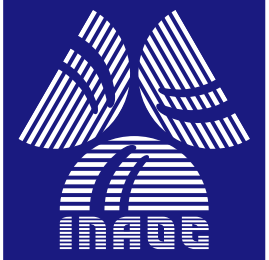


Quadrotor Flight in Constrained Indoor Environments

by

Antonio Matus Vargas

Technical Report CCC-17-005

Department of Computational Sciences

Instituto Nacional de Astrofísica, Óptica y Electrónica
Tonantzintla, Puebla, Mexico
November 15, 2017

Supervisors:

Dr. Gustavo Rodríguez Gómez
Dr. José Martínez Carranza

Abstract

Drones are being used in an increasing number of civilian applications. The quadcopter is a type of drone which has undergone extensive research making it an excellent testbed for novel control techniques. Several intended uses of quadcopters require operation in confined environments, in which objects are in close proximity to the vehicle. In these conditions, the flight is affected by aerodynamic interactions (force and torque). Intuitively, these interactions can be viewed as air flow rebounding on the surroundings back to the vehicle. The development of efficient computational approaches for describing such interactions remains open to improvement as existing accurate models require a considerable amount of computational load and cannot be used in the real-time control loop of the quadrotor. This research hypothesizes that with a simplified mathematical model, which can be deployed in real-time and approximates the behavior of the aerodynamic interactions, the flight control of the quadrotor is improved. In seeking to confirm this hypothesis, the aim is to develop an efficient model of aerodynamic interactions which can be retrieved from simulated and experimental data. Three major areas of knowledge shall be explored to solve the problem: control theory, artificial intelligence, and fluid mechanics. As a preliminary progress, numerical optimization techniques for nonlinear quadrotor control are proposed.

Contents

Chapter 1

Introduction

Drones or unmanned aerial vehicles (UAVs), aircraft systems which do not carry human operators, have the potential of changing the world in a positive way. They were conceived to be used in environments that are considered to be dangerous, dull or dirty; in other words, for situations where it is undesirable to employ a human pilot [1]. UAVs are equipped with on-board computing and sensing to enable their control and can be operated remotely and/or with certain degrees of autonomy. Although drones were predominantly deployed for military missions, they are being used in an increasing number of civilian appliances such as surveillance, inspection, goods delivery, photography, cinematography, and emergency response, just to mention a few.

The quadcopter or quadrotor is a four rotor helicopter. Each motor with a fixed-pitch propeller is mounted over a symmetrical cross-like frame. The movement of the quadrotor is controlled by the throttle in each motor. They are mechanically simple and low-cost. There has been extensive research about the quadrotor making it an excellent testbed for novel control techniques.

Several intended uses of small UAVs (included quadrotors), commonly referred to as micro air vehicles (MAVs), require operation in confined environments, in which objects are in close proximity to the drone. During the flight in this spaces, there will be aerodynamic vehicle-environment interactions which will affect the movement of the aircraft. Intuitively, these interactions can be viewed as airflow rebounding on the surroundings back to the vehicle. Even when drones are being tele-operated in line-of-sight those effects are difficult to handle.

1.1 Motivation

The flight of quadcopters in indoor scenarios is a challenging endeavor. Most research on UAV control has concentrated in handling dynamics when operating in free space, where aerodynamic interactions with nearby surfaces can be neglected. Although aerodynamic effects are mentioned in some works, most of them treat these interactions as random disturbances which the control must compensate using feedback. One of the most mentioned

aerodynamic actions is the ground effect, which is the upward lift caused by compressed air beneath the vehicle. Nevertheless, this "air bed" is not the only interaction that may occur. Another example is the ceiling effect, in which the drone is sucked upwards to the ceiling. Vertical surfaces also exert perturbations when a rotorcraft flies near them.

In robotics, physical interactions with the environment are essential, but not straightforward to model. In some cases they can be simulated, for the UAV case however, airflow simulations are computationally expensive to be implemented in a real-time control loop. Interactions can be learned from experience, but this is time-consuming. Particularly, the study of the ground effect on dates back to the 1930's. It started with the investigation of such effect on full-size helicopters [49, 50]. Mathematical approximations for a single rotor have been proposed [3, 52]. More recent works have applied flat surface approximations for helicopter and quadcopter UAVs. On the other hand, little research has taken into account the ceiling and wall effects.

There are problems yet to be solved to fully enable indoors flight of a quadrotor although extensive research has been carried in this area. Interior scenarios present the latent complication of object proximity to the quadrotor. This problem could be comfortably ignored for a small vehicle mass and using protective hulls, but by doing so, the benefits of controlling the vehicle near objects would also be ignored. There is a knowledge gap when it comes to the modeling of aerodynamic interactions between the quadcopter and its surroundings for the improvement of a control scheme. The encouraging results by Bartholomew et al. [5] are the main motivation for this research, they provided an upward acceleration prediction scheme which used the shape of the object below as input. However, a more general, platform-independent, and time-compact vehicle-environment interaction model needs to be further investigated.

1.2 Justification

Autonomous flight of UAVs is still a challenge in the robotics community. It is becoming harder as the promising tasks get more involved. For example, outdoors autonomous flight of an aircraft can be achieved using a global positioning system (GPS). However, many interesting applications require indoors flight. In such cases, satellites' signals may be degraded or even not receivable at all. Another problem is the precision of the GPS signal, it may not be suitable for navigating small buildings since a deviated measurement could lead to drone collisions with adjacent structures. Considering a very accurate and precise GPS with reliable

signal indoors, it would anyway be prone to cause undesirable contacts with GPS-invisible obstacles inside the construction. Here is where vision systems come into action. An on-board eye is being used by researchers to determine the relative movement of the MAV with respect to the environment. Also, it can help to detect obstacles and then retrieve traversable spaces or to obtain the shape of the surroundings. Other types of sensors can be used to indirectly measure other relevant flight variables, such as altitude. Nevertheless, sensors have a drawback: random inherent noise. Scientists have proposed methods for data fusion, they can be applied to mix two or more sensor measurements to obtain an overall result with reduced uncertainty. Fused data can be then consumed to plan a path. Control techniques stabilize the MAV and execute the path. Finally, once the drone has concluded the task, it is typically required to land safely. In summary, there are many tools for trying to achieve autonomous indoor flight, but they need to be proficiently unified for a given platform.

Although the aforementioned considerations must be taken into account, the focus will be on the investigation of algorithms to enhance the quadrotor flight in near-surface conditions. Such algorithms are expected to improve well-known flight missions like hovering, picking-up and releasing object, taking-off and landing in hard conditions, and operating near the ground in general. Also, the study of the flow physics during proximity flight could aid to the design/selection of control schemes. In this part, being able to model aerodynamic interactions has multiple potential appliances for quadrotor UAVs. At the path planning stage, predictions can be used to select paths that are less likely to suffer disturbances or to supply safety flight margins. A path planner could also make use of predictions to deliberately select paths that provide some impulse, thus reducing energy consumption. At the control level, the aerodynamic model could be incorporated into the control loop, allowing to track trajectories more accurately. Drones operated with control remote could also take the benefits of such model as a stabilizing assistance when piloting near objects.

1.3 Problem Statement

Micro quadcopters are able to access confined indoor environments such as small windows and doors, ducts, narrow corridors, and collapsed spaces. Naturally, operation in aforesaid confined environments will require quadcopters to fly in close proximity to fixed obstacles. Here, the adjective “indoor” has two connotations: no GPS signal, and negligible environmental wind.

In operation, the quadcopter is wanted to move safely from a starting point to a goal point.

The first step could be to control the quadrotor in free space, i.e., this stage would consist on how to effectively change the state of the vehicle in a three-dimensional Euclidean space with no objects. The following step could consist on controlling the quadcopter in presence of a loosely constrained environment, in which a sufficiently large distance between the vehicle and every object is guaranteed. The restraint is herein put on the states that the vehicle is able to visit, no further adjustments are required for the controller. Now, consider a confined setup, a particular phenomenon is observed: the vehicle is affected by aerodynamic disturbances due to fluid interactions that occur between rotors and nearby surfaces (the exterior boundary of objects). Though the previously designed control could compensate disturbances, it is prone to cause a critical failure due to collisions. A modification of the control scheme is demanded.

The treatment of aerodynamic interactions as random disturbances limits the control performance. A more detailed model of said interactions can leverage their suppression, and therefore improve the control response. The problem is now how to encode the aerodynamic disturbances in a feasible model that can be used in the real-time control loop for the quadrotor. It is assumed that the aircraft is equipped with on-board computing and sensing.

1.4 Research Questions

The stated problem pose the following inquiries.

1. In which way could the expensive airflow simulation be dispensed with for the retrieval of the force and torque acting on the quadrotor when flying near a surface?

A precise method for retrieving forces and torques for specific flight conditions is to perform a finite volume fluid simulation, such method does take significant computational power. The previous question alludes to a form of retrieving forces and torques disturbing the aircraft when flying in close proximity to a surface with limited computation capabilities.

2. Which set of variables would allow approximately model aerodynamic interactions between the quadrotor and its environment?

For instance, quadrotor's blades rotational speed, the flatness of the environment, normal directions of the surface, vehicle-surface distance, attitude of the aircraft, forward velocity (if any). Here, the aim is to try to build up a model that can be computed as close as possible to real-time.

3. Could the inclusion of a real-time simplified model of the aerodynamic interactions in the control loop outperform a cutting edge controller in terms of trajectory tracking?

The unraveling of this question will use the information gathered from the previous queries. It is important to note that the enhanced control loop may be based on a well-known control methodology (e.g., adaptive control), so a fair comparison shall be made against another control strategy that already beats the base control.

1.5 Hypothesis

It is hypothesized that with a simplified mathematical model, which can be deployed for computations in real-time and approximates the dynamics of the interactions between the vehicle and its environment (aerodynamic interactions), the flight control of the quadrotor is improved.

A more concise supposition is that the aerodynamic interactions experienced by the quadrotor due to nearby surfaces can be described by a certain set of internal quadrotor variables (rotational speed of propellers, attitude, drone-surface distance) and a set of the environmental characteristics mainly related with its geometry. Furthermore, the model would simplify the description of the behavior of a spatially distributed system (i.e., aerodynamic interactions) into a topology of discrete entities that would approximate the distributed scheme under certain assumptions.

Formally, let Q be a set of internal quadrotor variables, and E a set of environment characteristics, then the solution of a model $\dot{y} = a : Q \times E \rightarrow \mathbb{R}^n$ could be used to retrieve the vectors $d, \tau_d \in \mathbb{R}^3$ representing force and torque caused by aerodynamic interactions. In addition, another function $b : \mathbb{R}^6 \rightarrow \mathbb{R}^4$ would be required to implement the correction terms in the controller for the four motors.

1.6 Objectives

The general aim of the research is to design and implement a novel algorithm that is capable of stabilizing a quadrotor when flying close to planar surfaces. The new algorithm will be divided into two parts: one part is devoted to approximately capture the fluid interactions between the drone and nearby surfaces, and the other will consume this information for the final end of flight stabilization.

Specific goals are:

- To analyze the mathematical relations between the quadrotor and environment variables and the aerodynamic interactions (force and torque) using indirect observations of the phenomenon.
- To find the relevant dynamics of the model for simplifying the computational complexity with multi-rate integration methods [76].
- To establish a computational model of the aerodynamic interactions that can be executed using on-board capabilities.
- To design and to implement a flight control scheme that takes advantage of the retrieved aerodynamic interactions model.
- To conduct concurrent validation of the aerodynamic interactions model (comparing the retrieved model with an well-established model) and nomological validation of the proposed system.

1.7 Scope and Limitations

This research focuses on the development of algorithms for stabilizing the quadrotor flight near planar surfaces. Although the thesis requires knowledge related to instrumentation, dynamical systems, control theory, and optimization techniques, the target will be on the development of a computational model of aerodynamic interactions between the aircraft and its environment whose results can be used by other custom algorithm to enhance the flight tracking accuracy. It is out of the scope to search for a simplified analytic solution of the Navier-Stokes equations.

The following restrictions are set. The aerodynamic interactions will be studied in the moments when the aircraft is disturbed from equilibrium conditions (i.e., constant velocity), for which the vehicle is most susceptible to disturbances. The dynamical model shall have a low computational complexity to allow the consumption of its results in the control loop, the maximum computational time will be staged as a multiple of the period of the fastest sensor available. A trade-off is expected between the computational complexity and the accuracy of the model, but results shall be accurate enough to actually improve the flight control near surfaces. The model will be valid for a given drone-environment separation distance, and for operation in a predefined range of temperature, altitude, and humidity.

1.8 Expected Contributions

The main expected contributions of this research are summarized in the following bullets:

- A novel algorithm that allows stabilizing the quadrotor in near planar surface flight.
- A numerical strategy to solve in real-time the dynamic model of the quadcopter.
- The physical and mathematical relation between navigation variables, surface characteristics and the exerted force and torque on the aircraft.
- A new feasible computational model for relating flight information and aerodynamic interactions between the quadrotor and surfaces.
- Exploitation of the new computational model for improving the quadrotor trajectory tracking in near-surface conditions.

Chapter 2

Background

This segment provides a general overview of the concepts needed to understand the following sections.

2.1 Multirotor UAVs

There exist a wide variety of UAVs sizes, shapes, and configurations. They have been classified using different nomenclature, but they can be intuitively classified fixed-, rotary-, flapping-wing, and hybrid, which can be in-mission transitioned between classes (e.g., rotary-wing to fixed-wing). The last three classes are well suited to flight missions in confined spaces; however, the simpler mechanical design and the well established free-space models rotary-wing UAVs are the main reasons to choose this class. Initially, rotorcrafts can be sub-classified according to the number or rotors [8]. Furthermore, other configurations have been formed by stacking, fixing at some tilt angle, and dynamically tilting the rotors. An UAV classification diagram is shown in Fig. 2.1.

The fixed-pitch quadcopter provides features that favor flight in confined spaces. For the indoors flight, the size, flight time, and costs are the main points to take into account. Helicopters are energy efficient because the design allows the use of large propellers; however, their tilting mechanisms make them pricey to maintain. More than four rotors can increase safety owing to the fact that the vehicle may be able to maintain or recover flight if one propeller is lost. On the other hand, rotor redundancy implies an increase of the overall vehicle size, more energy consumption, and more maintenance costs. With respect to tilted configurations, they have been used to passively move in canted channels or to resist any wrench for aerial manipulation. The ability to exert horizontal forces for this canted configurations comes with a decrease in total thrust. Dynamical rotor tilting mechanism has been proposed to enhance the controllability properties of quadrotors; sadly, tilting mechanisms complicates the mechanical design and construction. In summary, the quadrotor is not the most efficient platform, but they provide a compact footprint with a low overall cost.

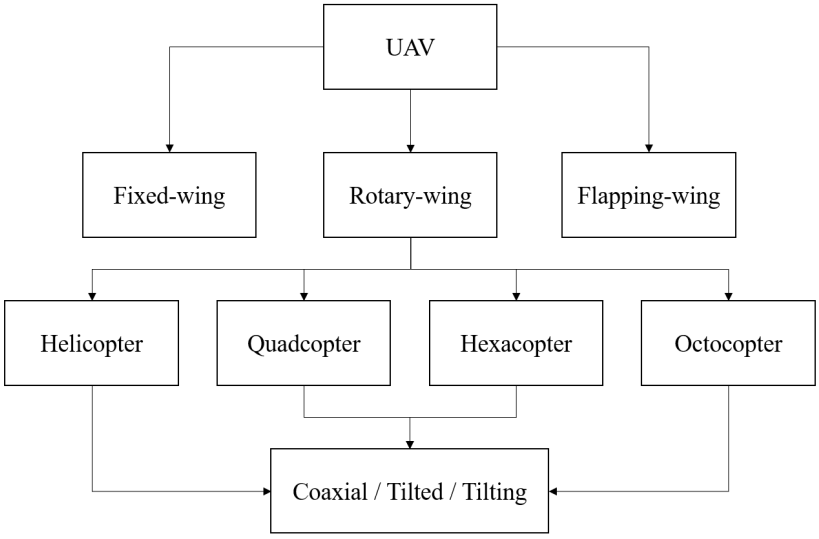


Figure 2.1. UAV classification, based on [8].

