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

J. P. R. Silva, G. C. H. E. de Croon and C. De Wagter

***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 standard 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 [Hehn and D'Andrea, 2015]. Application scenarios such as surveillance, package delivery or plant monitoring reflect the competence that quadrotors possess to follow predefined trajectories [Hoffmann et al., 2007]. 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 *a priori* information about the wind in order to increase the quadrotors performance becomes a natural solution [J.A. Guerrero, 2013].

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, and the trajectory needs to be promptly computed to serve real-time applications. 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.

J.P.R. Silva is a Master student with the Control and Simulations Department, Faculty of Aerospace, Delft University, 2629 HS Delft, The Netherlands.

G.C.H.E. de Croon and C. De Wagter are with the Micro Air Vehicle laboratory of the Faculty of Aerospace, Delft University, 2629 HS Delft, The Netherlands.

Trajectory generation for quadrotors has been studied in several approaches. Smooth point to point trajectories have been studied with Non Linear Programming (NLP) [Lai et al., 2006] and using the Pontryagin's Minimum Principle (PMP) [Mueller et al., 2013, Hehn and D'Andrea, 2015, Mueller et al., 2015]. For multiple waypoints the usual approaches are Sequential Convex Programming (SCP) [Augugliaro et al., 2012] or Quadratic Programming (QP) [Mellinger, 2012, Richter et al., 2013], extended to Multi Integers Linear Programming (MILP) by Mellinger et al. [2012]. Those approaches, however, aim at minimizing accelerations, jerks or snaps, disregarding wind and assuming fixed traveling times. J.A. Guerrero [2013] is among the few who include wind in the trajectory formulation, but plans only point to point trajectories. Bipin et al. [2014] achieve the goal of minimizing the time for multiple waypoints, but fail to include the wind.

Regarding quadrotor control, the typical controllers available are based on standard PID theory [Hoffmann et al., 2007, Waslander and Wang, 2009, Martin and Salaun, 2010]. 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 [Sydney et al., 2013] or Integral-Backstepping [Araar and Aouf, 2014] exist, but are only implemented in simulations. In a further step, Mellinger and Kumar [2011] consider the trajectory generation in the controller design, using a thrust vectoring approach, but neglect wind. The same approach is used by Omari et al. [2013] but without the trajectory information. Extending both the work of Mellinger and Kumar [2011] and Omari et al. [2013] to incorporate both the wind and the trajectory information seems to be an interesting approach that fulfills the controller requisites.

A general method to achieve the proposed goal is thus absent in literature. Therefore, a new approach is proposed, that consists in three sequentially linked aspects. The initial aspect of the approach is to address an aerodynamic study in order to understand the three most important aerodynamic effects that influence the quadrotors dynamics: blade flapping drag, induced drag 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, that is divided in three steps, so that the process is structurally fast, therefore 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 a QP optimization problem that minimizes 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 influence and the associated aerodynamic effects in the quadrotor. The time optimal trajectory is defined and constructed in Section III. Section IV presents the quadrotor mathematical model and the proposed cascade controller, while Section V shows the results. This work ends with the conclusion in Section VI and with Section VII where future work is suggested.

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 describe each effect with importance for this work individually (see Sections II-B-II-E). In Section II-F the other drag effects are shown.

In order to plan the optimal trajectory in the presence of wind fields it is firstly necessary to understand how the wind affects the performance of the quadrotor, during its maneuvers, and also how can the controller include the wind influence, instead of only considering it as a disturbance to be further rejected. In the absence of wind, the velocity of the quadrotor w.r.t. the ground, $\dot{\mathbf{r}}$, is equal to the velocity of the quadrotor w.r.t. the airflow, $\dot{\mathbf{r}}_\infty$. When this is not the case, and a stream of wind exists, then

$$\dot{\mathbf{r}} = \dot{\mathbf{r}}_\infty + \dot{\mathbf{r}}_w \tag{II.1}$$

where $\dot{\mathbf{r}}_w$ is the wind speed w.r.t the ground. Equation (II.1) can also be written with velocity vectors as $\mathbf{v} = \mathbf{v}_\infty + \mathbf{v}_w$. As it will be seen later, all the aerodynamic effects are established w.r.t. the airstream, the apparent wind. For this fact, the wind effect is only to alter that airstream, and studying the airstream influence on the aerodynamics of the quadrotor alone is sufficient to withdraw the necessary conclusions when wind is present.

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 distant parent of quadrotors, precise studies have been performed, such as Prouty [2002] and Leishman [2006]. In the latest, the aerodynamic properties of rotating blades are studied in detailed. However, for the case of small and more recent aircraft, such as quadrotors, the literature is more scarce. Hoffmann et al. [2007] are one of the first 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, Huang et al. [2009] also consider the first two aspects and Waslander and Wang [2009] modeled the total drag as a lumped factor linear w.r.t. to the airspeed. Bangura and Mahony [2012] are among the first who pursue a detailed explanation of all aerodynamic effects that affect the quadrotors dynamics. They consider in detail several aspects, such as blade flapping, induced drag, translational drag, profile drag, parasitic drag and also others such as ground effect and vertical descent. Mahony et al. [2012] refers to induced drag and blade flapping and also treats them as a lumped parameter linear w.r.t. the horizontal air stream. Later they also refer to additional aerodynamic effects that are important for high-speed and highly dynamic maneuvers. In the Allibert et al. [2014] approach, translational drag and blade flapping are considered bilinearly influenced by the thrust and the horizontal velocity w.r.t. the wind. They neglect parasitic drag since they operate near hovering. Moreover, they also consider the vertical thrust force variation due to the induced velocity. More recently, Ryll et al. [2015] considers the aerodynamic effects and perform experiment to obtain the model corresponding coefficients. There, it is shown that the aerodynamic effects can be neglected. They also consider the induced drag and blade flapping as first order dynamic effects but still neglect them. However, they test the model with velocities smaller than 1 [m/s], which is clearly not adequate for this work.

To summarize, there are several aerodynamic effects that influence the quadrotor dynamics which have been reported, however briefly, in literature. The following Sections II-B-II-E analyze each effect separately, in a similar fashion as Bangura and Mahony [2012]. Emphasis is given to the aerodynamic drag effects which are considered of significant importance for this work. The formulas have some simplifications, since the interest of this work is to obtain a relatively simple model of the wind/drag effects, in order to use them in the trajectory generation method proposed in Section III and to include them in the controller to be designed in Section IV. Nevertheless, the introduced literature is suggested for further understanding of the drag effects.

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 (see Figure II.1). The angle β can be decomposed for each rotor in two components, parallel and perpendicular to the airstream, as

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

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 and one can consider that $\frac{1}{2}\mu_i^2 \ll 1$ due to the high rotation speed of the rotors [Bangura and Mahony, 2012].

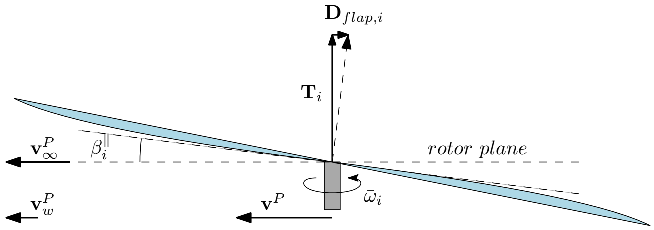

Figure II.1: Blade Flapping angle illustration.

Being so, most authors who describe blade flapping as a drag force due to aerodynamic effects [Mahony et al., 2012, Allibert et al., 2014] represent it as a function of thrust magnitude, rotor linear velocity and airspeed as

$$\mathbf{D}_{flap,i} = T_i \frac{\mathbf{A}_{flap}}{\bar{\omega}_i r} \mathbf{v}_\infty^P \tag{II.3}$$

with

$$\mathbf{A}_{flap} = \begin{pmatrix} a_1 & a_2 & 0 \\ -a_2 & a_1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

a constant matrix. Recently, Omari et al. [2013] uses the properties of blade theory to better estimate some states of the state vector, using accelerometers and including blade flapping in the quadrotor model. There, blade flapping is not written as in Equation II.3, although representing the same. This has to do with the fact that thrust can be written as a function of the rotor angular speed (see Section IV-A2) as

$$T_i = k_T \bar{\omega}_i^2, \tag{II.4}$$

with k_T the thrust coefficient, leading thus to

$$\mathbf{D}_{flap,i} = \bar{\omega}_i \mathbf{A}'_{flap} \mathbf{v}_\infty^P \tag{II.5}$$

with

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

a constant matrix similar in structure to $\mathbf{A}_{flap}$.

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. In order to model it, a linear drag coefficient can be introduced, leading to

$$\mathbf{D}_{ind,i} = T_i \mathbf{A}_{ind} \mathbf{v}_\infty^P, \tag{II.6}$$

with

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

a constant matrix with k_{ind} the induced drag coefficient.

D. Lumped Drag Coefficient

The formulas obtained for blade flapping and induced drag are only 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 $\sum_{i=1}^4 T_i = T$. For the blade flapping case it is different. In Martin and Salaun [2010], 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

$$\mathbf{D}_{flap} = T_h \mathbf{A}''_{flap} \mathbf{v}_\infty^P \tag{II.7}$$

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

$$\mathbf{D}_{bla} = T \mathbf{A}_{bla} \mathbf{v}_\infty^P \tag{II.8}$$

with

$$\mathbf{A}_{bla} = \begin{pmatrix} a_{bla} & 0 & 0 \\ 0 & a_{bla} & 0 \\ 0 & 0 & 0 \end{pmatrix} \approx \begin{pmatrix} \frac{\sum \bar{\omega}_i a'_1}{T} + k_{ind} & \frac{\sum \bar{\omega}_i a'_2}{T} & 0 \\ -\frac{\sum \bar{\omega}_i a'_2}{T} & \frac{\sum \bar{\omega}_i a'_1}{T} + k_{ind} & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{II.9}$$

the lumped drag matrix and a_{bla} the lumped drag coefficient.

