UNIVERSITY OF SOUTHAMPTON

Faculty of Social Sciences
School of Mathematical Sciences

Optimal Control and Routing of Autonomous Flying Vehicles

by

Luigi Bobbio
ORCID: 0000-0001-7123-6449

A thesis submitted in partial fulfillment for the
degree of Doctor of Philosophy

May 2024

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

Faculty of Social Sciences
School of Mathematical Sciences

Doctor of Philosophy

Optimal Control and Routing of Autonomous Flying Vehicles

by Luigi Bobbio

Autonomous flying vehicles are becoming more and more prominent nowadays and have manifold applications, some only about to emerge: collecting traffic data, surveillance and security, disaster management, wildlife observation, delivery services, and defence. Providing such vehicles with enough computational intelligence to automatically steer according to given objectives is a challenging task, only recently tackled with control algorithms working in soft-real time and computationally feasible for the limited on-board resources. Researchers and practitioners historically address the problem by separately concentrating on the two sub-problems of vehicle routing (VR) and trajectory optimisation (TO), often neglecting or oversimplifying vehicles dynamic. In this work we introduce methodologies contributing to the newborn research area unifying the two aspects in a single Vehicle Routing and Trajectory Optimisation problem (VRTOP), also taking into account a sufficiently accurate representation of vehicles dynamic and providing solution algorithms to the overall problem. First, a model for the VRTOP is proposed where trajectory constraints are linearised through a Taylor expansion around a known solution to vehicles' equations of motion within predefined settings. Distance constraints are also linearised to arrive at a Mixed-Integer Linear Programming formulation of the VRTOP with prior information. Second, this work provides approximation techniques to gain valuable insights and find initial trajectory guesses when prior knowledge is not available, overcoming the local nature of Taylor's approximation and resulting in enhanced and more robust solutions. Finally, a Two Step Solution Algorithm is proposed to integrate the two phase of the problem, enabling us to solve the VRTOP in absence of prior information. Numerical results showing the efficacy of the presented approach are furnished and further research developments are ultimately discussed.

Declaration of Authorship

I, Luigi Bobbio, declare that this thesis entitled ”Optimal Control and Routing of Autonomous Flying Vehicles” and the work presented in it is my own and has been generated by me as the result of my own original research.

I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University;
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
- Where I have consulted the published work of others, this is always clearly attributed;
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
- I have acknowledged all main sources of help;
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
- None of this work has been published before submission.

Signed:... Date:.................

Contents

List of Figures

List of Tables

Acknowledgements

I would like to express my gratitude to all the people that give me the possibility to embark in this incredible journey and shared it with me.

First, I would like to thank my family, my parents, my brother and my sister, who always support my decisions and well-being and with whom I shared beautiful life moment around the world. Likewise, my girlfriend must be thanked for never giving up and for rendering this period a 'ccezionale roller coaster, also offering new unpredictable life perspectives and new dreams.

A special thank goes also to my ex-manager, who helped me with the interviews and without whom I would not be here today. To my supervisors, who believed in me and gave me the opportunity to work on this amazing project. To my old Professors, who wrote my recommendations letters supporting my candidature.

I would then like to thank my friends, the ones who has always been there and came here to visit me, making some UK places become traditional. Thanks also to the new ones, with whom I lived the PhD life, before, during and after the pandemic, and who support me with the extra software licenses.

Finally, I would like to thank myself for my efforts and my stubbornness, that permitted me to persevere even when it did not seem to be the most rational and wise way of progressing.

Guarda il mondo come dentro Dio.

Chapter 1

Introduction

This manuscript aims at discussing about the development of models and algorithms for the path planning and optimal control of unmanned aerial vehicles, with particular focus on routing and trajectory optimisation problems. In this chapter, we provide the reader with an overview on unmanned aerial vehicles, including their origins and importance for the society, and we introduce the objectives and structure of the arguments presented.

1.1 Motivation

Unmanned aerial vehicles (UAVs) are defined as *aerial vehicles that does not carry a human operator on-board* and that can fly autonomously or can be piloted remotely. Quadcopters, remote controlled helicopters and unmanned gliders are all examples of UAVs, the latter differing from the other types due to the absence of on-board propulsion (i.e. an electric or combustion engine). Unmanned aerial vehicles made their first appearance in the late 19^{th} century, as reported in Buckley (1999) and McKenna (2016), being deployed for offensive military use in naval aviation, when the Austrian armies used incendiary balloons to carry and drop bombs over the besieged city of Venice. However, the offensive resulted in an undesired outcome where, due to a reversion in the wind direction after launch, balloons and bombs missed their target with some of them being pushed back over the Austrian lines. We need to wait then until 1917, during World War I, before seeing remotely controlled pilotless aircraft in action, as recorded by Low (1958). Since then, astonishing advancements have been made, with the use of unmanned aerial vehicles pervading several applications and areas. Cases where successful deployment of UAVs show their advantages and versatility have been already documented in target observation, Rysdyk (2006), traffic monitoring and management, Kanistras et al. (2013), meteorological sampling, Elston et al. (2015), aerial surveillance, Ruzgienė et al. (2015), forest fire detection, Yuan et al. (2015), geographic vigilance, Uysal et al. (2015), data collection, Stöcker et al. (2015), and last mile delivery, Wang et al. (2016), as well as disaster assessment and response, Nedjati et al. (2016), Coutinho et al. (2019), Chowdhury et al. (2017) and Laville et al. (2017), who report the Grenfell Tower disaster. In addition, Hayat et al. (2016), Farhan et al. (2014) and Shakhatreh et al. (2019) provide a review of a number of applications of UAV networks. The interested reader can also refer to extensive surveys on the evolving field of civil applications for UAVs and possible optimization approaches in Otto et al. (2018) and Yuan et al. (2015). Finally, future directions predicting the disruptive nature of UAVs used for last mile delivery are shown in Cornell et al. (2023) and a report on Drone Industry Insights (2022) claims for 237 applications of unmanned aerial vehicles. In the latter, UAVs' commercial market is forecasted to grow at a CAGR of 9.4% until 2026, where it is expected to generate over 41 billion USD.

