DESIGN AND CONTROL OF QUADROTORS WITH APPLICATION TO AUTONOMOUS FLYING

THÈSE N^{O} 3727 (2007)

PRÉSENTÉE LE 23 FÉVRIER 2007

À LA FACULTÉ DES SCIENCES ET TECHNIQUES DE L'INGÉNIEUR
Laboratoire de systèmes autonomes 1
SECTION DE MICROTECHNIQUE

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Samir BOUABDALLAH

ingénieur d'état, Université Aboubekr Belkaid, Tlemcen, Algérie
de nationalité algérienne

acceptée sur proposition du jury:

Prof. R. Clavel, président du jury
Prof. R. Siegwart, directeur de thèse
Prof. P. Corke, rapporteur
Dr Ph. Müllhaupt, rapporteur
Prof. J.-D. Nicoud, rapporteur

EPFL
ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Lausanne, EPFL
2007

Remerciements

Ce travail est le fruit de quelques années passées au laboratoire de systèmes autonomes de l'EPFL, dirigé par le Professeur Roland Siegwart que je remercie très profondément de m'avoir donné ma chance, et surtout de m'avoir guidé, soutenu et encouragé tout au long de la thèse. Roland, avec ses compétences, son tact et son sens des relations humaines, a fait de son groupe, non seulement une équipe soudée, mais une famille unie.

Mes remerciements vont également au Professeurs Peter Corke et Jean-Daniel Nicoud ainsi qu'au Docteur Philippe Müllhaupt pour leur participation à mon jury de thèse.

Lors de certains passages particuliers de votre vie, il ya des gens qui sont là, qui vous écoutent, vous soutiennent. Dr. Mohamed Bouri en fait partie, j'ai trouvé en lui un frère et un ami, toujours prêt à conseiller et à rendre service. Merci "*moh*" pour ce que tu es. En font partie aussi, Daniel Burnier et Marie-José Pellaud, leurs présence et leurs innombrables conseils ont rendu ma vie au labo, tout simplement plus humaine. Je leurs transmet toute ma reconnaissance.

Le campus de l'EPFL regorge de personnes très compétentes, toujours prêtes à donner de leur temps et partager leurs connaissances. C'est le cas du Dr. Philippe Müllhaupt, ses conseils ont jalonné ce doctorat du début à la fin. Je l'en remercie vivement.

Plusieurs étudiants ont participé à ce travail, André Noth, Thierry Frank, Julien Epiney, Vincent De Perrot, Yves Stauffer, Fabien Gigon, Fabien Tâche, Marc Tosetti, Florent Guenter, ainsi que d'autres qui, à travers leur travail ont eu un impact certain sur cette thèse. Merci à tous pour votre temps et vos efforts. Merci aussi à Marcelo Becker pour sa contribution dans la partie "évitement d'obstacles". Il ne faut pas oublier Peter Brühlmeier, André

Guignard, Tarek Baaboura, Philippe Vosseler ainsi que les mécaniciens du BM, pour leur support dans les diverses réalisations techniques.

Le dynamisme ambiant à l'ASL est dû à son excellente équipe. C'est ainsi que je tiens à remercier André Noth, Jean-Christophe Zufferey et Gilles Caprari pour les discussions passionnantes sur la robotique aérienne, Rolf Jordi, Frédéric Pont et Roland Philippsen pour leur support en Linux, Agostino Martinelli pour ses conseils en mathématique, Pierre Lamon, Davide Scaramuzza pour les discussions sur la vision, Kristian Maček pour les échanges sur le contrôle, Jan Weingarten pour... son humour et Massoud pour son "*Sabah el-kheiiir ya djari*", en passant chaque matin. Sans oublier Michel, Adriana et Marcelo, aussi Sandra pour avoir créé cette sympathique atmopshère au bureau et bien sûr tous les membres de l'ASL pour l'environnement motivant qu'ils entretiennent.

Enfin, à ma famille et belle famille je dis: Merci pour votre soutien et votre compréhension. Merci aussi à mes frères et soeur, surtout à toi mon grand frère Hmida pour m'avoir encouragé à frapper à la porte de l'EPFL et à toi maman de m'avoir donné tant d'amour et d'attention. Papa, saches, toi qui nous a quitté durant cette thèse, que les principes que tu as forgés en moi et l'éducation que tu m'a offerte ont été, le long de ce parcours doctoral, mon unique parure. Je te dédie cette thèse. Quant à ma plus belle réussite, elle restera toujours ma femme et mon fils auxquels je dis: Merci.

Samir Bouabdallah
Zürich, Dec. 2006

Abstract

This thesis is about modelling, design and control of Miniature Flying Robots (MFR) with a focus on Vertical Take-Off and Landing (VTOL) systems and specifically, micro quadrotors. It introduces a mathematical model for simulation and control of such systems. It then describes a design methodology for a miniature rotorcraft. The methodology is subsequently applied to design an autonomous quadrotor named OS4. Based on the mathematical model, linear and nonlinear control techniques are used to design and simulate various controllers along this work.

The dynamic model and the simulator evolved from a simple set of equations, valid only for hovering, to a complex mathematical model with more realistic aerodynamic coefficients and sensor and actuator models.

Two platforms were developed during this thesis. The first one is a quadrotor-like test-bench with off-board data processing and power supply. It was used to safely and easily test control strategies. The second one, OS4, is a highly integrated quadrotor with on-board data processing and power supply. It has all the necessary sensors for autonomous operation.

Five different controllers were developed. The first one, based on Lyapunov theory, was applied for attitude control. The second and the third controllers are based on PID and LQ techniques. These were compared for attitude control. The fourth and the fifth approaches use backstepping and sliding-mode concepts. They are applied to control attitude. Finally, backstepping is augmented with integral action and proposed as a single tool to design attitude, altitude and position controllers. This approach is validated through various flight experiments conducted on the OS4.

Key words: Quadrotor, Dynamic Modelling, Backstepping, MAV.

Résumé

Cette thèse traite de la modélisation, de la conception et du contrôle de robots volants miniatures. Le travail est focalisé sur les systèmes à vol vertical, plus précisément sur les quadrotors. La thèse introduit un modèle mathématique pour la simulation et le contrôle. Elle décrit une méthodologie de conception pour hélicoptères miniatures, qui est par la suite appliquée pour la réalisation d'un quadrotor autonome nommé OS4. Basé sur le modèle, des techniques linéaires et nonlinéaires ont été utilisées pour concevoir divers contrôleurs.

Le modèle dynamique et le simulateur ont évolué d'un simple set d'équations, seulement valide en vol stationnaire, à un modèle complexe comprenant des coefficients aérodynamiques plus réalistes ainsi que des modèles de capteurs et d'actuateurs.

Deux plateformes furent développées, la première est un banc de test de quadrotors. Ce banc a permis de tester facilement des stratégies de contrôle. La seconde, OS4, est un quadrotor autonome hautement intégré.

Cinq techniques de contrôle ont été utilisées le long de cette thèse. La première est basée sur la théorie de Lyapunov, elle a été appliquée au contrôle de l'assiette. La deuxième et la troisième techniques sont respectivement le PID et le LQ. Leurs performances en contrôle d'assiette ont été comparées. La quatrième et la cinquième approches utilisent les concepts de backstepping et de mode-glissant, appliqués au contrôle d'assiette. Finalement, la technique du backstepping est renforcée avec une action intégrale et proposée comme seul outil de contrôle. Cette approche est validée sur OS4 dans diverses expériences en vol.

Mots clés: Quadrotor, Modélisation Dynamique, Backstepping, MAV.

List of symbols

a	lift slope
A	propeller disk area
A_c	fuselage area
Au	Operational time (autonomy)
b	thrust factor
B_W	propulsion group bandwidth
c	propeller chord
C	propulsion group cost factor
C_{bat}	battery capacity
$\overline{C_d}$	drag coefficient at 70% radial station
C_H	hub force coefficient
C_Q	drag coefficient
C_{R_m}	rolling moment coefficient
C_T	thrust coefficient
d	drag factor
g	acceleration due to gravity
h	vertical distance: Propeller center to CoG
H	hub force
i	motor current
$I_{xx,yy,zz}$	inertia moments
J_m	motor inertia
J_r	rotor inertia
J_t	total rotor inertia seen by the motor
k_e	motor electrical constant
k_m	motor torque constant
l	horizontal distance: propeller center to CoG
m	overall mass
m_{af}	airframe mass
m_{av}	avionics mass
m_{bat}	battery mass

$m_{bat_{av}}$	avionics’ battery mass
m_{hel}	helicopter mass
m_{pg}	propulsion group mass
$M_{BAT_{max}}$	maximum battery mass possible
$M_{max_{possible}}$	maximum mass one motor can lift
$M_{max_{requested}}$	requested mass for one motor to lift
n	number of propellers
P_{av}	avionics’ power consumption
P_{el}	electrical power
P_{in}	gearbox input power
P_{out}	gearbox output power
Q	drag moment
Q_{pg}	propulsion group quality factor
Q_{in}	design quality index
r	gearbox reduction ratio
R	rotation matrix
R_{rad}	rotor radius
R_{mot}	motor internal resistance
R_m	rolling moment
T	thrust force
T_w	propulsion group thrust/weight ratio
u	motor input
U	control inputs
V	body linear speed
x, y, z	position in body coordinate frame
X, Y, Z	position in earth coordinate frame
β	thrust/weight ratio
ζ	position vector
η	gearbox efficiency
η_m	motor efficiency
θ	pitch angle
θ_0	pitch of incidence
θ_{tw}	twist pitch
λ	inflow ratio
μ	rotor advance ratio
ν	speed vector
ρ	air density
ϱ_{equ}	percentage of time in equilibrium
σ	solidity ratio
τ	motor time-constant

τ_a	torque in body coordinate frame
τ_d	motor load
τ_m	motor torque
υ	induced velocity
ϕ	roll angle
ψ	yaw angle
ω	body angular rate
ω_m	motor angular rate
Ω	propeller angular rate
Ω_r	overall residual propeller angular speed

Acronyms

ASL Autonomous Systems Laboratory

BLDC Brush-Less Direct Current

HTA Heavier Than Air

IB Integral Backstepping

IGE In Ground Effect

IMU Inertial Measurement Unit

LQ Linear Quadratic

LTA Lighter Than Air

MAV Micro Aerial Vehicle

MEMS Micro Electro-Mechanical Systems

MFR Miniature Flying Robots

OS4 Omnidirectional Stationary Flying OUtstretched Robot

PG Propulsion Group

PID Proportional Integral Derivative

PPM Pulse Position Modulation

UAV Unmanned Aerial Vehicle

VTOL Vertical Take-Off and Landing

Face à la roche, le ruisseau l'emporte toujours, non pas par la force mais par la persévérance.

Jackson Brown

Contents

Remerciements **i**

Abstract **iii**

Résumé **v**

List of symbols **vii**

Acronyms **ix**

1 Introduction **1**

1.1 Motivations and Objectives 1

1.2 State of the Art 2

1.3 Contribution of this Work 4

1.4 A Short Helicopter History 6

1.5 MAV Configurations 9

1.5.1 Helicopters vs Other Flying Principles 9

1.5.2 Short VTOL Configurations Comparison 11

1.5.3 Candidate VTOL Configurations for Future MAV . . 11

2 System Modelling **15**

2.1 Concept and Generalities 15

2.2 Modelling with Euler-Lagrange Formalism 16

2.2.1 Kinematics 17

2.2.2 Energy 17

2.2.3 Equation of Motion 17

2.2.4 The Derived Dynamic Model 18

2.2.5 Rotor Dynamics 18

2.3 Modelling with Newton-Euler Formalism 19

2.3.1 Aerodynamic Forces and Moments 19

2.3.2 General Moments and Forces 22
2.3.3 Equations of Motion 24
2.3.4 Rotor Dynamics 24
2.4 OS4 Simulator 25
2.5 Conclusion 27

3 System Design 29
3.1 Concept and Generalities 29
3.2 Test-Bench Design 29
3.3 Small-Scale Rotorcraft Design 30
3.3.1 The General Method 31
3.3.2 The Iterative Algorithm 32
3.4 OS4 Quadrotor Design 34
3.4.1 Design Results 36
3.4.2 Propulsion Group 39
3.4.3 Computer Module 39
3.4.4 Position Sensor 41
3.4.5 Obstacle Detection Setup 42
3.4.6 OS4 vs Other Quadrotors 42
3.5 Conclusion 44

4 System Control 45
4.1 Concept and Generalities 45
4.2 Modelling for Control 45
4.3 Control using Lyapunov Theory 47
4.3.1 Simulations 48
4.3.2 Experiments 49
4.4 Control using PID Technique 51
4.4.1 PD Controller Synthesis and Simulation 52
4.4.2 PID Controller on the Real System 52
4.5 Control using Optimal Control Theory 54
4.5.1 Adaptive Optimal Control 54
4.5.2 First LQ Controller Synthesis and Simulation 55
4.5.3 Second LQ Controller Synthesis and Simulation 56
4.5.4 LQ Controller on the Real System 56
4.6 Control using Backstepping Technique 57
4.6.1 Backstepping Controller Simulation 60
4.6.2 Backstepping Controller on the Real System 60
4.7 Control using Sliding-Mode Technique 63
4.7.1 Sliding-mode Controller Simulation 64
4.7.2 Sliding-mode Controller on the Real System 64

4.8 First Autonomous Flight 64
4.9 The Proposed Control Approach 68
4.10 Control using Integral Backstepping 69
4.10.1 Attitude Control 70
4.10.2 Altitude Control 72
4.10.3 Position Control 75
4.10.4 Obstacle Avoidance 79
4.11 Conclusion 83

5 General Conclusion 85
5.1 Review 85
5.1.1 Modelling 86
5.1.2 Design 86
5.1.3 Control 87
5.1.4 Originalities 87
5.2 Outlook 88

A Additional Control Plots 91
A.1 Investigations in Obstacle Avoidance 92

B Modelling 95
B.1 Rotation Matrix 95
B.2 Mathematical Derivation 97
B.2.1 Test-Bench's Actuator Model Derivation 103

C Sensor Data 107
C.1 Inertial Measurement Unit 107
C.2 Range Sensor 110
C.3 Position Sensor 110

D Implementation Details 115
D.1 Software Architecture 115
D.2 Implementation Requirements 115

E The OS4 helicopter 117
E.1 OS4's Parameters 118

Bibliography 119

Index 124

Curriculum Vitæ 127

List of Tables

1.1 Some quadrotor projects. 3
1.2 Common MAV requirements. 8
1.3 Common MAV configurations. 10
1.4 Flying principles comparison (1=Bad, 3=Good). 11
1.5 VTOL concepts comparison (1=Bad, 4=Very good). 12
1.6 Quadrotors' main advantages & drawbacks. 14

2.1 Main physical effects acting on a helicopter. 16

3.1 OS4 propulsion group design variables. 40
3.2 Models of the propulsion group components. 40
3.3 Some parameters of OS4 in comparison with other quadrotors. 43

4.1 Simulation parameters of Backstepping controller. 60
4.2 Experimental parameters of Backstepping controller. 61
4.3 Simulation parameters of Sliding-mode controller. 64
4.4 Experimental parameters of Sliding-mode controller. 65

E.1 OS4 parameters. 118

List of Figures

1.1 The two platforms developed in this thesis. 5

1.2 Spaceship of the prophet Ezekiel (left). A flying dreamer (right). 6

1.3 Helical Air Screw (left). *Gyroplane n:01* (right). 7

1.4 UCAR concept from Lockheed Martin. 8

1.5 General classification of aircraft. 9

1.6 CoaX helicopter, developed at EPFL (pict. Alain Herzog). . . 13

1.7 The quadrotor concept. The width of the arrows is proportional to the propellers' angular speed. 13

2.1 Test-bench's coordinate system. 16

2.2 OS4 coordinate system. 20

2.3 Simulation: Ground effect influence on the inflow velocity. . . 22

2.4 Rotor and model step response, measured at propeller's shaft. 25

2.5 OS4 simulator block diagram. 26

2.6 Control block in OS4 simulator. 26

3.1 OS4 test-bench. **(1)** RS232 to I^{2}C translator, **(2)** Motor modules, **(3)** Universal joint, **(4)** Micro IMU, **(5)** Propulsion group. 30

3.2 OS4 test-bench block diagram. 30

3.3 The design method flowchart. The user has to define a target mass, span and thrust/weight ratio of his system. 31

3.4 The iterative algorithm flowchart. 33

3.5 The maximum mass one motor can lift, and the requested one over the range of the battery mass for the initial design of OS4. 35

3.6 The four design indicators in the initial design of OS4. 36

3.7 OS4 block diagram. A DSP processor handles attitude and altitude control. Then, a miniature PC handles obstacles avoidance control and communication tasks. The robot communicates through a wifi interface and accepts standard remote control signals. 37

3.8 Mass and power distributions of the initial design of OS4. The battery mass represents almost one half of the total mass and the actuators sink 90% of the power. 37

3.9 Initial design of OS4. 38

3.10 Mass and power distributions of the last version of OS4. The battery mass represents one third of the total mass and the actuators sink 90% of the power. 38

3.11 The last version of OS4 (see details in Appendix: E). 38

3.12 OS4 propulsion group. The module is interfaced through I^{2}C bus and has a local PI speed controller. 39

3.13 The x-board based, 40 g and 56x71 mm computer module. . . 40

3.14 Position sensing setup on OS4. 41

3.15 Possible US sensors arrangement on OS4. Position (3) was adopted after various testing. 42

3.16 OS4 in comparison with other quadrotors. 43

4.1 Connection of rotations and translations subsystems. 47

4.2 Simulation: The system has to maintain attitude angles to zero in spite of the added noise. 49

4.3 Experiment: The controller has to maintain attitude angles to zero. In spite of the huge initial condition on the yaw, the system is rapidly brought back to equilibrium. 50

4.4 Simulation: The controller has to stabilize the orientation angles, starting from $\pi/4$ in roll, pitch and yaw as initial condition (P=0.8, D=0.4 for roll and pitch. P=0.8, D=0.5 for yaw angle). 52

4.5 Experiment: The controller has to stabilize the attitude. A higher priority was given to roll and pitch control. An integral term was added to eliminate the steady-state error (P=0.9, I=0.3, D=0.2 for roll and pitch. P=0.06, I=0.3, D=0.02 for yaw angle). A PID control loop was applied locally to control every propeller's speed. 53

4.6 Simulation: The system has to stabilize the orientation angles starting from $\pi/2$ with an LQ controller designed using Pearson method. 55

4.7 Simulation: The system has to stabilize the orientation angles starting from $\pi/2$ with an LQ controller generated using Sage-Eisenberg method. 57

4.8 Experiment: The system has to stabilize the orientation angles. The experiment was performed with an LQ controller using Sage-Eisenberg method. The control performance is weak, especially in roll and yaw angles. 58
4.9 Simulation: The backstepping controller has to stabilize the system and maintain the roll, pitch and yaw angles to zero. . 61
4.10 Experiment: The backstepping controller has to stabilize the system and maintain the roll, pitch and yaw angles to zero. The helicopter is stabilized very quickly despite the hard initial conditions. A slight drift in yaw angle is observed due to the vibrations and EMI influence on the yaw sensor. 62
4.11 Simulation: The sliding-mode controller has to stabilize the system and maintain the roll, pitch and yaw angles to zero. . 65
4.12 Experiment: The Sliding-mode controller has to stabilize the system and maintain attitude angles to zero. The controller stabilizes correctly roll and pitch angles, but the shattering effect is visible on them. However, yaw angle is weakly controlled. The big negative overshot in the pitch angle is due to the huge initial condition on the yaw angle and the roll. . . . 66
4.13 Experiment: First successful autonomous flight. The controller stabilizes the orientation angles with a PID controller. 67
4.14 The flying part of the test-bench in a tethered flight. 68
4.15 The proposed control approach. 69
4.16 The control structure implemented on OS4. 70
4.17 Simulation: Integral backstepping attitude controller has to maintain roll, pitch and yaw angles to zero. Despite of the hard initial conditions and the white noise, the helicopter is quickly brought back to equilibrium. 73
4.18 Experiment: Integral backstepping attitude controller has to maintain attitude angles to zero in flight. The helicopter is stabilized despite the numerous disturbances due to yaw drift, sensors noise and unmodeled effects. 74
4.19 OS4 in hover. A training frame was added for safety. 74
4.20 Autonomous landing flowchart. Altitude reference is gradually reduced taking into account the dynamics of the robot. 76
4.21 Autonomous take-off, altitude control and landing in simulation and in real flight. 76
4.22 Simulation: Integral backstepping position controller drives attitude controller in order to maintain the helicopter over a given point. 78

4.23 Four way-points for a square trajectory tracked by OS4. . . . 79
4.24 Simulation: The position and attitude signals generated to track the square trajectory. 80
4.25 The 4 flight directions (left) and the security zone (right). . . 81
4.26 Simulation: OS4 avoiding static obstacles. 82
4.27 Obstacle detection with and without the filter. 82
4.28 Experiment: Collision avoidance with OS4. The helicopter flies back until the obstacle disappears. 83

A.1 Simulation: Investigation of the Integral Backstepping attitude controller. Reference (green), Output (red). 91
A.2 Simulation: Investigation of the Integral Backstepping position controller. Reference (green), Output (red). 92
A.3 Experiment: Acceleration and angular rate data in flight. . . 93
A.4 The different obstacle avoidance maneuvers simulated. 94

B.1 Step-response of the DC motor used in the first platform, measured at motor shaft. The model output fits remarkably well the output measured in open-loop. 105

C.1 Roll and pitch angles behavior with the motors OFF and ON. 107
C.2 Accelerometers' and gyroscopes' responses of the MT9-B (green) and the 3DM-GX1 (red). 108
C.3 Filtered Z acceleration of the 3DM-GX1. 109
C.4 Yaw drift with motors turned-off and turned-on. 109
C.5 Range measurement with SRF10 and GP2Y0A02YK. 110
C.6 Beam volume with tubes (blue) and without tubes (orange). Side view (top left), top view (top right), 3D view (bottom). 111
C.7 Maximum range with different gains and 3 cm plastic tubes. . 111
C.8 Left: Image from a tiny CMOS camera (<1 g). Right: Image from our CCD camera (<16 g). Both images were taken at 1 m/s, our camera is almost motion-blur free. 112
C.9 The pattern before and after detection. 112
C.10 Translation at 1 m/s measured with our sensor and with an encoder. 113

E.1 Side view of the initial design of OS4. 117

E.2 Sensors, actuators and electronics of OS4. **(a)** inertial measurement unit, **(b)** altitude sensor below the robot, **(c)** obstacle avoidance sensor with tubes, **(d)** mini camera below the robot, **(e)** DSP, **(f)** mother board, **(g)** motor module, **(h)** propeller, **(i)** battery, **(j)** RC antenna, **(k)** wifi dongle. 118

Chapter 1

Introduction

A helicopter is a collection of vibrations held together by differential equations.

John Watkinson

1.1 Motivations and Objectives

Flying objects have always exerted a great fascination on man encouraging all kinds of research and development. This thesis started in 2003, a time at which the robotics community was showing a growing interest in Micro Aerial Vehicle (MAV) development. The scientific challenge in MAV design and control in cluttered environments and the lack of existing solutions was very motivating. On the other hand, the broad field of applications in both military and civilian markets was encouraging the funding of MAV related projects. At the same time, the Autonomous Systems Laboratory (ASL) had already accumulated a large experience on ground-based robots with excellent results. Several theses were conducted on localization, navigation, obstacle avoidance etc. The limitations of ground-based robots in rough terrain and the recent progress in micro technology pushed us towards developing new mobility concepts. This includes flying systems on which one could apply the techniques already developed on ground-based robots. However, the task is not trivial due to several open challenges. In the field of sensing technologies, industry can currently provide a new generation of integrated micro Inertial Measurement Unit (IMU) composed generally of Micro Electro-Mechanical Systems (MEMS) technology inertial and magneto-resistive sensors. The latest technology in high density power storage offers about 190 Wh/kg which is a real jump ahead especially for micro aerial robotics. This technology was originally developed for hand-held applications and is now widely used in aerial robotics. The cost and size reduction of such systems makes it very interesting for the civilian market. Simultaneously, this reduction of cost and size implies performance limitation and thus a more challenging control. Moreover, the miniaturization of inertial sensors imposes the use of MEMS technology, which is still much less accurate than the conventional sensors because of noise and drift. The use of low-cost IMUs is synonym of less efficient data processing and thus a bad orientation data prediction in addition to a weak drift rejection. On the other hand, and in spite of the latest progress in miniature actuators, the scaling laws are still unfavorable and one has to face the problem of actuator saturation. That is to say, even though the design of micro aerial robots is possible, the control is still a challenging goal. It was decided from the beginning of this thesis to work on a particular VTOL configuration: the quadrotor. The interest comes not only from its dynamics, which represent an attractive control problem, but also from the design issue. Integrating the sensors, actuators and intelligence into a lightweight vertically flying system with a decent operation time is not trivial.

1.2 State of the Art

The state of the art in quadrotor control has drastically changed in the last few years. The number of projects tackling this problem has considerably and suddenly increased. Most of these projects are based on commercially available toys like the Draganflyer [1], modified afterwards to have more sensory and communication capabilities. Only few groups have tackled the MFR design problem, and even fewer did it in the optimal way (simultaneous consideration of design and control) for a quadrotor. Table 1.1 lists probably the most important quadrotor projects of the last 10 years. Mesicopter project, started in 1999 and ended in 2001. It aimed to study the feasibility of a centimeter scale quadrotor. The project's driving application was the deployment over large areas or planets of a huge number of micro vehicles providing atmospheric and meteorological data. Starmac, another interesting project, it targets the demonstration of multi agent control of quadrotors of about 1 kg. However, none of these systems was built based on a clear and systematic design optimization methodology.

Table 1.1: Some quadrotor projects.

Project	University	Status	Picture
Mesicopter	Stanford	Ended	
E. Altuğ's thesis	Univ. Pennsylvania	Ended	
P. Castillo's thesis	Univ. Compiègne	Ended	
A. Clifton's thesis	Univ. Vanderbilt	Ended	
P. Pounds's thesis	ANU	in progress	
N. Guenard's thesis	CEA	in progress	
Starmac	Stanford	in progress	
M. Kemper's thesis	Univ. Oldenburg	in progress	
P. Tournier's thesis	MIT	in progress	
MD4-200®	microDrones GmbH	in progress	

1.3 Contribution of this Work

This thesis focuses on design and control of unmanned, autonomous micro helicopters with application to a quadrotor helicopter. The contribution of this work lies in three fields.

- Dynamic *modelling* of quadrotors: the goal is to obtain a faithful mathematical representation of the mechanical system for system analysis and control design.
- System *design* and optimization: the objective is to maximize the operation time and minimize the weight of the helicopter.
- System *control*: the aim is to understand and then master the dynamics of quadrotors by applying the appropriate control techniques.

Modelling

An accurate simulation model was developed in successive steps. The first version was a simple model describing the vehicle in hover flight, see paper [2]. It included only the gyroscopic effects and the action of the actuator. A faithful CAD model allowed the easy extraction of the physical parameters. The model then evolved into a more complete set of equations, describing the vehicle dynamics not only in hover, but also in motion (see Section 2.3). This was achieved through the introduction of several effects like friction force, hub force and propeller rolling moment. The implementation of a simulator which includes an aerodynamics block allowed the consideration of variable aerodynamic coefficients, namely: the thrust, drag, hub force and rolling moment factors. Finally, the propulsion group model was identified, simplified and validated. The entire model was implemented under Simulink and used to optimize the design and to tune the control parameters. Right from the beginning, the control parameters tuned in simulation were implemented on the Omnidirectional Stationary Flying OUtstretched Robot (OS4) for successful flight stabilization. The evolution of the model is perceptible through the successive papers issued by the author.

Design

Designing a miniature autonomous helicopter is basically dealing with numerous design parameters that are closely linked. Taking a decision about all these parameters requires a clear methodology. This thesis proposes a practical method to handle the design problematic of a small-scale rotorcraft by combining the theoretical knowledge of the system and the result of a system level optimization analysis. The method is driven by the application. The user defines a target size and weight for the system and the algorithm selects (from a database) the best components to be used and estimates iteratively the most important design parameters. The method was used to design the OS4 and CoaX flying robots. A description of the method can be found in Chapter 3. Two quadrotor platforms were designed, the first one was mainly a test-bench (see Fig. 3.1). The second quadrotor design called OS4 (see Fig. 3.11) is a 650 g quadrotor equipped with all the necessary sensors and actuators for fully autonomous operation. It is capable of a maximum 20 min operation time and has an almost 53% thrust margin. OS4 is undoubtedly one of the most integrated small-scale quadrotors ever designed.

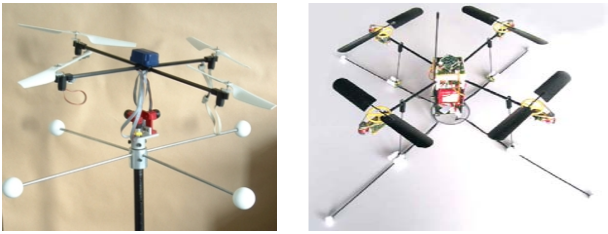


Figure 1.1: The two platforms developed in this thesis.

Control

An important part of this thesis was dedicated to finding a good control approach for quadrotors. Several techniques were explored from theoretical development to final experiments. Firstly, Lyapunov theory was applied to the attitude control of the helicopter on its test-bench only. Secondly, two linear controllers, a Proportional Integral Derivative (PID) and a Linear Quadratic (LQ), were investigated based on a simplified model. The main result was an autonomous hover flight, see paper [3]. In the third attempt, backstepping and sliding-mode techniques were tested. This time, we were able to elegantly reject strong disturbances, but the stabilization in hover flight was delicate, this is detailed in paper [4]. Another improvement was then introduced due to integral backstepping [5]. By means of this technique, OS4 is able to perform autonomous hovering with altitude control and autonomous take-off and landing. Obstacle avoidance on MAV remains an open problem and is in fact beyond the focus of this thesis. However, several approaches were developed and simulated in an attempt to perform obstacle avoidance on a quadrotor based on four ultra-sound sensors. Despite the difficulties due to ultra-sound interference, sensor noise, and vehicle dynamics, the experiment showed that OS4 is now able to escape an approaching obstacle.

1.4 A Short Helicopter History

Mankind's fascination for flying has been clearly expressed throughout man's history. It took the form of legends, myths or even religious accounts as the spaceship vision in the book of prophet Ezekiel or the intriguing Vimanas in ancient Indian mythology, or even the feat of Daedalus and Icar. The dream of flying, one of the biggest challenges for humans, has generated centuries of frustration and hundreds of dramatic attempts without being faded [6]. However, by inventing the wheel, man made it better than nature which knows only the alternative motion of wings. The helicopter's history is short in comparison with fixed-wing aircraft. Chinese tops are probably the first flying toys ever designed, they were inspired by observations of the auto-rotating seeds of trees such as the Sycamore. In 1490, Leonardo Da Vinci created the Helical Air Screw (see Fig. 1.3), it has been often cited as the first serious attempt to produce a working helicopter. Ponton d'Amécourt was the first one to use the word "Helicopter" (from two old Greek words: Helix and Pteron, screw and wing) in 1863. He also described a coaxial helicopter and several ways to steer it. On this basis, in 1877 Forlanini realized a reduced scale, steam powered model able to fly 20 seconds at 12 meters. The first electrical model was built in 1887. Many years later on 29 September 1907, the Frenchmen Louis and Jacques Breguet and the professor Richet have demonstrated the first manned flight with their *Gyroplane n:01* (see Fig. 1.3), a huge quadrotor with a double layer of propellers and no control surfaces. The first manually controlled free flight was achieved on 13 November 1907 by Paul Cornu on his tandem-like helicopter [7]. The helicopter was born.