2.2 Quadrotor Anatomy

The major components of a quadrotor are described in the following paragraphs. In Fig. 2.2 the main components of a typical quadcopter are indicated.

The frame is a cross-like structure to which all the other essential components are attached, it is lightweight, rigid, and can be bought off the shelf or it could be designed to any specification. Motor mounts are located at the end of each arm and the electronics are housed in a flat area at the center of the cross.

The quadcopter uses four brush-less motors to power the propellers. Each motor is independently controlled by an electronic speed controller (ESC), which takes input from a logic board and sends varying pulses to the motor to make it spin at different rates. The propellers are mechanically attached to the motors and are used to provide lift and thrust for maneuvering. Motors and propellers are configured in such a way that two of them spin clockwise and the other two spin counter-clockwise. This arrangement produces a balance of forces and moments, giving some stability to the quadrotor.

A major drawback of quadrotors is their high energy consumption. They are normally suited with lightweight rechargeable batteries that have the capacity to power all the components during the desired flight endurance period. Most manufacturers use lithium-ion polymer (LiPo/LiPoly) batteries.

Quadrotors are equipped with flight sensors, including accelerometers, gyroscopes, barom-

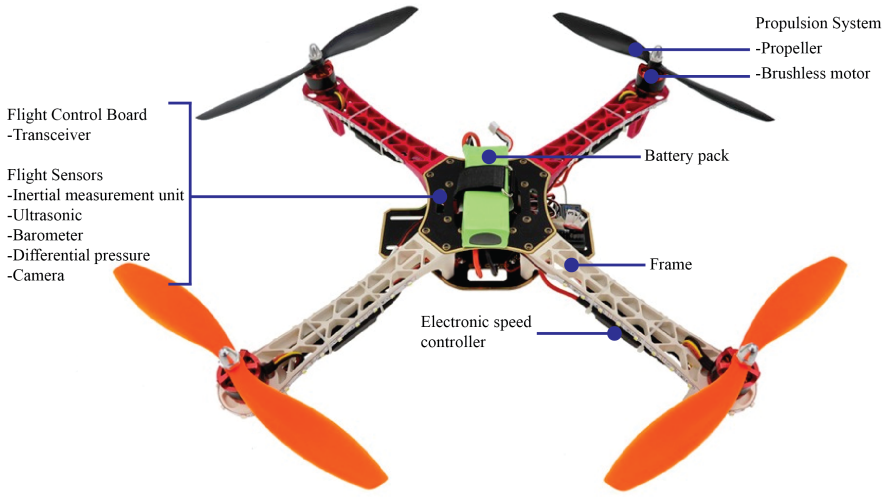


Figure 2.2. Main components of a quadcopter.

eters, ultrasonic transducers, GPS receivers, cameras, etc. All of these sensors are continually sending information to an on-board flight computer. These signals are then processed to keep the helicopter aloft.

An accelerometer is an electromechanical device that measures the physical acceleration experienced by the object to which it is attached. With this sensor, the vehicle angle of tilt can be determined by measuring the amount of static pull due to gravity acceleration. Furthermore, analysis of the movement of the device can be done by measuring the amount of dynamic acceleration.

A gyroscope is a device for measuring the orientation of the vehicle based on the principle of conservation of angular momentum. It measures angular velocity and is typically used for retrieving the vehicle's rate of orientation. The heading and motion of a craft can be tracked using an arrangement of three accelerometers and two gyroscopes, with their axes lined up at right angles. These measurements are obtained using a single device called inertial measurement unit (IMU) which is an electronic device that allows sensing a body's specific force and angular rate.

Ultrasonic transducers are designed to use sound propagation to detect obstacles in the robot's environment; they create a pulse of sound and then listen for reflections of the pulse. The distance from the transducer to an object is obtained by measuring the broadcast-reception time of a pulse and knowing the speed of sound. Another common element is the barometric pressure transducer, which generates an electrical signal as a function of the pressure

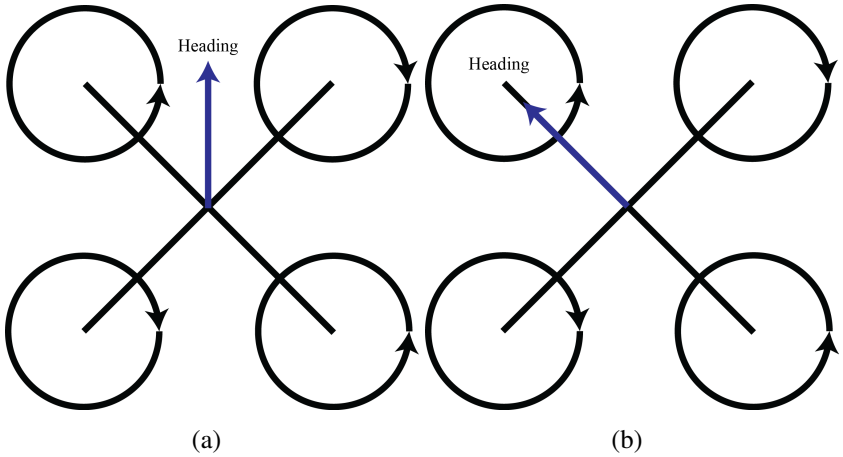


Figure 2.3. Common quadrotor configurations (top view), a) "x" style, b) "+" style.

imposed. This kind of sensor is often used for monitoring and control altitude.

The flight control board is the brain of the vehicle. It houses the sensors and processes the data. The only thing that keeps the quadrotor aloft is the precise control of the motors. The flight control board can track the vehicle's position and orientation by monitoring how it flies. Transmitters and receivers are connected to the board, these enable the communication between the aircraft and the remote control or ground station. There is a wide range of receiver and transmitter combinations, however, the most widely used technology is radio-frequency.

2.3 Basic Concepts of the Quadcopter

Quadcopters are commonly flown and modeled in two configurations: the "x" (cross, square) configuration and the "+" (plus, diamond) configuration, which are shown in Fig. 2.3. Either way, two of the motors rotate counter-clockwise, while the other two rotate clockwise. This arrangement helps the vehicle to not spin on its vertical axis since the rotational inertia is canceled out, eliminating the need for a tail rotor which is used to stabilize the conventional helicopter.

The following lines make sense considering a coordinate system aligned with the positive x-axis aligned with the heading direction, the z-axis pointing against gravity, and the y-axis following the right-hand rule. Now, observing how the rotors are distributed, it is easy to note that when the pair of rotors that are spinning in one direction is faster than the other pair, the vehicle will spin on its vertical axis, subsequently leading to a change of heading

of the aircraft's direction of movement. Angular acceleration about the x-axis and y-axis can be caused without affecting the rotation with respect to the z-axis. Increasing thrust for some rotors and decreasing it for others will maintain the vertical axis stability and induce a net torque about the other two. Translational acceleration is achieved by maintaining an inclination angle at the front or side of the structure. This way, fixed rotor blades can be made to maneuver the quadcopter in all dimensions.

Hovering is the term used to describe when the rotorcraft maintains a constant position at a selected point in the air. At this point, the forces of lift and weight reach a state of equilibrium. To reach the hovering state, all the propellers must rotate at the same speed to generate collective lift force that offsets the weight of the quadcopter. In general, rotorcrats can climb or descend by altering the vertical balance of forces acting on them. This alteration can be performed by applying throttle, which is the action of increasing (or decreasing) all the propeller speeds by identical amounts. If the quadcopter is tilted then the provided thrust generates accelerations in both vertical and horizontal directions.

The quadrotor has six degrees of freedom, three translational and three rotational, represented in Fig. 2.4. Here, the term pose refers to the position and orientation of the quadcopter in a three-dimensional space. The position on the x and y axes shall be named as such, while the position on the z axis may be referred to as altitude. The orientation angles measured shall be named roll (ϕ), pitch (θ), and yaw (ψ), being the rotation angles about the x, y, and z axis, respectively. The yaw angle may also be referred to as heading.

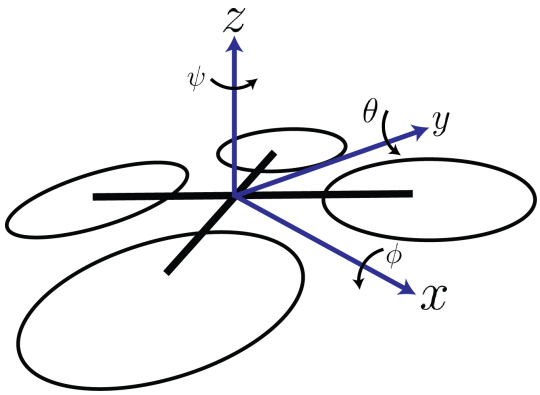


Figure 2.4. Degrees of freedom of a quadrotor.

2.4 Attitude Representations

The focus issue on this section is to give a summary of the different formalisms used to represent the attitude of a rigid body. A more detailed explanation is given in [6]. The pose of a rigid body is the position and attitude (orientation) of that body. The data for this case is expressed in two different coordinate systems:

- The world coordinate system is fixed to the earth; it is also common to said that is fixed to inertial space. The origin of this coordinate system is denoted by x_w.
- The body-fixed coordinate system is rigidly attached to the object of interest. The origin of this coordinate system is denoted by x_b.

Points and vectors are expressed in the body-fixed coordinates are distinguished from those expressed in the world coordinates by a prime symbol.

2.4.1 Rotation Matrix

A rotation matrix is a matrix whose multiplication with a vector rotates the vector while preserving its length. The special orthogonal group of all 3×3 rotation matrices is denoted by $SO(3)$. Thus, if $R = SO(3)$, then

$$\det R = \pm 1 \text{ and } R^{-1} = R^T.$$

Two possible conventions exist for defining the rotation matrix that encodes the attitude of a rigid body: a) the matrix that maps from the body-fixed to coordinates to the world coordinates, b) the matrix that maps from the world coordinates to the body-fixed coordinates. Though converting between the two conventions is as trivial as performing the transpose matrix, it is necessary to check that two different sources are using the same convention prior to use results from both sources together.

Here, the rotation matrix that encodes the attitude of a rigid body is defined to be the matrix that when pre-multiplied by a vector expressed in the world coordinates yields the same vector expressed in the body-fixed coordinates. Let $z \in \mathbb{R}^3$ be a vector in the world coordinates and $z' \in \mathbb{R}$ be the same vector expressed in the body-fixed coordinates, then the following relations hold:

$$z' = Rz,\ z = R^T z'.$$

To transform a point from one coordinate system to the other one must subtract the offset to the origin of the target coordinate system before applying the rotation matrix. Thus, if $x \in \mathbb{R}^3$ is a point in the world coordinates and $x' \in \mathbb{R}^3$ is the same point expressed in the body-fixed coordinates, then:

$$x' = R(x - x_b) = Rx + x'_w \tag{2.1}$$

$$x = R^T(x' - x'_w) = R^T x' + x_b. \tag{2.2}$$

Substituting $x = 0$ into Eq. 2.1 and $x' = 0$ into Eq. 2.2 yields

$$x'_w = -Rx_b,\ x_b = -R^T x'_w.$$

In the computer graphics community, it is quite common to write Eqs. 2.1 and 2.2 as matrix-vector products:

$$\begin{bmatrix} x' \\ 1 \end{bmatrix} = \begin{bmatrix} R & x'_w \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} x \\ 1 \end{bmatrix} = \begin{bmatrix} R & -Rx_b \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} x \\ 1 \end{bmatrix},$$

$$\begin{bmatrix} x \\ 1 \end{bmatrix} = \begin{bmatrix} R^T & x_b \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} x' \\ 1 \end{bmatrix} = \begin{bmatrix} R^T & -Rx'_w \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} x' \\ 1 \end{bmatrix}.$$

A coordinate rotation is a rotation about a single coordinate axis. Enumerating the x-, y-, and z-axes with 1, 2, and 3, the coordinate rotations, $R_i : \mathbb{R} \to SO(3)$, for $i \in \{1, 2, 3\}$, are

$$R_1(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & \sin\alpha \\ 0 & -\sin\alpha & \cos\alpha \end{bmatrix},\ R_2(\alpha) = \begin{bmatrix} \cos\alpha & 0 & -\sin\alpha \\ 0 & 1 & 0 \\ \sin\alpha & 0 & \cos\alpha \end{bmatrix},\ R_3(\alpha) = \begin{bmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

A rotation matrix may also be referred to as direction cosine matrix because its elements are the cosines of the unsigned angles between the body-fixed axes and the world axes. Denoting the world axes by (x, y, z) and the body-fixed axes (x', y', z'), let $\theta_{x',y}$ be, for example the unsigned angle between the x'-axis and the y-axis. In terms of these angles, the rotation

matrix may be written

$$R = \begin{bmatrix} \cos(\theta_{x',x}) & \cos(\theta_{x',y}) & \cos(\theta_{x',z}) \\ \cos(\theta_{y',x}) & \cos(\theta_{y',y}) & \cos(\theta_{y',z}) \\ \cos(\theta_{z',x}) & \cos(\theta_{z',y}) & \cos(\theta_{z',z}) \end{bmatrix}.$$

The rotation matrix may also be thought of as the matrix of basis vectors that define the world and body-fixed coordinate systems. The rows of the rotation matrix are the basis vectors of the body-fixed coordinates expressed in world coordinate, and the columns are the basis vectors of the world coordinates expressed in the body-fixed coordinates.

The multiplication of two rotation matrices yields another rotation matrix whose application to a point effects the same rotation as the sequential application of the two original rotation matrices. For example, let

$$z' = R_a z,\ z'' = R_{b/a} z' = R_{b/a} R_a z = R_b z,$$

where

$$R_b = R_{b/a} R_a.$$

Note that the rotations are applied in the reverse order.

