Quadrotor Thrust Vectoring Control with Time Optimal Trajectory Planning in Constant Wind Fields

João Paulo da Rocha Silva
joaopaulo@tecnico.ulisboa.pt
Instituto Superior Técnico, Universidade de Lisboa, Portugal
March 2017

***Abstract*—This paper proposes a control strategy to follow time optimal trajectories planned to visit a given set of waypoints in windy conditions. The aerodynamic effects of quadrotors are investigated, with emphasis on blade flapping, induced and parasitic drag. An extended method to identify all the aerodynamic coefficients is developed, and their influence on the performance is analyzed. A computationally efficient three steps approach is suggested to optimize the trajectory, by minimizing aerodynamic drag and jerk while still guaranteeing near optimal results. The derived smooth trajectory is compared with standard discrete point to point followed by low-pass filtering trajectories, showing energetic improvements in thrust and reductions in Euler angles aggressiveness. By exploiting the non-linear aerodynamic effects and using *a priori* trajectory information, a thrust vectoring controller is designed and compared with a classic PID controller, showing an increase in performance by reducing the tracking delay and extending the flight envelope.**

***Index Terms*—Quadrotor, Control, Thrust Vectoring, Wind, Optimal Trajectory, Drag Effects, Minimum Jerk, Waypoint Sequencing.**

I. Introduction

Quadrotors are a popular type of Multicopter Unmanned Aerial Vehicles (UAVs) used for applications in which fast and aggressive trajectories on a three dimensional space are required [10]. Application such as surveillance, package delivery or plant monitoring reflect the skills that quadrotors possess to follow predefined trajectories [11]. However, due to a variety of limitations such as maximum thrust, reduced energetic capacity or bounded bank angles, their performance is not efficient and worsens when the wind is present. Planning the trajectory and designing the controller to include information about the wind to increase the quadrotors performance becomes then a natural solution [13].

The goal of this work is to plan the time optimal trajectory for quadrotors in the presence of constant wind fields. The trajectory is formulated such that m predefined desirable waypoints are visited, without restrictions in the visiting sequence. The total trajectory time has to be minimized while still maintaining feasibility. Wind influence on the quadrotors dynamics forces the inclusion of aerodynamic effects in the trajectory generation phase and to understand those effects a literature study is required. To control the quadrotor, a state of the art controller needs to be designed, which should minimize the aerodynamic effects and follow the aggressive trajectory efficiently.

Trajectory generation for quadrotors has been studied in several approaches. Smooth point to point trajectories have already been studied with Non Linear Programming (NLP) [14] and using the Pontryagin's Minimum Principle (PMP) [10, 22]. For multiple waypoints the usual considered approaches are Sequential Convex Programming (SCP) [3] or Quadratic Programming (QP) [21, 26], extended to Multi Integers Linear Programming (MILP) by [20]. Those approaches aim at minimizing accelerations, jerks or snaps, but disregard wind and assume fixed traveling times. [13] is among the few who include wind in the trajectory, but plans only point to point trajectories. [6] achieve the goal of minimizing the time for multiple waypoints, but fail to the include wind.

Regarding quadrotor control, the typical controllers available are based on classic PID theory [11, 29, 18]. In these approaches the wind is disregarded and treated as a disturbance to be further rejected by the controller. More complex approaches that include aerodynamic effects, such as Feedback Linearization [28] or Integral-Backstepping [2] exist, but are only implemented in simulations. In a further step, [19] considers the trajectory generation in the controller design, using a thrust vectoring approach, but neglects wind. The same approach is used by [23] but without the trajectory information. Extending the work of [19] and [23] seems to be an engaging approach that fulfills the controller requisites.

A general method to achieve the proposed goal is thus absent in literature. Thus, a new approach is proposed, that consists in three linked aspects. The initial aspect of the approach is to address an aerodynamic study in order to understand the three most important effects that influence the quadrotors dynamics: blade flapping, induced and parasitic drag. To complement the study, a contribution is made with an extended method to identify the drag terms and an experiment is performed to validate their influence on the controller performance. The next aspect of the approach is the definition and construction of the time optimal trajectory. The trajectory is defined to be as fast as possible while maintaining the velocity with respect to (w.r.t.) the wind below a certain limit such that the wind effects are diminished the most. A method to plan the trajectory is proposed and divided in three steps, so that the process is structurally fast, saving computational time while still guaranteeing near optimal results. The first step is to determine the time optimal sequence of waypoints using a heuristic search algorithm. The second step is to determine the optimal trajectory throughout the waypoint sequence by solving minimizing the jerk. The third step is to ensure that the velocity respects the mentioned limit. The last aspect of the approach is to design a controller optimized to explore the *a priori* information obtained about the wind and the trajectory. A cascade thrust vectoring controller is proposed and compared with a typical PID controller, showing an increase in performance, in particular by reducing tracking delay.

This work is structured as follows. Section II describes the wind and associated aerodynamic effects, the time optimal trajectory is derived in Section III and Section IV presents the quadrotor model and the proposed controller. Section V shows the results, while the conclusion is presented in Section VI.

II. Wind and Aerodynamic Effects

This Section aims at understanding the wind and associated aerodynamic effects in the quadrotor. It starts with a literature review in Section II-A and the following Sections II-B-II-E describe individually each effect with importance for this work.

A. Literature Review

Detailed studies on aerodynamic effects are already available in the literature for the case of almost all aircraft types. Even for helicopters, which can be seen as a parent of quadrotors, precise studies have been performed, such as [25] and [16]. In the latest, the aerodynamic properties of rotating blades are studied in detailed. However, for the case of small and more recent quadrotors, the literature is more scarce. [11] are one of the firsts to consider aerodynamic effects into the model of the quadrotor. There, drag effects such as the total thrust variation, blade flapping and airflow disruption are considered. Later on, [12] also considers the first two aspects and [29] modeled the total drag as a lumped factor linear w.r.t. to the airspeed. This lumped factor as also been described by [17, 1]. [4] are the firsts who pursue a detailed explanation of all aerodynamic effects that affect the quadrotors dynamics, and consider blade flapping, induced drag, translational drag, profile drag, parasitic drag and also others such as ground effect and vertical descent.

B. Blade Flapping

Blade Flapping is a phenomena that occurs due to the flexibility of the rotor blades. When in translational movement, the tip advancing and retrieving blades, for each rotor, will not have the same velocity w.r.t. the airstream. This causes the advancing blade to flex upwards, while the retrieving blade flexes downwards. In the literature, it is mentioned that this behavior creates roll and pitch moments at the blade root, and shifts the thrust vector $\mathbf{T}$ by a blade flapping angle β. The angle β can be decomposed in two components, parallel and perpendicular to the airstream, as

$$\beta_i^{\parallel} = -\frac{\mu_i a_1}{1-\frac{1}{2}\mu_i^2}, \qquad \beta_i^{\perp} = -\frac{\mu_i a_2}{1+\frac{1}{2}\mu_i^2}, \tag{II.1}$$

where $\mu_i = \frac{\mathbf{v}_\infty^P}{\bar{\omega}_i r}$ is the advancing ratio, i.e., the ration between the airspeed projected in the rotor plane (similar to all rotors) and the rotor linear velocity. The constants a_1 and a_2 depend on blade properties, with $a_2 \ll a_1$, and one can consider that $\frac{1}{2}\mu_i^2 \ll 1$ due to the high rotation speed of the rotors [4]. Being so, most authors who describe blade flapping as a drag force due to aerodynamic effects [17, 1] represent it as a function of thrust magnitude, rotor linear velocity and airspeed. Recently, [23] uses the properties of blade theory to better estimate some states of the state vector, including blade flapping in the quadrotor model. There, blade flapping is written differently by using the thrust as a function of the rotor angular speed with $T_i = k_T \bar{\omega}_i^2$, leading to

$$\mathbf{D}_{flap,i} = \bar{\omega}_i \mathbf{A}'_{flap} \mathbf{v}_\infty^P \tag{II.2}$$

with $\mathbf{A}'_{flap}$ a constant matrix given by

$$\mathbf{A}'_{flap} = \begin{pmatrix} \frac{a_1 k_T}{r} & \frac{a_2 k_T}{r} & 0 \\ -\frac{a_2 k_T}{r} & \frac{a_1 k_T}{r} & 0 \\ 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} a'_1 & a'_2 & 0 \\ -a'_2 & a'_1 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