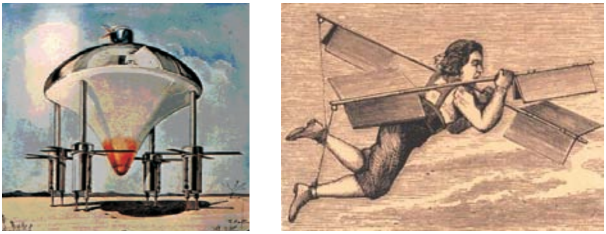


Figure 1.2: Spaceship of the prophet Ezekiel (left). A flying dreamer (right).

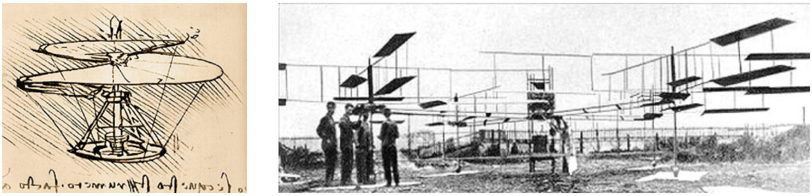


Figure 1.3: Helical Air Screw (left). *Gyroplane n:01* (right).

The Era of UAVs

A few years after the first manned airplane flight, Dr. Cooper and Elmer Sperry invented the automatic gyroscopic stabilizer, which helps to keep an aircraft flying straight and level. This technology was used to convert a U.S. Navy Curtiss N-9 trainer aircraft into the first radio-controlled Unmanned Aerial Vehicle (UAV). The first UAVs were tested in the US during World War I but never deployed in combat. During World War II, Germany took a serious advantage and demonstrated the potential of UAVs on the battlefields. After the two wars, the military recognized the potential of UAVs in combat and started development programs which led, few decades after, to sophisticated systems, especially in the US and Israel, like the Predator (General Atomics) or the Pioneer (PUAV). The first unmanned helicopter was the one built by Forlanini in 1877. It was neither actively stabilized nor steerable. With the outstanding technological advancements after WW II it became possible to build and control unmanned helicopters. The company Gyrodyne of America started the famous DASH program for the navy. The military market of unmanned helicopters became evident. An intensive research effort was deployed and impressive results achieved; like the A160 Hummingbird, a long-endurance helicopter able to fly 24 h within a range of 3150 km. The battlefield of the future would belong to the Unmanned Combat Armed Rotorcraft (see Fig. 1.4), if DARPA decides one day to restart this ambitious project. The academic researchers have also shown their interest in the development of autonomous helicopters over the last decades. An extensive research effort is being conducted on VTOL UAVs and MAVs. This research is not only directed towards civilian applications like search and rescue, but also towards military ones. A good survey of helicopter projects during the 90s is proposed in [8]. Presently, an important effort is invested in autonomous MAVs, where the challenges of the miniaturization, autonomy, control, aerodynamics and sources of energy are tackled. The WASP from AeroVironment [9] is considered as the state of the art in fixed-wing MAVs.

Figure 1.4: UCAR concept from Lockheed Martin.

On the other hand, the state of the art in rotary-wing is not clearly defined but the MFR II from Epson [10] seems to be the most integrated helicopter in this size. The future of MAVs is facing numerous technological challenges often identified as:

- Low Reynolds Number Aerodynamics
- Analytical and computational models
- Lightweight multi-functional materials and structures
- Robust flight navigation and control
- Miniaturized navigation and control electronics.

A short set of requirements was defined in [11] as guidelines for a typical MAV urban mission in the future. Comparing Table 1.2 to the state of the art reveals that the road to mission-level MAVs is still long. This will undoubtedly require several years of intensive research.

Table 1.2: Common MAV requirements.

Specification	Requirement
Size	<15 cm
Weight	<100 g
Range	1 to 10 km
Endurance	60 min
Altitude	<150 m
Speed	15 m/s
Payload	20 g
Cost	$ 1500

1.5 MAV Configurations

In general, aerial vehicles can be divided into two categories: Lighter Than Air (LTA) and Heavier Than Air (HTA). Figure 1.5 presents a general classification of aircraft depending on the flying principle and the propulsion mode. Table 1.4 gives a non-exhaustive comparison between different flying principles from the miniaturization point of view. One can easily conclude from this table that VTOL systems like helicopters or blimps have an unquestionable advantage compared with the other concepts. This superiority is owed to their unique ability for vertical, stationary and low speed flight. The key advantage of blimps is the auto-lift and the simplicity of control which can be essential for critical applications, such as aerial surveillance. Table 1.3 lists different configurations commonly used in MAV research and industry.

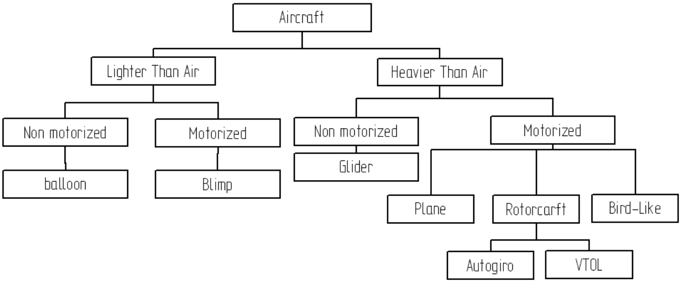


Figure 1.5: General classification of aircraft.

1.5.1 Helicopters vs Other Flying Principles

Compared with the other flying principles discussed above, VTOL systems have specific characteristics which allow the execution of applications that would be difficult or impossible with other concepts. Table 1.4 gives a non-exhaustive comparison between different flying principles from the MAV point of view. From this table, one can easily conclude that VTOL systems like helicopters or blimps have an advantage compared with other concepts in the MAV class. This superiority is owed to their unique ability for vertical, stationary and low speed flight. The key advantage of blimps is the "auto-lift" and the simplicity of control which can be essential for critical applications such as space exploration [12]. However, VTOL vehicles in different configurations represent today one of the most promising flying concepts seen in terms of miniaturization.

Table 1.3: Common MAV configurations.

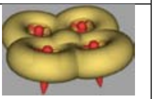
Configuration *e.g.*	Advantages	Drawbacks	Picture
Fixed-wing (AeroVironment)	Simple mechanics, silent operation	No hovering	
Single rotor (A. V de Rostyne)	Good controllability, good maneuverability	Complex mechanics, large rotor, long tail boom	
Axial rotor (Maryland Univ.)	Simple mechanics, compactness	Complex aerodynamics	
Coaxial rotors (EPSON)	Simple mechanics, compactness	Complex aerodynamics	
Tandem rotors (Heudiasyc)	Good controllability, simple aerodynamics	Complex mechanics, large size	
Quadrotor (EPFL-ETHZ)	Good maneuverability, simple mechanics, increased payload	High energy consumption, large size	
Blimp (EPFL)	Low power, long flight operation, auto-lift	Large size, weak maneuverability	
Hybrid quadrotor-blimp (MIT)	Good maneuverability, good survivability	Large size, weak maneuverability	
Bird-like (Caltech)	Good maneuverability, compactness	Complex mechanics, complex control	
Insect-like (UC Berkeley)	Good maneuverability, compactness	Complex mechanics, complex control	
Fish-like (US Naval Lab)	Multi-mode mobility, efficient aerodynamics	Complex control, weak maneuverability	

Table 1.4: Flying principles comparison (1=Bad, 3=Good).

	Airplane	Helicopter	Bird	Autogiro	Blimp
Power cost	2	1	2	2	3
Control cost	2	1	1	2	3
Payload/volume	3	2	2	2	1
Maneuverability	2	3	3	2	1
Stationary flight	1	3	2	1	3
Low speed fly	1	3	2	2	3
Vulnerability	2	2	3	2	2
VTOL	1	3	2	1	3
Endurance	2	1	2	1	3
Miniaturization	2	3	3	2	1
Indoor usage	1	3	2	1	2
Total	19	25	24	18	25

1.5.2 Short VTOL Configurations Comparison

Table 1.5 gives a short and not exhaustive comparison between different VTOL configurations. This is an adaptation of the larger comparison in [13]. One can see in this table that the quadrotor and the coaxial helicopter are among the best configurations if used as MFR.

1.5.3 Candidate VTOL Configurations for Future MAV

The different configurations listed in Table 1.3 are not all suitable candidates for future VTOL MAV systems. The coaxial and the quadrotor are the most promising ones. Moreover, a number of variants for these configurations appeared recently, like the rotary wing MAV with an active structure [14].

Coaxial configuration

The development of full-scale coaxial helicopters was historically slower than the one of single rotor. This is mainly due to the incredible complexity of their swashplate mechanisms. However, the advantage of coaxials was recognized for unmanned vehicles and on naval vessels, where space is limited [15]. In the coaxial configuration, one propeller is located above the other with a common shaft. The rotors turn in opposite directions, which removes the need for a tail rotor, and makes the helicopter a lot more compact. Typical coaxial MAVs

Table 1.5: VTOL concepts comparison (1=Bad, 4=Very good).

	A	B	C	D	E	F	G	H
Power cost	2	2	2	2	1	4	3	3
Control cost	1	1	4	2	3	3	2	1
Payload/volume	2	2	4	3	3	1	2	1
Maneuverability	4	3	2	2	3	1	3	3
Mechanics simplicity	1	2	3	1	4	4	1	1
Aerodynamics complexity	1	1	1	1	4	3	1	1
Low speed flight	4	3	4	3	4	4	2	2
High speed flight	2	4	1	2	3	1	3	3
Miniaturization	2	3	4	2	3	1	2	4
Survivability	1	3	3	1	1	3	2	3
Stationary flight	4	4	4	4	4	3	1	2
Total	24	28	32	23	33	28	22	24

[1] A=Single rotor, B=Axial rotor, C=Coaxial rotors, D=Tandem rotors, E=Quadrotor, F=Blimp, G=Bird-like, H=Insect-like.

use the residual torque, due to angular speed difference between the two rotors to rotate the helicopter vertically, left or right. Increasing or decreasing the angular speed of the rotors simultaneously permits climbing and descending. Finally, by using simplified swashplates or by shifting the center of gravity [16], it is possible to control rotation about the longitudinal and the lateral axis and thus control horizontal motion (see Fig. 1.6). The coaxial in hover behaves like a single rotor with the same total solidity, if the two rotors are not too far apart. However, if the separation between the upper and lower rotor is significant, the lower rotor will encounter increased inflow velocity and will require more power. Coaxial configuration fits remarkably well the requirement for MAVs. However, this is conditioned by the abandoning of complex swashplate mechanisms and the availability of rigid and efficient propellers.

Quadrotor configuration

The development of full-scale quadrotors experienced limited interest in the past. Nevertheless, the first manned short flight in 1907 was on a quadrotor, as shown before in Fig. 1.3. Nowadays, their development is limited to the UAV/MAV category. Present quadrotors have four fixed-pitch propellers in cross configuration. Driving the two pairs of propellers in opposite directions removes the need for a tail rotor. Consequently, vertical rotation is achieved by creating an angular speed difference between the two pairs of

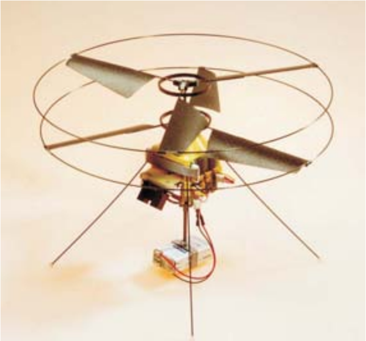


Figure 1.6: CoaX helicopter, developed at EPFL (pict. Alain Herzog).

rotors. Increasing or decreasing the speed of the four propellers simultaneously permits climbing and descending. Rotation about the longitudinal and the lateral axis and consequently horizontal motions are achieved by tilting the vehicle. This is possible by conversely changing the propeller speed of one pair of rotors as described in Fig. 1.7. In spite of the four actuators, the quadrotor remains an under-actuated and dynamically unstable system. In fact, MAV class quadrotors require a very small rotor diameter which is very penalizing in terms of efficiency. However, the inherent simple mechanics of quadrotors and the increased payload are their main advantage in the MAV class.

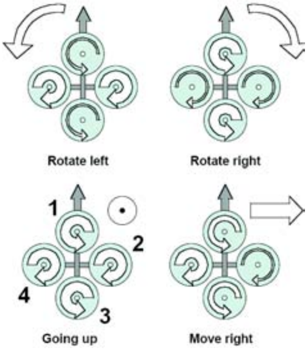


Figure 1.7: The quadrotor concept. The width of the arrows is proportional to the propellers' angular speed.

Advantages and Drawbacks

Size and energy requirements are definitely the main disadvantages of the quadrotor. However, this concept offers better payload and is simpler to build and control, which is a decisive advantage. Table 1.6 gives a rapid idea about quadrotors' advantages and drawbacks.

Table 1.6: Quadrotors' main advantages & drawbacks.

Advantages	Drawbacks
Simple mechanics	Large size and mass
High payload	High energy consumption
Reduced gyroscopic effects	

Chapter 2

System Modelling

2.1 Concept and Generalities

The concept followed in this thesis for dynamic modelling was to write physical equations, get the parameters from the CAD model and identify only the dynamics of the actuators which are important in the case of a quadrotor. This approach makes it easy to build dynamic models of instable systems, since we don't have to perform closed loop identification in flight. Euler-Lagrange formalism and DC motor equations were used to model the test-bench. Newton-Euler formalism, model identification and blade element and momentum theories were used to model the OS4 quadrotor. Tait-Bryan angles were used for the parametrization. The latter model is implemented as a simulator. The model developed in this thesis assumes the following:

- The structure is supposed rigid.
- The structure is supposed symmetrical.
- The CoG and the body fixed frame origin are assumed to coincide.
- The propellers are supposed rigid.
- Thrust and drag are proportional to the square of propeller's speed.

The helicopter is a complex mechanical system, it collects numerous physical effects from the aerodynamics and the mechanics domains [17]. The model of the quadrotor should consider all the important effects including the gyroscopic ones. A short list of the main effects acting on a helicopter [18] are described briefly in Table 2.1.

Table 2.1: Main physical effects acting on a helicopter.

Effect	Source	Formulation
Aerodynamic effects	Propeller rotation Blades flapping	$C\Omega^2$
Inertial counter torques	Change in propeller rotation speed	$J\dot{\Omega}$
Gravity effect	Center of mass position	
Gyroscopic effects	Change in orientation of the rigid body	$I\theta\psi$
	Change in orientation of the propeller plane	$J\Omega_r\theta, \phi$
Friction	All helicopter motion	$C\dot{\phi}, \dot{\theta}, \dot{\psi}$

2.2 Modelling with Euler-Lagrange Formalism

The rotation dynamics of the test-bench are modelled in this section using Euler-Lagrange Formalism. The detailed mathematical derivation is provided in Appendix B.2. Let us consider earth fixed frame E and body fixed frame B, as seen in Fig. 2.1. The airframe orientation in space is given by a rotation R from B to E, where $R \in SO3$ is the rotation matrix.

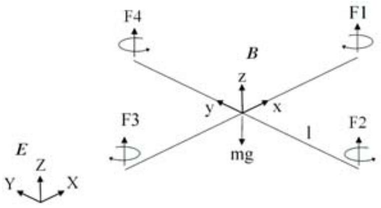


Figure 2.1: Test-bench's coordinate system.

2.2.1 Kinematics

For any point of the airframe expressed in the earth fixed frame, we can write (with, c:cos, s:sin).

$$\begin{cases} r_X = (c\psi c\theta)x + (c\psi s\theta s\phi - s\psi c\phi)y + (c\psi s\theta c\phi + s\psi s\phi)z \\ r_Y = (s\psi c\theta)x + (s\psi s\theta s\phi + c\psi c\phi)y + (s\psi s\theta c\phi - c\psi s\phi)z \\ r_Z = (-s\theta)x + (c\theta s\phi)y + (c\theta c\phi)z \end{cases} \tag{2.1}$$

The corresponding velocities are obtained by differentiation of (2.1), and the squared magnitude of the velocity for any point is given by:

$$\nu^2 = \nu_X^2 + \nu_Y^2 + \nu_Z^2 \tag{2.2}$$

2.2.2 Energy

From the equation (2.2), and by assuming that the inertia matrix is diagonal, one can extract the kinetic energy expression (see Appendix: B for details):

$$T = \frac{1}{2}I_{xx}(\dot{\phi} - \dot{\psi}s\theta)^2 + \frac{1}{2}I_{yy}(\dot{\theta}c\phi + \dot{\psi}s\phi c\theta)^2 + \frac{1}{2}I_{zz}(\dot{\theta}s\phi - \dot{\psi}c\phi) \tag{2.3}$$

Using the well known potential energy formula, one can express (2.3) in the earth fixed frame as:

$$V = \int xdm(x)(-gs\theta) + \int ydm(y)(gs\phi c\theta) + \int zdm(z)(gc\phi c\theta) \tag{2.4}$$

2.2.3 Equation of Motion

Using the Lagrangian and the derived formula for the equations of motion:

$$L = T - V \quad , \quad \Gamma_i = \frac{d}{dt}(\frac{\partial L}{\partial \dot{q}_i}) - \frac{\partial L}{\partial q_i} \tag{2.5}$$

where $\dot{q}_i$ are the generalized coordinates and Γ_i the generalized forces. The three equations of motion are then:

$$\begin{cases} I_{xx}\ddot{\phi} = \dot{\theta}\dot{\psi}(I_{yy} - I_{zz}) \\ I_{yy}\ddot{\theta} = \dot{\phi}\dot{\psi}(I_{zz} - I_{xx}) \\ I_{zz}\ddot{\psi} = \dot{\phi}\dot{\theta}(I_{xx} - I_{yy}) \end{cases} \tag{2.6}$$

On the other hand, the nonconservative torques acting on OS4 result, firstly from the action of the thrust difference of each pair, see Fig. 2.1:

$$\begin{cases} \tau_x = bl(\Omega_4^2 - \Omega_2^2) \\ \\ \tau_y = bl(\Omega_3^2 - \Omega_1^2) \\ \\ \tau_z = d(\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2) \end{cases} \tag{2.7}$$

Secondly, from the gyroscopic effect resulting from the propellers rotation:

$$\begin{cases} \tau_x^{'} = J_r\omega_y(\Omega_1 + \Omega_3 - \Omega_2 - \Omega_4) \\ \\ \tau_y^{'} = J_r\omega_x(\Omega_2 + \Omega_4 - \Omega_1 - \Omega_3) \end{cases} \tag{2.8}$$

2.2.4 The Derived Dynamic Model

The quadrotor dynamic model describing the roll, pitch and yaw rotations contains then, three terms which are the gyroscopic effect resulting from the rigid body rotation, the gyroscopic effect resulting from the propeller rotation coupled with the body rotation and finally the actuators action:

$$\begin{cases} I_{xx}\ddot{\phi} = \dot{\theta}\dot{\psi}(I_{yy} - I_{zz}) - J\dot{\theta}\Omega_r + \tau_x \\ \\ I_{yy}\ddot{\theta} = \dot{\phi}\dot{\psi}(I_{zz} - I_{xx}) + J\dot{\phi}\Omega_r + \tau_y \\ \\ I_{zz}\ddot{\psi} = \dot{\phi}\dot{\theta}(I_{xx} - I_{yy}) + \tau_z \end{cases} \tag{2.9}$$

2.2.5 Rotor Dynamics

The test-bench's rotors are driven by DC motors with the well known equations:

$$\begin{cases} L\frac{di}{dt} = u - R_{mot}i - k_e\omega_m \\ \\ J_m\frac{d\omega_m}{dt} = \tau_m - \tau_d \end{cases} \tag{2.10}$$

As we use a small motor with a very low inductance, the second order DC motor dynamics may be approximated by:

$$J_m\frac{d\omega_m}{dt} \quad = \quad -\frac{k_m^2}{R_{mot}}\omega_m - \tau_d + \frac{k_m}{R_{mot}}u \tag{2.11}$$

By introducing the propeller and the gearbox models, the equation (2.11) may be rewritten:

$$\begin{cases} \dot{\omega}_m = -\frac{1}{\tau}\omega_m - \frac{d}{\eta r^3 J_t}\omega_m^2 + \frac{1}{k_m \tau}u \\ \\ \frac{1}{\tau} = \frac{k_m^2}{R J_t} \end{cases} \tag{2.12}$$

The equation (2.12) can be linearized around an operation point $\dot{w}_0$ to the form $\dot{w}_m = -Aw_m + Bu + C$ with:

$$A = \left(\frac{1}{\tau} + \frac{2dw_0}{\eta r^3 J_t}\right), \quad B = \left(\frac{1}{k_m \tau}\right), \quad C = \left(\frac{d\omega_0^2}{\eta r^3 J_t}\right) \tag{2.13}$$

2.3 Modelling with Newton-Euler Formalism

This simulation model was developed through some successive steps as presented in papers [2, 4, 19]. This last version includes hub forces H, rolling moments R_m and variable aerodynamical coefficients. This makes the model more realistic particularly in forward flight. With the preliminary versions of the model it was often necessary to slightly adjust the control parameters for successful experiments. This section presents the model used for the last version of the OS4 simulator with which the Integral Backstepping (IB) controller was developed. This time, the simulated control parameters were directly used on the real helicopter for successful autonomous flights. The dynamics of a rigid body under external forces applied to the center of mass and expressed in the body fixed frame are in Newton-Euler formalism [20]:

$$\begin{bmatrix} mI_{3\times 3} & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \dot{V} \\ \dot{\omega} \end{bmatrix} + \begin{bmatrix} \omega \times mV \\ \omega \times I\omega \end{bmatrix} = \begin{bmatrix} F \\ \tau \end{bmatrix} \tag{2.14}$$

Let us consider an earth-fixed frame E and a body-fixed frame B as seen in Fig. 2.2. Using Euler angles parametrization, the airframe orientation in space is given by a rotation R from B to E, where $R \in SO3$ is the rotation matrix. The frame system (Fig. 2.2) is in conformity with the N, E, D (North, East, Down) standard, following by the way the coordinate system of our inertial sensor (3DM-GX1).

2.3.1 Aerodynamic Forces and Moments

The aerodynamic forces and moments are derived using a combination of momentum and blade element theory [21]. This is based on the work of Gary

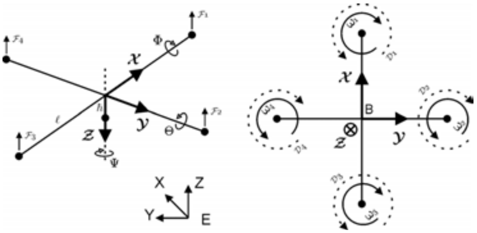


Figure 2.2: OS4 coordinate system.

Fay during Mesicopter project [22]. For an easier reading of the equations below, we recall some symbols.

σ : solidity ratio	λ : inflow ratio
a : lift slope	υ : induced velocity
μ : rotor advance ratio	ρ : air density

Thrust Force

Is the resultant of the vertical forces acting on all the blade elements.

$$\begin{cases} T = C_T \rho A (\Omega R_{rad})^2 \\ \frac{C_T}{\sigma a} = (\frac{1}{6} + \frac{1}{4}\mu^2)\theta_0 - (1+\mu^2)\frac{\theta_{tw}}{8} - \frac{1}{4}\lambda \end{cases} \tag{2.15}$$

Hub Force

Is the resultant of the horizontal forces acting on all the blade elements.

$$\begin{cases} H = C_H \rho A (\Omega R_{rad})^2 \\ \frac{C_H}{\sigma a} = \frac{1}{4a}\mu \overline{C_d} + \frac{1}{4}\lambda\mu(\theta_0 - \frac{\theta_{tw}}{2}) \end{cases} \tag{2.16}$$

Drag Moment

This moment about the rotor shaft is caused by the aerodynamic forces acting on the blade elements. The horizontal forces acting on the rotor are multiplied by the moment arm and integrated over the rotor. Drag moment determines the power required to spin the rotor.

$$\begin{cases} Q = C_Q \rho A (\Omega R_{rad})^2 R_{rad} \\ \frac{C_Q}{\sigma a} = \frac{1}{8a}(1+\mu^2)\overline{C_d} + \lambda(\frac{1}{6}\theta_0 - \frac{1}{8}\theta_{tw} - \frac{1}{4}\lambda) \end{cases} \tag{2.17}$$

Rolling Moment

The rolling moment of a propeller exists in forward flight when the advancing blade is producing more lift than the retreating one. It is the integration over the entire rotor of the lift of each section acting at a given radius. This is not to be confused with propeller radius or the rotation matrix R or the overall rolling moment which is caused by a number of other effects.

$$\begin{cases} R_m = C_{R_m} \rho A (\Omega R_{rad})^2 R_{rad} \\ \frac{C_{R_m}}{\sigma a} = -\mu(\frac{1}{6}\theta_0 - \frac{1}{8}\theta_{tw} - \frac{1}{8}\lambda) \end{cases} \tag{2.18}$$

Ground Effect

Helicopters operating near the ground (∼ at half rotor diameter) experience thrust augmentation due to better rotor efficiency. This is related to a reduction of the induced airflow velocity. This is called Ground Effect. In the literature one can find different approaches to deal with this effect, for instance by using adaptive techniques [23]. However, the principal need in this project is to find a model of this effect for OS4 to improve the autonomous take-off and landing controllers. The goal is to obtain a simple model capturing mainly the variation of the induced inflow velocity. Cheeseman [24] states (based on the images method [25]) that at constant power $T_{OGE} v_{i,OGE} = T_{IGE} v_{i,IGE}$. The velocity induced at the rotor center by its image is $\delta v_i = A v_i / 16\pi z^2$. Cheeseman obtained the relation (2.19) by assuming that v_i and δv_i are constant over the disk which allows $v_{i,IGE} = v_i - \delta v_i$.

$$\frac{T_{IGE}}{T_{OGE}} = \frac{1}{1 - \frac{R_{rad}^2}{16z^2}} \tag{2.19}$$

Another simple way to proceed is to consider that the inflow ratio In Ground Effect (IGE) is $\lambda_{IGE} = (v_{i,OGE} - \delta v_i - \dot{z})/\Omega R_{rad}$, where the variation of the induced velocity is $\delta v_i = v_i/(4z/R_{rad})^2$. We can then rewrite the thrust coefficient (2.15) IGE as follows:

$$\begin{cases} T_{IGE} = C_T^{IGE} \rho A (\Omega R_{rad})^2 \\ \frac{C_T^{IGE}}{\sigma a} = \frac{C_T^{OGE}}{\sigma a} + \frac{\delta v_i}{4\Omega R_{rad}} \end{cases} \tag{2.20}$$

Then we compared the variation of the inflow velocity in and out of ground effect using OS4 simulator. In Fig. 2.3 we plot the ratio $(\delta v_i / v_i)$ obtained by simulation and by analytical derivation. The influence is perceptible for

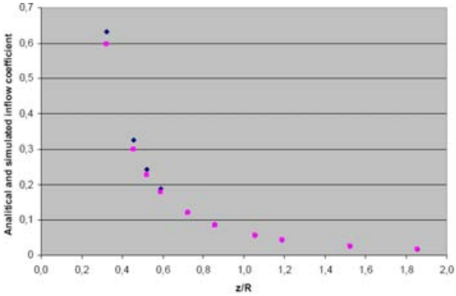


Figure 2.3: Simulation: Ground effect influence on the inflow velocity.

$z/R_{rad} \approx 2$ but becomes important near $z/R_{rad} < 1$. It seems then, that in the case of a quadrotor the ground effect influence is already present at one rotor diameter and becomes really important at one rotor radius. In order to empirically verify this assumption, we conducted a simple experiment which proved that a quadrotor deprived of altitude control can hover at a constant altitude at nearly one rotor diameter from the ground. It is clear that this result is only an indication of validity and does not constitute a formal proof.

2.3.2 General Moments and Forces

Quadrotor motion is obviously caused by a series of forces and moments coming from different physical effects. This model considers the following ones (with c:cos, s:sin).

Rolling Moments

body gyro effect	$\dot{\theta}\dot{\psi}(I_{yy} - I_{zz})$
propeller gyro effect	$J_r\dot{\theta}\Omega_r$
roll actuators action	$l(-T_2 + T_4)$
hub moment due to sideward flight	$h(\sum\limits_{i=1}^{4} H_{yi})$
rolling moment due to forward flight	$(-1)^{i+1}\sum\limits_{i=1}^{4} R_{mxi}$

Pitching Moments

body gyro effect	$\dot{\phi}\dot{\psi}(I_{zz} - I_{xx})$
propeller gyro effect	$J_r \dot{\phi} \Omega_r$
pitch actuators action	$l(T_1 - T_3)$
hub moment due to forward flight	$h(\sum\limits_{i=1}^{4} H_{xi})$
rolling moment due to sideward flight	$(-1)^{i+1} \sum\limits_{i=1}^{4} R_{myi}$

Yawing Moments

body gyro effect	$\dot{\theta}\dot{\phi}(I_{xx} - I_{yy})$
inertial counter-torque	$J_r \dot{\Omega}_r$
counter-torque unbalance	$(-1)^{i} \sum\limits_{i=1}^{4} Q_i$
hub force unbalance in forward flight	$l(H_{x2} - H_{x4})$
hub force unbalance in sideward flight	$l(-H_{y1} + H_{y3})$

Forces Along z Axis

actuators action	$c\psi c\phi(\sum\limits_{i=1}^{4} T_i)$
weight	mg

Forces Along x Axis

actuators action	$(s\psi s\phi + c\psi s\theta c\phi)(\sum\limits_{i=1}^{4} T_i)$
hub force in x axis	$-\sum\limits_{i=1}^{4} H_{xi}$
friction	$\frac{1}{2} C_x A_c \rho \dot{x} \lvert \dot{x} \rvert$

Forces Along y Axis

actuators action	$(-c\psi s\phi + s\psi s\theta c\phi)(\sum\limits_{i=1}^{4} T_i)$
hub force in y axis	$-\sum\limits_{i=1}^{4} H_{yi}$
friction	$\frac{1}{2} C_y A_c \rho \dot{y} \lvert \dot{y} \rvert$

2.3.3 Equations of Motion

The equations of motion are derived from (2.14) and all the forces and moments listed in Subsection 2.3.2.

$$\begin{cases} I_{xx}\ddot{\phi} = \dot{\theta}\dot{\psi}(I_{yy} - I_{zz}) + J_r\dot{\theta}\Omega_r + l(-T_2 + T_4) - h(\sum_{i=1}^{4} H_{yi}) + (-1)^{i+1}\sum_{i=1}^{4} R_{mxi} \\ I_{yy}\ddot{\theta} = \dot{\phi}\dot{\psi}(I_{zz} - I_{xx}) - J_r\dot{\phi}\Omega_r + l(T_1 - T_3) + h(\sum_{i=1}^{4} H_{xi}) + (-1)^{i+1}\sum_{i=1}^{4} R_{myi} \\ I_{zz}\ddot{\psi} = \dot{\theta}\dot{\phi}(I_{xx} - I_{yy}) + J_r\dot{\Omega}_r + (-1)^{i}\sum_{i=1}^{4} Q_i + l(H_{x2} - H_{x4}) + l(-H_{y1} + H_{y3}) \\ m\ddot{z} = mg - (c\psi c\phi)\sum_{i=1}^{4} T_i \\ m\ddot{x} = (s\psi s\phi + c\psi s\theta c\phi)\sum_{i=1}^{4} T_i - \sum_{i=1}^{4} H_{xi} - \frac{1}{2}C_x A_c \rho \dot{x}|\dot{x}| \\ m\ddot{y} = (-c\psi s\phi + s\psi s\theta c\phi)\sum_{i=1}^{4} T_i - \sum_{i=1}^{4} H_{yi} - \frac{1}{2}C_y A_c \rho \dot{y}|\dot{y}| \end{cases} \tag{2.21}$$

2.3.4 Rotor Dynamics

OS4 is equipped with four fixed-pitch rotors (no swash plate), each one includes a Brush-Less Direct Current (BLDC) motor, a one-stage gearbox and a propeller. The entire rotor dynamics was identified and validated using the Matlab Identification Toolbox. A first-order transfer function (2.22) is sufficient to reproduce the dynamics between the propeller's speed set-point and its true speed.

$$G(s) = \frac{0.936}{0.178s + 1} \tag{2.22}$$

It is worthwhile to note the non-unity gain in (2.22), this is visible in Fig 2.4, which superimposes the model output (red) and the sensor data (blue) to a step input (green). In fact, sensorless BLDC motors require a minimum speed to run thus, the set-point does not start from zero. The motor used does not incorporate hall effect sensors; the identification was carried out using a reflective encoder placed under the propeller gear.

