Pontifícia Universidade Católica
do Rio de Janeiro

Yessica Rosas Cuevas

Modeling and control of a quadcopter for autonomous navigation in agricultural fields

Dissertação de Mestrado

Thesis presented to the Programa de Pós–graduação em Engenharia Elétrica da PUC-Rio in partial fulfillment of the requirements for the degree of Mestre em Engenharia Elétrica .

Advisor: Prof. Antonio Candea Leite

Rio de Janeiro
August 2020

Pontifícia Universidade Católica
do Rio de Janeiro

Yessica Rosas Cuevas

Modeling and control of a quadcopter for autonomous navigation in agricultural fields

Thesis presented to the Programa de Pós–graduação em Engenharia Elétrica da PUC-Rio in partial fulfillment of the requirements for the degree of Mestre em Engenharia Elétrica . Approved by the Examination Committee.

Prof. Antonio Candea Leite
Advisor
Departamento de Engenharia Elétrica – PUC-Rio

Prof. Wouter Caarls
President
Departamento de Engenharia Elétrica – PUC-Rio

Prof. Eduardo Costa da Silva
Departamento de Engenharia Elétrica – PUC-Rio

Prof. Karla Tereza Figueiredo Leite
Universidade do Estado do Rio de Janeiro – UERJ

Rio de Janeiro, August the 3rd, 2020

Yessica Rosas Cuevas

Graduate in System Engineering from Universidad Nacional de San Agustin (Arequipa, Perú)

Bibliographic data

Rosas Cuevas, Yessica

Modeling and control of a quadcopter for autonomous navigation in agricultural fields / Yessica Rosas Cuevas; advisor: Antonio Candea Leite. – Rio de janeiro: PUC-Rio, Departamento de Engenharia Elétrica, 2020.

v., 103 f: il. color. ; 30 cm

Dissertação (mestrado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Engenharia Elétrica.

Inclui bibliografia

1. Engenharia Elétrica – Teses. 2. Quadricóptero;. 3. Modelo Dinâmico;. 4. Controle Adaptativo;. 5. Planejamento de Trajetórias. 6. Navegação Autônoma.. I. C. Leite, Antonio. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Engenharia Elétrica. III. Título.

CDD: 621.3

To my parents, for their support
and encouragement.

Acknowledgments

I gratefully acknowledge the support of my advisor Professor Antonio C. Leite, and I am also indebted to the Professors of the examination committee, for their support, their approach to research and writing, help me to improve this work.

Many thanks go to the other students and staff in PUC-Rio, with whom it was a pleasure to share the study room 604 and laboratories, by its many scientific and unscientific discussions but always with a cup of coffee.

I am grateful to the National Council for Scientific and Technological Development (CNPq) for the opportunity and the financial support. This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

Finally, I would like to thank my parents Paulina and Juan, without their continuous love and support, this thesis would never have been.

Abstract

Rosas Cuevas, Yessica; C. Leite, Antonio (Advisor). **Modeling and control of a quadcopter for autonomous navigation in agricultural fields**. Rio de Janeiro, 2020. 103p. Master Dissertation – Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro.

In this work, we address the modeling and control design of a quadrotor for autonomous navigation in agricultural environments. The kinematic and dynamic models of the aerial vehicle are derived following the Newton-Euler formalism. The motion system of the quadrotor can be split into two subsystems, that is, translational and rotational subsystems, responsible for controlling the position along the longitudinal, transverse and vertical axes of the Cartesian space as well as its orientation about the corresponding axes. The first linear control approach is based on the proportional-derivative (PD) controller, whereas the second nonlinear control approach is based on an adaptive controller in order to deal with the presence of uncertainties in the system parameters. Numerical simulations are carried out in Matlab to illustrate the performance and feasibility of the proposed control methodology. Gazebo was used to perform the 3D simulations for verifying autonomous navigation in agricultural fields.

Keywords

Quadrotor; Dynamic Model; Adaptive Control Design; Trajectory Planning; Autonomous Navigation.

Resumo

Rosas Cuevas, Yessica; C. Leite, Antonio. **Modelagem e controle de um quadricóptero para navegação autônoma em campos agrícolas**. Rio de Janeiro, 2020. 103p. Dissertação de Mestrado – Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro.

Neste trabalho, aborda-se a modelagem e controle de um quadricóptero para navegação autônoma em ambientes agrícolas. Os modelos cinemático e dinâmico do veículo aéreo são computados a partir do formalismo de Newton-Euler, incluindo efeitos aerodinâmicos e características das hélices. O sistema de movimento do quadricóptero pode ser dividido em dois subsistemas, um translacional e outro rotacional, responsáveis pelo controle de posição nos eixos x, y, z, and atitude do veículo no espaço Cartesiano. A primeira abordagem de controle é linear, se presenta dois controladores, um controlador proporcional-derivativo (PD) e o adaptativo baseado no espaço de estados. A segunda abordagem é não-linear e baseada em um controlador adaptativo a fim de lidar com a presença de incertezas nos parâmetros do sistema. Simulações numéricas são executadas em Matlab para ilustrar o desempenho e a viabilidade da metodologia de controle proposta. Simulações computacionais 3D são executadas em Gazebo para verificar a navegação autônoma em um campo agrícola.

Palavras-chave

Quadricóptero; Modelo Dinâmico; Controle Adaptativo; Planejamento de Trajetórias Navegação Autônoma.

Table of contents

1 Introduction **18**
1.1 Classification of UAVs 18
1.1.1 Fixed-Wing Aircraft 19
1.1.2 Rotary-wing aircraft 19
1.1.3 Bio-inspired aircraft 21
1.2 Applications of Aerial Robots 22
1.2.1 Precision Agriculture 24
1.3 Motivation 27
1.4 Review of the State of the Art 28
1.4.1 Quadrotor models 29
1.4.2 Quadrotor controllers 29
1.4.3 Vision-Based Navigation 32
1.5 Objectives 34
1.6 Methodology 34
1.7 Outline of thesis 35

2 Background **36**
2.1 Preliminaries 36
2.2 Nomenclature 37
2.3 Reference Frames 37
2.3.1 Inertial Reference Frame 37
2.3.2 Body Reference Frame 38
2.4 Basics of the Quadrotor Motion 38
2.4.1 Roll Motion 39
2.4.2 Pitch Motion 39
2.4.3 Yaw Motion 40
2.4.4 Throttle Motion 40
2.5 The Quadrotor Kinematic Model 40
2.6 Quadrotor Dynamic Model 44
2.6.1 Translational Motion Equation 44
2.6.2 Rotational Motion Equation 45
2.7 Newton-Euler model 46
2.8 State-space model 48
2.8.1 Controllability and Observability of the Linear System 51
2.8.2 Conclusions 52

3 Vision based Navigation **53**
3.1 Digital image 53
3.1.1 Perspective projection 54
3.2 Crop Detection 54
3.2.1 Image collection 56
3.2.2 Agronomic Index 56
3.2.3 Image Processing 57
3.2.3.1 Canny Edge Detector 58

3.2.3.2 Hough Transform Method 58
3.3 The Vanishing Point Algorithm 59
3.4 Evaluation of Vision Module 60
3.5 Autonomous navigation of UAV 62
3.6 Experiments and results 65
3.7 Conclusions 66

4 Quadrotor Control Strategies **67**
4.1 Mathematics of quadrotors Control 68
4.2 Proportional Integration Derivative Control 69
4.2.1 Position Controller 69
4.2.2 Altitude Controller 70
4.2.3 Attitude Controller 71
4.2.4 Rate Controller 71
4.2.5 Control Mixer 71
4.2.6 Simulation Results 71
4.3 Adaptive Linear Control 76
4.3.1 Pole Placement 76
4.3.2 Model Reference Adaptive Control 77
4.3.3 Linear Controller Design 77
4.3.4 Simulation Results 80
4.4 Adaptive Non-linear Control 85
4.4.1 Altitude Control 85
4.4.2 Attitude Control 87
4.4.2.1 Reparameterization 89
4.4.3 Simulation Results 90
4.5 Conclusions 92

5 Concluding Remarks **93**
5.1 Conclusions 93
5.2 Perspective and Future Works 93

Bibliography **95**

A Quadrotor mechanical characteristics **103**

List of figures

Figure 1.1 UAVs classification based on aerodynamic configuration 19

Figure 1.2 Examples of fixed-wing aircraft 20

1.2(a)Airbus A380 aircraft 20

1.2(b)Albatross aircraft 20

Figure 1.3 Rotary wing aircraft configuration 21

1.3(a)Swiss UAV Neo S-300 21

1.3(b)Nasa Airbus H135 Helicopter 21

1.3(c)DJI Phantom 4 Pro 21

1.3(d)DJI Agras MG-1P 21

Figure 1.4 Bio-inspired aircraft robots. 22

1.4(a)RoboBee 22

1.4(b)SmartBird 22

1.4(c)eMotionButterflies 22

Figure 1.5 Precision Agriculture 25

1.5(a)Remote Sensing 25

1.5(b)Nitrogen Status 25

Figure 1.6 Research articles on Quadrotors. 27

Figure 1.7 Search analysis in Scopus. 28

Figure 2.1 Earth-fixed reference frame $\mathcal{F}_I$ and body-fixed reference frame $\mathcal{F}_B$. The yellow color denotes the front rotor of the quadrotor. 38

Figure 2.2 Positive roll motion. 39

Figure 2.3 Negative pitch motion. 40

Figure 2.4 Negative yaw motion. 41

Figure 2.5 Throttle motion. 41

Figure 2.6 Forces and moments acting on a quadrotor. 44

Figure 3.1 System architecture modules 54

Figure 3.2 The central projection model. 55

Figure 3.3 Geometry of the central projective model 56

Figure 3.4 Image acquisition 57

3.4(a)Strawberry polytunnel 57

3.4(b)Simulation 57

3.4(c)Row crop 57

3.4(d)Simulation 57

Figure 3.5 Hough line representation on an image plane 58

Figure 3.6 The framework of crop row detection using vanishing point. 60

Figure 3.7 Vanishing point detection test in real crop images. 61

Figure 3.8 Vanishing point detection test in simulated images 62

Figure 3.9 Possible vanishing point location 64

Figure 3.10 Various stages of VP estimation 65

Figure 3.11 Various stages of VP estimation 66

Figure 4.1 Overall control frame of a quadrotor 67

Figure 4.2 The Simulink model with a closed-loop simulation of the quadrotor with PID Control, with a circular trajectory at a constant altitude. [60] 70
Figure 4.3 Measured position during a test flight, 3D trajectory. 72
Figure 4.4 Simulation results for Position and Attitude Controller vs Time. 73
Figure 4.5 Velocity signal controller. 73
Figure 4.6 3D Trajectory for second test 74
Figure 4.7 Simulation resutls for Position and Attitude Controller in second test 75
Figure 4.8 Velocity signal controller for circular trajectory 75
Figure 4.9 Tracking Error for circular trajectory 76
Figure 4.10 3D Trajectory test for linear model 81
Figure 4.11 Simulation resutls for Position and Attitude Controller for linear State-Space Controller 81
Figure 4.12 Velocity signal controller for linear State-Space Controller 82
Figure 4.13 Tracking Error for linear State-Space Controller 82
Figure 4.14 3D Trajectory test for linear model for linear State-Space Controller 83
Figure 4.15 Simulation resutls for Position and Attitude Controller in linear model 83
Figure 4.16 Velocity signal controller for circular trajectory in for linear State-Space Controller 84
Figure 4.17 Tracking Error for circular trajectory for linear State-Space Controller 84
Figure 4.18 Attitude Controller for Adaptive linear controller 85
Figure 4.19 3D Trajectory test with linear model for Adaptive linear Controller 86
Figure 4.20 Tracking Error for altitude and attitude controller in non-linear model. 90
Figure 4.21 Control input signals (T, τ_x, τ_y, τ_z) 91
Figure 4.22 Parameters estimation 91

List of tables

Table 1.1 UAV applications in Precision Agriculture. 26
Table 1.2 Summary of discussed control techniques for Quadrotors. 32

Table 3.1 Vanishing Point recognition. 62

Table A.1 Mechanical constants Parameters of quadrotor 103

Nomenclature

$\mathcal{F}_a$	Orthonormal frame a.
$\vec{x}_a$	Unit vector of the a frame in x-axis.
$\vec{y}_a$	Unit vector of the a frame in y-axis.
$\vec{z}_a$	Unit vector of the a frame in z-axis.
$\mathcal{F}_I$	Earth-fixed inertial reference frame
$\mathcal{F}_B$	Body-fixed reference frame
$\mathcal{I}$	Diagonal inertia matrix of quadrotor
I_{xx}	Inertia about the x-axis [kg m^2]
I_{yy}	Inertia about the y-axis [kg m^2]
I_{zz}	Inertia about the z-axis [kg m^2]
η	angular position (RPY Euler angles) in the body frame.
ξ	linear position in the inertial frame
q	quadrotor generalized position and orientation vector
${}^{I}x$	quadrotor linear position along x-axis, in the inertial frame
${}^{I}y$	quadrotor linear position along y-axis, in the inertial frame
${}^{I}z$	quadrotor linear position along z-axis, in the inertial frame
ϕ	roll angle [rad]
θ	pitch angle [rad]
ψ	yaw angle [rad]
v	quadrotor linear velocity vector, in the body frame
${}^{B}\omega$	quadrotor angular velocity vector, in the body frame
${}^{B}\dot{x}$	quadrotor linear velocity along x-axis, in the body frame

${}^{B}\dot{y}$	quadrotor linear velocity along y-axis, in the body frame
${}^{B}\dot{z}$	quadrotor linear velocity along z-axis, in the body frame
p	quadrotor angular velocity around x-axis, in the body frame
q	quadrotor angular velocity around y-axis, in the body frame
r	quadrotor angular velocity around z-axis, in the body frame
m	mass of quadrotor [kg]
b	aerodynamic friction coeficient
$\vec{x}_{_I}$	Unit vector in the x-axis of the inertial frame.
$\vec{y}_{_I}$	Unit vector in the y-axis of the inertial frame.
$\vec{z}_{_I}$	Unit vector in the z-axis of the inertial frame.
τ	body torques from motors [N-m]
τ_x	quadrotor torque around $x-$axis, in the body frame.
τ_y	quadrotor torque around $y-$axis, in the body frame.
τ_z	quadrotor torque around $z-$axis, in the body frame.
Ω_i	Angular speed of rotor i.
f_i	aerodynamic force produced by the i-th rotor.
M_i	aerodynamic moment produced by the i-th rotor.
l	length of the arm.
α	angle between the perpendicular with $x-$axis for PHT
$\ddot{\xi}$	quadrotor linear acceleration in inertial frame
$\ddot{x}$	quadrotor linear acceleration along $x-$axis in inertial frame
$\ddot{y}$	quadrotor linear acceleration along $y-$axis in inertial frame
$\ddot{z}$	quadrotor linear acceleration along $z-$axis in inertial frame
$\Delta\Omega$	rotor speed increasing (or decreasing) coefficient.
$\dot{\phi}$	quadrotor angular velocity around $x-$axis in inertial frame (roll)
$\dot{\psi}$	quadrotor angular velocity around $z-$axis in inertial frame (yaw)

$\dot{\theta}$	quadrotor angular velocity around $y-$axis in inertial frame (pitch)
$\dot{p}$	quadrotor angular acceleration around $x-$axis in bodyframe
$\dot{q}$	quadrotor angular acceleration around $y-$axis in bodyframe
$\dot{r}$	quadrotor angular acceleration around $z-$axis in bodyframe
$\dot{x}$	quadrotor linear velocity along $x-$axis in inertial frame
$\dot{y}$	quadrotor linear velocity along $y-$axis in inertial frame
$\dot{z}$	quadrotor linear velocity along $z-$axis in inertial frame
ϕ_{des}	desired roll angle
ψ_{des}	desired yaw angle
ρ	length of the perpendicular drawn from the origin for PHT
θ_{des}	desired pitch angle
ς_{des}	desired state position and orientation for linear system
${}^{B}e_{x}$	error position in $x-$axis in body frame
${}^{B}e_{y}$	error position in $y-$axis in body frame
${}^{B}e_{z}$	error position in $z-$axis in body frame
${}^{B}x_{des}$	desired position along $x-$axis in body frame
${}^{B}y_{des}$	desired position along $y-$axis in body frame
${}^{B}z_{des}$	desired position along $z-$axis in body frame
${}^{I}e_{x}$	error position in $x-$axis in inertial frame
${}^{I}e_{y}$	error position in $y-$axis in inertial frame
${}^{I}e_{z}$	error position in $y-$axis in inertial frame
${}^{I}x_{des}$	desired position along $x-$axis in inertial frame
${}^{I}y_{des}$	desired position along $y-$axis in inertial frame
${}^{I}z_{des}$	desired position along $z-$axis in inertial frame
A	constant matrix
B	Red channel in RGB color space

B	constant matrix
C	constant matrix
D	constant matrix
f	focal length
G	Red channel in RGB color space
g	acceleration due to gravity (9.81 $m\ s^{-2}$)
$K_{D,\phi}$	pitch derivative gain
$K_{D,\psi}$	yaw derivative gain
$K_{D,\theta}$	pitch derivative gain
$K_{D,p}$	angular acceleration around $x-$axis derivative gain
$K_{D,q}$	angular acceleration around $y-$axis derivative gain
$K_{D,r}$	angular acceleration around $z-$axis derivative gain
$K_{D,x}$	x position derivative gain
$K_{D,y}$	x position derivative gain
$K_{D,z}$	z position derivative gain
$K_{I,z}$	z position integral gain
$K_{P,\phi}$	pitch proportional gain
$K_{P,\psi}$	yaw proportional gain
$K_{P,\theta}$	pitch proportional gain
$K_{P,p}$	angular acceleration around $x-$axis proportional gain
$K_{P,q}$	angular acceleration around $y-$axis proportional gain
$K_{P,r}$	angular acceleration around $z-$axis proportional gain
$K_{P,x}$	x position proportional gain
$K_{P,y}$	y position proportional coefficient
$K_{P,z}$	z position proportional gain
p_{des}	desired angular velocity around $x-$axis in the body frame

q_{des}	desired angular velocity around $y-$axis in the body frame
R	Red channel in RGB color space
R_B^I	rotation matrix from the body frame $\mathcal{F}_B$ to the inertial frame $\mathcal{F}_I$.
r_{des}	desired angular velocity around $z-$axis in the body frame
u_1	vertical thrust respect to the body frame
u_2	roll torque respect to the body frame
u_3	pitch torque respect to the body frame
u_4	yaw torque respect to the body frame
U_{eq}	input system at equilibrium point
X	vector state of the system
X_{eq}	equilibrium point at hovering

1
Introduction

In these last years, several industries require more robots to replace men in dangerous or heavy works for different tasks, and with the development in microelectronics and increase of the computing efficiency, aerial vehicles have encountered a significant focus among robotics. There are several structures and aerial configurations to perform the flight movement with greater freedom such as blimps, fixed-wing planes, single rotor helicopters, bird prototypes, and quadrotors. Features such as size, speed, and maneuverability vary widely as a function of the application and help to classification aerial vehicles [1, 3, 4].

Currently, with the technology and the variety of tasks, aerial vehicles may have different levels of autonomy: remote-controlled, semi-autonomous, and intelligent systems, or fully autonomous. The main part of an autonomous UAVs is the navigation system utilizes information from other subsystems to reach a position, or identify obstacles and act in consequence, and send commands to stabilize the attitude and follow guidance objectives [1, 2, 5, 29].

In this first chapter we will present types of aerial vehicles, platforms and applications in different areas. Also the motivation deals to develop this project, and review of others related projects.

1.1
Classification of UAVs

Aerial vehicles can be categorized by their performance characteristics. Features including weight, range, maximum altitude, speed, endurance, and production costs, are important design parameters. For example, the type of propulsion systems that are used in UAVs is different based on their models, and the number of engines that used is another kind of classification [1, 13].

Based on the aerodynamics configuration, there are three main classes of aerial vehicles as shown the Figure 1.1. The review of literature shown there is another type of aircraft as know as convertible aircraft. Each of them has advantages and drawbacks, the main drawback of aerial vehicles is the high energy consumption [9, 12, 15, 29].

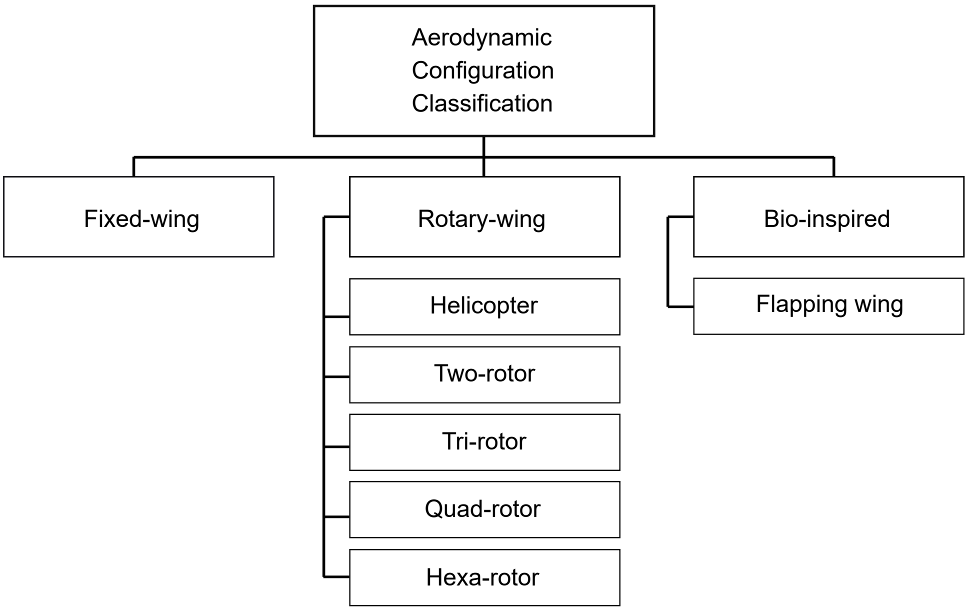


Figure 1.1: UAVs classification based on aerodynamic configuration

1.1.1 Fixed-Wing Aircraft

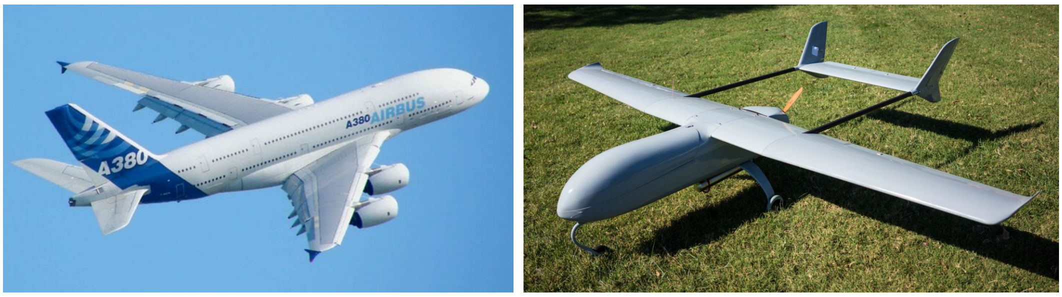
A fixed-wing aircraft consists of a rigid wing, fuselage, and tails which use a motor and propeller as their propulsion system. They use their wings to generate lift force so it compensates for the weight of the vehicle [1, 13].

They require a run-way to take off and land, they can fly for a long time at high altitude and at high speed. Fixed-wing aircraft has longer range and endurance and cover larger areas and has a fast flight speed [9, 29]. They are used mainly in scientific application such as meteorological and environmental monitoring [3, 12]. Figure 1.2 shown some fixed wing aircraft.

1.1.2 Rotary-wing aircraft

Rotating wing UAVs have the advantage above fixed-wing UAVs that they can fly in every direction, horizontally, vertically, and also can hover in a fixed position with high maneuverability [2, 42].

- **Single-rotor**: they have a main rotor on top and another rotor at the tail for stability, same as the helicopter configuration as shown in Figure (1.3)(a-b).
- **Quadrotor**: a very successful design for smaller UAVs. They have two pairs of counter-rotating rotors. The first and the third rotors rotate clockwise whereas the second and the fourth rotors rotate anticlockwise.



1.2(a): Airbus A380 aircraft

1.2(b): Albatross aircraft

Figure 1.2: Some examples of fixed-wing aircraft (a) Commercial aircraft [31], (b) Albatross is an aircraft well-suited to projects where long flight times is required, this aircraft is perfect to surveillance and monitoring [33]

They can be controlled by varying the speed of the rotors and thus fixed-pitch blades can be used, which simplifies the design and control of the vehicle [13, 37, 42, 61]. Figure (1.3)(c) shows a commercial quadrotor using for entertainment and topographic assessment.

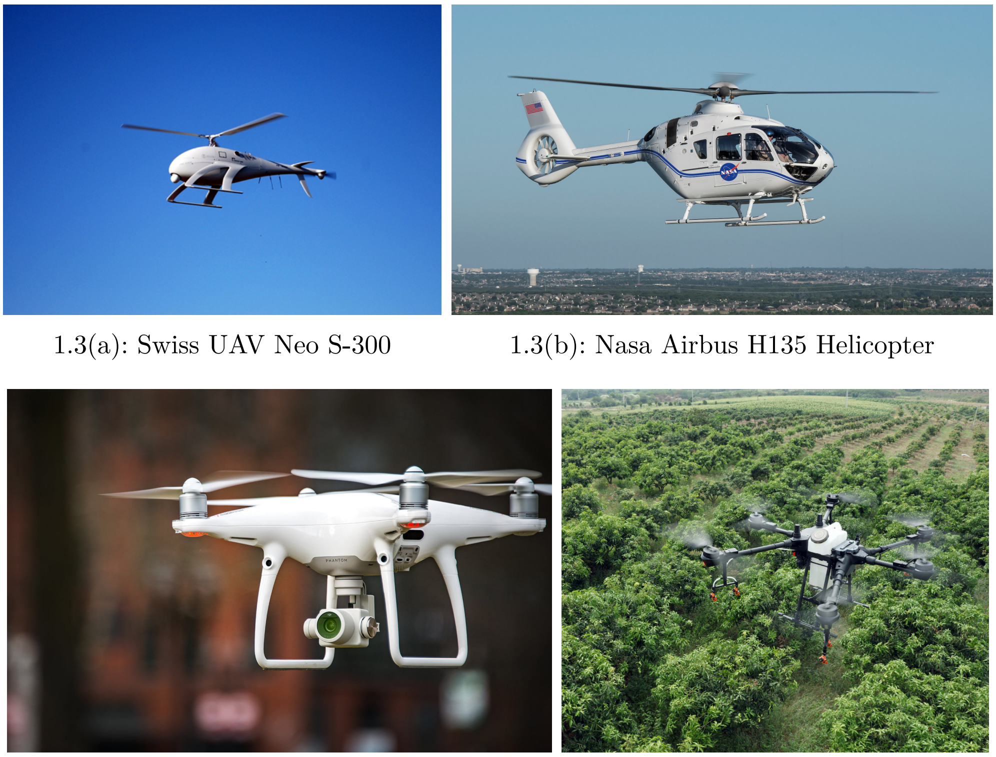
Quadrotor has several advantages since they are small, compared with a single-rotor, its rotor is smaller and highly improved the security of flight; they are low cost, and the principal advantage its vertical take-off and landing (VTOL) can easily realize static state hovering. Among the disadvantages of quadrotors, the main is their poor energy performance as compared to helicopters with similar sizes and weights. [1, 9, 42]

In practice, most quadrotor are symmetric around their center of mass (CoM), i.e. the distances between each rotor to the quadrotor CoM are equal. [1]

- **Multi-rotor**: Rotary wing with two, three, five, six and eight motors are called twin-rotor, tri-rotor, penta-rotor, hexa-rotor, and octo-rotor. Among rotary wing UAVs, the quadrotorss and hexa-rotors are the best known drones.