However, despite all the innovations, UAVs remain predominantly costly in their present state and they frequently necessitate the (remote) control of experienced pilots for operation. Nonetheless, within aerial operations, a fleet of affordable autonomous gliders can be 3D printed, Keane et al. (2017), equipped with cameras and launched via balloons, Crispin (2016), potentially eliminating the need for a specialized team to operate them. This approach can serve as an initial foundation for enabling the utilization of swarms of autonomous flying vehicles, facilitating efficient, cost-effective and sustainable development of future applications. Equipping these swarms with adequate computational intelligence to autonomously navigate the individual vehicles based on predefined objectives poses a formidable challenge and represents the ultimate objective of our research project. Developing control algorithms that operate in soft-real time (minutes or seconds, rather than hours) and are computationally viable for the constrained resources of on-board computational units has only recently become possible. Still, the corresponding mathematical models and algorithms generally fail to provide the UAVs with the big picture, usually treating vehicle routing (VR) and trajectory optimisation (TO) aspect of the problem separately, as noted by Coutinho et al. (2018). According to Gasparetto et al. (2015), it is possible to see the VR as a geometric problem since its objective is to find a geometric path regardless of any defined time law, while on the other hand, TO involves the task of assigning a time law to a controlled geometric path. As of today, if in the routing literature there has been a tendency to overlook constraints arising from the flight dynamics of UAVs, the field of TO lacks of consideration for routing decisions and, while it is possible to search for a feasible trajectory for a generic UAV given a set of ordered waypoints, it remains unclear whether the previously computed sequence of waypoints is appropriate or optimal. In certain real-world applications involving sophisticated UAV systems, like unmanned gliders and fixed-wing vehicles, it becomes then crucial to integrate the route definition with the design of flyable trajectories. Failure to do so can result in inefficient or even infeasible routes for these UAVs. The intricate question of how to effectively combine routing and trajectory decisions into a unified optimization problem remains open, yet it is essential to guarantee the quality of the UAV routes.

Nowadays research is expanding and starts addressing the problem by leveraging on the utilisation of mixed-integer linear programming (MILP) and exact methods to find solutions. Fügenschuh and Müllenstedt (2015) and Schmidt and Fügenschuh (2023) proposed an integrated model for civil applications, considering vehicles presenting low flight autonomy. The authors design a way to simultaneously assign targets to an inhomogeneous fleet of UAVs, while finding the best trajectories to reduce the consumption of fuel. However, their approach considers a simplified dynamic and can be computationally expensive for large problem instances. Therefore, approaches involving the use of heuristic and metaheuristic algorithms that can provide good quality solutions in a reasonable amount of time might be sometimes preferred. Probabilistic and machine learning techniques have also been used to improve the performance of these methods. Zhang et al. (2012) propose an updatable probabilistic roadmap and a fast heuristic searching algorithm, emphasising the importance of integrating routing and trajectory optimization to ensure both the safety of the vehicle and the feasibility of the trajectories. The authors consider a scenario where a UAV is asked to visit a specific set of targets and once the UAV reaches a predetermined distance from an objective, it needs to modify its flight attitude to carry out a payload delivery. Following the successful delivery, the UAV is required to execute an escape maneuver preparing for the next shipment.

All this considered, embedding routing and UAV dynamic conditions into the planning process might result in a problem very complicated to solve. The focus of our research revolves around the modeling and integration of problems related to routing and trajectory optimization specifically for unmanned gliders. Our study involves a fleet of gliders that need to visit a predefined set of waypoints, while adhering to flight dynamics constraints. This problem is known as *the Glider Routing and Trajectory Optimization Problem* (GRTOP). The primary motivation for tackling this problem stems from aerial photographing missions conducted during disaster events. The missions require a fleet of gliders to capture images of objects located in various points of interest, such as hospitals, schools, residential areas, and inaccessible regions, in the least possible time. Nevertheless, the problem can be generalised to other different scenarios. In addition, the use of gliders is motivated by the implicit difficulties deriving from their dynamic - not having on board propulsion makes the problem harder to solve since the equations of motion cannot be reasonably oversimplified - making it also easier to generalise. At the same time, it is impossible to neglect that gliders might result in a significant help to the green economy. In this work, we propose a two-step algorithmic formulation where at first we describe a mixed-integer linear programming (MILP) model for the GRTOP when an initial trajectory estimate is provided. In a second moment, we provide methods and problem formulation to find the initial trajectory in a context of reduced information. Consequently, we joint the two models in a unique algorithmic formulation that allows to find paths that simultaneously optimise routes and flight trajectories along these routes while the flight dynamic of the gliders are modelled in details. Linearising and approximating dynamic effects described by the equations of motion, we are able to circumvent the handling of nonconvex constraints and reduce the problem to two mixed-integer linear programming subproblems, one utilised to find an initial trajectory and the second one to refine the initial solution, obtaining dynamically sound and feasible paths. Along the manuscript, we provide also computational experiments conducted using a substantial number of randomly generated instances that show the efficacy of the proposed models and algorithms.

1.2 Thesis Outline and Objectives

In what follows, we dedicate Chapter 2 to a more detailed review of the Vehicle Routing (VR) and Trajectory Optimisation (TO) Problems, highlighting the main aspects and possible solution approaches. In addition, we examine some of the methods proposed so far to tackle the VR and TO simultaneously, also proposing a generic mathematical formulation of the VRTOP. In Chapter 3, we aim at describing the equations governing glider's dynamics in detail, also providing methods used to solve them as well as examples of generated trajectories. Chapter 4 presents the Glider Routing and Trajectory Optimisation Problem (GRTOP), including a detailed and precise mathematical MILP formulation when an initial trajectory is known. A second model is then proposed in Chapter 5 to manage situations where guesses on the initial trajectory are unavailable. Along the chapter, we provide the reader with a Two Step Solution Algorithm for the GRTOP in absence of prior knowledge, integrating the second model into the established framework when initial information is available. Numerical examples validating the procedure and demonstrating the efficacy and reliability of the proposed approach are then shown in Chapter 6. Finally, Chapter 7 summarises conclusive remarks and discusses possible future research developments.

Chapter 2

Literature Review

In this chapter, we start examining more in deep the literature about the aforementioned (i) Vehicle Routing (VR) and (ii) Trajectory Optimisation (TO) problems, the latter seen as special case of Optimal Control (OC) problems. Here we discuss their typical mathematical description as well as possible solution approaches that are presented in the literature. Then, we discuss generic mathematical formulations for the combined Vehicle Routing and Trajectory Optimisation Problem (VRTOP), merging typical aspects of VR and TO. Finally, an overview of the newborn and developing research about the joint problem is provided.