2.4.2 Euler Angles

Leonhard Euler introduced that three coordinate rotations in sequence can describe any rotation. Consider triple rotations in which the first rotation is an angle ψ about the k-axis, the second rotation is an angle θ about the j-axis, and the third rotation is an angle ϕ about the i-axis. For notational brevity, let arrange these angles in a three-dimensional vector called the Euler angle vector, defined by

$$u := [\phi,\ \theta,\ \psi]^T.$$

The function that maps an Euler vector to its corresponding rotation matrix $R_{ijk} : \mathbb{R}^3 \to SO(3)$, is

$$R_{ijk}(\phi, \theta, \psi) := R_i(\phi) R_j(\theta) R_k(\psi). \tag{2.3}$$

The time derivative of the Euler angle vector is the vector of Euler angle rates. The relationship of the Euler angle rates and the angular velocity of the body is encoded in the Euler

angle rates matrix. Multiplying this matrix by the vector of Euler angle rates gives the angular velocity in the global coordinates. Letting $\hat{e}_i$ be the i^{th} unit vector, the function that maps an Euler angle vector to its corresponding Euler angle rates matrix, $E : \mathbb{R}^3 \to \mathbb{R}^{3\times3}$, is

$$E_{ijk}(\phi, \theta, \psi) := [R_k(\psi)^T R_j(\theta)^T \hat{e}_i, R_k(\psi)^T \hat{e}_j, \hat{e}_k], \tag{2.4}$$

and the related conjugate Euler angle rates matrix function, $E' : \mathbb{R}^3 \to \mathbb{R}^{3\times3}$, whose multiplication with the vector of Euler angle rates yields the body-fixed angular velocity is

$$E'_{ijk}(\phi, \theta, \psi) := [\hat{e}_i, R_i(\phi)\hat{e}_j, R_i(\phi)R_j(\theta)\hat{e}k]. \tag{2.5}$$

Hence,

$$\omega = E_{ijk}(u)\dot{u},\ \omega' = E'_{ijk}(u)\dot{u}.$$

Noting also that the angular velocity in the body-fixed coordinates may be related to the angular velocity in the global coordinates by

$$\omega' = R_{ijk}(u)\omega,\ \omega = R_{ijk}(u)^T\omega'.$$

Angular velocities, ω and ω', and $\dot{u}$ can be eliminated to yield

$$R_{ijk}(u) = E'_{ijk}(u)[E_{ijk}(u)]^{-1},\ R_{ijk}(u)^T = E_{ijk}(u)[E'_{ijk}(u)]^{-1}.$$

Thus far, the sequences of coordinate rotations that are able to span all dimensional rotations have not been specified. In fact, only 12 of the 27 possible sequences of three integers in $\{1, 2, 3\}$ satisfy the constraint that no two consecutive numbers in a valid sequence may be equal. These are

$$\begin{aligned}(i, j, k) \in \{&(1, 2, 1), (1, 2, 3), (1, 3, 1), (1, 3, 2),\\ &(2, 1, 2), (2, 1, 3), (2, 3, 1), (2, 3, 2),\\ &(3, 1, 2), (3, 1, 3), (3, 2, 1), (3, 2, 3)\}.\end{aligned}$$

Euler angle representations have singularities that are said to arise from gimbal lock. Gimbal lock may be understood in different ways. Intuitively, it arises from the indistinguishability of changes in the first and third Euler angles when the second Euler angle is at some critical value. The phenomenon may also be seen in the mathematics, where it manifests

itself as singularities. For all parametrizations of the form (i, j, i) there exist a singularity at the home position, $[\phi, \theta, \psi] = [0, 0, 0]$. On the other hand, all the Euler angle sequences that do not have a repeated axis have singularities at pitch values of $\theta = \pi/2 + n\pi$, for $n \in \mathbb{Z}$. A common strategy for dealing with singularities is to change representations whenever an object nears a singularity. Even more popular is the use of unit quaternions to represent an object's attitude.

2.4.3 Quaternions

William Rowan Hamilton first devised quaternions in the 19th-century. A quaternion, $q \in \mathbb{H}$, may be represented as a vector,

$$q = [q_0, q_1, q_2, q_3]^T = \begin{bmatrix} q_0 \\ q_{1:3} \end{bmatrix}.$$

The adjoint, norm, and inverse of the quaternion, q, are

$$\bar{q} = \begin{bmatrix} q_0 \\ -q_{1:3} \end{bmatrix}, \ \|q\| = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2}, \ q^{-1} = \frac{\bar{q}}{\|q\|}.$$

Quaternion multiplication is not commutative. Multiplication between quaternion q and p is defined by

$$\begin{aligned} q \cdot p = q_m(q, p) &= \begin{bmatrix} q_0 p_0 - q_{1:3}^T p_{1:3} \\ q_0 p_{1:3} + p_0 q_{1:3} - q_{1:3} \times p_{1:3} \end{bmatrix} \\ &= \begin{bmatrix} q_0 & -q_{1:3}^T \\ q_{1:3} & q_0 I_3 - C(q_{1:3}) \end{bmatrix} \begin{bmatrix} p_0 \\ p_{1:3} \end{bmatrix} = \begin{bmatrix} p_0 & -p_{1:3}^T \\ p_{1:3} & p_0 I_3 + C(p_{1:3}) \end{bmatrix} \begin{bmatrix} q_0 \\ q_{1:3} \end{bmatrix}, \end{aligned}$$

where the skew-symmetric cross product matrix function $C : \mathbb{R}^3 \to \mathbb{R}^{3\times 3}$ is defined by

$$C(x) = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ x_2 & x_1 & 0 \end{bmatrix}.$$

More compactly, quaternion multiplication may be written as the second quaternion pre-

multiplied by a matrix-valued function of the first quaternion. That is,

$$q \cdot p = q_m(q,p) = Q(q)p = \bar{Q}(p)q$$
$$p \cdot q = q_m(p,q) = Q(p)q = \bar{Q}(q)p,$$

where the quaternion matrix function, $Q : \mathbb{H} \to \mathbb{R}^{4\times4}$ is defined by

$$Q(q) = \begin{bmatrix} q_0 & -q_{1:3}^T \\ q_{1:3} & q_0 I_3 + C(q_{1:3}) \end{bmatrix} = \begin{bmatrix} q_0 & -q_1 & -q_2 & q_3 \\ q_1 & q_0 & q_3 & -q_2 \\ q_2 & -q_3 & q_0 & q_1 \\ q_3 & q_2 & q_1 & q_0 \end{bmatrix},$$

and the related conjugate quaternion matrix function, $\bar{Q} : \mathbb{H} \to \mathbb{R}^{4\times4}$ is defined by

$$\bar{Q}(q) = \begin{bmatrix} q_0 & -q_{1:3}^T \\ q_{1:3} & q_0 I_3 - C(q_{1:3}) \end{bmatrix} = \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{bmatrix}.$$

Unit quaternions are quaternions with unity norm, which means that

$$\|q\| = 1.$$

A unit quaternion can be used to represent the attitude of a rigid body. Recalling the vector $z \in \mathbb{R}^3$ in the global coordinates, and $z' \in \mathbb{R}^3$ the same vector in the body-fixed coordinates, then the following relations hold:

$$\begin{bmatrix} 0 \\ z' \end{bmatrix} = q \cdot \begin{bmatrix} 0 \\ z \end{bmatrix} \cdot q^{-1} = q \cdot \begin{bmatrix} 0 \\ z \end{bmatrix} \cdot \bar{q}$$
$$= \bar{Q}(q)^T Q(q) \begin{bmatrix} 0 \\ z \end{bmatrix} = \begin{bmatrix} 1 & 0^T \\ 0 & R_q(q) \end{bmatrix} \begin{bmatrix} 0 \\ z \end{bmatrix},$$

where

$$R_q(q) = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2q_1q_2 + 2q_0q_3 & 2q_1q_3 - 2q_0q_2 \\ 2q_1q_2 - 2q_0q_3 & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2q_2q_3 + 2q_0q_1 \\ 2q_1q_3 + 2q_0q_2 & 2q_2q_3 - 2q_0q_1 & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix}. \tag{2.6}$$

Inverse mappings $q_R^i : SO(3) \to \mathbb{H}$ for $i \in \{0, 1, 2, 3\}$, can be retrieved by inspection of Eq. (2.6), however, they are not displayed for the sake of brevity.

Just as with rotation matrices, sequences of rotations are represented by products of quaternions. That is, for unit quaternions q and p, it holds that

$$R_q(q \cdot p) = R_q(q)R_q(p).$$

The main disadvantage of unit quaternions is that they are constrained to have unit length. Furthermore, the unity norm constraint is quadratic in form and thus can lead to complications when attempting to optimize over the quaternion parameters.

Alternatively, it is possible to parameterize the attitude of a rigid body with an angle $\alpha \in \mathbb{R}$ and a unit vector $n \in \mathbb{S}^2$, where $\mathbb{S}^2 := \{v \in \mathbb{R}^3 \mid \|v\| = 1\}$. The quaternion that arises from a rotation α about an axis n is given by the axis-angle quaternion function, $q_a : \mathbb{R} \times \mathbb{S}^2 \to \mathbb{H}$, defined by

$$q_a(\alpha, n) := \begin{bmatrix} \cos(1/2\alpha) \\ n \sin(1/2\alpha) \end{bmatrix}.$$

This representation, while perhaps more intuitive than quaternion, is functionally equivalent to it: both require four parameters and a single quadratic constraint.

The inverse mappings, from unit quaternion to the corresponding axis and angle of rotation, are $\alpha_q : \mathbb{H} \to \mathbb{R}$ and $n_q : \mathbb{H} \to \mathbb{S}^2$, defined by

$$\alpha_q(q) := 2 \arccos(q_0),\ n_q(q) := \frac{q_{1:3}}{\|q_{1:3}\|} = \frac{q_{1:3}}{\sqrt{1 - q_0^2}}.$$

2.4.4 Rotation Vector

One of the major drawbacks of quaternions is that they require a quadratic norm constraint in order to be valid rotations. This problem can be overcome by folding the unity norm constraint into the parametrization. The rotation vector appears to be the most natural three-dimensional parametrization of the quaternion representation of an object's attitude.

The rotation vector is a function of the axis and angle of a rotation, $\mathrm{v}_a : \mathbb{R} \times \mathbb{S}^2 \to \mathbb{R}^3$, given by

$$\mathrm{v}_a(\alpha, n) := \alpha n.$$

Noting that $\|n\| = 1$, the previous definition may be inverted to yield the functions, α_v :

$\mathbb{R}^3 \to \mathbb{R}$ and $n_v : \mathbb{R}^3 \to \mathbb{S}^2$, defined by

$$\alpha_v(\mathrm{v}) := \|\mathrm{v}\|,\ n_v(\mathrm{v}) := \frac{\mathrm{v}}{\|\mathrm{v}\|}.$$

The function that maps a rotation vector to a unit quaternion, $q_v : \mathbb{R}^3 \to \mathbb{H}$, is defined by

$$q_v(\mathrm{v}) := q_a(\alpha_v(\mathrm{v}), n_v(\mathrm{v})) = \begin{bmatrix} \cos(\|\mathrm{v}\|/2) \\ \frac{\mathrm{v}}{\|\mathrm{v}\|}\sin(\|\mathrm{v}\|/2) \end{bmatrix}.$$

The inverse mapping, $\mathrm{v}_q : \mathbb{H} \to \mathbb{R}^3$, which maps a unit quaternion to a rotation vector, is given by

$$\mathrm{v}_q(q) := \alpha_q(q) n_q(q) = 2\arccos(q_0)\frac{q_{1:3}}{\|q_{1:3}\|} = \frac{2\arccos(q_0)}{(1-q_0^2)^{1/2}} q_{1:3}$$

The multiplication of two rotation vectors u and $v \in \mathbb{R}^3$ is defined in terms of the product of quaternions:

$$v * u = \mathrm{v}_m(v, u) = \mathrm{v}_q(q_m(q_v(v), q_v(u))).$$

Basically, the expression above indicates that the product computed by converting each rotation vector to a unit quaternion, performing the quaternion product, and then converting back to a rotation vector.

Now, consider an object with a body-fixed angular velocity of $\omega'(t)$ and the change in attitude from time t_0 to time t_1. The rotation vector over this interval is defined to be

$$\mathrm{v}_{\omega'}(t_0, t_1) := \int_{t_0}^{t_1} \omega'(t) d\tau.$$

If the body-fixed angular velocity is provided as discrete samples (from a set of gyros for example) the integration will have to be carried numerically. If at time t_0 the body has a quaternion attitude of q_0, then the attitude at time t_1 is

$$q_1 = q_v(\mathrm{v}_{\omega'}(t_0, t_1)) \cdot q_0.$$

This equation may be generalized to read

$$q_{i+1} = q_v(\mathrm{v}_{\omega'}(t_i, t_{i+1})) \cdot q_i,$$

which gives a simple update rule for tracking the attitude of an object over time. This method

is more accurate than integrating the Euler angles rates.

2.5 Control Systems Concepts

The autonomous flight of a quadrotor requires the implementation of a controller. This subsection shall present concepts in order to provide a better understanding of a control system implementation.

In modern control theory, the physical object to be controlled is referred to as the plant. A set of variables from the plant is measured and controlled, they are also named as the output of the system. The user feeds the system with a desired value/reference/set point. The task of the controller is to issue a control signal to compensate for the difference between the output and the set point value. In other words, the action of control means measuring the value of the controlled variables to correct or limit the deviation of the measured value from the desired value [7]. On the contrary, a disturbance is a signal that tends to adversely affect the value of the system output. If a disturbance is generated within the system it is called internal, while an external disturbance is caused outside the system and is an input.

Control systems can be classified in open-loop and closed-loop systems. The former only uses the user input to obtain the desired output, however, it cannot compensate for external disturbances. Closed-loop control systems operate with output feedback. A sensor provides the output measurement which is compared with the reference value in order to produce the desired value at the output. Therefore, the closed-loop system is able to compensate for external disturbances. General block diagrams for both aforementioned control systems are displayed in Fig. 2.5.

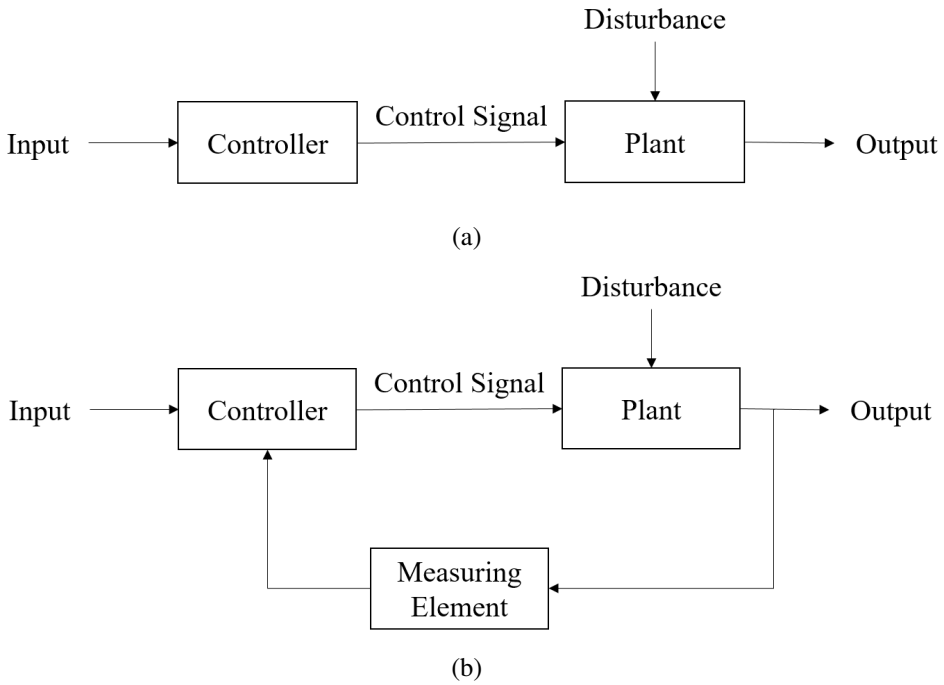


Figure 2.5. Block diagrams of a) the open-loop and b) the closed-loop control systems.

Chapter 3

State of the Art

In this section, a literature review for the main topics of the research is offered. A brief review of the control projects and relevant control techniques are provided in the starting subsection. Then, the review turns more specific for the revision of optimization techniques for control. Next, the most used methods for localization of the vehicle are discussed. The penultimate subsection is devoted to discussing the origins of the study of ground effect. The last subsection provides the explanation of the flight control attempts for unmanned quadrotors in proximity to surfaces.

3.1 Quadcopter Control Projects

UAVs have continued to capture the attention of many people. Research and development significance in this field is increasing due to the emergence of a large variety of potential civil applications. Several research groups are using the quadrotor design as a testbed for various projects. Table 3.1 is an summary of the compilation given in [8].

As seen before, different control methodologies have been formulated for quadrotors. A review of the prominent controller can be found in [9]. Control methodologies developed for unmanned quadcopters can be categorized into three classes [18], as shown in Fig. 3.1.

- Linear controllers
- Nonlinear controllers
- Intelligent controllers

Linear control techniques have been first choices for flight control of UAVs since their design and implementation is not cumbersome. Different simulation tools are available to analyze stability and performance. Furthermore, linear controllers have been successfully implemented and proved efficient in real world quadrotors. However, linear techniques are designed for linearized models and when quadcopters deviate from the operating regime, the performance of these controllers is badly affected.