In contrast, rotary-wing have lower speed, shorter flight duration, and limited payload capacity. Their agile maneuvering, ability to hover around a particular area and the ability to operate in confined areas make them ideal for surveillance and for detecting crop pests, monitoring, diseases, and weeds. [9, 12, 16]

As in the case of fixed-wing aircraft, rotary aircraft are usually under-actuated systems. For example, the standard quadrotor has four independent control inputs (four propellers spinning velocity) versus 6 DoFs parameterizing the quadrotor position/orientation in space. Due to this limited mobility, other projects add more DoFs using tilting propellers to improve the versatility of the quadrotor. Different types of rotary wings are presented in Figure 1.3



1.3(a): Swiss UAV Neo S-300

1.3(b): Nasa Airbus H135 Helicopter

1.3(c): DJI Phantom 4 Pro

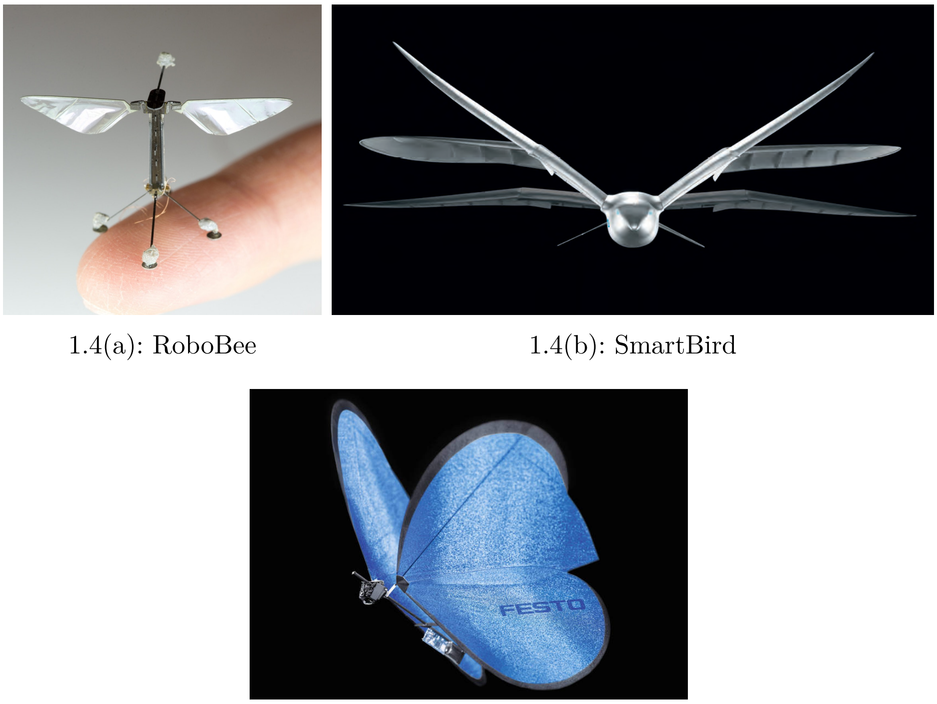
1.3(d): DJI Agras MG-1P

Figure 1.3: Some examples of rotary wing aircraft (a-b)Conventional helicopter configuration [34, 32]. (c)Quadrotor configuration [35]. (d) Hexarotor configuration[36].

1.1.3
Bio-inspired aircraft

Different animals from insects to birds are capable to do flight maneuvers that are now impossible in aerial vehicles, such as flying in turbulence or cluttered airspace. Animals are more maneuverable and can fly longer distances. Due to these, most studies searching optimal wing for hovering bio-inspired micro air vehicles (MAVs) assumed flat and rigid wings. There are researchers who have made attempts at building flapping robots. While several are successful, others either never take off or fly for a short time duration due to their higher complexity or poor design, the results in researchers suggest that the best performing wing should have nearly straight leading edges with large surface area [1, 10, 11].

While flapping wing aerodynamics is complex and not fully understood, it is simple from a robot design perspective, because it is scalable from insect to bird size. Flapping wings generate more lift than translating wings because they generate a stable leading-edge vortex that enables the wing to operate at high angles of attack [1, 11].



1.4(a): RoboBee

1.4(b): SmartBird

1.4(c): eMotionButterflies

Figure 1.4: Bio-inspired aircraft robots. (a) Insect-inspired robots that could perform myriad roles in agriculture or disaster relief, Robobee [10]. (b) Smartbird, robot inspired by the herring gull [88]. (c) Bionic butterflies, the project combine the ultralight construction of artificial insects with coordinated flying behavior in a collective [88].

1.2 Applications of Aerial Robots

The UAVs have been used for different civilian and military applications. These vehicles are able to carry different kind of sensors based on the type of their missions and to enhance their performance and efficiency, researches have focused on the design, development, and fabrication of various types of UAVs with diverse capabilities. In the literature review several applications were found:

– **Surveillance**, is an application in which the UAVs support public safety, search and rescue operations, and disaster management. I case of natural disasters like floods, earthquakes, tsunamis, or terrorist attacks, transportation, and telecommunications systems can be partially or fully affected by a disaster. UAVs can provide timely disaster warning and assist in speeding up rescue and recovery operations. UAVs can also carry medical supplies to areas that are classified as inaccessible or it is too away. The UAVs can quickly provide coverage of a large area without ever risking the safety of the person involved. [21, 27, 30]

- **Construction**, is another application, using aerial robots that lift building elements and position them at their precise locations. Possibly more than one UAV is required to lift the structures and it becomes a distributed processing through by a team of heterogeneous UAVs. In this case, depending upon the payload capacity of the robot and the weight of the building block in order to estimate the number of required robot [3, 12, 29].
- **Transport** is another interesting application. UAVs can be used to transport food, packages, and other goods. The aerial vehicle is equipped with a control processor and a GPS module. With the massive growth of e-commerce, postal companies have been forced to find new methods to expand beyond their traditional mail delivery business models, different companies have undertaken various trial test, such as Amazon's Prime Air project is an attempt to use UAVs for delivery of commercial products [3, 16, 29].
- **Monitoring** collect information about traffic on roads, the UAV can monitor large continuous road surfaces or focus on a specific road. In mining industry UAVs can make a quick inspection of an area in order to identification an emergency situation, or make 3D mapping [3, 9, 30].

All applications need the information of other subsystems to reach the objective, these processes utilized remote sensing system to processing videos/images for object detection, image transmission based on the data gathered by data sources such as cameras [6, 20, 25]. The object detection depends on the image resolution and the visibility of the target. Often times the mission goal states, that once the target is detected, then it is being tracked by a UAV, in which collision avoidance and navigation methods may be used [65, 66]. Collision avoidance approaches have been proposed in order to avoid potential collisions by UAVs, methods such as path planning approaches, potential field approaches, and vision-based approaches [74]. Several collision avoidance techniques can be utilized for an indoor environment, methods using cameras optical sensors, onboard sensors using IR, ultrasonic, laser, scanner, etc [9, 29].

1.2.1 Precision Agriculture

Precision agriculture is considered an area in which the use of aerial robots may become a key factor in the growth and improving the production quality, it has become an area that has attracted great attention [12]. Medium size drones are used for analysis applications, while larger drones are used when there are activities that need to carry a load [4, 7].

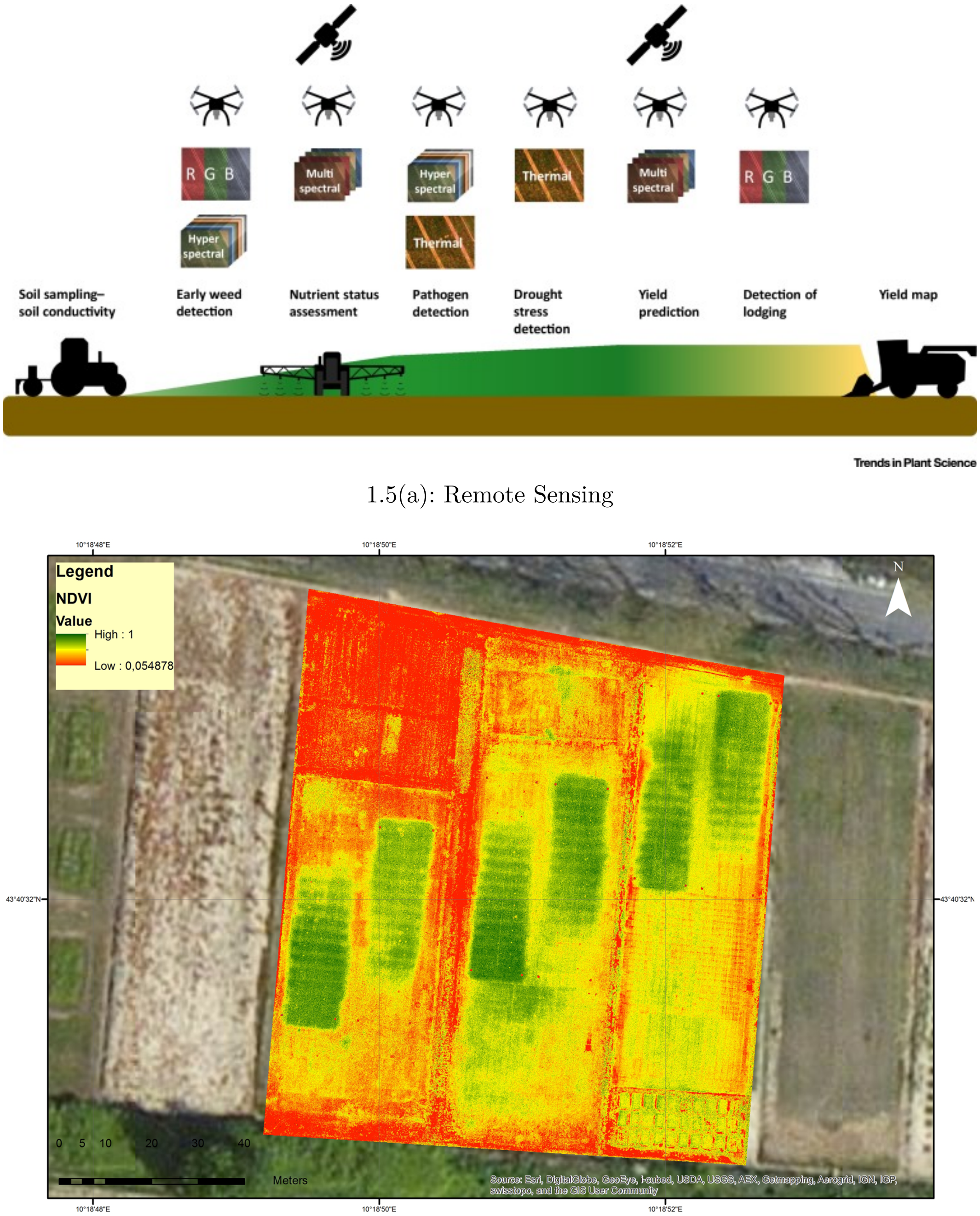
Nowadays, in agriculture, multirotor drone configurations seem to be the favorite due to their lower cost and high level of simplicity and maneuverability [9]. The skillfulness of UAVs provides different ways in order to improve agricultural processes like soil and field analysis, crop monitoring, health assessment, irrigation, crop spraying, and seed planting [4, 5, 8, 9, 12, 16].

UAVs, are able to produce maps, quickly and cheaply, which are then used for the design of seed-planting patterns and the generation of a wide range of data types with many applications. In crop monitoring, working with satellite image was previously the most advanced form, but there are some major drawbacks as imagery is very costly and can be imprecise, also poor weather affects data quality [6, 20]. Figure 1.5(a) shows the role of UAVs for remote sensing with two different aircrafts (multirotor and satellite).

On the other hand, UAVs can also be monitoring crops much more accurately, because they can fly lower and close to the field, frequently and affordably, delivering higher quality data that is updated regularly to provide vision into crop development and highlight inefficient practices [12]. For example, nitrogen status management or identification of planting rows and crop monitoring, as shown in the Figure 1.5 (b) [4, 17, 18].

The various sensors generally integrated with UAVs can be divided into two broad groups based on the role of the sensor in the functioning of the aerial vehicle. First group, sensors are directly integrated into the UAV and affect the functioning and flight of the UAV such as accelerometers, gyroscopes, Global Positioning System (GPS), temperature sensors, and barometer can be broadly used for localisation, and measure of the motion. Second group, sensors are not linked directly to the functioning and controlling of the UAV and can be changed based on the UAV's application, sensors as temperature sensors, color sensor, RGB camera, hyper-spectral camera, multispectral camera, Light Detection and Ranging (LiDAR) sensor, thermal imaging camera, and gimbals-based stabilization sensors. Various sensors are needed for the operation of UAVs in agriculture.

Table 1.1 outlines some optimization objectives, and methodologies that have applied to improve the UAV solutions in precision agriculture, and involve



Trends in Plant Science

1.5(a): Remote Sensing

Source: Esri, DigitalGlobe, GeoEye, i-cubed, USDA, USGS, AEX, Getmapping, Aerogrid, IGN, IGP, swisstopo, and the GIS User Community

1.5(b): Nitrogen Status

Figure 1.5: Precision Agriculture: (a) The Role of Unmanned Aerial Vehicles (UAVs) for Assessing Field and Crop Status Spatially, Schematic overview of the different ways to extract spatial information, the useful platforms, and the optimal UAV sensors, throughout a growing season in a wheat crop [6]. (b)The RGB image of the turfgrass fields acquired by Tetracam ADCMicro mounted on the unmanned aerial vehicle [20].

the different sensors listed above.

There is no universal agriculture solution for every crop, every crop is different, each crop needs to be measured differently to generate actionable data. Therefore specialists are needed for each particular crop type, and they will ultimately deliver different solutions to farmers.

Table 1.1: UAVs applications in Precision Agriculture.

Objective	**UAV type**	**Methodology**
Spray system	Helicopter	Developing architecture enables a UAV to adjust its route to the concentration of deposited pesticides. Using Wireless Sensor Networks (WSN) to deploy the information in a matrix format covering the crops in the field of a UAV in different weather conditions for control pest. [7]
Manage crop	Octarotor	Collecting information about the crop, framework with 4 modules including job management, monitoring settings, map module, and database management.[5]
Remote sensing	Fixed-wing/eBee Ag	Developing a method for mapping skips in sugarcane fields using UAV images. High-resolution UAV images provide accurate information on sugarcane skip incidence. [18]
	MK-Oktokopter	Three cameras (RGB, color infrared (NIR-G-B), and multispectral (MS) camera) were mounted on a UAV to rice plant detection in the early season. The algorithm classifies into rice plants, road, and background (soil, water, and duckweed)[19].
	micro UAV	Infrared camera for estimating biomass of willow to get information and Normalized Difference Vegetation Index (NDVI).[23]
Monitoring	Octorotor	From data obtained from the imagery of turfgrasses. A GreenSeeker crop sensor and a multispectral sensor were equipped in a UAV, to determine NDVI and monitor plant nutrition, reduce nitrogen (N) application.[17]
	Hexa-rotor/ VulcanHex	Photo-interpretation of crop growth using VNIR wavelengths of the system, vegetation indices will be computed to reveal several insights in plant growth such as nitrogen concentration, photosynthesis, transpiration, and crop yield.[24]
	quadrotor/Ar Drone	Developing a framework for monitoring and inspection using single cameras exchange information between UAVs.[25]
Water stress	Quadrotor	Data spatial information of multispectral images of the ornamental crop, a relatively small area of production, in order to evaluate levels of water. [26]
	Fixed-wing	Showing that a 0.3 m per pixel was needed to produce precise “crop water stress index” (CWSI) mapping in a vineyard. [28]

1.3 Motivation

The quadrotor's structure is attractive for several application particularly in surveillance, imaging sensing [6, 21], indoor and outdoor navigation and mapping [3, 12, 8, 18]. There are several researches were conducted on localization, navigation, obstacle avoidance, etc [38, 53]. On the other hand, the use of aerial robots like quadrotores for agriculture services has increased in recent years, especially by their flexibility, mobility and low-cost of ownership and operation [9].

The most common operations performed by aerial robots in agriculture are detection of plants and soil parameters, spraying chemicals and repellents, and detection of wild animals or damage they can cause [8, 9]. Some previous works have shown the use of different types of sensors on quadrotors, hexacopters or other UAVs to navigation [9, 6, 15]. It was decided from the beginning of this dissertation to work on a particular configuration: the quadrotor. The interest comes not only from its dynamics, which represent an attractive control problem. It supports a wide range of payloads and offers the type of under-actuated system that is still manageable for a lot of well-established control approaches [43].

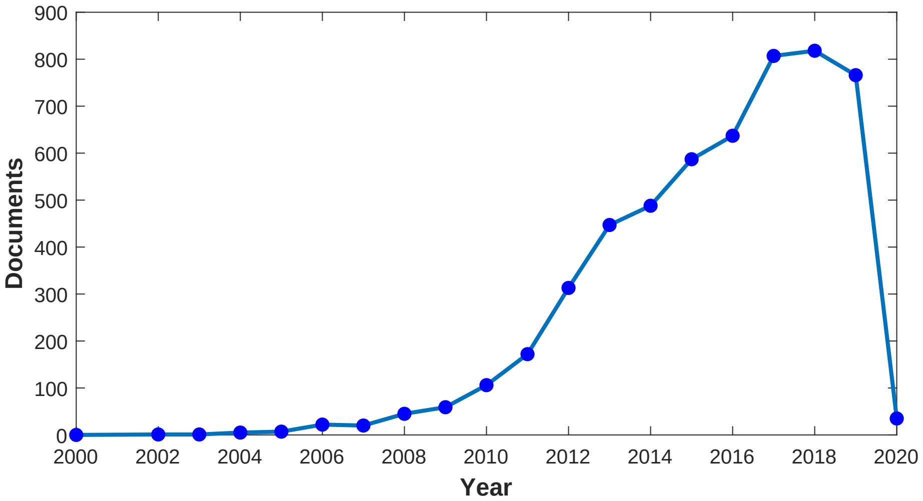


Figure 1.6: Research articles on Quadrotors.

Figure 1.6 presents the increment of quantity research papers on Quadrotors from 2000 to 2019. Notice that, during the last decade, the number of scientific publications increased almost five times. This Figure was based on the query of the Scopus research database. The search includes several research domains such as Imaging, Networks, Swarms, Localization, Path Planning, Mapping, Stabilization and Controls, and Applications.

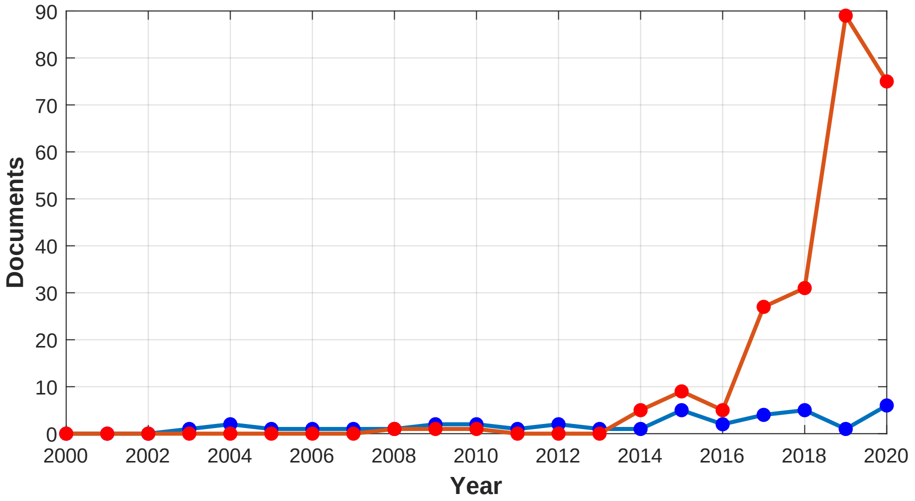


Figure 1.7: Search analysis in Scopus for the past 20 years: The red plot represents the number of researches per year for principal key words: "drones" and "precision agriculture". The blue plot is the number of researches per year for principal key words :"Wheel robot" and "precision agriculture".

In recent years, in precision agriculture, aerial robots have become an attractive research area on wheeled robots, as illustrated in Figure 1.7, for the last 19 years, the number of scientific papers for drones has increased compared to wheeled robots. This increase is due to an important characteristic, the size of UAVs allows advantages of greater agility and maneuverability than wheeled robots. These advantages lead to their largest drawbacks, which are the limited onboard payload, reduced flight, disturbances by the wind, and limited computational resources.

In crop monitoring and spraying, some parameters of the UAV model may be changed. Given the wide variety of sensors that can use to perform a specific task, the unknown parameters of the environment, or the payload of multirotor, all involve adjustment in the model, so that the multirotor fulfills a mission in a robust and autonomous way. So, a controller that adapts the unknown parameters could be ideal, with a navigation algorithm non-complex to move inside a crop field.

1.4
Review of the State of the Art

The state of the art in quadrotor control has drastically changed in the last few years. The number of projects has suddenly increased (Figure 1.6). Most of these projects are based on commercially available platform like ArDrone platform[47]. Many others focused on designing and modeling their own quadrotor as OS4 [38] or X4yer modeled in [53] . A number of open-source projects for quadrotors are shown in [15]. In most of the selected configurations, each pair of blades spins in the same direction and is implemented on one of the axes of the body frame coordinate system. Newton-Euler formulation or Lagrange equation are the classical methods used to modeling and getting the differential equations [57].

1.4.1 Quadrotor models

There are several configurations of quadrotors, starting from the traditional + - shaped to design × - shaped structure. A quadrotor is a rotary-wing UAV, controlled by varying the speed of each rotor. In most of the selected configurations, each pair of blades spins in the same direction and its implemented on one of the axes of the body frame coordinate system. All propellers' axes of rotation are fixed and parallel and the airflow down to lifting upwards [13, 43, 57].

There are a several works about modelling of quadrotors and the derivation of all forces, torques and differential equation of motion. One outcome of these models is a static model for thrust and rotor torque as the square of rotor speed [39, 40, 41]. It is important to note that modelling and control strategies applied to quadrotor can not be applied directly on others UAV as helicopter, due to the many differences between each other [1]. The mathematical model should be developed based on the chosen configuration. So, modelling help to analyse the system and develop control algorithms.

In order to obtain a realist model of the quadrotor, researchers consider the forces produced by the four motors, the gravity, the gyroscopic effects, drag forces and moments in order to obtain a nonlinear model [57]. In [38] and [42], blade flapping and total thrust in hub forces and moments for their model are quantified. However, some researchers do not include hub and roll forces for their controlling models simplifying the system to a linear model[44, 47, 50].

1.4.2 Quadrotor controllers

In practice, a control system depends on a lot of factors. The rotary-wings need motor parameters and aerodynamic parameters given by the environment and rotor blades [1]. Some common linear and nonlinear methods were reviewed for quadrotor control and we realized the combination of these techniques for the quadrotor dynamic model are gain good performance flight.

The linear control consists of a complicated algebraic manipulation for state variables under certain conditions. Linear control can be applied only if the trajectory and the flying conditions for the quadrotor are not complex and difficult. However, the flight controller for rotary-wing aircraft are an inherent nonlinear system, the quadrotor is open-loop unstable, which requires a very fast control response and a large operation range [43, 57].

Researches such as [38, 45, 51, 53] and [60] used a Proportional Integral Derivative (PID) for fully autonomous quadrotor to stabilize the attitude of the quadrotor. In [38] a complete model was derived by Euler-Lagrange formulation including gyroscopic effects and aerodynamic effects and the results were compared with a Linear Quadratic (LQ) controller. The PID controller showed the ability to control the orientation with small perturbations. Some difficulties of the system were the inaccurate model that neglected some of the couplings in the LQ controller.

In[45], a quadrotor trajectory tracking control for outdoor was developed in which the measurements by the GPS sensors are noisy and imprecise. In the same way as previous work, there are other researches as [49, 53] and [61], where proposed use PID to stabilize the position, and attitude were controlled with other controllers.

So, we concluded that PID is efficient in hovering but the performance decrease with the presence of high disturbance. The controller parameters must be tuned by trial and error so this is a time-consuming process. On the other hand, [52] used a self tuning PID controller based on adaptive pole placement to control the attitude and heading of a quadrotor. Simulation showed that the proposed controller performed well with online tuning of the parameters.

Under non-linear approach there are a large body of papers developing nonlinear control theory for the autonomous quadrotors. These approaches include, feedback linearization, backstepping control, sliding mode control, and adaptive control to mention only a few.

Backstepping is a recursive algorithm, which decomposes the controller into several steps and makes each step stable progressively [41]. In [38] backstepping technique is applied using position tracking-errors and the Lyapunov theorem. With this approach, three inputs are extracted for control of the rotation, and one input is obtained for the altitude controller. Similarly, in [?], altitude tracking control was accomplished using a backstepping controller.

Sliding mode control(SMC) is an algorithm works by applying a discontinuous control signal to the system to command it to slide along a prescribed path. Its main advantage is that it does not simplify the dynamics through linearization and has good tracking. The sliding mode controller was applied to stabilize cascaded under-actuated systems in [46, 47] In [38], a SMC was designed in which the sliding surface was defined and verified using the Lyapunov theory. At the final step, control inputs were extracted using a backstepping approach. The proposed SMC was compared with the backstepping controller, and demonstrated that the backstepping controller was better than the SMC because of the switching nature of the controller.

Other works as [37, 47] and [49] using SMC to make the quadrotor fly to the desired position with a desired heading angle. Their simulation results presented that the proposed controller successfully achieved their goal with acceptable outputs even when the simulation was conducted with parametric uncertainties.

Finally, we have the Adaptive control, this technique are aimed to adapt uncertain or time varying parameters with changes in the system [41]. There are different kind of adaptive controllers as: Adaptive Pole Placement Control(APPC), Robust Control (RC), Direct and Indirect Adaptive Control, and Model reference adaptive control (MRAC).

In the literature, there are already works regarding adaptive control. In order to control each one of the angles that define the attitude of the quadrotor, a discrete-time direct technique was proposed in [44]. An identifier was introduced in the baseline MRAC, called augmented–MRAC, in order to cope with the parametric uncertainties, and simulation results demonstrated that the proposed controller improved the stability. A continuous time-varying adaptive controller, which shows good performance, was implemented by [46]. This research considers all quadrotor parameters bounded but fully unknown. To overcome the under-actuation problem, the system has been split into three sub-systems (roll, pitch, and yaw) with fully actuation properties. In [50] a controller design is provided considering the position and attitude dynamics as outer and inner loops respectively. Reference signals for roll and pitch angles are obtained from the outer loop controller while the inner loop controller drives the tracking error of roll and pitch angle dynamics to zero. The stability is proved using a Lyapunov function. In [62], the attitude tracking system was designed for a small quadrotor UAV with uncertainties on the rotors through the model reference adaptive control method.

The linear control methods are simple and low cost, but the flight performance and robustness are less effective than the nonlinear control methods. Each technique has different advantages and disadvantages as shown in the Table 1.2. The quadrotor controller for flying missions is still a challenge. The flight performance is affected by external disturbances such as noise, wind, and rain; or uncertain parameters such as mass, arm length and payload. So, the robustness for the UAVs is only going to gradually improve.

Table 1.2: Summary of discussed control techniques for Quadrotors.

Controllers	**Advantanges**	**Disadvantanges**	**Authors**
PID	Simplicity Robustness	Low intelligent Low optimal	Dong[37], Bouabdallah[38], Mercado et al.[45], Yin et al.[49], Hernandez et al.[51], Pounds[53], Corke [60], Yali et al. [61].
Feedback linearization	Flexible design Good tracking performance	Poor external disturbance rejection	Mahony et al.[40], Malo et al.[59]
LQR	Optimal method	Not simple Lack of robustness	Bouabdallah[38], Hernandez et al.[51], Faessler et al.[56] Sevkuthan [58].
Backstepping	Fast convergence Can handle disturbances	Poor robustness Needs information about all system states	Bouabdallah[38], Yali et al. [61].
Sliding Mode Control	Insensitive to external disturbances Good tracking performs Good robustness	Chattering phenomenon	Dong[37], Bouabdallah[38], Mercado et al.[45], Xiong et al. [47], Yin et al.[49].
Adaptive control	Good robustness Flexible design High optimization Reduce the influence of uncertainties	Requires exact modelling	Fan et al. [44], Emran et al. [46], Akgun [50], Whitehead et al.[62], Mihailescu et al.[63], Dydek et al. [64].

1.4.3 Vision-Based Navigation

Vision-Based robot navigation is defined as the technique guides a mobile robot to the desired destination or along a desired path in an environment by avoiding static and may be dynamic obstacles, primarily using vision sensors (cameras). In general, vision-based robots have a vision system that perceives the external environment [16, 79].

The ability of obstacle detection and avoidance are relatively complicated to develop, as extracting information from a stream of images from the site can be a complex task to achieve the desired real-time performance with as little computing processing as possible. Accordingly, researches focuses on the use of computer vision to achieve vision-based autonomous mobile robotic systems capable of navigation by logically acting on the sensed data to avoid such obstacles [78, 79]. Hence, a UAV must be aware of its states, including location, navigation speed, heading direction as well as starting point and target location. Several navigation methods have been proposed as inertial navigation, satellite navigation and vision-based navigation.