2.1 The Vehicle Routing Problem

The problem of designing optimal delivery or collection routes from one or several depots to a number of geographically scattered cities or customers, subject to side constraints, is at the foundation of operational research and combinatorial optimisation and it is known as the *Vehicle Routing* Problem (VR or VRP here). The VRP is not only interesting as a mathematical problem per se, but it carries a plethora of possible applications relevant for businesses, governments and organisations which include cost and risk reduction, improvement of the supply chain, also helping the implementation of a "green" economy. It is introduced for the first time by the paper of Dantzig et al. (1954), where the authors study a relatively large scale Travelling Salseman Problem (TSP), a special case of the VRP, also proposing a possible solution method. From then onwards, many variants of the TSP are studied until 1972, when the first paper explicitly referring to the *vehicle routing* problem appears, see Golden et al. (1972). Many other versions of the VRP are then proposed between the 1970s and the 1980s, but, due to their computational complexity, they do not raise the deserved interested of the industrial and scientific communities. In the 1990s however, due to the terrific acceleration of computers capabilities and availability, researchers reignite the interest towards the problem, developing and proposing newer formulations and solution algorithms. Heuristic and metaheuristic methods for the VRP are proposed for the first time in these years, see for example Osman and Kelly (1996), Aarts and Lenstra (1997) and Cordeau et al. (2022). Dynamic and Stochastic versions of the VRP (DVRP and SVRP) also receive a boost, see Gendreau and Potvinm (1998), Larsen (2000), Gendreau et al. (1996) and Yaohuang et al. (2002). Due to this, it is easy to understand why nowadays the VRP constitutes a central pillar in operational research and combinatorial optimisation, with its class of problems presenting an extensive literature, as well as a broad spectrum of proposed solution approaches. We refer the reader to the surveys of Bodin et al. (1983), Christofides (1985), Laporte and Nobert (1987), Laporte (1990) and Laporte (1992) as well as the classification scheme introduced by Desrochers et al. (1990) and Eksioglu et al. (2009) for a more detailed study of the VRP, while in the following sections, we propose some of its mathematical formulations, also mentioning possible solution techniques frequently used in the literature.

2.1.1 Mathematical Problem Formulations

To describe the VRP, we can proceed as follows. Let $G = (V, A)$ be a graph where $V = \{1, \dots, n\}$ is a set of vertices representing a set of locations and $A \subseteq V \times V$ is a set of arcs. Lets $C = [c_{ij}]_{(i,j)\in A}$ be the matrix representing the non-negative costs $c_{ij} \geq 0$ required to travel from vertex i to vertex j and associated to arc (i, j). We note here that sometimes, when C is symmetrical, it is convenient to replace the set A with E, identifying arcs (i, j) and (j, i) with an undirected edge e, and define the corresponding cost matrix $C_E = [c_e]_{e\in E}$. Assume then that m vehicles based are available at the depot $d \in V$. It is often reasonable to associate a fixed cost $F_g \geq 0$ to the employment of a vehicle g, especially when some vehicle can remain unused. Then, the vehicle routing problem can be described as follows.

Definition 2.1. The VRP is the problem of designing a set of *least-cost vehicle routes* in such a way that

- all vehicle routes start and end at the depot $d \in V$;
- each city in $V \setminus \{d\}$ is visited exactly once by exactly one vehicle;

With this in mind, the VRP can be clearly associated to the multiple Travelling Salesman Problem (m-TSP), which is a widely studied problem in the literature, known to be NP-Hard, and whose mathematical formulations can be as follows. We define binary variables x_{ij} for each arc $(i, j) \in A$, with x_{ij} taking value 1 if arc (i, j) is included in a tour, while x_{ij} taking value 0 otherwise. We also define integer variables O_i to denote the position of vertex i in a tour and we define a value $q = 1, \dots, n$ to be the maximum number of vertices that can be visited by any vehicle. Thus, considering the number of vehicles employed to be fixed and equal to m, the VRP (or m-TSP) can then be written as

$$\begin{align}
\min \quad & \sum_{(i,j)\in A} c_{ij}x_{ij} \tag{2.1}\\
\text{subject to} \quad & \sum_{j\in V:(d,j)\in A} x_{dj} = m \tag{2.2}\\
& \sum_{j\in V:(j,d)\in A} x_{jd} = m \tag{2.3}\\
& \sum_{i\in V:(i,j)\in A} x_{ij} = 1,\ \forall j \in V\setminus\{d\} \tag{2.4}\\
& \sum_{j\in V:(i,j)\in A} x_{ij} = 1,\ \forall i \in V\setminus\{d\} \tag{2.5}\\
& O_i - O_j + qx_{ij} \le q-1,\ \forall (i,j)\in A, i,j \ne d, \tag{2.6}
\end{align}$$

where objective (2.1) requires to minimise the cost of the tours, equations (2.2) and (2.3) enforce the use of m vehicles starting and ending the tour at depot d, constraints (2.4) and (2.5) require that a node is visited once by exactly one vehicle and finally, like proposed in Miller et al. (1960), inequalities (2.6) guarantee that routes presenting subtours are excluded from the solutions.

It is worth introducing here another possible formulation of the VRP that considers the case where the number of vehicles employed is not predetermined. Having at disposal a set of vehicles $\Upsilon = \{1,\ldots,m\}$, we define variables x_{gij} equal to 1 if arc $(i,j) \in A$ is included in the tour of vehicle $g \in \Upsilon$, and 0 otherwise. We can note that basically we split the decision variables x_{ij} of the previous formulation to take into account the vehicle travelling through that road. Then, we can define new variables $\mathcal{Y}_g$ taking value 1 if vehicle g is used, while taking value 0 otherwise. In this case, since the number of used vehicles is not predetermined, we consider in the formulation the associated costs $F_g \ge 0$ incurred when vehicle g is utilized. All the rest remaining the same, the VRP can be formulated as the problem of minimising

$$\sum_{g\in\Upsilon}\left(F_g\mathcal{Y}_g + \sum_{(i,j)\in A} c_{ij}x_{gij}\right),$$