Table 3.1. Research projects on quadrotors, loosely based on [8].

University/ Organization	Project	Year	Recent Studies
Stanford	Starmac I, Starmac II	2004-2011	Collision avoidance and control in aggressive maneuver with hybrid decomposition and reachable set theory [10].
U. Pennsylvania	E. Altug	2002-2012	Improving disturbance rejection and robustness using fuzzy logic controller [11].
UTC	P. Castillo et al.	2003-2014	Precise measurement and prediction of position and orientation in presence of external perturbation [12].
EPFL	Bouabdallah, Siegwart	2004-2011	Robust control in presence of uncertainties and external disturbances [13].
Cornell U.	A. Saxena et al.	2004-2012	Parallel algorithms for obstacle avoidance [14].
METU	F. B. Camlica, A. T. Kutay	2004-2014	Simulations of linear quadratic tracking control [15].
Cranfield U.	I. D. Cowling, J. F. Whidborne	2007-2010	Real time trajectory generation and tracking in presence of gust [16].
U. of Maryland	AVL's Micro Quad	2009-2015	Robust stabilization and command tracking behavior in obstacle-laden environments [17].

Various nonlinear control schemes have been established to overcome the shortcomings of linear methods applied to UAVs. Nonlinear control techniques are based on nonlinear dynamical models. Techniques such as feedback linearization, adaptive control, and MPC have been successfully applied for real quadrotors. They perform better than linear controls in terms of disturbance rejection, system uncertainties, and tracking precision. In spite of this, a significant implementation progress has not been witnessed. The reason is the higher computational cost for the implementation.

For intelligent controllers, dynamic model of the unmanned vehicle is not necessarily required, the system is trained via data obtained through flight experiments. This kind of con-

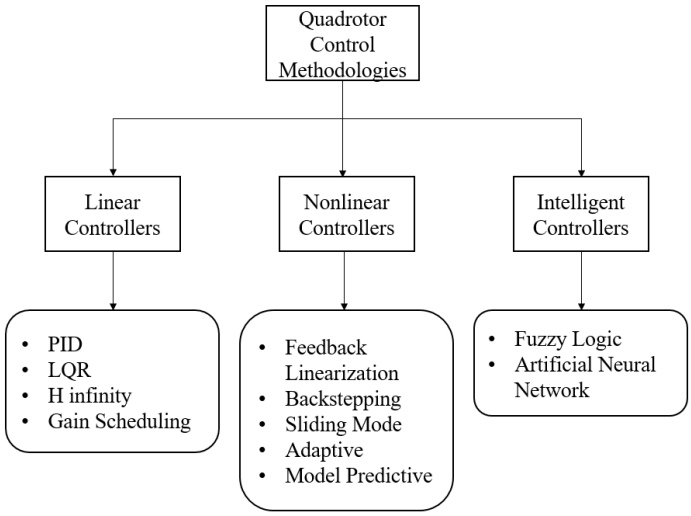


Figure 3.1. Classification of control methodologies for quadrotors, modified from [18].

trollers has been validated through several experiments. The model-free approach enables these controllers to be used on different UAV configurations. The inconvenient lies on the difficulty in the analysis of stability and robustness of these methodologies.

3.2 Optimization Techniques for Control

Conventionally, controllers have parameters that need to be set to certain values in order to achieve some desired response. A common strategy is to choose those values by trial and error. But, how does one make sure that the system behaves as good as possible? This question can be answered by introducing optimization theory. In fact, some of the control techniques shown in Fig. 2.5 already employ optimization to some extent (LQR, adaptive, model predictive). Since optimization is a very desirable feature, a brief review of its relationship with control, with a special focus on the quadcopter, is related here.

One of the most popular for industrial process and even for quadrotors is the PID control, a popular technique because it can be easily applied to a plant without knowing the dynamical model by adjusting its gains heuristically. Some authors have controlled a linearized model of a quadrotor using PID [19, 20]. The linearized system, however, cannot be used for rapid maneuvering. To overcome this issue, more complete models have been used applying cascaded PID controllers [21]. Within this context, cascaded means that there are an inner loop and an outer one, the former stabilizes the angular rates, and the latter takes care of Euler angles stabilization. PID controllers have, in general, the potential to perform very well but

their quality is, obviously, highly dependent on the gains parameters. The problem of how to tune the PID gains so it behaves optimally has been tackled using two main approaches: relying on the properties of the system, such as stability or linearity [22]; using extremum searching, i.e., mathematical optimization.

Researchers have been using optimization to tune PID controllers for quadrotors. Kim et al. [23] compared the steepest descent and Newton's method for tuning gain of a position-yaw PD control. Both methods converged to the same point, and authors noted that the computational load of the first algorithm was not significant but it required a large number of iterations to meet the halt condition, the other one converged in fewer iterations demanding higher computational load. A group at the University of Hangzhou proposed a PD and PID tuning approach for Euler angles and height control [24]. Their method was based on gradient optimization through a variational system. They provide comparative results for manually tuned and optimized PD and PID controllers. Heuristic, stochastic and combinations of these forms of optimization have also been applied to this problem [25, 26].

Variational calculus, a small area of the optimization field, has allowed the development of the optimal control theory. This theory provides a framework to determine control signals that will cause a process to maximize or minimize a performance criterion [27]. The linear quadratic regulator (LQR) is one kind of optimal control technique, in which the controller is given by a negative feedback of the system's states. This approach was used to obtain vertical position controllers [28]. Three techniques were compared, a PID tuned by integral time-weighted absolute error (ITAE), a classic LQR controllers, and a PID tuned by an LQR loop. It was shown that the ITAE-tuned PID gave the faster results but not with robust gains as the LQR-based ones. The LQR technique has been widely applied to different quadrotor dynamic models and to real-world experiments [29, 30, 31]. However, this approach works with the assumption of a linear system to be controlled, which may impose certain limitations.

Research has been carried out in which nonlinear models of quadcopters were adopted to obtain optimal transitions between states. In [32] a twelve-state model was used to generate a time-optimal trajectory between two states. Authors proposed a method combining genetic algorithms and nonlinear programming. It numerically minimizes the terminal transition time assuming piecewise constant control inputs. A group from the ETH Zürich proposed an algorithm that calculates the control inputs necessary to make the transition between a pair of known states in optimal time using the minimum principle [33, 34]. Though the method was applied to a simplified five-state quadrotor model it still remained nonlinear. In [35], a nonlinear model with six states was considered in order to compare direct and

indirect methods for dynamic optimization. The Legendre Pseudo-spectral (direct) method was compared against the Pontryagin's maximum (indirect) method. Both solutions were assumed to be optimal.

3.3 Vehicle Localization

One important thing to apply control is to measure. So, in order to implement a controller for the position of an aircraft, we require to have some measure of its position. The central theme of this section is how the position of the quadcopter is being procured nowadays.

Nowadays, the most common method to obtain the position of a drone in outside environments is the GPS. There are commercial MAVs with such technology. One example is the Parrot AR Drone 2.0 which can be controlled using GPS with the software Flight Recorder released in 2013. In 2015, the Minnesota Department of Transportation reported a bridge inspection system using a quadrotor, they involved GPS for navigation purposes [36].

Another system to determine the position of an MAV is the Motion Capture or Mo-cap. It consists of external cameras that detect the position using markers in the quadrotor. The GRASP laboratory of the University of Pennsylvania is highly involved with this kind of system. This group has accomplished impressive maneuvers, such as flying through windows with few inches clearance [37, 38].

Dead reckoning or inertial navigation is an algorithm for estimating position. For the case of a UAV, the location constantly estimated using data from the IMU, which consists of a three axis gyroscope, three axis accelerometer, and a three axis magnetometer. The dead reckoning algorithm is typically combined with GPS. When the GPS signal is weak the dead reckoning switches on to continue the position estimation using the last data of the GPS. The great disadvantage of this method is that error is accumulated over time due to sensors' noise and drift.

The estimation of the state of a robot with on-board sensors and the construction of a model of the environment is referred as Simultaneous Localization And Mapping (SLAM). Monocular cameras are being used for SLAM. Although this type of cameras natively provides 2D image, a 3D model can be retrieved using a sequence of images. Several algorithms for monocular SLAM have been developed. Parallel Tracking And Mapping (PTAM) [39], ORB-SLAM [40], and Large-Scale Direct monocular SLAM (LSD-SLAM) [41] are some examples of this kind of algorithms. A slightly different algorithm is the Semi-direct monocular Visual Odometry (SVO) [42], the difference lies in that this algorithm uses existing maps

and does not generate them.

Several works have proposed the use of a camera and an IMU to get what is called the visual-inertial navigation. The motivation is that the minimal sensor suite for autonomous localization consists of the use of those two sensors [43, 44, 45, 46]. Similar to this, a quadcopter was equipped with a stereo camera and IMU [47]. The use of the stereo camera facilitates the handling of the scaling problems that arise with monocular camera systems.

3.4 Ground Effect on Copters

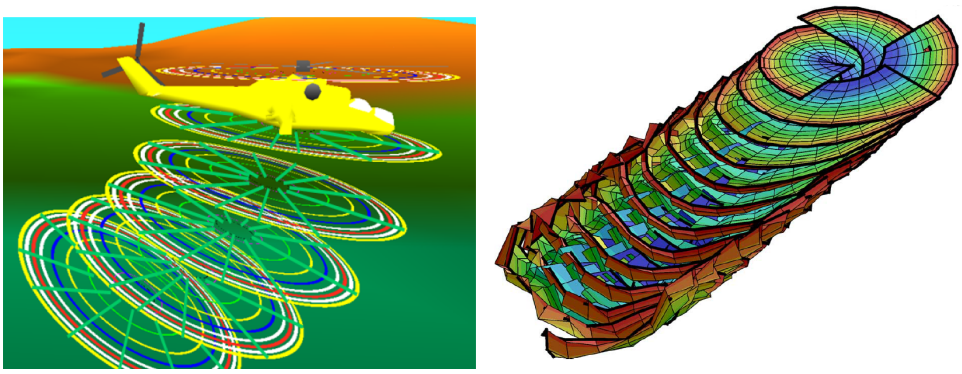
Proximity to the ground has a pronounced effect on the aerodynamic characteristics of rotary-wing crafts. Ground effect is therefore of importance in the study of helicopters. This problem has been mentioned in the 1930's literature [48, 49, 51]. An approximate mathematical analysis has been made in [50], where the mentioned effect was studied considering zero air speed. One of the most referenced models of the ground effect on rotary-wing crafts was introduced by Cheeseman and Bennet in 1957 [3], it predicts the ratio of in-ground effect (IGE) to out-of-ground effect (OGE) thrust for a single rotor operating at constant power, as a function of the rotor height and propeller radius. The ratio was expressed as

$$\frac{T_{IGE}}{T_{OGE}} = \frac{1}{1 - \frac{1}{16}\left(\frac{R}{z}\right)^2 \left[\frac{1}{1 + (V/v)^2}\right]}, \tag{3.1}$$

where T_{IGE} and T_{OGE} are the thrust inside and outside the ground cushion, R and z are the radius of the rotor and the height of the rotor above the ground, V is the airspeed and v is the induced velocity at the rotor.

In the 1970's, Heyson conducted extensive analysis on induced velocities for helicopters [52]. His work paid particular attention to the changes in airflow when the aircraft were operating in ground effect. It showed that ground effect produces a quadratic decrease in power required to hover, with 20% less power needed when the rotor was one radius from the ground. A significant amount of research has been generated since then, exploring experimentally, analytically, and computationally different effects of the ground on the aerodynamics of the main rotor in conventional helicopters [53, 54, 55, 56].

The computational fluid dynamics (CFD) area has been enabling the study of wake produced by rotors in presence of boundary conditions. The focus lies in the wake of full-size helicopters. Simple models for flat surfaces have been proposed, and also complex models



(a) (b)

Figure 3.2. Rotor wake simulations, a) SilAnx geometry [57], b) full 4-blade free wake after some time of simulation [58].

for proximity to objects of arbitrary shapes. Simpler models consider one rotor, and their performance is degraded by using more than one rotor. Complex models consider any number of rotors and arbitrary shapes, but they are computationally expensive, and yet they only provide approximations.

One example of a simple computation wake technique is the SilAnx model [57]. It is a vortex wake model made of groups of multi-circular vortices. The geometry of the rotor wake is exposed in Fig. 3.2(a). The idea behind its creation was to get a rotor vortex wake model simple enough in its geometry to require a low computational time for simulations. Thus, simple vortex shapes are used: circular vortices for the trailing vortices and rectilinear segments for the shed vortices. A new group of concentric vortex rings is emitted at each rotor revolution. By the principle of mass flow conservation, the rotor wake geometry is altered depending on the height of the rotor above the ground.

Another approach in wake modeling is to describe wake as a set of vortex lattices which are freely convected with time. Here, the assumption of a specific geometry during the propagation is removed. For this reason, they are called free wake models. A case proposed by Lebouar et al. [58] consists on the discretization of the wake with a high order panel representation. In this case, the unsteady evolution of the wake is described by a potential discontinuity surface. In Fig. 3.2(b) a visualization of the free wake far from the ground is shown. In this model, the simulation of flat ground is currently realized by a method of images.

The previous two methods are commonly referred as lifting line and surface methods, re-

spectively. The so called boundary element methods are one step up from lifting line and surface methods. They are capable of modeling bodies other than lifting surfaces. As such, they can be programmed to simulate aircraft in full configuration. A method of images was initially implemented for ground effect simulation in conditions in which a symmetry plane could be identified. This method is accurate and efficient, but is only possible when symmetry conditions on the flow field can be detected [60]. This is not the case when of a rotor flying in proximity of obstacles or arbitrary shapes. A solution to this problem is represented by the surface singularity method (SSM) which is a more computationally elaborate technique but may be applied with greater flexibility. In this approach the surface of the obstacle is discretized into quadrilaterals panels on which a singularity is placed, The strengths associated with each panel are evaluated by the boundary condition of flow tangency at all control points on the surface.

More complex simulation techniques exist providing numerical solutions to the Navier-Stokes equations. They describe both the viscous and the rotational phenomenon which constitute the wake. In some implementations, discretization is realized following the method of lines, where in space second order accuracy is obtained via central differencing stabilized by artificial dissipation operators [59]. Temporal discretization can be done of local time stepping in the case of steady-state of dual time stepping for unsteady flow scenarios. Unfortunately, these models require significant computational power to predict forces and torque for specific flight conditions.

3.5 Control of Aerodynamic Interactions

In the small-size UAVs community, the ground effect has a much shorter history of inquiry. An adaptive control technique for the landing of a scale model helicopter was proposed in [61], in which the ground effect was coped by estimating the lift coefficient online. The same technique was used by Guenard et al. [62] for the altitude control of a tele-operated quadrotor. In this case, the system was valid for stationary and quasi-stationary flight, and it relied on an ultrasonic sensor for estimating the altitude velocity. Another adaptive technique was presented in [63]. Here, the adaptation was made within a sliding mode framework, simulations were carried out considering uncertainty and sensor noise. The resulting technique was then implemented for the vision-based landing of a small quadcopter on a moving platform [64]. A group of the University of Shiraz exposed a similar sliding control [65]. The main difference is that they used a Takagi-Sugeno adaptive fuzzy technique to approximate augmented

input control and disturbances (including ground interaction). They provide simulations for a quadrotor with sensor noise. In the same year, 2011, Nonaka and Sugizaki implemented an altitude control for a small coaxial helicopter, utilizing an external charge-coupled device (CCD) camera [66]. The authors recurred to an empirical formula for ground effect compensation; though simple, this formula cannot be generalized to other vehicles. Other approach employed an intuitive visual feedback to estimate ground effect [67]. The main idea is that streamers lying on the ground will move around faster due to the pressurized airflow causing the ground effect. Then, the ground cushion force was learned with an ϵ-Support Vector Regression model using experimental data. The major disadvantage of this work is that it heavily depends on artificial cues.