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] [Bangura and Mahony, 2012], are in question, since it is proportional to the square of the translational velocity. It can be expressed as

$$\mathbf{D}_{par} = k_{par} \|\mathbf{v}_\infty\| \mathbf{v}_\infty, \tag{II.10}$$

with $k_{par} = \frac{1}{2}\rho S C_{D_{par}}$ depending on the non-lifting surfaces area, the air density and the non-lifting drag coefficient, with $C_{D_{par}}$ negligible w.r.t. the blades drag coefficient C_D [Bangura and Mahony, 2012], all assumed constant.

In environments where the wind speed and the ground speed 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.

F. Other Drag Forces

The blade flapping, induced and parasitic drag represent most of the important aerodynamic effects useful for this work. Nevertheless, other aspects are referred in the literature, which will be briefly described here for the sake of completeness:

- Total Thrust Variation - is the phenomena that occurs when a quadrotor undergoes a translational motion or changes the angle of attack, both w.r.t. the airflow. When this occurs, there is an induced velocity given by

$$V_{ind} = \frac{V_h^2}{\sqrt{(V_\infty \cos\gamma)^2 + (V_\infty \sin\gamma + V_{ind})^2}} \tag{II.11}$$

where V_h is the induced velocity at hover thrust and γ is the angle of attack. The induced velocity will alter the necessary thrust to perform the maneuver, and the ratio between the ideal thrust and the thrust at hovering point can be then obtained as function of γ and V_∞ with

$$\frac{T}{T_h} = \frac{V_h}{V_{ind} + V_\infty \sin\gamma} \tag{II.12}$$

where T_h is the nominal thrust when hovering;

- Translational Drag - also known as momentum drag, it appears when the induced velocity passes the rotors, creating a drag proportional to the lift. For small velocities, it can be discarded when compared with induced drag or the blade flapping effect. The same happens for high velocities, as translational drag starts decreasing after a velocity threshold;
- Profile Drag - is the result of the transverse velocity of the rotor blades moving through the air. It is zero at hovering and usually does not depend on the angle of attack. It can be discarded when compared with the induced drag or the blade flapping effect;
- Ground Effect - it appears for flights near the ground, and results in a reduced necessary power to hover when closer to it. This effect will not appear in this work, as the quadrotor will not fly close to the ground;
- Vertical Descent - it appears when a vertical descent maneuver is in place, causing opposite directions in the induced velocity and airspeed. This can lead to vortices or turbulence generation.

III. Time Optimal Trajectory

The problem addressed in this Section is to determine the time optimal trajectory that covers multiple waypoints and considers both the quadrotor dynamics and wind information. 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 and the quadrotor dynamics.

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 online applications. Therefore, a second approach that separates the problem into three smaller problems, or steps, will be considered.

The first smaller problem of the proposed approach is to estimate the optimal sequence of waypoints (see Figure III.2) and it will be addressed in Section III-A. In order 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 purely geometric approach with constraints 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 second, and more complex, one. It is necessary to note that 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 in the proposed trajectory generation method.

The possible trajectories are thus restrained to lines segments, circles and splines [Barrientos et al., 2009] or to B-splines [Bouktir et al., 2008, Lorenz and Adolf, 2010]. Another geometrical approach is to consider vector fields such as in Zhou and Schwager [2014]. Since at this point the goal lies in selecting the optimal sequence, the complexity of the trajectory itself between each point does not need to be significant.

For the remaining of this Section III-A, the focus will be on line segments, that are evidently less complex. 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 to solve 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 [Bryson and Ho, 1975] since the wind is constant. 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

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

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 J.A. Guerrero [2013] but not in constant wind fields.

Using the Pontryagin's Minimum Principle [Pontryagin, 1987] to optimize the trajectory time T the cost function becomes

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

the Hamiltonian is expanded as

$$H = 1 + \lambda^\top f = 1 + \lambda_x(V\cos(\chi) + u) + \lambda_y(V\sin(\chi) + v) \tag{III.3}$$

and co-state Euler-Lagrange equations are

$$\dot{\lambda}_x = -\frac{\partial H}{\partial x} = -\lambda_x \frac{\partial u}{\partial x} - \lambda_y \frac{\partial v}{\partial x}, \tag{III.4}$$

$$\dot{\lambda}_y = -\frac{\partial H}{\partial y} = -\lambda_x \frac{\partial u}{\partial y} - \lambda_y \frac{\partial v}{\partial y}, \tag{III.5}$$

$$0 = \frac{\partial H}{\partial \chi} = V(-\lambda_x \sin(\chi) + \lambda_y \cos(\chi)) \Rightarrow \tan(\chi) = \frac{\lambda_y}{\lambda_x}. \tag{III.6}$$

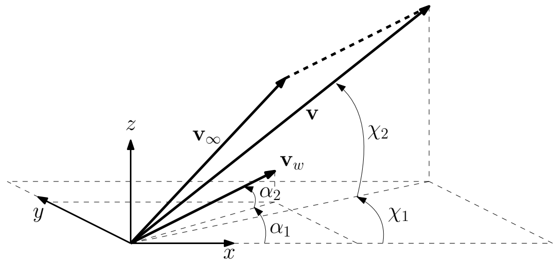

Figure III.1: Illustration of the different velocities and angles.

As it can be seen in Equation (III.1), in the original Zermelo's problem the sea current velocity is dependent on the position and it influences the solution of the problem. In the problem regarded in this work, the wind velocity is constant and independent of the position, meaning that all derivatives w.r.t. the position are zero. This will lead to

$$\left.\begin{aligned}\dot{\lambda}_x = 0 \Rightarrow \lambda_x = a\\ \dot{\lambda}_y = 0 \Rightarrow \lambda_y = b\end{aligned}\right\} \Rightarrow \chi = c \tag{III.7}$$

for some constants a, b and c. Thus, the desirable course angle is constant and the optimal trajectory is a line segment between two consecutive points. The problem can be expanded to include the z direction and still straight lines will be the solution.

Minimizing the time to get the time optimal solution implies maximizing the velocity, and thus the velocity can be determined using the restriction on the limit velocity w.r.t. the airstream, so that parasitic drag effects can be diminished, as in

$$V^2 = \|\mathbf{v}_\infty\|^2 = \|\mathbf{v} - \mathbf{v}_w\|^2 \leq V_{lim}^2 \tag{III.8}$$

in which the quadratic form is used to allow for negative velocities. Converting the inequality to equality to maximize the velocity, the solution of Equation (III.8) is

$$V = V_w g \pm \sqrt{(V_w g)^2 + V_{lim}^2 - V_w^2} \tag{III.9}$$

where

$$\begin{aligned} g(\chi_1, \chi_2, \alpha_1, \alpha_2) = \cos(\chi_1)\cos(\chi_2)\cos(\alpha_1)\cos(\alpha_2) + \\ + \sin(\chi_1)\cos(\chi_2)\sin(\alpha_1)\cos(\alpha_2) + \sin(\chi_2)\sin(\alpha_2) \end{aligned}$$

and the plus sign is always used since V was defined as a positive constant. The angles can be seen in Figure III.1.

With m waypoints, there are $m!$ possible waypoint sequences. For each possible sequence, the χ_{1i} and χ_{2i} angles are determined, the velocity V_i is computed and the time T_i is estimated. It is evident that depending on the geometry of the waypoints and on the wind velocity, different trajectory times will result. In the next Section III-A2 the optimal sequence which optimizes the time $T = \sum_{i=1}^{m} T_i$ will be determined using search algorithms.

2) The Traveling Salesman Problem: The Traveling Salesman Problem (TSP) is a well known NP-hard problem in the Artificial Intelligence field [Laporte, 1991]. It aims at determining the optimal sequence between m possible cities that a salesman has to cover, with no restrictions on each city to visit first. Determining the optimal sequence of waypoints the quadrotor has to pass is analogous to choosing in which order the salesman travels between cities. In the original problem, the traveling times between each city are independent on the direction. In this work, however, when wind is present this condition does not prevails and for that reason an asymmetrical TSP is defined, which can be represented in a directed graph as in Figure III.2.

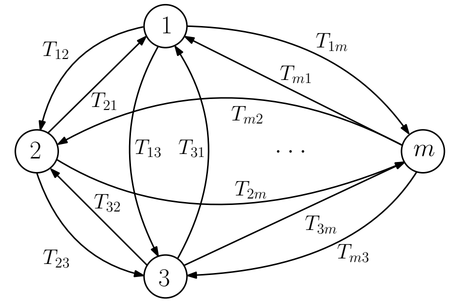

Figure III.2: Directed Graph illustrating a generic distribution of m waypoints and traveling times.

In order to solve the TSP several approaches can be considered. The most intuitive is to evaluate all possible sequences and checking which one is the best, a brute force approach. Another approach is to write the problem in an Integer Linear Programming (ILP) formulation and numerically solve it [Papadimitriou and Steiglitz, 1998]. One last approach is to use search algorithms to start from the origin and incrementally extend the sequence until the least costing solution is found [Rego et al., 2011].

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

Search Algorithms for the TSP

The search algorithms used were of two types. The first type, the uninformed (or blind) search, is general and allows to solve every search problem type. The second type, the informed (or heuristic) search, is dependent on the specific problem as it uses extra information that varies from problem to problem. Both methods start at the origin node (or a set of origin nodes). Every created, and still unexplored, node is part of the open-list of nodes. At each iteration the better node is chosen from the open-list, using a selection criteria, and it is expanded. The different search algorithms usually differ in the selection criteria. When a goal node is reached, the search stops.