called the *total travel costs*, subject to the following constraints

$$\sum_{(i,j)\in A} x_{gij} \leq M\mathcal{Y}_g \tag{2.7}$$

$$\sum_{g\in\Upsilon} \mathcal{Y}_g \geq m_L \tag{2.8}$$

$$\sum_{i\in V:(i,j)\in A} \sum_{g\in\Upsilon} x_{gij} = 1, \ \forall j \in V \setminus \{d\} \tag{2.9}$$

$$\sum_{j\in V:(i,j)\in A} \sum_{g\in\Upsilon} x_{gij} = 1, \ \forall i \in V \setminus \{d\} \tag{2.10}$$

$$\sum_{j\in V:(d,j)\in A} x_{gdj} = \mathcal{Y}_g, \ \forall g \in \Upsilon \tag{2.11}$$

$$\sum_{j\in V:(j,d)\in A} x_{gjd} = \mathcal{Y}_g, \ \forall g \in \Upsilon \tag{2.12}$$

$$\sum_{i\in V:(i,j)\in A} x_{gij} = \sum_{i\in V:(j,i)\in A} x_{gji}, \ \forall j \in V, \forall g \in \Upsilon \tag{2.13}$$

$$O_i - O_j + q\sum_{g\in\Upsilon} x_{gij} \leq q - 1, \ \forall (i,j) \in A, i, j \neq d, \tag{2.14}$$

with M a sufficiently large number, possibly equal to the number of arcs, and m_L is a number between 1 and m, the latter representing the minimum number of vehicles to be used. Again constraints (2.9) and (2.10) require that a node is visited just once by exactly one vehicle, while inequalities (2.14) ensure the absence of subtours in the solutions. On the other hand, in place of constraints (2.2)-(2.3), we introduce inequalities (2.7)-(2.8), ensuring that the unused vehicles remain at the depot and that at least m_L vehicles are used, as well as equations (2.11)-(2.12) asking to the selected vehicles to start their tour exiting the depot and guaranteeing their return to the initial location. Finally, constraints (2.13) are defined, imposing the net flow and coherence of the tours: a vehicle g entering customer location j must also leave it.

The literature include a number of variants of the VRP, all distinguished by the side constraints required to be satisfied. Typical side conditions are:

- *capacity restrictions*: here each location $i \neq d$ is associated with a corresponding weight (or demand) $w_i \geq 0$. Each vehicle then has a maximum capacity Q and it can not satisfy a demand greater than its capacity, meaning that the sum of weights of any vehicle route may not exceed vehicle's capacity Q. This problem is normally referred to as the Capacity-constrained vehicle routing problem (CVRP). We can also observe that the above VRP can be considered as a special case of the CVRP where $w_i = 1$ for all $i \neq d$ and $Q = q$;

- *total time restrictions*: here the length of a route, measured as the sum of travel times/ costs c_{ij} and possibly of stopping times δ_i at each location i on the route, is limited by a predefined bound $L \geq 0$;
- *time windows*: here each customer location or node i is required to be visited within the time interval (or window) $[t_{0,i}, t_{f,i}]$. Waiting can also be allowed at a node location;
- *precedence relations between pairs of cities*: here it is required that some location i appears in the tour before another location j.

Other interesting variants can be found in Golden and Assad (1988) and Toth and Vigo (2002). As a last example, we would like to give here a possible integer linear programming formulation for the symmetric CVRP, proposed for the first time by Laporte et al. (1985). Lets consider C symmetrical and lets indicate with x_e the integer variables representing the number of times an edge e is included in the solution. Lets then denote with $r(S)$ the minimum number of vehicles needed to serve the locations in $S \subseteq V$. Also, for each $S \subset V$, let $\delta(S)$ be the set $\{(i,j) \mid (i \in S \text{ and } j \notin S) \text{ or } (i \notin S \text{ and } j \in S)\}$. If S is equal to the single location $\{i\}$, then we can simply write $\delta(i)$ instead of $\delta(\{i\})$. The proposed CVRP formulation is then

$$\min \sum_{e \in E} c_e x_e \tag{2.15}$$

$$\text{subject to} \sum_{e \in \delta(i)} x_e = 2, \ \forall i \in V \setminus \{d\} \tag{2.16}$$

$$\sum_{e \in \delta(d)} x_e = 2m \tag{2.17}$$

$$\sum_{e \in \delta(S)} x_e \geq 2r(S), \ S \subseteq V \setminus \{d\}, S \neq \emptyset \tag{2.18}$$

$$x_e \in \{0,1\}, \ e \notin \delta(d) \tag{2.19}$$

$$x_e \in \{0,1,2\}, \ e \in \delta(d), \tag{2.20}$$

where constraints (2.16) guarantee that each location is visited exactly once, while equation (2.17) enforces the utilisation of m vehicles. On the other hand, constraints (2.18) force a sufficient number of edges to enter in S, so ensuring that both subtour elimination and the vehicle capacity requirements are satisfied. Conditions (2.19) and (2.20) impose that edges connecting two different locations are used at most once, whereas edges between the depot d and any other node are used at most twice, with x_e being equal to 2 when a route visiting a single customer is selected. The value of $r(S)$ can be determined by solving a corresponding Bin Packing Problem (BPP) with item set S and bins of capacity Q. However, since the BPP is NP-hard, $r(S)$ may be approximated by any BPP lower bound.

In light of this broad overview on the VRP and some of its possible formulations, in the following section we would like to discuss some of the solution approaches commonly found in the literature.

2.1.2 Solution Approaches

The VRP is usually tackled using two different approaches: by means of exact algorithms, where a proven optimal solution to the (approximated) problem is found, or by using heuristic or metaheuristic methods, where the quality of the solution in terms of optimality or feasibility can not be always guaranteed. In fact, when heuristic algorithms are utilised for solving the VRP, extra care might be necessary to ensure the feasibility of the generated routes. Hybrid and combined approaches can also be found in the literature. We present here some of the solution procedures employed, also considering some of the potential applications stemming from real-life applications.

2.1.2.1 Exact Methods

Following Laporte and Nobert (1987) survey, in the literature it is possible to find many different exact algorithms solving the VRP that can mainly be classified in the categories: (i) direct tree search methods, (ii) dynamic programming, and (iii) integer linear programming. The first one leverages on a VRP formulation allowing the solutions of the problem to be organised in a tree. Consequently, a tree exploration strategy is deployed. The second one, is a mathematical optimisation method aiming at simplifying a complicated problem by breaking it down into simpler sub-problems and solving them in a recursive manner. The last one takes advantage of a mathematical formulation similar to the one proposed above where (some of) the variables are integers and constraints can be expressed in a linear form. Regardless the category, common techniques used to find the optimal solutions include *branch-and-bound*, *branch-and-cut*, *branch-and-price* or *set partitioning* with *column generation*.

