not injure a human being or, through inaction, allow a human being to come to harm.
2. A robot must obey the orders given it by human beings except where such orders would conflict with the First Law.
3. A robot must protect its own existence as long as such protection does not conflict with the First or Second Laws.

Until now, this high risk of injure humans have prevented the use of robots for several applications, constraining them to industrial tasks where the environment is controlled and security can be extremed, or in dangerous scenarios were humans can hardly operate. Otherwise, small size robots that are harmless for humans have to be employed, mainly for entertainment and research, to cite some famous examples we have the vacuum cleaning robot *Roomba* from *iRobot*, or the humanoids *Nao* from *Aldebaran Robotics* and *ASIMO* produced by *Honda*.

With all this in mind, let us focus on the kind of robots that are of interest for the present work, the aerial robots, better known as Unmanned Aerial Vehicles. Some interesting applications for UAVs have popped out from new emerged technologies, for example, by using the *Myo* bracelet produced by *Thalmic Labs* we are able to control UAVs just with the movements of the arm [3]. This is a devise equipped with Electromyography (EMG) sensors along with a 9-axis IMU to observe activity from your muscles to detect hand gestures and orientation. Another nice technology implemented to drones for human interaction is the one presented in [32], where the control of a quadrotor is achieved by a computer vision algorithm and a 3D camera, able to detect hand gestures and interpret them as commands. A not so new but still powerful tool is the *Microsoft Kinect* sensor. It has been conceived to recognize gestures and body postures to friendly control the navigation of UAVs through Natural User Interface (NUI) [6], [67]. Additionally, haptic devices offer a different way of interaction with human operators by providing force feedback, this

allows to improve the piloting experience and assist the user in complicated tasks such as simultaneously controlling multiple UAVs [49].

As for the present work, it is intended to offer a practical solution to the problem of an UAV capable of following a human face. As a first approximation to the problem, things are kept simple whenever possible, and the simplest solution was preferred over the sophisticated one, as long as it was effective in accomplishing its objective and always keeping in mind the users security as the design priority. To do so, a wide variety of problems, all of great interest in the robotics field, were successfully solved and the resulting system can be safely used. They go from perception using monocular computer vision, to automatic control and state estimation, passing through signal filtering.

Such a face follower drone is envisioned for several interesting applications, such as to follow a suspect or a fugitive in a persecution. Other possible use is to video record someone in hazardous situations, like while practicing extreme sports. This idea was already conceived by the commercial quadrotor *LILY* [2], but in that case the system depends on an external tracking device to localize the human user, while our proposed algorithm depends only on an already integrated frontal camera. Also, the face-follower can be employed for entertainment, for example, trying to hide and run away from the drone, or even to play with it some popular games like *tag*, or the *hot-potato*. Furthermore, the classifier can be trained to detect any other mostly rigid object to be followed for the drone.

The contribution of the present work can be summarized in the following points:

- A simple but effective technique to estimate the distance between a human face and the image plane is developed from the size of the detected face on the image.
- A complete computer vision algorithm is developed to detect and track a face despite the presence of false positives or other faces on the image.
- Several powerful available tools and techniques are merged together to offer a complete working solution for a face follower drone.
- A real time relative position control algorithm is proposed for an UAV following a moving object with unknown dynamics, including the observation of the missing state.

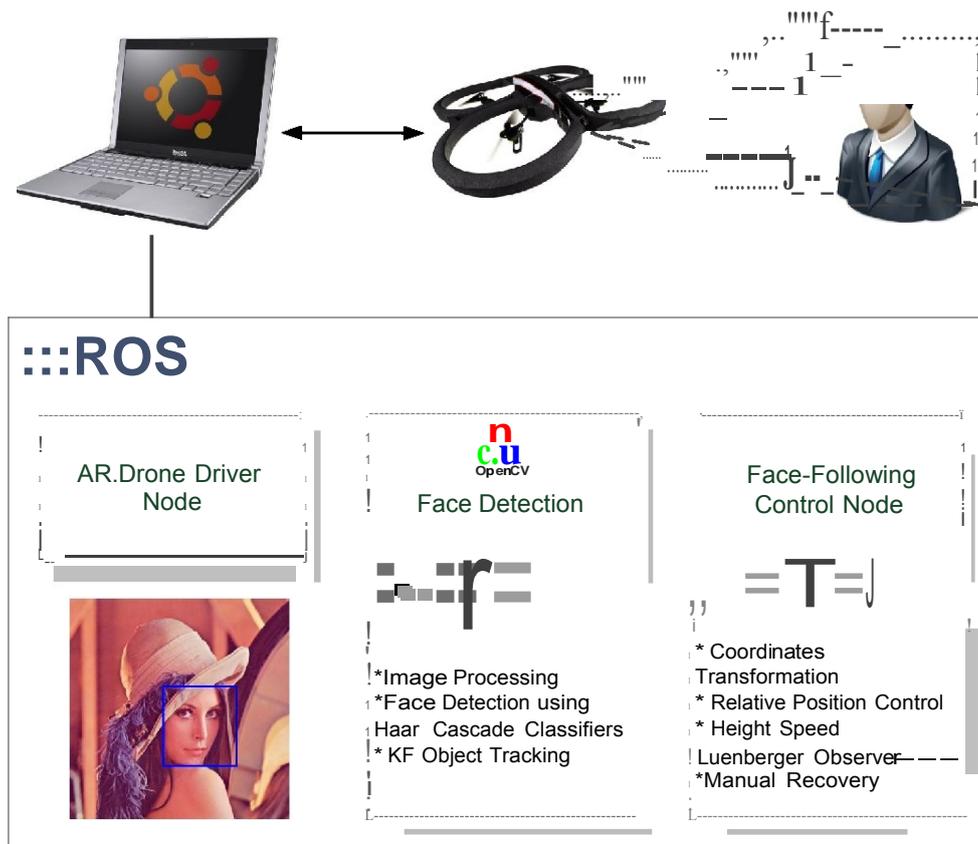


Figure 5-2: Overall system description.

- To the best of our knowledge, this is an innovative application for UAVs that can be used as a base for other future interesting applications.

The outline of this chapter is the following: the general operation of the overall system is described in section 5.2. In section 5.3 the computer vision algorithm for face detection and tracking, as well as the necessary transformation from the image to the real world are presented. The relative position control by means of a linear PD is introduced in section 5.4, along with some techniques for estimate the altitude speed, needed for the state feedback. Last but not least, section 5.5 contains the experimental results of the proposed algorithms.

## 5.2 Overall System Description

We consider the use of a commercial inexpensive quadrotor *AR.Drone*, as described in section 2.2.1. The frontal embedded camera is used by the vision algorithm to detect faces.

At the ground station, the algorithm is decomposed in three main ROS nodes: the *AR.Drone* driver node, already available, an image processing and face detection node and the control node, both developed for this work. An extra node is employed for manual recovery in case of any problem, using a *sixaxis* wireless controller. Furthermore, a graphical users interface allows to select between diverse operation modes as well as to tune in real time the parameters of interest. The overall system architecture can be observed in Fig. 5-2.

The *AR.Drone* driver node connects the UAV with ROS by a wireless communication, retrieving and publishing important information from the sensors, along with the videos from the cameras. It also sends the control inputs to the helicopter.

A second node receives the video stream from the frontal camera and, with the help of the OpenCV libraries, performs the image processing for filtering and face detection using a Haar feature-based cascade classifier. This node is also in charge of the face tracking by means of a Kalman Filter. The result is the tracked face's position and size on the image coordinates, which are computed and published at 6Hz.

Position control is carried out by a different node, using the estimated relative position with respect to the face obtained by the vision algorithm and key information from the embedded sensors, such as the quadrotor horizontal speeds. A suitable coordinates transformation is executed at this point, from the image space to the real world for the lateral and altitude relative position. Meanwhile, the distance to the tracked object is estimated from the face size. Also, a Luenberger observer deals with the estimation of the missing altitude speed.

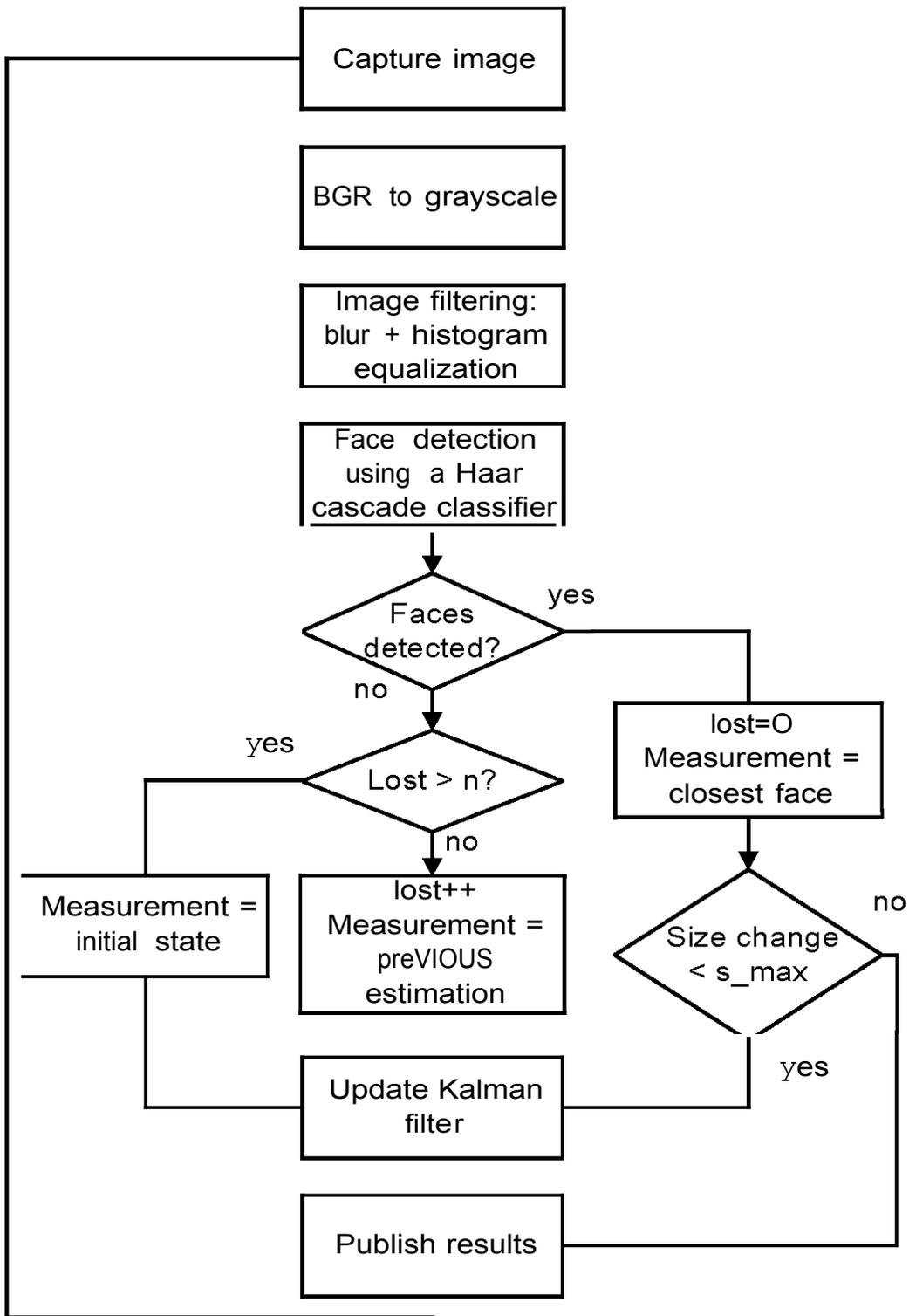


Figure 5-3: Flowchart of the vision algorithm.