A strategy to solve uninformed search problems is to, at each iteration, select the node with lower path cost, the so called uniform cost search. This method is optimal and complete thus is a suitable method for this work. For this type of search algorithms, the total path cost is

$$f(i) = g(i) \tag{III.10}$$

where i is the $i$ths current waypoint in the sequence and

$$g(i) = \sum_{j=1}^{i} T_j \tag{III.11}$$

for the js waypoints in the sequence.

Informed search algorithms use heuristic functions that estimate the cost of going from a specific node until the goal node. This cost is obviously problem dependent and the challenge lies in how to evaluate it. If only the heuristic is used as the selecting criteria, then it is a greedy best-first search algorithm, which is not optimal. However, if the information of getting from the origin node until the current node, as well as the information of getting from the current node until the goal node are joined, it is an A^* search method. If the heuristic function used for the A^* search is admissible (never underestimates the true cost) then this approach is optimal. For informed search algorithms the total path cost is

$$f(i) = g(i) + h(i) \tag{III.12}$$

where $h(i)$ is the heuristic function and in the greedy best-first search $g(i) = 0$.

Heuristics for the TSP

Two heuristic functions were used. The first one is admissible, and it is given as

$$h_1(i) = (m - i)T_{min} \tag{III.13}$$

where $m - i$ is the remaining number of waypoints to get to the goal node and T_{min} is the minimum traveling time between two nodes. The second heuristic method is not admissible, since it may happen that it underestimates the true cost of getting to the goal, and is given as

$$h_2(i) = K_{heu}(m - i)T_{ave} \tag{III.14}$$

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 as it will be discussed later. 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 [Mellinger and Kumar, 2011, Mellinger et al., 2012, Richter et al., 2013, Bipin et al., 2014] or using the Pontryagin's Minimum Principle, which results in an analytical solution [Mueller et al., 2013, Hehn and D'Andrea, 2015, Mueller et al., 2015]. Other approaches to plan the trajectory between two points are Look-Up Tables [Corbets and Langelaan, 2007], and discretizing the model or the trajectory [Lai et al., 2006, Augugliaro et al., 2012]. 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 of the QP and continuity in the derivatives can be guaranteed as far as wanted.

The trajectory is defined in Section III-B1 and the QP problem is generally constructed in Section III-B2. In Sections III-B3-III-B6 the specifications of the QP problem are determined.

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 Fliess et al. [1992] and, because of their properties, are well suited for trajectory definition and generation.

According to van Nieuwstadt and Murray [1997], the differential flatness theory states that the set of outputs must be equal in number to the set of inputs, in order to allow a direct algebraic mapping. Generically, the non-linear state-space system

$$\begin{aligned} \dot{\mathbf{x}}(t) &= f(\mathbf{x}, \mathbf{u}, t) \quad \mathbf{x} \in \mathbb{R}^n, \ \mathbf{u} \in \mathbb{R}^m \\ \mathbf{y}(t) &= h(\mathbf{x}, \mathbf{u}, t) \quad \mathbf{y} \in \mathbb{R}^m \end{aligned} \tag{III.15}$$

is differentially flat if it is possible to find outputs of $\mathbf{z} \in \mathbb{R}^m$ in the form

$$\mathbf{z}(t) = \zeta(\mathbf{x}, \mathbf{u}, \dot{\mathbf{u}}, ..., \mathbf{u}^{(l)}, t) \tag{III.16}$$

such that

$$\begin{aligned} \mathbf{x}(t) &= \mathbf{x}(\mathbf{z}, \dot{\mathbf{z}}, ..., \mathbf{z}^{(l)}, t) =: \mathbf{x}(\bar{\mathbf{z}}, t), \\ \mathbf{u}(t) &= \mathbf{u}(\mathbf{z}, \dot{\mathbf{z}}, ..., \mathbf{z}^{(l)}, t) =: \mathbf{u}(\bar{\mathbf{z}}, t), \end{aligned} \tag{III.17}$$

with $\mathbf{y}$ the tracking outputs and $\mathbf{z}$ the flat outputs. This means that every element of the state $\mathbf{x}$ and the input $\mathbf{u}$ are covered by $\mathbf{z}$.

Differentially flat systems are useful when explicit trajectory generation is required and for quadrotors the flat outputs are usually [Zhou and Schwager, 2014, Mellinger and Kumar, 2011] chosen as

$$\mathbf{z}(t) = [\ r_x(t) \quad r_y(t) \quad r_z(t) \quad \psi(t)\ ]^\top, \tag{III.18}$$

with the velocity, acceleration and the pitch θ and roll ϕ angles the other elements of the state vector to be withdrawn. A proof that quadrotors are differentially flat can be found in Zhou and Schwager [2014]. Using differential flatness, a target time trajectory is defined as

$$\mathbf{r}_T(t) = [\ r_{T_x}(t) \quad r_{T_y}(t) \quad r_{T_z}(t) \quad \psi_T(t)\ ]^\top \tag{III.19}$$

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 in that single direction can be defined as a n order time polynomial such that

$$\sigma_T(t) = \begin{cases} c_{10} + c_{11}t + c_{12}t^2 + \cdots + c_{1n}t^n & 0 \leq t \leq T_1 \\ c_{20} + c_{21}t + c_{22}t^2 + \cdots + c_{2n}t^n & T_1 \leq t \leq T_2 \\ \vdots & \\ c_{m0} + c_{m1}t + c_{m2}t^2 + \cdots + c_{mn}t^n & T_{m-1} \leq t \leq T \end{cases} \tag{III.20}$$

with $m(n+1)$ coefficients c_{ij}. In the previous definition time is monotonically increasing, meaning that the c_{i0} do not represent the m waypoints coordinates. To do so, a time-shift can be applied to Equation (III.20) such that

$$t = T_{i-1} + t_i \qquad T_{i-1} \leq t \leq T_i \tag{III.21}$$

and

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

The time trajectory of the position and its derivatives, without the i indexes for nomenclature convenience, is then given by

$$\begin{aligned} \sigma_T(t) &= c_0 + c_1 t + \cdots + c_n t^n \\ \dot{\sigma}_T(t) &= c_1 + 2c_2 t + \cdots + n c_n t^{n-1} \\ \ddot{\sigma}_T(t) &= 2c_2 + 6c_3 t + \cdots + n(n-1)c_n t^{n-2} \\ &\vdots \\ \frac{\mathrm{d}^k \sigma_T(t)}{\mathrm{d}t^k} &= \sum_{j=0}^{n-k} \frac{(k+j)!}{j!} c_{k+j} t^j \end{aligned} \tag{III.23}$$

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

$$\begin{aligned} \min \ J(\mathbf{c}) &= \frac{1}{2}\mathbf{c}^\top \mathbf{H}\mathbf{c} + \mathbf{f}^\top \mathbf{c} \\ \text{subjected to } & \mathbf{A}_{in}\mathbf{c} \leq \mathbf{b}_{in} \\ \text{and } & \mathbf{A}_{eq}\mathbf{c} = \mathbf{b}_{eq} \end{aligned} \tag{III.24}$$

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 [Mellinger and Kumar, 2011, Mellinger et al., 2012], a sum of different derivatives [Richter et al., 2013] or the Boor control points for the B-Spline [Bipin et al., 2014].

If the objective is to minimize the power of a general derivative of the position then it can be formulated as

$$\sigma_T^*(t) = \min \int_0^T \sum_{i=1}^4 K_i \left(\frac{\mathrm{d}^k \sigma_{T_i}(t)}{\mathrm{d}t^k} \right)^2 \mathrm{d}t \tag{III.25}$$

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}. By choosing to minimize a derivative of the position, the $\mathbf{H}$ matrix will result in a block diagonal of m independent $(n+1) \times (n+1)$ matrices as

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_1 & 0 & \cdots & 0 \\ 0 & \mathbf{H}_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{H}_m \end{bmatrix} \tag{III.26}$$

and the $\mathbf{f}$ vector will be null. Nevertheless, the waypoints are still 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 also 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) Minimizing the Velocity w.r.t. the Wind: In order to minimize the effects of the parasitic drag, an interesting cost function is thus to minimize the power of the velocity of the quadrotor w.r.t. the wind. Considering a general direction $\sigma_T(t)$, the optimal trajectory solution is given by

$$\sigma_T^*(t) = \min \int_0^T \left(\dot{\sigma}_T(t) - V_{w,\sigma} \right)^2 \ \mathrm{d}t \tag{III.27}$$

with $V_{w,\sigma}$ the wind velocity in the direction of the trajectory direction σ_T. Equation (III.27) can be expanded, and the fact that wind is constant can be used to obtain

$$\sigma_T^*(t) = \min \int_0^T \dot{\sigma}_T^2(t) \ \mathrm{d}t - 2V_{w,\sigma} \int_0^T \dot{\sigma}_T(t) \ \mathrm{d}t. \tag{III.28}$$

The first term will give origin to the $\mathbf{H}$ matrix while the second term will give origin to the $\mathbf{f}$ vector. The necessary constraint to this problem is the displacement in the position, that is

$$\int_0^T \dot{\sigma}_T(t) \ \mathrm{d}t = \Delta\sigma_T \tag{III.29}$$

where the first term will give origin to $\mathbf{A}_{eq}$ and the second one to $\mathbf{b}_{eq}$. Thus, comparing Equation (III.29) with Equation (III.28) one can see that, by construction, the second part is a constant term and thus irrelevant to the minimization problem. Finally, it can be concluded that minimizing Equation (III.27) is the same as

$$\sigma_T^*(t) = \min \int_0^T \dot{\sigma}_T^2(t) \ \mathrm{d}t \tag{III.30}$$

and a conclusion can be withdrawn: wind has no influence in the trajectory formulation. Thus, other derivatives of the position can be chosen that better represent the quadrotors dynamics.