The most useful sensors for navigation are mainly GPS, accelerometer, gyroscope, and inertial navigation system, all of them help to estimate position and it can be enhance the information by multiple sensor data fusion but a more general approach is obtain from images. Comparing to the GPS, LIDAR sensor, and other traditional sensors, visual sensors (cameras) acquire more information about environment as color, texture and other visual information and works better in indoor scenarios. Navigation with monocular camera is an emerging, highlighting that a camera is a lighter sensor than other sensors, as for example LIDAR sensor [29, 79].

On the other hand, a scenario where we know is possible to navigate without a known map, by extracting distinct features of the environment that has been observed, is a mapless system. Currently, the most used methods in mapless systems are optical flow methods and feature tracking methods. The optical flow is the pattern of objects, surfaces, and edges in a visual scene caused by the relative motion between an observer (an eye or a camera) and the scene. The feature tracking method has become a robust and standard approach for localization and mapping. It primarily tracks invariant features of moving elements, including lines, corners, and so on and determines the movement of an object by detecting the features and their relative movement in sequential images [79].

In precision agriculture, many of the sensors mentioned above are used for plant detection and mapping to support the navigation of autonomous robots for tasks such as cultivation or guiding a combine for harvest operation. The principal problems of the different vision-based techniques of navigation are the processing time, shadows, and illumination [9, 79].

In this dissertation, the navigation problem is approached by monocular vision. The UAV is equipped with a frontal camera and the video extracted from the camera is processed to generate the next control command to follow a row crop or navigate between two rows, Vanishing Point is used in order to find the head direction of UAV.

1.5 Objectives

In order to perform navigation and control of a quadrotor is proposed a methodology for semi-autonomous monitoring crops based on computer vision and control, to be applied to precision agriculture.

- Analyze the mathematical model that describes the dynamics of a quadrotor.
- Design and experiment with the theoretical state-space model for the quadrotor with uncertainties.
- Apply adaptive control technique for linear and non-linear models in quadrotor. Evaluate the controller performance in presence of uncertainties on the model.
- Develop a simulation environment to control the position and orientation of a quadrotor in Matlab.
- Implement a vision-based navigation system for crop fields using vanishing points in Gazebo.

1.6 Methodology

We modeling the quadrotor within the Newton-Euler formalism, and control the quadrotor as a linear and non-linear system. The quadrotor mass is the adapted parameter in the linear system. For non-linear system, we adapting the mass and inertia parameters from the requirements of the kinematics and dynamics with a parameterizing in the body frame that one may simplify the representation of the adaptation scheme.

The motion control problem can be tackled using two different approaches: regulation and tracking. The stability for non-linear control strategy is carried out by using the Lyapunov theory.

For navigation we use a UAV equipped with a front camera and the video extracted from the camera is processed to generate the control command in order to follow a row crop or navigate between two rows.

1.7 Outline of thesis

This work is organized according to the following chapters:

- **Chapter 2**: general UAV kinematics and dynamics models are derived, including a brief discussion on the choice of rotation parameterisation. This section seeks to present a well-known UAV modeling, which has already been validated in numerical simulation, to assist in the design of the control system.

- **Chapter 3:** describes the image processing pipeline in detail and the algorithm used to compute the vanishing points for navigation simulation in ROS.

- **Chapter 4**: shows the development of the control algorithms needed to stabilize the quadrotor. The model of the quadrotor is simplified to allow for the design of a simpler controller and to reduce the complexity of the stability analysis. On the other hand, an adaptive controller as an advanced control technique with robustness properties is developed in order to deal with modeling uncertainties and external disturbances.

- **Chapter 5**: presents the concluding remarks obtained in this work as well as discussion and future works.

2 Background

In this chapter, the kinematic and dynamic models of a quadrotor are introduced. The model of the dynamic equations describing the attitude and the position of the quadrotor are basically those of a rotating rigid body, with six degrees of freedom (DoF) and four control inputs. Therefore, a quadrotor is an under-actuated system due to it has only four rotors to fly in the 3D Cartesian space.

2.1 Preliminaries

The study of the kinematics and dynamics of a quadrotor require the following simplifying assumptions, for more details see for example [60]:

(A1) The structure is rigid and symmetric, with the four arms aligned with the body x- and y-axes. Thus, the inertia matrix I is a diagonal matrix with entries are $I_{xx}=I_{yy}=I_{zz}=\mathcal{I}$, that is:

$$I = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} = \begin{bmatrix} \mathcal{I} & 0 & 0 \\ 0 & \mathcal{I} & 0 \\ 0 & 0 & \mathcal{I} \end{bmatrix},$$

where I_{xx}, I_{yy} and I_{zz} are respectively the area moment of inertia with respect to the x-, y- and z-axes.

(A2) The origin of the body frame is in the center of mass of the quadrotor;

(A3) The propellers are rigid;

(A4) Thrust and drag are proportional to the square of the propeller's speed;

(A5) Inertia of rotors are neglected.

2.2 Nomenclature

The following convenient notation is adopted throughout this work:

(i) $\mathcal{F}_a = [\,\vec{x}_a \;\; \vec{y}_a \;\; \vec{z}_a\,]$ denotes the orthonormal frame a, whereas $\vec{x}_a$, $\vec{y}_a$ and $\vec{z}_a$ denote the unit vectors of the frame axes.

(ii) For a given vector $v \in \mathbb{R}^n$, its elements are denoted by v_i, for $i = 1, \ldots, n$, that is, $v = [\,v_1 \;\; v_2 \;\; \cdots \;\; v_n\,]^\mathsf{T}$.

(iii) $\mathcal{S}(\cdot)$ denotes the skew-symmetric operator. In other words, for any vector $v \in \mathbb{R}^3$ its matrix representation is given by

$$\mathcal{S}(v) = \begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix}.$$

(iv) Sinusoidal functions of a given angle $\alpha \in \mathbb{R}$, i.e., $\sin(\alpha)$, $\cos(\alpha)$, $\tan(\alpha)$ and $\sec(\alpha)$ will be replaced respectively by s_α, c_α, tan_α and sec_α.

2.3 Reference Frames

Before discussing converting orientations from one reference frame to another, it is essential to establish a set of reference frames. There are two main reference frames used to represent a given object that moves in 3D Cartesian space, as depicted in Figure 2.1, the inertial reference frame and the body reference frame. The yellow color on the rotor is used to denote the front of the quadrotor.

2.3.1 Inertial Reference Frame

The earth-fixed inertial reference frame $\mathcal{F}_I$ is a right-handed orthogonal axis with the origin at the center of gravity of the quadrotor at the beginning of the considered motion. This reference frame is fixed to the Earth, its origin can be set everywhere but once chosen it cannot vary usually.

It is convenient to align the axes of the reference frame with the compass; one with the North (N), one with the East (E), and the last with the normal to the surface generated by the previous two-axis, pointing to the center of the Earth and labeled down (D).

2.3.2 Body Reference Frame

The body-fixed reference frame $\mathcal{F}_B$ is a right-handed orthogonal axis-system with the origin at the center of gravity of the quadrotor. The body-fixed reference frame remains fixed to the quadrotor even in perturbed motions.

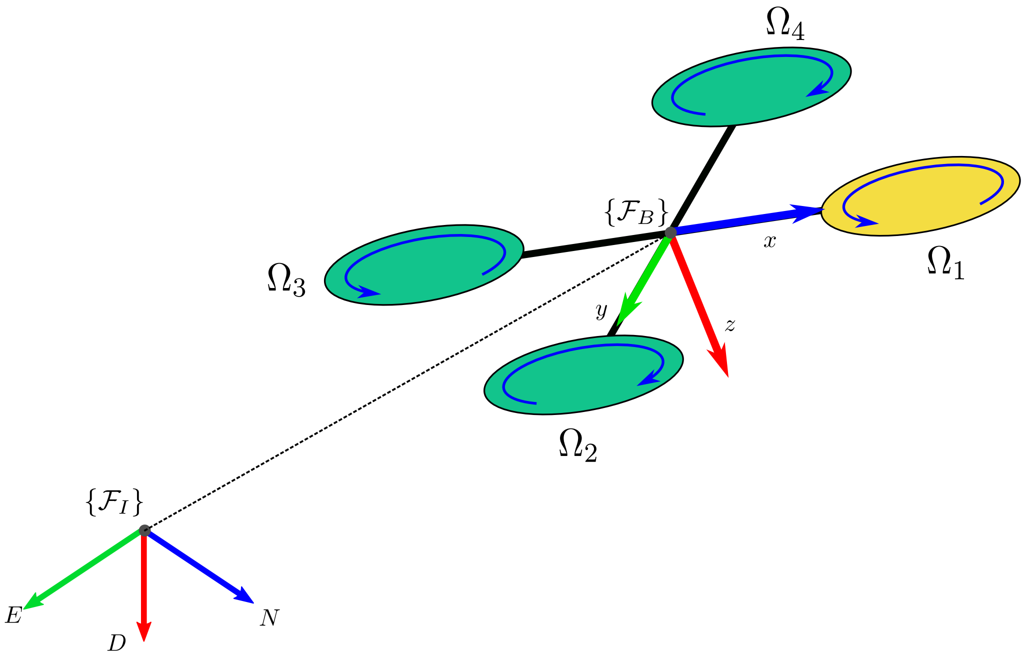


Figure 2.1: Earth-fixed reference frame $\mathcal{F}_I$ and body-fixed reference frame $\mathcal{F}_B$. The yellow color denotes the front rotor of the quadrotor.

Remark 1: To follow the notation commonly found in the literature and to avoid confusing the reader, the quadrotor reference frames will be named as: (i) the inertial frame, denoted by $\mathcal{F}_I$; (ii) the body frame, denoted by $\mathcal{F}_B$.

2.4 Basics of the Quadrotor Motion

For a basic multi-rotor model, the rotors or propellers generate thrust in the negative direction of the z-axis of the body frame $\mathcal{F}_B$. So, in order to move the quadrotor in a certain direction, the propulsion system has to point its overall thrust vector in that direction. The amount of thrust generated by each of the rotors changes autonomously to the desired motion, meaning speeding up or slowing down certain rotors as necessary. From now on, to facilitate understanding, the motors will be listed by motor Ω_i for $i = 1, 2, 3, 4$.

2.4.1 Roll Motion

The roll motion command generates a sideways motion, provided that the quadrotor slightly rotates around its roll axis (x-axis). To achieve this, motor M4 has to generate more thrust than motor M2 so that the torque around the roll axis is generated. Then, Motor 4 needs to speed up and motor 2 needs to slow down, by amounts of $\Delta\Omega_4$ and $\Delta\Omega_2$ respectively, as shown in Figure 2.2.

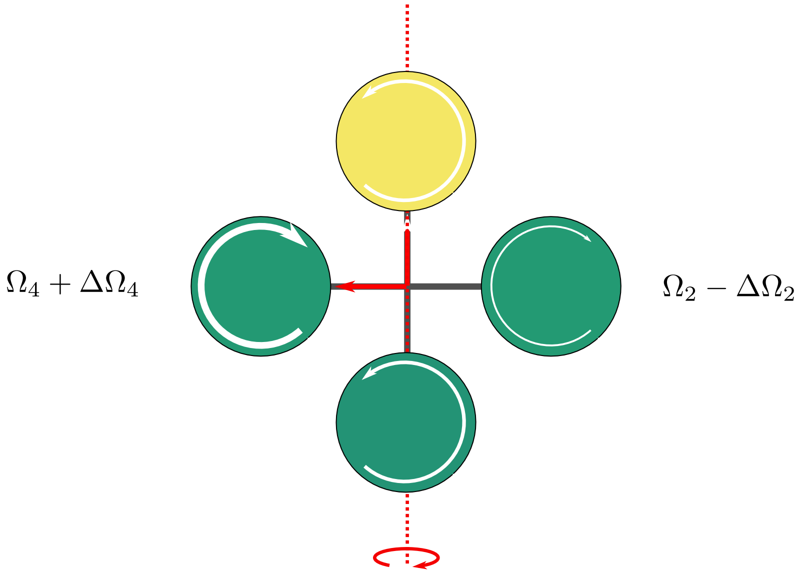


Figure 2.2: In a positive roll motion, the quadrotor moves right.

2.4.2 Pitch Motion

The pitch motion command generates a forward/back motion, provided that the quadrotor slightly rotates around its pitch axis (y-axis). To achieve this, motor 3 has to generate more thrust than motor 1 so that the torque around the pitch axis is generated. Then, motor 3 needs to speed up and motor 1 needs to slow down, by amounts of $\Delta\Omega_3$ and $\Delta\Omega_1$ respectively, as shown in Figure 2.3.

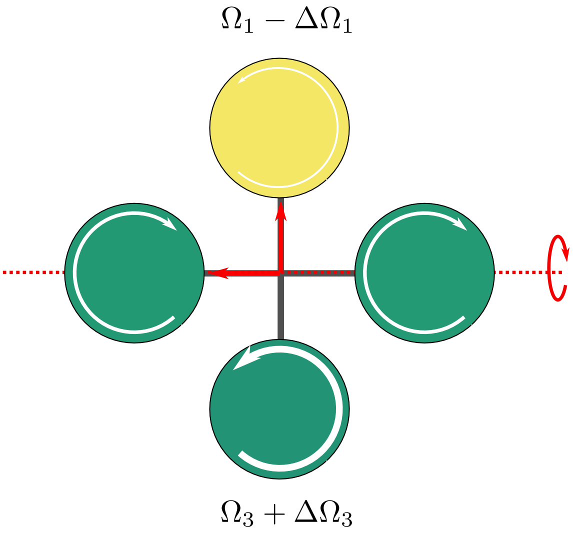


Figure 2.3: In a negative pitch motion, the quadrotor moves forward.

2.4.3 Yaw Motion

The yaw motion command is provided by increasing (or decreasing) the speeds of the front-rear propellers (motors 1 and 3) and by decreasing (or increasing) the speeds of the left-right ones (motors 2 and 4). The resulting motion produces a net torque in the z-axis coordinates of the body frame $\mathcal{F}_B$, making the quadrotor turn around, either clockwise or counterclockwise, as shown in Figure 2.4.

2.4.4 Throttle Motion

The throttle motion command is provided by increasing (or decreasing) all propeller speeds by the same amount $\Delta\Omega_i$ for $i = 1, 2, 3, 4$. The resulting motion leads to the emergence of a net vertical force in the body frame $\mathcal{F}_B$ which counterbalances the gravitational force, raising or lowering the quadrotor. Figure 2.5 shows the throttle command on a quadrotor, where $\Delta\Omega$ is a positive variable that represents an increase respect Ω.

2.5 The Quadrotor Kinematic Model

The kinematics analysis describes the translations and rotations of the body frame $\mathcal{F}_B$ with respect to the inertial frame $\mathcal{F}_I$. The linear position of the quadrotor, expressed in the inertial frame $\mathcal{F}_I$, is defined by $\xi \in \mathbb{R}^3$ in terms of x,

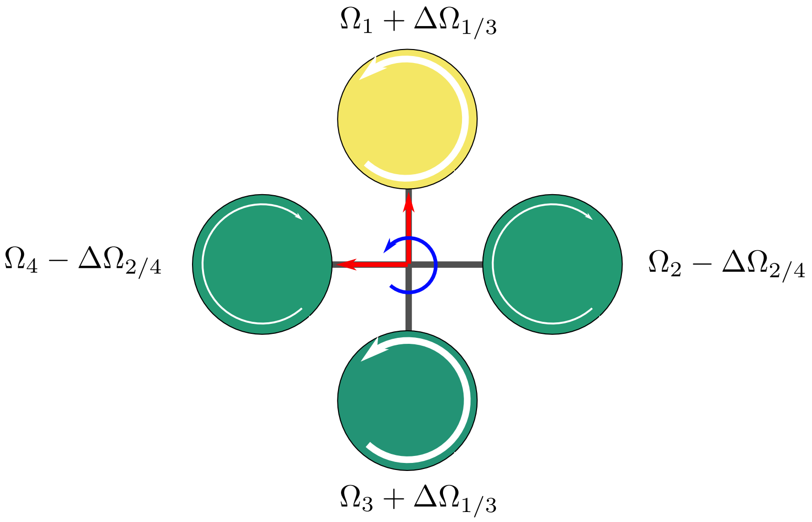

$\Omega_1 + \Delta\Omega_{1/3}$
$\Omega_4 - \Delta\Omega_{2/4}$
$\Omega_2 - \Delta\Omega_{2/4}$
$\Omega_3 + \Delta\Omega_{1/3}$

Figure 2.4: In a negative yaw motion, the quadrotor turns around counterclockwise.

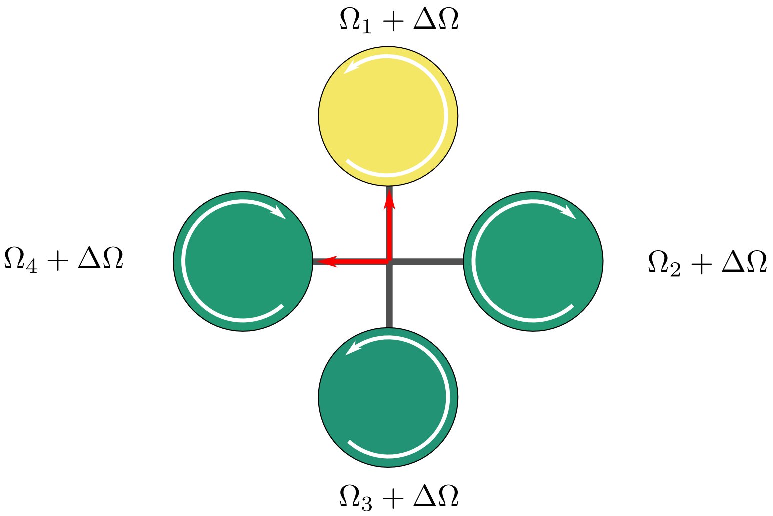

$\Omega_1 + \Delta\Omega$
$\Omega_4 + \Delta\Omega$
$\Omega_2 + \Delta\Omega$
$\Omega_3 + \Delta\Omega$

Figure 2.5: Quadrotor throttle motion.

y, and z-axes coordinates. On the other hand, the angular position (attitude) of the quadrotor, expressed in the body frame $\mathcal{F}_B$, is defined by $\eta \in \mathbb{R}^3$ in terms of RPY Euler angles: the roll angle (ϕ) gives the rotation around the x-axis; the pitch angle (θ) gives the rotation around the y-axis, and the yaw angle (ψ) gives the rotation around the z-axis. Then, the quadrotor generalized pose (position + orientation) vector $q \in \mathbb{R}^6$ is composed of the linear and angular position vectors as:

$$\xi = \begin{bmatrix} {}^I x \\ {}^I y \\ {}^I z \end{bmatrix}, \quad \eta = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix}, \quad q = \begin{bmatrix} \xi \\ \eta \end{bmatrix}. \tag{2-1}$$

In the body frame $\mathcal{F}_B$, the linear velocity vector is denoted by ${}^B v \in \mathbb{R}^3$ whereas the angular velocity vector (or the angular body rates) is denoted by ${}^B \omega \in \mathbb{R}^3$ as:

$$v = \begin{bmatrix} {}^B \dot{x} \\ {}^B \dot{y} \\ {}^B \dot{z} \end{bmatrix}, \quad {}^B \omega = \begin{bmatrix} p \\ q \\ r \end{bmatrix}, \tag{2-2}$$

where p, q, and r denote respectively the quadrotor angular velocities around the x-, y- and z-axes of the body frame $\mathcal{F}_B$.

The rotational velocity vector $\dot{\eta} \in \mathbb{R}^3$, or the time-derivative of the RPY Euler angles (ϕ, θ, ψ), can be related to the angular velocity vector ${}^B \omega$ as:

$$\begin{aligned} {}^B \omega &= W(\eta)\ \dot{\eta}, \\ &= \begin{bmatrix} 1 & 0 & -s_\theta \\ 0 & c_\phi & s_\phi c_\theta \\ 0 & -s_\phi & c_\phi\ c_\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}, \end{aligned} \tag{2-3}$$

where $W(\eta)$ is the so-called Jacobian Representation matrix that depends only on the roll (ϕ), pitch (θ) and yaw (ψ) angles. Notice that, $W(\eta)$ is invertible only if the quadrotor attitude is far from the singularities of the RPY Euler angles, which occur when $\theta = \pi/2\ rad$ and implies that $c_\theta = 0$. Then, if the pitch angle θ satisfy the singularity boundaries $\theta \in \left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$, it follows that:

$$\begin{aligned} \dot{\eta} &= W^{-1}(\eta)\,{}^{B}\omega\,, \\ &= \begin{bmatrix} 1 & s_\phi\, tan_\theta & c_\phi\, tan_\theta \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi\, sec_\theta & c_\phi\, sec_\theta \end{bmatrix} {}^{B}\omega\,. \end{aligned} \tag{2-4}$$

Similarly, the linear velocity of the quadrotor expressed in the inertial frame $\mathcal{F}_I$, denoted by $\dot{\xi} \in \mathbb{R}^3$, is given as:

$$\dot{\xi} = \begin{bmatrix} {}^{I}\dot{x} \\ {}^{I}\dot{y} \\ {}^{I}\dot{z} \end{bmatrix}\,. \tag{2-5}$$

Now, the linear velocity of the quadrotor expressed in the body frame $\mathcal{F}_B$, denoted by $v \in \mathbb{R}^3$, it can be related to $\dot{\xi} \in \mathbb{R}^3$ using:

$$\dot{\xi} = R^{I}_{B}(\eta)\, v\,, \tag{2-6}$$

where $R^{I}_{B}(\eta) \in \mathbb{SO}(3)$ is the rotation matrix from the body frame $\mathcal{F}_B$ to the inertial frame $\mathcal{F}_I$, which is given by:

$$R^{I}_{B}(\eta) = \begin{bmatrix} c_\theta\, c_\psi & s_\phi\, s_\theta\, c_\psi - c_\phi\, s_\psi & c_\phi\, s_\theta\, c_\psi + s_\phi\, s_\psi \\ c_\theta\, s_\psi & s_\phi\, s_\theta\, s_\psi + c_\phi\, c_\psi & c_\phi\, s_\theta\, s_\psi - s_\phi\, c_\psi \\ -s_\theta & s_\phi\, c_\theta & c_\phi\, c_\theta \end{bmatrix}\,. \tag{2-7}$$

Notice that, the rotation matrix $R^{B}_{I}(\eta)$ describing the rotation of the inertial frame $\mathcal{F}_I$ with respect to the body frame $\mathcal{F}_B$ can be obtained from $R^{B}_{I} = (R^{I}_{B})^{\mathsf{T}}$. It is worth mentioning that $R^{I}_{B}(\eta)$ will be very useful in the derivation of the quadrotor dynamic model because some states are measured in the body frame $\mathcal{F}_B$ (e.g., thrust forces produced by the propellers and the quadrotor's attitude) whereas other states are measured in the inertial frame $\mathcal{F}_I$ (e.g., gravitational forces and the quadrotor's position). The rotation matrix R^{I}_{B} will be used in the formulation of the dynamics model, its due to the fact some states are measured in the body frame (e.g. thrust forces produces by the propellers) while others are measured in the inertial frame (e.g. gravitational forces and the quadrotor's position).

2.6 Quadrotor Dynamic Model

The dynamic motion of the quadrotor can be divided into two subsystems: translational and rotational. In the following subsections, the different forces and moments acting on the quadrotor will be defined for each rotor. In Figure 2.6 these forces and moments are clearly visible.

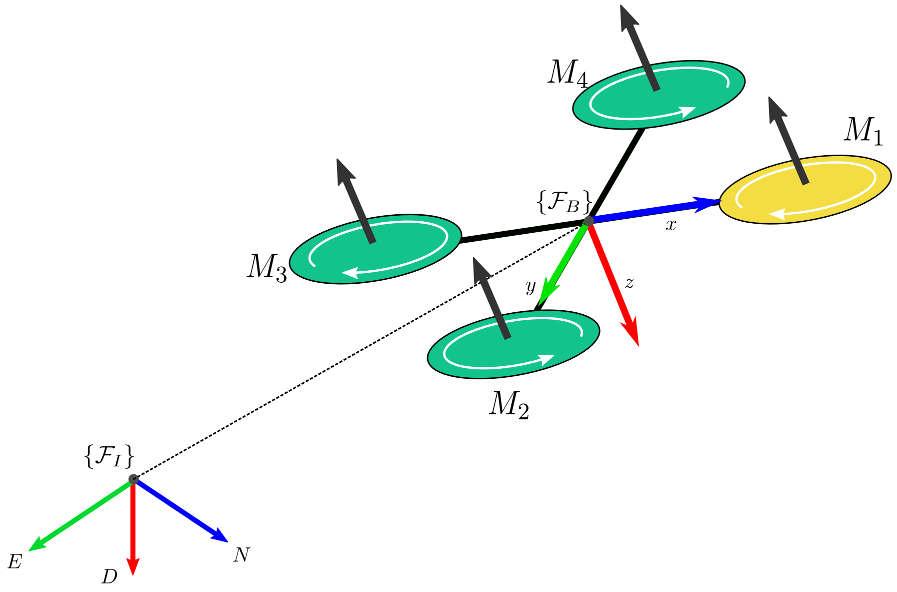


Figure 2.6: Forces and moments acting on a quadrotor.

2.6.1 Translational Motion Equation

The translational motion equation for the quadrotor, expressed in the inertial frame $\mathcal{F}_I$, is computed by using Newton's second law as in [39, 40, 60]:

$$\begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & m \end{bmatrix} \begin{bmatrix} {}^{I}\ddot{x} \\ {}^{I}\ddot{y} \\ {}^{I}\ddot{z} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} - R_B^I \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix} - \begin{bmatrix} b & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & b \end{bmatrix} \begin{bmatrix} {}^{I}\dot{x} \\ {}^{I}\dot{y} \\ {}^{I}\dot{z} \end{bmatrix} , \tag{2-8}$$

where m is the total mass of the quadrotor, g is the gravitational acceleration, b is the aerodynamic friction, and T is the total upward thrust. The quadrotor dynamic model (2-8) can be written in a more compact form as:

$$\mathcal{M}\,\ddot{\xi} = mg\,\vec{z}_{_I} - R_B^I\,T\,\vec{z}_{_I} - \mathcal{B}\,\dot{\xi}\,, \tag{2-9}$$

where $\mathcal{M} = \text{diag}(m, m, m)$ is the mass matrix, $\mathcal{B} = \text{diag}(b, b, b)$ is the aerodynamic drag matrix and $\vec{z}_{_I}$ is the unit vector along the z-axis of the inertial frame $\mathcal{F}_I$. Notice that, starting from left to right, the first term in (2-9) is the gravity force acting downward in the inertial frame $\mathcal{F}_I$, the second term is the total thrust of the quadrotor in the body frame $\mathcal{F}_B$ rotated to the inertial frame $\mathcal{F}_I$, and the third term is the aerodynamic drag force acting opposite to the motion direction.

2.6.2 Rotational Motion Equation

The rotational motion equation for the quadrotor, expressed in the body frame $\mathcal{F}_B$, is computed by using the Euler's equation of motion as [39, 40, 60]:

$$\begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} \mathcal{I} & 0 & 0 \\ 0 & \mathcal{I} & 0 \\ 0 & 0 & \mathcal{I} \end{bmatrix} \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} + \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} \mathcal{I} & 0 & 0 \\ 0 & \mathcal{I} & 0 \\ 0 & 0 & \mathcal{I} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} + \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ I_r \Omega_r \end{bmatrix} \tag{2-10}$$

The quadrotor dynamic model (2-10) can be written in a more compact form as

$$\tau = \mathcal{I}\,{}^B\dot{\omega} + {}^B\omega \times (\mathcal{I}\,{}^B\omega) + M_g\,. \tag{2-11}$$

Starting from left to right, the first term of (2-11) denotes the the rate of the angular momentum relative to the body frame $\mathcal{F}_B$, the second term represents the gyroscopic torque induced by the dependence of $\mathcal{I}$ on body orientation, and the term M_g represents the gyroscopic moments due to the rotor's inertia I_r, in which $\Omega_r = \Omega_1 - \Omega_2 + \Omega_3 - \Omega_4$ is the relative speed of the rotors. The gyroscopic moment of a rotor is a physical effect in which gyroscopic torques or moments attempts to align the spin axis of the rotor along the z-axis of the inertial frame $\mathcal{F}_I$.