## 5.3 Face Detection and Tracking

### 5.3.1 Computer Vision Algorithm for Face Detection

In order to perform the image acquisition and processing for the computer vision algorithm, the OpenCV libraries are used on ROS using the *cv\_bridge* node. Figure 5-3 shows the flowchart describing the developed computer vision algorithm for face detection and tracking. Firstly, images from the quadrotor's frontal camera are received at a rate of 30Hz, in BGR format and transformed to gray scale. Then, a *blur* operation is effectuated to smooth the image and filter the white noise, this is useful to decrease the number of false positive detection due to high frequency image noise. Before the face detection function is applied, a histogram equalization is carried out to expand the intensities range and improve the contrast on the image.

Face detection is accomplished through the Haar classifier on OpenCV [17]. It consists in a Machine Learning technique first developed in [75] which uses Haar-like features in cascade through different levels of the image to determine whether or not a pre-specified rigid object, for which it was trained a priori, is present on the image. The Haar classifier is a supervised classifier that uses a form of AdaBoost organized as a rejection cascade and designed to have high detection rate at the cost of low rejection rate, producing many false positives. One of the main advantages of this method is the computational speed achieved in real time detection, once the classifier was trained off-line for the desired object, in this case a face. It is important to notice that this method can be trained for almost any mostly rigid object with distinguishing views.

Up to now, only frontal faces are detected using the pre-trained classifier already provided. In order to detect rotated faces it is preferable to train different classifiers for each specific pose and run the detectors sequentially.

### 5.3.2 Kalman Filter for Object Tracking

As stated before, one of the main drawbacks of the face detection algorithm is its low rejection rate, or in other words, its high number of false positive detections. To overcome

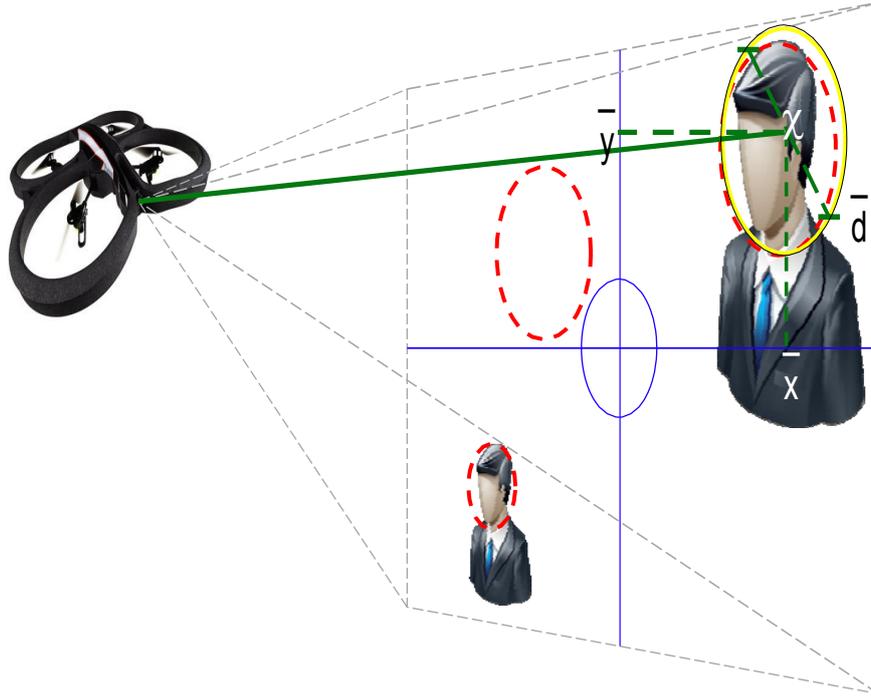


Figure 5-4: Face detection and tracking with a flying quadrotor.

this issue and in order to add robustness to the algorithm against other faces present on the image and missed detections, it is helpful to track along time the selected face. To do so, we consider the use of the Kalman filter, a powerful technique for optimal state estimation and filtering of linear dynamic systems perturbed with Gaussian noise [43].

In this case, the discrete-time version of the Kalman filter [35], [19] is applied to the kinematic model of the detected face (see Fig. 5-4), i. e.

$$\begin{bmatrix} \chi_k \\ V_k \end{bmatrix} = \begin{bmatrix} 1 & T_s \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \chi_{k-1} \\ V_{k-1} \end{bmatrix} + \begin{bmatrix} 0 \\ a_{k-1} \end{bmatrix} \quad (5.1)$$

where  $\chi = [\bar{x} \ \bar{y} \ \bar{d}]^T$  contains the center position of the circle enclosing the face of interest, directly on the image and its diameter  $d$ , while  $V = [\dot{\bar{x}} \ \dot{\bar{y}} \ \dot{\bar{d}}]^T$  has its time derivative.  $k$  is the discrete time index and  $T_s$  defines the sampling period. Finally, the process noise  $a \in \mathbb{R}^3$  is used as a tuning parameter, analogous to the acceleration, which determines how fast the variables can move.

As for the measurement update in the Kalman filter, a few different cases are consid-

ered, depending on the state of the vision algorithm, they are:

- One or more faces detected.
- No face detected.
  - Iterations since last detection  $\leq n$ .
  - Iterations since last detection  $> n$ .

for certain trigger  $n$ . For the first case, if more than one face is detected, either due to a false positive detection or to the presence of other faces on the image, it is chosen the closest to the previous estimation. If no face is detected for a few iterations, a missed detection is assumed and the measurement is updated with the last estimation, allowing to continue with the same motion for a while, and avoiding sharp displacements in the closed loop system. However, if the face is lost for several iterations, it is assumed that the face is not present on the image anymore, hence, a more conservative approach is applied and the measure vector is updated with its initialization value  $\chi_{init}$ , which implies no motion from the quadrotor. Then, the measurement vector  $\zeta$  update equation takes the form

$$\zeta_k = \begin{cases} \chi_{min_k} & N_{faces} > 0 \\ \chi_{k-1} & (N_{faces} = 0) \& (I_{lost} \leq n) \\ \chi_{init} & (N_{faces} = 0) \& (I_{lost} > n) \end{cases} \quad (5.2)$$

with

$$\chi_{min} = \min_{\chi_{\delta_i} \in \Omega} \sqrt{(\bar{x} - x_{\delta_i})^2 + (\bar{y} - y_{\delta_i})^2} \quad (5.3)$$

where  $N_{faces}$  is the number of faces detected at the present frame,  $I_{lost}$  defines the number of iterations since the last valid detection.  $\Omega$  represents the set of all the faces detected on the image, described by  $\chi_{\delta} = [x_{\delta} \ y_{\delta} \ d_{\delta}]^T$ .  $\chi_{min}$  denotes the closest detection to the previous estimation.

The previous is illustrated on Figure 5-4, where a flying UAV captures a scene with multiple faces. Positive detections obtained by the classifier are identified by the dotted red circles, including false positive detections where there are no faces. Then, the Kalman filter

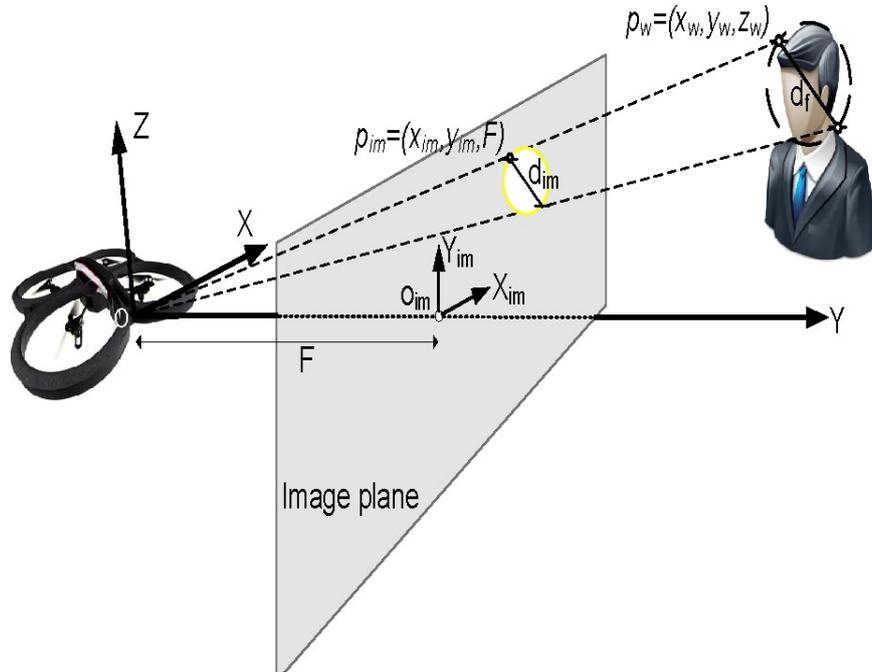


Figure 5-5: From real world to image coordinates transformation.

is employed to track the chosen face along time, and its result is displayed by a unique solid yellow circle with center at  $(x,y)$  and diameter  $d$ . Only the closest face is used to update the measurement vector of the Kalman filter, while the others are ignored.

### 5.3.3 Relative Position Estimation: From Image to Real World

For simplicity as a first approximation, let us consider the idealized pinhole camera model to describe the relationship between coordinates in the real world to their projection on the image. Then, any point in the real world  $P_w = (x_w, y_w, z_w)$  is projected to the image coordinates  $P_{im} = (x_{im}, y_{im})$ , see Figure 5-5, according to the following expressions

$$x_{im} = f_x \left[ \frac{x_w}{y_w} \right] + c_x; \quad y_{im} = f_y \left[ \frac{z_w}{y_w} \right] + c_y; \quad (5.4)$$

where the constant parameters for the focal lengths  $f_x = F s_x$ ,  $f_y = F s_y$  are actually the product of the physical focallength  $F$  and the number of pixels per meter  $s_x$ ,  $s_y$ , along each image axis. Those constants, together with the principal point position  $e_x$ ,  $c_y$  determine the

intrinsic parameters of the camera and are known from calibration. Note that, in contrast to the standard notation used in computer vision, here  $z$  stands for the altitude and  $y$  for the depth between the camera and the point, this is done just for consistency with the other used frames.

From the previous equation, it is straightforward to recover from the image projection, the real world position of a point in the  $x_w$  and  $z_w$  axis, normalized by the depth  $y_w$ . However, this depth  $y_w$  remains unknown and can not normally be obtained from the information from one single camera. The use of several cameras for stereo-vision is precluded due to the inflexibility of the selected hardware. Another option is to obtain extra information from the already available sensors, such as altitude and inertial measurements, to fusion it with the vision data to estimate this depth, similar to the work developed in [7]. Nevertheless, in our considered scenario, it can be estimated easily from the size of the detected face on the image. Even though everyone's face is different in size and shape, let us assume that we are dealing with an adult with an average face whose enclosing circle has a diameter of  $d_f$ . Also, consider the diameter  $d$  of the enclosing circle for the face projected in the image (see Fig. 5-4). Then, it is easy to show that

$$y_w \approx \frac{F d_f}{d} \quad (5.5)$$

Finally, substituting in (5.4) we get the position of the face in the physical world as

$$x_w = \frac{(x_{im} - c_x) s_x d_f}{d}; \quad z_w = \frac{(y_{im} - c_y) s_y d_f}{d}; \quad (5.6)$$

## 5.4 Relative Position Control

Once the vision algorithm detects a face and the Kalman filter tracks it through time, after a suitable transformations, we obtain an estimation of the relative position between the drone and the face of interest. The next step consists for the quadrotor to autonomously keep a constant predefined distance to the face. This is equivalent to a three dimensional position control problem where the reference continuously changes, with unknown dynamics.

To deal with this control problem, a linear PD controller is designed in cascade with the internal inner control loop from the attitude autopilot already included on the UAV under consideration.