Early works on exact methods for the VRP start emerging from the 1960s, when researchers recognised the challenges associated with goods distribution. Notable pioneers in this field include Balinski and Quandt (1964), proposing a set-partitioning formulation for VRP, and Eilon et al. (1971), who tackle the problem utilising a dynamic programming approach. Novel branch-and-bound and branch-and-cut approaches are then introduced by Christofides et al. (1981), Laporte et al. (1985) and Fischetti et al. (1994) among others. In the late 1990s instead, Applegate et al. (2000) suggest a cutting-plane method for solving bigger instances of the TSP, a special case of the VRP. The authors develop also an open source code, known as the Concorde TSP Solver of Applegate et al. (2000). In more recent years, Baldacci et al. (2004), Cordeau (2006), Baldacci et al. (2007) and Baldacci and Mingozzi (2009) propose exact algorithms based on set partitioning and using branch-and-cut techniques, also exploiting bounds stemming from the LP and Lagrangian relaxation of the VRP. A branch-and-cut-and-price algorithm is instead deployed by Pessoa et al. (2007), extending the method proposed for the CVRP by Fukasawa et al. (2006). The computational findings indicate that their solutions outperform the lower bounds suggested by Golden et al. (1984), Yaman (2006), and Choi and Tcha (2007) and reach optimality for problem instances with up to 75 vertices. Variants of the VRP flourished from then on, with the above techniques deployed in many different contexts and applications. However, due to their NP-hardness, VR problems remain challenging to solve exactly at scale. Therefore, heuristic methods gain popularity and are still at the core of research nowadays.

2.1.2.2 Heuristics

As introduced earlier, a heuristic method refers to any approach to problem solving that utilizes a practical procedure, which may not guarantee an optimal or feasible solution, but it is nonetheless effective in providing a prompt and satisfactory result or approximation. Heuristic algorithms tackling the VRP can be categorized into three primary classes: (i) two-phase, (ii) construction-based, and (iii) iterative improvement approaches. Algorithms belonging to the first class, as suggested by the name, typically divide the problem into two phases: one involves a customer allocation (clustering), while another phase deals with routes generation (routing), determining the order of the visit. As an illustration, Beasley (1983) propose a series of routing first and clustering second methods for the VRP. On the other hand, construction heuristic commence with an initial empty solution, gradually filled by the evaluation of feasible alternatives. This type of heuristic encompasses the savings algorithm, introduced by Clarke and Wright (1964), and insertion heuristics which create a solution by inserting customers who have not yet been assigned to a route, see Mole and S. Jameson (1976) and Laporte et al. (2002). Ultimately, iterative improvement heuristics are algorithms starting from a feasible solution and gradually improving it by applying techniques such as nearest neighbour, more general local search, hill climbing and many others, see Corona (2005) for a more detailed description.

Example of real-world problem addressed heuristically can be found in Hadjiconstantinou and Baldacci (1998), where practical issue concerning the provision of maintenance services are investigated. The authors aim at determining the service boundaries for each depot, the daily customer visits, and the routes followed by the service crews. They introduced a five-level heuristic for a multi-depot variant of the problem that involves solving the periodic VRP, i.e. the VRP for each day, a TSP for each route and optimising the routes. In a similar fashion, Salhi and Sari (1997) propose a multi-level composite heuristic that dramatically improves computational times for achieving solutions and, at the same time, maintains comparable quality with those found in the existing literature. The authors apply their heuristic also to the problem involving a heterogeneous fleet of vehicles. Nagy and Salhi (2005) expand the insertion-based heuristic proposed by Salhi and Nagy (1999), in order to handle scenarios involving pickup and delivery. Helsgaun (2000) propose a solution approach for the TSP based on a careful implementation of the Lin and Kernighan (1973) algorithm, an iterative improvement heuristics taking an Hamiltonian cycle as initial solution, attempting to improve it by neighbourhood search and repeating the process until a local minimum is found. One such implementation is now incorporated in the Concorde TSP Solver of Applegate et al. (2000). Chiu et al. (2006) propose a two-phase heuristic approach to examine the problem with time windows. The authors account for waiting time in the objective function. Waiting time result in having a substantial impact on the overall distribution time and the number of vehicles required, particularly in scenarios with tight time windows. The authors carry out also an empirical investigation by conducting a case study with a logistics company in Taiwan.

In the literature, we find that direct heuristic methods can sometimes be enhanced by general higher-level procedures to find, generate or select heuristics, especially in case of incomplete information or when limited computation capacity is available. These procedures are known as *metaheuristics*.

2.1.2.3 Metaheuristics

With the objective of strategically guide the search process and efficiently explore the search space, metaheuristics select a subset of solutions that would be otherwise impractical to exhaustively enumerate and handle the optimisation problem without relying heavily on specific problem assumptions. They are widely and extensively utilised to address several variants of the VRP, with the most recurring types being simulated annealing, tabu search, genetic algorithms, ant colony optimisation and iterative location search, among others.

Renaud et al. (1996) are pioneers in proposing metaheuristic for tackling the capacitated multi-depot VRP. Authors' approach involves the utilization of a tabu search algorithm, incorporating the improved petal heuristic for constructing the initial solution. Tabu search algorithms are also introduced by Cordeau et al. (1997), Tüzün and Burke (1999), Cordeau et al. (2001) and Parthanadee and Logendran (2006), the latter presenting three tabu search heuristics applied to a mixed-integer linear programming formulation of the VRP. Alongside tabu search metaheuristics, genetic algorithms designed for (multi-depot) capacitated VRP are proposed independently by Filipec et al. (1997), Salhi et al. (1998) and Skok et al. (2000). Thangiah and Salhi (2001) introduce an approach nominated ”genetic clustering”, where they deploy genetic algorithms to create clusters of clients, while route optimisation is achieved by solving a TSP utilizing an insertion heuristic. On the other hand, Yücenur and Demirel (2011) propose and implement genetic algorithms employing a geometric shape-based clustering algorithm, demonstrating improved clustering performance and computational efficiency compared to the nearest neighbor algorithm. Aras et al. (2011) provide a supplementary example of genetic algorithm, while Vidal et al. (2012) and Vidal et al. (2013) introduce an iterative metaheuristic based on genetic algorithms to address large scale multi-depot and periodic VRP. An evolutionary algorithm coupled with a local search heuristic is introduced by Vianna et al. (1999). Other examples of metaheuristics utilised in the literature are the simulated annealing algorithm proposed by Lim and Zhu (2006), while Polacek et al. (2005) and Polacek et al. (2008) introduce variable neighborhood search procedures. Another approach is suggested in a case study involving waste collection systems in 202 Portuguese localities and it is presented by Matos and Oliveira (2004), who propose an ant colony optimisation algorithm. Also, Li and Liu (2008) consider an ant colony optimisation algorithm for the multi-depot open vehicle routing problem with replenishment during the execution of routes.