C. Induced Drag

As discussed before, the blades are flexible, meaning that they can only bend to a certain angle β when a translational movement w.r.t. the airstream exists. Nevertheless, the blades still have stiff properties, meaning that when they produce produce lift, they produce an associated and proportional drag called induced drag. To model it, a linear drag coefficient can be introduced as

$$\mathbf{D}_{ind,i} = T_i \mathbf{A}_{ind} \mathbf{v}_\infty^P, \tag{II.3}$$

with $\mathbf{A}_{ind}$ constant defined by the induced drag coefficient as

$$\mathbf{A}_{ind} = \begin{pmatrix} k_{ind} & 0 & 0 \\ 0 & k_{ind} & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

D. Lumped Drag Coefficient

The formulas obtained for blade flapping and induced drag are w.r.t. one rotor. In the final model the four rotors should be added. For the induced drag it corresponds to linearly add the thrust of each motor, as $\sum_{i=1}^4 T_i = T$. In [18], the sum of all rotation speeds is found to be approximately constant during flights of near hovering thrust, leading to the blade flapping drag as

$$\mathbf{D}_{flap} = T_h \mathbf{A}''_{flap} \mathbf{v}_\infty^P \tag{II.4}$$

with $T_h \mathbf{A}''_{flap} \approx \sum_{i=1}^4 \bar{\omega}_i \mathbf{A}'_{flap}$ a constant matrix and T_h the hovering thrust. When comparing with the drag caused by blade flapping and induced drag, it can be seen that the expressions are similar. Thus, it is possible to lump these two drag forces into only one, and obtain a drag force due to the mixed flexibility and stiffness of the blade propellers as

$$\mathbf{D}_{bla} = T \mathbf{A}_{bla} \mathbf{v}_\infty^P \tag{II.5}$$

with $\mathbf{A}_{bla}$ constant defined by the lumped drag coefficient as

$$\mathbf{A}_{bla} = \begin{pmatrix} a_{bla} & 0 & 0 \\ 0 & a_{bla} & 0 \\ 0 & 0 & 0 \end{pmatrix} \approx \begin{pmatrix} \frac{\sum \bar{\omega}_i a'_1}{T} + k_{ind} & \frac{\sum \bar{\omega}_i a'_2}{T} & 0 \\ -\frac{\sum \bar{\omega}_i a'_2}{T} & \frac{\sum \bar{\omega}_i a'_1}{T} + k_{ind} & 0 \\ 0 & 0 & 0 \end{pmatrix}. \tag{II.6}$$

E. Parasitic Drag

Parasitic Drag is caused by the non-lifting surfaces of the quadrotor in a translational movement. It must be considered when high velocities, usually greater that 10 [m/s] [4], are in question, as it is proportional to the square of the velocity and the parasitic drag coefficient by

$$\mathbf{D}_{par} = k_{par} \|\mathbf{v}_\infty\| \mathbf{v}_\infty. \tag{II.7}$$

In environments where the wind and ground speeds are opposite, relatively big airspeeds will appear. Thus, the parasitic drag is of utmost importance in the trajectory generation phase, providing information on which direction to follow in order to minimize the global drag forces to optimize the trajectory.

III. Time Optimal Trajectory

The problem addressed in this Section is to determine the time optimal trajectory that covers multiple waypoints. The problem is defined by m waypoints distributed around the quadrotor, and there are no restrictions either in time or space. Nevertheless, the objective is to minimize the trajectory time and thus some inherent restrictions have to be weighed, which are mainly imposed by the drag forces that where studied in Section II and by the quadrotor dynamics so that the controller to be designed in Section IV follows the time optimal trajectory in an efficient manner.

In order to solve the time optimal problem, two approaches can be considered. One approach is to consider the problem as a global one, i.e., to put all objectives and constraints into a global optimization problem. However, this optimization problem would be too complex to process within an acceptable time for on-line applications. Therefore, a second approach that separates the problem into three smaller problems, or steps, will be considered.

The first smaller problem is to estimate the optimal sequence of waypoints and it will be addressed in Section III-A. To do so, geometric trajectory definitions can be used, including constraints related with wind, but not including the vehicles dynamics. The second smaller problem, solved in Section III-B, is to determine the optimal trajectory between each of the sequential waypoints, and can be solved using a Quadratic Programming (QP) approach. The third smaller problem of the time optimal trajectory planning is to adapt the QP problem solution so that the airspeed magnitude requirement is fulfilled, and it is covered in Section III-C.

A. Optimal Sequencing

When determining the optimal trajectory sequence a geometric approach will be used that includes wind information. Although this approach can be considered simple, it can critically reduce the computational time for this step in the trajectory generation process, and lend time to the more complex one. If this step is not considered, instead of having just one second smaller problem, there would be $m!$, increasing factorially the processing time. This would push the selecting process towards the final third step.

The possible trajectories are thus restrained to lines segments, circles and splines [5] or to B-splines [7]. Another geometrical approach is to consider vector fields such as in [30]. Since the goal lies in selecting the optimal sequence, the complexity of the trajectory does not need to be significant, and straight lines will be used. In Section III-A1 it will be seen that in constant wind fields a straight line is the correct approach and in Section III-A2 the solution of the optimal sequencing problem will be obtained.

1) The Zermelo's Problem: Line segments are a direct geometric solution but they can also be considered as a simplification of the Zermelo's problem [8]. In the original Zermelo's problem a sea current (analogue to wind velocity) pushes the ship directional velocity. The current is dependent on the ship position and the trajectory is modeled in state-space formulation as

$$f(x,y,u,v,\chi,t) = \begin{pmatrix} \dot{x}(t) \\ \dot{y}(t) \end{pmatrix} = \begin{pmatrix} V\cos(\chi) \\ V\sin(\chi) \end{pmatrix} = \begin{pmatrix} V_\infty \cos(\pi) + u(x,y) \\ V_\infty \sin(\pi) + v(x,y) \end{pmatrix} \tag{III.1}$$

where V is the constant velocity of the ship. The states x and y are the position that defines part of the trajectory, while χ defines the course angle. The inputs are the current velocities u and v in the x and y direction respectively. This theory was firstly introduced in the quadrotors field by [13] but not in constant wind fields. Using the Pontryagin's Minimum Principle [24] to optimize the trajectory time T the cost function becomes

$$\min T = \min \int_0^T 1 \ \mathrm{d}t. \tag{III.2}$$

In the case that the current (or wind) is constant then it can be proven that the solution of Equation III.2 results in a constant χ, showing that the optimal trajectories are in fact straight lines. With m waypoints, there are $m!$ possible waypoint sequences. For each possible sequence, the χ_i, V_i and T_i are determined. It is evident that depending on the geometry of the waypoints and on the wind velocity, different trajectory times will result.

2) The Traveling Salesman Problem: The Traveling Salesman Problem (TSP) is a well known NP-hard problem in the Artificial Intelligence field [15]. It aims at determining the optimal sequence between m cities that a salesman has to cover, with no restrictions on each city to visit first, and is analogue to our problem

To solve the TSP several approaches can be considered. One is to evaluate all possible sequences and check which one is the best. Another is to write the problem in an Integer Linear Programming (ILP) formulation and numerically solve it One last approach is to use search algorithms to incrementally extend the sequence until the least costing solution is found.