The torque applied to the body frame $\mathcal{F}_B$, denoted by $\tau = [\,\tau_\phi \;\; \tau_\theta \;\; \tau_\psi\,]^\mathsf{T}$, defines two physical effects created by the aerodynamic force f_i and moment M_i produced by the i-th rotor, which are given by:

$$f_i = \frac{1}{2}\rho\, A\, K\, r^2\, \Omega_i^2\,, \tag{2-12}$$

$$M_i = \frac{1}{2}\rho\, A\, B\, r^2\, \Omega_i^2\,, \tag{2-13}$$

where ρ is the air density, A is the blade area, K and B are the aerodynamic coefficients, r is the radius of the blade, and Ω_i is the angular velocity of i-th rotor. These definitions was taken from [1, 38].

It is worth mentioning that the aerodynamic forces and moments depend on the geometry of the propeller and the air density. For quadrotors, the maximum altitude during the flight is usually limited, thus the air density ρ can be considered constant. According [39, 40], the aerodynamic force and moment, given by (2-12) and (2-13) respectively, can be simplified as:

$$f_i = k\,\Omega_i^2\,, \tag{2-14}$$

$$M_i = b\,\Omega_i^2\,, \tag{2-15}$$

in which k is the lift constant and b is the drag constant. As an effect of rotation, the aerodynamic force (2-14) causes an upwards thrust force. On the other hand, the aerodynamic moment (2-15) has an opposite direction of rotation with respect to the i-th rotor.

The combined forces of rotors create thrust force in the direction of the body z-axis, as shown in equation (2-16). The torque τ, is in the direction of the corresponding body frame.

$$T = -\sum f_i = -k\sum \omega_i\,, \tag{2-16}$$

$$\tau = \begin{bmatrix} l\;k(\Omega_2^2 - \Omega_4^2) \\ l\;k(\Omega_3^2 - \Omega_1^2) \\ b(\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2) \end{bmatrix}. \tag{2-17}$$

where l is the arm length, which is the distance between the axis of rotation of each rotor to the origin of the body reference frame.

Remark 1: For the purposes of this work, the simplified model does not consider the gyroscope moment.

2.7 Newton-Euler model

This section provides the specific model information of the quadrotor with 6 degrees of freedom (DoF) rigid body equation, derived according to the formalism defined in previous sections. The kinematics of a 6 DoF rigid-body is defined by:

$$\dot{q} = J\nu\,, \tag{2-18}$$

where $\dot{q}$ is the velocity vector in $\mathcal{F}_I$, ν is the velocity vector in $\mathcal{F}_B$ and J is the generalized matrix.

The linear and angular position in the inertial frame are represented by the vector ξ and η in equation (2-19). The linear and angular velocity vector in body frame equation as represented by v and ${}^{B}\omega$ in equation (2-20).

$$q = \begin{bmatrix} \xi & \eta \end{bmatrix}^{\mathsf{T}} = \begin{bmatrix} {}^{I}x & {}^{I}y & {}^{I}z & \phi & \theta & \psi \end{bmatrix}^{\mathsf{T}}, \tag{2-19}$$

$$\nu = \begin{bmatrix} v & {}^{B}\omega \end{bmatrix}^{\mathsf{T}} = \begin{bmatrix} {}^{B}\dot{x} & {}^{B}\dot{y} & {}^{B}\dot{z} & p & q & r \end{bmatrix}^{\mathsf{T}}. \tag{2-20}$$

The generalized matrix J is composed by 4 sub-matrix, according to:

$$J = \begin{bmatrix} R_B^I & 0_{3\times3} \\ 0_{3\times3} & W^{-1}(\phi,\theta,\psi) \end{bmatrix}. \tag{2-21}$$

The notation $0_{3\times3}$ means a zero matrix in dimension 3×3. The dynamics of a generic 6 DoF rigid body takes into account the mass of the body and inertia matrix, as shown in the equations (2-9) and (2-11). Hence, the dynamics of quadrotor by Newton-Euler method is described by the following equation:

$$\begin{bmatrix} {}^{B}F \\ \tau \end{bmatrix} = \begin{bmatrix} m\ I_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & I \end{bmatrix} \begin{bmatrix} \dot{v} \\ {}^{B}\dot{\omega} \end{bmatrix} + \begin{bmatrix} {}^{B}\omega \times (m\ v) \\ {}^{B}\omega \times (I\ {}^{B}\omega) \end{bmatrix}, \tag{2-22}$$

where $I_{3\times3}$ means a identity matrix $\in R^{3\times3}$, $\dot{v}$ is the quadrotor linear acceleration vector, ${}^{B}\dot{\omega}$ is the quadrotor angular acceleration vector. In addition, ${}^{B}F$ is the quadrotor forces vector and τ is the quadrotor torques vectors, it can be defined as:

$$\begin{bmatrix} {}^{B}F & {}^{B}\tau \end{bmatrix}^{\mathsf{T}} = \begin{bmatrix} F_x & F_y & F_z & \tau_x & \tau_y & \tau_z \end{bmatrix}^{\mathsf{T}}, \tag{2-23}$$

Now, we might try to write the linear and angular equations of motion separately:

- Linear motion equation in inertial frame.

$$m\ddot{\xi} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + R_B^I \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix}. \tag{2-24}$$

- Angular motion equation in the body frame:

$$I \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times I \begin{bmatrix} p \\ q \\ r \end{bmatrix} + \begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix}. \tag{2-25}$$

Substituting equation (2-17) into the equation (2-25) yields

$$I \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times I \begin{bmatrix} p \\ q \\ r \end{bmatrix} + \begin{bmatrix} l\ k(\Omega_2^2 - \Omega_4^2) \\ l\ k(\Omega_3^2 - \Omega_1^2) \\ b(\Omega_2^2 - \Omega_4^2 1 \Omega_1^2 - \Omega_3^2) \end{bmatrix} . \tag{2-26}$$

It should be noticed that, in this model, the aerodynamic effects have been neglected for the sake of simplification.

2.8 State-space model

After understanding the equations of motion, the state-space representation can be implemented. These states are $(x_i, i = 1, \cdots, 12)$, which together make up the state vector X as follows:

$$X = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 & x_9 & x_{10} & x_{11} & x_{12} \end{bmatrix}^{\mathsf{T}} . \tag{2-27}$$

The components of the state vector X correspond to the following (in the same order):

$$X = \begin{bmatrix} x & y & z & \phi & \theta & \psi & \dot{x} & \dot{y} & \dot{z} & p & q & r \end{bmatrix}^{\mathsf{T}} . \tag{2-28}$$

The dynamics of the state vector X, also known as state dynamic, are given by the vector $\dot{X} = [\dot{x}_1,\ \dot{x}_2, \cdots, \dot{x}_{12}]^{\mathsf{T}}$, whose components are defined as follows:

$$\dot{X} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \\ \dot{x}_7 \\ \dot{x}_8 \\ \dot{x}_9 \\ \dot{x}_{10} \\ \dot{x}_{11} \\ \dot{x}_{12} \end{bmatrix} = \begin{bmatrix} x_7 \\ x_8 \\ x_9 \\ x_{10} + s_{x_4}\ tan_{x_5}\ x_{11} + c_{x_4}\ tan_{x_5}\ x_{12} \\ c_{x_4}\ x_{11} - s_{x_4}\ x_{12} \\ \frac{s_{x_4}}{c_{x_5}}\ x_{11} + \frac{c_{x_4}}{c_{x_5}}\ x_{12} \\ -\frac{T}{m}[c_{x_4}\ s_{x_5}\ c_{x_6} + s_{x_4}\ s_{x_6}] \\ -\frac{T}{m}[c_{x_4}\ s_{x_5}\ c_{x_6} - s_{x_4}\ c_{x_6}] \\ -\frac{T}{m}c_{x_4}\ c_{x_5} + g \\ \frac{\tau_x}{I_{xx}} + \frac{I_{yy}-I_{zz}}{I_{xx}}\ x_{11}\ x_{12} \\ \frac{\tau_y}{I_{yy}} + \frac{I_{yy}-I_{zz}}{I_{xx}}\ x_{11}\ x_{12} \\ \frac{\tau_z}{I_{zz}} + \frac{I_{xx}-I_{yy}}{I_{zz}}\ x_{10}x_{11} \end{bmatrix}, \tag{2-29}$$

where $T,\ \tau_x,\ \tau_y,\ \tau_z$ are the thrust force (u_1) and moment inputs (u_2, u_3, u_4) defined in the equations (2-16) and (2-17). Particularly, a time invariant linear system (SLIT) may be represent by a state-space model of the form:

$$\begin{aligned} \dot{x}(t) &= A\ x(t) + B\ u(t)\,, \\ y(t) &= C\ x(t) + D\ u(t)\,. \end{aligned} \tag{2-30}$$

where $x(t)\ \in\ \mathbb{R}^N$, $y(t)\ \in\ \mathbb{R}^N$, $u(t)\ \in\ \mathbb{R}^N$ are time dependent vectors, while $A\ \in \mathbb{R}^{N\times N}$, $B\ \in \mathbb{R}^{N\times N}$, $C\ \in \mathbb{R}^{M\times N}$, $D\ \in \mathbb{R}^{M\times R}$ are constant matrices.

Since the quadrotor is a multi-input multi-output(MIMO) system, with 12 state variables, 4 inputs and 4 output states. After linearization, the matrix A is 12 × 12, B is 12 × 4, and C is 4 × 12.

To linearize the dynamics at the hover configuration, we have: $u_1 \approx mg$, $\theta \approx 0$, $\phi \approx 0$, $\psi \approx \psi_0$, $p \approx 0$, $q \approx 0$ and $r \approx 0$. (equilibrium point X_{eq}). As defined previously the state-space dynamics as $\dot{X}$:

$$\dot{X} = f(X, U), \qquad f\ \in \mathbb{R}^{12}\,. \tag{2-31}$$

At hover f must be $f(X_{eq}, U_{eq}) = 0$. The equilibrium point at hover, and the achived input correspond to:

$$X_{eq} = \begin{bmatrix} x_{eq} & y_{eq} & z_{eq} & 0 & 0 & \psi_{eq} & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^{\mathsf{T}}, \tag{2-32}$$

$$U_{eq} = \begin{bmatrix} \omega_{eq}^2 & \omega_{eq}^2 & \omega_{eq}^2 & \omega_{eq}^2 \end{bmatrix}^{\mathsf{T}}, \tag{2-33}$$

where U_{eq} is the input system (U), it is necessary to achieve hover state when the rotational speed of all the propellers must be equal. The *thrust* is equal to the force of gravity on the drone $(T = mg)$. The definition of thrust in the equation (2-16) leads to:

$$w_{eq} = \sqrt{\frac{mg}{4k}}\,. \tag{2-34}$$

The linearization of a time-invariant nonlinear system leads to local properties. In order to obtain this linear representation of the system

$$\Delta\dot{X} = A\Delta X + B\Delta U\,. \tag{2-35}$$

The Jacobian matrices A and B are defined as:

$$A = \frac{\partial f}{\partial X}(X,U)|_{(X_{eq},U_{eq})}\,, \tag{2-36}$$

$$B = \frac{\partial f}{\partial U}(X,U)|_{(X_{eq},U_{eq})}\,. \tag{2-37}$$

Then, the actual state and input are re-defined as:

$$X = X_{eq} + \Delta X\,, \tag{2-38}$$

$$U = U_{eq} + \Delta U\,. \tag{2-39}$$

Using the previous two equations and the system defined in (2-31), we have that A and B can be defined as:

$$A = \begin{bmatrix} O_{6\times6} & I_{6\times6} \\ \Lambda & O_{6\times6} \end{bmatrix}, \tag{2-40}$$

$$B = \begin{bmatrix} O_{8\times4} \\ \Delta \end{bmatrix}. \tag{2-41}$$

where $O_{i\times j}$ and $I_{i\times j}$ are the zero matrices and the identity matrix respectively, of size $i \times j$ and the remaining terms, namely Λ and Δ are defined as follows:

$$\Lambda = \begin{bmatrix} 0 & 0 & 0 & gs_{\psi_{eq}} & gc_{\psi_{eq}} & 0 \\ 0 & 0 & 0 & -gc_{\psi_{eq}} & gs_{\psi_{eq}} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{2-42}$$

$$\Delta = \begin{bmatrix} \frac{k}{m} & \frac{k}{m} & \frac{k}{m} & \frac{k}{m} \\ 0 & -\frac{kl}{I_x} & 0 & \frac{kl}{I_x} \\ \frac{kl}{I_y} & 0 & -\frac{kl}{I_y} & 0 \\ \frac{b}{I_z} & -\frac{b}{I_z} & \frac{b}{I_z} & -\frac{b}{I_z} \end{bmatrix}. \tag{2-43}$$

2.8.1 Controllability and Observability of the Linear System

Controllability and Observability represent two principal concepts of modern control system theory as in [87]. From now on, we consider the following system described by (4-15)

$$\begin{aligned} \dot{x}(t) &= A\ x(t) + B\ u(t)\,, \qquad x(t_0) = x_0\,, \\ y(t) &= C\ x(t) + D\ u(t)\,. \end{aligned} \tag{2-44}$$

where A and B are given in (2-40) and (2-41).

Definition 2.1 (Controllability.) *A system is said to be controllable at time t_0 if it is possible by means of an unconstrained control vector to transfer the system from any initial state $x(t_0)$ to any other state in a finite interval of time. If the system is completely controllable requires that the rank ρ of the controllability matrix $\mathcal{C} \in \mathbb{R}^{n \times n}$:*

$$\mathcal{C} = [\mathbf{B} \mid \mathbf{AB} \mid \mathbf{A}^2\mathbf{B} \mid \cdots \mid \mathbf{A}^{n-1}\mathbf{B}]\,, \tag{2-45}$$

be n, that is, $\rho(\mathcal{C}) = n$.

Definition 2.2 ***Observability**. A system is said to be observable at time t_0 if, with the system in state $x(t_0)$, it is possible to determine this state from the* *observation of the output over a finite time interval. If the system is completely observable requires that the rank of the $nm \times n$ matrix:*

$$\mathcal{O} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \\ \vdots \\ \mathbf{CA}^{n-1} \end{bmatrix} \tag{2-46}$$

be n, that is, $\rho(\mathcal{O})=n$.

2.8.2 Conclusions

In this chapter we derived the dynamic model of the quadrotor. This model is obtained in a system equation form governed by the Newton-Euler formalism using Euler angles in order to obtain the kinematics and dynamics equations. The model contains basic physics equations and we also assume fixed aerodynamics coefficients.

3
Vision based Navigation

An aerial vehicle can fly without a human pilot. In order to successfully complete the mission, a UAV must be fully aware of their states, including location, speed, direction as well as a starting point and target location. To date, various navigation methods have been proposed, which can be mainly divided into three categories: inertial navigation, satellite navigation, and vision-based navigation. However, none of these methods are perfect, therefore, it is crucial to choose the most appropriate for each specific task.

Vision-based navigation is important and promising research field, because of the rapid development of computer vision. The visual sensors can provide abundant information about the surroundings, allowing is the perception of the dynamic environment, because of their remarkable anti-interference ability [78].

Figure 3.1 shows the architecture of the system, which has two components: a UAV and the vision module. The video from the UAV front camera is extracted and processed frame by frame by the vision module. First, the vanishing point is located, and depending on its relative position on the image plane in regards to the image boundaries, the control command to move forward is set, following a virtual path. The PID controller of the UAV is responsible for modifying the control commands (change velocities).

3.1
Digital image

A digital image is an array of digital data, obtained by an optical sensor (camera); interpreted by a computer, and displayed as a photograph. In order for a digital image to be produced, light must originate from a source, reflect off the various elements of the subject, enter the aperture of an optical sensor and stimulate the sensor's photoelectric array. This process involve a projection of the 3D world onto a 2D surface.

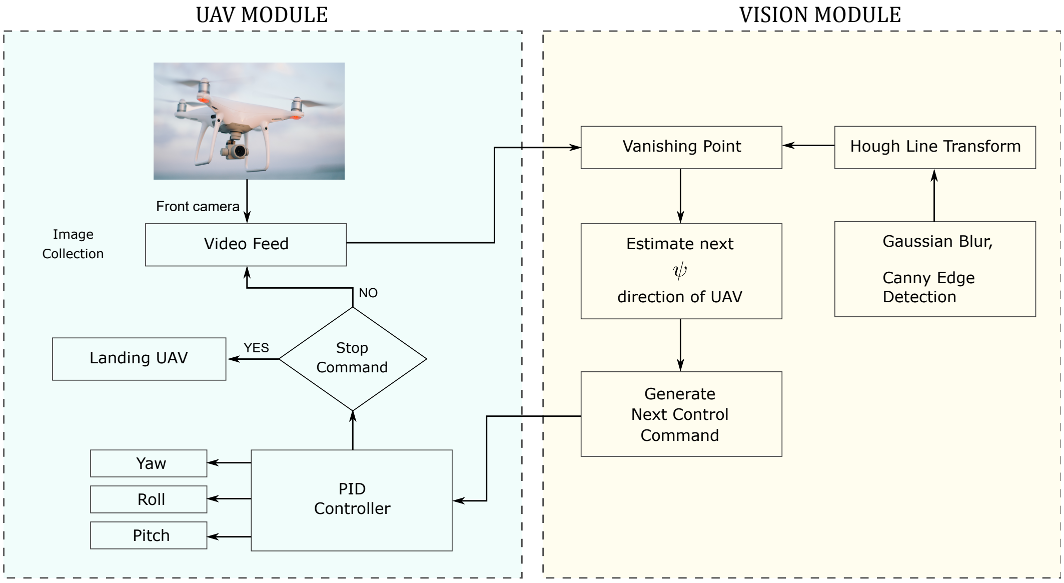


Figure 3.1: System architecture with two components: UAV Module and a Vision Module. The video feed from the UAV front camera is transmitted frame-by-frame to the Robot Operating System (ROS) interface for further processing. ROS tries to find the position of the vanishing point in each frame with the proposed algorithm. The relative position of the vanishing point on the image plane with respect to any of the image boundary axes is then utilized for deriving the next set of control commands that will ensure safe navigation of the UAV through the poly-tunnel without colliding with plants neither gutter tabletops.

3.1.1 Perspective projection

A small hole in the wall of a darkened room will cast a inverted image of the outside world on the opposite wall – a so-called pin-hole camera. The pinhole is called the center of projection, the front screen is the image plane, the distance between the center of projection and the image plane is the focal distance (f), the optical axis is the line through the center of projection that is perpendicular to the image plane, the point where the optical axis pierces the sensor plane is the principal point, as shown in Figure 3.2.

3.2 Crop Detection

Crops such as maize, sugar beet, potato, soybean, and so on in the open field, or strawberries, tomatoes, and many others in greenhouses are arranged in multiple parallel rows. This arrangement facilitates cultivation and other farming operations. In a crop field environment, there are a lot of parallel lines in three-dimension (3D) space, that can be mapped into some intersection lines in a two-dimension (2D) image, and the intersection point formed by

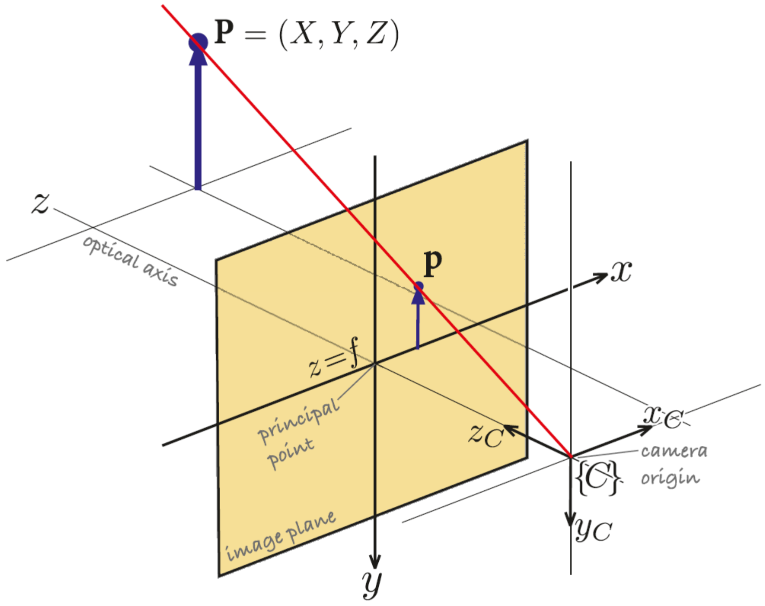


Figure 3.2: The central projection model. [60]

these intersection lines is called vanishing point. [82]

The vanishing point is defined as the convergence point of lines in an image plane produced by the projection of parallel lines in a real space. When a 3D real-world scene is projected onto a 2D image, parallel lines in 3D space transform into converging lines in the image plane due to perspective [80].

The model for human vision and cameras is the central projective model (or perspective). Light rays passing through one unique point (the focal point) form the projected image. For example, Figure 3.3 shows the geometry of the perspective projection of a crop field. The points p_1 and p_2 belonging to a row crop in the horizontal plane were projected onto the image plane following a line through the focal point. Other points (p_3 and p_4) in the next row crop, as well as in a parallel row crop were projected in the same way. All these points belong to parallel row crops in the field plane, but these rows are not parallel in the image plane due to the non-linear aspects of perspective.

The mapping relationship between point $A(x_0, y_0, z_0)$ in the three-dimension space and point $B(x_0', y_0', z_0')$ in the two-dimension image, is given by [66]:

$$(x_0', y_0', z_0') = \left(\frac{x_0\ f}{y_0}, \frac{z_0\ f}{y_0}, f \right), \tag{3-1}$$

where f represents the focal length of visual system.

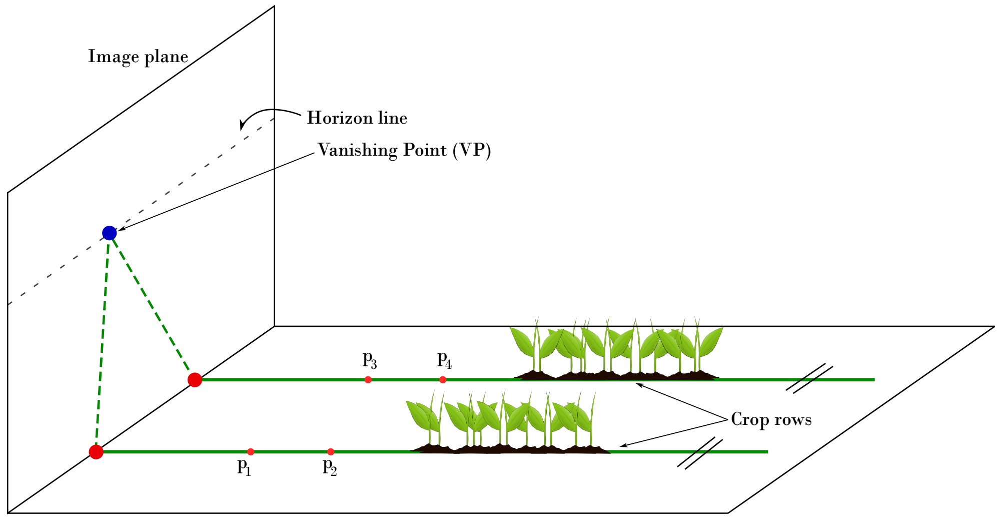


Figure 3.3: Geometry of the central projective model. Parallel row crops (3D) are not parallel in the field plane perspective and intersect at the vanishing point, in the image plane (2D).

3.2.1 Image collection

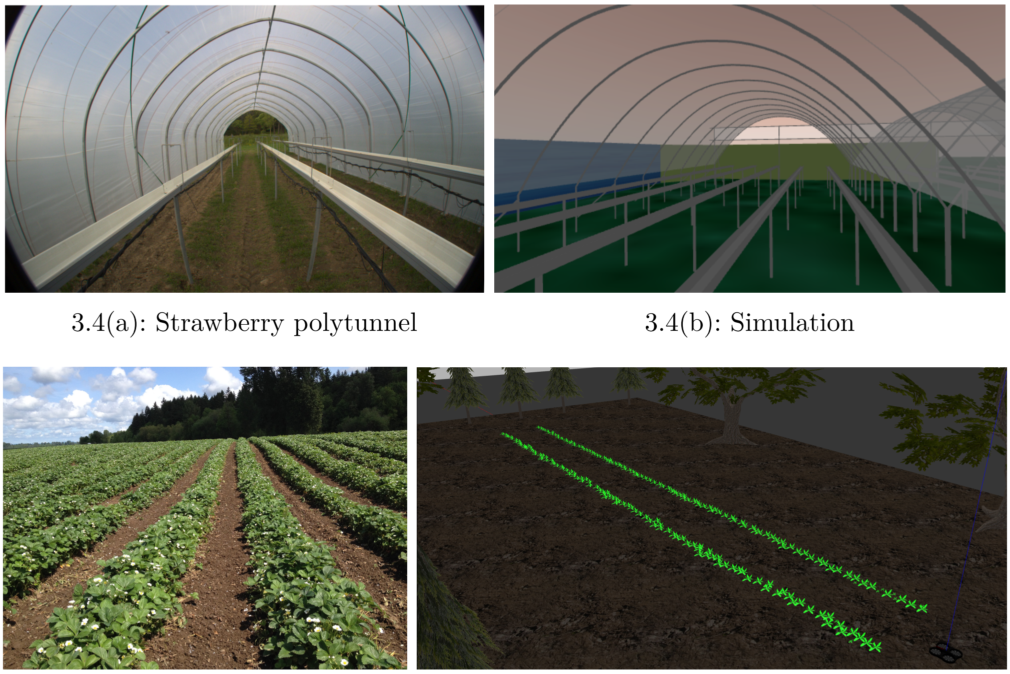
In order to evaluate the proposed methodology and its accuracy under various conditions with different initial parameters, several videos and images of natural scene was used. However, to complete the two modules of the system are architecture need environments with a robot.

Gazebo was the tool used to simulate the complete environment as shown in Figure 3.4. ArDrone is the robot used in Gazebo to test to command control and navigation. For the real-world, the tests were performed with videos available on the Internet about fields of grapes, soybeans, and paprika at open fields and greenhouse.

The digital images were captured from the simulator and stored as 8-bit color images, with resolutions of 360 × 640 pixels, saved in RGB color space. Nevertheless, only a reduced area on the perspective image is of interest either for detecting row crops or as a reference for guiding, denoted the Region of Interest (ROI). In this study, the ROI always has a trapezoidal shape, considering that it only needs two-row crops to be detected, with sufficient points to estimate the equations that define the row crops.

3.2.2 Agronomic Index

First, the vegetation needs to be recognized and there is color index that help it stand out from other elements in the scene such as green. For RGB



3.4(a): Strawberry polytunnel

3.4(b): Simulation

3.4(c): Row crop

3.4(d): Simulation

Figure 3.4: Example of real and simulated images and environment: (a) Real strawberry poly-tunnel (greenhouse) in Norway, (b) Simulated environment of (a) with height camera in 1.5 m, (c) Real image field, and (d) Simulated virtual field.

image, the excess green color is a good agronomic contrast index [82]. It is defined as:

$$I = 2 \times G - R - B \tag{3-2}$$

where I is the pixel value and R, G and B are red, green and blue channels.

This equation is insensitive for differences in lighting conditions and shadows. So, I is equivalent to a gray image as shown in the Figure 3.6(b).

3.2.3 Image Processing

Line detection generally needs low-level pre-processing of the image such as: smoothing, erosion, dilatation and edge detection. Two principal methods of edge and line detection are presented: Canny edge detector and Hough transform, followed by their application towards the detection of vanishing points.

3.2.3.1 Canny Edge Detector

Canny algorithm is an edge detection operator characterized by a low error rate. It is composed by several steps. First, it smooths the image and then finds the intensity gradient of the image. Finally, edge thinning and thresholding are applied by hysteresis, the algorithm only works on gray scale images [60]. OpenCV library implements the Canny edge detector [85].

3.2.3.2 Hough Transform Method

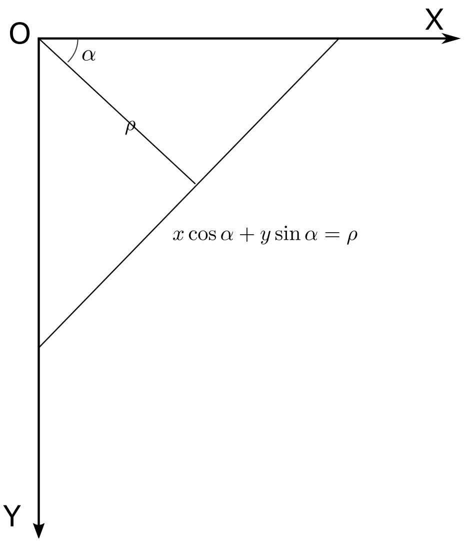


Figure 3.5: Hough line representation on an image plane

The Hough transformation (HT) is applied on the binary image to acquire a set of perspective lines. It uses polar coordinate system to represent a line on the image plane given by the parametric representation:

$$\rho = x\,c_\alpha + y\,s_\alpha\,, \tag{3-3}$$

where ρ represents the length of the perpendicular drawn from the origin onto the line and α represents the angle between the perpendicular and the positive $x-$axis [60, 86], as shown in Figure 3.5.