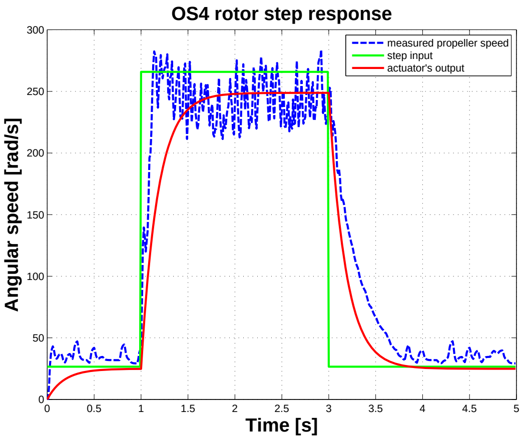


Figure 2.4: Rotor and model step response, measured at propeller's shaft.

2.4 OS4 Simulator

OS4 simulator development followed the successive improvements made to the dynamic model, the control scheme and the robot hardware. The last version includes identified actuator's dynamics, aerodynamics block, obstacle avoidance controller (OAC) and a high level planner for way-points definition. Each block is described by one or several Matlab files and can be easily incorporated into other simulators. The simulation starts with the initial state taken from the dedicated block "initial conditions" (see Fig. 2.5). After that, the set of data is degraded with delay and white noise and then filtered. It is then used in the control block and the inputs are sent to the motor dynamics block. The estimated rotors' speed feeds the aerodynamics block, which outputs the forces and moments of each propeller. This is sent to the system dynamics block, along with the state and the actual rotors' speed to process the new state. The block "control" is in fact dedicated to attitude, altitude and position controllers as schematized in Fig. 2.6. The simulator permits also various high level simulations like way-point following (see Fig. 4.23) or obstacle avoidance (see Fig. 4.26).

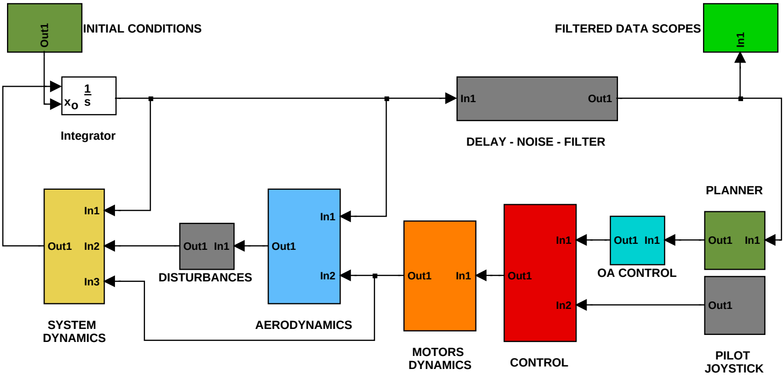


Figure 2.5: OS4 simulator block diagram.

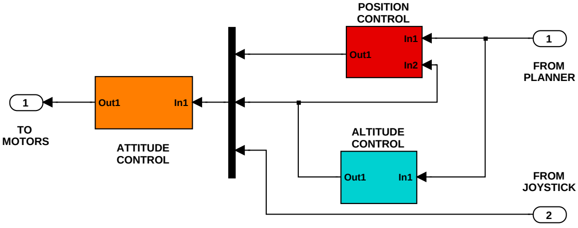


Figure 2.6: Control block in OS4 simulator.

2.5 Conclusion

This chapter presented the dynamic modelling using two different approached, of the two platforms designed in this thesis. The test-bench and the OS4. The first model contains basic physics equations and assumes fixed aerodynamics coefficients. The Propulsion Group (PG) dynamics are also tackled based on standard DC motor and gearbox equations. The second model is more realistic as it introduces variable aerodynamics coefficients. Moreover, the rotor dynamics were identified into a first order model. Both vehicles' parameters were extracted from a faithful CAD model for more reliability. This is to say, real flight experiments were conducted with the same control parameters tuned in simulation. On the other hand, an advanced simulator was developed based on this model, it takes into account realistic sensors' delay and noise. Moreover, it allows the simulation and control design of low and high level controllers. Namely, attitude, altitude, position controls in addition to obstacle avoidance and autonomous take-off and landing.

Chapter 3

System Design

3.1 Concept and Generalities

The concept followed in this thesis for system design was first to build a test-bench for easy testing of control strategies, then develop a methodology adapted to rotorcraft optimal design, and finally apply it to design the quadrotor. In general, the focus was on small vehicles able to fly indoor and having simple mechanics.

3.2 Test-Bench Design

The development of a control system for flying robots requires the development of an adequate test-bench for the preliminary experiments. This helps to lock some degrees of freedom in order to reduce control complexity and avoid system damage. OS4 test-bench allows only the three rotations (roll, pitch and yaw). The cross is made with carbon rods and the flying part weighs about 240 g as shown in Fig. 3.1. The test-bench is interfaced with a serial link to a computer. The RS232 to I^2C module translates serial data to the I^2C bus motor modules (see Fig. 3.2). These modules include a PID regulator on a PIC16F876 microcontroller and are capable of open or closed-loop operation in position, speed or torque control. The MT9-B IMU estimates with a Kalman filter the 3D orientation data and gives the calibrated data of acceleration and angular velocity. It weighs about 33 g and communicates at 115 kbps. The captured motion from the 3D universal joint can be decoded to extract absolute orientation information, thanks to the micro optical encoders in each axis. The flying part is thus only about

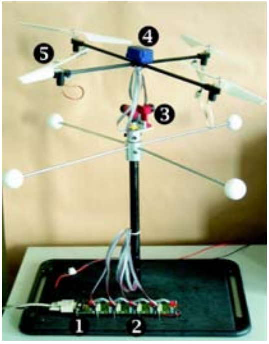


Figure 3.1: OS4 test-bench. **(1)** RS232 to I^2C translator, **(2)** Motor modules, **(3)** Universal joint, **(4)** Micro IMU, **(5)** Propulsion group.

235 g. OS4 test-bench has 4 propulsion groups, each one is composed of a 29 g motor including magnetic encoders, a 6 g gearbox and a 6 g propeller.

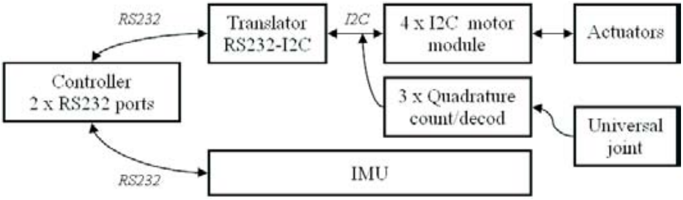


Figure 3.2: OS4 test-bench block diagram.

3.3 Small-Scale Rotorcraft Design

The interdependency of components during the design phase makes the choice of each one strongly conditioned by the choice of all the others. Starting such a design and taking a decision concerning all variables requires to follow an appropriate methodology. OS4 was designed following a practical method we developed to handle the design problematic of a small scale rotorcraft. The method combines models and databases of components and produces the best selection. Moreover, it provides the required battery mass to use in order to comply with the target mass constraint.

3.3.1 The General Method

The design process starts by setting three design constraints from the application definition: Maximum mass m_{max}, maximum span s_{max} and target thrust/weight ratio β. This gives a good idea about the propeller diameter to use d_{prop}. In practice, the propeller span defines the overall span of the helicopter (one can hardly imagine a big helicopter with a small propeller!). Using the propeller diameter d_{prop}, one can estimate the characteristics of the propeller in term of thrust, drag and power for a range of angular speeds. Paper [26] proposes for this purpose, the models $T \propto \Omega^2 L^4$, $D \propto \Omega^2 L^5$ and $P \propto \Omega^3 L^5$, where L is a reference dimension, *e.g.* the distance between the propeller center and the center of the blade. So, the mass m_{max}, the drag moment D and the thrust/weight ratio β are enough to fully define the motor power requirements. This allows the algorithm to select from the database a list of candidate actuators which offer the required power. Then, a rough estimation of the mass of the airframe m_{af} and avionics m_{av} is necessary to have a first estimation of the total mass without battery (see Fig. 3.3). This estimation can come from a formulation of the scaling laws or simply from an avionics data base as it was done for OS4. So, the helicopter mass is $m_{hel} = m_{af} + m_{av} + m_{pg} + m_{bat}$, where m_{pg} is the mass of the propulsion group (propeller, gearbox, motor) and m_{bat} is the mass of the battery. The iterative algorithm will find m_{pg} and m_{bat} as described hereafter.

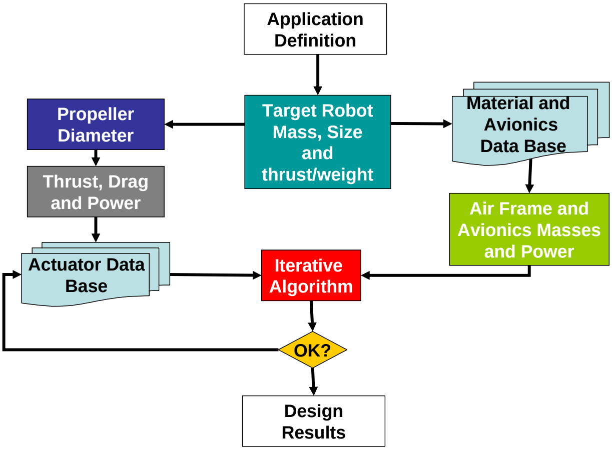


Figure 3.3: The design method flowchart. The user has to define a target mass, span and thrust/weight ratio of his system.

3.3.2 The Iterative Algorithm

This algorithm starts by picking up one candidate actuator from the database, its mass is $m_{pg}(i)$. Then, an initial value $m_{bat}(j_0)$ is given to m_{bat} variable. So far, we have all the variables to determine $m_{hel}(i,j) = m_{af} + m_{av} + m_{pg}(i) + m_{bat}(j)$. The temporary total mass of the helicopter $m_{hel}(i,j)$ is used to estimate various variables (see Fig. 3.4) at two operational points: At hover and at maximum thrust. For each candidate actuator, the variable $m_{bat}(j)$ is incremented until the maximum possible battery mass $M_{BAT_{max}}$ is reached. This process makes it possible to estimate, at hover and at maximum thrust, for each candidate actuator, and at each increment of $m_{bat}(j)$ the following four design indicators.

- Propulsion group cost factor C
- Propulsion group quality factor Q_{pg}
- Operational time (autonomy) Au
- Design quality index Q_{in}

Propulsion Group Cost Factor

The PG cost factor describes the cost in power of each gram lifted. It often happens that a MFR designer has to decide what to select, a lightweight component which consumes more power or a heavier component which consumes less power. The cost factor is an indicator that helps to solve the PG selection dilemma by directly looking at the power spent for each gram of a given solution. The cost factor C formulation proposed in this thesis is:

$$C = \frac{P_{el}}{(\frac{T}{g} - m_{pg})} \quad [W/g] \tag{3.1}$$

Propulsion Group Quality Factor

This PG quality factor describes the quality of mass lifting. It is somehow the answer to *how good am I lifting this mass?*, *am I disturbing the system while lifting?*, *am I changing the lift with high bandwidth?*. The propulsion group quality factor is necessary to take into account the notions of actuator bandwidth and thrust/weight ratio. The quality factor Q_{pg} formulation proposed in this thesis is:

$$Q_{pg} = \frac{B_W\,\beta}{\Omega\,C} \quad [Hzg/RadW] \tag{3.2}$$

Operational Time (Autonomy)

This system level indicator describes the endurance of the system in flight. Its links the battery mass to the power consumption and it depends directly on the battery capacity C_{bat} which is a technological limitation. A common formulation of the operational time is:

$$Au = \frac{m_{bat} C_{bat}}{P_{el}} \quad [h] \tag{3.3}$$

Design Quality Index

This system level indicator constraints the operational time with respect to the total power consumption. This factor is proposed as an indicator of the efficiency at the system level. The design quality index formulation proposed in this thesis is:

$$Q_{in} = \frac{Au}{P_{el}} \quad [h/W] \tag{3.4}$$

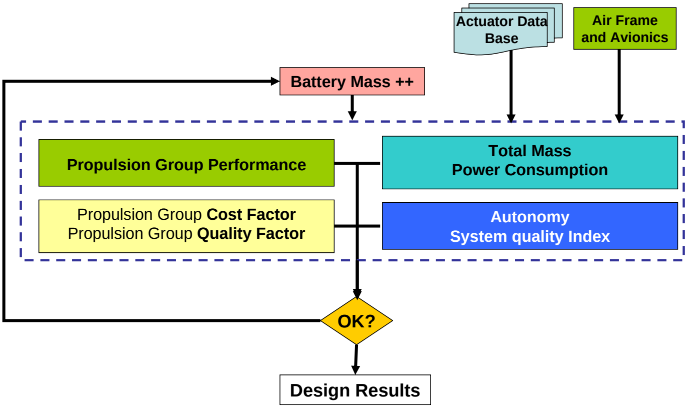

Actuator Data Base
Air Frame and Avionics
Battery Mass ++
Propulsion Group Performance
Total Mass Power Consumption
Propulsion Group Cost Factor Propulsion Group Quality Factor
Autonomy System quality Index
OK?
Design Results

Figure 3.4: The iterative algorithm flowchart.

The Method Formulation

The design methodology is formulated using standard physics equations, in addition to usual electrical DC motor formulas, and three equations introduced as design indicators. These equations are implemented as indicated in the Algorithm 1. The subscript $_{equ}$ is used with the variables at the equilibrium point (roll and pitch at zero), the subscript $_{max}$ is for the variables at the maximum thrust point (maximum roll and pitch without loosing altitude). The subscript $_{mean}$ is for the variables at the weighted average between the equilibrium an the maximum points. One exception holds for the masses where $_{max}$ stand for the maximum mass.

Algorithm 1 A simplified formulation of the general method.

```
$M_{BAT_{max}} = m_{max} - (m_{af} + m_{av} + m_{bat_{av}} + n * m_p)$
for $i = 0$ to $i_{last\_actuator}$ do
  $P_{out_{max}}(i) = P_{el}(i) * \eta$
  $\Omega_{max}(i) = [P_{out_{max}}(i)/d]^{\frac{1}{3}}$
  $M_{max_{possible}}(i) = b * \Omega_{max}(i)^2/g$
  for $j = 0$ to $j_{M_{BAT_{max}}}$ do
    $m_{hel}(i,j) = m_{af} + m_{av} + m_{bat_{av}} + n * m_{pg}(i) + m_{bat}(j)$
    $M_{max_{requested}}(i,j) = m_{hel}(i,j) * \beta/n$
    if $M_{max_{possible}}(i) \geq M_{max_{requested}}(i,j)$ then
      $M_{equ}(i,j) = m_{hel}(i,j)/n$
      $\Omega_{equ}(i,j) = [M_{equ}(i,j) * g/b]^{\frac{1}{2}}$
      $P_{out_{equ}}(i,j) = d * \Omega_{equ}(i,j)^3$
      $P_{el_{equ}}(i,j) = P_{out_{equ}}(i,j)/[\eta_{m_{equ}}(i) * \eta]$
      $P_{el_{max}}(i,j) = P_{out_{max}}(i,j)/[\eta_{m_{max}}(i) * \eta]$
      $P_{el_{mean}}(i,j) = P_{av} + [P_{el_{equ}}(i,j) * \varrho_{equ} + P_{el_{max}}(i,j) * (1 - \varrho_{equ})] * n$
      $m_{bat_{tot}}(i,j) = m_{bat}(i,j) + m_{bat_{av}}$
      $C_{equ}(i,j) = P_{el_{equ}}(i,j)/[M_{equ}(i,j) - m_{pg}(i)]$
      $Q_{pg_{equ}} = B_w * \beta/\Omega_{equ} * C_{equ}(i,j)$
      $Au(i,j) = m_{bat_{tot}} * C_{bat}/P_{el_{mean}}(i,j)$
      $Q_{in}(i,j) = Au(i,j)/P_{el_{mean}}(i,j)$
    end if
  end for
end for
```

3.4 OS4 Quadrotor Design

The method presented in Subsection 3.3.1 was used to design OS4 starting from three main constraints: 500 g maximum mass, 800 mm maximum span and a desired thrust/weight ratio of $\beta = 2$ (the helicopter has to lift two times its weight). On the other hand, 300 mm is a reasonable choice for our propeller diameter. This is done by taking the largest possible propeller which

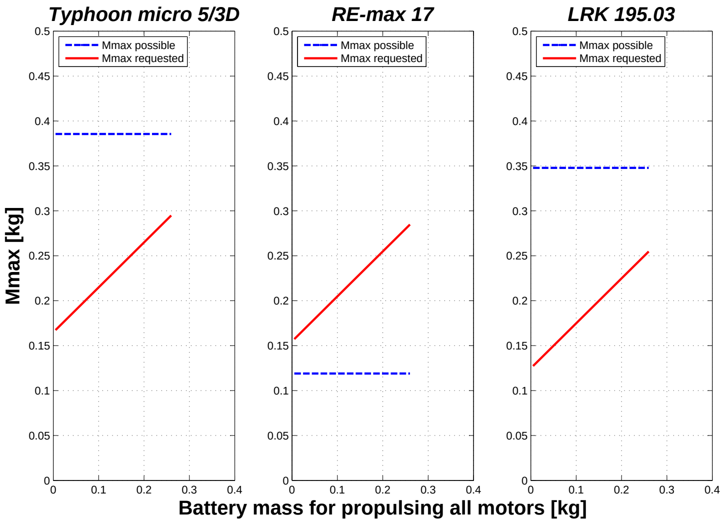


Figure 3.5: The maximum mass one motor can lift, and the requested one over the range of the battery mass for the initial design of OS4.

respects the target span constraint. After running the program implemented based on the algorithm in 1, we obtain the plots in Fig 3.5 and 3.6. One can see in Fig. 3.5, that the maximum mass the motor RE-max 17 can lift is always lower than what is requested to meet the $\beta = 2$ requirement. This means that the RE-max 17 is not adapted to our design, so we discard it. One can also see that the two other motors, the Typhoon and the LRK fulfill the requirement up to a battery mass of about 250 g. At this stage, the Typhoon motor seems to be better as it shows a higher maximum lifted mass. These two selected motors are then compared through the evaluation of the four factors like shown in Fig. 3.6. The cost factor, which is the cost in power of each gram lifted, increases because we lift more mass with the same rotor. Then, the quality factor decreases because by lifting more we have a higher cost factor and potentially more gyroscopic effects. The third indicator is simply the endurance (autonomy), it is directly linked to the battery mass. Finally, we look at the system design index which is an indicator of the efficiency at the system level. This is to say, C and Q_{pg} are propulsion group level indicators, while Au and Q_{in} are system level indicators. Based on this, the LRK motor is finally selected as the best actuator for OS4 quadrotor. Moreover, the program allows maximum 250 g

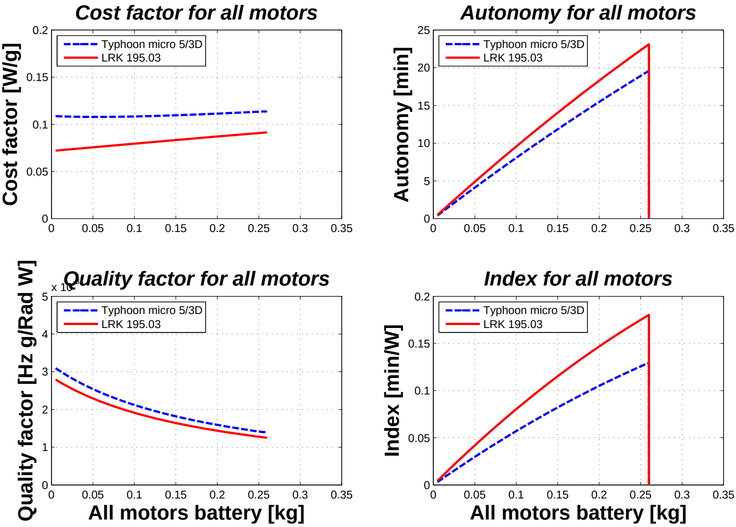


Figure 3.6: The four design indicators in the initial design of OS4.

for the battery. The best combination of Lithium-Polymer cells available on the market leads to: $m_{bat} = 230\,g$ (11 V, 3.3 Ah). The motors and battery are part of a whole set of components like presented in Fig. 3.7 which is OS4's block diagram.

3.4.1 Design Results

The initial design of OS4 is the result of the direct application of the method described before. The initial robot mass and power distributions are shown in Fig. 3.8. The total mass is about 520 g, where the battery takes almost one-half and the actuators only one-third thanks to BLDC technology. All the actuators obviously take the lion's part, 60 W of 66 W average power consumption. However, the latter depends on flight conditions and represents a weighted average between the equilibrium (40 W) and the worst possible inclination state (120 W) without loosing altitude. Fig. 3.9 shows the initial robot. Several mechanical and electrical parts were added since then in order to reinforce the structure or add functionalities. The mass and power distributions of the final version of the robot are shown in Fig. 3.10. A detailed description of OS4 is available in Appendix: E, along with the table of parameters. Fig. 3.11 shows the final version of OS4 helicopter.

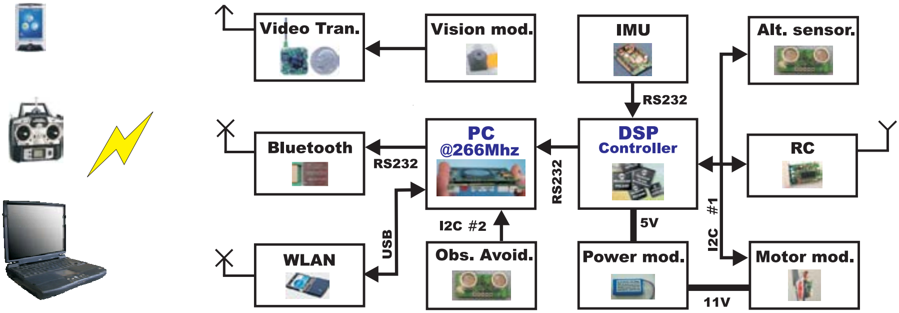


Figure 3.7: OS4 block diagram. A DSP processor handles attitude and altitude control. Then, a miniature PC handles obstacles avoidance control and communication tasks. The robot communicates through a wifi interface and accepts standard remote control signals.

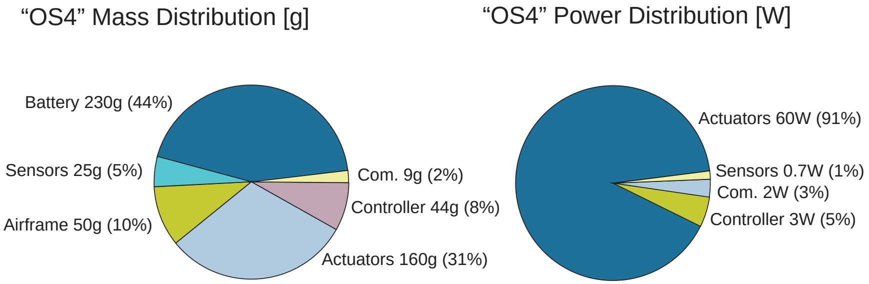


Figure 3.8: Mass and power distributions of the initial design of OS4. The battery mass represents almost one half of the total mass and the actuators sink 90% of the power.

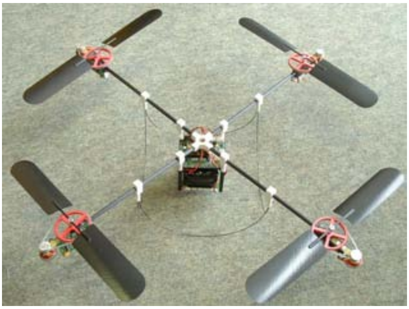


Figure 3.9: Initial design of OS4.

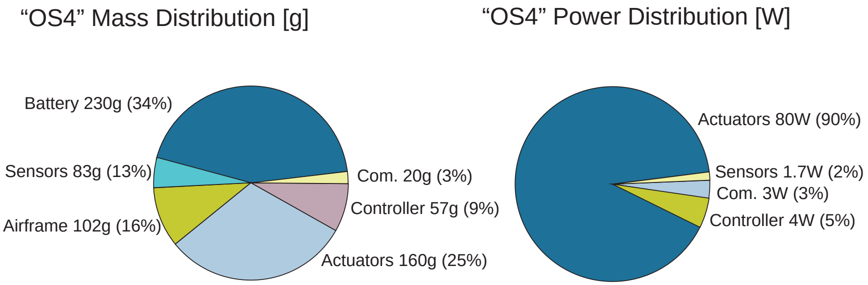


Figure 3.10: Mass and power distributions of the last version of OS4. The battery mass represents one third of the total mass and the actuators sink 90% of the power.

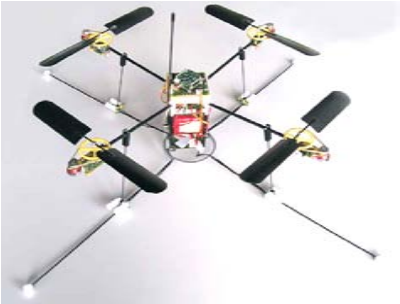


Figure 3.11: The last version of OS4 (see details in Appendix: E).

3.4.2 Propulsion Group

The main design variables of a propulsion group are listed in Table 3.1. They were used in the models of Table 3.2. On the rotor side, the tests revealed that a gearbox is necessary. In fact, a direct-drive propulsion group would allow only a thrust/weight ratio of $\beta = 0.75$, which is obviously not enough to lift the robot. The selected motor (LRK) is a brushless DC motor (12 g, 35W) with a high power/weight ratio. A 6 g I^{2}C controller was specially designed for the sensorless outrunner LRK195.03 motor as shown in Fig. 3.12. Obviously, BLDC motors offer high life-time and low electromagnetic noise. The ready to plug propulsion group weighs 40 g and lifts more than 320 g, intentionally limited to 260 g in order to avoid high currents and motor heating.

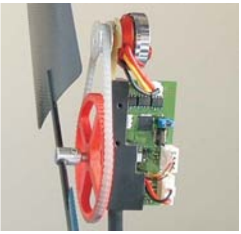


Figure 3.12: OS4 propulsion group. The module is interfaced through I^{2}C bus and has a local PI speed controller.

3.4.3 Computer Module

Embedding the controller for our application is definitely advisable as it avoids all the delays and the discontinuities in wireless connections. A miniature computer module, based on Geode 1200 processor running at 266 MHz with 128 M of RAM and flash memory was developed. The computer module is x86 compatible and offers all standard PC interfaces. The whole computer is 44 g in mass, 56 mm by 71 mm in size and runs Linux. The controller includes an MCU for interfacing Bluetooth with the computer module. The same chip is used to decode the Pulse Position Modulation (PPM) signal picked up from a 1.6 g, 5 channels commercially available RC receiver. This decoding on our MCU makes it possible to interface the RC receiver to I^{2}C bus and at the same time detect any anomaly in the channels. It is also possible to control the helicopter using a standard remote control.

Table 3.1: OS4 propulsion group design variables.

propeller		OS4	unit
mass	m_p	5.2	g
thrust coeff.	b	3.13e-5	$\mathrm{N\,s^2}$
drag coeff.	d	7.5e-7	$\mathrm{Nm\,s^2}$
inertia	J_r	6e-5	$\mathrm{kg.m^2}$
gearbox		OS4	unit
efficiency	η	90	%
mass	m_{gb}	7	g
max. torque		0.15	Nm
max. speed		1000	rad/s
red. ratio	r	4:1	
motor		OS4	unit
effi. at hover	η_m	64	%
mass	m_m	12	g
max. power	P_{el}	35	W
internal res.	R_{mot}	0.6	Ω
inertia	J_m	4e-7	$\mathrm{kg\,m^2}$
torque cst.	k	5.2	mNm/A

Table 3.2: Models of the propulsion group components.

component	model
Propeller	$(b,d)\Omega^2 = (T,D)$
Gearbox	$P_{in}\eta = P_{out}$
DC motor	$-\frac{k^2}{R_{mot}}\omega - D + \frac{k}{R_{mot}}u = J\frac{d\omega}{dt}$

Figure 3.13: The x-board based, 40 g and 56x71 mm computer module.