4) What Derivative to Minimize?: It was proven that minimizing the power of the velocity w.r.t. the wind is not relevant. Thus minimizing the following position derivatives powers, which have influence on the quadrotors dynamics (see Equation (IV.13)), is proposed: acceleration, that is directly implied in the quadrotor model and can be linked to thrust; jerk, the first derivative of the acceleration which directly corresponds to the quadrotor body rates; and snap, second derivative of the acceleration and proportional to the motor commands and attitude accelerations.

Acceleration is the simplest of all but it is also the 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, and this is acceptable due to the thrust vectoring controller to be proposed in Section IV-D, which will allow a great flexibility in Euler angles.

According to Mueller et al. [2015] the jerk cost is a better representative of the aggressiveness of the true system inputs, and the jerk, like the acceleration, has a direct link with thrust. Plus, they say they can bound the body rates as functions of the jerk and the thrust. Moreover Hehn and D'Andrea [2015] affirm that maintaining constraints on the acceleration and jerk leads to a continuous thrust during the maneuver, which is then supported by [Bipin et al., 2014] when affirming that constraints on jerk are necessary for a smooth trajectory. Finally, it has also been studied [Flash and Hogan, 1985] that humans tend to minimize the integral of the square magnitude of the jerk, in order to increase their performance in motion coordination.

According to Richter et al. [2013] minimum snap trajectories have also been proven effective to generate quadrotor trajectories, due to the linkage in the motors commands and body rate derivatives. The same says Mellinger [2012] and supports that with a study of human movement [Kawato et al., 1990] that claims that the best criterion for modeling motion is minimizing the integral of the square norm of torque torque derivatives, which are related to snap.

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.

5) 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. In the case that $n = 5$ the jerk is a second order time polynomial which is sufficient to give reliable results. Moreover, continuity constraints were imposed until $p = 2$, in order to guarantee continuity at least until the second derivative of the position, the acceleration.

6) 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 by allowing more time to one segment than another, 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 [Mellinger and Kumar, 2011].

C. Constraint on the Maximum Velocity

At this point, the total time optimal trajectory time T is only 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 third step of the approach in the planning of the total time optimal trajectory, 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. Usual controllers neglect the aerodynamic effects and are generally naive when following a trajectory, since they only account for discrete values of desired position setpoints. Thus, powerful and real-time information that can be used to design the controller is being neglected for the standard cases, such as the the Paparazzi UAV embedded controller. Paparazzi UAV 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 "$\times$" pairwise symmetrical counter-rotating fashion, and attached to the quadrotor rigid body as it can be seen in Figure IV.1.

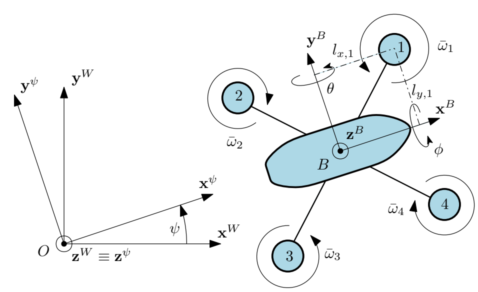

Figure IV.1: Top view of a typical quadrotor in "$\times$" configuration with reference frames scheme.

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 $\mathbf{R}_W^B = (\mathbf{R}_B^W)^{-1} = (\mathbf{R}_B^W)^\top$ since the rotation matrix is orthogonal. $\mathbf{R}_B^W$ is composed of three consecutive $Z - Y - X$ rotations of the Euler angles yaw, pitch and roll, with

$$\mathbf{R}_z = \begin{pmatrix} c\psi & -s\psi & 0 \\ s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \mathbf{R}_y = \begin{pmatrix} c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta \end{pmatrix} \quad \mathbf{R}_x = \begin{pmatrix} 1 & 0 & 0 \\ 0 & c\phi & -s\phi \\ 0 & s\phi & c\phi \end{pmatrix} \tag{IV.1}$$

using c and s for the $\sin(\cdot)$ and $\cos(\cdot)$ notation, such that

$$\mathbf{R}_B^W = \begin{pmatrix} c\theta c\psi & s\theta s\phi c\psi - c\phi s\psi & s\theta c\phi c\psi + s\phi s\psi \\ c\theta s\psi & s\theta s\phi s\psi + c\phi c\psi & s\theta c\phi s\psi - s\phi c\psi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{pmatrix}. \tag{IV.2}$$

With this convention of a right-handed coordinated system, ψ is positive if the quadrotor is yawing left, θ is positive if the quadrotor is pitching down and ϕ is positive if the quadrotor is rolling right. 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 roll rate is measured directly in the body frame, the pitch rate is measured in an intermediate frame rotated by the roll angle and the yaw rate is measured in the next intermediate frame rotated by pitch and roll angles, such that, in body coordinates

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \mathbf{I}\begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + \mathbf{R}_x^\top \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + (\mathbf{R}_y\mathbf{R}_x)^\top \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\phi & c\theta s\phi \\ 0 & -s\phi & c\theta c\phi \end{bmatrix}\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{IV.3}$$

where $\mathbf{I}$ is a 3×3 identity matrix. 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, such that

$$\mathbf{a}_\times := \begin{pmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{pmatrix}. \tag{IV.4}$$

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

$$\begin{cases} m\ddot{\mathbf{r}} = \mathbf{R}\sum_{i=1}^4 \mathbf{F}_i - mg\mathbf{e}_3 + \mathbf{F}_{aero} \\ \dot{\mathbf{R}} = \mathbf{R}\boldsymbol{\omega}_\times \\ \mathbf{J}\dot{\boldsymbol{\omega}} = -\boldsymbol{\omega}_\times\mathbf{J}\boldsymbol{\omega} + \sum_{i=1}^4(\mathcal{T}_i + \mathbf{l}_i\times\mathbf{F}_i) + \mathcal{T}_{aero} \end{cases} \tag{IV.5}$$

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.8). Thus

$$\sum_{i=1}^4 \mathbf{F}_i = \sum_{i=1}^4(\mathbf{F}_{T,i} + \mathbf{D}_{bla,i}) = T\mathbf{e}_3 + \sum_{i=1}^4 T_i\mathbf{A}_{bla}\mathbf{v}_{\infty,i}^P \tag{IV.6}$$

with

$$\sum_{i=1}^4 T_i\mathbf{A}_{bla}\mathbf{v}_{\infty,i}^P = TA_{bla}\mathbf{R}^\top(\dot{\mathbf{r}} - \mathbf{v}_w). \tag{IV.7}$$

The aerodynamic force is the parasitical drag force (see Equation (II.10)) given by

$$\mathbf{F}_{aero} = \mathbf{D}_{par} = k_{par}\|\mathbf{v}_\infty\|\mathbf{v}_\infty = k_{par}\|\dot{\mathbf{r}} - \mathbf{v}_w\|(\dot{\mathbf{r}} - \mathbf{v}_w) \tag{IV.8}$$

The torque terms are

$$\sum_{i=1}^4(\mathcal{T}_i + \mathbf{l}_i\times\mathbf{F}_i) = \mathcal{T} + \sum_{i=1}^4 T_i\mathbf{l}_i \times \mathbf{A}_{bla}\mathbf{v}_{\infty,i}^P \tag{IV.9}$$

with

$$\mathcal{T} = \sum_{i=1}^4 \lambda_i k_\tau\bar{\omega}_i^2\mathbf{e}_3 - k_T\bar{\omega}_i^2\mathbf{l}_i\times\mathbf{e}_3 \tag{IV.10}$$

and

$$\sum_{i=1}^4 T_i\mathbf{l}_i \times \mathbf{A}_{bla}\mathbf{v}_{\infty,i}^P = 0 \tag{IV.11}$$

due to symmetrical properties of the quadrotor. Assuming that the external aerodynamic forces cause no torque on the quadrotor then

$$\mathcal{T}_{aero} \approx 0 \tag{IV.12}$$

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

$$\begin{cases} m\ddot{\mathbf{r}} = T\mathbf{R}\mathbf{e}_3 - mg\mathbf{e}_3 + T\mathbf{R}\mathbf{A}_{bla}\mathbf{R}^\top(\dot{\mathbf{r}} - \mathbf{v}_w) + k_{par}\|\dot{\mathbf{r}} - \mathbf{v}_w\|(\dot{\mathbf{r}} - \mathbf{v}_w) \\ \dot{\mathbf{R}} = \mathbf{R}\boldsymbol{\omega}_\times \\ \mathbf{J}\dot{\boldsymbol{\omega}} = -\boldsymbol{\omega}_\times\mathbf{J}\boldsymbol{\omega} + \mathcal{T} \end{cases} \tag{IV.13}$$

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

$$\begin{pmatrix} T \\ \mathcal{T} \end{pmatrix} = \begin{pmatrix} k_T & k_T & k_T & k_T \\ k_Tl_y & k_Tl_y & -k_Tl_y & -k_Tl_y \\ -k_Tl_x & k_Tl_x & k_Tl_x & -k_Tl_x \\ k_\tau & -k_\tau & k_\tau & -k_\tau \end{pmatrix}\begin{pmatrix} \bar{\omega}_1^2 \\ \bar{\omega}_2^2 \\ \bar{\omega}_3^2 \\ \bar{\omega}_4^2 \end{pmatrix}. \tag{IV.14}$$

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.

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]. Paparazzi is widely used at TUDelft and it was chosen since 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. The vertical loop is responsible for the commanded thrust while the horizontal loop is responsible for the commanded Euler angles. Afterwards, the Motor-Mixing Unit (see Equation (IV.14) is responsible for translating those commands into the four motors rotating speeds.

[1] http://www.parrot.com/products/bebop-drone/
[2] http://wiki.paparazziuav.org/wiki/Main_Page

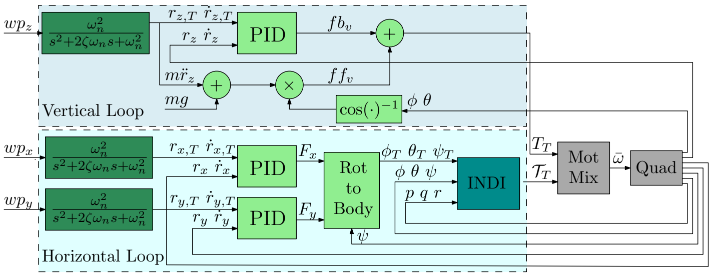

Figure IV.3: Illustration of a standard controller.