The HT is one of the most commonly used algorithms for line detection in perspective vision [65, 66, 80, 81]. In order to successfully perform Hough Transform and detect the right lines, there are different parameters that need to be taken care of [85].

In this work, a crop row detection method for real and simulated images is proposed:

1. Grayscale image.

2. Canny edge detection, the Gaussian blurring filter was used, with a 5 × 5 window was used to remove the noise and unimportant edges.

   From our experiments, we try to compensate the two parameters of the *high* and *low* thresholds that check the gradient value of a pixels.

3. Extract the set of lines intersecting at infinity by Probabilistic Hough Transform (PHT) method. However, these lines could be discontinuous and random. So, a fit method is used in order to find one line per side of image (left and right). The lines are discarded based on the following condition:

   Condition 1: The horizontal and vertical lines do not play any role in the estimation of vanishing point (VP) since these lines are parallel to image coordinate axes. The lines are grouped into two classes: *positive-slopped lines* with slope of the perpendicular $m > 0$ and *negative slopped lines* with slope of the perpendicular $m < 0$.

4. Region masking, this function is used to fit masks the unwanted parts of the image so that only the ROI appears, as shown the figure 3.6(c) and (d).

3.3 The Vanishing Point Algorithm

First, the unhelpful lines must be deleted, until only 2 lines remain. There is a number of straight lines in a set as $C\{L_1,\ L_2,\ \cdots,\ L_k \quad | \quad k = 0, \cdots,\ m\}$, where m is the number of straight lines system that has been detected and k is a integer number.

Each line can be expressed by their starting points and end points $\{(x_{s0}, y_{s0}), (x_{e0}, y_{e0})\}, \{(x_{s1}, y_{s1}), (x_{e1}, y_{e1})\}, \cdots, \{(x_{sm}, y_{sm}), (x_{em}, y_{em})\}$. The angle between each line and the horizontal coordinate axis (or slope) can be calculated by:

$$deg_m = slope = \arctan\left(\frac{|y_{em} - y_{sm}|}{|x_{em} - x_{sm}|}\right). \tag{3-4}$$

All lines in the image recognized by PHT are analyzed aiming at deleting the undesired by means of judging Condition 1, which allows knowing whether a line is close to horizontal or vertical straight lines.

All undesired lines are deleted until the number of lines in the line set is 2, one on the left side and one on the right side of the center image. Each straight lines can be expressed on the cartesian plane as:

$$y = mx + b, \tag{3-5}$$

where m is the slope or gradient of the line, b is the $y-$intercept of the line and x is the independent variable of the function $y = f(x)$.

In order to find the intersection point of the 2 remaining lines or vanishing point (X_{VP}, Y_{VP}), the following formula was applied:

$$X_{VP} = \frac{(m_{L_1}\ x_{s0}) - y_{s0} - (m_{L_2}\ x_{s1}) + y_{s1}}{m_{L_1} - m_{L_2}}, \tag{3-6}$$

$$Y_{VP} = m_{L_1}(X_{VP} - x_{s0} + y_{s0}), \tag{3-7}$$

where m_{L_1} and m_{L_2} are the slopes of the lines L_1 and L_2, (x_{s0}, y_{s0}) is the start point of the L_1, and (x_{s1}, y_{s1}) is the start point of the L_2. If $m_{L_1} = m_{L_2}$, then the two lines are parallel and there is no intersection. Figure 3.6(e) shows an example of the vanishing point detection.

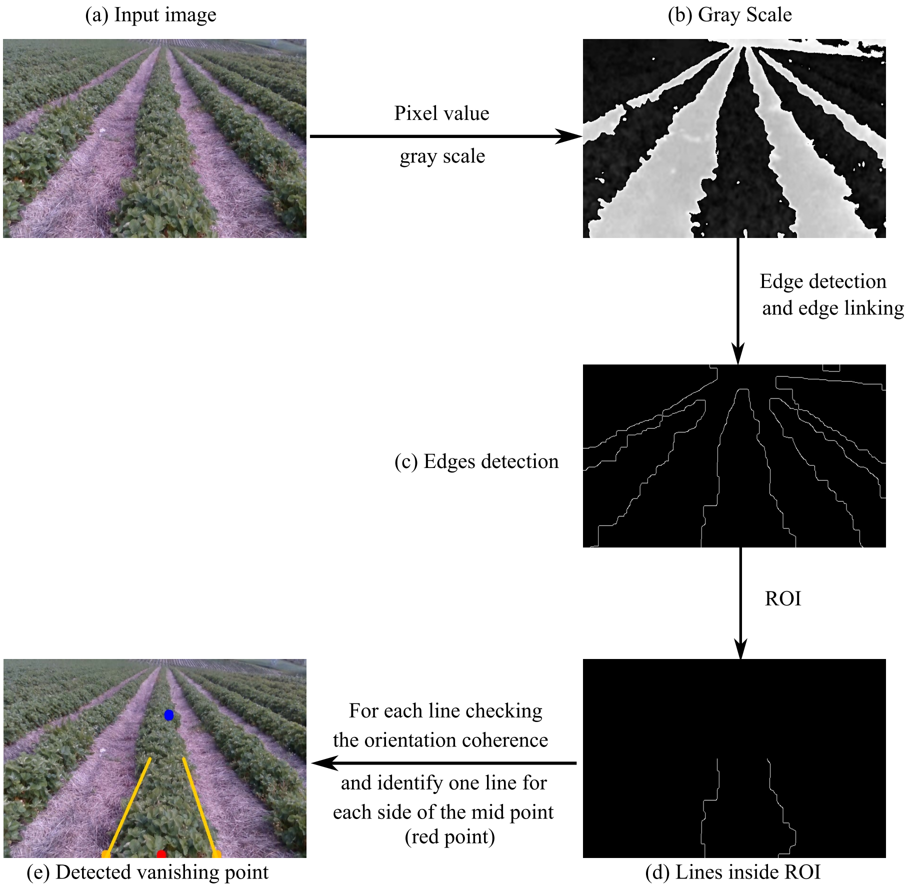


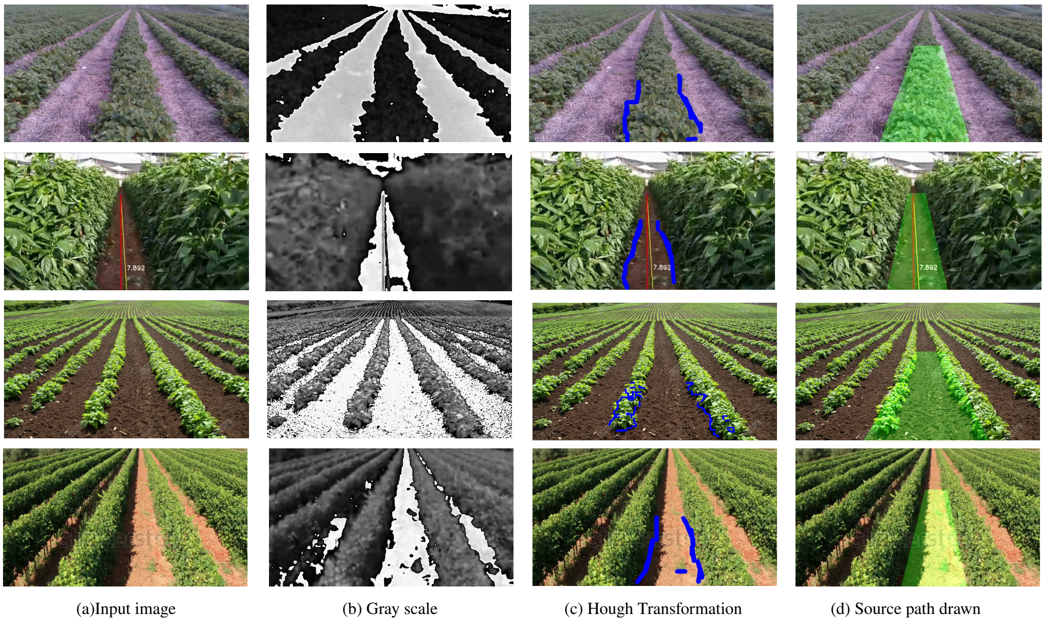
Figure 3.6: The framework of crop row detection using vanishing point.

3.4 Evaluation of Vision Module

The proposed framework for VP is validated with videos from real and simulated crop rows, with varying dimensions of frames. Information about the camera setup for tests in the real-world are unknown. In the simulated environment, the altitude of the camera is known (around 1.5m).

The Figure 3.7 shows four different crops: soybean (camera over a crop row), papikra, soybean (camera over the space between two rows), and grapes. These images allows to analysis two different situations: when the robot have to follow a row crop field, and when the robot go forward in the middle of two row crops.

The Figure 3.8 shows the VP recognition in simulated images. In this case, the first step in pre-processing of images is different, in simulated images the equation (3-2) is not robust, and the crop fields not highlighting. So, grayscale image equivalent is obtained with the OpenCV function `cv2.COLOR_BGR2GRAY`, the rest of the process is the same that was enumerated in the section 3.2.3.



(a)Input image (b) Gray scale (c) Hough Transformation (d) Source path drawn

Figure 3.7: Vanishing point detection test in real crop images: (a) Original input RGB image, (b) The input image converted to its gray equivalent(green agronomic index is applied), (c) Lines found by PHT inside the ROI, and (d) Source path drawn.

At this point, the VP location will be responsible for yaw navigation and the movements on the y-axis quadrotor. Table 3.1 summarized the result of path recognition for every environment (real and simulated), in a real environment. In the soybeans crop, there are different percents of recognition for the first and second crop due to the illumination and contrast of color. In paprika crop, the recognize is better because there is less relation between land and green color. Grape crops, present a similar little problem with the high

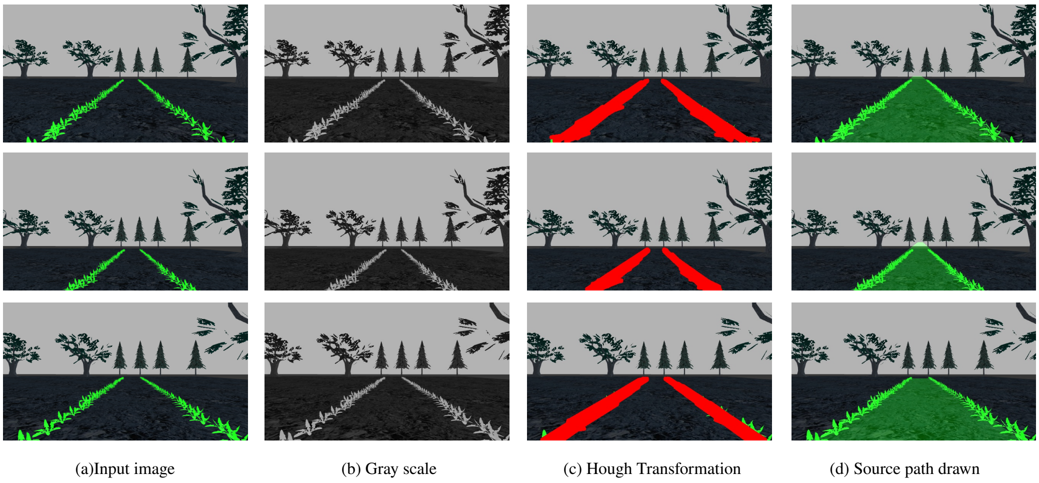


Figure 3.8: Vanishing point detection test in simulated images for different altitudes camera: (a) Original input RGB image, (b) Grayscale image, (c) Lines found by PHT inside the ROI, and (d) Source path drawn.

plants' shadow. Finally, the simulated environment has better percent of edge recognition because the crop row is tight a line with high contrast, with error only when the quadrotor is a very high or small height.

3.5 Autonomous navigation of UAV

In the real world, the position of the UAV is related to the location of the vanishing point (VP) on the image plane. VP has to close to the central vertical line of the image. So, the UAV flies near the medium axis of the two-row crops (or in the middle of the crop row). However, any significant deviation of VP from the central-vertical line on the image plane implies that the UAV is out from the desired path, and then it may lose the path (if following a row) or collide against either of the side row crops (if flying between them).

To overcome this issue, the variable δ is created and defined as the safe

Table 3.1: Vanishing Point recognition.

Environment	Frames	Crop detection
Soybean (I)	120	110 (91.67%)
Papikra	120	115 (95.83%)
Soybean (II)	120	103 (85.83%)
Grapes	120	111 (92.5%)
Simulated environment	120	112 (91.67%)

region where the $x-$ coordinate of the VP can be located, which guarantees that the UAV is close enough to the middle of the path. The Figure 3.9 shows different and possible reliable scenarios of VP location: when VP is within the safe region, and also with it outside, in case (b) and (c), other situations could be when some point meet with the $x-$ axis or $y-$axis and the VP may beyond the image, in which cases the quadrotor is too high (more to 2.5m) or close to the ground (less 1m). The ROI window guarantees to recognize any of the 3 cases mentioned above, in any other situation the quadrotor not find a VP and continues with its previous control command.

From Figure 3.9 six principal points were identified, for positive-slopped-line, the start and end points correspond to the point of intersection of the line with the reference $x-$axis and its intersection with the image plane (R and Q), in the same way for negative-slopped-line (S and T). W and H are respectively the width and height of the image plane and (X_{VP}, Y_{VP}) the position of VP at any time. So, the *positive-slopeed-line-ratio* (PSLR), *negative-slopped-line-ratio* (NSLR), and PSLR-*to*-NSLR-*ratio* (PTNR), can be defined as:

$$PSLR = \frac{VR}{VQ} \tag{3-8}$$

$$NSLR = \frac{VT}{VS} \tag{3-9}$$

$$PTNR = \frac{PSLR}{NSLR} \tag{3-10}$$

In order to calculate the control command, VP and PTNR are required to decide the direction of the quadrotor. In the best case, VP should lie on the central vertical line, given by PTNR = 1, or within the safe region defined above ($x = \frac{W}{2} \pm \delta$). Particularly, we have taken $\delta = 0.04 \times$ image-width. Similary, we consider an acceptable range for PTNR, given by $1 \pm \epsilon$, with $\epsilon = 0.02$. When the quadrotor is not in the middle of the path, at least one of the following condition holds:

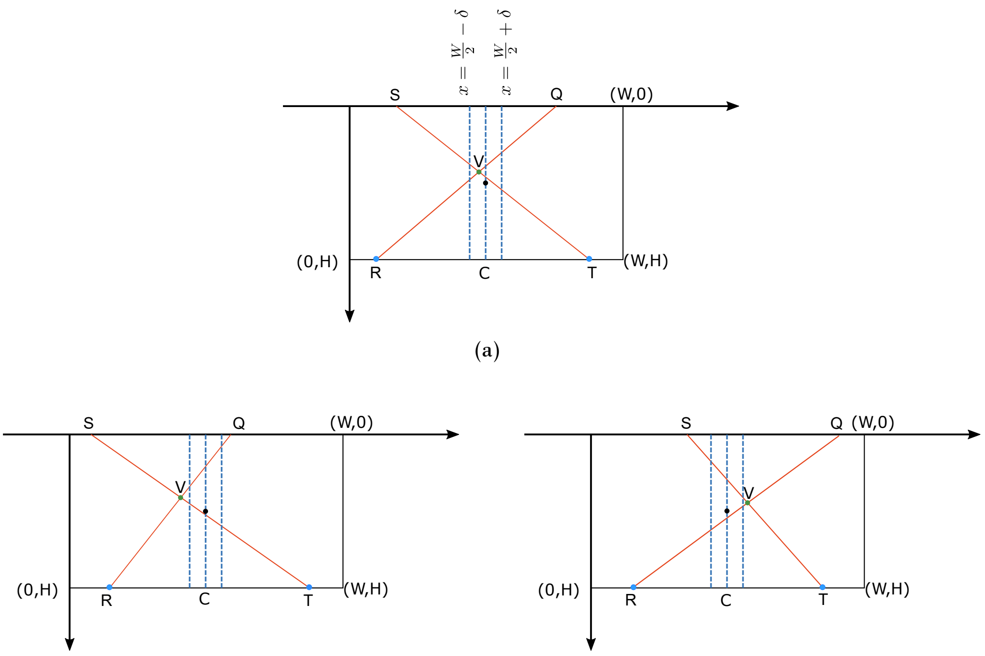
$$x_{VP} > \frac{W}{2} + \delta, \tag{3-11}$$

$$x_{VP} < \frac{W}{2} + \delta, \tag{3-12}$$

$$PTNR \leq 1 - \epsilon, \tag{3-13}$$

$$PTNR \geq 1 + \epsilon \tag{3-14}$$

The possible scenarios of X_{VP} and PTNR are taken into consideration to prepare a rule that generate the control command to navigate the UAV in



(a)

(b)

(c)

Figure 3.9: Possible vanishing point location yields to recognize if the UAV is out-of-track from the desired path.

crops, which is enumerated in Algorithm 1.

Algorithm 1: Control command UAV

```
Input  : Vanishing point abscissa: $X_{VP}$
Output: UAV yaw direction ($\psi$)
Data: UAV initial position ← $[x, y, z]$ and initial velocity ← $v_y$
if Vanishing point not found then
    Actuate UAV in previous direction
else
    if $1 - \epsilon <ptnr< 1 + \epsilon$ then
        if $X_{VP} < C_x + \delta$ then
            Turn left UAV (yaw left)
        else if $X_{VP} > C_x + \delta$ then
            Turn right UAV (yaw right)
        else
            Move the UAV on the X axis with a positive speed
    else if $ptnr \leq 1 - \epsilon$ then
        Move the UAV on the Y axis with a negative speed
    else if $ptnr \leq 1 - \epsilon$ then
        Move the UAV on the Y axis with a positive speed
```

3.6 Experiments and results

We performed experiments on 3 different video sequences captured by UAVs flying taken from internet and 1 video sequence captured by a wheeled robot on strawberries crop. In simulation experiment we use Gazebo and ROS for test the command control of the UAV module.

The figures 3.10 and 3.11 are extracted from the simulation environment, the first column is the original image taken from the camera of robot, the second column show the canny processing image, and the third column show the VP and the direction that the robot has to taken to navigate. The situations in th figure 3.10 are when the value of factor $ptnr$ is inside the range defined before $(1 \pm \epsilon)$. (a) is when the robot start to navigate, it is an altitute less than optimal (1.6m), (b) the UAV reaches the optimal altitude, (c) the quadrotor is above the optimum height, and (d) When within the ROI a complete line and another small line are identified, then the robot have to move in the y direction.

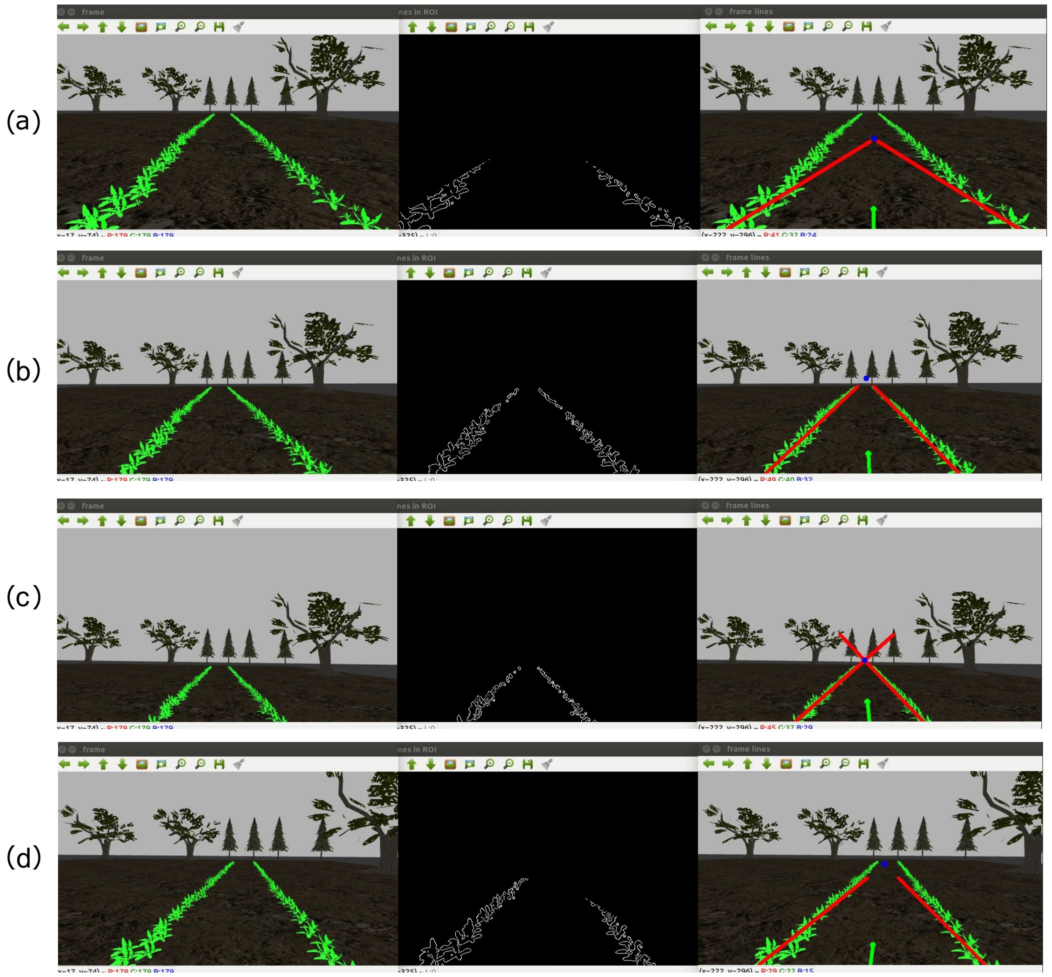


Figure 3.10: Various stages of VP estimation with direction motion to forward.

Figure 3.11 shown situation when the value of the factor $ptnr$ is out of the range $1 \pm \epsilon$ and the quadrotor have to move on the $y-$ axis to be in the middle of the crop rows.

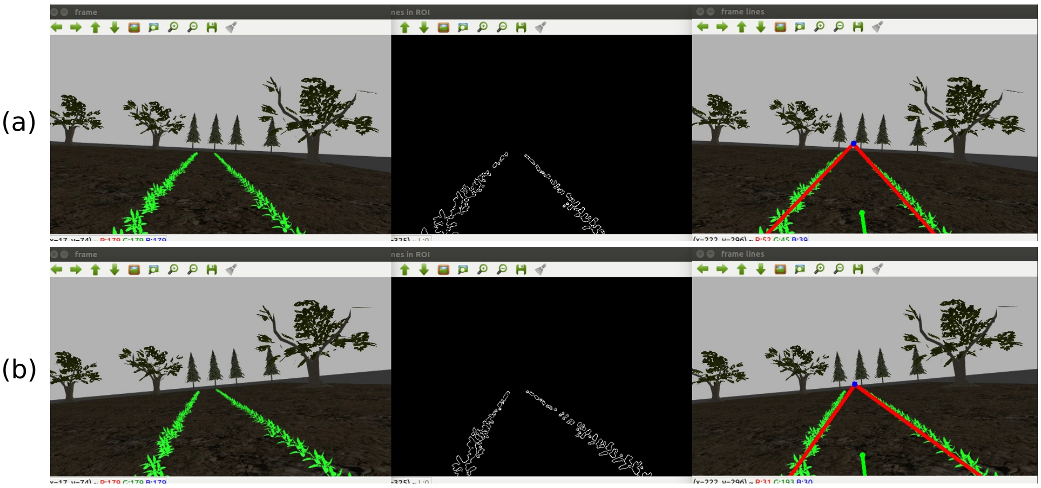


Figure 3.11: Various stages of VP estimation with direction motion to right and left.

3.7 Conclusions

This chapter has presented a description of the vision algorithm used within this work. A simple and fast method of vanishing point estimation is developed. An algorithm, based on the position of the vanishing point is proposed to modify the yaw to navigate the UAV in a crop field. The simulation shows how the quadrotor flying between two crop rows using rates inside a safe region.

4
Quadrotor Control Strategies

The quadrotor model estimated in Chapter 2 is med to evaluate the proposed control strategies. The test were performed on MATLAB simulated model, where it was easy to evaluate the performance with a mathematical approach.

This work covers three motion controllers for a quadrotor based on linear and non-linear control techniques. In the scope of linear control theory, the PID (Proportional, Integral, and Derivative) controller and the Adaptive Linear Pole Placement Control have been studied to evaluate the flight performance regarding the control gains tuning. Furthermore, a non-linear adaptive control technique has been investigating to analyze the flight behavior with respect to parametric uncertainty in the robot model. The effectiveness and feasibility of all controllers will be demonstrated through numerical simulations.

In general, the position and the attitude control can be treated as separate modules and various combinations of control laws can be applied to each controller. For example, the position control can be done with PD control while attitude stabilization can be done using Lyapunov theory and the other way around.

The quadrotor is controlled independently by nested feedback loops as shown in Figure 4.1. The inner loop is an attitude control loop while the outer loop is the position control loop. In hover configurations, the dynamic of attitude does not matter much in general, but in cases where the robot has to make maneuvers, it is imperative to have a faster attitude control loop.

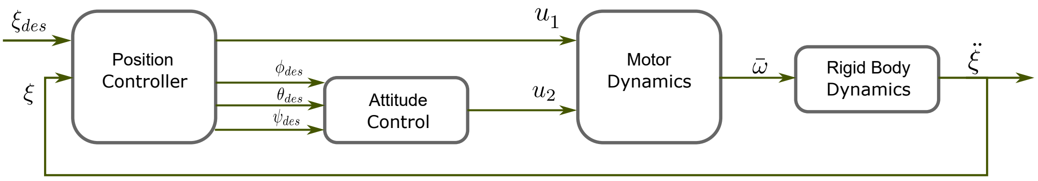


Figure 4.1: Overall control frame of a quadrotor

4.1 Mathematics of quadrotors Control

The control input are the thrust force (T) and the moment torques (τ_ϕ, τ_θ, τ_ψ).

$$u = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} T \\ \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} \tag{4-1}$$

$$m\ddot{\xi} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + R_B^I \begin{bmatrix} 0 \\ 0 \\ u_1 \end{bmatrix}, \tag{4-2}$$

$$I \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times I \begin{bmatrix} p \\ q \\ r \end{bmatrix} + \begin{bmatrix} u_2 \\ u_3 \\ u_4 \end{bmatrix}. \tag{4-3}$$

where $[p, q, r]$ are the angular velocities (ω) measured by the gyroscope; $[\phi, \theta, \psi]$ are the roll, pitch and yaw angles; m is the quadrotor mass; I is the moment of inertia; u_1, u_2, u_3 and u_4 are the thrust force and moment input.

For any quadrotor, normally there are nine equations of motion that we defined with the model, in order to determine the dynamics of the linear acceleration $(\ddot{x}, \ddot{y}, \ddot{z})$; with respect to $\mathcal{F}_e$, angular accelerations $(\dot{p}, \dot{q}, \dot{r})$ with respect to the $\mathcal{F}_b$; and, lastly, the Euler rates $(\dot{\phi}, \dot{\theta}, \dot{\psi})$.

$$\ddot{x} = -\frac{u_1}{m}[c_\phi\ s_\theta\ c_\psi + s_\phi\ s_\psi], \tag{4-4a}$$

$$\ddot{y} = -\frac{u_1}{m}[c_\phi\ s_\theta\ s_\psi - s_\phi c_\psi], \tag{4-4b}$$

$$\ddot{z} = -\frac{u_1}{m}\ c_\phi\ c_\theta + g, \tag{4-4c}$$

and

$$\dot{p} = \frac{u_2 + I_{yy}qr - I_{zz}qr}{I_{xx}}, \tag{4-5a}$$

$$\dot{q} = \frac{u_3 - I_{xx}pr + I_{zz}pr}{I_{yy}}, \tag{4-5b}$$

$$\dot{r} = \frac{u_4 + I_{xx}pq - I_{yy}pq}{I_{zz}}. \tag{4-5c}$$

Equation (2-4) can be written:

$$\dot{\phi} = p + \frac{r\ c_\phi\ s_\theta}{c_\theta} + \frac{q\ s_\theta\ s_\phi}{c_\theta}, \tag{4-6a}$$

$$\dot{\theta} = q\ c_\phi - r\ s_\phi, \tag{4-6b}$$

$$\dot{\psi} = \frac{r\ c_\phi}{c_\theta} + \frac{q\ s_\phi}{c_\theta}. \tag{4-6c}$$

4.2 Proportional Integration Derivative Control

PID control is one of the simplest linear control laws. It can be easily implemented in real-time systems using micro-controllers [53]. It is easy of understanding and computationally efficient becoming one of the most used control laws for UAVs.