A more recent study reported results from a set of experiments conducted with a micro-quadrotor demonstrating both the ground and ceiling effects on the rotorcraft [68]. Their findings indicate that the cushion effect begins to manifest at heights significantly higher than predicted by Cheeseman and Bennet's model in Eq. (3.1). A closely related work assessed the applicability of the mentioned model in the hovering of an octocopter with two rotors on four arms [69]. For this case, the results showed that the effect is noticeable at three to five radii. Both works used a mo-cap system for the experiments.

The most closely related work to this research is the one presented by Bartholomew et al. [4]. In 2014, they described a system for predicting upwards acceleration produced when flying above obstacles. A Gaussian process regression approach was trained with experimental data obtained in several test flights over obstacles with a conventional PID control. The model takes as input raw point clouds of objects synthesized from a forward point of view of the quadrotor. A year later, authors applied the developed prediction scheme to the control loop [5]. The prediction of the model was implemented as a feed-forward term after the PID action on the throttle command.

Chapter 4

Research Proposal

In this section, the methodology and schedule for the development of this research are presented. Also, the publications' plan is described.

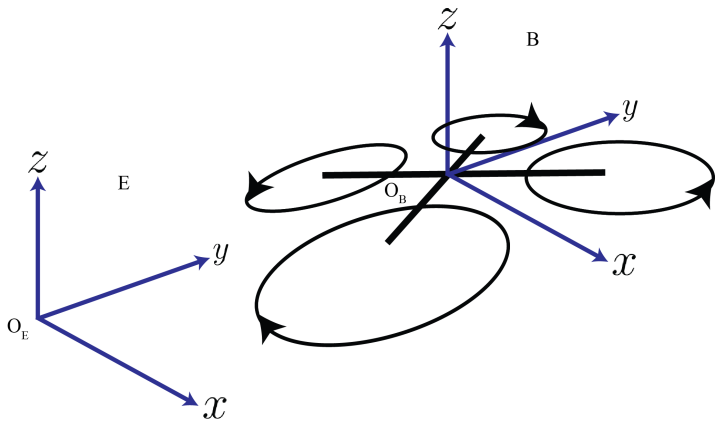


Figure 4.1. The quadrotor coordinate frames definition.

4.1 Methodology

The first step is to establish the coordinate frames that shall be used for modeling the quadrotor. As stated in Sec. 2.4, there are two coordinate systems to be clearly defined: the earth inertial frame (E-frame), and the body-fixed frame of the vehicle (B-frame). These coordinate systems are depicted in Fig. 4.1.

In general, the following reasonable assumptions are made concerning the modeling of the quadcopter:

- The origin of the body-fixed frame coincides with the center of mass of the vehicle.
- The acceleration of gravity is constant and perpendicular to the surface of the Earth.
- The quadrotor, as well as the propellers, will be treated as rigid bodies.

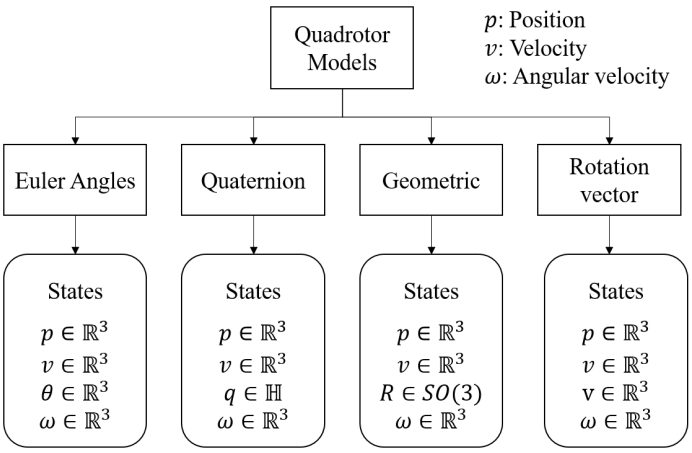


Figure 4.2. Summary of the quadcopter models and their corresponding states.

According to Sec. 2.4 and to the literature review, there exist four different types of quadcopter models. The main difference lies on how the attitude is represented. The most common model corresponds to a sequence of Euler angle rotations, in which the resulting rotation matrix has a fixed rotation sequence. The use of quaternions has been introduced to avoid the singularities of Euler angle sequences and to increase the accuracy the attitude integration process. On the other hand, the geometric approach, also named as the globally defined model, was adopted to avoid the ambiguities of quaternions when representing attitude. Finally, the rotation vector has been proposed as a compact attitude representation without singularities or ambiguities. However, the operations of this representation require the conversion between other formalisms, mostly to quaternions or to a rotation matrix. It is important to note that the state vector for every representation changes in every case, as depicted in Fig. 4.2.

The previous models will be tested in order to choose the best parametrization. The criteria of evaluation will be computational complexity and generalization. It appears that the rotation matrix and the rotation vector provide the best measure of generality because they were used to avoid singularities and ambiguities. Nevertheless, the computational complexity must be tested to guarantee the optimal execution of the model.

Apart from the model, the configuration of sensors needs to be established. Sensors discussed in Sec. 2.2 have been used primarily to measure the state of the vehicle. In addition, external disturbances can be observed or estimated indirectly with those sensors. There are other less ordinary sensors that are being employed to directly measure disturbances. For example, various differential pressure sensors installed on the quadcopter were used to detect

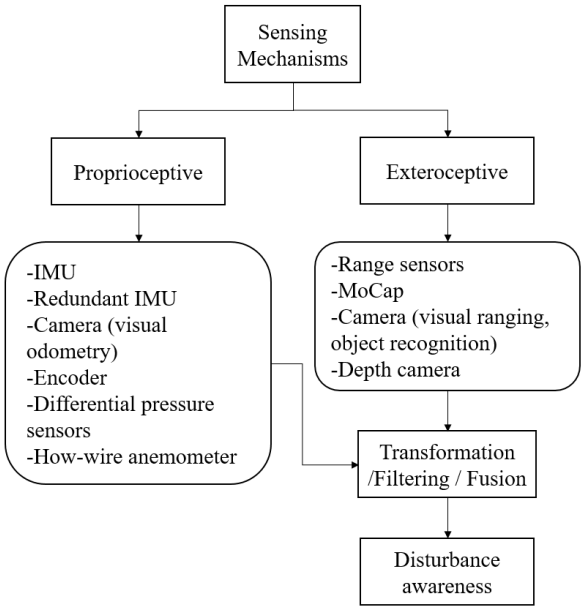


Figure 4.3. Classification of sensing mechanisms.

the downwash of other rotorcrafts. Also, wind measurements can be obtained using hot-wire anemometers on-board the vehicle. The use of a depth camera to gather information about the environment to later predict the disturbance in the function of the shape is another idea to take into account. On the other hand, the increase of accuracy of the measured state of the vehicle can improve the detection of disturbances. In this context, the use of redundant IMUs is known to improve the information about the state of an aircraft [77, 78]; here, the geometrical configuration is important. Also, optical sensors (encoders) commonly aid in fixed thrust benches, but they can be installed on-board to directly measure the rotation velocity of each propeller. To sum up, all sensors can be classified into two broad classes: proprioceptive (internal state), and exteroceptive (information about the environment). The said classification are shown in Fig. 4.3, which also shows that sensors can help, directly or indirectly, to be aware of disturbances.

Once the measuring elements are set, control techniques can be implemented. General closed-loop schemes, like the one shown in Fig. 2.5, can be extended to have either an observer or an estimator of external disturbances. The former is based on a deterministic formulation, and the latter can take into account process and sensor noise. One of the advantages of the observer is the already available tools to prove stability. That aside, observers tend to degrade as the noise increase. Moreover, they would be more appropriate to handle persistent rather than instantaneous disturbances [79]. Estimators can adequately handle noisy mea-

surements have been developed to estimate force and torque [80]. This research will assume general disturbance to have the following form. Let Δ_s be the disturbance vector in the state space of the quadcopter, it will be separated as

$$\Delta_s = \delta_{\text{rand}} + \delta_{\text{aero}}, \tag{4.1}$$

where, δ_{rand} and δ_{aero} are the vector of disturbances due to random external disturbances and the disturbance vector of aerodynamic interactions.

Now, this part is concerned with the areas of knowledge that are related to the problem depicted in Sec. 1.3. Fluid mechanics is a broad field of knowledge that has been closely related to the study of fluid interactions that occur in spinning rotors. In this area, several helicopter wake models have been proposed, which can be used to approximate the wake produced by individual propellers of the quadcopter. Computational fluid dynamics simulations, even though expensive, can be used to gather qualitative data of the aerodynamic disturbances. Another big area of research is the control theory. The quadcopter platform has proved to be a great platform for testing advanced control algorithms. Control methodologies are applied to drive a system to a desired state or reference, commonly using feedback. For quadrotors, several sophisticated control schemes have been shown to provide robustness in presence of random disturbances. Control techniques can be augmented to have a disturbance observer, or signal processing techniques can be applied to fuse sensors' information to estimate disturbances. However, including prior information of the disturbance can improve the performance of the whole system. Here is where artificial intelligence (AI) has been applied. For example, machine learning systems have been set to consume previous flight information to learn vertical accelerations due to the presence of objects. The big picture is that three main areas of active research are closely related to the problem of aerodynamic interactions in the flight of rotorcrafts: fluid mechanics, control theory, and AI (see Fig 4.4).

Starting to work with control theory is a reasonable decision, since a control scheme is needed to remotely fly the quadcopter anyway. The control technique must be able to lead the vehicle to desired states. Computational optimization techniques may be implemented to obtain the controller that meets the desired response. Few carefully selected control techniques may be implemented to prove their disturbance rejection capabilities. In general, controllers are designed using a model of the real process. Then, the next step could be the evaluation of the inclusion of aerodynamic terms (such as blade flapping, drag coefficient, etc.) in the model (intersection 4). Natural overlaps between control theory and AI occur

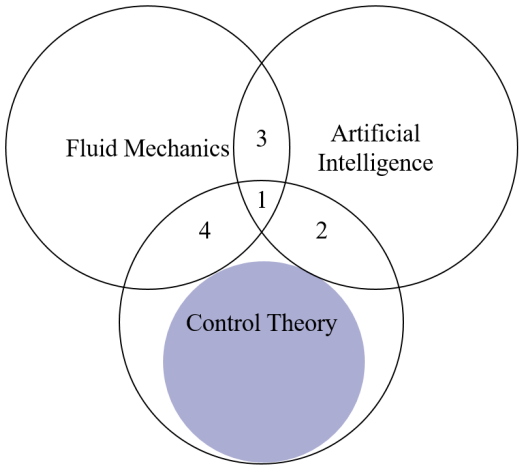


Figure 4.4. Branches of knowledge directly related to the problem of aerodynamic interactions.

in control strategies (intersection 2). For instance, an adaptive controller may change its parameters according to an intelligent computational entity. Also, computational fluid or wake simulations could be used to gather qualitative information about the behavior of disturbances when the vehicle is flying near obstacles. This information may later be analyzed with the aid of artificial intelligent algorithms to acquire new knowledge (intersection 3) or to reduce computational time. Hence, the three big areas and their intersections shall be explored to come up with a final strategy solve the problem stated in Sec. 1.3. The eventual result will probably fall on intersection 1.

The intended methodological procedure for the development of the proposed research is as follows:

1. Evaluation and selection of the sensor arrangement

Assimilation of the relevant types of sensors integrated on- and off-board the quadcopter. Particular emphasis shall be on the use of the Vicon mo-cap system, on-chip IMU, and depth cameras. The first mentioned sensor is selected because it offers high-precision high-frequency measurements of the states. The last two sensors were chosen according to literature which is converging to the same set of minimum on-board sensors for autonomous navigation: an IMU and a camera. In an intuitive way, the former is similar to the human sense of equilibrium and the latter to the sense of sight. A monocular camera could be used instead of a depth camera, but the implementation of algorithms to recover structure is out

of the scope of this research. Miniaturization of sensors poses the option to explore different configurations on-board the vehicle.

2. Design and development of a quadrotor platform

As a means of enabling motor-level control, a custom quadrotor platform is going to be developed. Concretely, the structure of an AR Drone will be used, with its main board substituted for an Odroid XU4 (2GHz Cortex-A7 octa-core processor, 2GB LPDDR3 RAM, 64GB eMMC storage), and the proprietary speed controllers will be replaced for a SimonK rated for 20A. This new board shall be used to load custom control algorithms. The board will be connected to a 4-channel bi-directional logic level converter (to step-up the 1.8V Odroid bus reference to 5V), then to an 8-channel PWM controller (to allow the control of the four motors through only two cables), and finally to the ESCs. Additionally, the contemplated IMU device to operate is the contemporary on-chip LPMS-B2, which is capable of output rates up to 400Hz. Due to limited payload capacity, the camera will not be mounted on the quadcopter.

3. Implementation of the quadrotor dynamic model and control

This stage will consist of the evaluation of existing dynamical models and controls for the quadrotor. The judgment criteria shall consider the time complexity of the computational model and its generality. A selective modal analysis will be used to get the relevant dynamics of the system [75] in order to find an adequate numerical strategy. The intended outcome will be an efficient computational model that describes the quadcopter dynamic behavior in non-restricted conditions (i.e., far from obstacles) with motor inputs, which will be commanded by two algorithms: the base control algorithm (or the one that is going to be enhanced with the aerodynamic interactions model), and a state-of-the-art control algorithm. The inclusion of an estimator for the base control will be evaluated. The spatial complexity will be analyzed to guarantee that the whole system fits in the hardware.

4. Experimentation in near-surface conditions

The base control system will be used to better understand the behavior of variables when the quadcopter is subjected to disturbances due to aerodynamic interactions with surfaces. Data from all available sensors and control inputs will be collected in each trial. Two main flight conditions are of interest: hover, and constant velocity (in essence, equilibrium conditions). The custom quadrotor will be set to fly in different near-surface conditions such as

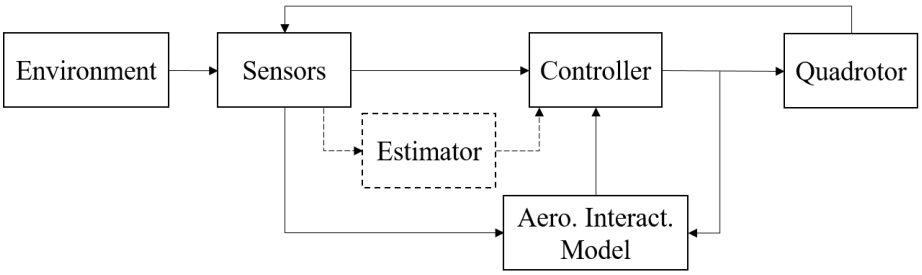


Figure 4.5. Suggested architecture for the aerodynamic interactions control.

close to the ground, ceiling, and wall. This task intends to answer the first research question. For every experiment, the dependent variables shall be the force and torque vectors, the independent variables include but are not restricted to: attitude, delivered commands for each motor (or efference copies, referring to internal copies of outflowing motion-producing signals), forward velocity, minimum distance between the aircraft's center of mass and the surface, point cloud of the environment's section coinciding with the location of the quadrotor. Special care shall be taken for minimizing (controlling) external wind gusts.

5. Encoding aerodynamic interactions in a mathematical model

Using the experimental and simulated data (from CFD) a process of knowledge discovery, based on AI algorithms (e.g., deep learning), will yield the variables that best describe aerodynamic interactions. A dimensional analysis may be used as a means to reduce the search space of input variables. Resulting models shall be validated in this step by comparing against well-established models. The second research question is expected to be answered here.