The first approach works well in both computational time and memory for small m, but when m increases the dimension increases by $\mathcal{O}(m!)$. The ILP formulation as a comparable computational complexity has the search algorithms, but it is slower. Therefore, the problem was solved using search algorithms.

Search Algorithm for the TSP: To solve the optimal sequence an A^* informed search algorithm is used. The heuristic function used is not admissible, since it may happen that it overestimates the true cost of the goal, and is given by

$$h_2(i) = K_{heu}(m-i)T_{ave}, \tag{III.3}$$

where T_{ave} is the average traveling time between two nodes and K_{heu} is a constant, with $0 < K_{heu} \leq 1$. This approach is not optimal, but the solutions are still acceptable. This is due to the fact that in waypoint navigation usually there is a nice distribution of the waypoints. The constant K_{heu} allows to tune the degree of optimality in contrast with the computational times. For a small K_{heu} the importance of the heuristic decreases and the performance gets similar to uniform search. For a big K_{heu} the performance moves towards a greedy best-first search.

B. Quadratic Programming

After determining the optimal sequence of waypoints, the next problem step is to determine the trajectory between each one of the waypoints. It is possible to obtain the trajectory by solving a QP problem [19, 20, 26, 6] or using the Pontryagin's Minimum Principle [10, 22]. Other approaches are discretizing the model [14] or discretizing the trajectory [3]. QP was considered due to its mathematical simplicity which leads to a general capability,

since the solutions can be extrapolated to multiple goals (minimizing the velocity, acceleration, jerk or snap) within the same framework. Moreover, the precision in the polynomials and continuity in the derivatives can be guaranteed as far as wanted.

1) Trajectory Definition: There is a special class of systems, called Differentially Flat systems, for which there is an one-to-one correspondence between trajectories of a set of "flat outputs" and the full state space and inputs. This means that the trajectory can be defined in output space, and then mapped algebraically to the state and input space. These type of systems were introduced in [9] and are well suited for trajectory definition and generation. For quadrotors the flat outputs are usually [30, 19] chosen as

$$\mathbf{r}_T(t) = [\ r_{T_x}(t) \quad r_{T_y}(t) \quad r_{T_z}(t) \quad \psi_T(t)\ ]^\top. \tag{III.4}$$

A proof that quadrotors are differentially flat can be found in [30], without considering aerodynamic drag. Considering a general direction on $\mathbf{r}_T(t)$, furthermore referred as $\sigma_T(t)$, then m trajectories between the origin and each one of the m waypoints can be defined as a n order time polynomial such that

$$\sigma_T(t) = \begin{cases} c_{10} + c_{11}t_1 + c_{12}t_1^2 + \cdots + c_{1n}t_1^n & 0 \le t_1 \le T_1 \\ c_{20} + c_{21}t_2 + c_{22}t_2^2 + \cdots + c_{2n}t_2^n & 0 \le t_2 \le T_2 \\ \vdots & \\ c_{m0} + c_{m1}t_m + c_{m2}t_m^2 + \cdots + c_{mn}t_m^n & 0 \le t_m \le T_m \end{cases} \tag{III.5}$$

and the derivatives of the time position given by

$$\frac{\mathrm{d}^k \sigma_{T_i}(t_i)}{\mathrm{d}t^k} = \sum_{j=0}^{n-k} \frac{(k+j)!}{j!} c_{i(k+j)} t_i^j \tag{III.6}$$

2) Creating the Quadratic Programming Optimization Problem: The QP optimization problem is formulated to minimize or maximize the cost J of the vector $\mathbf{c}$ as

$$\begin{aligned} \min\ J(\mathbf{c}) &= \frac{1}{2}\mathbf{c}^\top \mathbf{H}\mathbf{c} + \mathbf{f}^\top \mathbf{c} \\ \text{subjected to } \mathbf{A}_{in}\mathbf{c} &\le \mathbf{b}_{in} \\ \text{and } \mathbf{A}_{eq}\mathbf{c} &= \mathbf{b}_{eq} \end{aligned} \tag{III.7}$$

where $\mathbf{H}$ is a symmetrical matrix reflecting the quadratic form of the problem and $\mathbf{f}$ is a vector reflecting the linear one. The objective function can be formulated as a function of the k derivative of the position [19, 20], a sum of different derivatives [26] or the Boor control points for the B-Spline [6]. If the objective is to minimize the power of a general derivative of the position then it can be formulated as

$$\sigma_T^*(t) = \min \int_0^T \sum_{i=1}^{4} K_i \left(\frac{\mathrm{d}^k \sigma_{T_i}(t)}{\mathrm{d}t^k} \right)^2 \mathrm{d}t \tag{III.8}$$

with K_i a constant to turn the integral dimensionless. Since the trajectories are decoupled in all directions, the above can be seen as four different optimization problems, with $\mathbf{c}$ the vector with the concatenation of the referred c_{ij}. The waypoints are related by imposing constraints in the continuity up until the p derivative of the position, constraints to be imposed by $\mathbf{A}_{eq}$ and $\mathbf{b}_{eq}$. Maximum or minimum values for acceptable velocities, accelerations, jerks or snaps can be imposed using $\mathbf{A}_{in}$ or $\mathbf{b}_{in}$. However, minimizing a derivative of the position is already indirectly imposing acceptable trajectories, and in this work the inequality constraint is neglected.

3) What Derivative to Minimize?: Minimizing aerodynamic drag is the same as minimizing the power of the velocity w.r.t. the wind. In constant wind fields, it can be proven that this is the same as minimizing the overall velocity, which is contradictory to achieve the goal of minimizing the trajectory time, and thus the first position derivative was not considered. Minimizing the following position derivatives powers, which have influence on the quadrotors dynamics (see Equation (IV.7)), is then proposed.

- Acceleration is the simplest, but most naive to define as the goal, since it will imply the less possible thrust, thus constraining excessively the aggressiveness of the trajectory. Smooth trajectories are desirable, but with some aggressiveness to explore the time optimal possible trajectory, which is acceptable due to the thrust vectoring controller to be proposed in Section IV-D.

- Jerk is a better representative of the aggressiveness of the true system inputs [22] and, like the acceleration, has a direct link with thrust. Moreover [10] affirm that maintaining constraints on the acceleration and jerk leads to a continuous thrust during the maneuver, which is then supported by [6] when affirming that constraints on jerk are necessary for a smooth trajectory.

- Snap trajectories have also been proven effective to generate quadrotor trajectories [26], due to the linkage in the motors commands and body rate derivatives. The same says [21] and supports it with a study of human movement.

In this work, minimizing the power of jerk was defined as the goal, since it can balance all the aspects discussed before. Moreover, minimizing the jerk relates with minimizing the variation in the acceleration that is linked to the Euler angles caused by drag.

4) Degree of the Polynomial and Constraints: The degree of the polynomial was set to $n = 5$ so that there are still sufficient coefficients when minimizing the jerk power. Continuity constraints were imposed until $p = 2$, to guarantee continuity at least until the second derivative of the position, the acceleration.

5) Optimal Times: The time segments T_1, T_2 to T_m are constants and the optimization process is solved using the expected times obtained when solving the TSP problem. However, those times can be adapted, thus reducing the overall cost J while maintaining the total trajectory time T. Thus, an iterative process to optimize the time segments with a gradient descent method using a backtracking line search was used [19].

C. Constraint on the Maximum Velocity

At this point, the total time optimal trajectory time T is an estimate given by solving the TSP, meaning that the resulting maximum velocity of the QP problem can be greater or smaller than V_{lim} in some time intervals. Due to the non dimensional spatial property included by construction in the QP formulation, a new loop can be performed, by incrementally giving or taking more time to T, without the need to solve the QP problem again. By doing this, the velocity is guaranteed to respect V_{lim} so that the parasitic drag is minimized.