The traditional PID structure is composed of the addition of three components. The first component P, is proportional to the error. Inside this interval, the output will be proportional to the error while outside the output will be minimum or maximum. The second component I, varies according to the integral of the error. This component increases the overshoot and the settling time, it has eliminates the steady-state error. The third component D, varies according to the derivate of the error. This component helps to decrease the overshoot and the settling time [87].

P. Corke [60] implements a PID controller with a simplified quadrotor model (Figure 4.2). This model implements feed-forward control in order to counter the effect of gravity on the altitude control loop to work without disturbance. The alternatives to feed-forward control would be to have very high gain for the altitude loop which often leads to actuator saturation and instability.

4.2.1 Position Controller

The principal goal in $X-Y$ position controller is to regulate the attitude controller by calculating the roll and pitch angles to move the quadrotor between $X-Y$ plane. First, the error between the desired and actual position is calculated, then multiplying by the rotation matrix to transform the error to the $\mathcal{F}_b$ frame, as we can see in the equation. (4-9).

$${}^{I}e_x(t) = {}^{I}x_{des}(t) - {}^{I}y(t)\,, \tag{4-7}$$

$${}^{I}e_y(t) = {}^{I}y_{des}(t) - {}^{I}x(t), \tag{4-8}$$

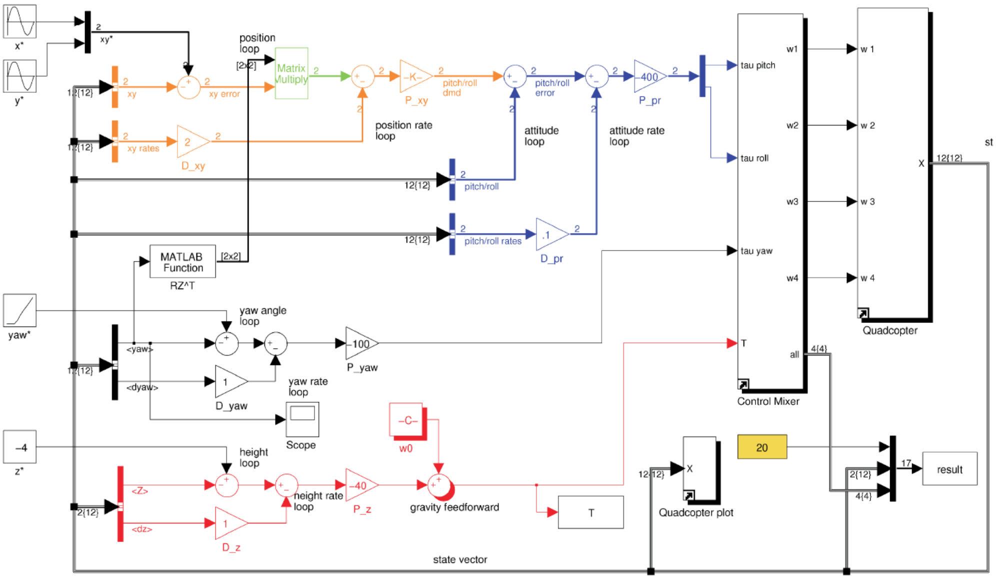


Figure 4.2: The Simulink model with a closed-loop simulation of the quadrotor with PID Control, with a circular trajectory at a constant altitude. [60]

$$\begin{bmatrix} {}^{B}e_x \\ {}^{B}e_y \end{bmatrix} = \begin{bmatrix} cos(\psi) & sin(\psi) \\ -sin(\psi) & cos(\psi) \end{bmatrix} \begin{bmatrix} {}^{I}e_x \\ {}^{I}e_y \end{bmatrix}. \tag{4-9}$$

The controllers that compute the desired attitude use the following control laws:

$$\phi_{des}(t) = K_{P,y}\, {}^{B}e_y(t) + K_{D,y}\, {}^{B}\dot{e}_y(t)\,, \tag{4-10a}$$

$$\theta_{des}(t) = K_{P,x}\, {}^{B}e_x(t) + K_{D,x}\, {}^{B}\dot{e}_x(t)\,. \tag{4-10b}$$

4.2.2 Altitude Controller

The altitude controller is a simple PID controller whose inputs are the error of the desired altitude and the altitude state that comes from the quadrotor's dynamics. The equation that describes this PID controller is the following:

$$u_1(t) = K_{P,z}(z_{des} - z) + K_{I,z}\int_0^t (z_{des} - z) + K_{D,z}(\dot{z}_{des} - \dot{z})\,. \tag{4-11}$$

The output of this PID controller (u_1) is the Thrust deviation from the equilibrium point set at the hover state.

4.2.3 Attitude Controller

An attitude controller was designed to control the quadrotor to orientations that are close to the hover state. These controllers are as follows:

$$p_{des}(t) = K_{P,\phi}(\phi_{des}(t) - \phi(t)) + K_{D,\phi}\ p(t)\,, \tag{4-12a}$$

$$q_{des}(t) = K_{P,\theta}(\theta_{des}(t) - \theta(t)) + K_{D,\theta}\ q(t)\,, \tag{4-12b}$$

$$r_{des}(t) = K_{P,\psi}(\psi_{des}(t) - \psi(t)) + K_{D,\psi}\ r(t)\,. \tag{4-12c}$$

4.2.4 Rate Controller

This controller calculates the input variation from the equilibrium point of the motors in order to create the angular momentum for the variables p, q and r. Three independent controllers are used:

$$u_2(t) = K_{P,p}(p_{des}(t) - p(t)) + K_{D,p}\ \dot{p}(t), \tag{4-13a}$$

$$u_3(t) = K_{P,q}(q_{des}(t) - q(t)) + K_{D,q}\ \dot{q}(t), \tag{4-13b}$$

$$u_4(t) = K_{P,r}(r_{des}(t) - r(t)) + K_{D,r}\ \dot{r}(t). \tag{4-13c}$$

4.2.5 Control Mixer

The output of the rate controller is the total input variation of the motors to generate a torque in the desired direction of movement. Subsequently, the input variation has to be distributed to the motors in the same way as in (2-16) and (2-17), they can move and rotate appropriately using the PWM input received. This analysis can be translated in the following equation:

$$\begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} = \begin{bmatrix} \frac{1}{4k} & 0 & \frac{1}{2lk} & \frac{1}{4b} \\ \frac{1}{4k} & -\frac{1}{2lk} & 0 & -\frac{1}{4b} \\ \frac{1}{4k} & 0 & -\frac{1}{2lk} & \frac{1}{4b} \\ \frac{1}{4k} & \frac{1}{2lk} & 0 & -\frac{1}{4b} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix}. \tag{4-14}$$

4.2.6 Simulation Results

The simulation environment was built in Matlab. The goal of the simulations is to test the control system for tracking a desired position $[x_{des},\ y_{des},\ z_{des}]$ and compare the behaviors of the different controllers.

First test. Try to reach a specific position. The setpoints are

$$
\begin{aligned}
x_{des} &= 1m\,,\\
y_{des} &= 1m\,,\\
z_{des} &= -1m\,,\\
\phi_{des} &= 0\,,\\
\theta_{des} &= 0\,,\\
\psi_{des} &= \frac{\pi}{4}\,.
\end{aligned}
$$

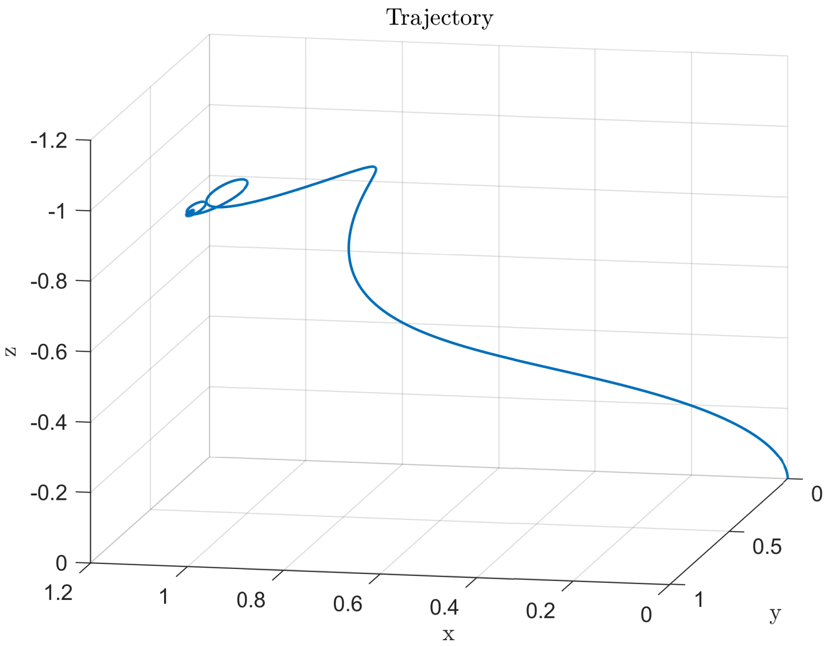


Figure 4.3: Measured position during a test flight, 3D trajectory.

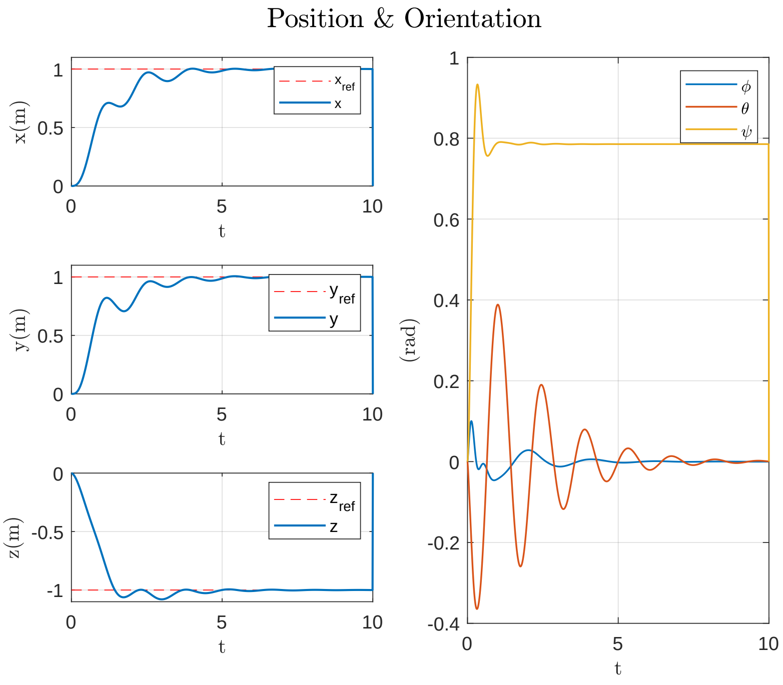


Figure 4.4: Simulation results for Position and Attitude Controller vs Time.

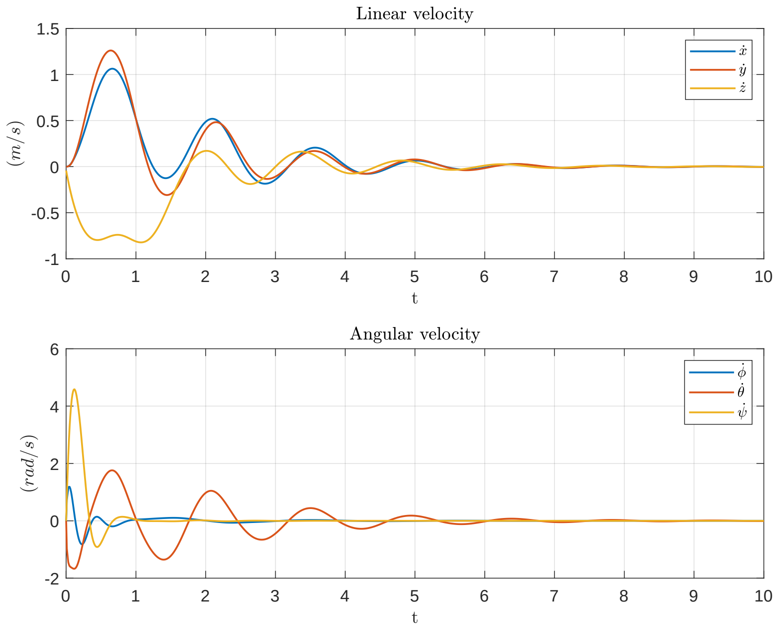


Figure 4.5: Velocity signal controller.

Second test. The quadrotor has to follow in the execution of the assigned motion. Try to track positions that change over time, as a circular path.

$$x(t) = 0.5\ sin(2\pi 0.15\ t - \pi/2)\,,$$
$$y(t) = 0.5\ sin(2\pi 0.15\ t).$$

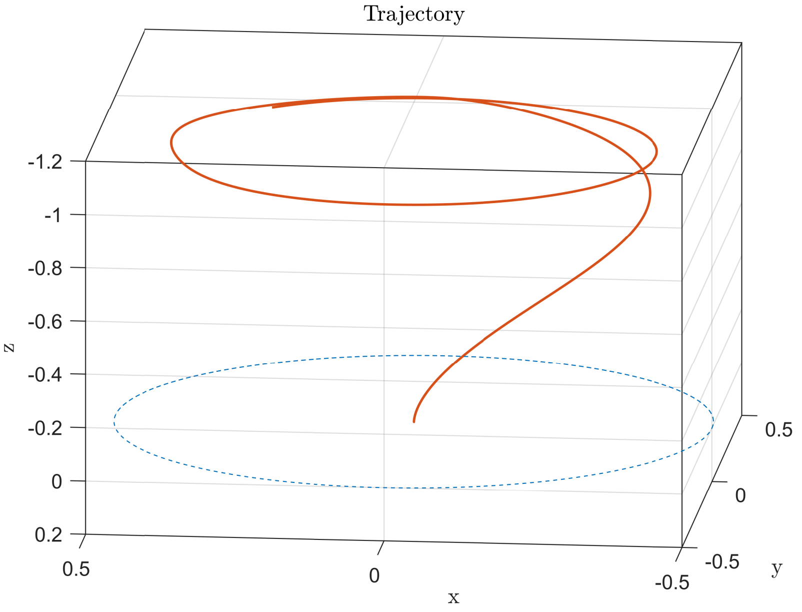


Figure 4.6: 3D Trajectory for second test

This trajectory derivatives a circle with a frequency of 0.15Hz and radius of 0.5 meters, at a constant altitude of 1 meters and constant ψ of $\pi/4$ rad.

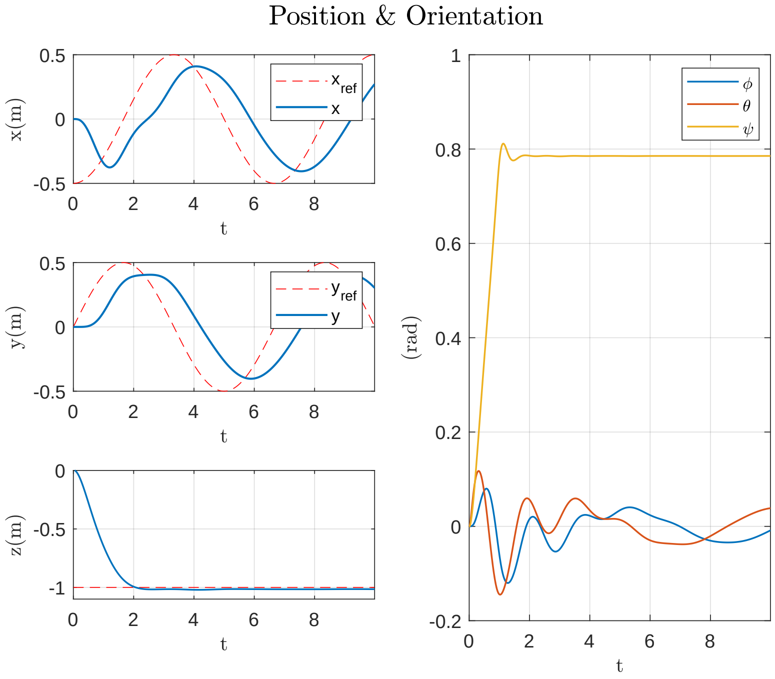


Figure 4.7: Simulation resutls for Position and Attitude Controller in second test

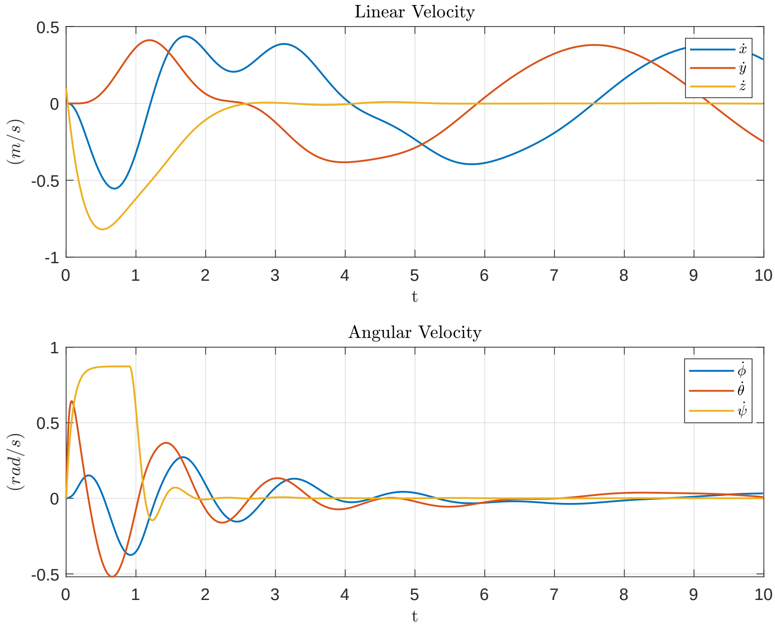


Figure 4.8: Velocity signal controller for circular trajectory

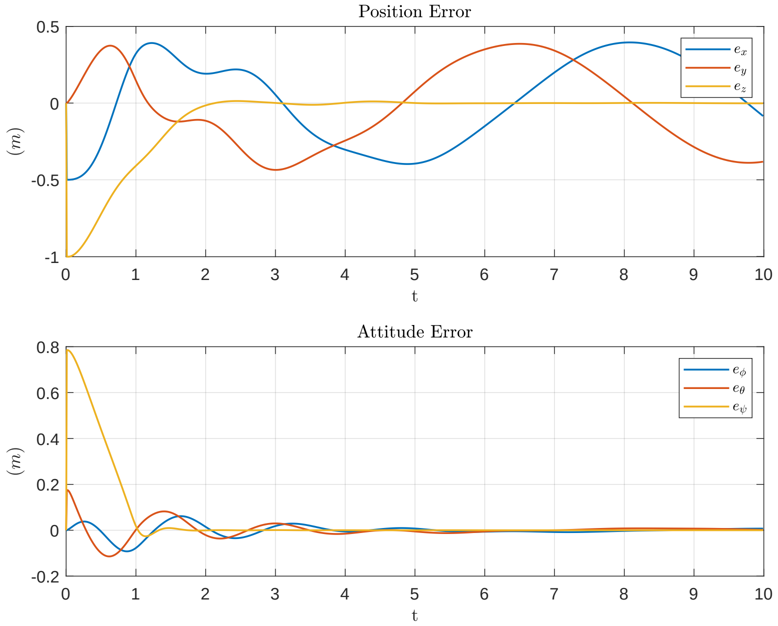


Figure 4.9: Tracking Error for circular trajectory

4.3 Adaptive Linear Control

4.3.1 Pole Placement

We assume that all state variables are measurable and are available for feedback. It will be shown that if the system considered is completely state controllable, then poles of the closed-loop system may be placed at any desired locations by means of state feedback through an appropriate state feedback gain matrix [87].

Consider a control system in the equation:

$$\begin{aligned} \dot{x}(t) &= A\ x(t) + B\ u(t)\,, \\ y(t) &= C\ x(t) + D\ u(t). \end{aligned} \tag{4-15}$$

where x, u, A, B, C and C are defined in the equation (4-15)

We shall choose the control signal to be:

$$u = -Kx. \tag{4-16}$$

The control law signal u is determined by an instantaneous state. The K matrix $1 \times n$, is called the state feedback gain matrix. We assume that all u is unconstrained.

The closed-loop system has no input. Its objective is to maintain the zero output. Because of the disturbances that may be present, the output will deviate from zero. The nonzero output will be returned to the zero reference input because of the state feed-back scheme of the system.

Substituting equation (4-16) into equation (4-15) gives:

$$\dot{x}(t) = (A - BK)x(t). \tag{4-17}$$

Remark 1: The matriz $A - BK$ can be made an asymptotically stable matrix, and for all $x(0) \neq 0$, it is possible to make $x(t)$ approach 0 as t approaches infinity. The eigenvalues of matrix $A - BK$ are called the regulator poles. If these regulator poles are placed in the left-half s plane, then $x(t)$ approaches 0 as t approaches infinity [87].

Remark 2: The pole placement for a given system is possible if and only if the system is completely state controllable [87].

4.3.2 Model Reference Adaptive Control

In adaptive control system there are usually parameters which vary in some unknown manner. So, a adaptive control monitors the performance characteristics of the system and then uses this information to modify the control action to make the overall system perform.

Model Reference Adaptive Control (MRAC) is one such method for adaptive control concerned with forcing the dynamic response of the controlled system to asymptotically approach that of a reference system, despite parametric uncertainties in the plant. The controller compares the output of the plant with the output of the reference model.

4.3.3 Linear Controller Design

After linearization, the design of the linear controller is simple; find a gain matrix K such that the closed-loop system has all the eigenvalues in the left half-plane.

$$\Delta\dot{X} = A\Delta X + B\Delta U\,, \tag{4-18}$$

$$\Delta U = -K\Delta X. \tag{4-19}$$

Closing the loop:

$$\Delta\dot{X} = (A - BK)\Delta X, \tag{4-20}$$

with ψ_{eq} as zero, we aimed to place all the eigen values of the close loop system completely in the $\mathbb{R}$ axis. After the pole placement, the gain matrix

$K = [1 \times 10^6 K_1, 1 \times 10^6 K_2]$ was obtained, where $K \in \mathbb{R}^{M \times N}$ and K_1, K_2 are defined as follows:

$$K_1 = \begin{bmatrix} 0.0673 & -1.2464 & 2.0012 & -2.3448 & 0.8883 & 2.5134 \\ 1.3955 & -1.3902 & 1.2290 & -2.2256 & -2.3260 & -1.9737 \\ 0.3328 & -1.3762 & 2.0032 & -2.4858 & -0.2368 & 2.4978 \\ 1.1913 & -0.9554 & 1.2013 & -0.8203 & -2.1007 & -1.9848 \end{bmatrix},$$

$$K_2 = \begin{bmatrix} 0.1872 & -1.0845 & 0.5054 & -0.1424 & 0.1070 & 0.6804 \\ 1.1099 & -1.0473 & 0.4049 & -0.1676 & -0.1423 & -0.6098 \\ 0.1272 & -1.1680 & 0.5058 & -0.1495 & -0.0453 & 0.6784 \\ 0.9775 & -0.5991 & 0.4016 & 0.0014 & -0.1309 & -0.6111 \end{bmatrix}.$$

The perturbations ΔX and ΔU from Equation (4-18) will be considered as the state and the input itself for convenience sake. Parameter values of the quadrotor will be purpose used incorrectly in order to create an *estimated plant* with parameter mismatch. Then, it will be shown that the adaptive control laws let us track the eqreference model signal more precisely compared to the linear control laws.

$$\text{Estimated plant: } \dot{x}(t) = A_{pm}\, x(t) + B_{pm}\, u(t)\,, \tag{4-21}$$

where pm stands for *parameter mismatch*, and $A_{pm} = A$, since the linearized matrix A does not depend on any parameter of the vehicle. B_{pm} was obtained assuming a miscalculation of m and l by a factor of 1.5, which gives us an overestimation of 50%

$$\text{Reference Model: } \dot{x}_m(t) = A_m\, x_m(t) + B_m\, r(t). \tag{4-22}$$

For the model in (4-22) we used the real linearized model of our system. In our case $r(t)$ is the reference model control input that makes $\dot{x}_m$ achieve hover at the desired location.

Controller:

$$u(t) = \hat{K}_x^{\mathsf{T}}(t)\, x(t) + \hat{K}_r^{\mathsf{T}}(t)\, r(t)\,, \tag{4-23}$$

where $\hat{K}_x \in \mathbb{R}^{N \times M}$, $\hat{K}_r \in \mathbb{R}^{M \times M}$ with $N = 12$ as the dimension of our state-space are the estimates of the ideal unknown matrices, and $M = 4$ is the dimension of the input space.

Adaptation laws:

$$\dot{\hat{K}}_x(t) = -\Gamma_x \, x(t) e^{\mathsf{T}}(t) \, P \, B \,, \tag{4-24}$$

$$\dot{\hat{K}}_r(t) = -\Gamma_r \, r(t) e^{\mathsf{T}}(t) \, P \, B \,, \tag{4-25}$$

where $e(t)$ is the tracking error defined as:

$$e(t) = x(t) - x_m(t). \tag{4-26}$$

P is the positive define matrix found solving the Lyapunov equation, and Γ is the arbitrary adaptation gain matrix. We are using the static matrix K that is being used in the Linear Controller as the initial condition of our dynamic matrix K_x and $I_{4\times4}$ as the initial condition of K_r, hence $K_x^{\mathsf{T}}(0) = K$ and $K_r^{\mathsf{T}}(0) = I_{4\times4}$.

Theorem 4.1 *Considering the MIMO system model* (4-15) *with a tracking system* (4-21) *and reference model dynamics* (4-22) *enforces global asymptotic stability with the control law* (4-23)*. Then, the tracking error* (4-26) *tends to zero, i.e.*

$$\lim_{t\to\infty} \|x(t) - x_m(t)\| = 0. \tag{4-27}$$

Proof. The ideal fixed-gain control law:

$$u(t) = K_x^{\mathsf{T}}(t) \, x(t) + K_r^{\mathsf{T}}(t) \, r(t) \,, \tag{4-28}$$

Substituting (4-28) into (4-15), obtain the closed-loop system:

$$\dot{x} = (A + BK_x^{\mathsf{T}}) \, x(t) + BK_r^{\mathsf{T}} \, r(t).$$

Comparing (4.3.3) with desired reference (4-22), for existence of a controller in (4-23), the ideal unknown control gains, K_x and K_r, have to satisfy:

$$\begin{aligned} A_m &= A + BK_x^{\mathsf{T}}, \\ B_m &= BK_r^{\mathsf{T}}. \end{aligned} \tag{4-29}$$

For any bounded reference input signal $r(t)$, the fixed-gain controller (4-28) provides global asymptotic tracking performance [62].

Assuming that gains K_x and K_r exist, consider the control law defined in (4-23) have estimated parameters generated online method.

Substituting (4-28) into (4-15), the closed–loop system dynamics can be written as:

$$\dot{x} = (A + B\hat{K}_x^{\mathsf{T}}) \, x(t) + B\hat{K}_r^{\mathsf{T}} \, r(t). \tag{4-30}$$

Subtracting (4-22) from (4-30) and consider (4-29), the tracking error is given by:

$$\dot{e} = A_m e + B(\tilde{K}_x^{\mathsf{T}} x(t) + \tilde{K}_r^{\mathsf{T}} r(t)), \tag{4-31}$$

with $\tilde{K}_x = \hat{K}_x - K_x$, $\tilde{K}_r = \hat{K}_r - K_r$ are the parameter estimation error [54].