6. Aerodynamic interactions control evaluation

Models from the previous step shall be incorporated into the base control. Then, evaluation of the schemes will be carried out by comparing tracking performance against the state-of-the-art controller. Root mean squared error of the tracking trajectories shall be used as evaluation indicator. This last step will result in the answer of the third research question.

In accordance with the given information in this section, the suggested architecture for the control of aerodynamic interactions is illustrated in Fig. 4.5. The estimator and the control are intended to suppress random external disturbances, whereas the aerodynamic interactions model is intended to cancel out δ_{aero}.

4.2 Work Plan

The Gantt chart, with the timing of the activities and dates of publications, is presented in Fig 4.6.

4.3 Publications Plan

The expected publications and their targets are presented below.

1. Conference paper (full length)

Target: The 2017 International Conference on Unmanned Aircraft Systems (ICUAS). Aim: Development and simulation of control algorithms for quadrotors. Expected results: Preliminary advances presented in this document. Submitted, deadline: February 26, 2017.

2. Conference paper (full length)

Target: Special Issue on Unmanned Aircraft Systems (UAS). Aim: Implementation of control algorithms in a quadrotor platform. Expected results: A quadrotor platform ready for experimental procedures for aerodynamic interactions. Submission date: September 30, 2017.

3. Journal paper

Target: Journal of Intelligent & Robotic Systems. Aim: Establishment of the basis for modeling quadrotor-environment aerodynamic interactions. Expected results: A model describing the aerodynamic interactions between the vehicle and its environment. Estimated submission date: September, 2018.

4. Journal paper

Target: IEEE Robotics and Automation Letters. Aim: Presentation and implementation of a control scheme for the autonomous flight of a quadrotor in constrained indoor environments based on the model of aerodynamic interactions. Expected results: Validation of the proposed control scheme. Estimated submission date: September, 2019.

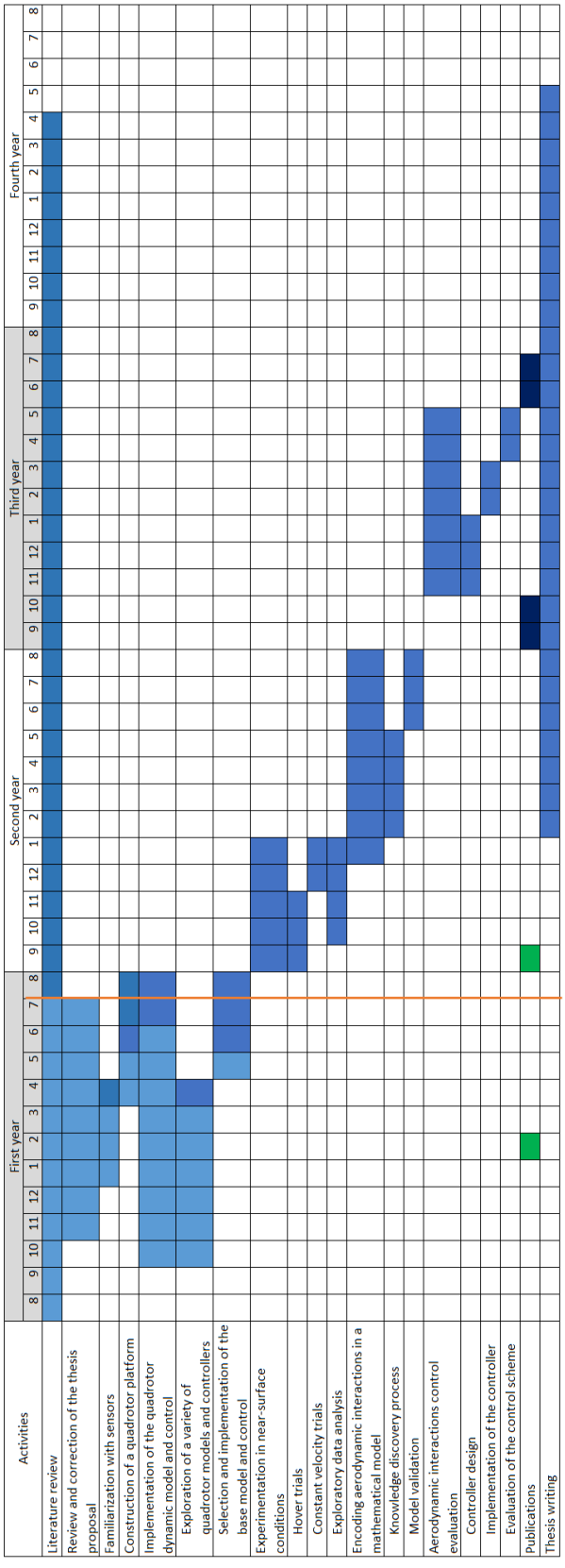


Figure 4.6. Task scheduling and current progress.

Chapter 5

Preliminary Progress

This last section presents some of the preliminary advances made during the first year. The actual area of knowledge in exploration is shaded in Fig. 4.4, which corresponds to control theory.

For quadcopter control, optimization methods have been applied to reduced models of the quadcopter, typically linear. In contrast, a first taken approach was to explore three optimization techniques to control a more elaborate nonlinear quadrotor model. Thus, the contributions are:

- Parameter tuning of a PD control for a quadrotor by means of the conjugate gradient algorithm with two cost function gradient computation strategies: approximated gradient by first-order difference formulas, and analytic Jacobian gradient.
- Optimal control law computation for a quadrotor applying optimal control theory alongside a numeric collocation technique.
- The quadrotor model to be controlled is nonlinear and coupled.

5.1 Quadrotor Model

The mathematical model of a quadcopter is derived introducing two operation frames, the inertial frame and the body frame. The first is fixed to the Earth, with gravity pointing in the negative z direction. The second frame is attached to the drone center of mass (CM) and describes its orientation. Both particular frames are illustrated in Fig. 5.1

The position and velocity of the quadrotor in the inertial frame is defined as (x, y, z) and (u, v, w), the body-frame Euler angles as (ϕ, θ, ψ), and the angular rate towards body axis as (p, q, r). It is important to note that $(p, q, r) \neq (\dot{\phi}, \dot{\theta}, \dot{\psi})$. The body angular rate is a vector pointing along the axis of rotation and the other quantities are just the time derivatives of the roll, pitch, and yaw. The relation between those variables and between the frames are

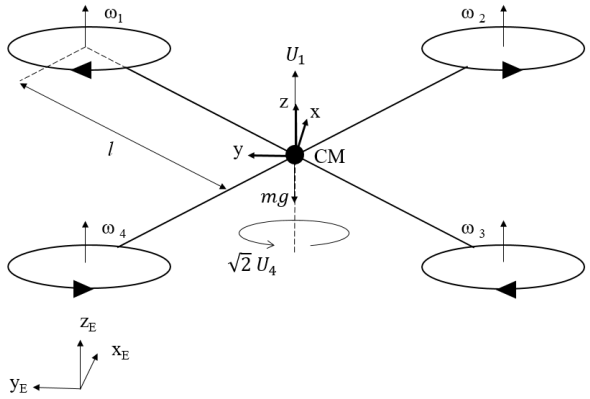


Figure 5.1. Frame system for modeling the quadrotor dynamics, more details are given than in Fig. 4.1.

expressed as transformation matrices, which details are not going to be discussed here for the sake of brevity. The state space equations, the same as in [24], are the following:

$$\begin{aligned}
\dot{z} &= w \\
\dot{w} &= U_1 \cos\phi \cos\theta / m - g \\
\dot{\phi} &= (p \cos\theta + q \sin\phi \sin\theta + r \cos\phi \sin\theta) / \cos\theta \\
\dot{\theta} &= q \cos\phi + r \sin\phi \\
\dot{\psi} &= (q \sin\phi + r \cos\phi) / \cos\theta \\
\dot{p} &= [\sqrt{2} l U_2 + qr(I_y - I_z)] / I_x \\
\dot{q} &= [\sqrt{2} l U_3 + pr(I_z - I_x)] / I_y \\
\dot{r} &= [\sqrt{2} U_4 + qp(I_x - I_y)] / I_z
\end{aligned} \quad , \tag{5.1}$$

where U_1, U_2, U_3, U_4, are the controller outputs, or the plant control inputs. The parameters values are shown in Table 5.1.

5.2 Mathematical Problem

The structure of the dynamic system (5.1) is given by an initial value problem **(IVP)** of the form

$$\dot{x}(t) = f(x(t), u(t), t) \qquad x(t_0) = x_0, \tag{5.2}$$

Table 5.1. Parameter description.

Parameter	Value	Description
g	9.81 m/s^2	Gravity acceleration
m	1 kg	Quadrotor mass
l	0.24 m	Distance between motor and CM
I_x, I_y, I_z	0.08 kg·m^2	Body inertia

where $\dot{x} = dx/dt$, $x \in \mathbb{R}^n$ is an Euclidean space, x_0 is the initial condition, and u is the control input that belongs to the vector space $\mathcal{U}$, f is a vector value-function on $\mathbb{R}^n \times \mathcal{U}$ linear or nonlinear. It is assumed that the initial time t_0 and the final time t_f are given. The performance index of this system over the time interval $[t_0, t_f]$ is considered to be given by the cost function

$$J(u) = h(x(t_f), t_f) + \int_{t_0}^{t_f} g(x(t), u(t), t) = \phi(x(t_f, u)), \tag{5.3}$$

where $x(t_f, u)$ is the state x at time t_f.

We can observe that the cost function is a function of the final state reached and measures the penalty that must be paid because of the dynamic system's trajectory (5.2).

The problem is to find $u^* \in \mathcal{U}$ that causes the system (5.2) to follow a trajectory x^* that minimizes the cost function (5.3); that is

$$u^* = \arg\min_{u \in \mathcal{U}} J(u), \tag{5.4}$$

where the function J is given by (5.3) and the state variable x satisfies the initial value problem (5.2)

To resolve the optimization problem, a simulation-based optimization methodology is used: a) the physical process given by (5.2) is simulated over $[t_0, t_f]$ given an input control u, b) the functional J in the reached final state is evaluated, c) an algorithm to find the optimal u is employed. This process is iterative and can be time-consuming.

5.3 Conjugate Gradient Methods

The Conjugate Gradient algorithm requires the gradient of the function to be optimized. There are two possibilities: to provide the analytical gradient or to provide the gradient ap-

proximated by differences formulas. In general, it is recommended to give the analytical gradient to reduce the numerical errors and to avoid difficulties to the unconstrained optimization problem. However, it was already pointed out in Sub. 5.2 that the cost function is the result of a computational simulation, and can be impractical or expensive to compute the analytical gradient.

The Conjugate Gradient methods are simple and easy to implement. The complexity computational of these types of algorithms is linear, $O(n)$, in comparison of the cubic computational complexity, $O(n^3)$, per iteration step of Newton's methods or the quadratic computational complexity, $O(n^2)$, for the Quasi-Newton methods. Likewise, with respect to the memory requirements, Newton-type methods and Quasi-Newton-type methods require quadratic memory complexity in comparison of the linear memory requirement complexity of the Conjugate Gradient type methods. Owing to the linear computational complexity and the linear memory requirement the CG-type methods are best suited for large problems, $n \geq 1000$, and they may outperform Newton type methods or Quasi-Newton type methods [70], [71].

Among the CG-type methods more popular are the Fletcher-Reeves method (FR-CG), the Polak-Ribière method (PR-CG) and the Polak-Ribière positive method (PR+). In practice, the PR-CG method performs better than FR-CG [72].

For this case, the analytical gradient of cost function (5.3) is given by [73]

$$\nabla J(u(t)) = f_u^T \lambda(t), \tag{5.5}$$

where $u = u(k_1, k_2, \ldots, k_m)$, the mathematical model is given by the system of ordinary differential equations (ODEs) (5.2), f_u is an $n \times m$ matrix $(f_u(x,u))_{ij} = \partial f_i(x,u)/\partial k_j$ $i = 1, 2, \ldots n$, $j = 1, 2, \ldots, m$ and $\lambda(t)$ satisfies the final value problem on $[t_0, t_f]$

$$\dot{\lambda} = -f_x^T \lambda(t) \qquad \lambda(t_f) = \nabla_x \phi(x(t_f, u)), \tag{5.6}$$

and $(f_x(x,u))_{ij} = \partial f_i(x,u)/\partial x_j \quad i,j = 1, 2, \ldots n$. Hence, computing the gradient requires the state variable $x(t_f)$ and the control variable u, it is necessary to find the solution of (5.2), and (5.6) for each input of u. Likewise, we need to approximate numerically the integral given by (5.5).

When using the gradient of the cost function approximated by difference formulas of first order, we take $J = J(k_1, \ldots, k_m)$ and calculate $\partial J/\partial k_j, j = 1, 2, \ldots, m$ as

$$\frac{\partial J(k)}{\partial k_j} \approx \frac{J(k + he_j) - J(k)}{h}, \tag{5.7}$$

where $k = (k_1, k_2, \ldots, k_m)$, $e_j = (0, 0, \ldots, 1, 0, \ldots, 0)$ with 1 in the place j–th and h is a constant greater than zero.

5.4 Optimal Control

To find the optimal control that causes the system (5.2) to follow a trajectory x^* that minimize the cost function (5.3), we need to apply variational methods to optimal control problems. In our case, by the Pontryagin's Maximum Principle (**PMP**) we need to solve the following equations

$$H = g(x(t), u(t), t) + \lambda^T f(x(t), u(t), t) \tag{5.8}$$

$$\dot{x}(t) = \frac{\partial H}{\partial \lambda}(x(t), u(t), \lambda(t), t) \tag{5.9}$$

$$\dot{\lambda}(t) = -\frac{\partial H}{\partial x}(x(t), u(t), \lambda(t), t) \tag{5.10}$$

$$0 = \frac{\partial H}{\partial u}(x(t), u(t), \lambda(t), t), \tag{5.11}$$

where $t \in [t_0, t_f]$, $H(x(t), u(t), \lambda(t), t)$ is the Halmitonian of the system, $\lambda' s$ are called the co-state equations, $f(x(t), u(t), t)$ is given by (5.2), $g(x(t), u(t), t)$ is the integrand of (5.3), and $\dot{\lambda}(t)$ is subject to the terminal condition

$$\lambda(t_f) = \frac{\partial h}{\partial x}(x(t_f), t_f). \tag{5.12}$$

It should be observed that with the equation (5.11) we can obtain the control law of the model. Besides, the equation (5.9) is the same that (5.2).

5.5 Boundary Value Problems

To find the numerical solution of (5.8), (5.9), (5.10) and (5.11) subject to (5.12), we can reformulate it as a Boundary Value Problem (**BVP**) and then use a collocation method for the solution of BVP's with the following structure

$$y' = f(x, y, p), \qquad a \leq x \leq b \tag{5.13}$$

subject to general nonlinear, two-point boundary conditions

$$g(y(a), y(b), p) = 0, \tag{5.14}$$

where p is a vector of unknown parameters. For the above, the function bvp4c of MATLAB was used. This function implements a collocation method for the solution of BVPs [74]. The idea of a collocation method is to choose a number of points in the domain function, which are called collocation points, and choose a vector space V whose dimension is finite. In this space V are the candidate solutions, for example polynomials, then select a solution which satisfies (5.13) with (5.14) at the collocation points. The function bvp4c use cubic polynomials.