3.4.4 Position Sensor

OS4's position sensor is based on an on-board down-looking CCD camera and a simple pattern on the ground. The camera provides a motion-blur free image of 320x240 at up to 25 fps as shown in Fig. C.8. The algorithm detects the pattern, estimates the pose and provides the camera position (x,y) and heading angle (ψ). The image is primarily sent to an off-board computer for processing and then the position data is sent-back to the helicopter for control purpose as shown in Fig. 3.14. Several possibilities were considered for pattern detection. The first method tested is the detection of five red dots on a A4 paper. This method suffers from sensitivity to lighting conditions. For the second test, we considered four LEDs with different colors on an A4 board. However, it was hard to tune the LEDs intensity for the overall working volume. Finally, we used a red A4 paper with a white spot shifted from the pattern center. This time the pattern was robustly detected. We use for that Canny edge detector and Douglas-Peucker algorithms already implemented in OpenCV [27]. In addition, we run a least-square based linear regression to refine the detection. The pattern before and after detection is shown in Fig. C.9. Pose estimation is then performed using PnP algorithm [28]. The sensor algorithm is afterwards enhanced with a management of different situations where the pattern is not or partially detected. All the processing takes about 7 ms. Image capture takes 1 ms with a PCI acquisition card and almost 20 ms with a USB 1.1 device on a Pentium 4, 2.4 GHz. Anyway, the algorithm is limited to 25 Hz by the camera frame rate (25 fps). Figure C.10 shows a comparison between position estimation obtained with our sensor

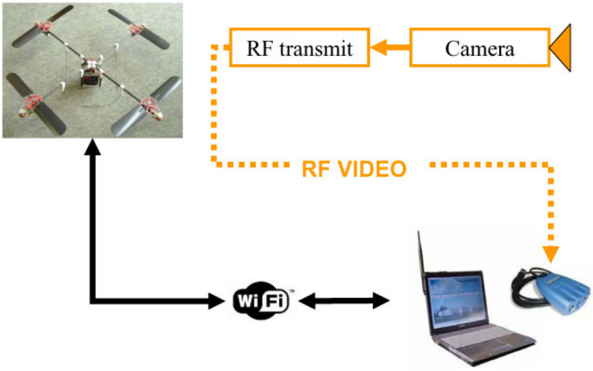


Figure 3.14: Position sensing setup on OS4.

and an encoder of a linear motorized slider at 1 m/s. The errors obtained in x and y position sensing are about 2 cm at 0.5 m/s. The error on the yaw is about 3 ° at 180 °/s.

3.4.5 Obstacle Detection Setup

Four ultrasound range finders are mounted on OS4 for obstacle detection, one under each propeller (see Fig. 3.15). Two short plastic tubes are mounted on each sensor in order to reduce the beam width.

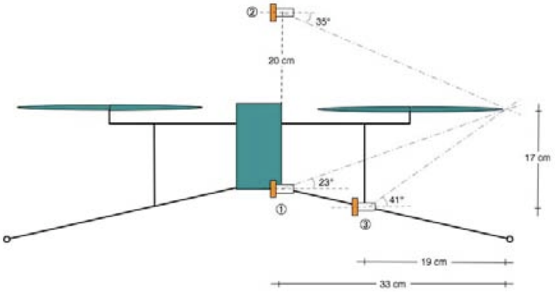


Figure 3.15: Possible US sensors arrangement on OS4. Position (3) was adopted after various testing.

3.4.6 OS4 vs Other Quadrotors

The methodology developed in this work shall permits better design performance if followed carefully. The evaluation of the design performance is done through the analysis of three parameters. The first one is the battery mass contribution to the total mass. In an ideal MAV this contribution would be 100%, that would be a flying battery! The second parameter is the motors mass contribution to the total mass. Ideally this contribution would be 0%, that would be like a motor without mass! The third parameter is the thrust margin. Ideally the thrust margin is infinite! In comparison with other similar quadrotors, OS4 demonstrate very good design performance like shown in Fig. 3.16, where OS4 is compared to three other robots for which the data are available. The X4-Flyer (ANU) [29], a robust 4 kg flying machine. Draganfly (DF Innov.) [1], a commercial platform widely used in research and the Starmac 2 (Stanford) [30], probably the closest design to OS4. The battery mass contribution in OS4 is about 35%, this is 50% better than the closest

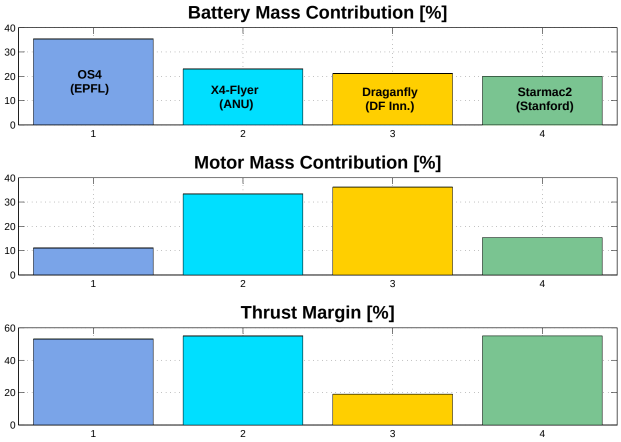


Figure 3.16: OS4 in comparison with other quadrotors.

competitor. The motors mass contribution in OS4 is only 11% which is at least 38% below the others. The thrust margin is more delicate to compare. In fact, it is not clear from the publications if the thrust margin is calculated with or without the payload. Anyway, OS4 seems to have comparable thrust/weight ratio to the other designs. These results highlight, not only the quality of OS4, but the utility of the design methodology.

Table 3.3: Some parameters of OS4 in comparison with other quadrotors.

	OS4 (EPFL)	X4-Flyer (ANU)	Draganfly (DF Inn.)	Starmac2 (Stanford)
Mass [kg]	**0.65**	4.34	0.52	1.3
Battery mass [kg]	**0.23**	1	0.11	0.26
Battery mass/Mass ratio	**0.35**	0.23	0.21	0.2
Battery mass contrib. [%]	**35.4**	23	21.1	20
Motors mass [kg]	**0.07**	1.45	0.19	0.2
Motors mass contrib. [%]	**11**	33.4	36.1	15.4
Thrust margin [%]	**53**	55	19	55

3.5 Conclusion

This chapter presented the design of the two vehicles developed in this thesis. The test-bench and the OS4. The first system is only capable of 3 DoF which facilitates the testing of the controllers. However, it is possible to detach the flying part in order to test free flights. Before designing the second system which is a free flying quadrotor, a new design methodology is introduced. It allows an optimal design of small-scale rotorcraft. Four new design indicators were introduced for a precise and complete evaluation of the design performance. This methodology appreciably facilitated the components selection process and battery dimensioning of OS4. This quadrotor exhibits higher capabilities and endurance than the competition. This is verified through the comparison of different design parameters. OS4 embeds all the necessary avionics and energy devices for a fully autonomous flight. This comprises a low cost IMU, a vision based position sensor specifically developed for this project and an obstacle detection setup.

Chapter 4

System Control

4.1 Concept and Generalities

The concept followed in this thesis for system control was to evaluate several control techniques, tune them in simulation and evaluate them on the test-bench. After which, implement the best and mature controller on the helicopter. In general, we used model based cascaded controllers implemented onboard for real-time operation. The focus was mainly on attitude control as it is the heart of the control problem.

4.2 Modelling for Control

The model (2.21) developed in subsection 2.3.3 describes the differential equations of the system. It is advisable for control design to simplify the model in order to comply with the real-time constraints of the embedded control loop. Hence, hub forces and rolling moments are neglected and thrust and drag coefficients are supposed constant. The system can be rewritten in state-space form $\dot{X} = f(X, U)$ with U inputs vector and X state vector chosen as follows:

State vector

$$X = [\phi \quad \dot{\phi} \quad \theta \quad \dot{\theta} \quad \psi \quad \dot{\psi} \quad z \quad \dot{z} \quad x \quad \dot{x} \quad y \quad \dot{y}]^T \tag{4.1}$$

$$\begin{array}{l|l} x_1 = \phi & x_7 = z \\ x_2 = \dot{x}_1 = \dot{\phi} & x_8 = \dot{x}_7 = \dot{z} \\ x_3 = \theta & x_9 = x \\ x_4 = \dot{x}_3 = \dot{\theta} & x_{10} = \dot{x}_9 = \dot{x} \\ x_5 = \psi & x_{11} = y \\ x_6 = \dot{x}_5 = \dot{\psi} & x_{12} = \dot{x}_{11} = \dot{y} \end{array} \tag{4.2}$$

$$U = [U_1 \quad U_2 \quad U_3 \quad U_4]^T \tag{4.3}$$

where the inputs are mapped by:

$$\begin{cases} U_1 = b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\ U_2 = b(-\Omega_2^2 + \Omega_4^2) \\ U_3 = b(\Omega_1^2 - \Omega_3^2) \\ U_4 = d(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2) \end{cases} \tag{4.4}$$

The transformation matrix between the rate of change of the orientation angles $(\dot{\phi}, \dot{\theta}, \dot{\psi})$ and the body angular velocities (p, q, r) can be considered as unity matrix if the perturbations from hover flight are small. Then, one can write $(\dot{\phi}, \dot{\theta}, \dot{\psi}) \approx (p, q, r)$. Simulation tests have shown that this assumption is reasonable. From (2.21), (4.1) and (4.3) we obtain after simplification:

$$f(X, U) = \begin{pmatrix} \dot{\phi} \\ \dot{\theta}\dot{\psi}a_1 + \dot{\theta}a_2\Omega_r + b_1U_2 \\ \dot{\theta} \\ \dot{\phi}\dot{\psi}a_3 - \dot{\phi}a_4\Omega_r + b_2U_3 \\ \dot{\psi} \\ \dot{\theta}\dot{\phi}a_5 + b_3U_4 \\ \dot{z} \\ g - (\cos\phi\cos\theta)\frac{1}{m}U_1 \\ \dot{x} \\ u_x\frac{1}{m}U_1 \\ \dot{y} \\ u_y\frac{1}{m}U_1 \end{pmatrix} \tag{4.5}$$

With:

$$\begin{array}{l|l} a_1 = (I_{yy} - I_{zz})/I_{xx} & b_1 = l/I_{xx} \\ a_2 = J_r/I_{xx} & b_2 = l/I_{yy} \\ a_3 = (I_{zz} - I_{xx})/I_{yy} & b_3 = 1/I_{zz} \\ a_4 = J_r/I_{yy} & \\ a_5 = (I_{xx} - I_{yy})/I_{zz} & \end{array} \tag{4.6}$$

$$\begin{aligned} u_x &= (\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi) \\ u_y &= (\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi) \end{aligned} \tag{4.7}$$

It is worthwhile to note in the latter system that the angles and their time derivatives do not depend on translation components. On the other hand, the translations depend on the angles. One can ideally imagine the overall system described by (4.5) as constituted of two subsystems, the angular rotations and the linear translations (see Fig. 4.1).

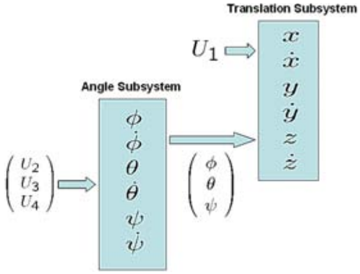


Figure 4.1: Connection of rotations and translations subsystems.

4.3 Control using Lyapunov Theory

This is the first control technique developed during this thesis and evaluated on the test-bench. It makes a direct use of Lyapunov control theory to stabilize the attitude of the quadrotor. From (2.9) and (4.2) we obtain:

$$f_\alpha(X,U) = \begin{pmatrix} x_2 \\ x_4x_6a_1 - x_4a_2\Omega_r + b_1U_2 \\ x_4 \\ x_2x_6a_3 + x_2a_4\Omega_r + b_2U_3 \\ x_6 \\ x_4x_2a_5 + b_3U_4 \end{pmatrix} \tag{4.8}$$

The angular rotations subsystem has as state the restriction X_α of X to the last 6 components which concerns roll, pitch, yaw angles and their time derivatives. The dynamics of these variables are described by $f_\alpha(X,U)$. This section mainly considers the stabilization of the angles in a particular configuration $X_\alpha^d = (x_1^d, 0, x_3^d, 0, x_5^d, 0)^T$. Let us consider the Lyapunov Function $V(X_\alpha)$ which is positive defined around the desired position X_α^d:

$$V(X_\alpha) = \frac{1}{2}[(x_1 - x_1^d)^2 + x_2^2 + (x_3 - x_3^d)^2 + x_4^2 + (x_5 - x_5^d)^2 + x_6^2] \tag{4.9}$$

The time derivative of (4.9), $\dot{V} = (\nabla V)^T f_\alpha$, in the case of a perfect cross VTOL ($I_{xx} = I_{yy}$) is drastically reduced to (4.10), where Ω_r does not appear.

$$\dot{V} = (x_1 - x_1^d)x_2 + x_2\frac{l}{I_{xx}}U_2 + (x_3 - x_3^d)x_4 + x_4\frac{l}{I_{yy}}U_3 + (x_5 - x_5^d)x_6 + x_6\frac{1}{I_{zz}}U_4 \quad (4.10)$$

By simply choosing:

$$\begin{cases} U_2 = -\frac{I_{xx}}{l}(x_1 - x_1^d) - k_1 x_2 \\ U_3 = -\frac{I_{yy}}{l}(x_3 - x_3^d) - k_2 x_4 \\ U_4 = -I_{zz}(x_5 - x_5^d) - k_3 x_6 \end{cases} \quad (4.11)$$

with k_1, k_2 and k_3 positive constants, we obtain for (4.10):

$$\dot{V} = -x_2^2\frac{l}{I_{xx}}k_1 - x_4^2\frac{l}{I_{yy}}k_2 - x_6^2\frac{1}{I_{zz}}k_3 \quad (4.12)$$

which is only negative semi-definite. By Lyapunov theorem [31] the simple stability for equilibrium is now ensured. By LaSalle invariance theorem we can ensure also that starting from a level curve of the Lyapunov function defined in (4.9), where $V(X_\alpha)$ is constant, the state evolution is constrained inside the region bounded by the level curve. This is very useful when trying to avoid a particular configuration; it is simply necessary to start with a level curve not containing these points and apply the previous defined controls. We can also ensure the asymptotic stability by applying LaSalle theorem because the maximum invariance set of rotations' subsystem under control (4.11) contained in the set $S = \{X_\alpha^S \in \Re^6 : \dot{V}|_{X_\alpha^S} = 0\}$ is restricted only to the equilibrium point. By the latter consideration we can ensure an asymptotical stability starting from a point in a set around the equilibrium. To ensure the global stability it is sufficient that the $\lim_{|X_\alpha| \to \infty} V(X_\alpha) = \infty$, which is our case.

4.3.1 Simulations

The controller's task in simulation was to compensate the initial error on the roll, pitch, yaw and stabilize these angles. The real system (test-bench) suffers from some delays and actuators's saturation. The delays are mainly due to RS232 communications and the actuators' dynamics. To emulate this lacks, two Simulink discrete-step delay blocks were introduced in the feedback loop and on the actuators. The motors have a maximum angular velocity of 600 rad/sec; a saturation block was placed between the controller and the delay block. Finally, the overall system was simulated at 30 Hz using a discrete time solver in order to model the behavior of the digital controller. In the simulation (see Fig. 4.2), the task is to hover in spite of the added normal Gaussian noise of variance 4 rad/sec on each angular velocity.

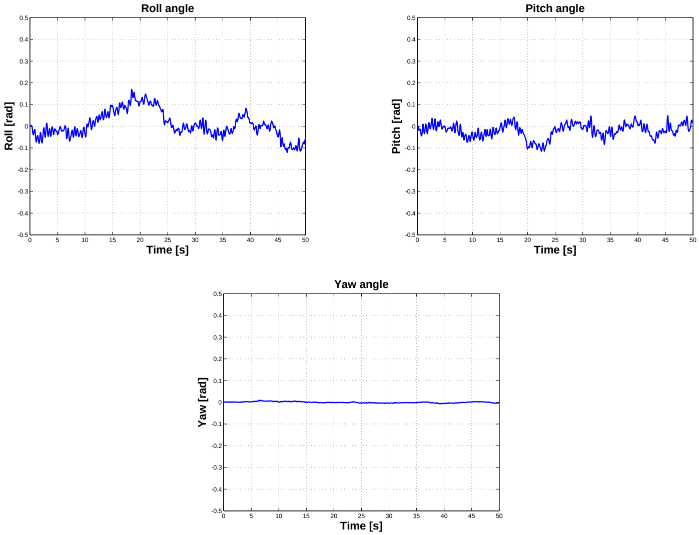


Figure 4.2: Simulation: The system has to maintain attitude angles to zero in spite of the added noise.

4.3.2 Experiments

In order to validate the control law developed in the previous section, we implemented the controller and we performed several experiments. The task was to control the orientation (roll, pitch and yaw), while the altitude was fixed by the test-bench (see Fig. 4.3). In spite of the test-bench limitations in term of delays and errors introduced by the tethering system, the experimental results obtained show that the proposed controller works well especially for the yaw angle.

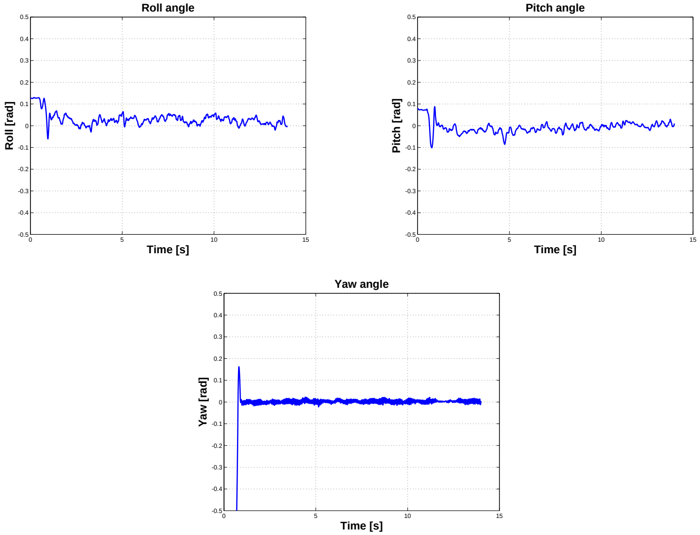


Figure 4.3: Experiment: The controller has to maintain attitude angles to zero. In spite of the huge initial condition on the yaw, the system is rapidly brought back to equilibrium.

4.4 Control using PID Technique

The dynamic model (2.9) presented above contains two gyroscopic effects. The influence of these effects is in our case less important than the motor's action. Especially if we consider a near-hover situation. In order to make it possible to design multiple PID controllers for this system [32], one can neglect these gyroscopic effects and thus remove the cross coupling. The model (2.9) is rewritten then:

$$\begin{cases} I_{xx}\ddot{\phi} = lU_2 \\ I_{yy}\ddot{\theta} = lU_3 \\ I_{zz}\ddot{\psi} = U_4 \end{cases} \tag{4.13}$$

If we include in (4.13), the rotor dynamics and rewrite the model in Laplace domain we obtain:

$$\begin{cases} \phi(s) = \frac{B^2 bl}{s^2(s+A)^2 I_{xx}}(u_4^2(s) - u_2^2(s)) \\ \theta(s) = \frac{B^2 bl}{s^2(s+A)^2 I_{yy}}(u_3^2(s) - u_1^2(s)) \\ \psi(s) = \frac{B^2 d}{s^2(s+A)^2 I_{zz}}(-1)^{i+1}\sum_{i=1}^{4} u_i^2(s) \end{cases} \tag{4.14}$$

Where A and B are the coefficients of the linearized rotor dynamics as described in (2.13), while C, too small comparing to B, is neglected. By using the control inputs U_i instead of the motor inputs u_i, (4.14) becomes:

$$\begin{cases} \phi(s) = \frac{A^2 l}{s^2(s+A)^2 I_{xx}}U_2 \\ \theta(s) = \frac{A^2 l}{s^2(s+A)^2 I_{yy}}U_3 \\ \psi(s) = \frac{A^2}{s^2(s+A)^2 I_{zz}}U_4 \end{cases} \tag{4.15}$$

The numerical application gives:

$$\begin{cases} \phi(s) = \frac{0.522}{0.004s^4+0.039s^3+0.009s^2}U_2 \\ \theta(s) = \frac{0.522}{0.004s^4+0.039s^3+0.009s^2}U_3 \\ \psi(s) = \frac{21.78}{0.008s^4+0.077s^3+0.18s^2}U_4 \end{cases} \tag{4.16}$$

4.4.1 PD Controller Synthesis and Simulation

A PD controller is introduced for each orientation angle:

$$U_{2,3,4} = k_{\phi,\theta,\psi}(\phi,\theta,\psi) + d_{\phi,\theta,\psi}(\phi,\theta,\psi) \tag{4.17}$$

We performed several simulations on Simulink using the complete model in order to tune the six control parameters. The controller's task was to stabilize the orientation angles. For these simulations, the dynamic model (2.9) was used, obtaining the results shown in Fig. 4.4. The simulated performance was satisfactory regarding the simple control synthesis approach.

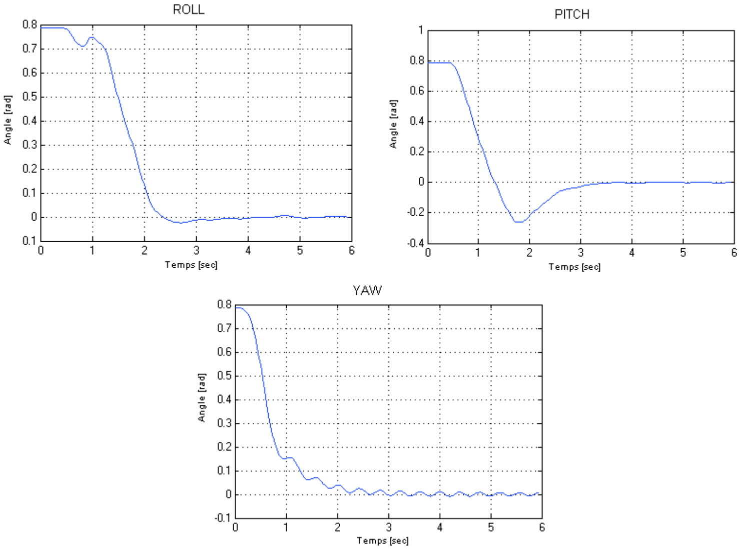


Figure 4.4: Simulation: The controller has to stabilize the orientation angles, starting from $\pi/4$ in roll, pitch and yaw as initial condition (P=0.8, D=0.4 for roll and pitch. P=0.8, D=0.5 for yaw angle).

4.4.2 PID Controller on the Real System

Finally, we implemented the controllers in C under Linux on a machine running at 450 MHz simulating the future integration of a single board computer. The experiment has shown that OS4 was not completely stabilized, as a small steady-state error remained. An integral term was then added and the experiment was performed including a closed-loop speed control on each rotor. The results are shown in Fig 4.5. The effect of propellers' speed control (inner loop) affects the general stabilization of the vehicle. In closed-loop, the orientation stabilization is faster and the yaw angle is well controlled. Contrarily, in open-loop, the response is much smoother. This highlights the importance of the actuators' fast response. In both cases, the simulations and the experiments have shown that the quadrotor can be controlled efficiently in hover using a classical approach. Obviously, this controller will not be able to stabilize the helicopter in presence of strong perturbations.

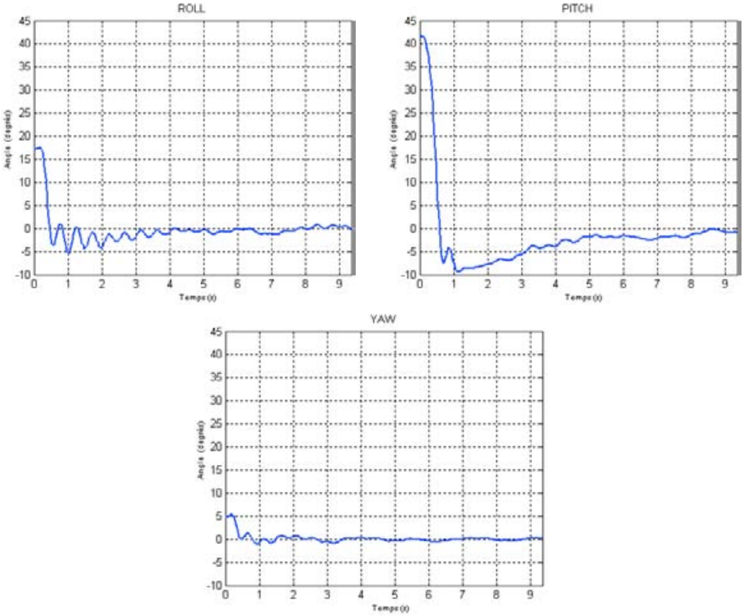


Figure 4.5: Experiment: The controller has to stabilize the attitude. A higher priority was given to roll and pitch control. An integral term was added to eliminate the steady-state error (P=0.9, I=0.3, D=0.2 for roll and pitch. P=0.06, I=0.3, D=0.02 for yaw angle). A PID control loop was applied locally to control every propeller's speed.

4.5 Control using Optimal Control Theory

Now we are considering the general equations for state-space system, cost function and state feedback for a linearized system:

$$\begin{cases} \dot{x} = Ax + Bu \\ \\ J_{cf} = \int (x^T Q x + u^T R u) dt \\ \\ u(t) = -K_c x(t) \end{cases} \tag{4.18}$$

Where Q and R are the weighting matrices. Here, the necessary condition for optimality of the time derivative of the Hamiltonian function is:

$$K_c = R^{-1} B^T P \tag{4.19}$$

where P obey to Riccati equation:

$$-PA - A^T P + PBR^{-1} B^T P - Q = \dot{P} \tag{4.20}$$

In order to solve Riccati equation, we first build the Hamiltonian matrix:

$$H = \begin{bmatrix} A & -BR^{-1}B^T \\ -Q & -A^T \end{bmatrix} \tag{4.21}$$

4.5.1 Adaptive Optimal Control

Applying the LQ control requires the system linearization to $\dot{X} = AX + BU$ form. In our specific system, a linearization around an equilibrium point will cause the model to be far from the reality (especially in large orientation angles) as all the couplings are neglected. In order to allow the system optimization for a larger flight envelope, one can linearize around each state. Each coupled term is represented twice by fixing and varying one state at each instant of time. This leads to the following linear state-space system:

$$\dot{X}^T = \begin{bmatrix} \dot{\phi} & \ddot{\phi} & \dot{\theta} & \ddot{\theta} & \dot{\psi} & \ddot{\psi} \end{bmatrix}^T \tag{4.22}$$

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{I_{yy}-I_{zz}}{2I_{xx}}\dot{\psi} & 0 & \frac{I_{yy}-I_{zz}}{2I_{xx}}\dot{\theta} \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & \frac{I_{zz}-I_{xx}}{2I_{yy}}\dot{\psi} & 0 & 0 & 0 & \frac{I_{yz}-I_{xx}}{2I_{yy}}\dot{\phi} \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & \frac{I_{xx}-I_{yy}}{2I_{zz}}\dot{\theta} & 0 & \frac{I_{xx}-I_{yy}}{2I_{zz}}\dot{\phi} & 0 & 0 \end{pmatrix} \tag{4.23}$$

$$B = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{l}{I_{xx}} & 0 & 0 & \frac{Jr}{I_{xx}}\dot{\theta} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{l}{I_{yy}} & 0 & 0 & \frac{Jr}{I_{yy}}\dot{\theta} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{I_{zz}} & 0 \end{pmatrix} \tag{4.24}$$

The matrices A and B are now being adapted through the robot trajectory. The linearization is thus more faithful.

4.5.2 First LQ Controller Synthesis and Simulation

If we consider Pearson method [33], we solve Riccati's equation assuming that we zero the second term of (4.20), solve the equation and get the feedback gain matrix. A first simulation was performed on a model without actuators' dynamics. The results were very satisfactory even if we start from a critical position as $\pi/2$ for the orientation angles. The same simulation including, this time, actuator's model was performed and showed the strong influence of the actuators' dynamics as presented in Fig. 4.6.

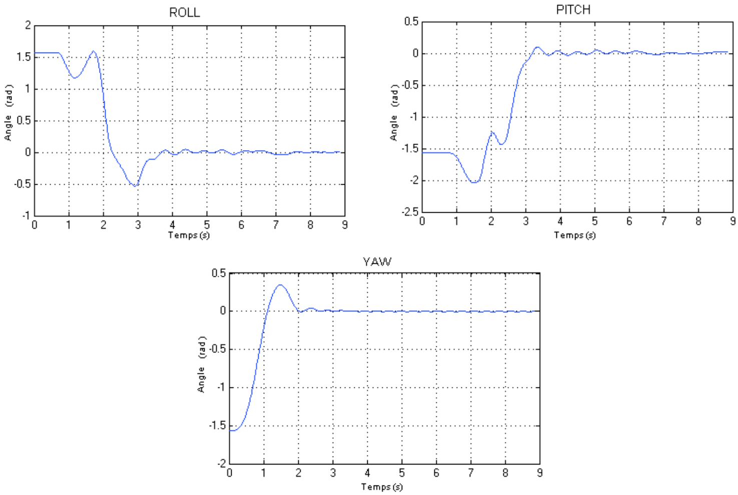


Figure 4.6: Simulation: The system has to stabilize the orientation angles starting from $\pi/2$ with an LQ controller designed using Pearson method.

4.5.3 Second LQ Controller Synthesis and Simulation

Considering a permanent solution to Riccati equation as simulated before gives medium results. Contrarily, Sage-Eisenberg method [33] proposes to consider a variable solution to Riccati equation and a fixed final condition $P(t_f) = 0$. Once discretized, Riccati equation can be rewritten as:

$$-P_t(hA - I) - (hA^T)P_t + P_t(hBR^{-1}B^T)P_t - (hQ + P_{t+h}) = 0 \quad (4.25)$$

Here, t_f is the final time, n the number of iterations and $h = \frac{t_f}{n}$ the iteration period. The equation (4.25) represents correctly the system in the P_t to P_{t+h} interval. The control using this method was simulated at 100 Hz under Simulink (see Fig. 4.7), with the full model including actuators' dynamics, the same Q and R matrix used in Subsection 4.5.2 and by taking $t_f = 0.3$ and $n = 10$. The gain matrix was then:

$$K = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 12.83 & 10.02 & 0 & 0 & 0 & 0 \\ 0 & 0 & 12.83 & 10.02 & 0 & 0 \\ 0 & 0 & 0 & 0 & 12.86 & 10.01 \end{pmatrix} \quad (4.26)$$

Comparing with the previous simulation presented in Subsection 4.5.2, Sage-Eisenberg method gives better results as it optimizes the cost function for every sub-trajectory in the P_t to P_{t+h} interval. According to Bellman principle [34], splitting an optimal trajectory generates several optimal sub-trajectories.

4.5.4 LQ Controller on the Real System

In order to validate the previous simulations, we implemented the controllers on the same 450 MHz PC. It was problematic to find weight matrices which satisfy the control stability. In addition, a slight change in Q or R matrices introduces an important variation of the controller behavior. Hence, by choosing $t_f = 0.05$, $n = 10$ and an appropriate Q and R matrices, the system stabilizes as shown in Fig. 4.8. The gain matrix K is then:

$$K = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1.059 & 0.391 & -0.001 & 0 & 0 & 0.001 \\ 0.0007 & 0 & 1.059 & 0.391 & 0 & -0.0004 \\ 0.005 & 0.002 & -0.0002 & -0.0001 & 0.015 & 0.028 \end{pmatrix} \quad (4.27)$$

As this can be seen from Fig. 4.8, a steady-state error remains on the three orientation angles. This is due to the slight differences of the propulsion

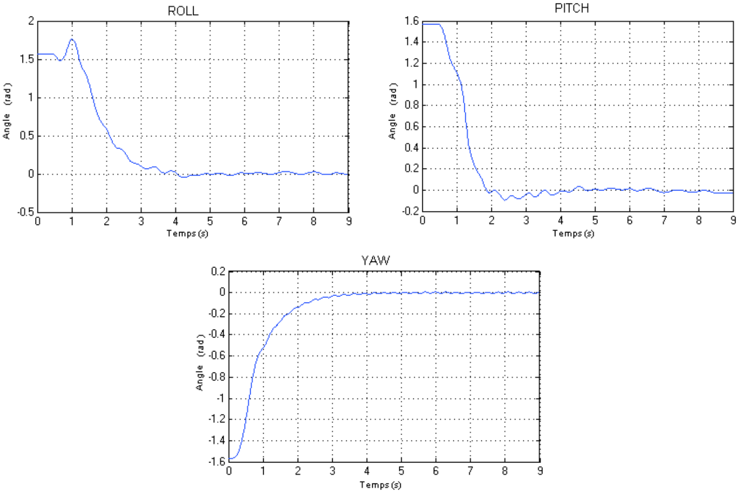


Figure 4.7: Simulation: The system has to stabilize the orientation angles starting from $\pi/2$ with an LQ controller generated using Sage-Eisenberg method.

groups and the disturbance caused by the power and data cables. On the other hand, the fact that the LQ controller was developed without considering actuators' dynamics is also responsible for the average performance. However, one can try to introduce an integral term in an LQ controller as shown in [35].