In order to guarantee stability of the tracking error dynamics (4-31), consider the Lyapunov function:

$$V(e, \tilde{K}_x, \tilde{K}_r) = e^{\mathsf{T}} P e + \tilde{K}_x^{\mathsf{T}} \Gamma_x^{-1} \tilde{K}_x + \tilde{K}_r^{\mathsf{T}} \Gamma_r^{-1} \tilde{K}_r, \tag{4-32}$$

where P is the solution of the Lyapunov equation:

$$P\,A_m + A_m^{\mathsf{T}}\,P = -Q. \tag{4-33}$$

The time derivative $\dot{V}$ is given by:

$$\dot{V} = \dot{e}^{\mathsf{T}} P e + e^{\mathsf{T}} P \dot{e} + 2\tilde{K}_x^{\mathsf{T}} \Gamma_x^{-1} \dot{\hat{K}}_x + 2\tilde{K}_r^{\mathsf{T}} \Gamma_r^{-1} \dot{\hat{K}}_r. \tag{4-34}$$

Considering the Lyapunov equation, expanding and rearranging terms yields

$$\begin{aligned} \dot{V} = -e^{\mathsf{T}} Q e + 2(e^{\mathsf{T}} P B \tilde{K}_x^{\mathsf{T}} x(t) + \tilde{K}_x^{\mathsf{T}} \Gamma_x^{-1} \dot{\hat{K}}_x) + \\ 2(e^{\mathsf{T}} P B \tilde{K}_r^{\mathsf{T}} r(t) + \tilde{K}_r^{\mathsf{T}} \Gamma_r^{-1} \dot{\hat{K}}_r). \end{aligned} \tag{4-35}$$

Thus, selecting the adaptive laws in (4-24) and (4-25):

$$\begin{aligned} \dot{\hat{K}}_x &= -\Gamma_x x(t) e^{\mathsf{T}} P B, \\ \dot{\hat{K}}_r &= -\Gamma_r r(t) e^{\mathsf{T}} P B. \end{aligned}$$

the time derivative $\dot{V}$ becomes $\dot{V} = -e^{\mathsf{T}} Q e \leq 0$. Therefore, the error dynamics is asymptotically stable. ■

4.3.4 Simulation Results

We show the result of the 12 linearized states the linear controller by the equation (4-19), as shown figures 4.10-4.13. The system convergs to the desired hover position, where the first three states are $[x, y, z]^{\mathsf{T}} = [1, 1, -1]^{\mathsf{T}}$.

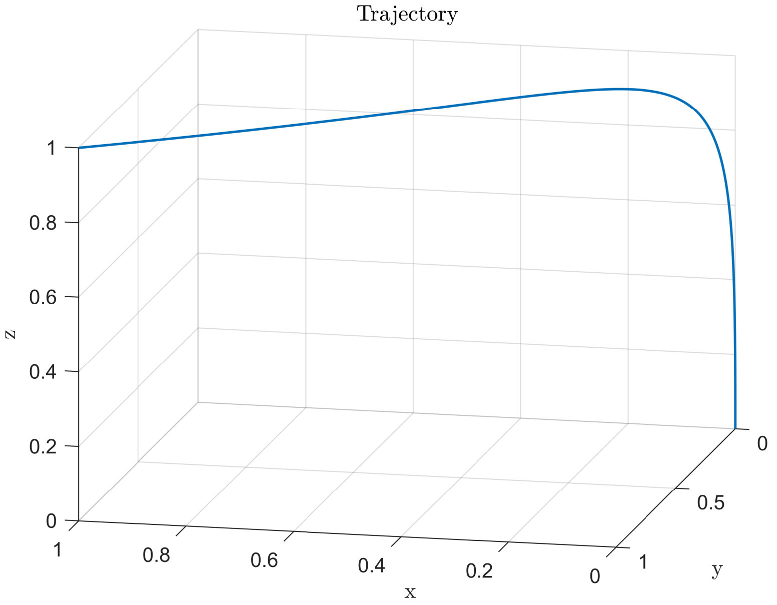


Figure 4.10: 3D Trajectory test for linear model

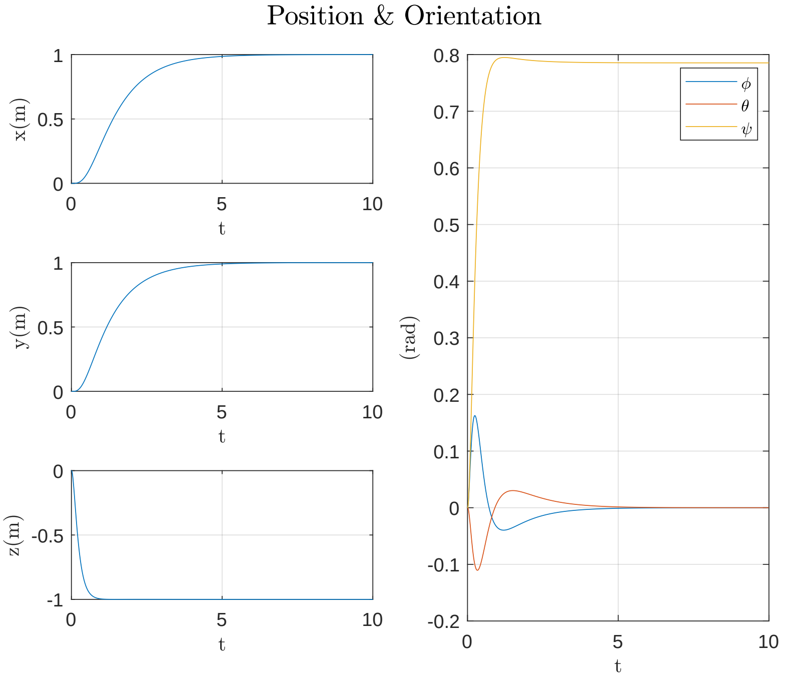


Figure 4.11: Simulation resutls for Position and Attitude Controller for linear State-Space Controller

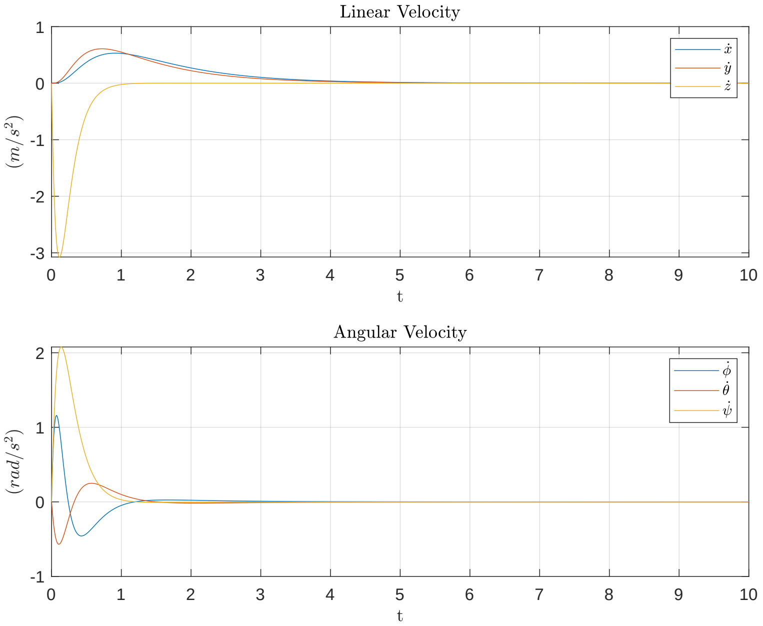


Figure 4.12: Velocity signal controller for linear State-Space Controller

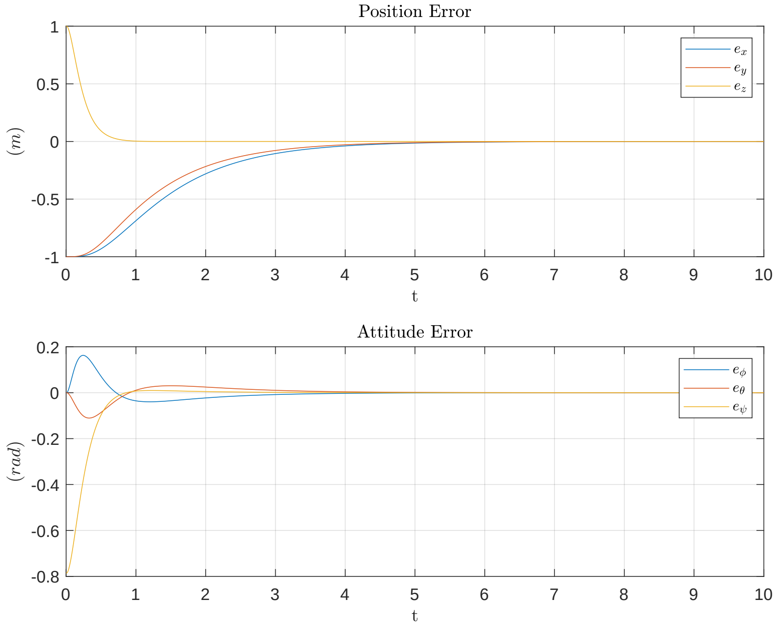


Figure 4.13: Tracking Error for linear State-Space Controller

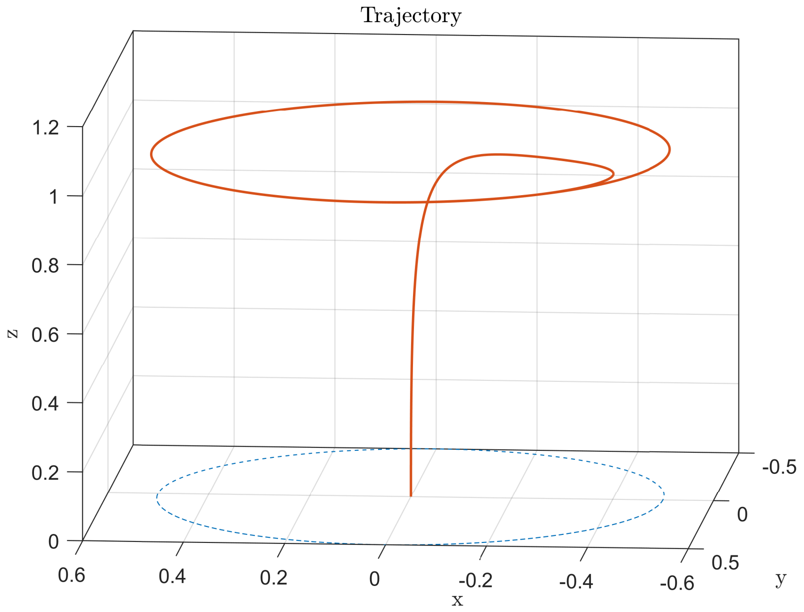


Figure 4.14: 3D Trajectory test for linear model for linear State-Space Controller

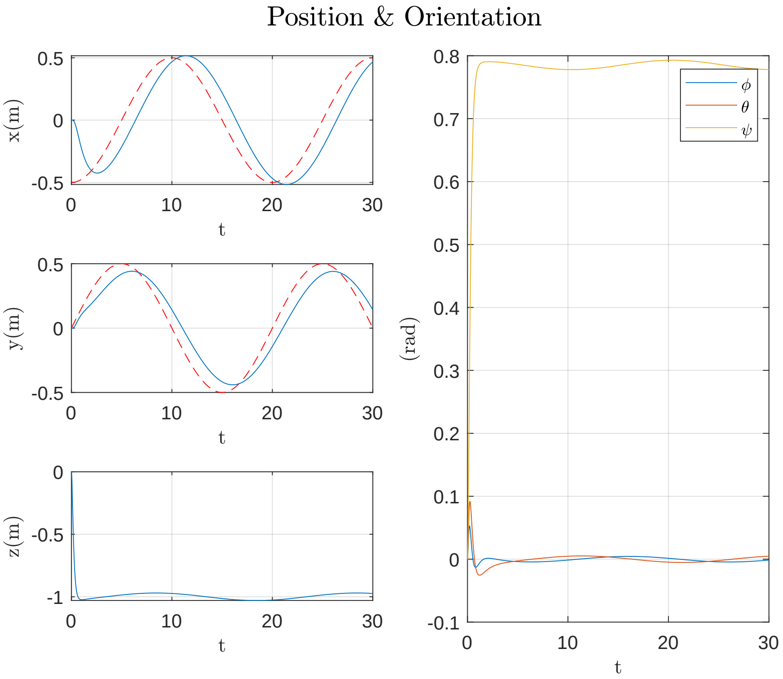


Figure 4.15: Simulation resutls for Position and Attitude Controller in linear model

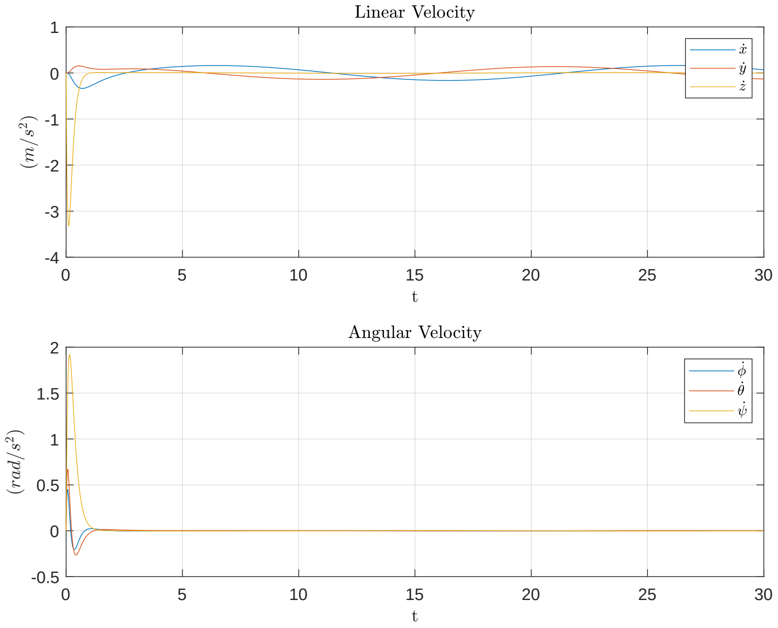


Figure 4.16: Velocity signal controller for circular trajectory in for linear State-Space Controller

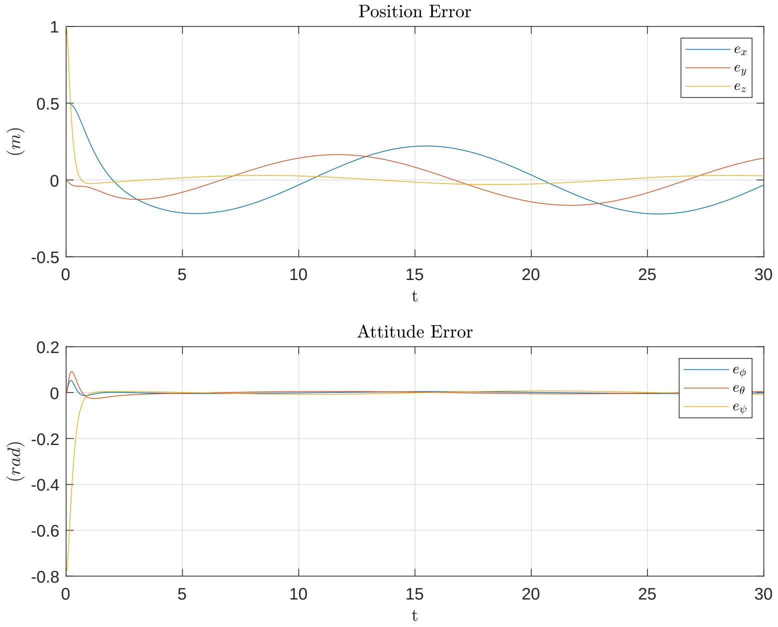


Figure 4.17: Tracking Error for circular trajectory for linear State-Space Controller

Figures 4.18 and 4.19 correspond to the response of the system with the linear adaptive controller with parameter uncertainties. Figure 4.18 shows the evolution of the vehicle's position and orientation during its flight. Figure 4.19 represents the tracking responses of the quadrotor position and attitude. It can be seen that the controller in the equation (**??**)-(4-25) performs is better tracking of the reference model than the regular linear controller. It is assumed that the quadrotor mass has an uncertainty of 80 % uncertainty and the arm length has 10% uncertainty.

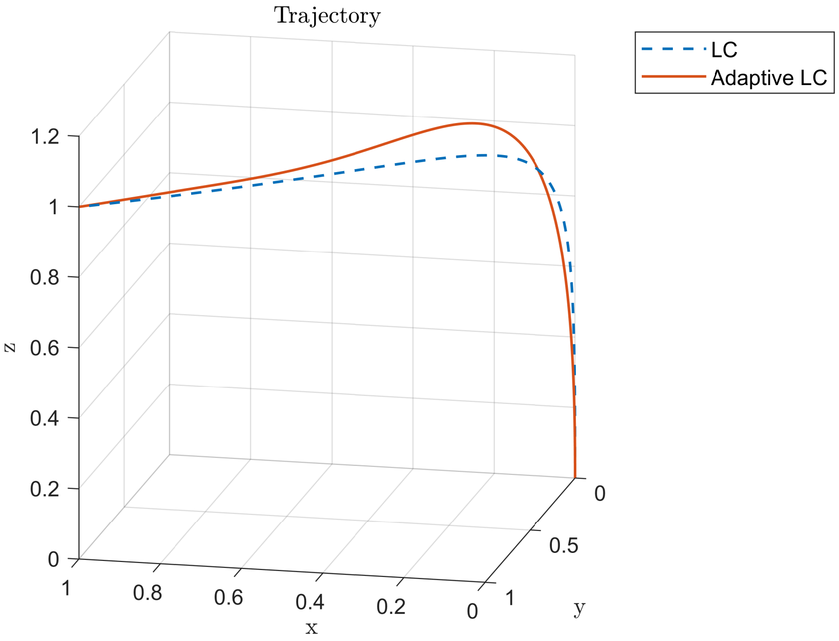


Figure 4.18: Attitude Controller for Adaptive linear controller

4.4 Adaptive Non-linear Control

Due to the high non-linearity and complexity of the dynamical model, the control problem of quadrotors has been considered extensively, since the underactuated system. Our control is the combination of the altitude control in the inertial coordinate frame and the attitude tracking parameterization in the body coordinate frame that simplifies the representation of the adaptation scheme.

4.4.1 Altitude Control

The equation (4-4c) represents the dynamics of $\ddot{z}$

$$\ddot{z} = -\frac{u_1}{m}\, c_\phi\, c_\theta + g. \tag{4-36}$$

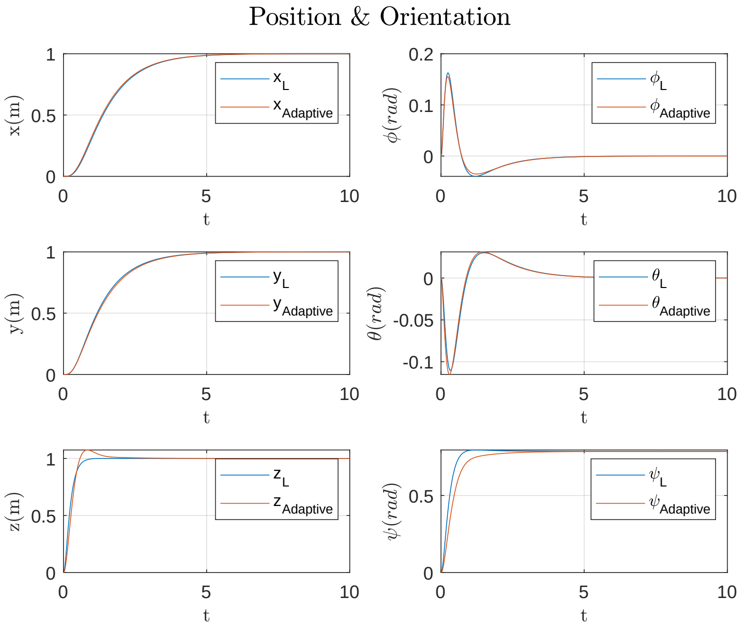


Figure 4.19: 3D Trajectory test with linear model for Adaptive linear Controller

The tracking error is defined as $\tilde{z} = z - z_{des}$, while

$$\boldsymbol{s}_z = \ddot{\tilde{z}} - \lambda_z * \tilde{z}. \tag{4-37}$$

is used as a measure tracking, where λ is positive constant. Equation (4-37) can be written as:

$$\boldsymbol{s}_z = \dot{z} - \dot{z}_r. \tag{4-38}$$

The proposed control law is as follows:

$$u_1 = \hat{m}\;\ddot{z} + \hat{m}\;g - K_{D_z}\;\boldsymbol{s}_z. \tag{4-39}$$

Theorem 4.2 *Consider the dynamical model of quadrotor for height equation* (4-36)*. Under the designed controllers by equations* (4-39), (4-37) *and* (4-38)*, the nonlinear system is asymptotically stable.*

Proof. Combining (4-38) and (4-39) yields

$$m\ddot{z} + mg = \hat{m}\ddot{z}_r + \hat{m}g - K_{D_z}\,, \tag{4-40}$$

$$m\ddot{z} - m\ddot{z}_r = (\hat{m} - m)\ddot{z}_r + (\hat{m} - m)g - K_{D_z}\,, \tag{4-41}$$

$$m\dot{\boldsymbol{s}}_z + K_{D_z} = \mathbf{Y}(\ddot{z}_r)\tilde{\Phi}_z. \tag{4-42}$$

The Lyapunov function candidate might to be positive definite and its time derivate will be negative definite. In other words:

$$2V(s,\tilde{\Phi}_z) = ms^2 + \gamma_z^{-1}\tilde{\Phi}^2\,, \tag{4-43}$$

$$\dot{V} = -K_{D_z}s^2 + s\mathbf{Y}(\ddot{z}_r)\tilde{\Phi}_z + \gamma_z^{-1}\tilde{\Phi}_z\dot{\hat{\Phi}}_z. \tag{4-44}$$

Assuming that $\dot{\Phi}_z = 0$, yields the parameter adaptation law:

$$\dot{\hat{\Phi}}_z = -\gamma_z\mathbf{Y}(\ddot{z}_r)s\,, \tag{4-45}$$

which implies that $\dot{V} = -K_{D_z}s_z^2 \leq 0$. This shows that $s_z \to 0$ and $\tilde{z} \to 0$. ■

4.4.2 Attitude Control

Consider a system written as follows:

$$\begin{aligned} H\ddot{q} + C(\dot{q},x)\dot{q} + g(x) &= \tau\,, \\ \dot{x} &= J(x)\ q. \end{aligned} \tag{4-46}$$

where $q\,in\,\mathbb{R}^N$ is the local coordinates, $x \in \mathbb{R}^N$ is a vector in the inertial frame and $\tau \in \mathbb{R}^N$ is the vector of input torques, H is the inertia matrix for the system, and $C\dot{q}$ is a non-linear matrix of coriolis and sentripetal forces, $g(x) \in \mathbb{R}^N$ is the gravity vector and J is an $n \times n$ transformation matrix. Slotine [54] write the equation (4-46) as follows:

$$H^\star(x)\ddot{x} + C^\star(x,\dot{x})\dot{x} + g^\star(x) = J^{-T}\ \tau. \tag{4-47}$$

For this work, the Slotine's theory was using for attitude control, using the gravity force. So, equation (4-47) can be write as:

$$H^\star(x)\ddot{x} + C^\star(x,\dot{x})\dot{x} = F. \tag{4-48}$$

Now, the matrices $H^\star$ and $C^\star$ have to be calculated. From equation (4-3) we have:

$$H(q)\dot{\omega} = C\omega + \tau, \tag{4-49}$$

where:

$$H = I\,,$$

$$C = -\omega \times I\omega.$$

From equation (2-4), we obtain:

$$\dot{\omega} = J^1\ddot{\eta} - J^{-1}\dot{J}\omega. \tag{4-50}$$

Replacing (4-50) in (4-49) and multiplying by $J^{-\mathsf{T}}$ leads to:

$$H^\star = J^{-\mathsf{T}} H J^{-1}\,, \tag{4-51}$$

$$C^\star = -J^{-\mathsf{T}}(HJ^{-1}\dot{J} + C)J^{-1}\,, \tag{4-52}$$

$$F = J^{-\mathsf{T}}\tau. \tag{4-53}$$

The desired trajectory $\ddot{\eta}_{des}$, $\dot{\eta}_{des}$, and η_{des} are assumed to be bounded. Moreover, change the variables $q = \omega$ and $x = \eta$.

The tracking error vector $\tilde{x}$ is defined as $\tilde{x} = x - x_{des}$, while

$$s = \ddot{\tilde{x}} + \lambda\tilde{x}\,, \tag{4-54}$$

is used as a measure of tracking, where λ is a strictly positive constant.

It is convenient to rewrite (4-54) as follows:

$$s = \dot{x} - \dot{x}_r, \quad \text{where} \quad \dot{x}_r = \dot{x}_{des} - \lambda\tilde{x}. \tag{4-55}$$

It is important to notice that the terms $H^\star$ and $C^\star$ are linear in their parameters. The unknown parameters are lumped together into a parameter vector Θ. Let $\hat{\Theta}$ be the time-varying parameter vector estimate and let $\tilde{\Theta} = \hat{\Theta} - \Theta$ be the parameter error vector.

Theorem 4.3 *Given the lyapunov equation for control law[54]:*

$$V(s, \tilde{\Theta}, t) = \frac{1}{2}s^{-\mathsf{T}}H^\star s + \frac{1}{2}\tilde{\Theta}^{-\mathsf{T}}\Gamma\tilde{\Theta}, \tag{4-56}$$

where $\Gamma \in \mathbb{R}^{3\times3}$*. Then, the error defined as* (4-55) *converges to zero and the quadrotor system* (4-47) *is globally stable.*

Proof. Differentiating V with respect to time yields:

$$\dot{V} = s^{\mathsf{T}}(J^{-\mathsf{T}}\tau - H^\star\ddot{x}_r - C^\star\dot{x}_r) + \tilde{\Theta}\Gamma^{\dot{-}1}\tilde{\Theta} \tag{4-57}$$

This suggests that the control law can be selected as

$$\tau = J^{\mathsf{T}}(\hat{H}^\star\ddot{x}_r + \hat{C}^\star\dot{x}_r - K_D\ s), \tag{4-58}$$

where $\hat{H}$ and $\hat{C}$ denotes the adaptive estimates and $K_D \in \mathbb{R}^{3\times3}$.

Combining (4-57) and (4-58) yields

$$\dot{V} = -s^{\mathsf{T}}K_D\ s + s^{\mathsf{T}}(\tilde{H}^\star\ddot{x}_r + \tilde{C}^\star\dot{x}_r) + \dot{\tilde{\Theta}}\Gamma^{-1}\tilde{\Theta}\,, \tag{4-59}$$

where $\tilde{H}^\star = \hat{H}^\star - H^\star$ and $\tilde{C}^\star = \hat{C}^\star - C^\star$. Using the parametrization from [54], implies that

$$H^\star\ddot{x}_r + C^\star\dot{x}_r = \Phi^\star(x, \dot{x}, \dot{x}_r, \ddot{x}_r)\Theta\,, \tag{4-60}$$

where $\Phi^\star(x, \dot{x}, \dot{x}_r, \ddot{x}_r)$ is a known regressor matrix of appropriate dimensions. Hence(4-57) may be written as follows:

$$\dot{V} = -s^{\mathsf{T}}K_D\ s + \tilde{\Theta}^{\mathsf{T}}(\Gamma^{-1}\dot{\tilde{\Theta}} + \Phi^{\star\mathsf{T}}\ s). \tag{4-61}$$

Assuming that $\dot{\Theta} = 0$, yields the parameter adaption law:

$$\dot{\hat{\Theta}} = -\Gamma \Phi^{\star\mathsf{T}}(x, \dot{x}, \dot{x}_r, \ddot{x}_r)s\,, \tag{4-62}$$

which implies that $\dot{V} = -s^{\mathsf{T}} K_D\ s \leq 0$, showing that $s \to 0$ and $\tilde{x} \to 0$. ■

4.4.2.1 Reparameterization

The previous scheme can be simplified by the definition of the vector $\dot{q}_r$:

$$\dot{x}_r = J(x)\dot{q}_r. \tag{4-63}$$

This implies that $\dot{q}_r$, and $\ddot{q}_r$, may be calculated as follows:

$$\begin{aligned} \dot{q}_r &= J^{-1}(x)\dot{x}_r\,, \\ \ddot{q}_r &= J^{-1}(x)\ddot{x}_r - J^{-1}(x)\dot{J}(x)J^{-1}(x)\dot{x}_r. \end{aligned} \tag{4-64}$$

So, equation (4-60) may be rewritten as follows:

$$\begin{aligned} J^{\mathsf{T}}(H^{\star}\ddot{x}_r + C^{\star}\dot{x}_r) &= H\ddot{q}_r + C\dot{q}_r \\ &= \Phi(x, \dot{q}, \dot{q}_r, \ddot{q}_r)\Theta. \end{aligned} \tag{4-65}$$

By using the vector q_r, instead of x_r, the known transformation matrix $J(x)$ is eliminated from the new regressor matrix $\Phi(x, \dot{q}, \dot{q}_r, \ddot{q}_r)$. Using the parameterization of (4-65), the following expression for $\dot{V}$ is found

$$\dot{V} = s^{\mathsf{T}} J^{-\mathsf{T}}(\tau - H\ddot{q}_r - C\dot{q}_r) + \dot{\hat{\Theta}}^{\mathsf{T}}\Gamma^{-1}\tilde{\Theta}. \tag{4-66}$$

The control law (4-58) then simplifies to

$$\tau = \hat{H}\ddot{q}_R + \hat{C}\dot{q}_r - J^{\mathsf{T}} K_D s\,, \tag{4-67}$$

while the adaption law (4-65) is modified to:

$$\dot{\hat{\Theta}} = -\Gamma \Phi^{\mathsf{T}}(x, \dot{q}, \dot{q}_r, \ddot{q}_r)J^{-1}s\,, \tag{4-68}$$

Notice that that the new control law is written in terms of $\hat{H}$, $\hat{C}$ instead of $\hat{H}^{\star}$ and $\hat{C}^{\star}$. Hence, Φ is given in body coordinates, which suggesting that the elements can be found by inspection or recursive methods.