Combined metaheuristic approaches are also present in the literature. Simulated annealing is combined with tabu search in Wu et al. (2012), where the authors define a novel approach to simultaneously solve the problems of setting the location of depots, routing vehicles, and assigning clients to them. A hybrid algorithm coupling multiple ant colony systems and simulated annealing is introduced by Ting and Chen (2009). Wang et al. (2011) incorporate simulated annealing and a modified variable neighborhood search in their algorithm, in addition to a clustering algorithm utilised to allocate customers and determine the starting solution. Liu and Yu (2013) solve a VRP with an heterogeneous fleet of vehicles by means of a genetic algorithm and ant colony optimisation procedure. Liu (2013) tackle the classical multi-depot VRP with a combination of genetic algorithm, bee colony optimisation and simulated annealing. Iterated local search together with dynamic programming and genetic algorithm is instead proposed by Vidal et al. (2014) to solve the multi-depot VRP with an unconstrained vehicle fleet.

Finally, we note that a combination of (meta)heuristic and exact methods is also present in the literature as well as in practice when solving combinatorial optimisation problems such as the VRP. Examples of hybrid methodology can be found in Özyurt and Aksen (2007), where the utilisation of a hybrid approach based on Lagrangian relaxation and tabu search improves the solutions found by Tüzün and Burke (1999), and in Flisberg et al. (2009), who integrate a mixed-integer linear program and a tabu search procedure to solve a case study of the VRP with an heterogeneous fleet of vehicles, time window constraints, pickups and split deliveries. In Escobar et al. (2013), the authors suggest a two-phase algorithm combining exact and heuristic methods in the first construction phase with a modified granular tabu search in the second improvement phase.

The examples mentioned above represent only a fraction of the diverse range of approaches and algorithms found in the literature for addressing different variations of the VRP. Nevertheless, it is worth noting that the effectiveness of metaheuristics might heavily rely on the quality of the initial solutions. Ho et al. (2008) deploy genetic algorithms to solve the VRP and compare solutions found by generating initial solution, first utilising the savings algorithm proposed by Clarke and Wright (1964), later improved via nearest neighbourhood search, and then randomly. The authors prove that a thoughtful construction of initial solutions can outperform the random generation in terms of improving the total delivery time.

Given that our focus is on the modelling side more than on the construction and implementation of new algorithms to solve the vehicle routing and trajectory optimisation problems, we leave the details of the above mentioned methods to the reader. Nonetheless, it is worth recalling here the strong interconnection between the m-TSP and the (C)VRP, and remember that the m-TSP can be reduced to the (C)VRP if all vehicles are considered to have infinite capacity. An extensive literature review on models and algorithms for the m-TSP is then presented by Bektas (2006). As a final remark, we would like to mention here that the VR problem and the m-TSP are problem widely studied for terrestrial applications, however, with the development of new technologies, new variants of these problems are gaining interest among the scientific community. The problem of routing unmanned aerial vehicle is more complex than the VRP, because it combines the VR combinatorial characteristics with the complexity of dealing with the dynamical systems of UAVs (i.e., flight dynamics, battery life, wind conditions), the latter being usually addressed separately within the so-called trajectory optimisation and optimal control problems.

2.2 The Trajectory Optimisation Problem

Trajectory optimisation (TO) is the process of designing a system's trajectory that minimises (or maximises) some measure of performance while satisfying a set of constraints, typically boundary conditions, dynamic constraints as well as path constraints and variable bounds. Trajectory optimisation can also be viewed as a special case of Optimal Control (OC), which aims at finding control functions or control feedback (inputs) that minimise an objective function or performance index, respecting differential equation constraints. OC establishes its roots in the seventeenth century when J. Bernoulli, together with other famous contemporary mathematicians, argued on the question of *which is the fastest path between two fixed points that a particle under the influence of gravity can follow*, see Bryson (1996) and Sussmann and Willems (1997). Further development in the 18^{th} century are due to Euler and Lagrange. In theoretical physics, astonishing results in connection with Hamilton's principle or the Principle of Least Action are also entangled with OC. In the late 1920s, applications of OC in economics start to emerge, but the strongest impulse come from military applications during and after the second world war. It is then with Pontryagin that we have a decisive step forward, when he and his team formulate and prove the so-called *Pontryagin Maximum Principle*, see Pontryagin et al. (1964). The principle provides necessary conditions for optimal control by deriving and solving the *Hamiltonian system* that will be introduced in the following sections. In the late 1950s, Bellman offers a new point of view on OC with the introduction of what he called *Dynamic Programming* which allows for reduction in computational times needed for finding optimal controls, see Kirk (1998). A further crucial breakthrough is reached with the advent of the computers, enabling a massive expansion of OC theory and applications to real, and therefore more complex, problems. Disrupting innovations can be seen in fields ranging from biology, see Joshi (2002), to economics, see Sun and Li (2008), to robotics and aerospace, see Goh (2008), to mention a few. Nowadays, there exists a broad OC theory, including many different solution approaches. In what follows, first we highlight the common aspects of these approaches, also proposing a general mathematical formulation of the OC problem, and second, we describe typical solution techniques.