4.6 Control using Backstepping Technique

Backstepping is the fourth control technique used in this thesis and evaluated on the test-bench. By considering the state (4.2) and the system in (4.8), one can synthesize the control law, forcing the system to follow the desired trajectory [36, 37]. For the first step we consider the tracking error:

$$z_1 = x_{1d} - x_1 \tag{4.28}$$

Then we use the Lyapunov theorem by considering the Lyapunov function z_1 positive definite and its time derivative negative semi-definite:

$$V(z_1) = \frac{1}{2} z_1^2 \tag{4.29}$$

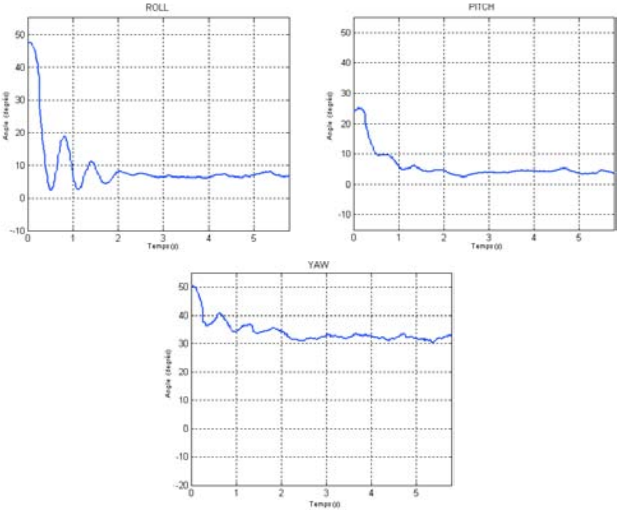


Figure 4.8: Experiment: The system has to stabilize the orientation angles. The experiment was performed with an LQ controller using Sage-Eisenberg method. The control performance is weak, especially in roll and yaw angles.

$$\dot{V}(z_1) = z_1(\dot{x}_{1d} - x_2) \tag{4.30}$$

The stabilization of z_1 is obtained by introducing a virtual control input x_2:

$$x_2 = \dot{x}_{1d} + \alpha_1 z_1 \quad (\alpha_1 > 0) \tag{4.31}$$

The equation (4.30) is then:

$$\dot{V}(z_1) = -\alpha_1 z_1^2 \tag{4.32}$$

Let us proceed to a variable change by making:

$$z_2 = x_2 - \dot{x}_{1d} - \alpha_1 z_1 \tag{4.33}$$

For the second step we consider the augmented Lyapunov function:

$$V(z_1, z_2) = \frac{1}{2}(z_1^2 + z_2^2) \tag{4.34}$$

Its time derivative is then:

$$\dot{V}(z_1, z_2) = z_2(a_1 x_4 x_6 + a_2 x_4 \Omega_r + b_1 U_2) - z_2(\ddot{x}_{1d} - \alpha_1(z_2 + \alpha_1 z_1)) - z_1 z_2 - \alpha_1 z_1^2 \tag{4.35}$$

The control input U_2 is then extracted ($\ddot{x}_{1,2,3d} = 0$), satisfying $\dot{V}(z_1, z_2) < 0$:

$$U_2 = \frac{1}{b_1}(z_1 - a_1 x_4 x_6 - a_2 x_4 \Omega_r - \alpha_1(z_2 + \alpha_1 z_1) - \alpha_2 z_2) \tag{4.36}$$

The term $\alpha_2 z_2$ with $\alpha_2 > 0$ is added to stabilize z_1.
The same steps are followed to extract U_3 and U_4.

$$\begin{cases} U_3 = \frac{1}{b_2}[z_3 - a_3 x_2 x_6 - a_4 x_2 \Omega_r - \alpha_3(z_4 + \alpha_3 z_3) - \alpha_4 z_4] \\ \\ U_4 = \frac{1}{b_3}[z_5 - a_5 x_2 x_4 - \alpha_5(z_6 + \alpha_5 z_5) - \alpha_6 z_6] \end{cases} \tag{4.37}$$

with:

$$\begin{aligned} z_3 &= x_{3d} - x_3 \\ z_4 &= x_4 - \dot{x}_{3d} - \alpha_3 z_3 \\ z_5 &= x_{5d} - x_5 \\ z_6 &= x_6 - \dot{x}_{5d} - \alpha_5 z_5 \end{aligned} \tag{4.38}$$

4.6.1 Backstepping Controller Simulation

Before testing the controller on the real system, we performed different simulations with this six parameters controller ($\alpha_1, ..., \alpha_6$), tuned simultaneously using the Nonlinear Control Design blockset (NCD) from the Optimization Toolbox in Matlab. Since Release 14, this tool became Signal Constraint block and it belongs to Simulink Response Optimization library. The initial condition was $\pi/4\,$rad for the three angles. The results were very satisfactory as shown in Fig. 4.9. The controller's parameters for this simulation are listed in Table 4.1.

Table 4.1: Simulation parameters of Backstepping controller.

Parameter	Value
α_1	10.689
α_2	2.048
α_3	9.535
α_4	3.843
α_5	2.223
α_6	2.177

4.6.2 Backstepping Controller on the Real System

In order to validate the control law developed in the previous section, we implemented the controller in C under Linux on a machine running at 450MHz simulating the future integration of a Single Board Computer. We performed several experiments on the real system, where the task was to control the vehicle orientation as shown in Fig. 4.10. The altitude was then fixed by the test-bench. The controller parameters for this experiment are listed in Table 4.2. The initial condition was about 32 degrees for the roll angle and we obtained the stabilization in less than 5 seconds. It was difficult to give the same initial angular speed to the roll angles on the test-bench as in simulation. In spite of the test-bench limitations in terms of delays and errors introduced by the tethering system, the experimental results obtained show that the proposed controller is able to stabilize the system even for relatively critical initial conditions.

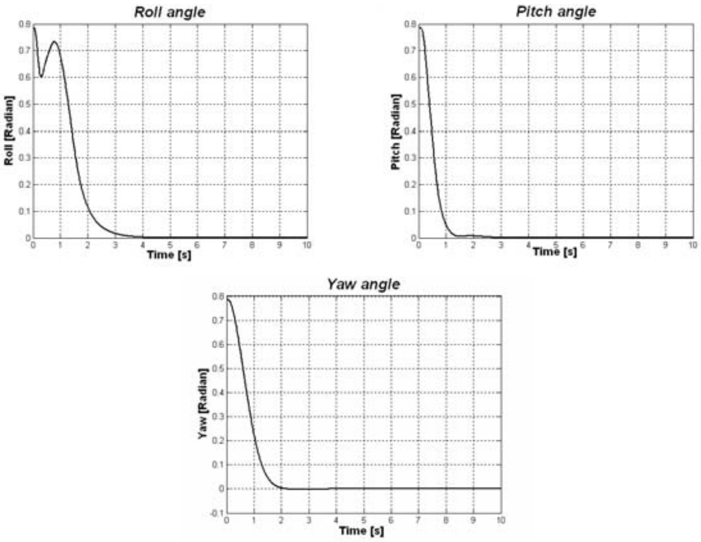


Figure 4.9: Simulation: The backstepping controller has to stabilize the system and maintain the roll, pitch and yaw angles to zero.

Table 4.2: Experimental parameters of Backstepping controller.

Parameter	Value
α_1	12
α_2	3
α_3	10
α_4	3
α_5	2.8
α_6	2.4

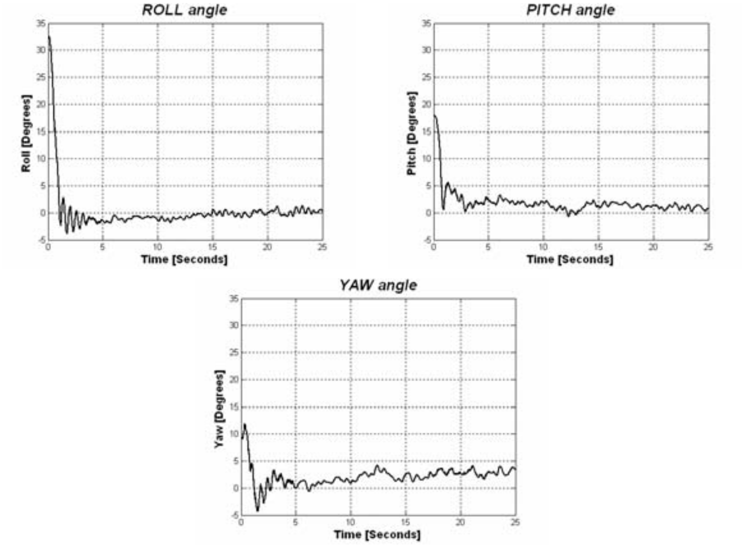


Figure 4.10: Experiment: The backstepping controller has to stabilize the system and maintain the roll, pitch and yaw angles to zero. The helicopter is stabilized very quickly despite the hard initial conditions. A slight drift in yaw angle is observed due to the vibrations and EMI influence on the yaw sensor.

4.7 Control using Sliding-Mode Technique

The mapping (4.8) is used to design the sliding-mode controller for the rotations subsystem of the OS4 helicopter. The first step in this design is similar to the one for the backstepping approach [38], except for the equation (4.31) were S_2 (Surface) is used instead of z_2 for more clearance:

$$s_2 = x_2 - \dot{x}_{1d} - \alpha_1 z_1 \tag{4.39}$$

For the second step, we consider the augmented Lyapunov function:

$$V(z_1, s_2) = \frac{1}{2}(z_1^2 + s_2^2) \tag{4.40}$$

The chosen law for the attractive surface is the time derivative of (4.39) satisfying ($s\dot{s} < 0$):

$$\begin{aligned} \dot{s}_2 &= -k_1 sign(s_2) - k_2 s_2 \\ &= \dot{x}_2 - \ddot{x}_{1d} - \alpha_1 \dot{z}_1 \\ &= a_1 x_4 x_6 + a_2 x_4 \Omega_r + b_1 U_2 - \ddot{x}_{1d} + \alpha_1 (z_2 + \alpha_1 z_1) \end{aligned} \tag{4.41}$$

With: $k_1, k_2 > 0$. The symbol *sign* stands for signum function. As for the backstepping approach, the control U_2 is extracted:

$$U_2 = \frac{1}{b_1}(-a_1 x_4 x_6 - a_2 x_4 \Omega_r - \alpha_1^2 z_1 - k_1 sign(s_2) - k_2 s_2) \tag{4.42}$$

The same steps are followed to extract U_3 and U_4:

$$\begin{cases} U_3 = \frac{1}{b_2}[a_3 x_2 x_6 - a_4 x_2 \Omega_r - \alpha_2^2 z_3 - k_3 sign(s_3) - k_4 s_3] \\ \\ U_4 = \frac{1}{b_3}[-a_5 x_2 x_4 - \alpha_3^2 z_5 - k_5 sign(s_4) - k_6 s_4] \end{cases} \tag{4.43}$$

with:

$$\begin{aligned} &z_3 = x_{3d} - x_3 \\ &s_3 = x_4 - \dot{x}_{3d} - \alpha_2 z_3 \\ &z_5 = x_{5d} - x_5 \\ &s_4 = x_6 - \dot{x}_{5d} - \alpha_3 z_5 \end{aligned} \tag{4.44}$$

4.7.1 Sliding-mode Controller Simulation

For these simulations we considered only the angular rotations subsystem in order to be able to verify the development on the real system. The controller above contains nine parameters $(\alpha_1, ..., \alpha_3, k_1, ..., k_6)$ tuned also using NCD and listed in Table 4.3. The initial conditions were $\pi/4$ rad for the three angles as shown in Fig. 4.11.

Table 4.3: Simulation parameters of Sliding-mode controller.

Parameter	Value
α_1	2.306
α_2	0.088
α_3	1.623
k_1	2.146
k_2	6.202
k_3	1.299
k_4	6.945
k_5	0.901
k_6	3.238

4.7.2 Sliding-mode Controller on the Real System

The experimental conditions were similar to the ones applied for the backstepping controller, see Fig. 4.12. The task was to control the vehicle orientation, and the altitude was fixed by the test-bench. The controller parameters for this experiment are listed in Table 4.4. The initial condition was about 26 degrees for the roll angle and we obtained the stabilization in less than 8 seconds. The experimental results obtained show that the proposed controller is able to stabilize the roll and pitch angles, but the shattering effect, even if reduced, disturbs the measurements and this is visible especially for the yaw angle.

4.8 First Autonomous Flight

After all the simulations and experiments performed on the test-bench with the control techniques presented before, it was time to test an autonomous flight. However, this was not possible with all the techniques presented before.

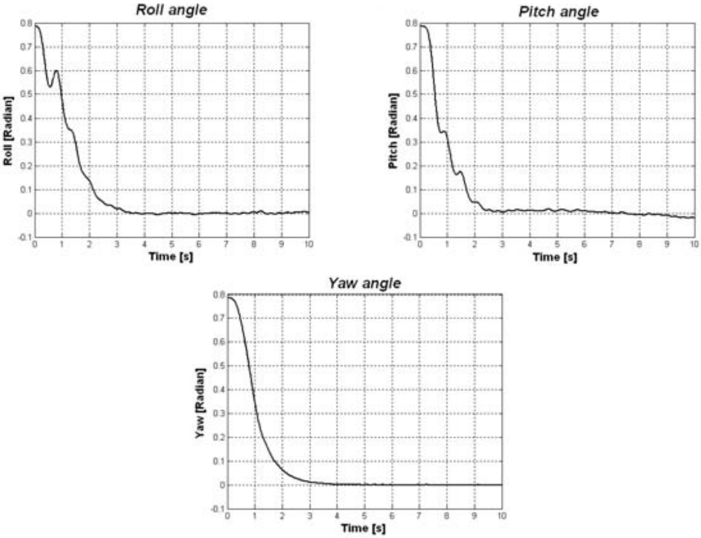


Figure 4.11: Simulation: The sliding-mode controller has to stabilize the system and maintain the roll, pitch and yaw angles to zero.

Table 4.4: Experimental parameters of Sliding-mode controller.

Parameter	Value
α_1	15
α_2	15
α_3	2
k_1	15
k_2	8
k_3	15
k_4	8
k_5	2
k_6	2

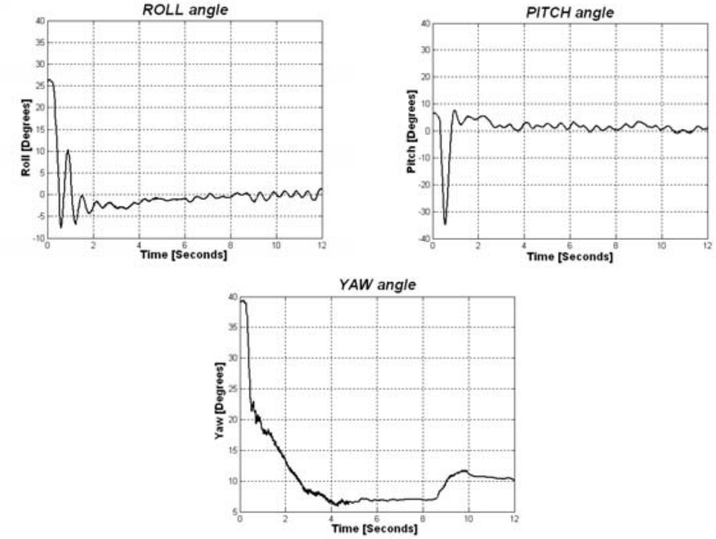


Figure 4.12: Experiment: The Sliding-mode controller has to stabilize the system and maintain attitude angles to zero. The controller stabilizes correctly roll and pitch angles, but the shattering effect is visible on them. However, yaw angle is weakly controlled. The big negative overshot in the pitch angle is due to the huge initial condition on the yaw angle and the roll.

The Lyapunov controller developed in section 4.3, proved to be very reactive. As one can see in Fig.4.3, the yaw angle was very well stabilized despite the large initial condition. However, at this stage of the project, it was not possible to release the flying part from the test-bench. Even though the rejection of disturbances was good, the stabilization in the direct neighborhood of the equilibrium point was not rigid enough. This caused the flying part to slide away when released.

The PID controller with its simple implementation (see Section 4.4) proved to be well adapted to the quadrotor when flying near hover. It was possible using this technique to successfully perform the first autonomous flight. We released the flying part as shown in Fig. 4.14, the behavior of the orientation angles is presented in Fig. 4.13. Some perturbations were introduced by the power cables and by us while trying to prevent the helicopter from colliding with the walls. Despite this success, the PID controller was only able to control the quadrotor in near hover in absence of large disturbances.

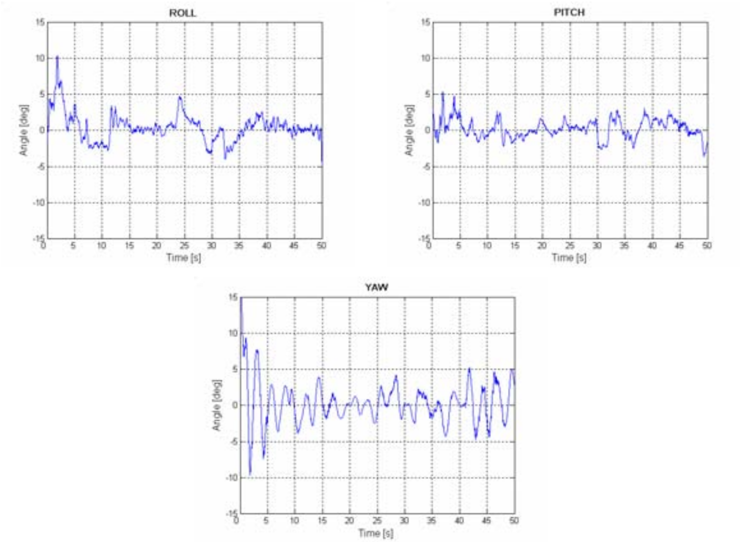


Figure 4.13: Experiment: First successful autonomous flight. The controller stabilizes the orientation angles with a PID controller.

The LQ controller shown in Section 4.5 displayed average stabilization results. In fact, a steady-state error remained because of unmodeled effects

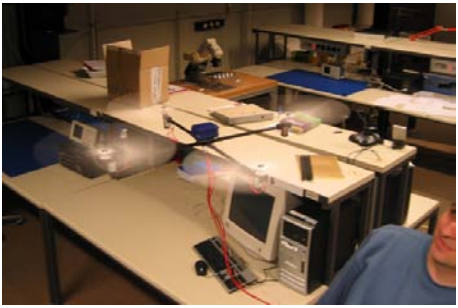


Figure 4.14: The flying part of the test-bench in a tethered flight.

and the systematic slight differences in the propulsion groups. In addition, the LQ controller we obtained showed to be less dynamic than the PID. Thus, we were not able to release OS4 for a free flight. However, another research group succeeded in flying a quadrotor using an LQ controller, refer to [30] for more details. This proves that the optimal control theory should give better results when applied to a quadrotor.

The Backstepping based controller developed in Section 4.6, proved its ability to control the orientation angles in presence of relatively high perturbations. This confirms the results of a previous study on underactuated systems [39]. However, 50% of the attempts to fly freely with this technique failed. This was obviously not acceptable. Anyhow, the impressive and elegant stabilization of large disturbances and strong initial conditions makes this approach very interesting.

The sliding-mode technique (see Section 4.7) did not provide excellent results. This is partly due to the switching nature of the controller which seems to be ill adapted to the dynamics of the quadrotor. Another similar project used this technique to control the altitude of a quadrotor as shown in this paper [40].

4.9 The Proposed Control Approach

After the evaluation of all the control approaches tested in this work, it became clear that the way to follow is a combination between PID and Backstepping into the so-called Integral Backstepping. The goal was to bring together the robustness against disturbances offered by Backstepping and robustness against model uncertainties offered by the integral action as schematized in Fig. 4.15. This shall permit more complex flight maneuvers than a simple hovering. After a phase of extensive simulation and experimentation, Integral Backstepping was proposed as a single approach for attitude, altitude and position control. This process is presented in the next section.

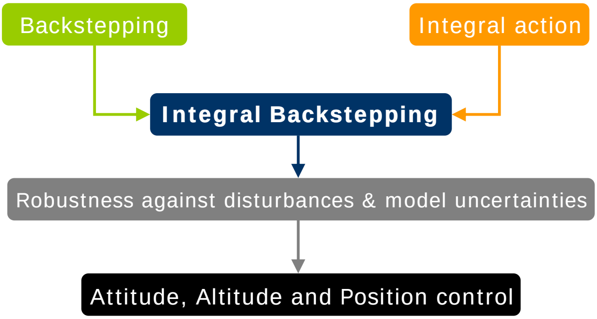

Backstepping
Integral action
Integral Backstepping
Robustness against disturbances & model uncertainties
Attitude, Altitude and Position control

Figure 4.15: The proposed control approach.

4.10 Control using Integral Backstepping

Another improvement is now introduced thanks to IB [5]. The idea of using integral action in the backstepping design was first proposed in [41] and applied in [42] from which this control design was derived. Thanks to this technique, OS4 is able to perform autonomous hovering with altitude control and autonomous take-off and landing. The augmentation of the Backstepping controller with integral action affects slightly the control law design. Moreover, as the derivation is similar for attitude, altitude and position controllers, only roll angle controller derivation will be presented. The OS4 control system is structured in six different controllers as illustrated in Fig. 4.16. Take-off and landing controller outputs the desired altitude (z_d) to altitude controller which outputs the desired overall thrust (T_d) based on sonar data. Position controller receives OS4 position (x, y) and desired thrust, it outputs desired roll (ϕ_d) and pitch (θ_d) while desired yaw (ψ_d) comes directly from the user. Attitude controller outputs then the desired motor speed to the motor controllers. Integral backstepping technique is used for attitude, altitude and position control. This permits a powerful and flexible control structure.

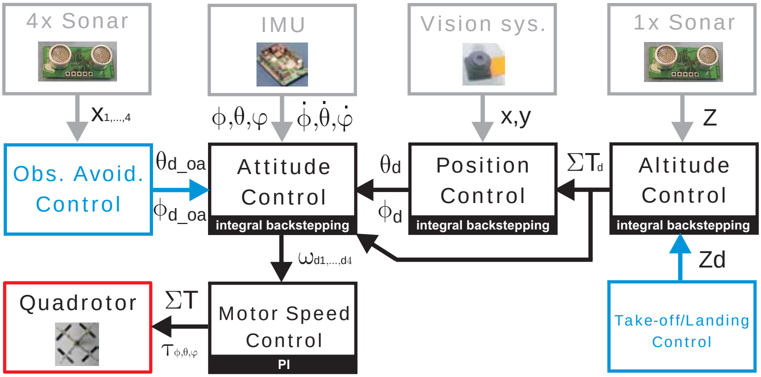


Figure 4.16: The control structure implemented on OS4.

4.10.1 Attitude Control

Attitude control is the heart of the control system, it keeps the 3D orientation of the helicopter to the desired value. Usually roll and pitch angles are forced to zero which permits hovering flight. Attitude control loop runs at 76 Hz which is the update rate of the IMU (Microstrain 3DM-GX1). The latter provides the rates of turn and orientations around (x, y, z) axes with an accuracy of ±2° in dynamic. The first step in IB control design is to consider the tracking-error $e_1 = \phi_d - \phi$ and its dynamics:

$$\frac{de_1}{dt} = \dot\phi_d - \omega_x \tag{4.45}$$

The angular speed ω_x is not our control input and has its own dynamics. So, we set for it a desired behavior and we consider it as our virtual control:

$$\omega_{xd} = c_1 e_1 + \dot\phi_d + \lambda_1 \chi_1 \tag{4.46}$$

with c_1 and λ_1 positive constants and $\chi_1 = \int_0^t e_1(\tau)d_\tau$ the integral of roll tracking error. So, the integral term is now introduced in (4.46).
Since ω_x has its own error e_2, we compute its dynamics using (4.46) as follows:

$$\frac{de_2}{dt} = c_1(\dot\phi_d - \omega_x) + \ddot\phi_d + \lambda_1 e_1 - \ddot\phi \tag{4.47}$$

where e_2, the angular velocity tracking error is defined by:

$$e_2 = \omega_{xd} - \omega_x \tag{4.48}$$

Using (4.46) and (4.48) we rewrite roll tracking error dynamics as:

$$\frac{de_1}{dt} = -c_1 e_1 - \lambda\chi_1 + e_2 \tag{4.49}$$

By replacing $\ddot{\phi}$ in (4.47) by its corresponding expression from model (4.5), the control input U_2 appears in (4.50):

$$\frac{de_2}{dt} = c_1(\dot{\phi}_d - \omega_x) + \ddot{\phi}_d + \lambda_1 e_1 - \dot{\theta}\dot{\psi}a_1 - \dot{\theta}a_2\Omega_r - b_1 U_2 \tag{4.50}$$

The real control input has appeared in (4.50). So, using equations (4.45), (4.49) and (4.50) we combine the tracking errors of the position e_1, of the angular speed e_2 and of the integral position tracking error χ_1 to obtain:

$$\frac{de_2}{dt} = c_1(-c_1 e_1 - \lambda_1\chi_1 + e_2) + \ddot{\phi}_d + \lambda_1 e_1 - \tau_x/I_{xx} \tag{4.51}$$

where τ_x is the overall rolling torque. The desirable dynamics for the angular speed tracking error is:

$$\frac{de_2}{dt} = -c_2 e_2 - e_1 \tag{4.52}$$

This is obtained if one chooses the control input U_2 as:

$$U_2 = \frac{1}{b_1}[(1 - c_1^2 + \lambda_1)e_1 + (c_1 + c_2)e_2 - c_1\lambda_1\chi_1 + \ddot{\phi}_d - \dot{\theta}\dot{\psi}a_1 - \dot{\theta}a_2\Omega_r] \tag{4.53}$$

where c_2 is a positive constant which determines the convergence speed of the angular speed loop. Similarly, pitch and yaw controls are:

$$\begin{cases} U_3 = \frac{1}{b_2}[(1 - c_3^2 + \lambda_2)e_3 + (c_3 + c_4)e_4 - c_3\lambda_2\chi_2 + \ddot{\theta}_d - \dot{\phi}\dot{\psi}a_3 + \dot{\phi}a_4\Omega_r] \\ \\ U_4 = \frac{1}{b_3}[(1 - c_5^2 + \lambda_3)e_5 + (c_5 + c_6)e_6 - c_5\lambda_3\chi_3] \end{cases} \tag{4.54}$$

with $(c_3, c_4, c_5, c_6, \lambda_2, \lambda_3) > 0$, and (χ_2, χ_3) the integral position tracking error of pitch and yaw angles respectively.

Stability Analysis

Stability analysis is performed using Lyapunov theory. The following candidate Lyapunov function is chosen:

$$V = \lambda\frac{1}{2}[\chi_1^2 + e_1^2 + e_2^2] \tag{4.55}$$

It includes the position tracking error e_1, its integration χ_1 and velocity tracking error e_2. Deriving (4.55) and using equations (4.49) and (4.52) gives:

$$\dot{V} = -c_1 e_1^2 - c_2 e_2^2 \leqslant 0 \tag{4.56}$$

The definition of (4.55) and the fact that $\dot{V} \leqslant 0, \forall(e_1, e_2)$ guarantees the boundedness of e_1, χ_1 and e_2. The desired position reference ϕ_d is bounded by assumption and $e_1 = \phi_d - \phi$ is also bounded, so, position ϕ is also bounded. This implies the boundedness of the virtual control ω_x. Finally, the boundedness of the overall control torque is due to our choice of the control law in (4.53). Global Asymptotic Stability is also ensured from the positive definition of V and the fact that $\dot{V}(e_1, e_2) < 0$, $\forall(e_1, e_2) \neq 0$ and $\dot{V}(0) = 0$ and by applying LaSalle theorem. In fact, the maximum invariance set of angular rotations subsystem under control (4.53) and (4.54) is restricted only to the equilibrium point.

Results

Attitude control performance is of crucial importance, it is directly linked to the performance of the actuators. OS4 is equipped with brushless sensorless motors powerful enough to avoid amplitude saturation. However, they suffer from low dynamics and thus from bandwidth saturation. This was taken into account in control design. Simulation results shown in Fig. 4.17 are performed with a model which includes actuators' dynamics and amplitude saturation. The simulation takes into account the delay and the noise inherent to sensors. The task was to stabilize roll, pitch and yaw angles and maintain them at zero. Control parameters were in the simulation $C_1 = 10$, $C_2 = 2$, $C_3 = 10$, $C_4 = 2$, $C_5 = 2$, $C_6 = 2$. The experiment shown in Fig. 4.19 is a free flight were attitude references are zero. One can see in Fig. 4.18 that roll and pitch plots show a bounded oscillation of $0.1\,rad$ in amplitude. This oscillation is not perceptible in flight, nevertheless it is due to the slow dynamics of OS4's actuators coupled with the differences between the four propulsion groups. Control parameters were in this experiment $C_1 = 10.5$, $C_2 = 2$, $C_3 = 10$, $C_4 = 2$, $C_5 = 2$, $C_6 = 2$. These are really close to the parameters used in simulation which highlights the quality of the model.