5.6 Optimal PD Controller

The PD controller feeds the plant with some weight of the error and other weight of its derivative. These weights are the parameters to be optimized by the algorithm. The entire set of parameters is $u = (k_{p1}, k_{p2}, k_{p3}, k_{p4}, k_{d1}, k_{d2}, k_{d3}, k_{d4})^T$, where k_{pi} and k_{di} are the so-called proportional and derivative parts, respectively. We want to drive the plant to the set-points or desired values $(z_d, \phi_d, \theta_d, \psi_d)$. The form of the quadrotor inputs in the case of a PD controller is given below

$$\begin{aligned} U_1 &= k_{p1}(z_d - z(t)) - k_{d1}\dot{z}(t), \quad U_2 = k_{p2}(\phi_d - \phi(t)) - k_{d2}\dot{\phi}(t), \\ U_3 &= k_{p3}(\theta_d - \theta(t)) - k_{d3}\dot{\theta}(t), \quad U_4 = k_{p4}(\psi_d - \psi(t)) - k_{d4}\dot{\psi}(t). \end{aligned} \tag{5.15}$$

A cost function is formulated to maintain the system state as close as possible to the desired state and to minimize all the coefficients. The second objective is attainable by introducing a weighting factor R. Therefore, the cost function is written as

$$\begin{aligned} J(u) = \int_{t_0}^{t_f} [&(z_d - z)^2 + (\phi_d - \phi)^2 + (\theta_d - \theta)^2 + (\psi_d - \psi)^2 \\ &+ R(k_{p1}^2 + k_{p2}^2 + k_{p3}^2 + k_{p4}^2 + k_{d1}^2 + k_{d2}^2 + k_{d3}^2 + k_{d4}^2)]d\tau. \end{aligned} \tag{5.16}$$

The cost function will be treated as a state of the system, which can be done differentiating

with respect to time the left-hand side and the right-hand side of (5.16) to yield

$$\dot{J}(u) = (z_d - z)^2 + (\phi_d - \phi)^2 + (\theta_d - \theta)^2 + (\psi_d - \psi)^2 + R(k_{p1}^2 + k_{p2}^2 + k_{p3}^2 + k_{p4}^2 + k_{d1}^2 + k_{d2}^2 + k_{d3}^2 + k_{d4}^2 + k_{d4}^2). \quad (5.17)$$

Substituting (5.15) into (5.1) and then concatenating (5.17) a new state space system $\dot{x} = f(t, x(t), u)$ can be formed with $x = [z, w, \phi, \theta, \psi, p, q, r, J(u)]^T$. Once this system is solved forwards in time using a fixed u, we can retrieve the criterion as $J \leftarrow x_9(t_f)$. Those would be all the requirements to apply CG if we adopt approximated Jacobian gradient. On the other hand, an extra effort is needed to utilize the analytic version of the gradient in the optimization process.

The analytic Jacobian gradient is calculated as follows. The adjoint (co-state) system, which is analytically formed with $f_x \in \mathbb{R}^{9\times9}$ as declared by (5.6), is integrated backwards in time with terminal condition $p(t_f) = \nabla_x \phi(x(t_f, u)) = (0, 0, 0, 0, 0, 0, 0, 0, 1)^T$. This integration delivers the solution $p(t) \in \mathbb{R}^9$ that is used to calculate our particular eight gradient directions as stated by (5.5), where f_u is in $\mathbb{R}^{9\times8}$

With respect to the numerical design, a fourth order fixed step solver based on Runge-Kutta algorithm is used to solve the dynamic model and the co-state system for the analytic gradient case. The optimization problem is solved using a Polak-Ribière conjugate gradient method which in turn uses a soft line search strategy. The initial conditions of the state vector are $x(t_0) = (0, 0, \pi/4, \pi/4, \pi/4, 0, 0, 0)$, the set-points are $(z_d, \phi_d, \theta_d, \psi_d) = (1, 0, 0, \pi/6)$, the weighting factor R is set to 10^{-5}, the initial guess of parameters is $k_{pi} = k_{di} = 1.5$, and the system is run from $t_0 = 0$ s to $t_f = 5$ s using a step of 0.01 s.

5.7 Optimal Controller

In a similar way to Sec. 5.6, we want to drive the plant to the set-points or desired values $(z_d, \phi_d, \theta_d, \psi_d)$. In this case, we have 16 ODEs, 8 correspond to the states x and 8 to the co-states λ. The initial conditions of the state vector are $x(t_0) = (0, 0, \pi/4, \pi/4, \pi/4, 0, 0, 0)^T$, the boundary condition of co-state vector $\lambda(t_f) = (0, 0, 0, 0, 0, 0, 0, 0)^T$, the set-points are $(z_d, \phi_d, \theta_d, \psi_d) = (1, 0, 0, \pi/6)$, the weighting factor R is set to 10^{-5}. Finally, the system is solved from $t_0 = 0$ s to $t_f = 5$ s.

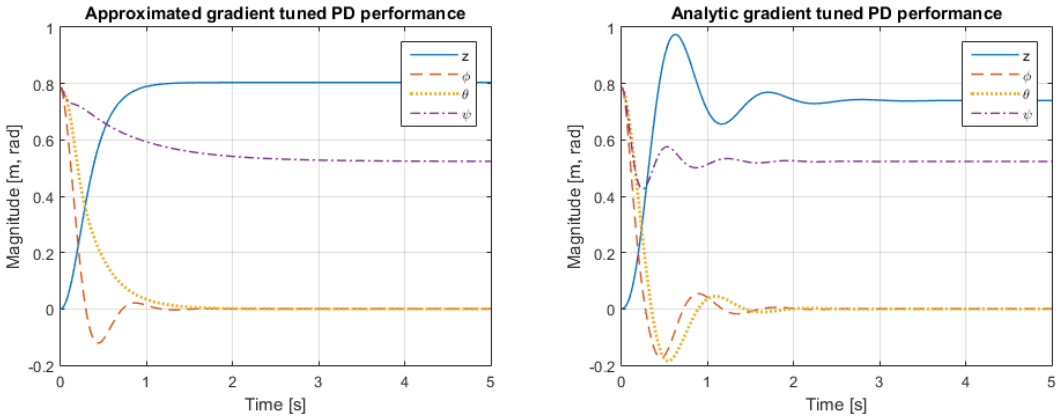


Figure 5.2. PD tuned controller performance: approximated gradient (left), analytic gradient (right).

The optimal control $u(t) = (u_1(t), u_2(t), u_3(t), u_4(t))^T$ is obtained from (5.11), where

$$\begin{aligned} u_1(t) &= -\frac{cos(x_3(t)cos(x_4(t))\lambda_2(t)}{mR}, \quad & u_2(t) &= -\frac{\sqrt{2}\,l\lambda_6(t)}{IxR}, \\ u_3(t) &= -\frac{\sqrt{2}\,l\lambda_7(t)}{IyR}, & u_4(t) &= -\frac{\sqrt{2}\lambda_8(t)}{IzR}. \end{aligned} \tag{5.18}$$

The cost function, in this case, is given by

$$J(u) = \int_{t_0}^{t_f} [(z_d - z)^2 + (\phi_d - \phi)^2 + (\theta_d - \theta)^2 + (\psi_d - \psi)^2 + R\sum_{i=1}^{4}(u_i(t)^2)]d\tau. \tag{5.19}$$

Substituting (5.18) into (5.1) we obtain the complete system of ODE $(\dot{x}(t), \dot{\lambda}(t))^T$ and we have setup the Two-Point Boundary Problem that can be solved with the function bvp4c of MATLAB.

5.8 Results and Analysis

The nonlinear quadrotor model (5.1) and the exposed algorithms were coded in MATLAB. All the values of model parameters are provided in Table 5.1.

Numeric results of u^* for the same PD structure using approximated and analytic gradients are shown in Table 5.2. The values of $J(u^*)$ are 0.8450 and 0.7904, respectively. The cost of the approximated gradient method represents a 6.9% increment with respect to the analytic

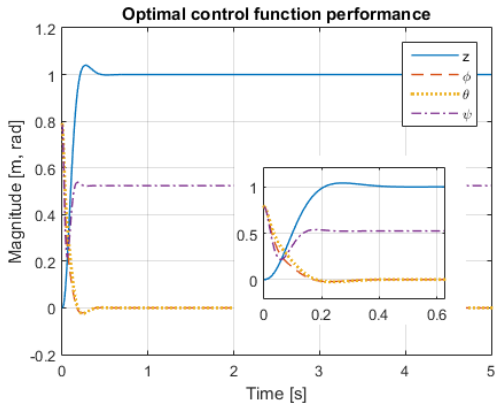


Figure 5.3. States behavior when subjected to the optimal control functions. The inner figure shows the same function performance zoomed in between the time interval 0 to 0.6 in seconds.

Table 5.2. Optimal control parameters and cost function values.

Symbol	Approx. gradient	Analytic gradient	Symbol	Approx. gradient	Analytic gradient
k_{p1}	50.0314	37.6668	k_{d1}	14.6970	3.8609
k_{p2}	16.4364	12.9335	k_{d2}	1.9649	1.2023
k_{p3}	14.9931	9.4117	k_{d3}	5.3477	1.2232
k_{p4}	6.4145	4.7500	k_{d4}	4.7844	0.2196

gradient cost.

Figure 5.2 shows the performance of the optimized PD controller with both approximated and analytic gradient. As expected, the two methods have a nonzero steady state error vector, and the approximate version and the analytic one transfer system variables (z, ϕ, θ, ψ) to $(0.80, 0, 0, \pi/6)$ and $(0.74, 0, 0, \pi/6)$, respectively. The approximated approach stabilizes height close to the mark of 1 s, without neither overshoot nor oscillations. The roll angle is taken to the reference in around 1 s with brief oscillations. The remaining angles, pitch, and yaw reach smoothly their set-points in about 2 s. On the other hand, the analytic version displays an aggressive response. All the plotted states exhibit overshoot and oscillations with a settling time of around 2 s.

For the case of the optimal controller, simulation results are presented in Fig. 5.3. From the figure, it can be seen that the quadrotor is transferred to the desired state values, i.e., the steady state error is zero. Brief overshoots are observed in each plotted state, in particular,

the z-variable exceeds its reference for 4 cm. From all the traces in the plot, height is the one with the maximum settling time of 0.5 s.

The best performance of the three exposed methods goes to the optimal controller, followed by the approximated Jacobian gradient. The 4% percent overshoot in height of the last approach is compensated with a nil steady state and overall faster response. The next position was given to the PD tuned by the approximated Jacobian gradient due to its smoothness and reduced settling time compared to the analytic version. It is important to note that these last two approaches yielded optimal results, each for a different numerical strategy, which is why here the performance is judged by the behavior quality instead of by the numerical value of the cost function $J(u^*)$.

The comparison above can be meaningful to a user of the so far discussed optimization methods. Arguably, the "easiest" approach to be implemented is that of the optimal control when considering the use of an existing boundary value problem solver. The main effort is put into the coding of a $2n$ ODEs system. The following in order is the parametric optimization with approximated Jacobian gradient. This method requires the use of a gradient conjugate algorithm and a numerical integration method. Having met those requirements, the only thing left to do is to implement an extended $n + 1$ model. The most intricate method was the analytic Jacobian gradient. It is an extension of the previous approach, which involves the elaboration of the $n + 1$ co-state. In sum, the last optimization mechanism demands $2(n + 1)$ coded ODEs, plus the numerical integration of m expressions.

5.9 Stochastic Ground Effect Emulation

The method exposed in Section 5.6 for obtaining the optimal PD controller can be directly used to take into account disturbances in the quadcopter model. To include the disturbances, four stochastic signals are added to the model, namely δ_z, δ_p, δ_q, and δ_r, which are the disturbance normal to the quadrotor, the disturbance along the body x-axis, y-axis, and z-axis, respectively. With this signals, disturbances caused by the ground effect may be emulated.

Stochastic signals were generated in the following manner. First, there is an available random number generator followed by a hold. The output of the hold then feeds directly into the input of a linear filter with a transfer function $W(s)$. By controlling of the probability distribution of the random number generator and of the transfer function of the linear filter, the amplitude characteristics and the frequency bandwidth characteristics of the resulting signal may be selected as desired. These considerations are taken to not upset the numerical

integration used to calculate the system response.

The generated random numbers were chosen to have Gaussian distribution with mean zero; the variance varies depending on the disturbed axis. The variances for δ_z, δ_p and δ_q were set to 5, and the variance for δ_r was set to 1. For the generation of δ_z, the linear shaping filter was chosen as a second order filter with dampening factor $\xi = 5.5$ and cutoff frequency $\omega_n = 4.18$ rad/sec, values which were obtained from the Jacobian in a simulation of the model with deterministic signals. The remaining stochastic signals were filtered by a first order transfer function with cutoff frequency $\omega = 1$ rad/sec. These values were chosen according to experimental results for the ground effect [81].

After applying the optimization method to tune the gains with the emulation of ground effect, the results in Fig. 5.4 were retrieved. As can be seen from the behavior of the states, the controller is able to drive the system around the reference values, with exception of the height variable which was taken to an average value below its reference. The qualitative behavior of this controller is in average the same as the one obtained using deterministic signals.

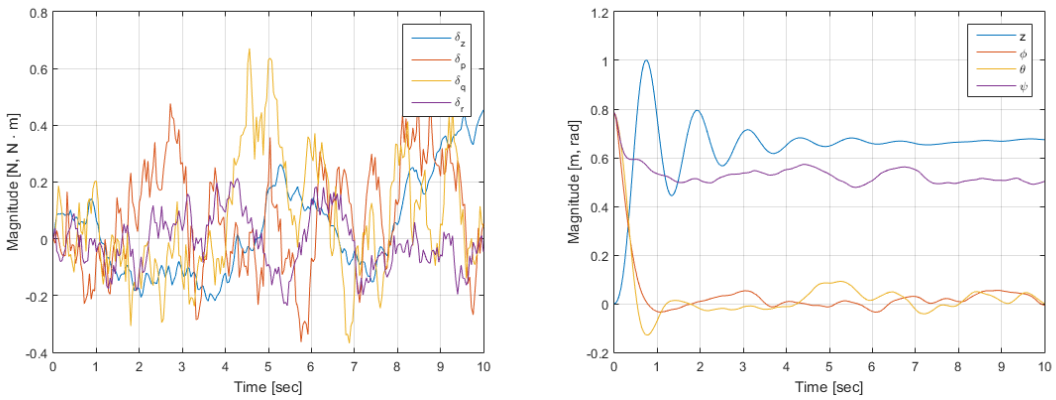


Figure 5.4. PD tuned controller performance with ground effect emulation: filtered stochastic signals (left), states behavior (right).

5.10 Conclusions

The preliminary progress has presented an analysis for three optimization techniques to control a quadcopter whose dynamic model is based on a highly nonlinear model [82]. For this two approaches for PD tuning based on conjugate gradient were presented. A first ap-

proach uses an approximation of the gradient by first-order differences formulas, and the second relies on the analytic Jacobian gradient. From the simulation results, it can be seen that the former demonstrates greater performance compared with the latter, with an arguably simpler implementation, which could be useful for rapid implementation of quadcopter-based applications. Furthermore, these optimization methods were applied to tune a successful controller in presence of a stochastic emulation of the ground effect.

Nevertheless, PD controllers fail to eliminate steady state error when the set-point is nonzero. One solution could be to extend the proposed parametric optimization to consider the whole PID structure. To eliminate the steady state error, the opportunity to explore the third method was taken, it consisted of an optimal controller for the nonlinear quadrotor dynamical model implemented by applying the Pontryagin's Maximum Principle. The proposed method treats the optimization as a boundary value problem. In addition, a numerical collocation method was used for the solution of the BVPs to retrieve the optimal trajectories. This control strategy achieved a null steady state error and fast response, thus outperforming those PD controllers whose tunning was obtained via CG methods.