2.2.1 General Formulation of Optimal Control Problems

As mentioned above, optimal control aims at determining the inputs to a dynamical system such that a specified performance index is minimised (or maximised), while constraints on the motion of the system are satisfied. Therefore, a typical OC problem requires the following:

- a time set $\mathscr{T} = \{\tau \mid \tau \in [\tau_0, \tau_f] \subseteq \mathbb{R}\}$;
- a set of state variables $\mathbf{y}(\tau) \in \mathscr{Y} \subseteq \mathbb{R}^n$;
- a set of control variables (inputs) $\mathbf{u}(\tau) \in \mathscr{U} \subseteq \mathbb{R}^u$;
- a vector of parameters $\mathbf{p} \in \mathbb{R}^q$;
- a performance index or objective functional, $J = J(\mathbf{y}(\tau), \mathbf{u}(\tau), \mathbf{p})$.

The main goal consists in finding control functions $u(\tau) \in \mathscr{U}$, $\tau_0 \leq \tau \leq \tau_f$, and associated states $y(\tau) \in \mathscr{Y}$, such that the objective function J is minimised (maximised). More precisely we can write a generic OC problem as the problem of finding the

$$\min_{\mathbf{y}(\tau),\mathbf{u}(\tau)} J(\mathbf{y}(\tau), \mathbf{u}(\tau), \tau, \mathbf{p}) = \Phi(\mathbf{y}(\tau_0), \tau_0, \mathbf{y}(\tau_f), \tau_f, \mathbf{p}) + \int_{\tau_0}^{\tau_f} w(\mathbf{y}(\tau), \mathbf{u}(\tau), \tau, \mathbf{p})\, d\tau, \tag{2.21}$$

subject to

$$\dot{\mathbf{y}} = \mathbf{f}(\mathbf{y}(\tau), \mathbf{u}(\tau), \tau, \mathbf{p}) \tag{2.22}$$

$$\mathbf{h}(\mathbf{y}(\tau), \mathbf{u}(\tau), \tau, \mathbf{p}) \leq 0 \tag{2.23}$$

$$\phi(\mathbf{y}(\tau_0), \tau_0, \mathbf{y}(\tau_f), \tau_f, \mathbf{p}) \leq 0, \tag{2.24}$$

where the performance index J embodies a boundary penalisation function Φ, discouraging non desirable features at the end points, and a path cost function , describing the price along the chosen trajectory. On the other hand, *dynamic constraints* (2.22) limit the manoeuvrability of the vehicle, while *path constraints* (2.23) and *boundary conditions* (2.24) are also respected.

Typically, OC problems are formulated as a collection of P different phases, indexed by $k \in \{1, \dots, P\}$, and these phases are then meaningfully aggregate or joined. In general, the independent variable τ is therefore defined in consecutive intervals $\tau_0^{(k)} \leq \tau \leq \tau_f^{(k)}$. In light of this, we can think of multiple-phase OC problems as follows. Optimise the total performance index

$$J = \sum_{k=1}^{P} J^{(k)},$$

where the performance index of phase k, $J^{(k)}$, can be written as in (2.21), subject to *dynamic constraints*

$$\dot{\mathbf{y}}^{(k)} = \mathbf{f}(\mathbf{y}^{(k)}(\tau), \mathbf{u}^{(k)}(\tau), \tau, \mathbf{p}^{(k)}),$$

path constraints

$$\mathbf{h}^{(k)}(\mathbf{y}^{(k)}(\tau), \mathbf{u}^{(k)}(\tau), \tau, \mathbf{p}^{(k)}) \leq 0,$$

boundary conditions

$$\phi^{(k)}(\mathbf{y}^{(k)}(\tau_0^{(k)}), \tau_0^{(k)}, \mathbf{y}^{(k)}(\tau_f^{(k)}), \tau_f^{(k)}, \mathbf{p}^{(k)}) \leq 0,$$

as well as *linkage constraints*

$$\mathbf{\Psi}(\mathbf{y}^{(k)}(\tau_f^{(k)}), \mathbf{u}^{(k)}(\tau_f^{(k)}), \tau_f^{(k)}, \mathbf{p}^{(k)}, \mathbf{y}^{(k+1)}(\tau_0^{(k+1)}), \mathbf{u}^{(k+1)}(\tau_0^{(k+1)}), \tau_0^{(k+1)}, \mathbf{p}^{(k+1)}) \leq 0$$

guaranteeing some regularity of the solution along the different phases. Given this type of formulations, we can proceed and explore typical approaches to solve optimal control problems.

2.2.2 Solution Approaches

Due to their complexity, usually optimal control problems cannot be solved analytically and need to be solved numerically. Numerical methods for solving OC problems are divided into two major classes:

- indirect methods and
- direct methods

In an *indirect method*, first (or higher) order optimality conditions are derived for the problem at hands through the calculus of variations. Consequently, the stemming (nonlinear) system of equations is then solved, leading to (possibly non unique) candidate optimal points. Each points is then evaluated to verify if it is indeed a local minimum. Finally, of the locally optimising solutions, a point associated with the best performance index is chosen. On the other hand, in a *direct method*, state and control variables are somehow discretised and the problem is then converted into a (nonlinear) optimisation or (nonlinear) programming problem (NLP). The stemming NLP is then solved leveraging on well known optimisation techniques.

In light of the this, it is also possible to state that well-founded methods for solving optimal control problems rely on three fundamental components:

1. methods for solving the differential equations and integrating functions;
2. methods for solving a system of nonlinear algebraic equations;
3. methods for solving a nonlinear optimisation problem.

with all numerical methods in optimal control requiring knowledge and usage of methods for solving differential equations and integrating functions. Then, while indirect methods generally combine the numerical solution of differential equations with the numerical solution of systems of nonlinear equations, direct methods instead integrate numerical solution of differential equations with nonlinear optimisation.

With this said, numerous advanced algorithms have been developed to address TO problems. Providing a comprehensive review of these works is beyond the scope of this thesis, nonetheless we outline below the general structure of these methodologies. For a more in-depth understanding, the interested reader can always refer to papers by Stryk and Bulirsch (1992), Betts (1998), Rao (2010) and Rao (2014), or consult books by Kirk (1998) and Betts (2009), which offer valuable insights on the subject.