4.10.2 Altitude Control

The altitude controller keeps the distance of the helicopter to the ground at a desired value. It is based on a sonar (Devantech SRF10) which gives the range to the closest obstacle at 15 Hz. The accuracy depends on the

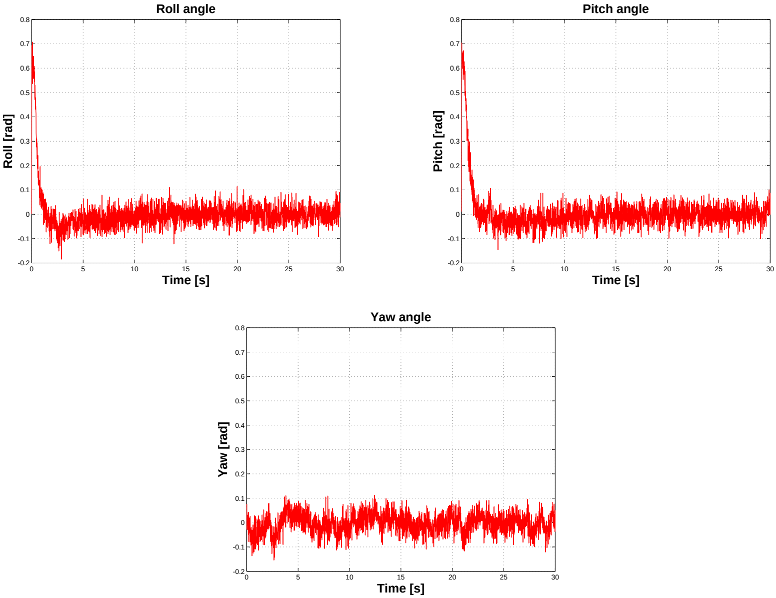


Figure 4.17: Simulation: Integral backstepping attitude controller has to maintain roll, pitch and yaw angles to zero. Despite of the hard initial conditions and the white noise, the helicopter is quickly brought back to equilibrium.

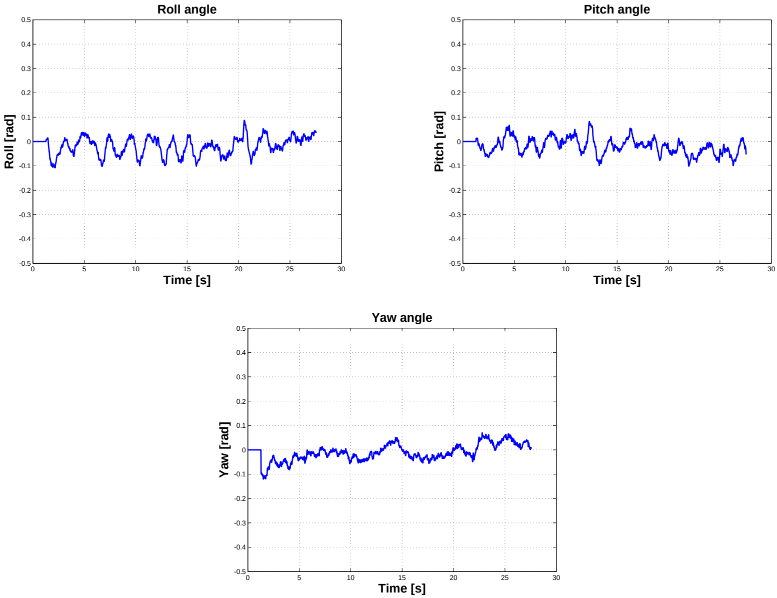


Figure 4.18: Experiment: Integral backstepping attitude controller has to maintain attitude angles to zero in flight. The helicopter is stabilized despite the numerous disturbances due to yaw drift, sensors noise and unmodeled effects.

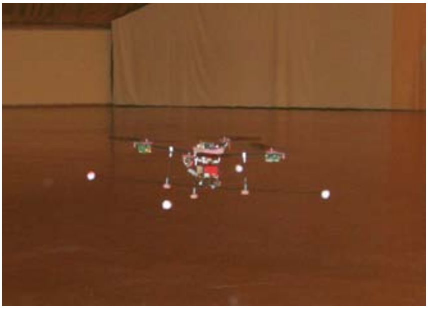


Figure 4.19: OS4 in hover. A training frame was added for safety.

distance, it is about 1 to 2 cm at 1 m. The necessary altitude rate of change is estimated based on the range. On the control law side, altitude tracking error is defined as:

$$e_7 = z_d - z \tag{4.57}$$

The speed tracking error is:

$$e_8 = c_7 e_7 + \dot{z}_d + \lambda_4 \chi_4 - \dot{z} \tag{4.58}$$

The control law is then:

$$U_1 = \frac{m}{cos\phi cos\theta}[g + (1 - c_7^2 + \lambda_4)e_7 + (c_7 + c_8)e_8 - c_7\lambda_4\chi_4] \tag{4.59}$$

where (c_7, c_8, λ_4) are positive constants.

Take-off and Landing

The autonomous take-off and landing algorithm adapts the altitude reference z_d to follow the dynamics of the quadrotor for taking-off or landing. One can see in Fig. 4.20 that the desired altitude reference is gradually reduced by a fixed step k $(k > 0)$ which depends on the vehicle dynamics and the desired landing speed. Moreover, the fact that the control loop is much faster than the vehicle dynamics, makes the landing very smooth. Ground effect was not implemented because the landing skids are long enough to keep the propellers out of ground effect even after touch-down.

Results

Altitude control works surprisingly well despite all the limitations of the sonar. Figure 4.21 shows an altitude reference profile (green) followed by the simulated controller (red) and the real controller (blue). The task was to climb to $0.5\,m$, hover and then land. Control parameters where $C_7 = 3.5$, $C_8 = 1.5$ in simulation and $C_7 = 4$, $C_8 = 2$ in experiment. The slight deviation between simulation and reality in take-off and landing phases is inherited from actuators' dynamics where the model was slightly slower in the raising edge, and slightly faster in the falling one. Take-off is performed in 2 s (0-0.5 m) and landing in 2.8 s (0.5-0 m). Altitude control has a maximum of 3 cm deviation from the reference.

4.10.3 Position Control

Position control keeps the helicopter over the desired point. It is meant here the (x, y) horizontal position with regard to a starting point. Horizontal motion is achieved by orienting the thrust vector towards the desired direction of

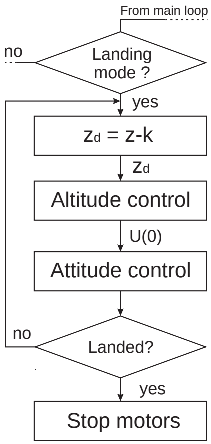


Figure 4.20: Autonomous landing flowchart. Altitude reference is gradually reduced taking into account the dynamics of the robot.

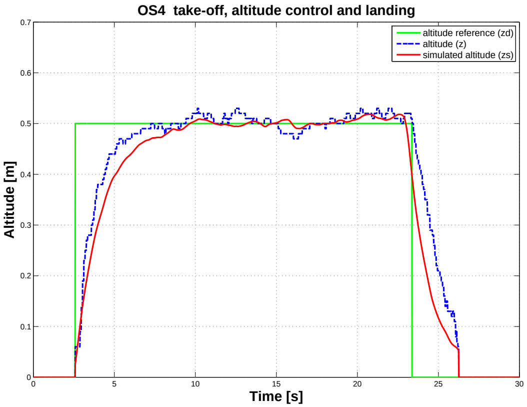


Figure 4.21: Autonomous take-off, altitude control and landing in simulation and in real flight.

motion. This is done by rotating the vehicle itself in the case of a quadrotor. In practice, one performs position control by rolling or pitching the helicopter in response to a deviation from the y_d or x_d references respectively. Thus, the position controller outputs the attitude references ϕ_d and θ_d, which are tracked by the attitude controller (see Fig. 4.16). The thrust vector orientation in the earth fixed frame is given by R, the rotation matrix. Applying small angle approximation to R gives:

$$R = \begin{bmatrix} 1 & \psi & \theta \\ \psi & 1 & -\phi \\ -\theta & \phi & 1 \end{bmatrix} \tag{4.60}$$

From (4.5) and using (4.60) one can simplify horizontal motion dynamics to

$$\begin{bmatrix} m\ddot{x} \\ m\ddot{y} \end{bmatrix} = \begin{bmatrix} -\theta U_1 \\ \phi U_1 \end{bmatrix} \tag{4.61}$$

The control law is then derived using IB technique. Position tracking errors for x and y are defined as:

$$\begin{cases} e_9 = x_d - x \\ e_{11} = y_d - y \end{cases} \tag{4.62}$$

Accordingly speed tracking errors are:

$$\begin{cases} e_{10} = c_9 e_9 + \dot{x}_d + \lambda_5 \chi_5 - \dot{x} \\ e_{12} = c_{11} e_{11} + \dot{y}_d + \lambda_6 \chi_6 - \dot{y} \end{cases} \tag{4.63}$$

The control laws are then:

$$\begin{cases} U_x = \frac{m}{U_1}[(1 - c_9^2 + \lambda_5)e_9 + (c_9 + c_{10})e_{10} - c_9 \lambda_5 \chi_5] \\ \\ U_y = -\frac{m}{U_1}[(1 - c_{11}^2 + \lambda_6)e_{11} + (c_{11} + c_{12})e_{12} - c_{11} \lambda_6 \chi_6] \end{cases} \tag{4.64}$$

where $(c_9, c_{10}, c_{11}, c_{12}, \lambda_5, \lambda_6)$ are positive constants.

Results

The main result in position control was obtained in simulation. Fig. 4.16 shows how the different controllers are cascaded. In fact, only the attitude is driven by the position, altitude controller is simply feeding them with U_1. Attitude and position loops run at 76 Hz and 25 Hz respectively. This spectral separation is necessary to avoid a conflict between the two loops; it is often accompanied with gain reductions in the driving loop. Control parameters were $C_9 = 2, C_{10} = 0.5, C_{11} = 2, C_{12} = 0.5$ in the simulation of Fig. 4.22.

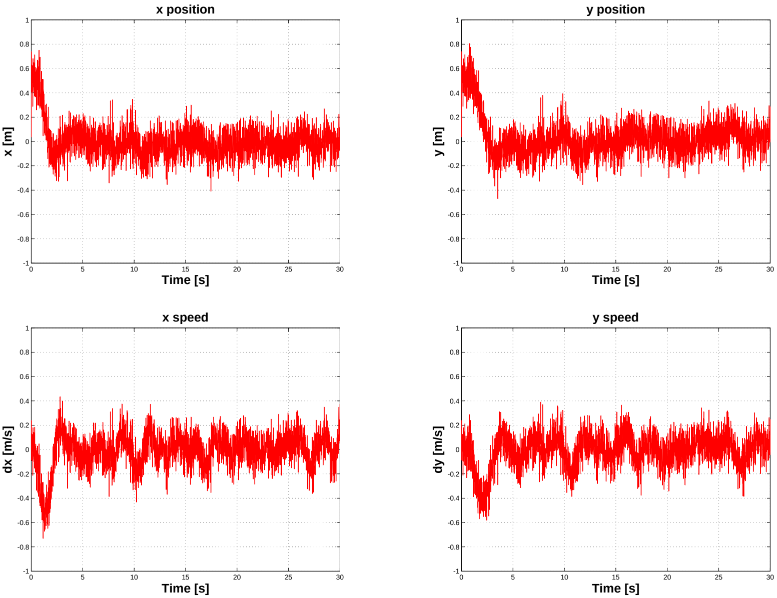


Figure 4.22: Simulation: Integral backstepping position controller drives attitude controller in order to maintain the helicopter over a given point.

Way-points following

The planner block in Fig. 2.5 defines the way-points and hence the trajectories OS4 has to follow. The position of the next way-point is sent to position controller which directs the vehicle towards the goal. A way-point is declared reached when the helicopter enters a sphere around this point. The radius of this sphere (0.1 m) is the maximum admitted error. Figure 4.23 shows a square trajectory defined by four way-points. The task was to climb to 1 m from the ground and then follow the four way-points of a square of 2 m side. In order to track the square trajectory, the planner generates the (x_d, y_d) position references, and consequently the position controller generates the (ϕ_d, θ_d) attitude references. Figure 4.24 depicts these signals and shows that the 2 m side square is tracked with about 10% overshoot (20 cm), while the trajectory is completed in 20 s.

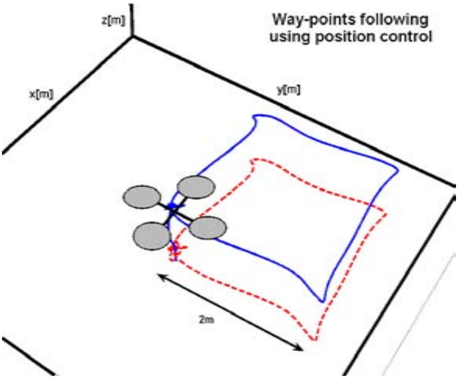


Figure 4.23: Four way-points for a square trajectory tracked by OS4.

4.10.4 Obstacle Avoidance

OS4 is equipped with a sonar-based obstacle avoidance system composed of four miniature ultrasound range finders in cross configuration. First of all, we introduced the obstacle avoidance controller into the Simulink model and inserted the environment and sensor libraries. Aiming to simplify the procedure, we decided to keep the altitude constant during evasive maneuvers. This would reduce the path planning complexity to a 2D problem. We also restricted its direction of flight: OS4 can move only on the four directions where the US sensors were placed. To increase the flight safety, a 90 cm radius security zone is constantly maintained between the helicopter and the

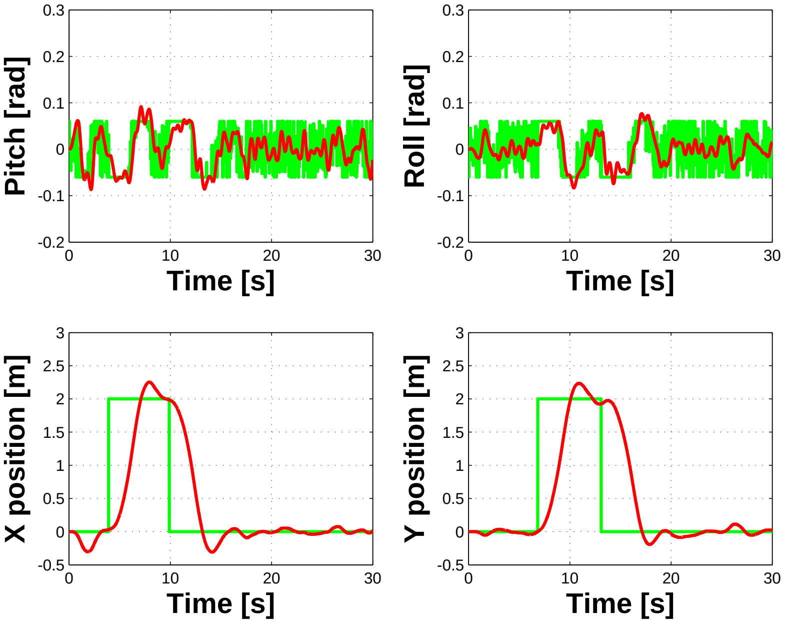


Figure 4.24: Simulation: The position and attitude signals generated to track the square trajectory.

environment (see Fig. 4.25). This zone assures a 50 cm distance between the helicopter rotors and any obstacle. If an obstacle is detected inside the security zone, a safety loop (that runs in parallel to the OAC) interferes in the helicopter flight control and generates an evasive maneuver. This maneuver is obtained by selecting a predefined pitch and/or roll angles that would avoid a collision between the helicopter and the obstacles. Several algorithms were simulated (see Appendix A) with various results in a 100 m^2 environment with obstacles represented as columns of 20 cm in diameter and 3 meters in height. The developed approaches can be divided into two categories: Relative position and speed-based approaches. The first OAC uses position controller to act on the relative position of the helicopter wrt. the closest obstacle (X_{oa}, Y_{oa}). The second one uses speed controller to act on the speed of the vehicle $(\dot{X}, \dot{Y})$ if an obstacle is detected. The latter approach was used to develop our main OAC algorithm. The idea was to act on $\dot{X}$ and $\dot{Y}$ while keeping the heading and altitude constants. When an obstacle is detected its distance to the helicopter is classified based on a given threshold as "far", "close" or "too close". If the obstacle is "far", no avoidance action is needed and the OAC does not interfere with the helicopter normal flight. On the other hand, if the obstacle distance is "close" or "too close" the OAC informs the helicopter flight control, reduces its speed, and generates evasive maneuvers using predefined flight directions, this is shown in Fig. 4.26. The selection of the direction of the evasive flight depends on the stimulated sensor and the desired flight direction previously selected by the user. However, if the quadrotor is surrounded by obstacles that are "too close", it reduces the speed and keeps a hovering behavior. The lack of precise sensors for linear speed made the implementation of this approach difficult. A simple collision avoidance algorithm was then developed. The idea was to avoid collision with walls or persons present in the flight area. The inherent noise of the sonar especially in absence of obstacles was threatening OS4 stability. This

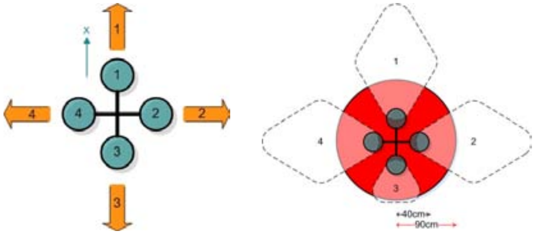


Figure 4.25: The 4 flight directions (left) and the security zone (right).

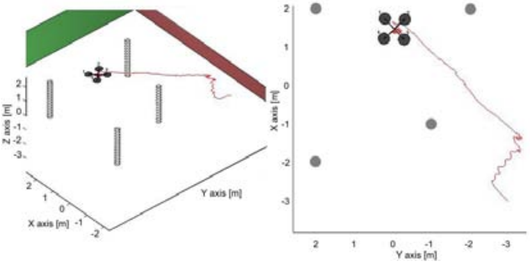


Figure 4.26: Simulation: OS4 avoiding static obstacles.

is mainly due to the interferences between the five sonar and the effect of the propellers on the ultrasound waves. Figure 4.27 shows a detection of an obstacle with and without the filter. The latter is based on the variation of successive samples and gives a reliable detection signal usable in flight.

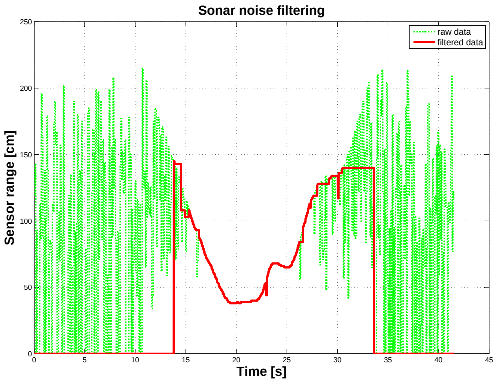


Figure 4.27: Obstacle detection with and without the filter.

Results

A collision avoidance behavior was practically obtained after numerous tests and tuning. Once the obstacle is detected, a pitch reference is given to fly away the helicopter from the obstacle. Figure 4.28 shows the reaction of OS4 to an obstacle at 40 cm, one can see the distance to the obstacle increasing until the latter disappears, then OS4 recovers a normal flight.

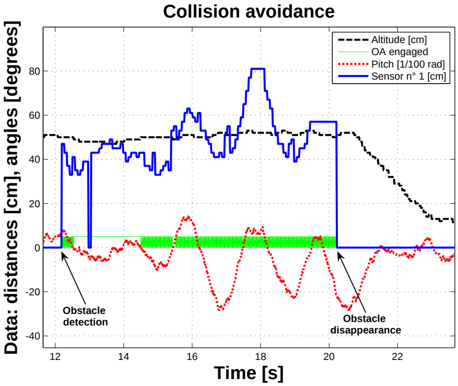


Figure 4.28: Experiment: Collision avoidance with OS4. The helicopter flies back until the obstacle disappears.

4.11 Conclusion

This chapter presented the simulation and the testing of five control techniques for the attitude control of a quadrotor. The first technique is based on Lyapunov theory, it proved to be very reactive, especially for the yaw angle control. However, the stabilization in the direct neighborhood of the equilibrium point was not rigid enough to permit hover flight. The second one is a PID controller, it proved to be well adapted to the quadrotor when flying near hover. It was possible using this technique to successfully perform the first autonomous flight. The PID controller was only able to control the quadrotor in near hover and absence of large disturbances. The third one is an LQ controller, it displayed average stabilization results. It showed to be less dynamic than the PID. The fourth control technique is the Backstepping, its ability to control the orientation angles in presence of relatively high perturbations is very interesting. The sliding-mode technique is the fifth approach, it did not provide excellent results. The switching nature of the controller seems to be ill adapted to the dynamics of the quadrotor. The results of all these control approaches conducted to a combination of PID and Backstepping into the so-called Integral Backstepping. This was proposed as a single tool to design attitude, altitude and position controllers. The experiment has shown that OS4 is currently able to take-off, hover, land and avoid collisions automatically. As far as we know, OS4 is the first quadrotor practically capable of a collision avoidance maneuver.

Chapter 5

General Conclusion

> *I have discovered that a screw-shaped device such as this, if it is well made from starched linen, will rise in the air if turned quickly.*
>
> Leonardo Da Vinci

This chapter is divided into a review of the main contributions of this thesis, followed by an outlook on future work, which takes a longer perspective on miniature aerial robots. It tries to identify possible future steps towards fully autonomous miniature flying robots.

5.1 Review

The explosion of the market for handheld devices a few years ago accelerated incredibly the development of technologies, not only useful for mobile phones or digital cameras but also determinant for micro aerial vehicles.

This project was born in this context, on the conviction that the development of miniature flying robots requires the simultaneous consideration of the system level optimization along with control design.

Understanding the vehicle's dynamics has been a constant concern, it has been investigated through system modelling, simulation and analysis. It represents the essential insight not only for an elaborated system design, but also for the selection of control approaches adapted to quadrotors.

A two-step approach for the system design was adopted throughout this work. A first platform was designed to allow easy experimentation and control validation. Then, following a new design methodology, a second platform was developed to permit free flight experiments, based on the control algorithms previously matured.
The investigation on control approaches was conducted along progressively increased complexity. Each approach was studied through controller synthesis, simulation and experimentation on the first platform. The proposed control approach was extensively simulated and fully implemented on the second platform.

5.1.1 Modelling

The whole dynamic model was built on basic physics and aerodynamics equations, and a faithful CAD model allowed easy extraction of the physical parameters. In addition, rotor dynamics was identified in order to accurately grasp the dynamics of the brushless motor, its power electronics, the gearbox and the propeller all together at once. The implementation of an aerodynamics block allowed the consideration of variable aerodynamic coefficients that were validated in hover. The result is a set of equations describing the vehicle dynamics not only in hover, but also in motion. A simulator was developed based on this model and is presently used in other quadrotor projects. The final experiments were all performed with strictly the same parameters found by simulation.

5.1.2 Design

The methodology developed and followed in the design of the OS4 robot appreciably facilitated the components selection process and battery dimensioning. Several requirements like target size and weight or target thrust/weigth ratio were considered. The challenges encountered in OS4 design included: the motor selection, electronics integration, structure design and design variables optimization. The actuators' saturation issue was alleviated by using brushless technology and the motor was selected based on considerations like the energy cost of the lifted mass or the quality of this lifting in terms of current losses, bandwidth available, etc. The electronics integration issue was addressed by developing all OS4's electronic modules from scratch. Special attention was given to the motherboard, a six-layer printed circuit board with all the electronics for a flying robot. Structure design issue was addressed by using a combination of carbon fibers (arms), lightweight polyurethane (molded parts) and aluminium (critical parts). Design variables optimization was carried out with the design method especially developed for miniature rotorcraft. As far as we know, OS4 is one of the most integrated quadrotors ever designed.

5.1.3 Control

An important part of this thesis was dedicated to finding a good control approach for quadrotors. Five techniques were explored from theoretical development to final experiments. The difficulties encountered in OS4 control included: Sensor quality, yaw drift, robustness against large disturbances and model uncertainties. Sensor noise is inherent to micro IMUs and is dramatically amplified on helicopters. This degrades sensor accuracy and accelerates drift. Yaw drift is one of the most annoying issues as the contribution of yaw control in the overall control is important. The best robustness against large disturbances was achieved using backstepping technique, while model uncertainties were cancelled thanks to integral action. Thus, integral backstepping has been proposed for full control of quadrotors. Thanks to this technique, OS4 has been able to perform autonomous hovering with altitude control and autonomous take-off and landing.

5.1.4 Originalities

In summary, the originality of this work lie in:

Simultaneous Consideration of The Design and Control Problems

This thesis tackles simultaneously the design and control problems. This makes it possible to simplify the control by directly acting on the design, which generally differs from other projects where commercially available platforms are used.

Easy Approach for Modelling Unstable Systems

The quadrotor model developed in this thesis is composed of physics and aerodynamics equations and identified model of the actuators. All the model parameters are easily extracted from a faithful CAD model. This makes it possible to avoid closed loop identification and thus, simplifies greatly the model development.

Single Tool for All Simulations

OS4 simulator handles all the necessary simulations from aerodynamics up to obstacle avoidance. Thanks to its flexible structure, one can remove, add or enhance any part if needed. This makes the simulation more convenient and gives a global view of the system behavior at once.

New Methodology for Small-scale Rotorcraft Design

This thesis introduced a new methodology for small-scale rotorcraft design and optimization. This methodology helps for tackling the design problematic in a systematic way. As far as we know, this is the first work which tackles the design problem using a clear methodology.

Powerful Brushless Sensorless Out-runner Motors

OS4 is probably the first quadrotor equipped with brushless sensorless out-runner motors. This technology brings very high power to mass ratio even with small motors. The result is a low contribution of the motors mass to the total mass of the helicopter.

Single Technique for Full Control of a Quadrotor

This thesis proposes a single control technique for full control of a quadrotor. This makes it an original approach as all the other systems combine several control techniques. The single technique approach brings simplicity, flexibility and a clearer view of the interaction between the different controllers.

5.2 Outlook

Extending the capabilities of OS4 requires a further improvement of the dynamics of its actuators, its sensory capability and a more integrated design. The improvement in the bandwidth of the actuators will release the power of backstepping controllers. This will allow OS4 to be more stable, to fly in more difficult environments and to enlarge its flight envelope to more complex maneuvers. The robot is now equipped with the necessary sensors for autonomous hover flights. A first step in enhancing its sensory capability is to investigate optical flow or feature tracking algorithms to estimate the heading, speed and/or position onboard. The second step is to apply appropriate sensor fusion algorithms between inertial and vision sensors to have a better estimation of the state. This will allow the application of more sophisticated obstacle avoidance algorithms. OS4 provides actually a good level of integration, the possible further improvements are: Integrating the IMU to the motherboard, performing vision processing on-board and simplifying the structure.

The present size and weight of OS4 constrain its flying environment to wide rooms, while the generated air flow makes it difficult to fly in ordinary indoor environment. Therefore, miniaturization takes all its meaning, and is the inevitable way to practical applications like indoor area surveillance and rescue missions in collapsed buildings. Unfortunately, the development of micro aerial vehicles is facing stringent challenges and technical barriers that must be alleviated. First of all, aerodynamics at low Reynolds numbers lacks accurate models of flow separation and unsteady aerodynamics. A possible solution is to get inspiration from nature and try to control the delayed stall and wake capture. The propulsion efficiency at a small scale is still poor. The modelling and optimization of micro brushless outrunner motors is actually a feasible improvement. In addition, electro-active polymers are opening new possibilities, for the moment limited to low bandwidth actuation. Structures and materials have also to be adapted to be used as multifunctional smart materials permitting for instance the active shape deformation and energy storage at the same time. Moreover, the available stabilization and navigation algorithms require too much processing power and high resolution sensors. A potential solution is to couple insect-inspired navigation strategies with low resolution panoramic vision sensors. On the other hand, the demand for higher capacity batteries will never cease, and miniaturized fuel-cells represent now a promising technology. However, the real problem is rather in energy management and conversion and not only in the source itself. Finally, the system level integration is the key to bring together all these technologies in the optimal way.

All these challenges and possible solutions are now being investigated at ETHZ, through a European project called "muFly" lead by the author. This project proposes, the development and implementation of the first fully autonomous micro helicopter comparable in size and weight to a small bird. This project is an evident continuation of the work done in this thesis, it fits perfectly the vision of going always smaller, only when it is worth doing so.

Appendix A

Additional Control Plots

The robustness of the proposed IB controllers is verified in simulation using OS4 simulator under Matlab-Simulink. The reference impulse for attitude controller test is 4 s long and 0.2 rad high (see Fig. A.1). The signal for position controller test is 10 s long and 1 m high.

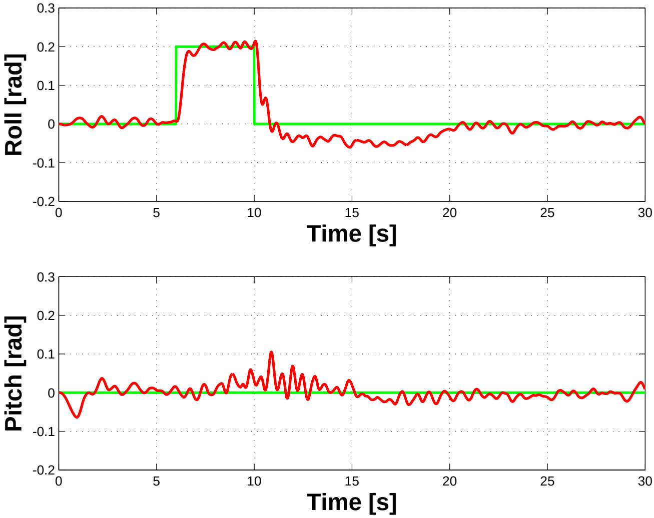


Figure A.1: Simulation: Investigation of the Integral Backstepping attitude controller. Reference (green), Output (red).

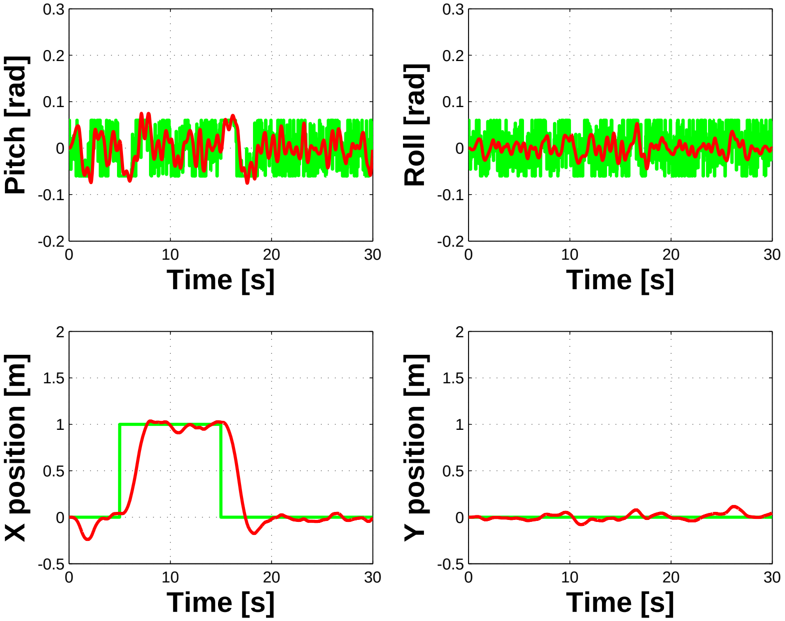


Figure A.2: Simulation: Investigation of the Integral Backstepping position controller. Reference (green), Output (red).

The control approach proposed in Section 4.10 is composed of position (*e.g.*: Angles) and velocity (*e.g.*: Angular rates) loops. Figure A.3 shows the control results of roll velocity loop in flight.

It is also possible to control only velocity loops using the same control laws (*e.g.*: 4.53) by simply zeroing the position error (*e.g.*: $e_1 = 0$).