The preliminary advances allow delving into, as mentioned in Chap. 4, the first big area of knowledge involved in the problem of aerodynamic interactions: control theory. The simulated control mechanism is en route to being implemented on a real platform. Meanwhile, more sophisticated algorithms are being studied (adaptive in particular) which could be employed as a reference scheme for evaluation. Hence, the research is transitioning to the overlap between control theory and artificial intelligence. Computational fluid simulations are planned for the near to medium-term. Two main software packages are being considered for the simulations: ANSYS Fluent and STAR-CCM+ from Siemens. Aerodynamic interactions have not yet been studied experimentally nor computationally. Despite this, the literature review provided evidence which agrees with the hypothesis of this research.

Bibliography

[1] I. Cowling, "Towards Autonomy of Quad-rotor UAV," PhD Thesis, Cranfield University, 2008.

[2] Y. Peng, W. Guo, M. Liu, and S. Xie, "Active modeling based yaw control of unmanned rotorcraft," International Journal on Smart Sensing and Intelligent Systems, Vol. 7, 2014.

[3] I. C. Cheeseman and W. E. Bennet, "The Effect of the Ground on a Helicopter Rotor in Forward Flight," Aeronautical Research Council Reports and Memoranda, 1957.

[4] J. Bartholomew, A. Calway, and W. Mayol-Cuevas, "Learning to Predict Obstacle Aerodynamics from Depth Images for Micro Air Vehicles," IEEE International Conference on Robotics and Automation, May, 2014.

[5] J. Bartholomew, A. Calway, and W. Mayol-Cuevas, "Improving MAV Control by Predicting Aerodynamic Effects of Obstacles," IEEE/RSJ International Conference on Intelligent Robots and Systems, 2015.

[6] J. Diebel, "Representing Attitude: Euler Angles, Unit Quaternions, and Rotation Vectors," 2006.

[7] K. Ogata, "Modern Control Engineering," Fifth Edition, Prentice Hall, 2012.

[8] S. N. Ghazbi, Y. Aghli, M. Alimohammadi, and A. A. Akbari, "Quadrotors unmanned aerial vehicles: A review," International Journal on Smart Sensing and Intelligent Systems, March, 2016.

[9] A. Zulu and S. John, "A Review of Control Algorithms for Autonomous Quadrotors," Open J. of Applied Sciences, Vol. 4, 2014.

[10] J. H. Gillula, G. M. Hoffman, H. Huand, M. P. Vitus, and C. Tomlin, "Applications of Hybrid Reachability Analysis to Robotic Aerial Vechicles," International Journal of Robotics Research, 2011.

[11] B. Erginer and E. Altug, "EKF Based Attitude Estimation and Stabilization of a Quadrotor UAV Using Vanishing Points in Design And Implementation Of A Hybrid Fuzzy

Logic Controller For A Quadrotor VTOL Vehicle," International Journal of Control, Automation and Systems, Vol. 10, 2012.

[12] R. Sanz, P. Garcia, P. Castillo, and P. Albertos, "Time-Delay Compensation Using Inertial Measurement Sensors For Quadrotor Control Systems," 17th International Conference on Information Fusion, July, 2014.

[13] M. D. Hua, G. Ducard, and S. Bouabdallah, "A Robust Attitude Controller And Its Application To Quadrotor Helicopters," 18th International Federation of Automatic Control World Congress, Vol. 18, Part 1, September, 2011.

[14] I. Lenz, M. Gemici, and A. Saxena, "Low-Power Parallel Algorithms for Single Image based Obstacle Avoidance in Aerial Robots," IEEE/RSJ International Conference on Intelligent Robots and Systems, 2012.

[15] E. C. Suicmez and A. T. Kutay, "Optimal path tracking control of a quadrotor UAV," International Conference on Unmanned Aircraft Systems, 2014.

[16] I. D. Cowling, O. A. Yakimenko, J. F. Whidborne, and A. K. Cooke, "Direct Method Based Control System for an Autonomous Quadrotor," Journal of Intelligent and Robotic Systems, Vol. 60, Issue 2, 2010.

[17] J. Keshavan, G. Gremillion, H. Alvarez-Escobar, and J. S. Humbert, "Autonomous Vision-Based Navigation of a Quadrotor in Corridor-Like Environments," Journal of Micro Air Vehicles, Vol. 7, 2015.

[18] R. Amin, L. Aijun, and S. Shamshirband, "A review of quadrotor UAV: control methodologies and performance evaluation," Int. J. of Automation and Control, Vol. 10, No. 2, 2016.

[19] S. Bouabdallah, A. Noth, and R. Siegwart, "PID vs LQ Control Techniques Applied to an Indoor Micro Quadrotor," IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, 2004.

[20] A. M. Singh, D. J. Lee, D. P. Hong, and K. T. Chong, "Successive Loop Closure Based Controller Design for an Autonomous Quadrotor Vehicle," Applied Mechanics and Materials, Vol. 483, 2014.

[21] G. Szafranski and R. Czyba, “Different Approaches of PID Control UAV Type Quadrotor,” in Proc. Int. Micro Air Vehicles Conf., 2011.

[22] H. Bolandi et al., “Attitude Control of a Quadrotor with Optimized PID Controller,” Intelligent Control and Automation, 2013.

[23] J. Kim, S. A. Wilkerson, and S. A. Gadsden, “Comparison of Gradient Methods for Gain Tunning of a PD Controller Applied on a Quadrotor System,” in Proc. SPIE, Vol. 9837, 2016.

[24] J. Zhu et al., “A Gradient Optimization based PID Tunning Approach on Quadrotor,” The 27th Chinese Control and Decision Conf., 2015.

[25] F. Berkenkamp, A. P. Schoelling, and A. Krause, “Safe Controller Optimization for Quadrotors with Gaussian Process,” IEEE Int. Conf. on Robotics and Automation, 2016.

[26] S. Bencharef and H. Boubertakh, “Optimal Tuning of a PD control by Bat Algorithm to Stabilize a Quadrotor,” 8th Int. Conf. on Modelling, Identification and Control, 2016.

[27] D. S. Naidu, “Optimal Control Systems,” CRC Press, 2002.

[28] L. M. Argentim et al., “PID, LQR and LQR-PID on a Quadcopter Platform,” Int. Conf. on Informatics, Electronics and Vision, 2013.

[29] T. Nuchkrua and M. Parnichkun, “Identification and Optimal Control of Quadrotor,” Thammasat Int. J. of Science and Technology, Vol. 17, No. 4, 2012.

[30] A. A. Rubio et al., “Optimal control strategies for load carrying drones,” 2014. ¡hal-009927766¿66¿

[31] M. Faesslet, D. Falanga, and D. Scaramuzza, “Thrust Mixing, Saturation, and Body-Rate Control for Accurate Aggressive Quadrotor Flight,” IEEE Robotics and Automations Letters, Vol. 2, Issue 2, 2017.

[32] L. C. Lai, C. C. Yang, and C. J. Wu,“Time-Optimal Control of a Hovering Quad-Rotor Helicopter,” J. of Intelligent and Robotic Systems, 2006.

[33] R. Ritz et al., “Quadrotor Performance Benchmarking Using Optimal Control,” IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, 2011.

[34] M. Hehn, R. Ritz, and R. D'Andrea, "Performance benchmarking of quadrotor system using time-optimal control," Auton Robot, 2012.

[35] T. Grüning, A. Rauh, and H. Aschemann, "Feedforward Control Design for a Four-Rotor UAV using Direct and Indirect Methods," 17th Int. Conf. on Methods & Models in Automation & Robotics, 2012.

[36] J. Zink and B. Lovelace, "Unmanned Aerial Vehicle Bridge Inspection Demonstration Project," Minnesota Department of Transportation, Research Project Final Report, 2015.

[37] D. W. Mellinger, "Trajectory Generation and Control for Quadrotors," PhD Thesis, University of Pennsylvania, 2012.

[38] D. Mellinger, N. Michael, and V. Kumar, "Trajectory Generation and Control for Precise Aggressive Maneuvers with Quadrotors," The 12th International Symposium on Experimental Robotics, 2014.

[39] G. Klein and D. Murray, "Parallel Tracking and Mapping for Small AR Workspaces," Proceeding of International Symposium on Mixed and Augmented Reality, 2007.

[40] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM: a Versatile and Accurate Monocular SLAM System", IEEE Transactions on Robotics, 2015.

[41] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-Scale Direct Monocular SLAM," European Conference on Computer Vision, 2014.

[42] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast Semi-Direct Monocular Visual Odometry," IEEE International Conference on Robotics and Automation, 2014.

[43] E. S. Jones and S. Soatto, "Visual-inertial navigation, mapping and localization: A scalable real-time causal approach," The International Journal of Robotics Research, Vol. 30, No. 4, 2011.

[44] J. Kelly and G. S. Sukhatme, "Visual-inertial sensor fusion: Localization, mapping and sensor-to-sensor self-calibration," International Journal of Robotics Research, Vol. 30, No. 1, 2011.

[45] J. A. Hesch, D. G. Kottas, S. L. Bowman, and S. I. Roumeliotis, "Camera-IMU-based localization: Observability analysis and consistency," International Journal of Robotics Research, Vol. 33, No. 1, 2014.

[46] S. Shen, N. Michael, and V. Kumar, "Tightly-coupled monocular visual-inertial fusion for autonomous flight of rotorcraft MAVs," IEEE International Conference on Robotics and Automation, 2015.

[47] S. Shen, Y. Mulgaonkar, N. Michael, and V. Kumar, "Vision-based state estimation and trajectory control towards high-speed flight with a quadrotor," Robotics: Science and Systems, 2013.

[48] H. H. Platt, "The Helicopter: Propulsion and Torque," J. of Aero. Sci., Vol. 3, No. 11, 1936.

[49] H. G. Küssner, "Helicopter Problems," T.M. No. 827, NACA, 1937.

[50] A. Betz, "The ground effect on Lifting Propellers," T.M. No. 836, NACA, 1937.

[51] H. Focke, "The Focke Helicopter," T.M. No. 858, NACA, 1938.

[52] H. H. Heyson, "Theoretical study of the effect of ground proximity on the induced efficiency of helicopter rotors," T.M. X-71951, NASA, 1977.

[53] N. Kang and M. Sun, "Simulated flowfields in near-ground operation of single- and twin-rotor configurations," Journal of Aircraft, 37(2), 2000.

[54] B. Ganesh and N. Komerath, "Unsteady Aerodynamics of Rotorcrafts in Ground Effect," American Institute of Aeronautics and Astronautics, Fluid Dynamics Meeting, 2004.

[55] D. P. Pulli, "A study of helicopter aerodynamics in ground effect," PhD Dissertation, Ohio State University, 2006.

[56] J. W. Lim, K. W. McAlister, and W. Johnson, "Hover performance correlation for full-scale and model-scale coaxial rotors," Journal of the American Helicopter Society, 54(3), 2009.

[57] P. M. Basset and A. Omari, "A rotor vortex wake model for helicopter flight mechanics and its application to the prediction of the pitch-up phenomenon," 25th ERF Forum, 1999.

[58] G. Lebouar, M. Costes, A. Leroy-Chesneau, and P. Devinant, "Numerical simulations of unsteady aerodynamics of helicopter rotor in maneuvering flight conditions," Aero. Science and Tech., 8(1), 2004.

[59] N. Kroll, B. Eisfeld, and H. M. Bleeke, "The Navier-Stokes code FLOWer," Notes on Numerical Fluid Mechanics, Vol. 71, 1999.

[60] A. Fillipone, R. Bakker, P. M. Basset, B. Rodriguez, R. Green, B. Kutz, F. Bensing, and A. Visingardi, "Rotor Wake Modelling in Ground Effect Conditions," European Rotorcraft Forum, 2011.

[61] R. Mahony and T. Hamel, "Adaptive Compensation of Aerodynamics Effects during Takeoff and Landing Manoeuvres for a Scale Model Autonomous Helicopter," European Journal of Control, 2001.

[62] N. Guenard, T. Hamel, and L. Eck, "Control Laws For The Tele Operation Of An Unmanned Aerial Vehicle Known As An X4-flyer," IEEE/RSJ International Conference on Intelligent Robots and Systems, 2006.

[63] D. Lee, H. J. Kim, and S. Sastry, "Feedback Linearization vs. Adaptive Sliding Mode Control for a Quadrotor Helicopter," International Journal of Control, Automation, and systems, 2009.

[64] D. Lee, T. Ryan, and H. J. Kim, "Autonomous Landing of a VTOL UAV on a Moving Platform Using Image-based Visual Servoing," IEEE International Conference on Robotics and Automation, May, 2012.

[65] M. Mirzaei, F. S. Nia, and H. Mohammadi, "Applying Adaptive Fuzzy Sliding Mode Control to an Underactuated System," 2nd International Conference on Control, Instrumentation and Automation, October, 2011.

[66] K. Nonaka and H. Sugizaki, "Integral Sliding Mode Altitude Control for a Small Model Helicopter with Ground Effect Compensation," American Control Conference, June, 2011.

[67] T. Ryan and H. J. Kim, "'Modelling of Quadrotor Ground Effect Forces via Simple Visual Feedback and Support Vector Regression," AIAA Guidance, Navigation, and Control Conference, August, 2012.

[68] C. Powers, D. Mellinger, A. Kushleyev, B. Kothmann, and V. Kumar, "Influence of Aerodynamics and Proximity Effects in Quadrotor Flight," Experimental Robotics: The 13th International Symposium on Experimental Robotics, 2013.

[69] I. Sharf, M. Nahon, A. Harmat, W. Khan, M. Michini, N. Speal, M. Trentini, T. Tsadok, and T. Wang, "Ground Effect Experiments and Model Validation with Draganflyer X8 Rotorcraft," International Conference on Unmanned Aircraft Systems, May, 2014.

[70] P. E. Frandsen and K. J. H. B. NIelsen and O. Tingleff, Unconstrained Optimizations, 2004

[71] O. Nells, *Nonlinear System Identification*, Springer-Verlag, 2001

[72] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer series in operations research, 1999

[73] L. Hasdorff, "Gradient Optimization and Nonlinear Control," John Wiley & Sons, 1976.

[74] Shampine, L.F., M.W. Reichelt, and J. Kierzenka, "Solving Boundary Value Problems for Ordinary Differential Equations in MATLAB with bvp4c," available at $http : //www.mathworks.com/bvp_tutorial$

[75] I. J. Perez-Arriaga, G. C. Verghese, and F. C. Scheweppe, "Selective modal analysis with applications to electric power systems, part I: Heuristic introduction," IEEE Transactions on Power Apparatus and Systems, 1982.

[76] G. Rodriguez-Gomez, P. Gonzalez-Casanova, and J. J. Martinez-Carballido, "Computing general companion matrices and stability regions of multirate methods," International Journal for Numerical Methods in Engineering, 2004.

[77] D. S. Shim and C. K. Yang, "Optimal Configuration of Redundant Inertial Sensors for Navigation and FDI Performance," Sensors, 2010.

[78] H. D. Escobar, "Geometrical Configuration Comparison Of Redundant Inertial Measurement Units," Master's Report, University of Texas at Austin, 2010.

[79] B. Xiao and S. Yin, "A New Disturbance Attenuation Control for Quadrotor Unmanned Aerial Vehicles," IEEE Transactions on Industrial Informatics, 2017.

[80] C. D. McKinnon and A. P. Schoeling, "Unscented External Force and Torque Estimation for Quadrotors," IEEE International Conference on Intelligent Robots and Systems, 2016.

[81] S. Aich, C. Ahuja, T. Gupta, and P. Arulmozhivarman, "Analysis of Ground Effect on Multi-Rotors," International Conference on Electronics, Communications and Computational Engineering, 2014.

[82] A. Matus-Vargas, G. Rodriguez-Gomez, J. Martinez-Carranza, A. Muñoz-Silva, "Numerical Optimization Techniques for Nonlinear Quadrotor Control," International Conference on Unmanned Aircraft Systems, Miami, Florida, 2017.