Let us consider a simplified version of the well-known dynamic model of a quadrotor [20]:

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} \approx \frac{T}{m} \begin{bmatrix} s\psi s\varphi + c\psi s\theta c\varphi \\ -c\psi s\varphi + s\psi s\theta c\varphi \\ c\theta c\varphi \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \quad (5.7)$$

$$\begin{bmatrix} \ddot{\varphi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} \approx \begin{bmatrix} \tau_\varphi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} \quad (5.8)$$

where  $x$ ,  $y$ ,  $z$  are the position of the quadrotor with respect to an inertial frame,  $T \in \mathbb{R}^+$  defines the total thrust produced by the motors,  $m$  and  $g$  represent the mass and gravity constant, respectively.  $\varphi$ ,  $\theta$  and  $\psi$  stand for the Euler angles roll, pitch and yaw, and  $\tau_\varphi$ ,  $\tau_\theta$  and  $\tau_\psi$  describe their respective control torques produced by the differential velocities of the rotors. Remember that the *AR.Drone* includes an internal autopilot to deal with the attitude ( $\varphi$ ,  $\theta$  and  $\psi$ ) control and the altitude ( $z$ ), to follow some desired reference. Therefore, our control input vector is  $u = [\varphi_d \ \theta_d \ \dot{z}_d \ \dot{\psi}_d]^T$ , i.e., the desired roll, pitch angles and altitude and yaw velocities. This way, we only need to worry about controlling the remaining states,  $x$  and  $y$  position, to follow a desired reference, using the roll and pitch references ( $\varphi_d$  and  $\theta_d$ ) as virtual control inputs. This is achieved thanks to a time scale separation of the translational and rotational dynamics, since the rotational is much faster than the translational one [14].

### 5.4.1 PD Control

Let us focus our study on the  $x$  and  $y$  equations. Thus, assuming small variations in the roll and pitch angles, i.e. only non-aggressive maneuvers are considered, we can linearize

around the equilibrium point  $\varphi_d \approx \theta_d \approx 0$ , it is:

$$m \begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} \approx T \begin{bmatrix} s\psi & c\psi \\ -c\psi & s\psi \end{bmatrix} \begin{bmatrix} \varphi_d \\ \theta_d \end{bmatrix} \quad (5.9)$$

Note the introduction of the desired angles instead of the real ones, this is possible since  $\varphi_d \approx \varphi$  and  $\theta_d \approx \theta$  due to the action of the autopilot attitude control loop. Denote now

$$\begin{bmatrix} \hat{\varphi} \\ \hat{\theta} \end{bmatrix} = \begin{bmatrix} s\psi & c\psi \\ -c\psi & s\psi \end{bmatrix} \begin{bmatrix} \varphi_d \\ \theta_d \end{bmatrix} \quad (5.10)$$

then

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} = \frac{T}{\bar{m}} \begin{bmatrix} \hat{\varphi} \\ \hat{\theta} \end{bmatrix} \quad (5.11)$$

where  $\hat{\varphi}$  and  $\hat{\theta}$  are used as the new virtual control inputs for the position, and are chosen such that it follows certain references  $x_d, y_d$ . Thus, we can propose a PD control

$$\begin{bmatrix} \hat{\varphi} \\ \hat{\theta} \end{bmatrix} = \frac{m}{T} \begin{bmatrix} -k_{px}(x - x_d) - k_{dx}(\dot{x} - \dot{x}_d) \\ -k_{py}(y - y_d) - k_{dy}(\dot{y} - \dot{y}_d) \end{bmatrix} \quad (5.12)$$

with the control gains  $k_{px}, k_{py}, k_{dx}, k_{dy} \in \mathbb{R}^+$ . Transforming to the original coordinates

$$\begin{bmatrix} \varphi_d \\ \theta_d \end{bmatrix} = \frac{m}{T} \begin{bmatrix} s\psi & -c\psi \\ c\psi & s\psi \end{bmatrix} \begin{bmatrix} -k_{px}(x - x_d) - k_{dx}(\dot{x} - \dot{x}_d) \\ -k_{py}(y - y_d) - k_{dy}(\dot{y} - \dot{y}_d) \end{bmatrix} \quad (5.13)$$

In this work, we are interested in the vehicle position relative to the human face, rather than global movements. Also, for this first experience, it is supposed that the tracked human is looking straight to the drone, hence the relative yaw rotation always remains small.

As for the yaw  $\psi$  and altitude  $z$  controllers, employing the control parameters  $k_{p\psi}, k_{pz}, k_{d\psi}$  and  $k_{dz} \in \mathbb{R}^+$ , we have

$$\dot{\psi}_d = -k_{p\psi}(\psi - \psi_d) - k_{d\psi}\dot{\psi} \quad (5.14)$$

$$\dot{z}_d = -k_{pz}(z - z_d) - k_{dz}\dot{z} \quad (5.15)$$

for some desired  $\psi_d, z_d$ . For the face following problem, the desired position  $[x_d y_d z_d]^T$  is obtained by the vision algorithm. However the desired yaw orientation  $\psi_d$  has to be manually set in order to coincide with the face orientation.

### 5.4.2 Altitude speed estimation

It is important to notice that the previous feedback control strategy relies on a good measurement of the system states and their derivatives. Relative position is used straight from the vision algorithm. However, velocities from the vision algorithm result to be imprecise and noisy, and not to trust for control feedback. This is not a problem for the  $x$  and  $y$  coordinates since the utilized platform is equipped with a down looking camera from which the optic flow is calculated to estimate the horizontal velocities  $\dot{x}$  and  $\dot{y}$ . Nevertheless, problems arrive because no measurement is available for the altitude speed  $\dot{z}$ .

To overcome this issue, two estimation techniques were implemented and tested for the altitude speed, based on the altitude measurements  $z_m$  from the ultrasound sensor integrated on the bottom of the helicopter. The first method consists on the classical Euler derivative of the measurement  $z_m$ , it is

$$\dot{z}_m(k) = \frac{z_m(k) - z_m(k - \Delta t)}{\Delta t} \quad (5.16)$$

where  $k$  and  $\Delta t$  stand for the discrete time variable and its increment. Although it offers a simple solution, this derivative is known to amplify the noise from the measurements, and can not be directly used for control feedback. For this reason, it was decided to implement a fourth order low-pass filter type Chebyshev I, at a cutoff frequency of 8Hz. This is a kind of recursive filter, and as such, it offers a fast response [71]. The following equation presents the filter algorithm with input  $\dot{z}_m$  and output  $\bar{\dot{z}}_m$

$$\bar{z}_m(k) = \sum_{i=0}^4 (a_i \dot{z}_m(k - i\Delta t) + b_i \bar{z}_m(k - i\Delta t)) \quad (5.17)$$

where the constant coefficients  $a_i$  and  $b_i$  are obtained with the help of *MATLAB/fdatool*.

The result is a smooth and quite acceptable estimation of the speed, except for a little delay of about 4 steps produced by the filter. This response is good enough for several applications where dynamics are slow or sampling frequency is large enough to neglect this delay. However, for UAV control feedback this can cause instability or poor performance in the closed-loop system.

The second explored method consists of a Luenberger state observer [51]. A state observer provides estimates of the internal states of the system from the inputs and output measurements. Let us consider the discrete-time kinematic model of the altitude at instant time  $k$

$$\begin{bmatrix} z_k \\ \dot{z}_k \end{bmatrix} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} z_{k-1} \\ \dot{z}_{k-1} \end{bmatrix} + \begin{bmatrix} 0 \\ \ddot{z}_k \Delta t \end{bmatrix} \quad (5.18)$$

$$y = [1 \ 0] \begin{bmatrix} z_{k-1} \\ \dot{z}_{k-1} \end{bmatrix} \quad (5.19)$$

where  $y$  is the output. Then, a Luenberger observer is proposed, according to the following equations

$$\begin{bmatrix} \hat{z}_k \\ \dot{\hat{z}}_k \end{bmatrix} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{z}_{k-1} \\ \dot{\hat{z}}_{k-1} \end{bmatrix} + \begin{bmatrix} L_1 \\ L_2 \end{bmatrix} [y - \hat{y}] + \begin{bmatrix} 0 \\ \ddot{z}_k \Delta t \end{bmatrix} \quad (5.20)$$

$$\hat{y} = [1 \ 0] \begin{bmatrix} \hat{z}_{k-1} \\ \dot{\hat{z}}_{k-1} \end{bmatrix} \quad (5.21)$$

Note the use of a hat on the variables to indicate that they are state estimates rather than the real ones. In order to guarantee that the estimated states  $[\hat{z} \ \dot{\hat{z}}]^T$  converge asymptotically to

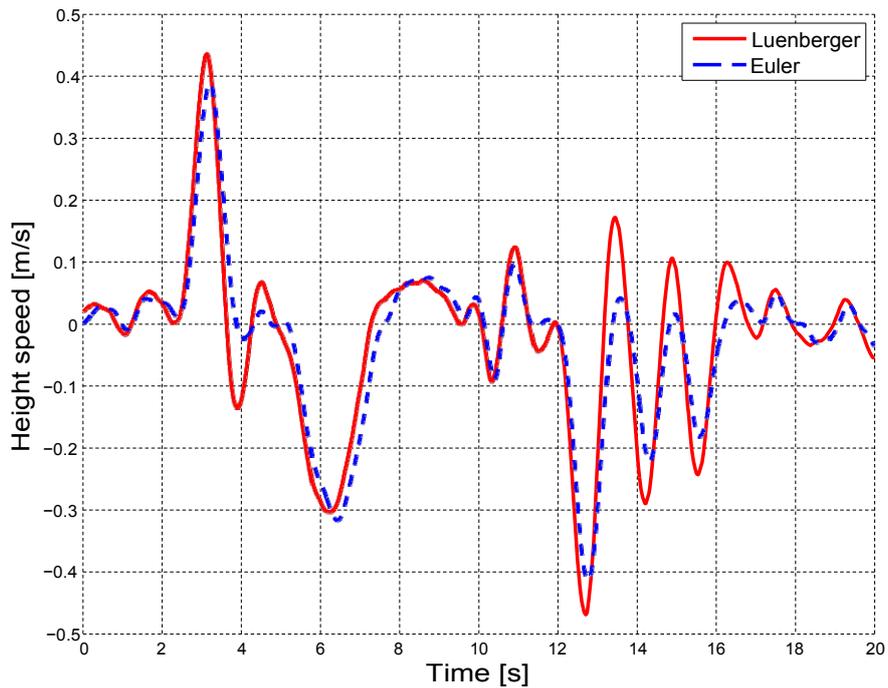


Figure 5-6: Altitude speed estimation comparison.

the real ones, the observer gains  $L_1$ ,  $L_2$  are chosen such that the matrix

$$\begin{bmatrix} 1 - L_1 \Delta t & \\ -L_2 & 1 \end{bmatrix}$$

has all its eigenvalues inside the unit circle.

Figure 5-6 allows us to compare both estimation techniques, where it is clear that the two of them obey in general the same behavior, however, the estimation given by the Euler derivative combined with the Chebyshev filter (dashed blue line) attenuates a little bit the signal amplitude and introduces a small undesired delay. This is corrected by using only a Luenberger observer instead (solid red line), which proves to be an excellent option for this case.

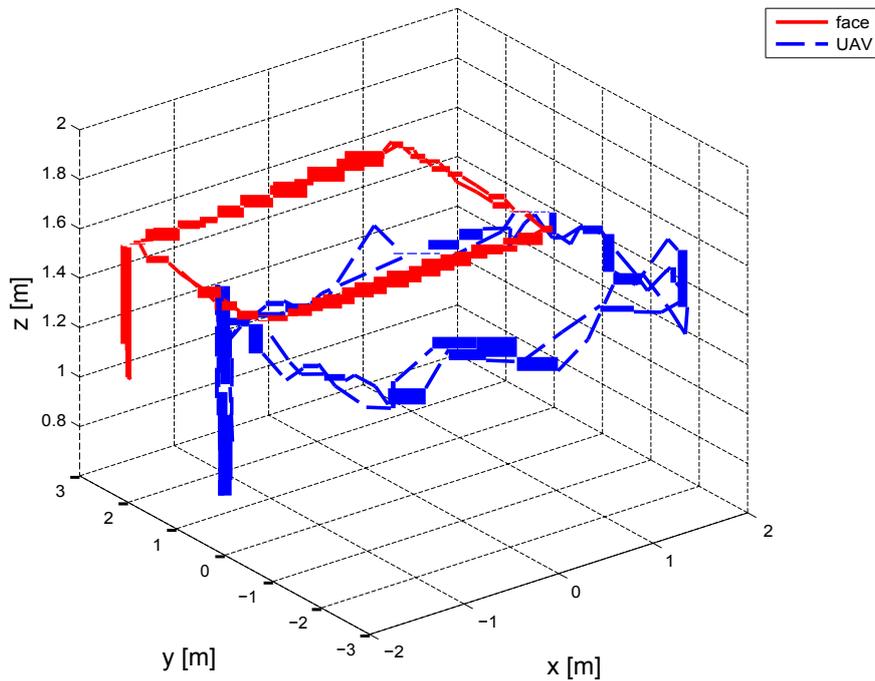


Figure 5-7: 3D trajectory.