1) Vertical and Horizontal Control Loops in Paparazzi: The vertical control loop is composed by a reference generator and a vertical controller. The reference input is the z vertical setpoint, a discrete value of desired altitude, and the output is the desired thrust. The horizontal control loop is similar but also includes a stabilization controller. The reference inputs are the x and y lateral setpoints and the outputs are the desired Euler angles.

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). Both Control Loops can be seen in Figure IV.3.

2) Incremental Non-Linear Dynamic Inversion: The INDI controller is an attitude based controller developed by Smeur et al. [2016] in TUDelft for the Parrot Bebop. The INDI approach is an improvement of the Non-Linear Dynamic Inversion in order to increase 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 INDI inputs are the Euler angles and derivatives, together with the body rates and it outputs the rotors torque (see Equation (IV.14)). 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

Due to the loops separation in Paparazzi the coding implementation firstly performs the horizontal loop and afterwards the vertical one. The only part in which the loops may be seen as linked is through the $(\cos(\phi)\cos(\theta))^{-1}$ term in the vertical loop, although these angles being the measured ones, and not the desired. This separation will then 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 obviously 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 instead of a low-pass filter there will be optimal trajectories, which minimize the jerk power. *A priori* information of the desirable velocity and acceleration is used so that there is no theoretical delay, which does not happen when a low-pass filter is implemented. Moreover, a thrust vectoring implementation allows to determine the desirable orientation of the quadrotor so that the thrust vector, in the direction of $\mathbf{z}^B$, 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 Mellinger and Kumar [2011], but divergences exist, which will be further described. Extending the PID approach of the traditional controller to include a double-derivative term for the acceleration and incorporating aerodynamic effects one has

$$\begin{aligned}\mathbf{F}_T = K_p\mathbf{e}_p + K_{pi}\int_0^{\Delta t}\mathbf{e}_p\ \mathrm{d}t + K_v\mathbf{e}_v + K_a\mathbf{e}_a + mg\mathbf{z}^W + \\ + m\ddot{\mathbf{r}}_T + T\mathbf{R}\mathbf{A}_{bla}\mathbf{R}^\top(\dot{\mathbf{r}} - \mathbf{v}_w) + k_{par}\|\dot{\mathbf{r}} - \mathbf{v}_w\|(\dot{\mathbf{r}} - \mathbf{v}_w)\end{aligned} \tag{IV.15}$$

with $\mathbf{F}_T$ the desirable thrust. The first four terms are feedback related and the others are feedforward related. Equation (IV.15) shows the first divergence with Mellinger and Kumar [2011], since here the drag terms are accounted, whereas they neglect them. The feedback errors are obtained as

$$\begin{pmatrix}\mathbf{e}_p\\ \mathbf{e}_v\\ \mathbf{e}_a\end{pmatrix} = \begin{pmatrix}\mathbf{r}_T - \mathbf{r}\\ \dot{\mathbf{r}}_T - \dot{\mathbf{r}}\\ \ddot{\mathbf{r}}_T - \ddot{\mathbf{r}}\end{pmatrix}. \tag{IV.16}$$

The desirable orientation of the quadrotor $\mathbf{z}^B$ axis is given by

$$\mathbf{z}_T^B = \frac{\mathbf{F}_T}{\|\mathbf{F}_T\|} \tag{IV.17}$$

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

$$\mathbf{x}_T^\psi = \mathbf{R}_z\mathbf{e}_1 = \begin{bmatrix}\cos(\psi_T) & \sin(\psi_T) & 0\end{bmatrix}^\top \tag{IV.18}$$

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

$$\mathbf{y}_T^B = \frac{\mathbf{z}_T^B \times \mathbf{x}_T^\psi}{\|\mathbf{z}_T^B \times \mathbf{x}_T^\psi\|} \tag{IV.19}$$

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

$$\mathbf{x}_T^B = \mathbf{y}_T^B \times \mathbf{z}_T^B. \tag{IV.20}$$

Equation (IV.19) is only valid if $\mathbf{z}_T^B \times \mathbf{x}_T^\psi \neq 0$ meaning that the axes are not collinear. In order for this to happen the quadrotor should be rotated on its side, which for practical reasons is never the case. The three axes define the desirable rotation matrix in the same way as in Equation (IV.2) as

$$\mathbf{R}_T = \begin{pmatrix}\mathbf{x}_T^B & \mathbf{y}_T^B & \mathbf{z}_T^B\end{pmatrix} \tag{IV.21}$$

and the other desirable Euler angles can finally be obtained using

$$\begin{pmatrix} \phi_T \\ \theta_T \end{pmatrix} = \begin{pmatrix} \operatorname{atan2}\left(\mathbf{R}_{T,32}, \mathbf{R}_{T,33}\right) \\ \operatorname{atan2}\left(-\mathbf{R}_{T,31}, \sqrt{\mathbf{R}_{T,32}^2 + \mathbf{R}_{T,33}^2}\right) \end{pmatrix} \tag{IV.22}$$

Equation (IV.22) shows the other major divergence from Mellinger and Kumar [2011], since here the angles are computed from the rotation matrix and inputted to the INDI stabilizer. In their case, they compute the *vee map* error between $\mathbf{R}_T$ and $\mathbf{R}$, as well as the error in the angular velocities $\boldsymbol{\omega}_T$ and $\boldsymbol{\omega}$ to feed directly to the inputs $\mathcal{T}_1$, $\mathcal{T}_2$ and $\mathcal{T}_3$. 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

$$T_T = \mathbf{F}_T \cdot \mathbf{z}^B \tag{IV.23}$$

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, as in Figure IV.4.

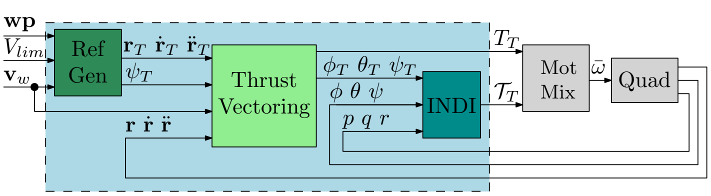

Figure IV.4: Illustration of the proposed Cascade Controller.

V. Results

In this Section the results of the work in hands will be analyzed, in a fashion that follows the previous Sections sequence. Firstly, in Section V-A, the lumped and parasitic drag coefficients will be identified through flight data, and their influence on the controller will be shown in Section V-B. Next, the optimal approach when solving the TSP will be determined in Section V-C, the influence of the wind in the time trajectory generation will be analyzed in Section V-D, and in Section V-E an overview of the trajectory formulation in computational times will be discussed. Later on, in Section V-F, the time optimal trajectory generation is compared versus a standard low-pass reference generator. Finally, in Section V-G, the thrust vectoring controller is compared with a standard PID controller.

A. Identification of the Drag Coefficients

In order to determine the coefficients related with aerodynamic properties of the quadrotor, discussed in Section II, a set of flight tests were conducted. The purpose of these tests is to identify the lumped drag coefficient, $\mathbf{A}_{bla}$ that combines blade flapping and induced drag, and the second order drag coefficient, k_{par}.

1) Proposed Identification Equations: Omari et al. [2013] 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, by including it in Equation V.1. In order to obtain the coefficients, Equation (IV.13) was used, and the movement of the quadrotor was constrained to the x axis direction, which allows some crucial simplifications. By restraining the movement, and assuming equilibrium of forces, the only non-zero Euler angle is the the pitch angle θ, and Equation (IV.13) results in

$$m \begin{bmatrix} a_x \\ 0 \\ 0 \end{bmatrix} = T \begin{bmatrix} s\theta \\ 0 \\ c\theta \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} + Ta_{bla} \begin{bmatrix} c^2\theta \\ 0 \\ -s\theta c\theta \end{bmatrix} v_x + k_{par}\|v_x\| \begin{bmatrix} v_x \\ 0 \\ 0 \end{bmatrix} . \tag{V.1}$$

Furthermore, using the small angles approximation for θ ($\cos(\theta) \approx 1$, $\sin(\theta) \approx \theta$) and assuming that

$$Ta_{bla}\theta << T - mg \tag{V.2}$$

a force equilibrium is achieved in the z axis naturally when $T = mg$, thus leading to the equation in the x axis as

$$ma_x = mg\theta + mga_{bla}v_x + k_{par}\|v_x\|v_x = mg\theta + D_x \tag{V.3}$$

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 of the collected flight data the specific drag force was determined as

$$d_{x,i} := \frac{D_{x,i}}{m} = a_{x,i} - g\theta_i \quad \left[\frac{\mathrm{m}}{\mathrm{s}^2}\right] \tag{V.4}$$

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

$$J = \sum_{i=1}^{N} \left(d_{x,i} - \hat{d}_{x,i}(v_{x,i})\right)^2 = \sum_{i=1}^{N} \left(d_{x,i} - g\hat{a}_{bla}v_{x,i} - \frac{\hat{k}_{par}}{m}\|v_{x,i}\|v_{x,i}\right)^2 \tag{V.5}$$

producing the following estimate of the drag coefficients

$$\begin{pmatrix} g\hat{a}_{bla} \\ \frac{\hat{k}_{par}}{m} \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{N} v_{x,i}^2 & \sum_{i=1}^{N} \|v_{x,i}\|v_{x,i}^2 \\ \sum_{i=1}^{N} \|v_{x,i}\|v_{x,i}^2 & \sum_{i=1}^{N} \|v_{x,i}\|^2 v_{x,i}^2 \end{pmatrix}^{-1} \begin{pmatrix} \sum_{i=1}^{N} v_{x,i}d_{x,i} \\ \sum_{i=1}^{N} \|v_{x,i}\|v_{x,i}d_{x,i} \end{pmatrix} \tag{V.6}$$

with N the number of samples.