IV. Thrust Vectoring Controller

From the two previous Sections II and III it is known that the quadrotor dynamics are linked to aerodynamic effects and trajectory dynamics. In this Section, the model of the quadrotor will be obtained, in Section IV-A, and the quadrotor used in this work will be presented, in Section IV-B. Paparazzi UAV, which uses a PID controller, is an open-source platform for UAV development that will be used as framework for this work, and will be analyzed in Section IV-C. However, the time optimal trajectory information and the aerodynamic effects can be used in an optimal way using a proposed thrust vectoring controller approach, to be following presented in Section IV-D.

A. Model of the Quadrotor

The configuration of a typical quadrotor consists of four rigid blade propeller motors mounted in a "+" or "×" pairwise symmetrical counter-rotating fashion, and attached to the quadrotor rigid body as it can be seen in Figure IV.1.

1) Frames of Reference and other Formalisms: The quadrotor center of mass is B with mass m and inertia matrix $\mathbf{J}$. The inertial world frame of reference is $\mathcal{W} = \{O, \mathbf{x}^W, \mathbf{y}^W, \mathbf{z}^W\}$ and the frame attached to the quadrotor, the body frame of reference, is $\mathcal{B} = \{B, \mathbf{x}^B, \mathbf{y}^B, \mathbf{z}^B\}$. To represent the relative frame orientation a rotation matrix from $\mathcal{B}$ to $\mathcal{W}$ is used, defined as $\mathbf{R}_B^W = \mathbf{R}_z\mathbf{R}_y\mathbf{R}_x = \mathbf{R}$. The inverse rotation is $\mathbf{R}_W^B = (\mathbf{R}_B^W)^{-1} = (\mathbf{R}_B^W)^\top$. $\mathbf{R}_B^W$ is composed of three consecutive $Z-Y-X$ rotations of the Euler angles yaw, pitch and roll, such that

$$\mathbf{R}_B^W = \begin{pmatrix} c\theta c\psi & s\theta s\phi c\psi - c\phi s\psi & s\theta c\phi c\psi + s\phi s\psi \\ c\theta s\psi & s\theta s\phi s\psi + c\phi c\psi & s\theta c\phi s\psi - s\phi c\psi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{pmatrix}. \quad \text{(IV.1)}$$

The canonical basis of $\mathbb{R}^3$ is $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$. The quadrotor position, velocity and acceleration, in the inertial frame, are $\mathbf{r}$, $\dot{\mathbf{r}}$ and $\ddot{\mathbf{r}}$ while the wind velocity is $\mathbf{v}_w$. The angular velocity, measured in the body frame, is $\boldsymbol{\omega} = \begin{bmatrix} p & q & r \end{bmatrix}^\top$. The Euclidean norm is $\|\mathbf{a}\| := \sqrt{a_1^2 + a_2^2 + a_3^2}$, the dot-product is $\mathbf{a} \cdot \mathbf{b} := a_1b_1 + a_2b_2 + a_3b_3$ and the notation $\mathbf{a}_\times\mathbf{b} := \mathbf{a} \times \mathbf{b}$ relates to the skew-symmetric matrix associated with the cross-product.

2) Motor Dynamics: The motors are symmetrically attached to B at a distance of $\mathbf{l}_i = \begin{pmatrix} l_{x,i} & l_{y,i} & 0 \end{pmatrix}^\top$. The motors dynamics are assumed relatively fast when compared to the rigid body dynamics and aerodynamics effects, so they can be neglected. Considering the body frame $\mathcal{B}$. each rotor rotating at $\bar{\omega}_i$ generates a thrust $\mathbf{F}_{T,i} = T_i\mathbf{e}_3$, with $T_i = k_T\bar{\omega}_i^2$, and an aerodynamic torque $\mathcal{T}_i = \lambda_i k_\tau \bar{\omega}_i^2 \mathbf{e}_3$. The thrust and torque constants, k_T and k_τ, depend on blade properties and the λ_i constant depends on the direction of rotation of the blade ($\lambda_i = 1$ for clockwise and $\lambda_i = -1$ for counterclockwise).

3) Dynamic Model: The dynamic model of the quadrotor, which represents the Newton-Euler equations, is given by

$$\begin{cases} m\ddot{\mathbf{r}} = \mathbf{R}\sum_{i=1}^4 \mathbf{F}_i - mg\mathbf{e}_3 + \mathbf{F}_{aero} \\ \dot{\mathbf{R}} = \mathbf{R}\boldsymbol{\omega}_\times \\ \mathbf{J}\dot{\boldsymbol{\omega}} = -\boldsymbol{\omega}_\times\mathbf{J}\boldsymbol{\omega} + \sum_{i=1}^4(\mathcal{T}_i + \mathbf{l}_i \times \mathbf{F}_i) + \mathcal{T}_{aero} \end{cases} \quad \text{(IV.2)}$$

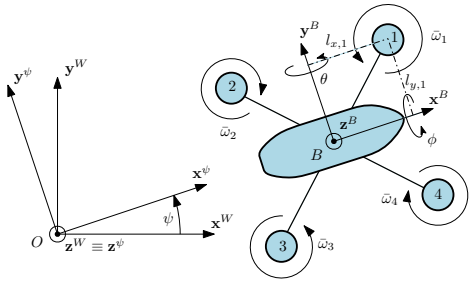

Figure IV.1: Top view of a typical quadrotor in "×" configuration with reference frames scheme.

All forces acting on the quadrotor, except for $\mathbf{F}_{aero}$, are the sum of the thrust and blade flapping and induced drag, lumped in one drag force as in Equation (II.5). Thus

$$\sum_{i=1}^4 \mathbf{F}_i = \sum_{i=1}^4(\mathbf{F}_{T,i} + \mathbf{D}_{bla,i}) = T\mathbf{e}_3 + \sum_{i=1}^4 T_i\mathbf{A}_{bla}\mathbf{v}_{\infty,i}^P. \quad \text{(IV.3)}$$

The aerodynamic force is the parasitical drag force given by

$$\mathbf{F}_{aero} = \mathbf{D}_{par} = k_{par}\|\mathbf{v}_\infty\|\mathbf{v}_\infty = k_{par}\|\dot{\mathbf{r}} - \mathbf{v}_w\|(\dot{\mathbf{r}} - \mathbf{v}_w) \quad \text{(IV.4)}$$

The torque terms are

$$\sum_{i=1}^4(\mathcal{T}_i + \mathbf{l}_i \times \mathbf{F}_i) = \mathcal{T} + \sum_{i=1}^4 T_i\mathbf{l}_i \times \mathbf{A}_{bla}\mathbf{v}_{\infty,i}^P = \mathcal{T}. \quad \text{(IV.5)}$$

Assuming that the external aerodynamic forces cause no torque on the quadrotor then

$$\mathcal{T}_{aero} \approx 0 \quad \text{(IV.6)}$$

Finally, the full system can be written, with manipulated inputs given by the rotors thrust and torque as

$$\begin{cases} m\ddot{\mathbf{r}} = T\mathbf{R}\mathbf{e}_3 - mg\mathbf{e}_3 + T\mathbf{R}\mathbf{A}_{bla}\mathbf{R}^\top(\dot{\mathbf{r}} - \mathbf{v}_w) + k_{par}\|\dot{\mathbf{r}} - \mathbf{v}_w\|(\dot{\mathbf{r}} - \mathbf{v}_w) \\ \dot{\mathbf{R}} = \mathbf{R}\boldsymbol{\omega}_\times \\ \mathbf{J}\dot{\boldsymbol{\omega}} = -\boldsymbol{\omega}_\times\mathbf{J}\boldsymbol{\omega} + \mathcal{T} \end{cases} \quad \text{(IV.7)}$$

where the thrust magnitude T and torque vector $\mathcal{T}$ relation with the angular velocity of the blade propellers is represented as

$$\begin{pmatrix} T \\ \mathcal{T} \end{pmatrix} = \begin{pmatrix} k_T & k_T & k_T & k_T \\ k_T l_y & k_T l_y & -k_T l_y & -k_T l_y \\ -k_T l_x & k_T l_x & k_T l_x & -k_T l_x \\ k_\tau & -k_\tau & k_\tau & -k_\tau \end{pmatrix} \begin{pmatrix} \bar{\omega}_1^2 \\ \bar{\omega}_2^2 \\ \bar{\omega}_3^2 \\ \bar{\omega}_4^2 \end{pmatrix}. \quad \text{(IV.8)}$$