## 5.5 Experimental Results

Several experiments were carried out to test the performance of the proposed algorithms on indoor and outdoor flight. In the following, some results are illustrated in Figures 5-7 to 5-12. The mission consisted for the quadrotor to try to keep a constant distance of  $2m$  in front of the face while the followed human walked through the room in a rectangular trajectory, as can be appreciated in Fig. 5-7. After completing a couple of trajectories, the third coordinate for the altitude is tested by descending and rising a few times. Two main results are studied, the performance of the computer vision algorithm for detecting and estimating the relative position between the quadrotor and the face, and the behavior of the relative position control while following the moving reference. An *OptiTrack* motion capture system, composed of twelve infrared cameras, is used to know with a precision within millimeters the exact location of the human and the drone. This information is used only as ground-truth to compare with the proposed algorithms and quantify its precision, but the real-time system does not depend nor makes any use of it. For a better understanding, a

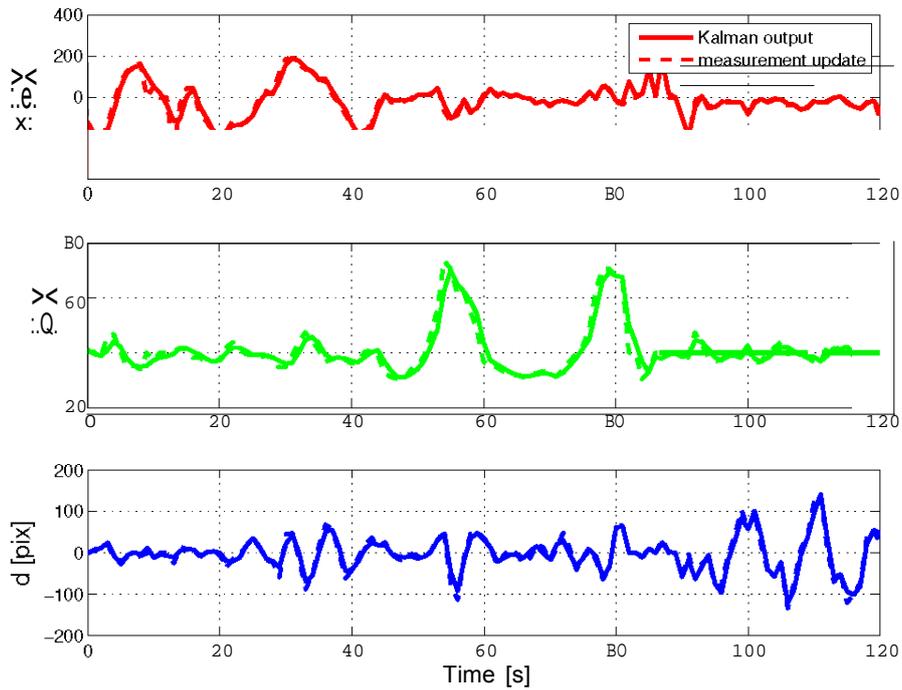


Figure 5-8: Kalman filter tracking.

video is available at <https://www.youtube.com/watch?v=p8zW4jNrsFO>.

### 5.5.1 Face Detection and Tracking

The performance of the face tracking by means of a Kalman filter is presented in Fig. 5-8. It is important to clarify that this is the only graph that does not correspond to the same experiment represented in the rest of the figures. This was done for the sake of better exemplifying the Kalman filter action, by imposing harder conditions to the detection algorithm to increase the number of false positive detections. From the figure we can observe how the tracking algorithm smooths the estimated position and handles the erroneous measurement produced by false positives and other faces on the image, like the one at second 13 for the x coordinate.

Let us now take a look at the estimation of the relative position obtained by the computer vision algorithm. For that purpose, it is crucial to have a good estimation of the depth  $Y_w$ , since the other two variables directly depend on it. A comparison between the relative

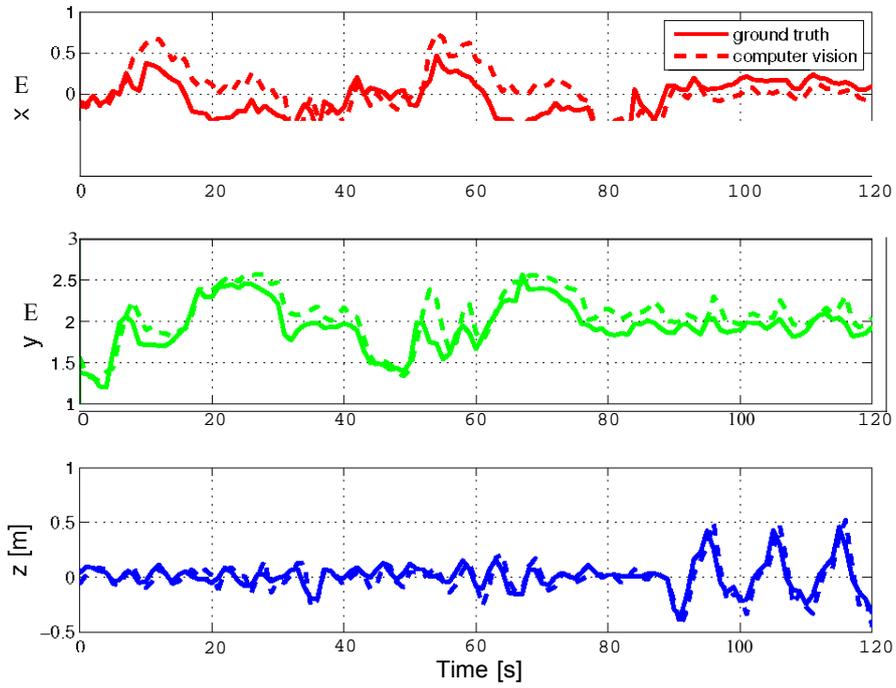


Figure 5-9: Relative position estimated by the computer vision algorithm vs ground truth.

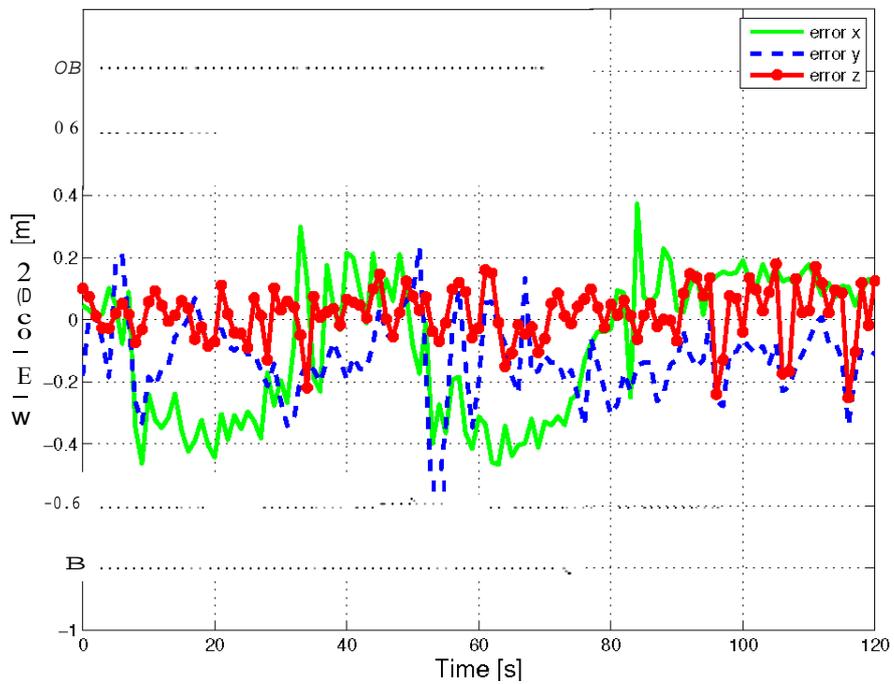


Figure 5-10: Computer vision estimation error.

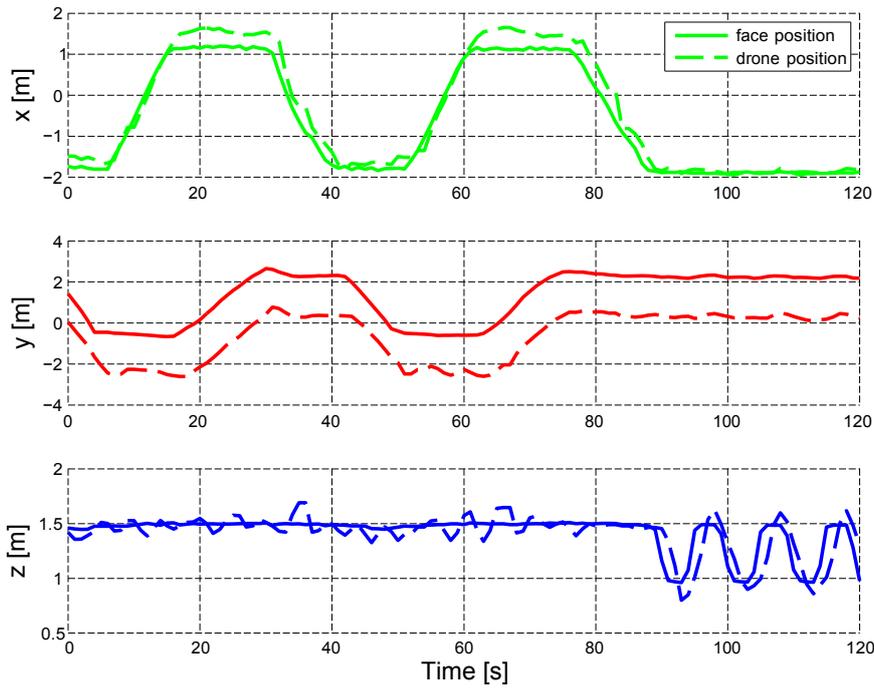


Figure 5-11: Relative position control performance.

position estimated by the proposed algorithm and ground truth data is exposed in Fig. 5-9, while Fig. 5-10 displays the estimation error, with a Root Mean Square Error (RMSE) of  $0.1294m$  in  $x$ ,  $0.2392m$  in  $y$  and  $0.1802m$  in  $z$ . Those results are quite satisfactory for this kind of application, taking into account that we are using a low-cost camera mounted on a moving object, and the simplicity of the employed algorithms.

### 5.5.2 Relative Position Control Performance

Finally, we can analyze the good behavior of the PD relative position controller from Figures 5-11 and 5-12, which present the face position against the drone position, and the relative position, respectively. Remember that the practical objective consists in keeping a distance of  $2m$  in front of the targeted face ( $y$  coordinate). Hence, Figures 5-11 and 5-12 prove the effectiveness of the overall system, being able to detect, track and follow a moving face.