Theorem 4.4 *The adaptive law:*

$$\dot{\hat{\Theta}} = -\Gamma \Phi^{\mathsf{T}}(x, \dot{q}, \dot{q}_r, \ddot{q}_r)J^{-1}s\,, \tag{4-69}$$

ensure the globally stability for the control law (4-66)*, which again implies that* $\dot{V} = -s^{\mathsf{T}} K_D\ s \leq 0$.

4.4.3 Simulation Results

In this section, simulation results are shown for the tracking attitude and altitude. The initial state is $[z, \phi, \theta, \psi] = [0, 0, 0, 0]$ meaning that, linear and angular velocities are chosen equal to zeros. The gains of the adaptive reaching laws are $\gamma_z = 0.01$ and $\Gamma = I_{3\times3}$. Figure 4.20 shows the altitude and attitude tracking. The Control input signals ($T(t)$ and $\tau(t)$) are shown in Figure 4.21. The adaptive estimations are presented in Figure 4.22.

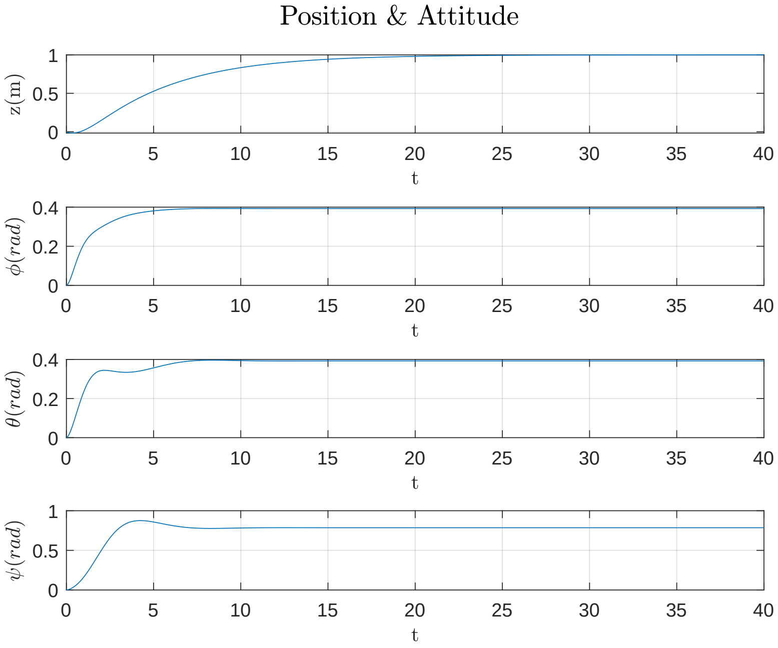


Figure 4.20: Tracking Error for altitude and attitude controller in non-linear model.

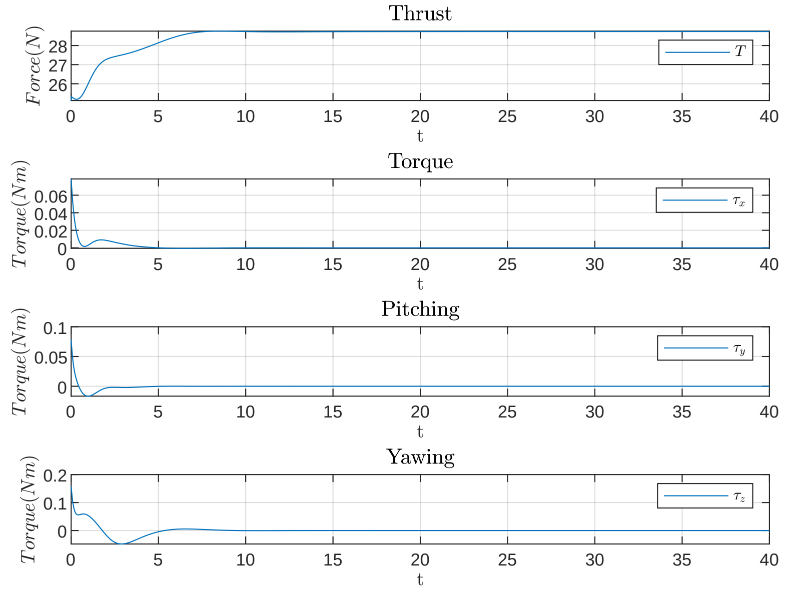


Figure 4.21: Control input signals (T, τ_x, τ_y, τ_z)

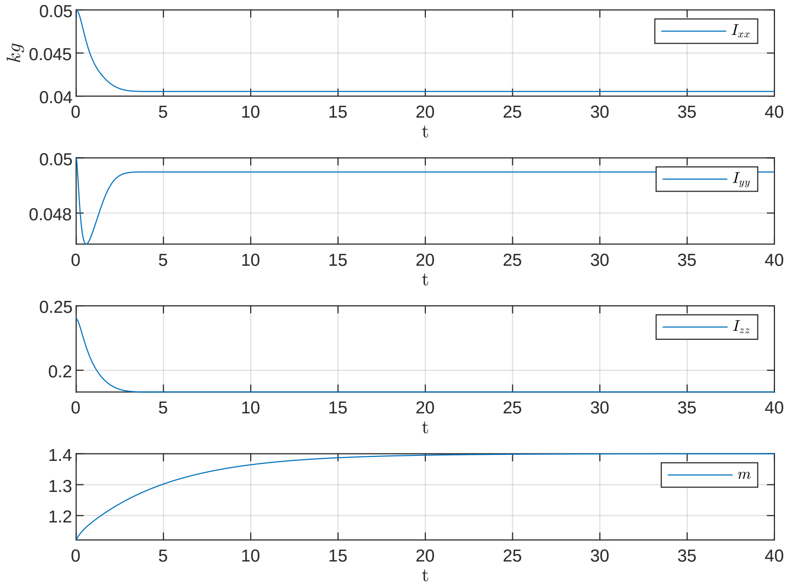


Figure 4.22: Parameters estimation

4.5 Conclusions

This chapter has presented a description of the control algorithms used within this dissertation. Three controllers were described: PD controller and Adaptive Control for Lineal and non-linear model, despite parametric uncertainties.

5
Concluding Remarks

Drones are becoming more readily available and the extent of their possible applications seems to grow endlessly.

5.1
Conclusions

In this thesis, we present and discussed a model that describes the quadrotor's behaviour, including the actuators. The model describes the equations of motion for a translating and rotating body in 3D space. This model receive PWM signals for motor control actuation. In Chapter 1, we review about UAVs (drones) and quadrotor models, controllers and applications, highlighting applications in agriculture Next, Chapter 2 shows the development of a mathematical model, with a simplified dynamics, that help to understand how to work with the principal forces and moments.

Chapter 3 presents the development of a simple monocular vision navigation for UAV in row crops, implemented and simulated in Gazebo. A simple method of vanishing point estimating is proposed. An algorithm based on the position of vanishing point modifies the velocities v_x, v_y, and v_z accordingly.

In the Chapter 4, the control laws was discussed, implemented and simulated in MATLAB. Initially, we introduce and discuss cascade internal and external control for a linear control like PID, showing that the aforementioned controller is easy to understand and implement, but is not robust to parameter variations. Then, we derive a state-space model to develop a controller for parameter changes, linear and non-linear adaptive control, with higher performance to adapt parameters such as mass, arm length, and inertia. In a non-linear case, we ensure Lyapunov stability, and not canceling of system non-linearities.

5.2
Perspective and Future Works

Following the research developed in this dissertation, corresponding to aspects that have not been solved or that requiere more analysis to improve the performance of the proposed strategies. We propose future works:

1. **Non-linear control.** Although we implemented several controllers we work mainly with linear control and adaptive non-linear but there was a topic that have been not fully covered and thus can be of interest for future research. The adaptive non-linear control work on altitude and attitude control, the position control can be research to work with Adaptive or Sliding mode which is more robust than linear control.

2. **Line recognition.** We recognize lines inside an ROI window that is accomplished when the drone is at a minimum height of 1.2m and maximum height of 2.5m for real environment. To improve the algorithm performance, it would be optimal recognize several row crops.

Bibliography

[1] LEUTENEGGER, S.; HÜRZELER, C.; STOWERS, A. K.; ALEXIS, K.; ACHTELIK, M. W.; LENTINK, D.; OH, P. Y. ; SIEGWART, R.. **Flying Robots**. In Siciliano and Khatib [14], p. 623–670.

[2] KENDOUL, F.. **Survey of advances in guidance, navigation, and control of unmanned rotorcraft systems**. Journal of Field Robotics, 29(2):315–378, 2012.

[3] SHAKERI, R.; AL-GARADI, M. A.; BADAWY, A.; MOHAMED, A.; KHATTAB, T.; AL-ALI, A. K.; HARRAS, K. A. ; GUIZANI, M.. **Design Challenges of Multi-UAV Systems in Cyber-Physical Applications: A Comprehensive Survey, and Future Directions**. CoRR, abs/1810.09729, 2018.

[4] WILD, K. J.; SCHMIEDEL, T. ; SCHUELLER, J. K.. **Concept for Using Unmanned Aerial Vehicles for a Continuous Provision of Information for Online Application in Precision Farming**. In: 2017 ASABE ANNUAL INTERNATIONAL MEETING, p. 1. American Society of Agricultural and Biological Engineers, 2017.

[5] DOERING, D.; BENENMANN, A.; LERM, R.; DE FREITAS, E. P.; MULLER, I.; WINTER, J. M. ; PEREIRA, C. E.. **Design and Optimization of a Heterogeneous Platform for multiple UAV use in Precision Agriculture Applications**. IFAC Proceedings Volumes, 47(3):12272 – 12277, 2014. 19th IFAC World Congress.

[6] MAES, W. H.; STEPPE, K.. **Perspectives for Remote Sensing with Unmanned Aerial Vehicles in Precision Agriculture**. Trends in Plant Science, 24(2):152–164, 2019.

[7] FAIÇAL, B. S.; FREITAS, H.; GOMES, P. H.; MANO, L. Y.; PESSIN, G.; DE CARVALHO, A. C.; KRISHNAMACHARI, B. ; UEYAMA, J.. **An Adaptive Approach for UAV-based Pesticide Spraying in Dynamic Environments**. Computers and Electronics in Agriculture, 138:210–223, 2017.

[8] MYERS, D.; ROSS, C. M. ; LIU, B.. **A Review of Unmanned Aircraft System (UAS) Applications for Agriculture**. In: 2015 ASABE ANNUAL INTERNATIONAL MEETING, p. 1. American Society of Agricultural and Biological Engineers, 2015.

[9] ROBBINS, J. A.. **Small Unmanned Aircraft Systems (sUAS)**. In: HORTICULTURAL REVIEWS, p. 33–71. John Wiley & Sons, Inc., Feb. 2018.

[10] NAN, Y.; KARÁSEK, M.; LALAMI, M. E. ; PREUMONT, A.. **Experimental optimization of wing shape for a hummingbird-like flapping wing micro air vehicle**. Bioinspiration & biomimetics, 12(2):026010, 2017.

[11] CHIN, D. D.; LENTINK, D.. **Flapping wing aerodynamics: from insects to vertebrates**. Journal of Experimental Biology, 219(7):920–932, 2016.

[12] SYLVESTER, G.. **E-agriculture in Action: Drones for Agriculture**. Technical report, Food and Agriculture Organization of the United Nations, 2018.

[13] MAHONY, R. E.; BEARD, R. W. ; KUMAR, V.. **Modeling and Control of Aerial Robots**. In Siciliano and Khatib [14], p. 1307–1334.

[14] Siciliano, B.; Khatib, O., editors. **Springer Handbook of Robotics**. Springer Handbooks. Springer, 2nd edition, 2016.

[15] EBEID, E.; SKRIVER, M.; TERKILDSEN, K. H.; JENSEN, K. ; SCHULTZ, U. P.. **A Survey of Open-Source UAV Flight Controllers and Flight Simulators**. Microprocessors and Microsystems, 61:11–20, 2018.

[16] MUKHERJEE, A.; MISRA, S. ; RAGHUWANSHI, N. S.. **A Survey of Unmanned Aerial Sensing Solutions in Precision Agriculture**. Journal of Network and Computer Applications, 148:1–24, 2019.

[17] CATUREGLI, L.; CORNIGLIA, M.; GAETANI, M.; GROSSI, N.; MAGNI, S.; MIGLIAZZI, M.; ANGELINI, L.; MAZZONCINI, M.; SILVESTRI, N.; FONTANELLI, M.; RAFFAELLI, M.; PERUZZI, A. ; VOLTERRANI, M.. **Unmanned Aerial Vehicle to Estimate Nitrogen Status of Turfgrasses**. PLOS ONE, 11:1–13, 06 2016.

[18] CARLOS HENRIQUE WACHHOLZ DE SOUZA AND RUBENS AUGUSTO CAMARGO LAMPARELLI AND JANSLE VIEIRA ROCHA AND PAULO SERGIO GRAZIANO MAGALHÃES. **Mapping Skips in Sugarcane**

Fields Using Object-based Analysis of Unmanned Aerial Vehicle (UAV) Images. Computers and Electronics in Agriculture, 143:49 – 56, 2017.

[19] ZHENG, H.; ZHOU, X.; HE, J.; YAO, X.; CHENG, T.; ZHU, Y.; CAO, W. ; TIAN, Y.. **Early Season Detection of Rice Plants Using RGB, NIR-G-B and Multispectral Images from Unmanned Aerial Vehicle (UAV)**. Computers and Electronics in Agriculture, 169:105223, 2020.

[20] DENG, L.; MAO, Z.; LI, X.; HU, Z.; DUAN, F. ; YAN, Y.. **UAV-based multispectral remote sensing for precision agriculture: A comparison between different cameras**. ISPRS Journal of Photogrammetry and Remote Sensing, 146:124 – 136, 2018.

[21] POLKA, M.; PTAK, S. ; KUZIORA, L.. **The use of uav's for search and rescue operations**. Procedia Engineering, 192:748 – 752, 2017.

[23] WANG, X.; SUN, H.; LONG, Y.; ZHENG, L.; LIU, H. ; LI, M.. **Development of Visualization System for Agricultural UAV Crop Growth Information Collection**. IFAC-PapersOnLine, 51(17):631 – 636, 2018. 6th IFAC Conference on Bio-Robotics BIOROBOTICS 2018.

[24] GAULTON, R.; TAYLOR, J. ; WATKINS, N.. **Unmanned aerial vehicles for pre-harvest biomass estimation in willow (salix spp.) coppice plantations**. ISPRS Geospatial Week, 2015.

[25] MISOPOLINOS, L.; ZALIDIS, C.; LIAKOPOULOS, V.; STAVRIDOU, D.; KATSIGIANNIS, P.; ALEXANDRIDIS, T. K. ; ZALIDIS, G.. **Development of a UAV system for VNIR-TIR acquisitions in precision agriculture**. In: Hadjimitsis, D. G.; Themistocleous, K.; Michaelides, S. ; Papadavid, G., editors, THIRD INTERNATIONAL CONFERENCE ON REMOTE SENSING AND GEOINFORMATION OF THE ENVIRONMENT (RSCY2015), volumen 9535, p. 478 – 487. International Society for Optics and Photonics, SPIE, 2015.

[26] DE CASTRO, A. I.; MAJA, J. M.; OWEN, J.; ROBBINS, J. ; PEÑA, J. M.. **Experimental approach to detect water stress in ornamental plants using sUAS-imagery**. In: Thomasson, J. A.; McKee, M. ; Moorhead, R. J., editors, AUTONOMOUS AIR AND GROUND SENSING SYSTEMS FOR AGRICULTURAL OPTIMIZATION AND PHENOTYPING III, volumen 10664, p. 178 – 188. International Society for Optics and Photonics, SPIE, 2018.

[27] ERDOS, D.; ERDOS, A. ; WATKINS, S. E.. **An experimental UAV system for search and rescue challenge**. IEEE Aerospace and Electronic Systems Magazine, 28(5):32–37, 2013.

[28] BELLVERT, J.; ZARCO-TEJADA, P. J.; GIRONA, J. ; FERERES, E.. **Mapping crop water stress index in a 'Pinot-noir'vineyard: comparing ground measurements with thermal remote sensing imagery from an unmanned aerial vehicle**. Precision agriculture, 15(4):361–376, 2014.

[29] HASSANALIAN, M.; ABDELKEFI, A.. **Classifications, applications, and design challenges of drones: A review**. Progress in Aerospace Sciences, 91:99 – 131, 2017.

[30] HILDMANN, H.; KOVACS, E.. **Review: Using Unmanned Aerial Vehicles (UAVs) as Mobile Sensing Platforms (MSPs) for Disaster Response, Civil Security and Public Safety**. Drones, 3(3), 2019.

[31] AIRBUS. **A380 - Passenger aircraft**. `https://www.airbus.com/aircraft/passenger-aircraft/a380.html`. [Accessed: 2020-02-01].

[32] AIRBUS. **Two airbus H135 Helicopters delivered to support space exploration at NASA's Kennedy Space Center**. `https://www.airbus.com/`. [Accessed: 2020-02-01].

[33] LINN AEROSPACE. **Albatross Fixed-Wing**. `https://linnaero.com/item/albatross/`. [Accessed: 2020-02-01].

[34] SWISSDRONES. **Blog & News**. `https://www.swissdrones.com/blog`. [Accessed: 2020-02-01].

[35] DJI. **Phantom 4 Pro**. `https://www.dji.com/phantom-4-pro`. [Accessed: 2020-02-01].

[36] DJI. **MG - 1P**. `https://www.dji.com/mg-1p`. [Accessed: 2020-02-01].

[37] DONG, W.; GU, G.-Y.; ZHU, X. ; DING, H.. **Modeling and Control of a Quadrotor UAV with Aerodynamic Concepts**. (77):437, 2013.

[38] BOUABDALLAH, S.. **Design and Control of Quadrotors with Application to Autonomous Flying**. PhD thesis, École Polytechnique Fédérale de Lausanne, 2007.

[39] POWERS, C.; MELLINGER, D. ; KUMAR, V.. **Quadrotor Kinematics and Dynamics**. In: HANDBOOK OF UNMANNED AERIAL VEHICLES, p. 307–328. Springer Netherlands, Aug. 2014.

[40] MAHONY, R.; KUMAR, V. ; CORKE, P.. **Multirotor Aerial Vehicles: Modeling, Estimation, and Control of Quadrotor**. IEEE Robotics Automation Magazine, 19(3):20–32, Sep. 2012.

[41] SHRAIM, H.; AWADA, A. ; YOUNESS, R.. **A Survey on Quadrotors: Configurations, Modeling and Identification, Control, Collision Avoidance, Fault Diagnosis and Tolerant Control**. IEEE Aerospace and Electronic Systems Magazine, 33(7):14–33, July 2018.

[42] IAN POUNDS, P. E.. **Design, Construction and Control of a Large Quadrotor Micro Air Vehicle**. PhD thesis, Australian National Univerisity, sep 2007.

[43] EMRAN, B. J.; NAJJARAN, H.. **A Review of Quadrotor: An Underactuated Mechanical System**. Annual Reviews in Control, 46:165 – 180, 2018.

[44] FAN, F.; LIN, M.; DING, R.; ZHENG, Z. ; LIU, Y.. **Augmented-MRAC for Quadrotor UAVs with Parameter Change**. In: 2016 IEEE INTERNATIONAL CONFERENCE ON ADVANCED INTELLIGENT MECHATRONICS, p. 501–506, July 2016.

[45] MERCADO, D.; CASTRO, R. ; LOZANO, R.. **Quadrotors Trajectory Tracking Control with Inaccurate GPS Measurements**. IFAC Proceedings Volumes, 46(30):142 – 146, 2013.

[46] EMRAN, B. J.; NAJJARAN, H.. **Switching Control of Quadrotor with Adaptation Mechanism**. In: 2016 IEEE INTERNATIONAL CONFERENCE ON SYSTEMS, MAN, AND CYBERNETICS, p. 004872–004877, Oct 2016.

[47] XIONG, J.-J.; ZHENG, E.-H.. **Position and Attitude Tracking Control for a Quadrotor UAV**. ISA Transactions, 53(3):725 – 731, 2014.

[49] YIN, H.; WANG, Q. ; SUN, C.. **Position and Attitude Tracking Control for a Quadrotor UAV via Double-loop Controller**. In: 2017 29TH CHINESE CONTROL AND DECISION CONFERENCE (CCDC), p. 5358–5363, May 2017.

[50] AKGÜN, O.; SUBAŞI, E. ; TÜRKER, T.. **A Lyapunov based Model Reference Adaptive Control of a Quadrotor**. In: 2017 10TH INTERNATIONAL CONFERENCE ON ELECTRICAL AND ELECTRONICS ENGINEERING, p. 732–736, Nov 2017.

[51] HERNANDEZ-MARTINEZ, E.; FERNANDEZ-ANAYA, G.; FERREIRA, E.; FLORES-GODOY, J. ; LOPEZ-GONZALEZ, A.. **Trajectory Tracking of a Quadcopter UAV with Optimal Translational Control**. IFAC-PapersOnLine, 48(19):226 – 231, 2015. 11th IFAC Symposium on Robot Control SYROCO 2015.

[52] YANG, J.; CAI, Z.; LIN, Q. ; WANG, Y.. **Self-tuning pid control design for quadrotor uav based on adaptive pole placement control**. In: 2013 CHINESE AUTOMATION CONGRESS, p. 233–237, 2013.

[53] POUNDS, P.; MAHONY, R. ; CORKE, P.. **Modelling and Control of a Large Quadrotor Robot**. Control Engineering Practice, 18(7):691 – 699, 2010.

[54] SLOTINE, J. .; DI BENEDETTO, M. D.. **Hamiltonian adaptive control of spacecraft**. IEEE Transactions on Automatic Control, 35(7):848–852, 1990.

[56] FAESSLER, M.; FALANGA, D. ; SCARAMUZZA, D.. **Thrust Mixing, Saturation, and Body-Rate Control for Accurate Aggressive Quadrotor Flight**. IEEE Robotics and Automation Letters, 2(2):476–482, 2017.

[57] ZHANG, X.; LI, X.; WANG, K. ; LU, Y.. **A Survey of Modelling and Identification of Quadrotor Robot**. Hindawi Publishing Corporation, 2(2):476–482, 2014.

[58] KURAK, S.; HODZIC, M.. **Control and estimation of a quadcopter dynamical model**. Periodicals of Engineering and Natural Sciences, 6(1):63–75, 2018.

[59] MALO TAMAYO, A. J.; VILLASEÑOR RÍOS, C. A.; IBARRA ZANNATHA, J. M. ; OROZCO SOTO, S. M.. **Quadrotor input-output linearization and cascade control**. IFAC-PapersOnLine, 51(13):437–442, 2018. 2nd IFAC Conference on Modelling, Identification and Control of Nonlinear Systems MICNON 2018.

[60] CORKE, P.. **Robotics, Vision and Control: Fundamental Algorithms in MATLAB**. Springer Publishing Company, Incorporated, 1st edition, 2013.

[61] YALI, Y.; YUANXI, W. ; OTHERS. **Controller Design of Quadrotor Aerial Robot**. Physics Procedia, 33:1254–1260, 2012.

[62] WHITEHEAD, B.; BIENIAWSKI, S.. **Model Reference Adaptive Control of a Quadrotor UAV**.

[63] MIHAILESCU-STOICA, D.; ACUNA, R. ; ADAMY, J.. **High performance adaptive attitude control of a quadrotor**. In: 2019 18TH EUROPEAN CONTROL CONFERENCE (ECC), p. 3462–3469, 2019.

[64] DYDEK, Z. T.; ANNASWAMY, A. M. ; LAVRETSKY, E.. **Adaptive control of quadrotor uavs: A design trade study with flight evaluations**. IEEE Transactions on Control Systems Technology, 21(4):1400–1406, 2013.

[65] AZINHEIRA, J. R.; RIVES, P.. **Image-Based Visual Servoing for Vanishing Features and Ground Lines Tracking: Application to a UAV Automatic Landing**. International Journal of Optomechatronics, 2(3):275–295, 2008.

[66] GOMEZ-BALDERAS, J.-E.; CASTILLO, P.; GUERRERO, J. A. ; LOZANO, R.. **Vision Based Tracking for a Quadrotor using Vanishing Points**. Journal of Intelligent & Robotic Systems, 65(1):361–371, Jan 2012.

[74] YANG, L.; QI, J.; SONG, D.; XIAO, J.; HAN, J. ; XIA, Y.. **Survey of Robot 3D Path Planning Algorithms**. Journal of Control Science and Engineering, 2016:5, 2016.

[78] LU, Y.; XUE, Z.; XIA, G.-S. ; ZHANG, L.. **A survey on vision-based uav navigation**. Geo-spatial Information Science, 21(1):21–32, 2018.

[79] CHATTERJEE, A.; RAKSHIT, A. ; SINGH, N. N.. **Mobile Robot Navigation**. In: VISION BASED AUTONOMOUS ROBOT NAVIGATION: ALGORITHMS AND IMPLEMENTATIONS, p. 1–14. Springer-Verlag Berlin Heidelberg, 2013.

[80] CANTONI, V.; LOMBARDI, L.; PORTA, M. ; SICARD, N.. **Vanishing point detection: representation analysis and new approaches**. In: PROCEEDINGS 11TH INTERNATIONAL CONFERENCE ON IMAGE ANALYSIS AND PROCESSING, p. 90–94, 2001.

[81] BASSO, M.; PIGNATON DE FREITAS, E.. **A UAV Guidance System Using Crop Row Detection and Line Follower Algorithms**. In: PROCEEDINGS 11TH INTERNATIONAL CONFERENCE ON IMAGE ANALYSIS AND PROCESSING, p. 605–621, 2020.

[82] GÉE, C.; BOSSU, J.; JONES, G. ; TRUCHETET, F.. **Crop/weed discrimination in perspective agronomic images**. Computers and Electronics in Agriculture, 60(1):49 – 59, 2008.

[85] OPENCV. **Opencv documentation**. `https://docs.opencv.org/3.4/index.html`. Accessed: 2019-05-20.

[86] TRIPATHI, A. K.; PATEL, V. V. ; PADHI, R.. **Vision based automatic landing with runway identification and tracking**. In: 2018 15TH INTERNATIONAL CONFERENCE ON CONTROL, AUTOMATION, ROBOTICS AND VISION (ICARCV), p. 1442–1447, 2018.

[87] OGATA, K.. **Modern control engineering**. Prentice Hall Upper Saddle River, NJ, 2009.

[88] FESTO. **FESTO**. `https://www.festo.com/group/en/cms/index.htm`. [Accessed: 2020-02-01].

A Quadrotor mechanical characteristics

Symbol	Unit	Value	Description
b	Ns^2	1.3858×10^{-6}	Drag torque coefficient
k	Nms^2	1.3328×10^{-5}	Thrust force coefficient
g	ms^{-2}	9.81	Gravity force
l	m	0.56	Arm of each motor
m	Kg	1.4	Vehicle's mass
I_{xx}	$Kg \cdot m^2$	0.05	Moment of inertia about $x-$axis
I_{yy}	$Kg \cdot m^2$	0.05	Moment of inertia about $y-$axis
I_{zz}	$Kg \cdot m^2$	0.24	Moment of inertia about $z-$axis

Table A.1: Mechanical constants Parameters of quadrotor