B. Parrot Bebop

The quadrotor used in this work is the Parrot Bebop (see Figure IV.2), a commercially available quadrotor[1] widely used at TUDelft as a research platform. It uses a "×" motor configuration and has features that are adequate for the work at hand: resistant structure, also protected by side bumpers to increase safety; lightweight; powerful dual-core CPU; and wide variety of sensors.

[1] http://www.parrot.com/products/bebop-drone/

Figure IV.2: Parrot Bebop with side bumpers.

C. Paparazzi UAV - Reference Generator and PID Framework

The Paparazzi UAV is an open-source platform firstly conceived as a tool for development of standard fixed-wing UAVs[2]. It incorporates a standard reference generator and PID quadrotor controller with whom the performances will be compared. The current controllers (see Figure IV.3) have a traditional separation between the vertical and horizontal movement, and thus two main and independent control loops are usually implemented.

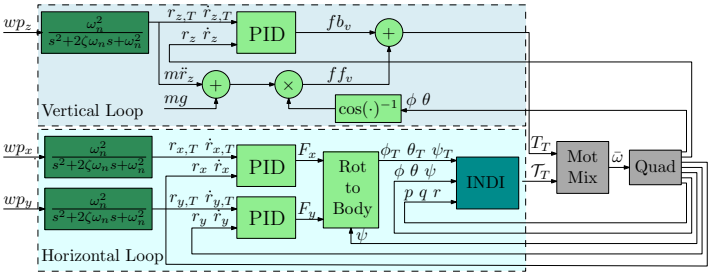

Figure IV.3: Illustration of a standard controller.

1) Control Loops in Paparazzi: Both control loops inputs are discrete values of waypoints. The reference generators create discrete point to point trajectory steps followed by second order low-pass filtering, used so that there are no aggressive requests of velocity and acceleration. The reference generators will be further referred as the standard low-pass filtering case. The vertical and lateral controllers are typical PID controllers, with feedback action for the position and velocity, and feedforward action for the acceleration. The stabilization controller currently implemented uses an Incremental Non-Linear Dynamic Inversion (INDI).

2) Incremental Non-Linear Dynamic Inversion: The INDI controller is an attitude based controller developed by [27] for the Parrot Bebop. It increases the controller robustness as it calculates increments in the control action based solely in the desirable increment in the angular acceleration, avoiding problems with unmodeled dynamics. The inclusion of an optimal trajectory generator and the thrust vectoring controller, together with the INDI is thus a major step towards an optimal quadrotor controller.

D. New Controller Approach

The loops separation will impose a natural oscillatory behavior by the exchange in vertical and horizontal performance. Moreover, in the usual PID controllers there is no information about the aerodynamic effects, manly composed of drag terms, and critically affected by wind. For these reasons, the performance of the current controller is not optimal. Thus, it is proposed the replacement of the first two outer loops by a single one, which weights both the vertical and horizontal movement at the same time. The current reference generator is replaced by the equations discussed in Section III, meaning that there will be optimal trajectories, which minimize the jerk power. *A priori* information of the trajectory is used so that there is no theoretical delay. Moreover, a thrust vectoring implementation allows to determine the desirable orientation of the quadrotor so that the thrust vector is aligned with the desired force. The stabilization controller based on the INDI implementation is also used.

1) Thrust Vectoring Equations: The proposed controller was inspired by [19], but divergences exist. Extending the PID approach to incorporate aerodynamic effects one has

$$\mathbf{F}_T = K_p\mathbf{e}_p + K_{pi}\int_0^{\Delta t}\mathbf{e}_p\ \mathrm{d}t + K_v\mathbf{e}_v + K_a\mathbf{e}_a + mg\mathbf{z}^W + \\ + m\ddot{\mathbf{r}}_T + T\mathbf{R}\mathbf{A}_{bla}\mathbf{R}^\top(\dot{\mathbf{r}} - \mathbf{v}_w) + k_{par}\|\dot{\mathbf{r}} - \mathbf{v}_w\|(\dot{\mathbf{r}} - \mathbf{v}_w) \tag{IV.9}$$

with $\mathbf{F}_T$ the desirable thrust. Equation (IV.9) shows the first divergence with [19], since here the drag terms are accounted, whereas they neglect them. The desirable orientation of the quadrotor $\mathbf{z}^B$ axis is given by

$$\mathbf{z}_T^B = \frac{\mathbf{F}_T}{\|\mathbf{F}_T\|} \tag{IV.10}$$

With the desirable yaw angle one can derive the intermediate $\mathbf{x}^\psi$ orientation of the first reference frame rotation, given by

$$\mathbf{x}_T^\psi = \mathbf{R}_z\mathbf{e}_1 = \begin{bmatrix}\cos(\psi_T) & \sin(\psi_T) & 0\end{bmatrix}^\top \tag{IV.11}$$

The desirable orientation of the quadrotor $\mathbf{y}^B$ axis can be obtained considering the orthogonality property of the axes

$$\mathbf{y}_T^B = \frac{\mathbf{z}_T^B \times \mathbf{x}_T^\psi}{\|\mathbf{z}_T^B \times \mathbf{x}_T^\psi\|} \tag{IV.12}$$

and the desirable orientation of the quadrotor $\mathbf{x}^B$ axis is

$$\mathbf{x}_T^B = \mathbf{y}_T^B \times \mathbf{z}_T^B. \tag{IV.13}$$

Equation (IV.12) is only valid if $\mathbf{z}_T^B \times \mathbf{x}_T^\psi \neq 0$ meaning that the axes are not collinear. The three axes define the desirable rotation matrix in the same way as in Equation (IV.1) as

$$\mathbf{R}_T = \begin{pmatrix}\mathbf{x}_T^B & \mathbf{y}_T^B & \mathbf{z}_T^B\end{pmatrix} \tag{IV.14}$$

and the other desirable Euler angles can finally be obtained using

$$\begin{pmatrix}\phi_T \\ \theta_T\end{pmatrix} = \begin{pmatrix}\operatorname{atan2}\left(\mathbf{R}_{T,32}, \mathbf{R}_{T,33}\right) \\ \operatorname{atan2}\left(-\mathbf{R}_{T,31}, \sqrt{\mathbf{R}_{T,32}^2 + \mathbf{R}_{T,33}^2}\right)\end{pmatrix} \tag{IV.15}$$

Equation (IV.15) shows the other major divergence from [19], since here the angles are computed from the rotation matrix and inputted to the INDI stabilizer. The proposed approach is less subjected to measurement errors, since the tracking of the angles is made directly through the INDI stabilizer, instead of propagating the error throughout a rotation matrix.

Finally, the desirable thrust magnitude can be obtained by projecting the desirable thrust into the quadrotor $\mathbf{z}^B$ axis

$$T_T = \mathbf{F}_T \cdot \mathbf{z}^B \tag{IV.16}$$

[2]http://wiki.paparazziuav.org/wiki/Main_Page

2) Three Stages Cascade Controller: With the equations derived in Section IV-D1, together with the equations derived from Section III and including the INDI stabilizer, the block diagram of the proposed three stages cascade controller can be represented.