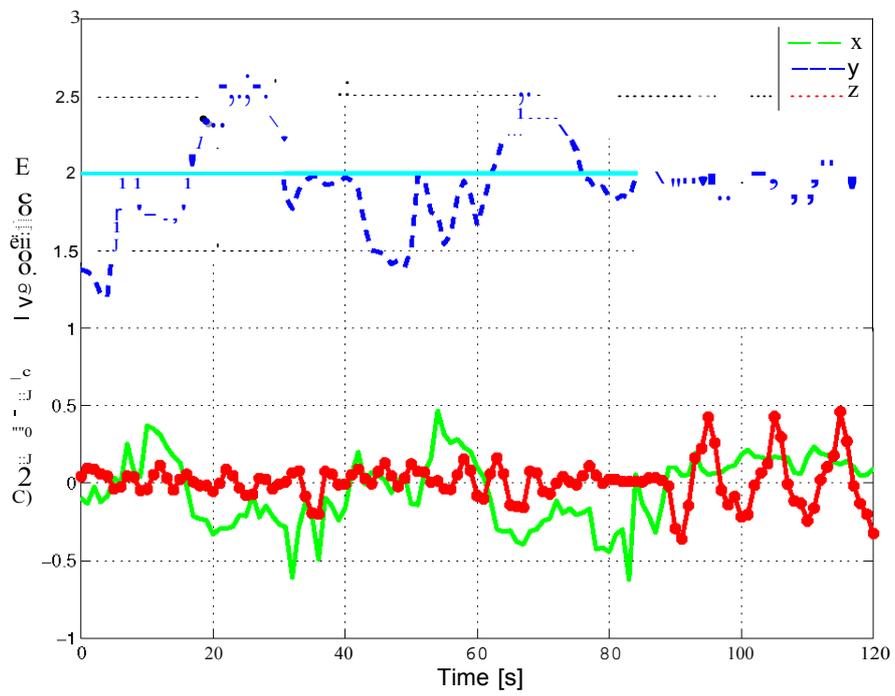


Figure 5-12: Quadrotor position relative to the face.

## **Chapter 6**

# **Implementation of a Fast and Safe Transportation System using a Quadrotor with a Cable-Suspended Payload**

This chapter deals with the problem of implementing a safe and fast transportation system using an Unmanned Aerial Vehicle kind quadrotor with a cable suspended load. The objective consists in transporting the load from one place to another in a fast way while reducing the swing on the cable. An Interconnection and Damping Assignment-Passivity Based Control (IDA-PBC) for the system of interest is successfully implemented and tested in real time experiments. The use of this control technique avoids the use of an extra sensor for the swing angle measurement. An experimental platform, composed by a quadrotor equipped with a rigid rod with two degrees of freedom in the joint, is developed. UAV autonomous navigation is achieved via monocular vision.



Figure 6-1: Experimental platform for a quadrotor with a cable suspended load.

## 6.1 Introduction

The autonomous aerial transportation of a cable-suspended payload is an interesting and important topic that has attracted the research interest in recent years. It can be applied to a wide range of useful tasks, such as deploying supplies in military operations, or delivering first-aid kits for personal assistance to the victims in disasters like floods, earthquakes, fires, industrial accidents, among others. It is also a basic technology for other future applications, such as, building platforms in hazardous environments [13]. Even more, important global companies for delivery services have announced their wide interest in using UAVs in the short term for package delivery.

These applications often require aggressive maneuvers for fast transportation of fragile payloads. The pendulum like behavior of the slung load gives a high risk of oscillations that can result in dangerous situations [42]. Furthermore, unstable oscillations can occur at high speeds due to the different aerodynamic shapes of the slung loads. Hence, it is very

important to cancel or reduce oscillations of the suspended payload to avoid damage on the load, the environment and the people around, such that safe operation is guaranteed.

The design of controllers that rapidly stabilize the payload swing and the general problem of an UAV transporting a cable-suspended payload, as the one depicted in Figure 6-1, has been investigated in a few recent works. For example, in [23] is designed an adaptive controller for a quadrotor UAV transporting a point-mass payload connected by a flexible cable. The cable is modeled as serially-connected rigid links and the payload is modeled as a point-mass. Meanwhile, in [62] is explored the effect of dynamic load disturbances introduced by instantaneously increased payload mass and how this affects a quadrotor under Proportional Integral Derivative flight control. Also, experiments of transportation and accurate deployment of payloads with single/multiple autonomous aerial vehicles are presented in [12] and [60].

On the other hand, in [34] and [22] is developed a geometric control for position tracking of a quadrotor, using a coordinate-free dynamic model of the referred system. As well, in [79] it is also developed a geometric control but for multiple quadrotors transporting a rigid-body load. A technique based on dynamic programming which ensures swing-free trajectory tracking is proposed in [61] and [31]. Moreover, in [72] and [73] is utilized the differentially-flat property to design trajectories and preliminary results have been demonstrated in simulations and in experiments for a quadrotor UAV transporting a cable-suspended load along the longitudinal plane. A model based algorithm in order to control the position and attitude of the system of interest is used in [66]. However, in some of the above mentioned works, the precise payload positioning with minimal oscillation is achieved for slow maneuvers. Also, it is noted that the control laws explicitly depend on the swing angle of the cable, and they are subjected to an expensive motion capture system for the implementation. In this article the IDA- PBC methodology is used for precise and fast payload positioning with minimum swing in a short time, without dependence on the angle of oscillation, and without requiring an external motion capture system for implementation.

In brief, this work introduces the practical implementation of a transportation system using a quadrotor with a cable suspended payload, using passivity based techniques to asymptotically stabilize the swing angle oscillations without a direct measurement from

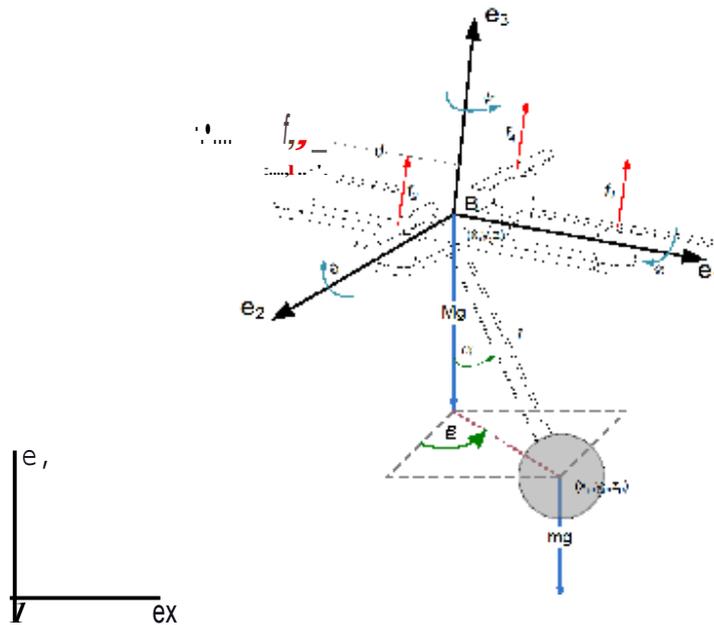


Figure 6-2: Quadrotor with a cable suspended payload.

them, see Figure 6-2. The proposed solution utilizes monocular vision techniques for precise localization and is not constrained to any external motion capture system.

This chapter is organized as follows: section 6.2 briefly introduces the dynamic model and the passivity based control strategy of the transportation system. Following, section 6.3 describes the practical implementation of the proposed configuration. Then, experimental results corroborate the validity of the overall proposed system and the good performance of the control scheme. A special acknowledgement is finally offered in section 6.5.

## 6.2 Control Strategy

Interconnection and Damping Assignment is a Passivity Based Control methodology applied to a special class of underactuated mechanical systems, which is based on systems described by the Port Controlled Hamiltonian (PCH) model. This methodology consists in assigning a new PCH model in closed loop with the interconnection and damping assign-

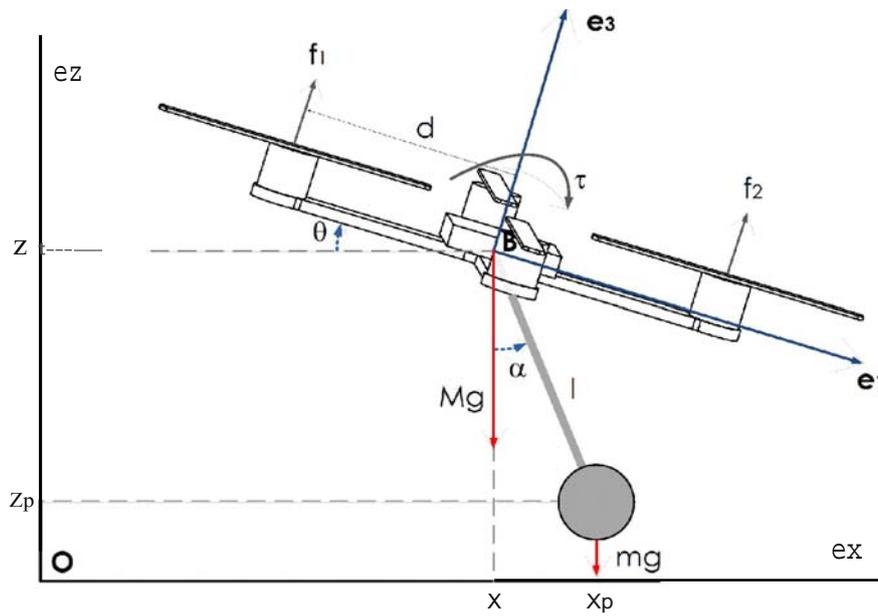


Figure 6-3: Quadrotor with a cable suspended payload, on the longitudinal plane.

ment.

In the present work, the IDA-PBC methodology is used to incorporate information about the underactuated system structure and to deal with the concept of energy in the control strategy. The main advantage of the IDA-PBC strategy is its inherent robustness with respect to unmodeled dynamics and nonparametric uncertainties. Furthermore, its inherent energy minimization allows to attenuate the undesired swing oscillations on the load without any direct measurement for it, hence, no extra sensors are required.

For simplicity, we consider only planar motions at a time, for both the longitudinal and transversal plane, then, a similar control law was implemented for both cases. Let us consider a quadrotor UAV with mass  $M$  evolving in the longitudinal plane, with a payload of punctual mass  $m$  connected to the drone by a rigid massless rod of length  $l$ , as illustrated at Figure 6-3. Consider as well an inertial coordinate frame  $\mathbf{O} = [e_x \ 0 \ e_z]^T$  fixed to the ground, and a body fixed frame  $\mathbf{B} = [e_1 \ 0 \ e_3]^T$ . The UAV position with respect to the inertial frame  $\mathbf{O}$  is described by  $x$  and  $z$ .  $\delta$  stands for the pitch Euler angle and  $\alpha$  represents

the swing angle. Then, the dynamical model for the studied system along the longitudinal plane can be obtained by the Euler-Lagrange method as [36]

$$-f \sin \theta = (M + m)\ddot{x} + ml(\ddot{\alpha} \cos \alpha - \dot{\alpha}^2 \sin \alpha) \quad (6.1)$$

$$f \sin \theta = (M + m)\ddot{z} + ml(\ddot{\alpha} \sin \alpha - \dot{\alpha}^2 \cos \alpha) + (M + m)g \quad (6.2)$$

$$\tau_\theta = I_\theta \ddot{\theta} \quad (6.3)$$

$$0 = l\ddot{\alpha} + \ddot{x} \cos \alpha + \ddot{z} \sin \alpha + g \sin \alpha \quad (6.4)$$

where  $g$  is the gravity constant and  $I_\theta$  is the mass moment of inertia of the quadrotor.  $f = f_1 + f_2$  and  $\tau_\theta$  are the total thrust and the input torque, and are considered as the control inputs. Observe that the system (6.1)-(6.4) is underactuated, with only two control inputs for four state variables. Note as well that a similar model can be used for the transversal plane.

Then, an IDA-PBC methodology can be applied to design a suitable control strategy for the longitudinal plane. In this case the following control law is chosen for implementation (further details about the development of the control scheme can be found at [37]).

$$f_{long} = -k_b \dot{x} - k_d \dot{z} - k_e \dot{\theta} + (M + m)g - K_{pz}(z - z_D) \quad (6.5)$$

$$\tau_\theta = -(k_a + k_c)\dot{x} - (k_b + k_e)\dot{z} - (k_c + k_f)\dot{\theta} - K_{px}(x - x_D) - K_{p\theta}(\theta - \theta_D) \quad (6.6)$$

where  $f_{long}$  is the thrust force in the longitudinal plan,  $\tau_\theta$  stands for the input torque on the pitch angle. Meanwhile,  $k_a, k_b, k_c, k_d, k_e, k_f, K_{px}, K_{pz}$  and  $K_{p\theta}$  are constant control parameters.