2) Methodology: For the experimental tests, the tracking position was set to be a ramp, resulting in a step in velocity and null acceleration. Due to the drag forces, an expected pitch angle can be computed. An illustration can be seen in Figure V.1 for a selected flight with velocity $V_T = 2$ [m/s]. The movement can be qualitatively described as follows: the quadrotor starts at the center of the test arena and moves diagonally to the corner, so it can maximize the flight distance; it does this at a constant negative V_T; then the quadrotor waits six seconds to stabilize its position and orientation; afterwards it moves to the other corner at positive V_T. Due to the space limitations in the test arena, the corners positions were restricted to $r_x \in \{\pm 3.0,\ \pm 3.5,\ \pm 4.0\}$ [m] accordingly to V_T, which 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 average result for $V_T = 2$ [m/s] can be seen in Figure V.1, which allow to withdraw some conclusions:

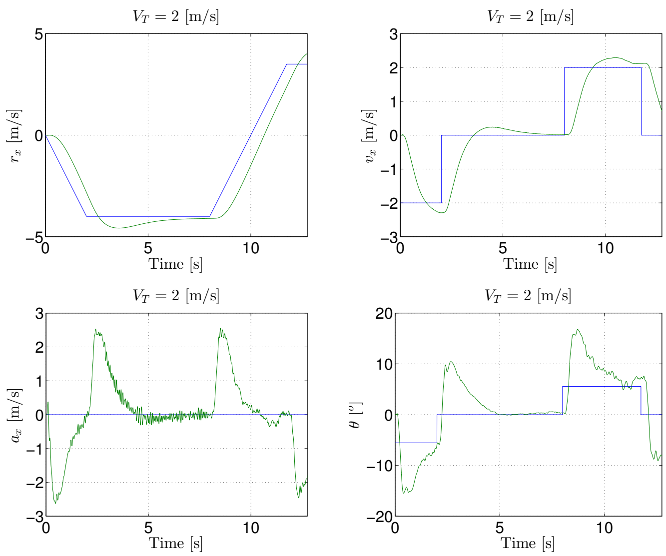

Figure V.1: Position, velocity, acceleration and pitch angle response with $V_T = 2$ [m/s].

- The position request is fulfilled with a constant delay and some overshoot, but with null steady-state error;
- The velocity request is fulfilled also with some delay but with worst performance relatively to the steady-state error;
- The acceleration and the pitch angle appear to compensate for each other (apart from a scale factor) when there is a discontinuity in the velocity, but later the acceleration goes to zero while the pitch angle goes to a constant value.

These conclusions are in line with the previous predictions, mainly the constant pitch angle when there is no acceleration, due to the compensation of the drag forces. The results of the LS fitting can be seen in Figure V.2, with the identified coefficients

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

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.2.

B. Influence of the Lumped Drag Coefficient in the Controller

To see the influence of the lumped linear drag coefficient previously identified in the controllers, trajectories of the same type as in the the previous Section V-A2 were tested. Two controllers were used, that differ only in Equation (IV.15). The manipulated variable was thus a_{bla}, with $a_{bla} = -0.0476 \vee a_{bla} = 0$ [s/m]. The second order parasitic drag coefficient was neglected for both.

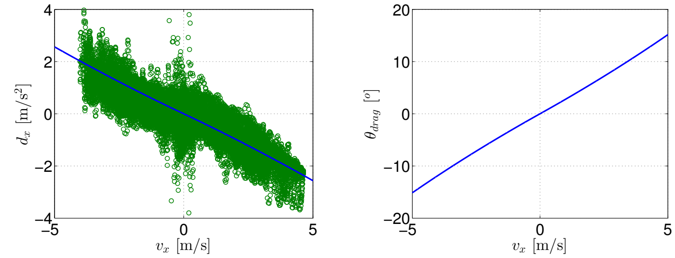

Figure V.2: LS fitting to obtain the drag coefficients (left) and necessary pitch angle to compensate for the drag force (right).

1) Results - Influence of the Lumped Drag Coefficient: Figure V.3 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. It is noted that the velocity is higher than the desirable, but that fact is explained by the position error.

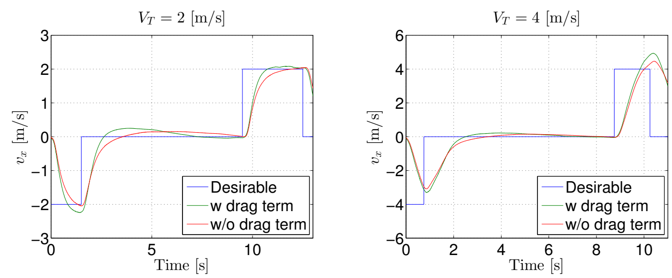

Figure V.3: Velocity for $V_T = 2, 4$ [m/s] (left, right).

Figure V.4 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 and should be accounted in order to increase the performance of the overall controller, thus sustaining the hypothesis that the drag terms should have a significant influence in the controller design.

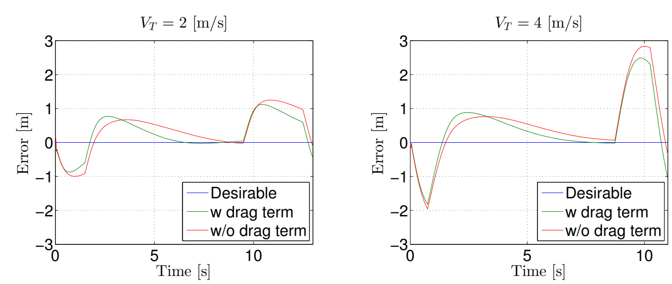

Figure V.4: Position error for $V_T = 2, 4$ [m/s] (left, right).

C. Optimal Sequencing - TSP Solution

This Section aims at comparing the performances of the alternatives in order to solve the TSP (see Section III-A2). The methodology will be described in Section V-C1 and the results will be shown in section V-C2.

1) Methodology: Unless stated otherwise, the experiments were set to randomly generate waypoints in a $50 \times 50 \times 50$ [m^3] box, while randomizing the wind velocity as fractions of the limit velocity, with a random angle $\alpha_1 \in [0, 360]$ [^{o}]. The value of K_{heu} is set to $K_{heu} = 0.9$, as it will be discussed later when optimizing K_{heu}. The experiments were performed in a 64 bits Windows 7 OS, running MATLAB 2013a, with an Intel i5-2430M 2.40 GHz CPU processor and 4.00 GB of memory.

2) Results - Optimal Sequencing: Firstly, the approaches are compared in terms of computational times (CPU) and problem dimension, resulting that the heuristic h_2 performs the best. However, this heuristic is not admissible, so an analysis in the difference between the heuristic h_2 solution and the optimal solution is accounted. Finally, the optimal value of K_{heu} is determined by means of a sensitivity analysis.

Computational Times: The computational times can be seen in Table V.1. One can see that for small m the permutations approach works well, even when comparing with the second heuristic h_2. Nevertheless, after $m = 5$ the second heuristic h_2 performs much better. Moreover, an increase in performance is achieved from uniform search, passing to heuristic h_1 and up until the heuristic h_2. Only with the second heuristic method it is reasonable to generate trajectories with large m, in particular greater than $m = 8$.

Table V.1: CPU Times for the different approaches.

		CPU Time [ms]			
		Per.	Uni.	h_1	h_2
	1	0.175	0.306	0.330	0.448
	2	0.295	0.729	0.763	1.112
	3	0.501	1.853	1.879	1.971
	4	1.339	7.097	5.413	4.022
m	5	6.061	34.95	25.62	7.197
	6	42.26	262.0	136.8	11.89
	7	413.7	5101	1816	19.36
	8	3159	-	20282	19.84
	9	-	-	-	20.58

Problem Dimension: The computational times are dependent on the computer in which the simulations are performed and code efficiency. Although the first aspect may be irrelevant, since all simulations were done in the same computer, the second is not. The code for all three search algorithms is similar but it is structured very differently than the permutations method, which uses MATLAB predefined functions. For these reasons, the computational times may not be a fair comparison criteria. With that in mind, another criteria is the problem dimension, a measure of its complexity. For the permutations method, this is measured as all the possible permutations, and for the search methods this is measured as the nodes in the open list. The results can be seen in Table V.2, which clearly shows the incremental increasing in performance from the permutations method up until the heuristic h_1 method, and a big increase in performance towards the heuristic h_2.

Trajectory Optimality: The second heuristic h_2 is not admissible and thus the resulting solution is not always optimal. The average error in determining the trajectory time is displayed in Table V.3, when comparing with the optimal solution. Values for m larger than $m = 8$ are not available since the optimal solution is obtained using one of the first three methods. It is possible to see that the error increases as m increases.

Table V.2: Problem dimension for the different approaches ($k_{h_1} < k_u < 1$).

		Problem Dimension [-]			
		Per.	Uni.	h_1	h_2
	1	1	1	1	1
	2	2	2	2	2
	3	6	6	6	5
	4	24	23	22	12
m	5	120	108	92	25
	6	720	536	411	44
	7	5040	2847	1963	61
	8	40320	-	8375	70
	9	-	-	-	75
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
	m	$\mathcal{O}(m!)$	$\mathcal{O}(k_u m!)$	$\mathcal{O}(k_{h_1} m!)$	$\mathcal{O}(m^2)$

Although it seems like the error is uncontrollably increasing, this is not the case, since the above experiments where taken for a random distribution of waypoints. When the waypoints are well distributed, which is the case for most applications, this heuristic gives better results, in particularly giving the correct solution, as it will be seen later in Section V-D.

Table V.3: Trajectory Time error for the heuristic h_2.