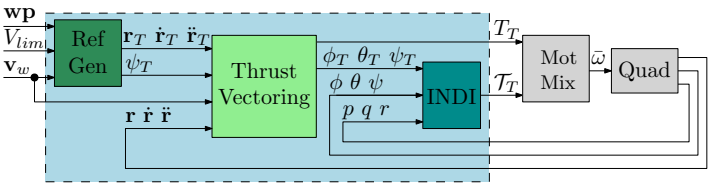

Figure IV.4: Illustration of the proposed Cascade Controller.

V. Results

In this Section the results of the work in hands will be analyzed. In Section V-A, the lumped and parasitic drag coefficients are identified and their influence on the controller is shown in Section V-B. In Section V-C, the time optimal trajectory generation is compared versus a standard reference generator. In Section V-D, the thrust vectoring controller is compared with a PID controller.

A. Identification of the Drag Coefficients

In order to determine the coefficients related with aerodynamic properties of the quadrotor, lumped a_{bla} and parasitic k_{par} drag coefficients defined in Section II, a set of flight tests were done.

1) Proposed Identification Equations: [23] proposed a method to obtain the lumped drag coefficient based on the measures of the acceleration and the velocity in body coordinates. Extending their work, an approach to obtain the second order drag coefficient is proposed. To do so, Equation (IV.7) was used, and the movement of the quadrotor was constrained to the x axis direction, which allows some crucial simplifications. Assuming equilibrium of forces , using the small angles approximation for θ and assuming that $Ta_{bla}\theta << T - mg$ leads to the equation in the x axis as

$$ma_x = mg\theta + mga_{bla}v_x + k_{par}\|v_x\|v_x = mg\theta + D_x \ , \quad \text{(V.1)}$$

showing a direct correspondence between velocity, acceleration and pitch angle that allows to obtain the desired coefficients by means of a statistic fitting. From every measure i of the N collected flight data the specific drag force was determined as

$$d_{x,i} := \frac{D_{x,i}}{m} = a_{x,i} - g\theta_i \ \left[\frac{\text{m}}{\text{s}^2}\right] \quad \text{(V.2)}$$

and a linear Least Squares (LS) fitting was used with cost

$$J = \sum_{i=1}^{N}\left(d_{x,i} - \hat{d}_{x,i}(v_{x,i})\right)^2 = \sum_{i=1}^{N}\left(d_{x,i} - g\hat{a}_{bla}v_{x,i} - \frac{\hat{k}_{par}}{m}\|v_{x,i}\|v_{x,i}\right)^2 . \quad \text{(V.3)}$$

2) Methodology: For the experimental tests, the tracking position was set to be a ramp, resulting in a step in velocity and null acceleration. Since drag forces exist, an expected pitch angle can be computed. Due to the space limitations in the test arena, V_T was varied in the set $V_T \in \{0.5,\ 1.0,\ 1.5,\ \cdots,\ 4.0\}$ [m/s].

3) Results - Identification of the Drag Coefficients: For each V_T, seven flights were performed. The results of the LS fitting can be seen in Figure V.1, with the identified coefficients

$$\hat{a}_{bla} = -0.0476 \ \left[\frac{\text{s}}{\text{m}}\right], \qquad \hat{k}_{par} = -0.0036 \ \left[\frac{\text{kg}}{\text{m}}\right].$$

The results indicate that in fact there is a linear term causing drag due to the velocity. However, the influence of the identified second order drag is too small when compared to the linear term, thus validating the assumption that second order drag can be neglected when flying at relatively slow speeds below 10 [m/s]. From the evaluation of the drag coefficients one can derive the necessary pitch angle to compensate for the drag in steady flight, which can be seen in Figure V.1.

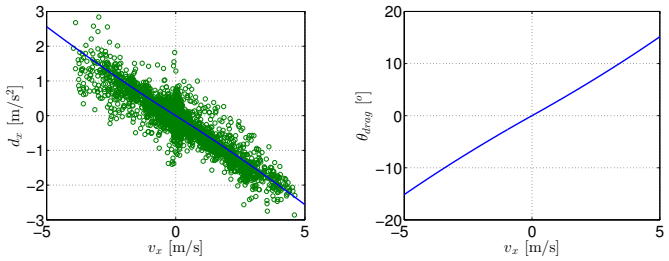

Figure V.1: LS fitting to obtain the drag coefficients (left) and necessary pitch angle to compensate for the drag force (right).

B. Influence of the Lumped Drag Coefficient in the Controller

To see the influence of the lumped drag coefficient two controllers were used, that differ only in Equation (IV.9). The manipulated variable was thus a_{bla}, with $a_{bla} = -0.0476 \ \vee \ a_{bla} = 0$ [s/m]. The parasitic drag coefficient was neglected for both.

1) Results - Influence of the Lumped Drag Coefficient: Figure V.2 shows that the velocity response is faster when the drag term is accounted. For $V_T = 4$ [m/s] the quadrotor is not able to get to the goal due to space limitations but still the response is faster. The velocity is higher than the desirable due to the position error.

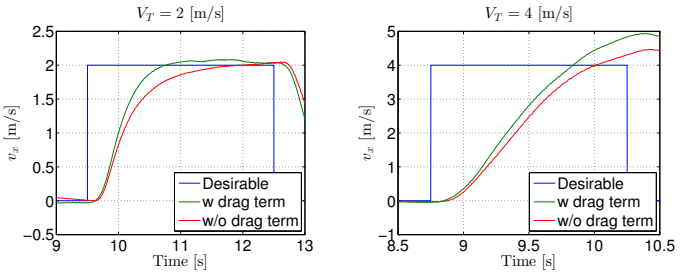

Figure V.2: Velocity for $V_T = 2 \vee V_T = 4$ [m/s].

Figure V.3 shows the position error, which is smaller when the drag term is accounted, by almost 0.5 meters. Concluding, these Figures show that for small velocities (smaller than 10 [m/s]) the linear drag is relevant, thus sustaining the hypothesis that the drag terms should have a significant influence in the controller design.

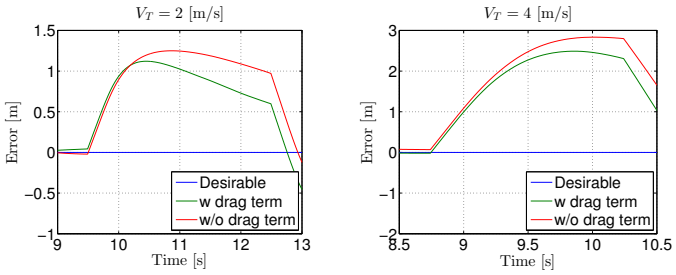

Figure V.3: Position error for $V_T = 2 \vee V_T = 4$ [m/s].

C. Reference Generator Comparison

This Section aims at comparing the performance of a standard trajectory generator, that uses a second order low-pass filter, versus the proposed optimal reference generator. Therefore, the difference in the two controllers is only in the first stage, the reference generator. The trajectory selected consists of $m = 8$ equally distributed waypoints and the limit velocity is set to $V_{lim} = 1$ [m/s]. The quadrotor starts at $(x, y) = (-2, 2)$ [m].

1) Results - Reference Generator: The optimal trajectory, seen in Figure V.4, is smooth and natural, which qualitatively validates the optimal reference generator, whereas the standard trajectory consists of straight lines with some overshoot after the waypoints.

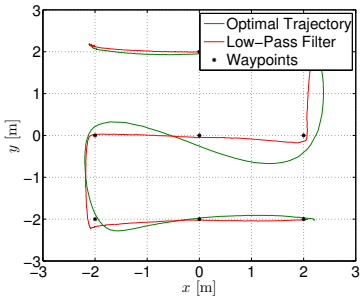

Figure V.4: Comparison of the 2D trajectory.