An equivalent control law can be found for the transverse plane ( $f_{trans}, \tau_\varphi$ ). Note that motions are only allowed in one plane at a time due to the coupling between them.

## 6.3 Real-Time Implementation

In order to validate the proposed control strategy, an experimental platform was developed. It consists in a low-cost commercial quadrotor type Parrot *AR.Drone*, as the one discussed at section 2.2.1, with an extra payload of 0.05Kg weight, connected to its bottom by a two-degrees of freedom rigid rod, as showed in Figure 6-1. The position of the rod was displaced to the back of the quadrotor to avoid interference with the optic flow and ultrasonic sensors placed at the bottom of the quadrotor. The quadrotor autonomous navigation is achieved using the monocular vision techniques described in chapter 4.3.2.

Finally, the motion capture system introduced in section 2.3 is used to estimate the swing angle for the payload. It is important to note that the described system does not depend on the motion capture system, and it is only used to obtain the graph of the payload's swing angle.

## 6.4 Experimental results

Real time experiments proved the good performance of the studied control scheme despite the absence of a direct measurement of the swing angle. The experiment consisted in rapidly carrying the load through the corners of a square of 2m, at a constant height of 0.8m, as depicted in Fig. 6-4. In Figure 6-5, the UAV position is presented for a segment of the experiment with reference changes at the seconds 7 and 28. The quadrotor position error is introduced in Fig. 6-6, where is evidenced the ability of the control scheme to regulate the UAV position, and its fast response despite the perturbations induced by the added load. Note the black dotted lines indicating the reference changes. The orientation of the UAV can be appreciated in Fig. 6-7. In Figure 6-8, the oscillation angles are presented, where  $\beta$  stands for the swing angle in the transverse plane. We can observe how these swing angles are kept bounded in small values (less than  $10^\circ$  if there are not reference changes, until  $15^\circ$  otherwise) even when there is not any direct measurement available. Finally, the control inputs are shown in Figure 6-9.

It is worth mentioning that due to interference with the sensors, the rigid rod is not

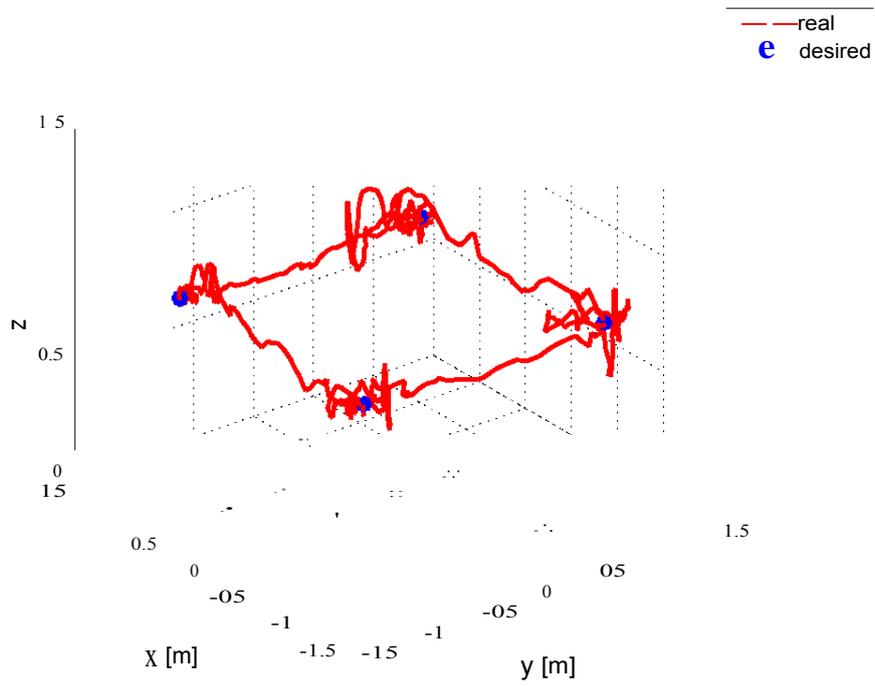


Figure 6-4: 3D view.

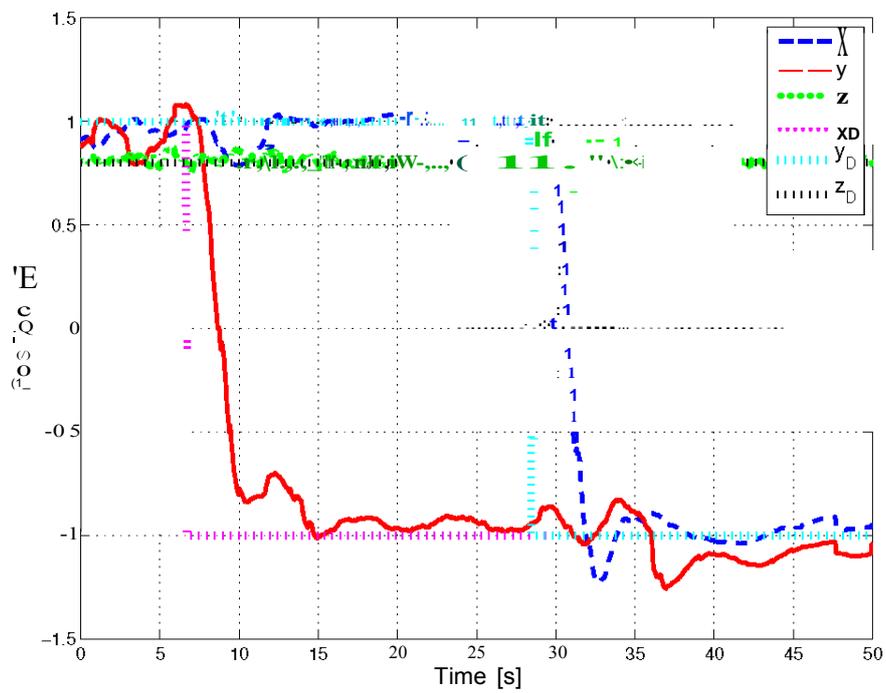


Figure 6-5: Quadrotor position.

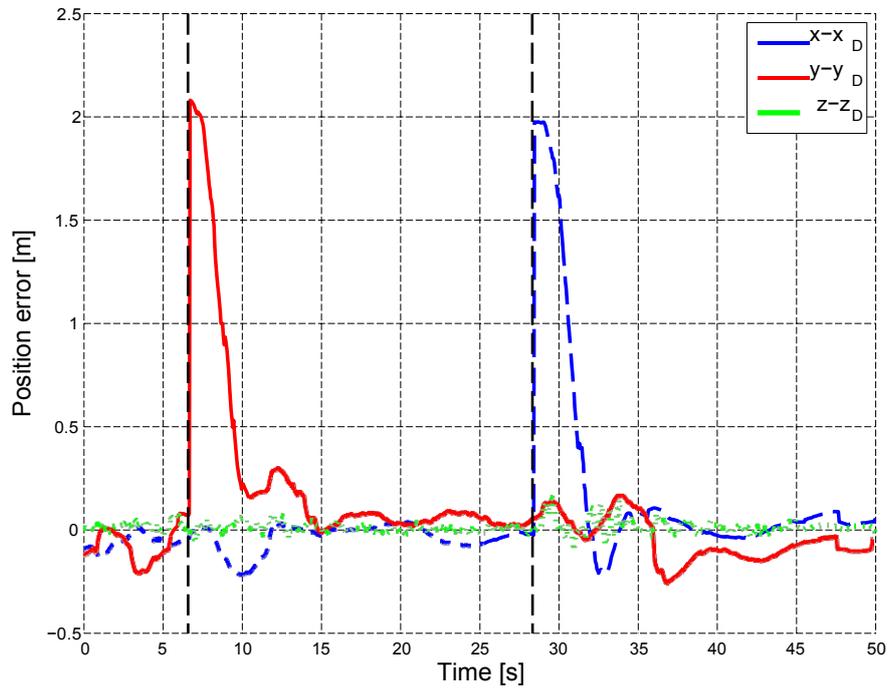


Figure 6-6: Quadrotor position error.

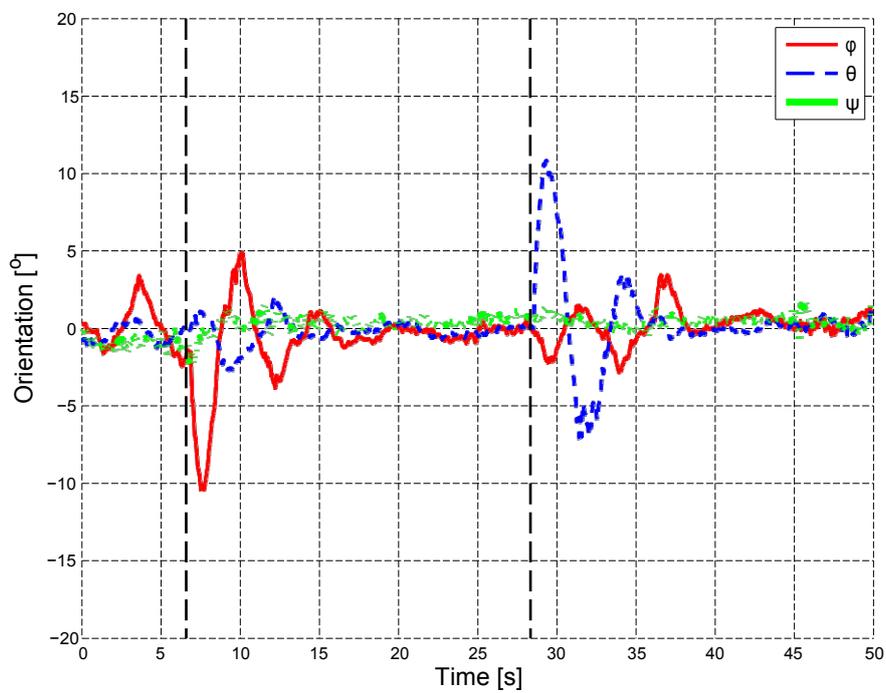


Figure 6-7: Quadrotor attitude.

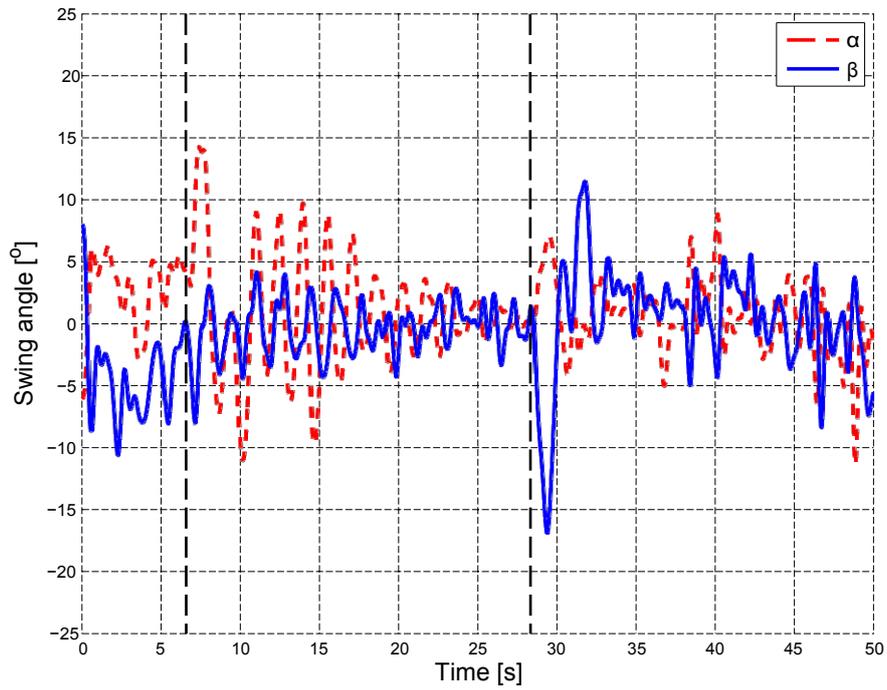


Figure 6-8: Swing angles.