		T [s]		Error [%]
		Optimal	Heuristic h_2	
	1	7.63	7.71	1.09
	2	15.54	15.90	2.30
	3	22.21	23.30	4.92
	4	28.23	30.44	7.82
m	5	33.27	37.17	11.74
	6	37.99	43.44	14.35
	7	42.88	49.00	14.26
	8	46.97	54.71	16.47

Optimal K_{heu}: To determine the optimal value of K_{heu}, which represents a trade-off between the optimal solution and the computational time, multiple values of K_{heu} were studied in a sensitivity analysis. The comparison is being done with an average wind speed V_w of half the limit velocity V_{lim}, which is already a considerable amount of wind, and the results can be seen in Figure V.5. For each K_{heu} the CPU times and trajectory times are determined for up until $m = 8$ waypoints, which reflects the increase in the trajectory time. When K_{heu} decreases the computational time increases but the average trajectory time decreases, and vice-versa. A good trade-off between trajectory optimality and CPU time is achieved for $K_{heu} = 0.90$.

D. Time Optimal Trajectories

This Section aims at showing the smooth time optimal planned trajectories after the three steps of the proposed approach are performed. It will be seen results for the wind influence when it is symmetrically and asymmetrically distributed around the origin.

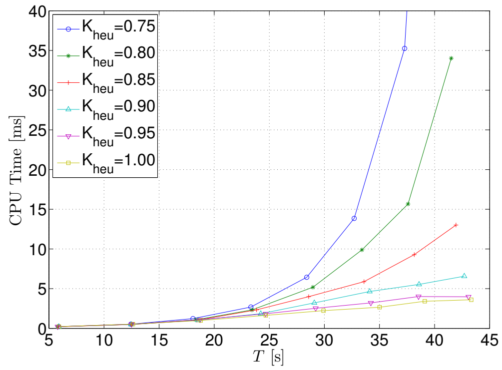

Figure V.5: Influence of K_{heu} in the trajectory time versus the computational (CPU) time.

Wind on a Symmetric distribution of Waypoints: Symmetrical wind is tested using $V_w = V_{lim}/3$, $V_{lim} = 3$ [m/s], and for four symmetric wind directions, only in the horizontal plane, with $\alpha_1 \in \{20,\ 110,\ 200,\ 290\}$ [^{o}] (shifted 90 [^{o}] in each quadrant). The results are shown in Figure V.6. The time intervals are obtained by solving the TSP and are then optimized as previously referred. Due to the adequate distribution of waypoints around the origin, the solution of the TSP either using an optimal method or using the heuristic h_2 are the same, which sustain the hypothesis given in Section V-C.

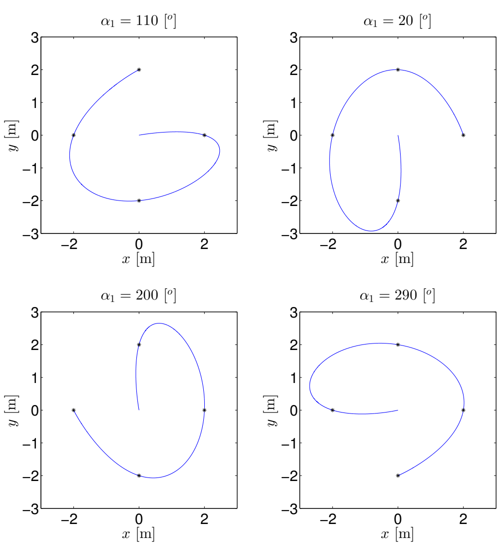

Figure V.6: Optimal trajectory for a symmetric distribution of $m = 4$ waypoints, with multiple wind directions.

The resulting trajectories are also symmetric and one can confirm the critical importance of wind when choosing the optimal sequence of waypoints. This is a result from the TSP and is illustrated here. The trajectories appear to be smooth and logical. The resulting velocity w.r.t. the airstream, acceleration and jerk are shown in Figure V.7 . It is validated that the maximum $V_\infty = 3$ [m/s], while the average velocity $V_{ave} = 2.1$ [m/s]. The total trajectory time is $T = 6.4$ [s], for every wind direction. The jerk is minimal by definition leading to accelerations of 4 [m/s^2] which are clearly acceptable and within limits used, for instance, by Mueller et al. [2013, 2015], Hehn and D'Andrea [2015].

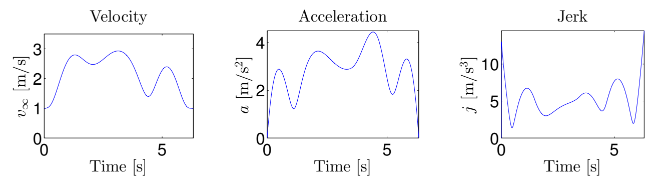

Figure V.7: Optimal velocity, acceleration and jerk.

Wind on an Asymmetric distribution of Waypoints: When the waypoints are not symmetrically distributed around the origin, the trajectories for different wind directions are not symmetrical. The results are shown in Figure V.8 and Table V.4. Once again the trajectories appear to be logical for passing through the set of waypoints, and the sequences appear to be natural as well. This qualitatively validates the use of the heuristic h_2 for selecting the waypoints sequence, because since $m = 15$ is relatively big, the optimal sequence could not be computed using the other three methods.

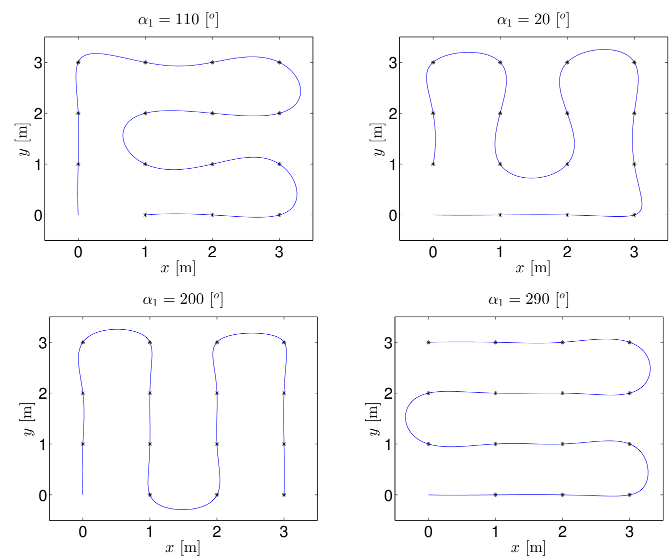

Figure V.8: Optimal trajectory for an asymmetric distribution of $m = 15$ wayoints, with multiple wind directions.

Due to the restriction on the maximum velocity, the trajectory times are different. In particular, the first two angles will create usually back wind, which allows faster trajectories, and the last two angles will create usually front wind, which allows slower trajectories. Thus, in environments where wind exists and it is known, the selection of the sequence of waypoints is critical to reduce the trajectory time and obtain the time optimal trajectory.

Table V.4: Trajectory times for multiple wind directions for an asymmetric distribution of waypoints.

α_1 [^{o}]	20	110	200	290
T [s]	8.2466	8.2160	9.5278	9.0250

E. Trajectory Planning Overview

It was proposed to divide the trajectory planning into three smaller problems, or steps, in order to reduce the complexity associated with finding the optimal sequence. Therefore, an initial simplification was used leading to the following question: "Is the initial obtained sequence the optimal one, after all steps?". It was checked that the sequence solution of the TSP is usually the best after all steps, but in some specific cases gives a trajectory time that is 5-10 [%] over the real optimal. Nevertheless, the CPU times when solving he TSP problem are negligible when compared to the other two next steps. There is a difference of milliseconds to seconds, three orders of magnitude. Therefore, by simplifying into three smaller steps, the overall CPU time is reduced by $\mathcal{O}(m!)$ and the purpose of the simplification is still satisfied, since the trade-off between CPU time and trajectory optimality is positive.

F. 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.9, is smooth and similar in shape to what has been seen in the previous Sections, whereas the standard trajectory consists of straight lines with some overshoot after the waypoints.

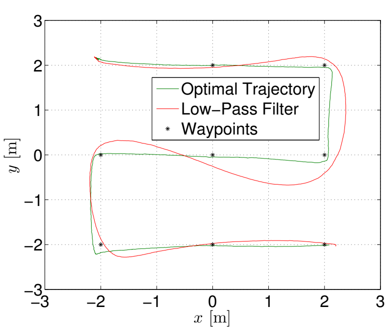

Figure V.9: Comparison of the 2D trajectory.

Velocity magnitude: In Figure V.10 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: there is an initial increase in velocity until steady state is reached; then the quadrotor senses that is near the next waypoint and the reference is changed; due to the low-pass filter the velocity reduces; the process repeats. Thus 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.

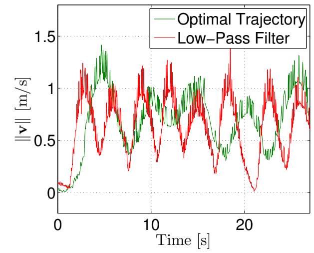

Figure V.10: Caparison of the velocities magnitudes.

Body velocity: Since the yaw angle ψ_T is constant in the standard case, depending on the turn direction, the body velocity behavior will be separated in both axes. This behavior can be seen in Figure V.11, where one can see a big velocity fluctuation for both axes. For the optimal case, the yaw angle ψ_T follows the desirable velocity direction and thus the velocity only varies in the x body axis. For the y body axis the velocity is near zero. Thus, the velocity variations are reduced in half and only one axis direction is used, increasing the stability.