A.1 Investigations in Obstacle Avoidance

Several obstacle avoidance maneuvers were imagined and simulated as shown in Fig. A.4). Most of them rely on the obstacle itself to stabilize the helicopter and operate the maneuver. The first approach (top-left) is to use vehicle's speed and distance to obstacles to generate evasive maneuvers in the safest direction. This approach can deal with any number of obstacles but it requires knowledge about the speed of the vehicle (see subsection 4.10.4). In

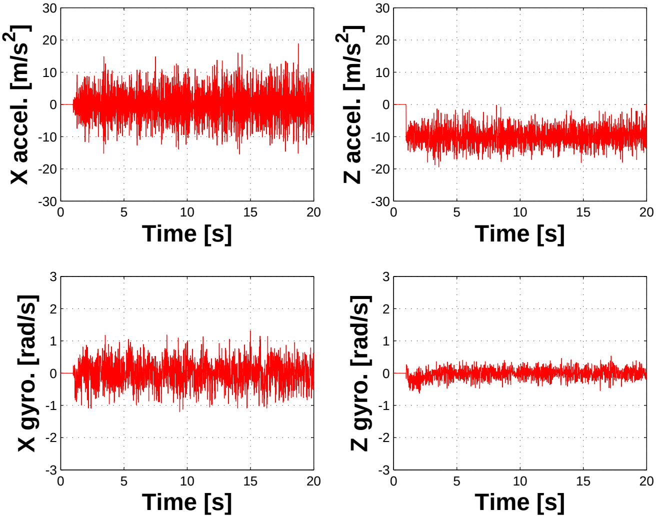


Figure A.3: Experiment: Acceleration and angular rate data in flight.

the second and third approaches (top-right and bottom-left), the helicopter detects the obstacle, stabilizes itself at a given distance and then generate skirting maneuvers around it. This way there is no need for knowledge about the global speed of the vehicle. However, this approach is slow and can deal with only one obstacle at a time. In the fourth approach (bottom-right), we assume that the maximum speed is about 1.5 m/s and thus, the helicopter is steered away from the obstacle without being stabilized in front of it, and then steered back to the original direction. This way, a smooth trajectory is obtained and the maneuver is much faster.

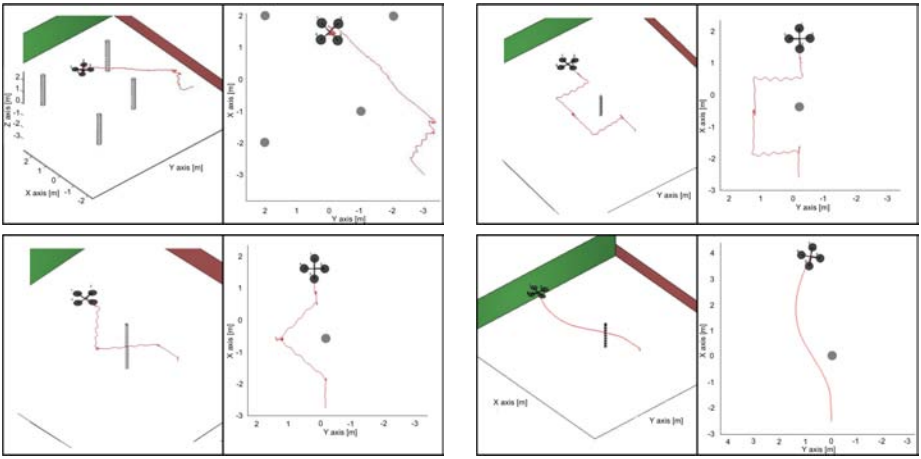


Figure A.4: The different obstacle avoidance maneuvers simulated.

Appendix B

Modelling

B.1 Rotation Matrix

The rotation of a rigid body in space can be parameterized using several methods like Euler angles, Quaternions and Tait-Bryan angles [43]. Tait-Bryan angles (also called "Cardano angles") are extensively used in aerospace engineering, where they are called "Euler angles". This conflicts with the real usage of "Euler angles", which is a mathematical representation of three successive rotations about different possible axes (numerous conventions) which are often confused in literature.

In aerospace engineering the axes are directed as for a craft moving in the positive x direction, with the right side corresponding to the positive y direction, and the vertical underside corresponding to the positive z direction. These three angles are individually called roll, pitch and yaw.

Considering a right-hand oriented coordinate system, the three single rotations are described separately by:

- $R(x, \phi)$, rotation around x-axis.
- $R(y, \theta)$, rotation around y-axis.
- $R(z, \psi)$, rotation around z-axis.

They are represented by:

$$R(x, \phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix} \tag{B.1}$$

$$R(y,\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \tag{B.2}$$

$$R(z,\psi) = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{B.3}$$

The complete rotation matrix is the product of the previous three successive rotations:

$$R(\phi,\theta,\psi) = R(x,\phi)R(y,\theta)R(z,\psi) \tag{B.4}$$

which results in:

$$R = \begin{bmatrix} \cos\psi\cos\theta & \cos\psi\sin\theta\sin\phi - \sin\psi\cos\phi & \cos\psi\sin\theta\cos\phi + \sin\psi\sin\phi \\ \sin\psi\cos\theta & \sin\psi\sin\theta\sin\phi + \cos\psi\cos\phi & \sin\psi\sin\theta\cos\phi - \sin\phi\cos\psi \\ -\sin\theta & \cos\theta\sin\phi & \cos\theta\cos\phi \end{bmatrix} \tag{B.5}$$

Angular Rates

The time variation of Tait-Bryan angles (ϕ,θ,ψ) is a discontinuous function. Thus, it is different from body angular rates (p,q,r), which are physically measured with gyroscopes. An IMU is generally used in aerospace to measure the body rotations. The transformation matrix from $[p \;\; q \;\; r]^T$ to $[\dot{\phi} \;\; \dot{\theta} \;\; \dot{\psi}]^T$ is given by [44]:

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = R_r \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{B.6}$$

where:

$$R_r = \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \sin\phi\cos\theta \\ 0 & -\sin\phi & \cos\phi\cos\theta \end{bmatrix} \tag{B.7}$$

B.2 Mathematical Derivation

The dynamic model is derived using Euler-Lagrange formalism [18] under the following assumptions:

- The structure is supposed rigid.
- The structure is supposed symmetrical.
- The CoG and the body fixed frame origin are assumed to coincide.
- The propellers are supposed rigid.
- Thrust and drag are proportional to the square of propeller's speed.

Recalling the Lagrangian $L = T - V$, and the general form of the equations of motion in Lagrange method.

$$\Gamma_i = \frac{d}{dt}\left(\frac{\delta L}{\delta \dot{q}_i}\right) - \frac{\delta L}{\delta q_i}$$

were:

q_i	generalized coordinates
Γ_i	generalized forces
T	kinetic energy
V	potential energy

Similarly to Section 2.2, we consider earth fixed frame E with the orthogonal base $[\vec{X}, \vec{Y}, \vec{Z}]$ and body fixed frame B with the orthogonal base $[\vec{x}, \vec{y}, \vec{z}]$, if any point of B experience three successive rotations, one can express any point of the body by:

$$r_{X,Y,Z}(x, y, z) = R(\phi, \theta, \psi)\begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

which is equivalent to:

$$r_X(x, y, z) = (\cos\psi\cos\theta)x + (\cos\psi\sin\theta\sin\phi - \sin\psi\cos\phi)y + (\cos\psi\sin\theta\cos\phi + \sin\psi\sin\phi)z$$
$$r_Y(x, y, z) = (\sin\psi\cos\theta)x + (\sin\psi\sin\theta\sin\phi + \cos\psi\cos\phi)y + (\sin\psi\sin\theta\cos\phi - \sin\phi\cos\psi)z$$
$$r_Z(x, y, z) = (-\sin\theta)x + (\cos\theta\sin\phi)y + (\cos\theta\cos\phi)z$$

Derivation with respect to time gives the corresponding speeds:

$$\begin{aligned}\nu_X(x,y,z) = {} & (-\sin\theta\cos\psi\dot{\theta} - \cos\theta\sin\psi\dot{\psi})x \\ & +(-\cos\psi\cos\phi\dot{\psi} + \sin\psi\sin\phi\dot{\phi} - \sin\psi\sin\phi\sin\theta\dot{\psi} + \cos\psi\cos\phi\sin\theta\dot{\phi} + \cos\psi\sin\phi\cos\theta\dot{\theta})y \\ & +(\cos\psi\sin\phi\dot{\psi} + \sin\psi\cos\phi\dot{\phi} - \sin\psi\cos\phi\sin\theta\dot{\psi} - \cos\psi\sin\phi\sin\theta\dot{\phi} + \cos\psi\cos\phi\cos\theta\dot{\theta})z\end{aligned}$$

$$\begin{aligned}\nu_Y(x,y,z) = {} & (-\sin\theta\sin\psi\dot{\theta} + \cos\theta\cos\psi\dot{\psi})x \\ & +(-\sin\psi\cos\phi\dot{\psi} - \cos\psi\sin\phi\dot{\phi} + \cos\psi\sin\phi\sin\theta\dot{\psi} + \sin\psi\cos\phi\sin\theta\dot{\phi} + \sin\psi\sin\phi\cos\theta\dot{\theta})y \\ & +(\sin\psi\sin\phi\dot{\psi} - \cos\psi\cos\phi\dot{\phi} + \cos\psi\cos\phi\sin\theta\dot{\psi} - \sin\psi\sin\phi\sin\theta\dot{\phi} + \sin\psi\cos\phi\cos\theta\dot{\theta})z\end{aligned}$$

$$\begin{aligned}\nu_Z(x,y,z) = {} & (-\cos\theta\dot{\theta})x \\ & +(\cos\phi\cos\theta\dot{\phi} - \sin\phi\sin\theta\dot{\theta})y \\ & +(-\sin\phi\cos\theta\dot{\phi} - \cos\phi\sin\theta\dot{\theta})z\end{aligned}$$

which could be rewritten:

$$\nu_X(x,y,z) = \nu_{Xx}x + \nu_{Xy}y + \nu_{Xz}z = \begin{pmatrix} \nu_{Xx} & \nu_{Xy} & \nu_{Xz} \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

$$\nu_Y(x,y,z) = \nu_{Yx}x + \nu_{Yy}y + \nu_{Yz}z = \begin{pmatrix} \nu_{Yx} & \nu_{Yy} & \nu_{Yz} \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

$$\nu_Z(x,y,z) = \nu_{Zx}x + \nu_{Zy}y + \nu_{Zz}z = \begin{pmatrix} \nu_{Zx} & \nu_{Zy} & \nu_{Zz} \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

The squared magnitude of the velocity for any point is given by:

$$\nu^2(x,y,z) = \nu_X^2(x,y,z) + \nu_Y^2(x,y,z) + \nu_Z^2(x,y,z)$$

$$\nu^2(x,y,z) = \begin{pmatrix} \nu_{Xx} & \nu_{Xy} & \nu_{Xz} \end{pmatrix} \Lambda \begin{pmatrix} \nu_{Xx} \\ \nu_{Xy} \\ \nu_{Xz} \end{pmatrix} + \begin{pmatrix} \nu_{Yx} & \nu_{Yy} & \nu_{Yz} \end{pmatrix} \Lambda \begin{pmatrix} \nu_{Yx} \\ \nu_{Yy} \\ \nu_{Yz} \end{pmatrix}$$

$$+ \begin{pmatrix} \nu_{Zx} & \nu_{Zy} & \nu_{Zz} \end{pmatrix} \Lambda \begin{pmatrix} \nu_{Zx} \\ \nu_{Zy} \\ \nu_{Zz} \end{pmatrix}$$

with:

$$\Lambda = \begin{pmatrix} x^2 & xy & xz \\ xy & y^2 & yz \\ xz & yz & z^2 \end{pmatrix}$$

then,

$$\begin{aligned}\nu^2(x,y,z) = {} & x^2 \cdot (\nu_{Xx}^2 + \nu_{Yx}^2 + \nu_{Zx}^2) \\ & + y^2 \cdot (\nu_{Xy}^2 + \nu_{Yy}^2 + \nu_{Zy}^2) \\ & + z^2 \cdot (\nu_{Xz}^2 + \nu_{Yz}^2 + \nu_{Zz}^2) \\ & + 2xy \cdot (\nu_{Xx} \cdot \nu_{Xy} + \nu_{Yx} \cdot \nu_{Yy} + \nu_{Zx} \cdot \nu_{Zy}) \\ & + 2xz \cdot (\nu_{Xx} \cdot \nu_{Xz} + \nu_{Yx} \cdot \nu_{Yz} + \nu_{Zx} \cdot \nu_{Zz}) \\ & + 2yz \cdot (\nu_{Xy} \cdot \nu_{Xz} + \nu_{Yy} \cdot \nu_{Yz} + \nu_{Zy} \cdot \nu_{Zz})\end{aligned}$$

$$\begin{aligned}\nu^2(x,y,z) = {} & x^2 \cdot (\cos^2\theta\dot{\psi}^2 + \dot{\theta}^2) \\ & + y^2 \cdot (\dot{\psi}^2(\cos^2\phi + \sin^2\phi\sin^2\theta) + \dot{\psi}(-2\sin\phi\cos\phi\cos\theta\dot{\theta} - 2\dot{\phi}\sin\theta) + \sin^2\phi\dot{\theta}^2 + \dot{\phi}^2) \\ & + z^2 \cdot (\dot{\psi}^2(\sin^2\phi + \cos^2\phi\sin^2\theta) + \dot{\psi}(2\sin\phi\cos\phi\cos\theta\dot{\theta} - 2\dot{\phi}\sin\theta) + \cos^2\phi\dot{\theta}^2 + \dot{\phi}^2) \\ & + 2xy(\dot{\psi}^2\sin\phi\sin\theta\cos\theta + \dot{\psi}(\cos\phi\sin\theta\dot{\theta} - \sin\phi\cos\theta\dot{\phi}) - \cos\phi\dot{\phi}\dot{\theta}) \\ & + 2xz(\dot{\psi}^2\cos\phi\sin\theta\cos\theta + \dot{\psi}(-\cos\phi\cos\theta\dot{\phi} - \sin\phi\sin\theta\dot{\theta}) + \sin\phi\dot{\phi}\dot{\theta} \\ & + 2yz(-\dot{\psi}^2\sin\phi\cos\phi\cos^2\theta + \dot{\psi}(\sin^2\phi\cos\theta\dot{\theta} - \cos^2\phi\cos\theta\dot{\theta}) + \sin\phi\cos\phi\dot{\theta}^2\end{aligned}$$

which could be rewritten as:

$$\begin{aligned}\nu^2(x,y,z) = {} & \left(y^2 + z^2\right)\left(\dot{\psi}^2\sin^2\theta - 2\sin\theta\dot{\phi}\dot{\psi} + \dot{\phi}^2\right) \\ & + \left(x^2 + z^2\right)\left(\dot{\psi}^2\sin^2\phi\cos^2\theta + 2\sin\phi\cos\phi\cos\theta\dot{\theta}\dot{\psi} + \cos^2\phi\dot{\theta}^2\right) \\ & + \left(x^2 + y^2\right)\left(\dot{\psi}^2\cos^2\phi\cos^2\theta - 2\sin\phi\cos\phi\cos\theta\dot{\theta}\dot{\psi} + \sin^2\phi\dot{\theta}^2\right) \\ & + 2xy(\dot{\psi}^2\sin\phi\sin\theta\cos\theta + \dot{\psi}(\cos\phi\sin\theta\dot{\theta} - \sin\phi\cos\theta\dot{\phi}) - \cos\phi\dot{\phi}\dot{\theta}) \\ & + 2xz(\dot{\psi}^2\cos\phi\sin\theta\cos\theta + \dot{\psi}(-\cos\phi\cos\theta\dot{\phi} - \sin\phi\sin\theta\dot{\theta}) + \sin\phi\dot{\phi}\dot{\theta} \\ & + 2yz(-\dot{\psi}^2\sin\phi\cos\phi\cos^2\theta + \dot{\psi}(\sin^2\phi\cos\theta\dot{\theta} - \cos^2\phi\cos\theta\dot{\theta}) + \sin\phi\cos\phi\dot{\theta}^2)\end{aligned}$$

The expression of the kinetic energy is then:

$$
\begin{aligned}
T = &\frac{1}{2}\int y^2 + z^2(R)dm(r) \cdot (\dot{\phi}^2 - \dot{\psi}\dot{\phi}2\sin\theta + \dot{\psi}^2\sin^2\theta) \\
&+\frac{1}{2}\int z^2 + x^2(R)dm(r) \cdot (\dot{\theta}^2\cos^2\phi + \dot{\theta}\dot{\psi}2\sin\phi\cos\phi\cos\theta + \dot{\psi}^2\sin^2\phi\cos^2\theta) \\
&+\frac{1}{2}\int x^2 + y^2(R)dm(r) \cdot (\dot{\theta}^2\sin^2\phi - \dot{\theta}\dot{\psi}2\sin\phi\cos\phi\cos\theta + \dot{\psi}^2\cos^2\phi\cos^2\theta) \\
&+\int xy(R)dm(r) \cdot (\dot{\psi}^2\sin\phi\sin\theta\cos\theta + \dot{\psi}(\cos\phi\sin\theta\dot{\theta} - \sin\phi\cos\theta\dot{\phi}) - \cos\phi\dot{\phi}\dot{\theta}) \\
&+\int xz(R)dm(r) \cdot (\dot{\psi}^2\cos\phi\sin\theta\cos\theta + \dot{\psi}(-\cos\phi\cos\theta\dot{\phi} - \sin\phi\sin\theta\dot{\theta}) + \sin\phi\dot{\phi}\dot{\theta} \\
&+\int yz(R)dm(r) \cdot (-\dot{\psi}^2\sin\phi\cos\phi\cos^2\theta + \dot{\psi}(\sin^2\phi\cos\theta\dot{\theta} - \cos^2\phi\cos\theta\dot{\theta}) + \sin\phi\cos\phi\dot{\theta}^2)
\end{aligned}
$$

where appear inertia moments (diagonal elements of the inertia matrix) and inertia products (off-diagonal elements). The mechanical symmetry of the quadrotor allows to neglect the inertia products and consider a diagonal inertia matrix. This was verified in the CAD model where the inertia products were a thousand times lower than the inertia moments. The kinetic energy become then:

$$
\begin{aligned}
T = &\frac{1}{2}I_{xx}(\dot{\phi} - \dot{\psi}\sin\theta)^2 \\
&+\frac{1}{2}I_{yy}(\dot{\theta}\cos\phi + \dot{\psi}\sin\phi\cos\theta)^2 \\
&+\frac{1}{2}I_{zz}(\dot{\theta}\sin\phi - \dot{\psi}\cos\phi\cos\theta)^2
\end{aligned}
$$

The expression of the potential energy is:

$$
V = g\int(-\sin\theta \cdot x + \sin\phi\cos\theta \cdot y + \cos\phi\cos\theta \cdot z)dm(r)
$$

$$
\begin{aligned}
V = &\int xdm(x) \cdot (-g\sin\theta) \\
&+\int ydm(y) \cdot (g\sin\phi\cos\theta) \\
&+\int zdm(z) \cdot (g\cos\phi\cos\theta)
\end{aligned}
$$

By considering the lagrangian $L = T - V$ and the equations of motion:

$$
\begin{aligned}
\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\phi}}\right) - \frac{\partial L}{\partial \phi} &= \tau_\phi \\
\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}}\right) - \frac{\partial L}{\partial \theta} &= \tau_\theta \\
\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\psi}}\right) - \frac{\partial L}{\partial \psi} &= \tau_\psi
\end{aligned}
$$

After development one get the following equations for Roll, Pitch and Yaw successively:

$$
\begin{aligned}
\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\phi}}\right) - \frac{\partial L}{\partial \phi} = {} & \ddot{\phi} \cdot I_{xx} \\
& -\ddot{\psi} \cdot \sin\theta \cdot I_{xx} \\
& -\dot{\psi}\dot{\theta} \cdot \cos\theta (I_{xx} + (I_{yy} - I_{zz})(2\cos\phi^2 - 1)) \\
& +\dot{\theta}^2 \cdot \frac{1}{2} \sin 2\phi (I_{yy} - I_{zz}) \\
& -\dot{\psi}^2 \cdot \frac{1}{2} \sin 2\phi \cos\theta^2 (I_{yy} - I_{zz}) \\
& + \int y dm(y) \cdot (-g \cos\phi \cos\theta) \\
& + \int z dm(z) \cdot (g \sin\phi \cos\theta)
\end{aligned}
$$

$$
\begin{aligned}
\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}}\right) - \frac{\partial L}{\partial \theta} = {} & \ddot{\theta} \cdot (I_{yy} \cos^2\phi + I_{zz} \sin^2\phi) \\
& +\ddot{\psi} \cdot \frac{1}{2} \sin 2\phi \cos\theta \cdot (I_{yy} - I_{zz}) \\
& +\dot{\psi}^2 \cdot \frac{1}{2} \sin 2\theta (-I_{xx} + I_{yy} \sin^2\phi + I_{zz} \cos^2\phi)) \\
& +\dot{\theta}\dot{\phi} \cdot \sin 2\phi (I_{zz} - I_{yy}) \\
& +\dot{\psi}\dot{\phi} \cdot \cos\theta (\cos 2\phi \cdot (I_{yy} - I_{zz}) + I_{xx}) \\
& + \int x dm(x) \cdot (-g \cos\theta) \\
& - \int y dm(y) \cdot (g \sin\phi \sin\theta) \\
& - \int z dm(z) \cdot (g \cos\phi \sin\theta)
\end{aligned}
$$

$$\begin{aligned}\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\psi}}\right) = &\ddot{\psi} \cdot (\cos^2\theta(I_{zz}\cos^2\phi + I_{yy}\sin^2\phi) + \sin^2\theta I_{xx})\\ &-\ddot{\phi} \cdot \sin\theta I_{xx}\\ &+\ddot{\theta} \cdot \frac{1}{2}\sin 2\phi\cos\theta(I_{yy} - I_{zz})\\ &+\dot{\theta}\dot{\psi} \cdot \sin 2\theta(I_{xx} - I_{zz}\cos^2\phi + I_{yy}\sin^2\phi)\\ &-\dot{\psi}\dot{\phi} \cdot \sin 2\phi\cos^2\theta(I_{yy} - I_{zz})\\ &+\dot{\theta}\dot{\phi} \cdot \cos\theta(I_{xx} + (2\cos^2\phi - 1)(I_{yy} - I_{zz}))\\ &-\dot{\theta}^2 \cdot \frac{1}{2}\sin 2\phi\sin\theta(I_{yy} - I_{zz})\end{aligned}$$

These equations can be simplified by expressing the speeds and accelerations of Euler angles in function of the instantaneous speeds and accelerations in the body fixed frame B by using the transformation matrix (B.7):

$$\begin{aligned}\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\phi}}\right) - \frac{\partial L}{\partial \phi} = &I_{xx}\dot{\omega}_x - (I_{yy} - I_{zz})\omega_y\omega_z\\ &+\int ydm(y) \cdot (-g\cos\phi\cos\theta)\\ &+\int zdm(z) \cdot (+g\sin\phi\cos\theta)\end{aligned}$$

$$\begin{aligned}\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}}\right) - \frac{\partial L}{\partial \theta} = &-\sin\phi(\dot{\omega}_z I_{zz} - \omega_x\omega_y(I_{xx} - I_{yy}))\\ &+\cos\phi(\dot{\omega}_y \cdot I_{yy} - \omega_x\omega_z(I_{zz} - I_{xx}))\\ &+\int xdm(x) \cdot (-g\cos\theta)\\ &-\int ydm(y) \cdot (g\sin\phi\sin\theta)\\ &-\int zdm(z) \cdot (g\cos\phi\sin\theta)\end{aligned}$$

$$\begin{aligned}\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\psi}}\right) = &-\sin\theta \cdot (\dot{\omega}_x I_{xx} - \omega_y\omega_z(I_{yy} - I_{zz}))\\ &+\sin\phi\cos\theta \cdot (\dot{\omega}_y I_{yy} - \omega_x\omega_z(I_{zz} - I_{xx}))\\ &+\cos\phi\cos\theta \cdot (\dot{\omega}_z I_{zz} - \omega_x\omega_y(I_{xx} - I_{yy}))\end{aligned}$$

The nonconservative moments are:

- $bl(\Omega_4^2 - \Omega_2^2)$: thrust imbalance between motor 2 and motor 4.
- $bl(\Omega_3^2 - \Omega_1^2)$: thrust imbalance between motor 1 and motor 3.
- $d(\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2)$: thrust imbalance of (1,3) and (2.4) propellers.

- $J_r\omega_y(\Omega_1 + \Omega_3 - \Omega_2 - \Omega_4)$: gyro. effect due to propellers' rotations.
- $J_r\omega_x(-\Omega_1 - \Omega_3 + \Omega_2 + \Omega_4)$: gyro. effect due to propellers' rotations.

The total moments acting about x, y and z axes are:

$$\tau_x = bl(\Omega_4^2 - \Omega_2^2) + J_r\omega_y(\Omega_1 + \Omega_3 - \Omega_2 - \Omega_4)$$
$$\tau_y = bl(\Omega_3^2 - \Omega_1^2) + J_r\omega_x(-\Omega_1 - \Omega_3 + \Omega_2 + \Omega_4)$$
$$\tau_z = d(\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2)$$

If now one applies small angle approximation, then the dynamics of the rotations subsystem become:

$$\ddot{\phi} = \frac{J_r\dot{\theta}(\Omega_1 + \Omega_3 - \Omega_2 - \Omega_4)}{I_{xx}} + \frac{I_{yy} - I_{zz}}{I_{xx}}\dot{\psi}\dot{\theta} + \frac{bl(\Omega_2^2 - \Omega_4^2)}{I_{xx}}$$

$$\ddot{\theta} = \frac{J_r\dot{\phi}(-\Omega_1 - \Omega_3 + \Omega_2 + \Omega_4)}{I_{yy}} + \frac{I_{zz} - I_{xx}}{I_{yy}}\dot{\psi}\dot{\phi} + \frac{bl(\Omega_3^2 - \Omega_1^2)}{I_{yy}}$$

$$\ddot{\psi} = \frac{d(\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2)}{I_{zz}} + \frac{I_{xx} - I_{yy}}{I_{zz}}\dot{\theta}\dot{\phi}$$

B.2.1 Test-Bench's Actuator Model Derivation

On the first platform (test-bench) we used standard DC motors obeying to the well known equations:

$$L\frac{di}{dt} = u - R_{mot}i - k_e\omega_m$$

$$J_m\frac{d\omega_m}{dt} = M_{em} - M_{fr}$$

with: $M_{em} = k_m \cdot i$ the motor torque and M_{fr} the friction moment. If we neglect the motor inductance L then:

$$i = \frac{u - k_e\omega_m}{R_{mot}}$$

Then:

$$J_m\frac{d\omega_m}{dt} = k_m \cdot \frac{u - k_e\omega_m}{R_{mot}} - M_{fr}$$

This is equivalent to:

$$J_m\dot{\omega}_m = -\frac{k_m^2}{R_{mot}}\omega_m - M_{fr} + \frac{k_m}{R_{mot}}u$$

In the case of a propeller with a gearbox, the friction torque experienced by the motor is:

$$M_{fr} = \frac{d\omega_m^2}{\eta r}$$

were η is the efficiency of the gearbox and r its reduction ratio. This is equivalent to:

$$\omega_{propeller} = \frac{\omega_m}{r} \qquad \Rightarrow \quad M_{fr} = \frac{d}{\eta r^3}\omega_m^2$$

The inertia seen by the motor is:

$$J_{propeller}\omega_{propeller}^2 = \eta \cdot J_{propeller \to motor}\omega_m^2$$

$$J_{propeller \to motor} = \frac{J_{propeller}}{\eta r^2}$$

$J_{propeller \to motor}$ is the propeller inertia seen by the motor. The motor's equation could be rewritten as:

$$\left(\frac{J_{propeller}}{\eta r^2} + J_m\right)\dot{\omega}_m = -\frac{k_m^2}{R_{mot}}\omega - \frac{d}{\eta r^3}\omega_m^2 + \frac{k_m}{R_{mot}}u$$

If J_t is the total inertia seen by the motor:

$$\dot{\omega}_m = -\frac{k_m^2}{RJ_t}\omega_m - \frac{d}{\eta r^3 J_t}\omega_m^2 + \frac{k_m}{RJ_t}u$$

this could be rewritten if we consider $\frac{1}{\tau} = \frac{k_m^2}{RJ_t}$:

$$\dot{\omega}_m = -\frac{1}{\tau}\omega_m - \frac{d}{\eta r^3 J_t}\omega_m^2 + \frac{1}{k_m\tau}u$$

This formula is linearized around ω_0 by using a first order Taylor series. We rewrite the system then if the form: $\dot{\omega}_m = -A\omega_m + Bu + C$. with:

$$A = \frac{1}{\tau} + \frac{2d\omega_0}{\eta r^3 J_t}, \quad B = \frac{1}{k_m\tau}, \quad C = \frac{d\omega_0^2}{\eta r^3 J_t}$$

Figure B.1 shows the speed step-response of the DC motor used in the first platform. The output of the actuator's model is plotted along with the measured data in open-loop and closed-loop. Closing the speed loop locally on each motor has a impact on the overall response of the quadrotor as shown in 4.4.2.

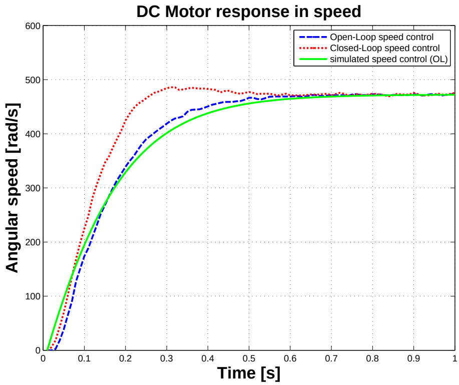


Figure B.1: Step-response of the DC motor used in the first platform, measured at motor shaft. The model output fits remarkably well the output measured in open-loop.

Appendix C

Sensor Data

C.1 Inertial Measurement Unit

During this thesis we used two different IMUs. The test-bench was equipped with the MT9-B from Xsens and the last version of OS4 uses the 3DM-GX1 from Microstrain. The influence of motors' vibrations on IMU's response is clear in Fig. C.1. It shows the behavior of roll and pitch angles of the MT9-B with the motors off and then on, while the IMU is kept static.

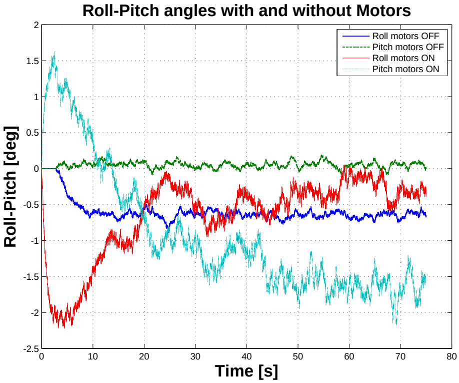


Figure C.1: Roll and pitch angles behavior with the motors OFF and ON.

On the other hand, it is interesting to compare the behavior of the MT9-B and the 3DM-GX1 under vibrations. So, both IMUs were kept static and subjected to a vibrations source. The result is shown in Fig. C.2.