Velocity magnitude: In Figure V.5 it is shown the time evolution of the velocity magnitude. In both approaches, the total trajectory time is approximately the same, i.e., $T = 27$ [s]. One can see the velocity constraint $V_{lim} = 1$ [m/s] to be respected for almost all time instants. The standard reference generator makes the velocity behavior similar between each waypoint and the velocity profile is in a sawtooth wave fashion. For the optimal trajectory the velocity profile behavior is much smoother and natural, in particular with less variations.

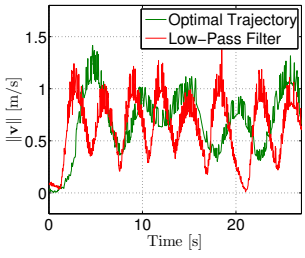

Figure V.5: Caparison of the velocities magnitudes.

Euler angles: The velocities contrasts have influence on the accelerations and thus on the Euler angles, which can be seen in Figure V.6. The Figure shows a clear difference in the results of both references generators, since the angles are bigger and more aggressive for the standard low-pass filter case, whereas in the optimal case the angles are smoother. A save in angles variation of 50 [%] is achieved for optimal trajectory generator.

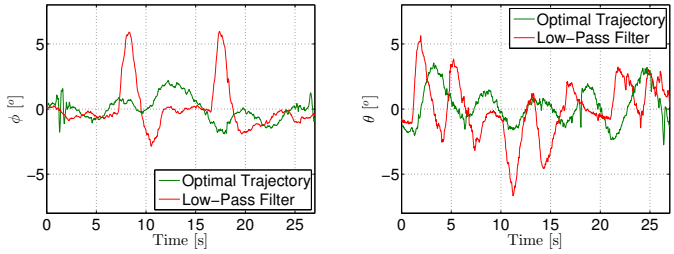

Figure V.6: Comparison of the Euler angles (ϕ left and θ right).

Thrust: Eventually the summed effects of the velocities, accelerations and angles have influence in the energetic performance. As discussed in Section III-B3, it was chosen to minimize jerk since it is a better representative of the aggressiveness of the system inputs and thus it is linked to thrust from Equation IV.16. Figure V.7 compares the thrust for both reference generators and shows that there is less energy consumption when an optimal trajectory is chosen. The hover thrust for the working quadrotor is $T_h = 3.874$ [N], while the optimal mean thrust is $T_{opt} = 3.886$ [N] and the mean thrust resulting from the low-pass filter is $T_{lp} = 3.923$ [N]. Thus, there is a reduction in thrust variation of 77 [%]. Moreover, the thrust during a flight is approximately equal to the hovering thrust, within 5 [%] of thrust variation, sustaining that almost all thrust is to compensate the gravity force.

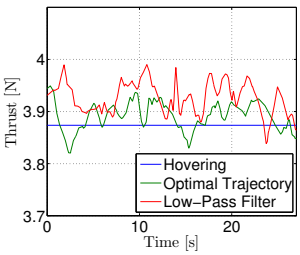

Figure V.7: Comparison of the Thrust.

D. Thrust Vectoring versus PID Control

This Section aims at comparing the performance of a classic PID controller versus the proposed thrust vectoring controller. The tested trajectory consists of $m = 3$ equally distributed waypoints. Two different limit velocities were tested, and throughout all this Section Figures V.8-V.12 show results for the limit velocity $V_{lim} = 1$ [m/s] on the left and for $V_{lim} = 2$ [m/s] on the right.

1) Results - Controllers Comparison: Figure V.8 shows the overall trajectory. One can clearly see the increase in performance on the thrust vectoring controller. Furthermore, the trajectory performance of the proposed controller is not affected by the increase in the velocity, which widens the quadrotor flight envelope.

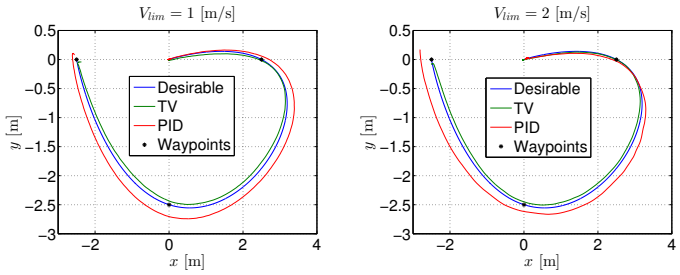

Figure V.8: Comparison of the 2D trajectory.

Horizontal Position: From Figure V.9 one can see that the performance of the PID controller worsens when the velocity increases, whereas in the proposed controller it is relatively the same, and a reduction of one second time delay is achieved. The delay caused in the PID controller is mainly due to the integrator term that is compensating the steady state error mainly caused by the drag term. Since the trajectory is aggressive the integrator can not perform fast enough. The results are similar for the y axis.

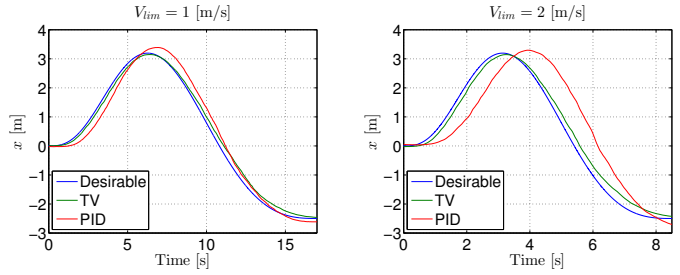

Figure V.9: Comparison of the x trajectory.

Vertical Position: Due to separation in the control loops it is expected that the performance in the vertical position is poorer for the PID controller, as it can be seen in Figure V.10, since the position has more error and variation in the PID controller. For the thrust vectoring case, there is an initial oscillatory behavior but then the following becomes evident, whereas in the PID controller the tracking never truly happens. Note the different scale when comparing with Figure V.9. This is a result of the vertical and horizontal loops inclusion into one single stage.

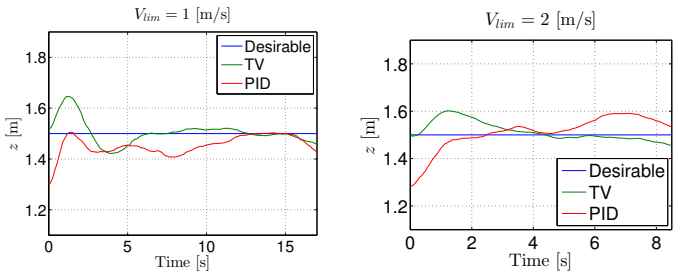

Figure V.10: Comparison of the z trajectory.

Euler Angles: Assuming a perfect following of the trajectory, the desirable pitch and roll angles can be computed, including the influence of the drag effects or not (see Figures V.11 and V.12). One can see that for the thrust vectoring case the performance is better when comparing with the PID case. The performance increases drastically when the velocity also increases, proving the robustness of our controller, which is able to follow trajectories in a wider flight envelope. It can be seen that by including the drag into the thrust vectoring controller the initial response is much faster (see Figure V.12). Furthermore, one can qualitatively validate the identification of the drag term, since the angles appear to follow the dark blue line, that accounts for the drag effects, better than the soft blue line, that does not.

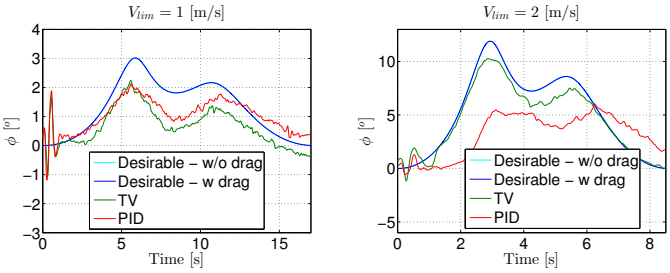

Figure V.11: Comparison of the roll angle ϕ.