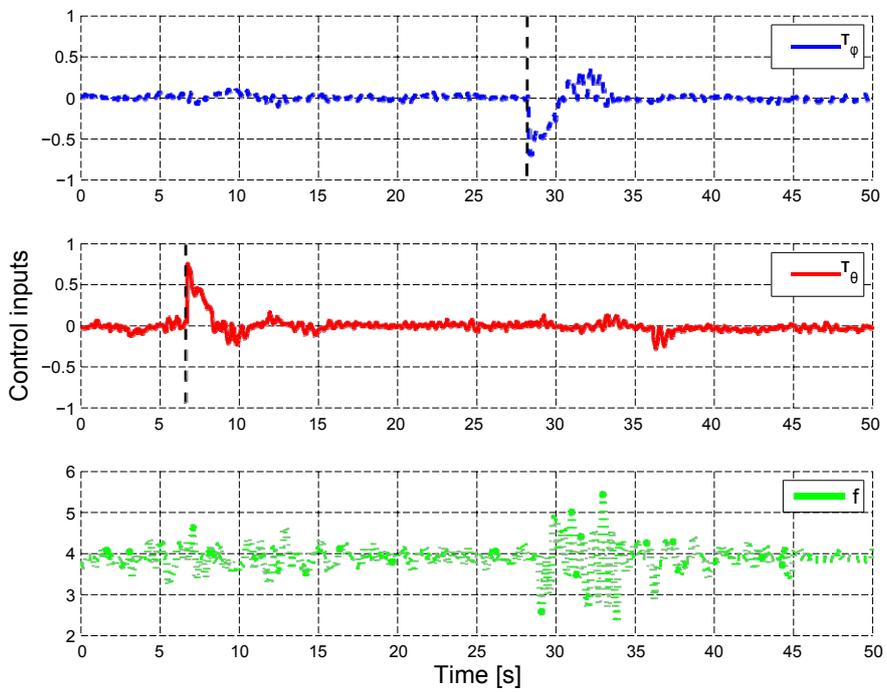


Figure 6-9: Control inputs.

aligned with the gravity center of the UAV, however, the tested controller showed robustness against this drawback and proved good performance in spite of this fact.

A video of the implemented system in action is available at:

[https://www.youtube.com/watch?v=l\\_ZP9tXLre8](https://www.youtube.com/watch?v=l_ZP9tXLre8).

## **6.5 Acknowledgements**

It is important to point out that the main objective and contribution of this work is the practical implementation of the IDA-PBC control technique for the considered problem. A special acknowledgement is given to M.Sc. Eusebia Guerrero for her theoretical contribution by obtaining the dynamic model and developing the control strategies applied in this work [36], [37].



# Chapter 7

## Conclusions and Future Work

A wide range of different interesting topics were covered in this document for UAVs autonomous navigation, including modeling, automatic control, data fusion, state estimation, computer vision, electronics and so on. This last chapter contains a discussion about the obtained results for each of the different applications investigated along this thesis, from both theoretical and practical point of view, and offers some perspectives for future developments to continue this work.

In general we can state that UAVs continue to be an exiting research field with a large number of possible civilian and military applications. A lot of research have been carried out around the world to try to solve the involved problems, however, in order to accomplish outdoor missions under real conditions a lot of work is still required.

Two main problems were considered and studied through this thesis, localization in GPS-denied environments and robustness against uncertainties and external unknown perturbations. Different approaches were studied and interesting solutions, according to the particular needs of the application, were proposed and tested in simulations and real-time experiments.

For the localization problem, data fusion techniques of commonly available sensors, and computer vision for localization with respect to either a fixed scene or to a predefined object, showed promising results that allowed successful outdoor navigation even in unknown scenarios.

Meanwhile, for the problem of robustness, two kinds of perturbations were studied,

external perturbations such as wind and oscillations due to a swinging payload. For the former case a second order sliding mode control was proposed, while for the oscillation attenuation a passivity based control was selected.

In the following, more details are presented according to the particular application.

## 7.1 Experimental Platforms

Two different experimental platforms were conceived and developed for this thesis. The first one was conceived for a fully embedded system on a quadrotor for autonomous outdoor flight. Meanwhile, the second experimental testbed consists on a non expensive commercial quadrotor and a ground computer, and was intended for fast, safe and robust implementing and evaluating new developed control strategies and computer vision algorithms.

As future work, it is intended to develop a new platform for autonomous navigation in outdoor applications, improving the one presented in this document by adding one or more video cameras and a micro computer for the vision algorithms, along with other sensors to measure optic flow and pressure, among others. Such a platform will be used to extend the proposed algorithms and techniques to a fully embedded system.

## 7.2 UAV Data Fusion for Localization

Simulations have shown promising results for both proposed data fusion strategies, considerably reducing the measurement error while handling the GPS signal losses.

The PF proved to be more adequate in this case than the EKF, since the measurement noise of a GPS in a loosely coupled scheme is not Gaussian. However, there is a trade-off between the precision of the PF and the computational cost, according to the number of employed particles. The computational cost of the PF with a good precision becomes then much higher than that for the EKF, deriving in a slower computation speed and making it unsuitable for embedded real time control of UAVs.

Therefore, none of the studied algorithms offers a fully satisfactory performance, but both can be used under certain conditions for a specific mission. This motivates our search

for other alternatives for UAVs localization.

Future work includes to implement the proposed strategies on a quadrotor equipped with a GPS sensor, for real time outdoor experiments.

### **7.3 Collision-Free Navigation and Haptic Teleoperation**

Collision-free navigation and teleoperation for a quadrotor using only a monocular camera were presented and experimentally validated in this thesis. Two main cases were studied: autonomous collision-free trajectory tracking and collision-free teleoperation. Despite the simplicity of the proposed collision avoidance technique, it satisfactorily accomplished its goal for both studied cases. Meanwhile, monocular-vision based navigation showed to be a powerful tool for MAVs in GPS denied environments, as well as an exciting research area to be explored.

Second order sliding mode control proved to be adequate for accurate trajectory tracking with the presented setup, accomplishing well the tracking objective while adding robustness against external perturbations. However, special attention must be paid when implementing this kind of controllers because of their discontinuous nature, in this case, the discontinuous terms approximated by a high gain saturation, and the limits were kept bounded by choosing small gains. A good solution for future developments could be to use the Super Twisting Algorithm instead, in order to avoid the chattering effect.

Haptic assisted teleoperation has been demonstrated to be an interesting alternative for safe UAV navigation, preventing the user from crashing the UAV and improving the flying experience.

Future work includes to extend the obtained results for collision avoidance in 3D unstructured scenarios and outdoor tests. Also, the design of a new sliding mode control is envisioned to avoid the problem of discontinuity in the input, using the super twisting algorithm.

## 7.4 The Face Follower UAV

An innovative application for an UAV capable to interact with humans by following a face was conceived and successfully developed in this work. In order to do so, several tools and techniques were merged together to offer a full-working solution.

Relative position from the face to the quadrotor was estimated by a computer vision algorithm. Face detection was accomplished by means of a Haar cascade classifier, while a Kalman filter was implemented to keep track of the face of interest, adding robustness against false positive detections and other faces on the image. Then a suitable transformation was proposed, by using the size of the detected face to compute the depth to the face, normally unknown for monocular vision algorithms.

A PD control strategy was implemented to deal in real-time with the relative position regulation to a desired value. To complete the state feedback for the controller, a Luenberger observer was employed to estimate the missing altitude speed. The overall system performance was extensively tested in real-time experiments under different conditions, and proved to be an excellent first-approach solution for the studied problem, despite the use of a low-cost quadrotor and the simplicity of the algorithms.

For the sake of security, the used algorithms and gains were chosen to be conservative. Hence, we envision to improve the proposed system by using more precise models, and applying more sophisticated algorithms for estimation and automatic control. Also, a wide angle lens would help to increase the field of view, allowing faster motions. Furthermore, it would be interesting to implement the proposed algorithms in a fully embedded UAV.

This new application presents an interesting problem that can be used as a base for future applications and developments, looking to improve the human-robot interaction experience. There, imagination is the limit, but we can cite some examples:

- The detection algorithm can be trained to track almost any other kind of mostly rigid object with distinguishing views. Then, the proposed system can be used as it is to follow the desired object.
- Detecting face gestures and body movements could be used to give further commands to the quadrotor.

- The face-follower UAV could be programmed to play some popular games like *tag* or the *hot potato*.
- A naive but powerful and interesting application would be to attach the image of a face to another quadrotor and use the developed system for formation flight applications.

## 7.5 Fast and Safe Transportation System

We have implemented an IDA-PBC strategy to minimize the swing of a payload connected by a cable to a quadrotor. The employed control law was found to be performing well for the control objective of moving a payload from any initial position to the desired position, with minimum swing. The advantage of the implemented control scheme is that it does not require an extra measurement for the oscillation angle, since it does not depend on it. The control strategy was successfully implemented and validated in real time experiments, showing its good performance. A monocular vision based algorithm was employed for localization, avoiding the dependence on an external motion capture system, or any other external reference. Future developments include to design a controller for 3D motions taking into account the coupling between planes. Also, study the problem for non-rigid cables is envisioned.



# Bibliography

- [1] Bionicopter web site. <<https://www.festo.com/group/en/cms/10224.htm>>.
- [2] Lily web site. <<https://www.lily.camera/>>.
- [3] Myo web site. <<https://www.myo.com/>>.
- [4] Smartbird web site. <<https://www.festo.com/group/en/cms/10238.htm>>.
- [5] Tms320f2810, tms320f2811, tms320f2812, tms320c2810, tms320c2811, tms320c2812 digital signal processors. Data manual, Texas Instrument, Computer Science Department, Fanstord, California, USA, 2001.
- [6] Controlling a quadrotor using kinect. <<http://spectrum.ieee.org/automaton/robotics/robotics-software/quadrotor-interaction>>, 2011. ETH Zurich.
- [7] M. Achtelik, M. Achtelik, S. Weiss, and R. Siegwart. Onboard imu and monocular vision based control for mavs in unknown in- and outdoor environments. *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 2011.
- [8] H. Alvarez, L. M. Paz, J. Sturm, and D. Cremers. Collision avoidance for quadrotors with a monocular camera. *Proc. of The 12th International Symposium on Experimental Robotics (ISER)*, 2014.
- [9] I. Asimov. *I, Robot*. 1950.
- [10] F. Augugliaro, S. Lupashin, M. Hamer, C. Male, M. Hehn, M.W. Mueller, J. Willmann, F. Gramazio, M. Kohler, and R. D'Andrea. The flight assembled architecture installation: Cooperative construction with flying machines. *IEEE Control Systems Magazine*, 34(4):46–64, 2014.
- [11] F. Augugliaro, A. Schoellig, and R. D'Andrea. Dance of the flying machines: Methods for designing and executing an aerial dance choreography. *IEEE Robotics and Automation Magazine*, pages 96–104, 2013.
- [12] M. Bernard and K. Kondak. Generic slung load transportation system using small size helicopters. *IEEE International Conference on Robotics and Automation (ICRA)*, 2009.