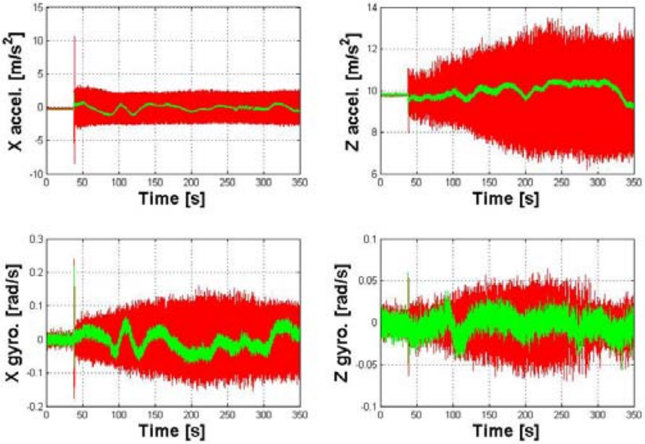


Figure C.2: Accelerometers' and gyroscopes' responses of the MT9-B (green) and the 3DM-GX1 (red).

The data from the 3DM-GX1 are obviously more noisy but once filtered, it becomes clear (see Fig. C.3) that this noise has zero mean. However, since then Xsens released new IMUs (MTi and MTx) with drastic improvements.

Yaw Sensor Issue

Most of commercially available IMU sensors use magnetometers to sense earth magnetic field direction and provide an absolute measurement of the yaw angle. This field is often disturbed in two different ways, namely by soft and hard iron interference. The latter is caused by permanent magnets such as those present on electrical motors. The sensor will measure the sum of both fields and thus provide wrong results. On the other hand, soft iron interference is caused by ferromagnetic objects in the vicinity of the sensor. These objects will distort earth magnetic field itself. The distortion is particularly strong inside buildings with metallic structures. Figure C.4 shows yaw drift with OS4 motors turned-off and turned-on. In the latter case, 5 seconds where enough to observe 0.25 rad of drift. Compensating these effects is only possible if they have a statical nature. This is generally not true with robots. These two interferences will cause the yaw data to drift in a very unpredictable way, especially in presence of variable speed motors, which is typically the case with the quadrotor. This drift will result in a growing error

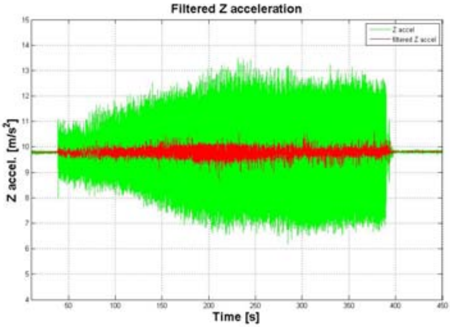


Figure C.3: Filtered Z acceleration of the 3DM-GX1.

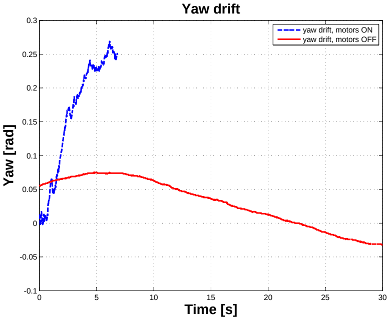


Figure C.4: Yaw drift with motors turned-off and turned-on.

in yaw (ψ) and yaw angular rate ($\dot{\psi}$). These errors cause control output saturation and vehicle instability after a while. It is possible to solve this issue by getting the yaw from another source like a vision sensor, or circumvent the problem by closing the loop on the yaw angular rate ($\dot{\psi}$) only. By doing so, one would reduce the drift to few degrees per second but will loose the absolute information about the yaw.

C.2 Range Sensor

Five range sensors are used on OS4, one for altitude control and four for obstacle avoidance. The hard constraints on mass and size restrict the choice to few models like the GP2Y0A02YK (infrared, 5 g) from *SHARP* or the SRF10 (ultrasound, 3.5 g) from *DEVANTECH*.
SRF10 is lighter and seems to be more accurate. Figure C.5 shows a distance measurement in static with both sensors.

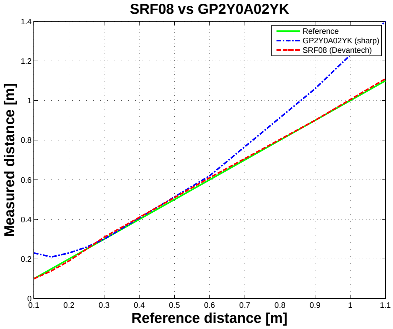


Figure C.5: Range measurement with SRF10 and GP2Y0A02YK.

However, the beam pattern of SRF10 is much wider which could be problematic in some situations. Therefore, we use plastic tubes (3 cm long) around the transceivers to reduce the beam width. Figure C.6 shows the beam volume with and without tubes. A reduction of the maximum gain (see Fig. C.7) was also necessary in order to reduce the sensitivity to side objects.

C.3 Position Sensor

Most of the small cameras available on the market require high lightening condition in order to deliver motion-blue free image. The position sensor

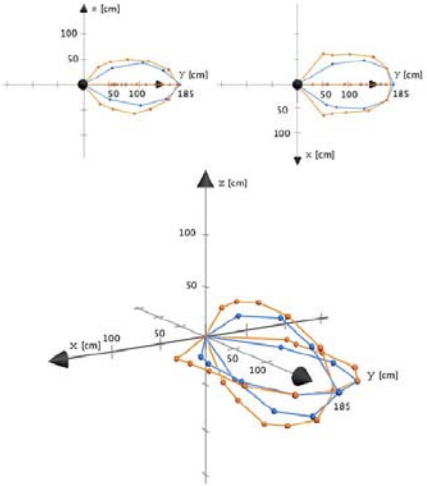


Figure C.6: Beam volume with tubes (blue) and without tubes (orange). Side view (top left), top view (top right), 3D view (bottom).

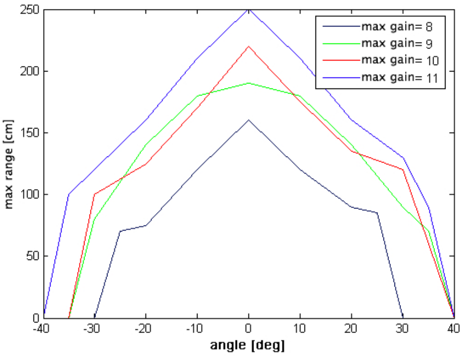


Figure C.7: Maximum range with different gains and 3 cm plastic tubes.

used on OS4 and presented in Subsection 3.4.4 uses a CCD camera which delivers an almost motion-blur free image of 320x240 at up to 25 fps as shown in Fig. C.8. The camera is used with a red A4 paper with a white spot shifted

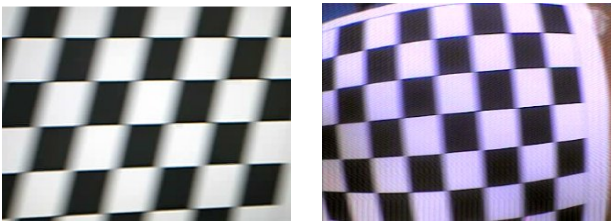


Figure C.8: Left: Image from a tiny CMOS camera (<1 g). Right: Image from our CCD camera (<16 g). Both images were taken at 1 m/s, our camera is almost motion-blur free.

from the pattern center as described in Subsection 3.4.4. The pattern is robustly detected. We use for that OpenCV [27] where Canny edge detector and Douglas-Peucker are algorithms already implemented. The pattern before and after detection is shown in Fig. C.9. The errors obtained in x and

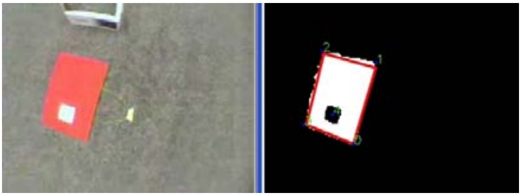


Figure C.9: The pattern before and after detection.

y position sensing are about 2 cm at 0.5 m/s. The error on the yaw is about 3 ° at 180 °/s. This is shown in Fig. C.10.

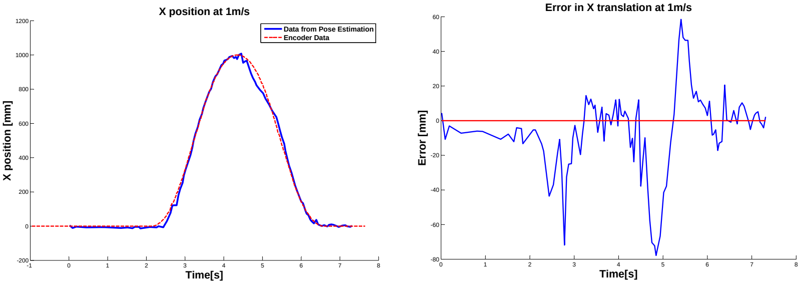


Figure C.10: Translation at 1 m/s measured with our sensor and with an encoder.

Appendix D

Implementation Details

D.1 Software Architecture

OS4 robot is equipped with several processing units namely:

- One processor Geode 1200 (AMD)
- One DSP 30F6014A (Microchip)
- Five Microcontrollers 16F876A (Microchip)

The motherboard shown in Fig. 3.13 hosts the Geode processor. The latter runs Linux and is mainly used for high level tasks which are often not time critical. It is also used for WIFI communication with the ground control station. On the other hand, the DSP (30 MIPS) runs the attitude, altitude and position control algorithms. All this processing is done in less than 4 ms. One microcontroller is interfaced with the RC receiver, it filters and decodes PPM signals into I^2C data. Finally, each of the four motor-control units is equipped with one microcontroller. All OS4's processing units are programmed in C, using standard GCC compiler for the processor and proprietary compilers for the DSP (C30) and the microcontrollers (CCS).

D.2 Implementation Requirements

Safety first, implementing a control software on a helicopter is primarily considering safety issues for the helicopter and the user. Therefore, we implemented a security layer which inhibits robot starting if the remote control is not detected and/or not turned in manual mode with the throttle at minimum. Proper operation of each sensor is also verified before take-off. If, for any reason, the contact is lost with the RC in flight, the safety layer automatically lands the helicopter. The implementation of a control loop on a helicopter must be done in a very careful way. In fact, the most critical part is the attitude loop, it must be strictly deterministic with the highest priority over the other processes (except safety layer). On OS4, this loop is synchronized with the IMU and is able to deal with temporary loss of its data. In general, a small bug would have catastrophic consequences. Therefore, extensive tests are required after each major code modification.

Integrators' Implementation

It is well known that a poor implementation of integrators would affect the performance of any control loop. In the specific case of flying systems, special care must be taken while implementing the integrators. The Algorithm 2 shows how we do it on OS4.

Algorithm 2 Roll integrator implementation.

```
if $e_1 > bandlimit \quad OR \quad e_1 < -bandlimit$ then
    $\chi_1 \leftarrow \chi_{1-1}$
else if $U_1 < saturationlimit \quad AND \quad \chi_1 < saturationlimit$ then
    $\chi_1 \leftarrow \chi_{1-1} + e_1 * sampligperiod$
else if $U_1 * \chi_1 > 0$ then
    $\chi_1 \leftarrow 0$
else

end if
```

Appendix E

The OS4 helicopter

The main platform developed during this thesis is a small-scale helicopter with four rotors in cross configuration. It is 772 mm in span and about 200 mm in height. The four arms are tilted by 5° wrt. the horizontal plane as shown in Fig. E.1.

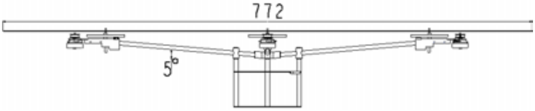


Figure E.1: Side view of the initial design of OS4.

The mechanics of a quadrotor are quite simple, however, one must keep in mind that any small failure would cause a crash in the next 500 ms! The design of the propulsion groups is delicate because they are subject to strong vibrations, and at the same time they are built with small mechanical parts. It is reasonable to say that 90% of the failures come from the mechanics. Figure E.2 depicts the position of the different components mounted on OS4.

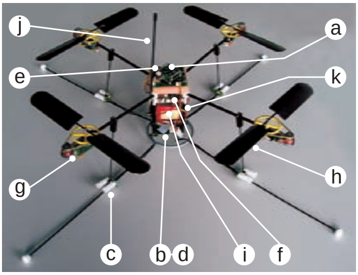


Figure E.2: Sensors, actuators and electronics of OS4. **(a)** inertial measurement unit, **(b)** altitude sensor below the robot, **(c)** obstacle avoidance sensor with tubes, **(d)** mini camera below the robot, **(e)** DSP, **(f)** mother board, **(g)** motor module, **(h)** propeller, **(i)** battery, **(j)** RC antenna, **(k)** wifi dongle.

E.1 OS4's Parameters

The design results presented in Subsection: 3.4.1 represent the results of the initial design, the one for which the design methodology was applied. Several mechanical and electrical parts were added since then in order to reinforce the structure or add functionalities. The parameters of the version of OS4 with all the new component are listed in Table: E.1.

Table E.1: OS4 parameters.

name	parameter	value	unit [mksA]
mass	m	0.650	kg
inertia on x axis	I_{xx}	7.5e-3	kg.m^2
inertia on y axis	I_{yy}	7.5e-3	kg.m^2
inertia on z axis	I_{zz}	1.3e-2	kg.m^2
thrust coefficient	b	3.13e-5	N s^2
drag coefficient	d	7.5e-7	Nm s^2
propeller radius	R_{rad}	0.15	m
propeller chord	c	0.04	m
pitch of incidence	θ_0	0.26	rad
twist pitch	θ_{tw}	0.045	rad
rotor inertia	J_r	6e-5	kg.m^2
arm length	l	0.23	m

Bibliography

[1] RCtoys, http://www.rctoys.com/.

[2] S. Bouabdallah *et al.*, "Design and control of an indoor micro quadrotor," in *Proc. (IEEE) International Conference on Robotics and Automation (ICRA'04)*, (New Orleans, USA), 2004.

[3] S. Bouabdallah, A. Noth, *et al.*, "PID vs LQ control techniques applied to an indoor micro quadrotor," in *Proc. (IEEE) International Conference on Intelligent Robots (IROS'04)*, (Sendai, Japan), 2004.

[4] S. Bouabdallah and R. Siegwart, "Backstepping and sliding-mode techniques applied to an indoor micro quadrotor," in *Proc. (IEEE) International Conference on Robotics and Automation (ICRA'05)*, (Barcelona, Spain), 2005.

[5] M. Krstić *et al.*, *Nonlinear and Adaptive Control Design.* New York, USA: Wiley Interscience, 1995.

[6] B. Marck, *Histoire de l'Aviation.* Flammarion, 1997.

[7] J. Boulet, *Histoire de l'Hélicoptère.* France-empire, 1991.

[8] C. Eck, *Navigation Algorithms with Applications to Unmanned Helicopters.* PhD thesis, ETHZ, 2001.

[9] Aerovironment, http://www.aerovironment.com.

[10] Epson, http://www.epson.co.jp/.

[11] D. Pines and F. Bohorquez, "Challenges facing future micro air vehicle development," *AIAA Journal of Aircraft*, vol. 43, no. 2, pp. 290–305, 2006.

[12] A. Elfes *et al.*, “Robotic airship for exploration of planetary bodies with an atmosphere autonomy challenges,” *Autonomous Robots*, 2003.

[13] MARV, http://www.enae.umd.edu/AGRC/Design00/MARV.html.

[14] P. Samuel *et al.*, “Design and testing of a rotary wing mav with an active structure for stability and control,” in *Proc. AHS Annual Forum 61*, (Grapevine, USA), 2005.

[15] J. Watkinson, *The Art of the Helicopter*. Elsevier, 2004.

[16] S. Bouabdallah *et al.*, “Design and control of an indoor coaxial helicopter,” in *Proc. (IEEE) International Conference on Intelligent Robots (IROS’06)*, (Beijing, China), 2006.

[17] G. Done and D. Balmford, *Bramwell’s Helicopter Dynamics*. Oxford Butterworth-Heinemann, 2001.

[18] P. Mullhaupt, *Analysis and Control of Underactuated Mechanical Nonminimum-phase Systems*. PhD thesis, EPFL, 1999.

[19] S. Bouabdallah and R. Siegwart, “Towards intelligent miniature flying robots,” in *Proc. of Field and Service Robotics*, (Port Douglas, Australia), 2005.

[20] R. Murray *et al.*, *A Mathematical Introduction to Robotic Manipulation*. CRC, 1994.

[21] J. G. Leishman, *Principles of Helicopter Aerodynamics*. Cambridge University Press.

[22] G. Fay, “Derivation of the aerodynamic forces for the mesicopter simulation,” 2001.

[23] N. Guenard *et al.*, “Control laws for the tele operation of an unmanned aerial vehicle known as an x4-flyer,” in *Proc. (IEEE) International Conference on Intelligent Robots (IROS’06)*, (Beijing, China), 2006.

[24] I. Cheeseman and W. Bennett, “The effect of the ground on a helicopter rotor in forward flight,” *Aeronautical Research Council*, no. 3021, 1957.

[25] D. Griffiths and J. Leishman, “A study of dual-rotor interference and ground effect using a free-vortex wake model,” in *Proc. of the 58th Annual Forum of the Ameriacan Helicopter Society*, (Montréal, Canada), 2002.

[26] J. Nicoud and J. Zufferey, "Toward indoor flying robots," in *Proc. (IEEE) International Conference on Intelligent Robots (IROS'02)*, (Lausanne, Switzerland), 2002.

[27] OpenCV, http://www.intel.com/research/mrl/research/opencv/.

[28] D. DeMenthon and L. Davis, "Model-based object pose in 25 lines of code," *International Journal of Computer Vision*, vol. 15.

[29] P. Pounds *et al.*, "A practical quad-rotor robot," in *Proc. Australasian Conference on Robotics and Automation (ACRA'06)*, (Auckland, New Zealand), Dec. 2006.

[30] G. Hoffmann *et al.*, "The stanford testbed of autonomous rotorcraft for multi agent control (starmac)," in *Proc. 23rd Digital Avionics Systems Conference (DASC'04)*, (Salt Lake City, USA), Oct. 2004.

[31] S. Arimoto, *Control Theory of Non-linear Mechanical Systems.* Oxford Science, 1996.

[32] R. Longchamp, *Commande Numérique de Systèmes Dynamiques.* PPUR, 1995.

[33] R. Longchamp, *Commande Sous-optimale Adaptative de Systèmes Nonlineaires.* PhD thesis, EPFL, 1978.

[34] P. Naslin, *Introduction à la Commande Optimale.* Dunod, 1966.

[35] F. Franklin *et al.*, *Digital Control of Dynamic Systems.* Addison Wesley, 1990.

[36] R. Sepulchre *et al.*, *Constructive Nonlinear Control.* Springer, 1997.

[37] I. Fantoni and R. Lozano, *Non-linear control for underactuated mechanical systems.* Springer, 2002.

[38] J. Slotine, *Applied nonlinear control.* Prentice-Hall, 1991.

[39] R. Olfati-Saber, *Nonlinear Control of Underactuated Mechanical Systems with Application to Robotics and Aerospace Vehicles.* PhD thesis, MIT, 2001.

[40] S. Waslander *et al.*, "Multi-agent quadrotor testbed control design: Integral sliding mode vs. reinforcement learning," in *Proc. International Conference on Intelligent Robots and Systems (IROS'05)*, (Edmonton, Canada), Aug. 2005.

[41] I. Kanellakopoulos and P. Krein, "Integral-action nonlinear control of induction motors," in *Proc. of the 12th IFAC World Congress*, (Sydney, Australia), 1993.

[42] Y. Tan *et al.*, "Advanced nonlinear control strategy for motion control systems," in *Proc. of (IEEE) Power Electronics and Motion Control Conference, (PIEMC'00)*, (Beijing, China), 2000.

[43] H. Goldstein *et al.*, *Classical Mechanics.* Addison Wesley, 2002.

[44] B. Etkin and L. Reid, *Dynamics of Flight: Stability and Control.* John Wiley and Sons, 1996.

[45] P. Pounds *et al.*, "Towards dynamically-favourable quad-rotor aerial robots," in *Proc. Australasian Conference on Robotics and Automation (ACRA'04)*, (Canberra, Australia), Dec. 2004.

[46] P. Pounds and R. Mahony, "Small-scale aeroelastic rotor simulation, design and fabrication," in *Proc. Australasian Conference on Robotics and Automation (ACRA'05)*, (Sydney, Australia), Dec. 2005.

[47] T. Hamel and R. Mahony, "Pure 2d visual servo control for a class of under-actuated dynamic systems," in *Proc. (IEEE) International Conference on Robotics and Automation (ICRA'04)*, (Canberra, Australia), Apr. 2004.

[48] P. Castillo *et al.*, "Real-time stabilization and tracking of a four rotor mini rotorcraft," *IEEE Transactions On Control Systems Technology*, vol. 12, pp. 510–516, July 2004.

[49] I. Kroo *et al.*, "The mesicopter: A miniature rotorcraft concept phase 2 interim report," 2000.

[50] University of Florida GATOR, http://www.mae.ufl.edu/mav/.

[51] Berkeley MFI, http://www.berkeley.edu.

[52] X. Deng *et al.*, "Attitude control for a micromechanical flying insect including thorax and sensor models," in *Proc. (IEEE) International Conference on Robotics and Automation (ICRA'03)*, (Teipei, Taiwan), 2003.

[53] Caltech MicroBat, http://touch.caltech.edu/research/bat/bat.html.

[54] AERO-EPFL, http://aero.epfl.ch/.

[55] S. Bouabdallah *et al.*, “Towards autonomous indoor micro vtol,” *Autonomous Robots*, vol. 18, Mar. 2005.

[56] H. K. Khalil, *Nonlinear Systems.* USA: Prentice Hall, 2001.

[57] VTI SCA100T, http://www.vti.fi.

[58] Analog Device ADXRS300, http://www.analog.com.

[59] SHARP GP2Y0A02YK, http://www.sharpsma.com.

[60] F. Bohorquez, “Design, analysis and performance of a rotary wing mav,” 2001.

[61] PROXFLYER, http://www.proxflyer.com/.

[62] O. Tanner, *Modeling, Identification and Control of Autonomous Helicopters.* PhD thesis, ETHZ, 2003.

[63] J. Zufferey and D. Floreano, “Evolving vision-based flying robots,” in *Proc. 2nd International Workshop on Biologically Motivated Computer Vision*, (Berlin, Germany), pp. 592–600, 2002.

[64] P. Mullhaupt *et al.*, “Cascade control of the toycopter,” in *Proc. of the European Control Conference*, (Karlsruhe, Germany), 1999.

[65] A. Chriette, *Contribution à la Commande et à la Modélisation des Hélicoptères : Asservissement Visuel et Commande Adaptative.* PhD thesis, Université d‘Evry, 2001.

[66] S. Pounds and R. Mahony, “Design of a four-rotor aerial robot,” in *Proc. Australasian Conference on Robotics and Automation*, (Auckland, New Zealand), 2002.

[67] M. Gafvert, “Modelling of the ETH helicopter,” 2001.

[68] S. Bouabdallah *et al.*, “Autonomous miniature flying robots: Coming soon!,” *Robotics and Automation Magazine*, 2007.

[69] W. Wang *et al.*, “Autonomous control for micro-flying robot and small wireless helicopter x.r.b,” in *Proc. (IEEE) International Conference on Intelligent Robots (IROS’06)*, (Beijing, China), 2006.

[70] R. W. Prouty, *Helicopter Performance, Stability, and Control.* PWS Engineering.

[71] R. V. Mises, *Theory of flight.* Dover Publications.

[72] G. D. Padfield, *Helicopter flight dynamics.* Blackwell Science.

[73] A. Fossard, *Commande des Systems Multidimentionnels.* Dunod.

[74] S. Skogestad, *Multivariable Feedback Control: Analysis and design.* Wiley Publishers.

[75] I. D. Landau, *Commande des systemes.* Hermès science.

[76] F. Weick, *Aircraft Propeller Design.* McGraw-Hill.

[77] W. H.Yeadon, *Handbook of Small Electric Motors.* McGraw-Hill.

[78] H. Baruh, *Analytical Dynamics.* McGraw-Hill.

[79] F. Lewis and C. Abdallah, *Control of Robot Manipulators.* Macmillan.

[80] R. A. Freeman and P. Kokotović, "A new lyapunov function for the backstepping design of 'softer' robust nonlinear control laws," tech. rep., Center for Control Engineering and Computation. University of California at Santa Barbara, 1992.

[81] P. V. Kokotović, "Joy of feedback: Nonlinear and adaptive, (1991 bode prize lecture)," tech. rep., Center for Control Engineering and Computation. University of California at Santa Barbara, 1992.

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

[83] P. Castillo *et al.*, *Modelling and Control of Mini-Flying Machines.* Springer.

[84] E. Altuğ *et al.*, "Control of a quadrotor helicopter using visual feedback," in *Proc. (IEEE) International Conference on Robotics and Automation (ICRA'02)*, (Washington, USA), 2002.

[85] E. Altuğ *et al.*, "Quadrotor control using dual camera visual feedback," in *Proc. (IEEE) International Conference on Robotics and Automation (ICRA'03)*, (Taipei, Taiwan), 2003.

[86] E. Altuğ, *Vision based control of unmanned aerial vehicles with applications to an autonomous four rotor helicopter, Quadrotor.* PhD thesis, University of Pennsylvania, 2003.

[87] M. Kemper and S. Fatikow, "Impact of center of gravity in quadrotor helicopter controller design," in *Proc. 4th IFAC-Symposium on Mechatronic Systems*, (Heidelberg, Germany), 2006.

Index

A
Aerodynamics, 4, 15, 25
Altitude, controller, 72
Altitude, sensor for, 110
Autonomy, 33, 35

C
Control, 45
Control, contribution in, 5
Control, state of the art, 2

E
Euler, angles, 19, 95, 102
Euler-Lagrange, formalism, 15, 16

G
Ground, effect, 21, 75

I
Identification, 15, 24, 86
IMU, 1, 87, 107
IMU, Microstrain 3DM-GX1, 70
IMU, MT9-B, 29
Integral backstepping, 69

L
LaSalle's, theorem, 48, 72
Lyapunov, theory of, 5, 47, 48

M
MAV, 1, 5, 8
MAV, classification of, 9
MAV, common configurations of, 10
MAV, common requirements of, 8
methodology, 4, 5, 29
methodology, formulation of the, 33
Modelling, for control, 45

N
Newton-Euler, formalism, 19

O
OS4, 5
OS4, last version of, 38

P
Position, control of, 75
Position, control results, 77

Q
Quadrotor, 2, 5, 87

S
Saturation, 86, 110
Saturation, actuator's, 2
Sliding-mode, 63
Stability, analysis, 71

T
Tait-Bryan, 96
Tait-Bryan, angles, 15, 95
Take-off and Landing, 75

U
Underactuated, systems, 68

Curriculum Vitæ

Samir Bouabdallah was born in Oran, Algeria on May 3^{rd} 1977. He received his M.Sc. in Electrical Engineering from Abu Bekr Belkaid University (ABBU) Tlemcen, Algeria in 2001. His master thesis was the development of an autonomous mobile robot for academic research. Late 2001, he started work at the Autonomous Systems Lab (ASL) at EPFL as research assistant and has been involved in several activities, for example designing the hardware for the educational robot Smartease, giving lectures, supervising student projects (more than 18 projects), co-organizing several times EPFL's robots contest, and participating in the design of RoboX the tour guide robot of Expo.02, an exposition organized in Switzerland every 25 years. In 2003, he started work on his doctoral thesis in the field of design and control of miniature flying robots at the ASL (EPFL). In 2006, he moved to Swiss Federal Institute of Technology Zürich (ETHZ), as project leader at the new Autonomous Systems Lab. He is the author of several conference and journal papers and one book chapter on aerial robotics. His current research interests are control systems and design optimization of VTOL Miniature Flying Robots.

List of Publications

Book Chapters

- Bouabdallah, S. and Siegwart, R. "Design and Control of a Miniature Quadrotor". Accepted for the book: *Advances in Unmanned Aerial Vehicles*, K.Valavanis, Springer, 2006.

Journals

- Bouabdallah, S., Becker, M. and Siegwart, R. "Autonomous Miniature Flying Robots: Coming Soon!" Accepted for the IEEE, *Robotics and Automation Magazine*, 2006.

- Bouabdallah, S., Murrieri, P. and Siegwart, R. "Towards Autonomous Indoor Micro VTOL". *Autonomous Robots*, Volume 18, Number 2, March 2005.

Peer-reviewed Proceedings

- Bouabdallah, S., Becker, M. and Siegwart, R. "Implementation of an Obstacle Avoidance Controller on the OS4 mini-helicopter". Accepted for XII Diname *International Symposium on Dynamic Problems of Mechanics.* Ilha Bela, Brazil, 2007.

- Bouabdallah, S. ,Caprari G. and Siegwart, R. "Design and Control of an Indoor Coaxial Helicopter". In Proceedings of the IEEE *International Conference on Intelligent Robots and Systems*, Beijing, China, 2006.

- Bouabdallah, S. and Siegwart, R. "Towards Intelligent Miniature Flying Robots". In Proceedings of *Field and Service Robotics*, Port Douglas, Australia, 2005.

- Bouabdallah, S. and Siegwart, R. "Backstepping and Sliding-mode Techniques Applied to an Indoor Micro Quadrotor". In Proceedings of IEEE *International Conference on Robotics and Automation*, Barcelona, Spain, 2005.

- Noth, A., Bouabdallah, S., Michaud, S., Siegwart, R. and Engel, W. "SKYSAILOR Design of an autonomous solar powered martian airplane". In Proceedings of the *8th ESA Workshop on Advanced Space Technologies for Robotics*, Noordwijk, The Netherlands, 2004.

- Bouabdallah, S., Noth, A. and Siegwart, R. "PID vs LQ Control Techniques Applied to an Indoor Micro Quadrotor". In Proceedings of the IEEE *International Conference on Intelligent Robots and Systems*, Sendai, Japan, 2004.

- Bouabdallah, S., Murrieri, P. and Siegwart, R. "Design and Control of an Indoor Micro Quadrotor". In Proceedings of *International Conference on Robotics and Automation*, New Orleans, USA, 2004.

- Tomatis, N., Bouabdallah, S., Piguet, R., Terrien, G. and Siegwart, R. "Autonomous Navigation and Security: A 13'000h/3'000km Case Study". In Proceedings of the 35th *International Symposium on Robotics*, Paris, France, 2004.

- Tomatis, N., Terrien, G., Piguet, R., Burnier, D., Bouabdallah, S., Arras, K.O. and Siegwart, R. "Designing a Secure and Robust Mobile Interacting Robot for the Long Term". In Proceedings of the IEEE *International Conference on Robotics and Automation*, Taipei, Taiwan, 2003.

- Tomatis, N., Terrien, G., Piguet, R., Burnier, D., Bouabdallah, S. and Siegwart, R. "Design and System Integration for the Expo.02 Robot". Workshop on Robots in Exhibitions, IEEE/RSJ *International Conference on Intelligent Robots and Systems*, Lausanne, Switzerland, 2002.