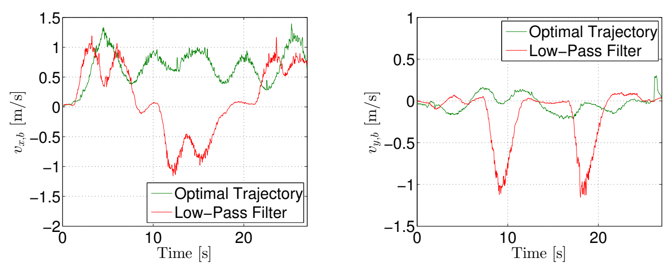

Figure V.11: Comparison of the velocity in x (left) and y (right) body axes directions.

Euler angles: The velocities contrasts have influence evidently on the accelerations and thus on the Euler angles, which can be seen in Figure V.12. 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. For the standard reference generator $\phi \in [-3, 6]$ [^{o}] and $\theta \in [-6, 6]$ [^{o}], whereas in the optimal generator $\phi \in [-2, 2]$ [^{o}] and $\theta \in [-2, 4]$ [^{o}]. Therefore, a save in angles variation of 50 [%] is achieved when the optimal trajectory generator is used.

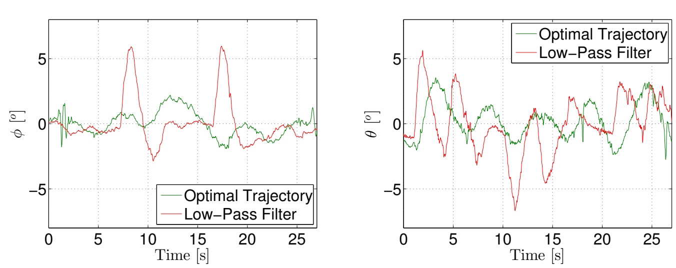

Figure V.12: Comparison of the Euler angles (ϕ left and θ right).

Thrust: Eventually the summed effects of the velocities, accelerations and angles have influence in the energetic performance of the quadrotor. As discussed in Section III-B4, it was chosen to minimize jerk since it is a better representative of the aggressiveness of the system inputs, the rotors angular velocities, and thus it is linked to thrust from Equation IV.23. Figure V.13 compares the thrust for both reference generators and shows that in fact 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, thrust is reduced from T_h plus 1.3 [%] to T_h plus 0.3 [%] when the optimal reference generator is chosen, meaning a reduction in thrust variation of 77 [%]. This Figure also shows that the thrust during a flight is approximately equal to the hovering thrust, within 5 [%] of thrust variation, validating the hypothesis that almost all thrust goes to compensate the gravity force.

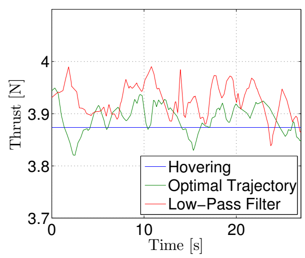

Figure V.13: Comparison of the Thrust.

G. Thrust Vectoring versus Standard PID Controller

This Section aims at comparing the performance of a standard 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.14-V.18 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.14 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

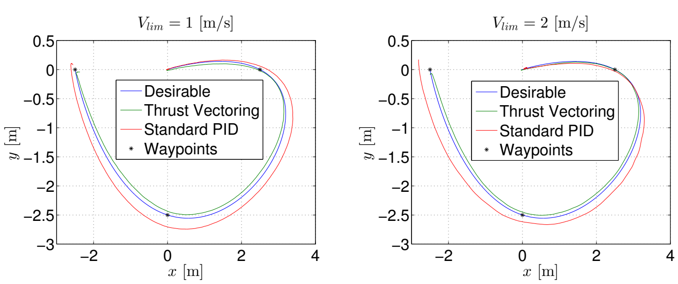

Figure V.14: Comparison of the 2D trajectory.

Horizontal Position: For the PID controller it seems that the performance increases when the velocity increases, but this is a false conclusion caused by the Figure type. Seeing Figure V.15 one can conclude that in fact the performance of the PID controller worsens, whereas in the proposed controller the performance is relatively the same (note the different time scales), and a reduction of one second time delay is achieved. The delay caused in the PID controller is mainly due to the integrator term. The integrator role is to compensate the steady state error mainly caused by the drag term, but since the trajectory is aggressive, as it is always changing, the integrator can not perform fast enough. The results are similar for the y axis, so they are not displayed.

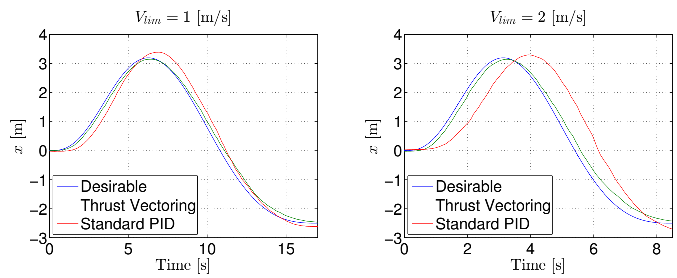

Figure V.15: Comparison of the x trajectory.

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.16, 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 standard PID controller the tracking never truly happens. Note the different scale when comparing with Figure V.15. This is a result of the vertical and horizontal loops inclusion into one single stage.

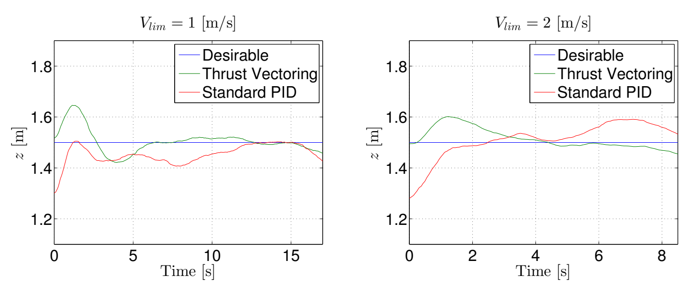

Figure V.16: Comparison of the z trajectory.

Euler Angles: The final results (Figures V.17 and V.18) show the performance in terms of the Euler angles. With the differential flatness outputs and derivatives, and 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. 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.18). 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.

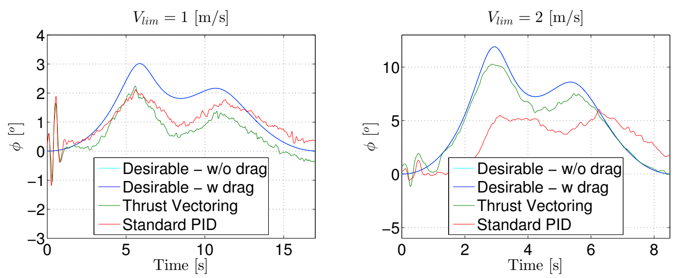

Figure V.17: Comparison of the roll angle ϕ.

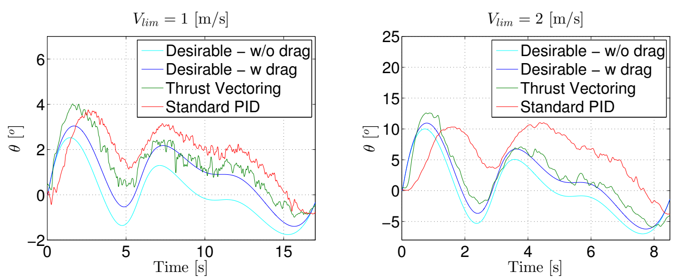

Figure V.18: Comparison of the pitch angle θ.

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. It was proven that the steps division saves computational time, reducing it approximately by an $\mathcal{O}(m!)$ order, while still maintaining acceptable time optimality. The QP was constructed and solved, showing results that indicate acceptable values for acceleration and jerk when compared to related work. 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 70 [%], 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.

VII. Future Work

The following future work is proposed, which covers the three main aspects of this work. For the aerodynamic effects, it is advised to test the quadrotor at higher velocities to obtain a relevant parasitic drag coefficient, as well as the identification of the thrust variation coefficients, extending the model to include them. For the optimal sequencing, it is recommend a study to explore the less optimal results of the TSP. In particular by verifying the increase in time optimality versus the increase in CPU time to be achieved with them, after the QP problem when assuring the constraint on the maximum velocity. Finally, it is proposed to test the performance of the thrust vectoring controller with higher aggressive maneuvers, higher velocities, and with 3D movement. This will allow to see the real extension of the possible flight envelope.

References

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.

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

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.

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

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

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.

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.

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

J. B. Corbets and J. W. Langelaan. Parameterized optimal trajectory generation for target localization. In *Guidance, Navigation and Control Conference*, August 2007.

T. Flash and N. Hogan. The coordination of arm movements: An experimentally confirmed mathematical model. *The Journal of Neuroscience*, pages 1688–1703, jul 1985.

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

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

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.

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.

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.

M. Kawato, Y. Maeda, Y. Uno, and R. Suzuki. Trajectory formation of arm movement by cascade neural network model based on minimum torque-change criterion. *Biological Cybernetics*, pages 275–288, 1990.

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.

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

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

S. Lorenz and F. M. Adolf. *A Decoupled Approach for Trajectory Generation for an Unmanned Rotorcraft*, chapter 1. Springer Berlin Heidelberg, 2010.

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.

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.

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.

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.

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

M. W. Mueller, M. Hehn, and R. D'Andrea. A computationally efficient algorithm for state-to-state quadrocopter trajectory generation and feasibility verification. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3480–3486, November 2013.

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.

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.

C. H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity*. Dover, 1998.

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

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

C. Rego, D. Gamboa, F. Glover, and C. Osterman. Traveling salesman problem heuristics: leading methods, implementations and latest advances. *European Journal of Operational Research*, June 2011.

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

M. Ryll, H. H. Bülthoff, and P. R. Giordano. A novel overactuated quadrotor unmanned aerial vehicle: Modeling, control, and experimental validation. *IEEE Transactions on Control Systems Technology*, pages 540–556, March 2015.

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.

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.

M. J. van Nieuwstadt and R. M. Murray. Real time trajectory generation for differential flat systems. *International Journal of Robust and Nonlinear Control*, May 1997.

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

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.