- [13] M. Bernard, K. Kondak, I. Maza, and A. Ollero. Autonomous transportation and deployment with aerial robots for search and rescue missions. *Journal of Field Robotics*, 28(6):914–932, 2011.
- [14] S. Bertrand, N. Guénard, T. Hamel, H. Piet-Lahanier, and L. Eck. A hierarchical controller for miniature vtol uavs: Design and stability analysis using singular perturbation theory. *Control Engineering Practice*, 19:1099–1108, 2011.
- [15] A. Beyeler, J. Zufferey, and D. Floreano. 3d vision-based navigation for indoor microflyers. *Proceedings IEEE International Conference on Robotics and Automation (ICRA)*, 2007.
- [16] C. Bills, J. Chen, and A. Saxena. Autonomous mav flight in indoor environments using single image perspective cues. *IEEE International Conference on Robotics and Automation (ICRA)*, 2011.
- [17] G. Bradski and A. Kaehler. *Learning OpenCV*. O’Reilly, first edition, 2008.
- [18] A. Brandt and M. Colton. Haptic collision avoidance for a remotely operated quadrotor uav in indoor environments. *IEEE International Conference on Systems, Man, and Cybernetics*, 2010.
- [19] R. Brown and Y. Hwang. Introduction to random signals and applied kalman filtering. 1992.
- [20] P. Castillo, R. Lozano, and A. Dzul. *Modelling and control of mini-flying machines*. Springer-Verlag, Londres, UK, 2005.
- [21] Z. Chen. *Bayesian filtering: From kalman filters to particle filters, and beyond*, volume 47. McMaster University, 2003.
- [22] P. Cruz and R. Fierro. Autonomous lift of a cable-suspended load by an unmanned aerial robot. *IEEE Conference on Control Applications (CCA)*.
- [23] S. Dai, T. Lee, and D. Bernstein. Adaptive control of a quadrotor uav transporting a cable-suspended load with unknown mass. *IEEE Conference on Decision and Control (CDC)*, 2014.
- [24] L. Derafa, L. Fridman, A. Benallegue, and Ouldali. Super twisting control algorithm for the four rotors helicopter attitude tracking problem. *In 11th International Workshop on Variable Structure Systems*, 2010.
- [25] W. Ding, J. Wang, and A. Almagbile. Adaptive filter design for uav navigation with gps/ins/optical flow integration. *International Conference on Electrical and Control Engineering*, 2010.
- [26] X. Ding and Y. Yu. Motion planning and stabilization control of a multipropeller multifunction aerial robot. *IEEE/ASME Transactions on Mechatronics*, 18(2):645–656, 2012.

- [27] C. Doyle, J. Bird, T. Isom, J. Kallman, D. Bareiss, D. Dunlop, R. King, J. Abbott, and M. Minor. An avian-inspired passive mechanism for quadrotor perching. *IEEE/ASME Transactions on Mechatronics*, 18(2):506–517, 2013.
- [28] J. Engel, J. Sturm, and D. Cremers. Camera-based navigation of a low-cost quadrotor. *Proc. IEEE Intl. Conf. on Intelligent Robot Systems (IROS)*, 2002.
- [29] J. Engel, J. Sturm, and D. Cremers. Scale-aware navigation of a low-cost quadrotor with a monocular camera. *Robotics and Autonomous Systems*, 62(11):1646–1656, 2014.
- [30] B. Etkin. *Dynamics of flight*. Wiley & Sons, New York, USA, 1959.
- [31] A. Faustl, I. Palunko, P. Cruz, F. Fierro, and L. Tapia. Learning swing-free trajectories for uavs with suspended load. *IEEE International Conference on Robotics and Automation (ICRA)*, 2013.
- [32] M. Fortsch and T. Endres. <<http://www.intelfreepress.com/news/drones-fly-hands-free-with-gestural-technology/6877/>>.
- [33] M. George and S. Sukkarieh. Tightly coupled ins/gps with bias estimation for uav applications. *In Proceedings of the 2005 Australian Conferences on Robotics and Automation*, 2005.
- [34] A. Goodarzi, D. Lee, and T. Lee. Geometric stabilization of a quadrotor uav with a payload connected by flexible cable. *American Control Conference (ACC)*, 2014.
- [35] M. Grewal and A. Andrews. *Kalman filtering: theory and practice using MATLAB*. John Wiley & Sons, 2001.
- [36] E. Guerrero, D. Mercado, R. Lozano, and C. García. Ida-pbc methodology for a quadrotor uav transporting a cable-suspended payload. *IEEE International Conference on Unmanned Aircraft Systems (ICUAS)*, 2015.
- [37] E. Guerrero, D. Mercado, R. Lozano, and C. García. Passivity based control for a quadrotor uav transporting a cable-suspended payload with minimum swing. *IEEE Conference on Decision and Control (CDC)*, 2015 (unpublished).
- [38] M. Guisser, H. Medromi, J. Saadi, and H. Ifassiouen. Stabilization and trajectory tracking in discrete-time of an autonomous four rotor mini-rotorcraft. *Proceedings of the 2th International Symposium on Communications, Control and Signal Processing*, 2006.
- [39] C. Hajiyev and H. Ersin. Robust adaptive kalman filter for estimation of uav dynamics in the presence of sensor/actuator faults. *Aerospace Science and Technology*, 28:376–383, 2013.
- [40] T. Hamel, R. Mahony, R. Lozano, and J. Ostrowski. Dynamic modeling and configuration stabilization for a x4-flyer. *IFAC 15th Triennial World Congress*, 2002.

- [41] G. Heredia, A. Jimenez-Cano, I. Sanchez, D. Llorente, V. Vega, J. Braga, J. Acosta, and A. Ollero. Control of a multirotor outdoor aerial manipulator. *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2014.
- [42] R. Hoh, R. Heffley, and D. Mitchel. Development of handling qualities criteria for rotorcraft with externally slung loads. NASA, 2006.
- [43] R. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 1960.
- [44] F. Kendoul, D. Lara, I. Fantoni, and R. Lozano. Nonlinear control for systems with bounded inputs: real-time embedded control applied to uavs. *In Proceedings of the 45th IEEE Conference on Decision and Control*, 2006.
- [45] G. Kitagawa. Monte carlo filter and smoother for non-gaussian nonlinear state space models. *Journal of Computational and Graphical Statistics*, 5(1):21–25, 1996.
- [46] G. Klein and D. Murray. Parallel tracking and mapping for small ar workspaces. *Proc. IEEE Intl. Symposium on Mixed and Augmented Reality (ISMAR)*, 2007.
- [47] A. Kushleyev, D. Mellinger, C. Powers, and V. Kumar. Towards a swarm of agile micro quadrotors. *Autonomous Robots*, 2013.
- [48] T. Lam, M. Mulder, and M. van Paasen. Haptic feedback for uav tele-operation - force offset and spring load modification. *IEEE International Conference on Systems, Man, and Cybernetics*, 2006.
- [49] D. Lee, A. Franchi, I. Hyung, H. ChangSu, H. Bulthoff, and P. Giordano. Semiautonomous haptic teleoperation control architecture of multiple unmanned aerial vehicles. *IEEE/ASME Transactions on Mechatronics*, 18(4):1334–1345, 2013.
- [50] Q. Lindsey, D. Mellinger, and V. Kumar. Construction of cubic structures with quadrotor teams. *Proceedings of Robotics Science and Systems (RSS)*, 2011.
- [51] D. Luenberger. An introduction to observers. *IEEE Transactions on Automatic Control*, AC-16(6), 1971.
- [52] S. Lupashin, A. Schölling, M. Sherback, and R. D'Andrea. A simple learning strategy for high-speed quadcopter multi-flips. *IEEE International Conference on Robotics and Automation (ICRA)*, 2010.
- [53] D. Mellinger, N. Michael, and V. Kumar. Trajectory generation and control for precise aggressive maneuvers with quadrotors. *The International Journal of Robotics Research*, 31(5):664–674, 2012.
- [54] D. Mercado, P. Castillo, R. Castro, and R. Lozano. 2-sliding mode trajectory tracking control and ekf estimation for quadrotors. *In the 19th IFAC World Conference*, 2014.

- [55] T. Mori and S. Scherer. First results in detecting and avoiding frontal obstacles from a monocular camera for micro aerial vehicles. *IEEE International Conference on Robotics and Automation (ICRA)*, 2013.
- [56] M. Mueller and R. D’Andrea. Stability and control of a quadcopter despite the complete loss of one, two, or three propellers. *IEEE International Conference on Robotics and Automation (ICRA)*, 2014.
- [57] L. E. Munoz, P. Castillo, and P. Garcia. Observer-control scheme for autonomous navigation: Flight tests validation in a quadrotor vehicle. *In International Conference on Unmanned Aircraft Systems (ICUAS)*, 2013.
- [58] M. Nieuwenhuisen, D. Droeschel, M. Beul, and S. Behnke. Obstacle detection and navigation planning for autonomous micro aerial vehicles. *Proceedings of International Conference on Unmanned Aircraft Systems (ICUAS)*, 2014.
- [59] Y. Orlov. Finite time stability and robust control synthesis of uncertain switched systems. *In SIAM Journal of Control Optimization* 43, pages 1253–1271, 2005.
- [60] I. Palunko, P. Cruz, and F. Fierro. Agile load transportation. *IEEE Robotics and Automation Magazine*, 19(3):69–79, 2012.
- [61] I. Palunko, F. Fierro, and P. Cruz. Trajectory generation for swing-free maneuvers of a quadrotor with suspended payload: A dynamic programming approach. *IEEE International Conference on Robotics and Automation (ICRA)*, 2012.
- [62] P. Pounds, D. Bersak, and A. Dollar. Stability of small-scale uav helicopters and quadrotors with added payload mass under pid control. *Auton Robot*, 33:129–142, 2012.
- [63] R. Ritz, M. Mueller, M. Hehn, and R. D’Andrea. Cooperative quadcopter ball throwing and catching. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.
- [64] L. Romo-Morales, A. Sanchez, V. Parra-Vega, O. Garcia, and F. Ruiz-Sanchez. Visual control for trajectory tracking of quadrotors and real-time analysis on an emulated environment. *American Control Conference (ACC)*, 2013.
- [65] J. Roskam. *Airplane Flight Dynamics and Automatic Flight Controls*. 1982.
- [66] S. Sadr, A. Moosavian, and P. Zarafshan. Dynamic modeling and control of a quadrotor with swing load. *Journal of Robotics*, 2014.
- [67] A. Sanna, F. Lamberti, G. Paravati, and F. Manuri. A kinect-based natural interface for quadrotor control. *Entertaining Computing*, 4(3):179–186, 2013.
- [68] D. Shevitz and B. Paden. Lyapunov stability theory of nonsmooth systems. *In IEEE Trans. Automat. Control*, 39:1910–1914, 1994.

- [69] Y. Shtessel, C. Edwards, L. Fridman, and A. Levant. *Sliding Mode Control and Observation*. Springer, New York, USA, 2010.
- [70] R. Siegwart and I. Nourbakhsh. *Introduction to Autonomous Mobile Robots*, chapter 6.2.1.3. The MIT press, 2004.
- [71] Steven W. Smith. *The Scientist and Engineer's Guide to Digital Signal Processing*. California Technical Publishing, second edition, 1997.
- [72] K. Sreenath, T. Lee, and V. Kumar. Geometric control and differential flatness of a quadrotor uav with a cable-suspended load. *IEEE Conference on Decision and Control (CDC)*, 2013.
- [73] K. Sreenath, N. Michel, and V. Kumar. Trajectory generation and control of a quadrotor with a cable-suspended load - a differentially-flat hybrid system. *IEEE International Conference on Robotics and Automation (ICRA)*, 2013.
- [74] A. Tayebi and S. McGilvray. Attitude stabilization of a vtol quadrotor aircraft. *IEEE Transactions on Control Systems Technology*, 14, 2006.
- [75] P. Viola and M. J. Jones. Rapid object detection using a boosted cascade of simple features. *IEEE CVPR*, 2001.
- [76] J. Wang, M. Garratt, A. Lambert, J. Jianguo, S. Han, and D. Sinclair. Integration of gps/ins/vision sensors to navigate unmanned aerial vehicles. Beijing, China, 2008.
- [77] S. Weiss, S. M. Achtelik, S. Lynen, M. Achtelik, L. Kneip, M. Chli, and R. Siegwart. Monocular vision for long-term micro aerial vehicle state estimation: A compendium. *Journal of Field Robotics*, 30(5), 2013.
- [78] S. Weiss, D. Scaramuzza, and R. Siegwart. Monocular-slam-based navigation for autonomous micro helicopters in gps-denied environments. *Journal of Field Robotics*, 28(6):854–874, 2011.
- [79] G. Wu and K. Sreenath. Geometric control of multiple quadrotors transporting a rigid-body load. *IEEE Conference on Decision and Control (CDC)*, 2014.
- [80] R. Xu and Ü. Özgüner. Sliding mode control of a quadrotor helicopter. *En Proc. 45th IEEE Conference on Decision & Control (CDC)*, 2006.