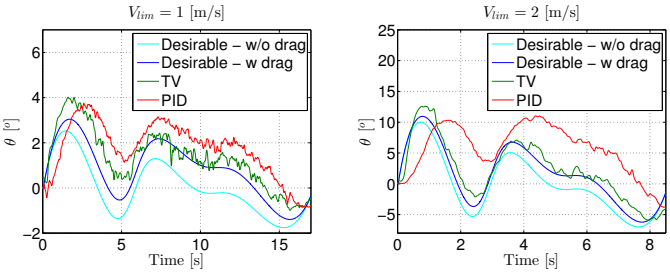

Figure V.12: Comparison of the pitch angle θ.

VI. Conclusion

The aim of this work resided on three topics, namely: studying the aerodynamic effects present in the dynamics of a quadrotor; planning the time optimal trajectory in the presence of wind and; designing a controller that uses wind and trajectory information.

A literature study was performed and the formulas for the most important aerodynamic effects were derived. With an experiment, the lumped and parasitic drag coefficients were identified, resulting in $\hat{a}_{bla} = -0.0476$ [s/m] and $\hat{k}_{par} = -0.0036$ [kg/m], and it was shown that even in environments where wind is absent, the inclusion of the drag terms in the controller increases significantly the overall performance. Therefore, the identification of the drag terms should be a step considered in controllers design stage.

Considering the time optimal trajectory, a simplified method to determine the solution was proposed, based on three steps. This steps division saves computational time, reducing it approximately by an $\mathcal{O}(m!)$ order. The reference generator was compared to a standard one, a low-pass filter, showing a performance increase. The profit comes from saving energy related with thrust variation, reduced by 77 [%], and from reducing the aggressiveness related with angle variations, reduced by half.

Finally, a thrust vectoring controller was proposed and augmented to include the drag terms. The controller was compared with a current available open-source PID controller and it was proven that a clear increase in performance of trajectory following is achieved. A reduction of time delay was obtained, and the position following was improved both in horizontal and vertical planes. Moreover, it was observed that the thrust vectoring controller maintains the performance in a higher flight envelope.

References

[1] G. Allibert, D. Abeywardena, M. Bangura, and R. Mahony. Estimating body-fixed frame velocity and attitude from inertial measurements for a quadrotor vehicle. In *IEEE Conference on Control Applications (CCA)*, pages 978–983, October 2014.

[2] O. Araar and N. Aouf. Quadrotor control for trajectory tracking in presence of wind disturbances. In *UKACC International Conference on Control*, July 2014.

[3] F. Augugliaro, A. P. Schoellig, and R. D'Andrea. Generation of collision-free trajectories for a quadrocopter fleet: A sequential convex programming approach. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1917–1922, October 2012.

[4] M. Bangura and R. Mahony. Nonlinear dynamic modeling for high performance control of a quadrotor. In *Australasian Conference on Robotics and Automation*, December 2012.

[5] A. Barrientos, P. Gutierrez, and J. Colorado. *Advanced UAV trajectory generation planning and guidance*, chapter 4. InTech, January 2009.

[6] K. Bipin, V. Duggal, and K. M. Krishna. Autonomous navigation of generic quadrocopter with minimum time trajectory planning and control. In *IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, pages 66–71, December 2014.

[7] Y. Bouktir, M. Haddad, and T. Chettibi. Trajectory planning for a quadrotor helicopter. In *16th IEEE Mediterranean Conference on Control and Automation*, pages 1258–1263, June 2008.

[8] A. E. Bryson and Y.-C. Ho. *Applied Optimal Control: Optimization, Estimation, and Control*. Taylor and Francis, 1975.

[9] M. Fliess, J. Lévine, P. Martin, and P. Rouchon. On differential flat nonlinear systems. *Comptes Rendus de L'Académie des Sciences*, January 1992.

[10] M. Hehn and R. D'Andrea. Real-time trajectory generation for quadrocopters. *IEEE Transactions on Robotics*, August 2015.

[11] G. M. Hoffmann, H. Huang, S. L. Waslander, and C. J. Tomlin. Quadrotor helicopter flight dynamics and control: Theory and experiment. In *AIAA Guidance, Navigation and Control Conference and Exhibit*, August 2007.

[12] H. Huang, G. M. Hoffmann, S. L. Waslander, and C. J. Tomlin. Aerodynamics and control of autonomous quadrotor helicopters in aggressive maneuvering. In *IEEE International Conference on Robotics and Automation*, pages 3277–3282, May 2009.

[13] Y. B. J.A. Guerrero, J.A. Escareno. Quad-rotor mav trajectory planning in wind fields. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 778–783, May 2013.

[14] L.-C. Lai, C.-C. Yang, and C.-J. Wu. Time-optimal control of a hovering quad-rotor helicopter. *Journal of Intelligent and Robotic Systems (2006)*, pages 115–135, 2006.

[15] G. Laporte. The traveling salesman problem: An overview of exact and approximate algorithms. *European Journal of Operational Research*, pages 231–247, July 1991.

[16] G. J. Leishman. *Principles of Helicopter Aerodynamics*. Cambridge University Press, second edition, July 2006.

[17] R. Mahony, V. Kumar, and P. Corke. Multirotor aerial vehicles - modelling, estimation and control of quadrotor. *IEEE Robotics & Automation Magazine*, pages 20–32, September 2012.

[18] P. Martin and E. Salaun. The true role of accelerometer feedback in quadrotor control. *IEEE International Conference on Robotics and Automation*, pages 1623–1629, May 2010. hal-00422423v1.

[19] D. Mellinger and V. Kumar. Minimum snap trajectory generation and control for quadrotors. In *IEEE International Conference on Robotics and Automation*, pages 2520–2525, May 2011.

[20] D. Mellinger, A. Kushleyev, and V. Kumar. Mixed-integer quadratic program trajectory generation for heterogeneous quadrotor teams. In *IEEE International Conference on Robotics and Automation*, pages 477–483, May 2012.

[21] D. W. Mellinger. *Trajectory Generation and Control for Quadrotors*. PhD thesis, University of Pennsylvania, 2012.

[22] M. W. Mueller, M. Hehn, and R. D'Andrea. A computationally efficient motion primitive for quadrocopter trajectory generation. *Transactions on Robotics*, pages 1294–1310, December 2015.

[23] S. Omari, M.-D. Hua, G. Ducard, and T. Hamel. Nonlinear control of vtol uavs incorporing flapping dynamics. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2419–2425, November 2013.

[24] L. S. Pontryagin. *Mathematical Theory of Optimal Processes*. CRC Press, 1987.

[25] R. W. Prouty. *Helicopter Performance, Stability, and Control*. Krieger Publishing Company, 2002.

[26] C. Richter, A. Bry, and N. Roy. Polynomial trajectory planning for quadrotor flight. In *International Symposium of Robotics Research (ISRR)*, 2013.

[27] E. J. Smeur, Q. Chu, and G. C. de Croon. Adaptive incremental nonlinear dynamic inversion for attitude control of micro aerial vehicles. In *AIAA Guidance, Navigation, and Control Conference*, January 2016.

[28] N. Sydney, B. Smyth, and D. A. Paley. Dynamic control of autonomous quadrotor flight in an estimated wind field. In *52nd IEEE Conference on Decision and Control*, December 2013.

[29] S. Waslander and C. Wang. Wind disturbance estimation and rejection for quadrotor position control. In *AIAA Infotech@Aerospace Conference*, April 2009.

[30] D. Zhou and M. Schwager. Vector field following for quadrotors using differential flatness. In *IEEE International Conference on Robotics & Automation (ICRA)*, pages 6569–6572, June 2014.