2.2.2.1 Indirect Methods

In indirect methods, first-order optimality conditions of the optimal control problem are typically derived using the augmented *Hamiltonian*, H, defined as

$$H(\mathbf{y}, \mathbf{u}, \lambda, \mu, \tau) = w + \lambda^T \mathbf{f} + \mu^T \mathbf{h},$$

where $\lambda(\tau) \in \mathbb{R}^n$, $\mu(\tau) \in \mathbb{R}^c$, with $c = dim(\mathbf{h})$ being the number of path constraints. The first-order optimality conditions of the continuous-time problem are then given by:

$$\begin{aligned}
\dot{\mathbf{y}} &= \left(\frac{\partial H}{\partial \lambda}\right)^T \\
\dot{\lambda} &= -\left(\frac{\partial H}{\partial \mathbf{y}}\right)^T \\
\mathbf{u}^* &= \arg\min_{\mathbf{u} \in \mathscr{U}} H \\
0 &= \phi(\mathbf{y}(\tau_0), \tau_0, \mathbf{y}(\tau_f), \tau_f, \mathbf{p}) \\
\lambda(\tau_0) &= -\frac{\partial \Phi}{\partial \mathbf{y}}\bigg|_{\tau_0} + \nu^T \frac{\partial \phi}{\partial \mathbf{y}}\bigg|_{\tau_0} \\
\lambda(\tau_f) &= \frac{\partial \Phi}{\partial \mathbf{y}}\bigg|_{\tau_f} - \nu^T \frac{\partial \phi}{\partial \mathbf{y}}\bigg|_{\tau_f} \\
H(\tau_0) &= \frac{\partial \Phi}{\partial \tau_0} - \nu^T \frac{\partial \phi}{\partial \tau_0} \\
H(\tau_f) &= \frac{\partial \Phi}{\partial \tau_f} - \nu^T \frac{\partial \phi}{\partial \tau_f} \\
\mu_j(\tau) &= 0 \text{ if } h_j(\mathbf{y}, \mathbf{u}, \tau, \mathbf{p}) < 0 \\
\mu_j(\tau) &\leq 0 \text{ if } h_j(\mathbf{y}, \mathbf{u}, \tau, \mathbf{p}) = 0,
\end{aligned}$$

where $\nu \in \mathbb{R}^b$. We can notice here that μ and ν can be seen as the Lagrange multipliers associated with the path constraints and the boundary conditions respectively. The above reported system of equation is also called *Hamiltonian boundary value problem* (HBVP) and it is typically solved numerically, using shooting methods or collocation methods, see Betts (2009) for reference.

Several difficulties arise when OC problems are solved by indirect methods. Firstly, it is necessary to calculate the Hamiltonian and the HBVP associated with the problem, resulting in limited flexibility, since every time a new problem is formulated, a new derivation is required. In addition, values of λ, μ and ν must be guessed and, due to their non-intuitive nature, this might lead to ill-conditioning issues. Conversely, direct methods do not rely on explicit derivation of necessary conditions and therefore, they gained popularity in solving OC problems over the past decades, especially with calculation capability advancements of modern computers.

2.2.2.2 Direct Methods

Direct methods start by transcribing the continuous OC problem into a (nonlinear) programming problem. Once in this form, it is possible to find solutions by constructing a sequence of points $(\mathbf{y}_1, \mathbf{u}_1)$, $(\mathbf{y}_2, \mathbf{u}_2)$, . . . , $(\mathbf{y}^*, \mathbf{u}^*)$, such that the objective function J to be minimised, or an approximation thereof, satisfies the following chain of inequalities: $J(\mathbf{y}_1, \mathbf{u}_1) \geq J(\mathbf{y}_2, \mathbf{u}_2) \geq \cdots \geq J(\mathbf{y}^*, \mathbf{u}^*)$. Alternatively, the problem can be passed to commercial solvers like SNOPT (Gill et al.), IPOPT, FMINCON, or CPLEX (IBM-ILOG-CPLEX), Gurobi (Gurobi), Xpress if the resulting problem has a limited amount of nonlinearities, that will construct the above sequence for us. Transcription algorithms are manifold and generally tailored to the specific problem, nonetheless it is possible to embed them into two broader classes: (i) shooting methods and (i) simultaneous or collocation methods. The difference originate from the way of dealing with system's dynamic and boundary constraints. In shooting methods, the strategy can be described as

1. guess initial conditions and control variables;
2. "shoot" propagating the differential equations from τ_0 to τ_f;
3. evaluate the error in the final boundary conditions $\phi(\mathbf{y}(\tau_f), \tau_f, \mathbf{p})$;
4. derive and solve a (nonlinear) programming problem to adjust the initial guess and satisfy the final boundary conditions;
5. iterate until a stopping criteria is reached.

From a practical standpoint, the shooting method is widely used primarily because the transcribed problem has a small number of variables. Unfortunately, shooting methods also suffers from being extremely sensitive to the initial guess. This can be limited deploying multiple shooting methods, paying the price in "shooting" computational time, especially when the evaluation of the the error in the final boundary conditions depends on a highly nonlinear function ϕ. On the other hand, collocation methods enforce the dynamics at a series of points along the trajectory, so avoiding the just mentioned issues. The idea here is to choose a finite-dimensional space for the candidate solutions, a number of points in the domain (called collocation points) and then to find a solution satisfying dynamic's equation and constraints at the collocation points. Certainly, hybrid solutions like *multiple shooting* can also be adopted, where the idea is to choose a set of collocation points and deploy a single shooting method between them, closing the gap between the arrival state at a collocation point and the state that would have been reached if a "shoot" is conducted from the previous step.

Despite the great variety of direct methods in the literature, all of them rely on discretisation of the solution space and integration schemes to describe/ solve dynamic constraints. The general approach is to partition each phase duration into smaller intervals

$$\tau_0 < \tau_1 < \cdots < \tau_N = \tau_f,$$

where for simplicity of notation, we drop the superscript highlighting dependency on phase (k). In addition, we will write $\mathbf{y}_t$ in place of $\mathbf{y}(\tau_t)$, $\mathbf{f}_t$ for $\mathbf{f}(\mathbf{y}(\tau_t), \mathbf{u}(\tau_t), \tau_t, \mathbf{p})$ and similarly for the other functions, variables or parameters when necessary, with t representing the grid point, or time step, of time partition. Defining then variables at the grid points

$$\mathbf{x}^T = (\mathbf{y}_1, \mathbf{u}_1, \ldots, \mathbf{y}_N, \mathbf{u}_N)$$

and *defects* ζ_t describing the dynamic equations, the problem can be transcribed to a (nonlinear) programming problem where the OC constraints are replace by

$$c_L \leq \mathbf{c}(\mathbf{x}) \leq c_U,$$

with

$$\begin{aligned}
&\mathbf{c}(\mathbf{x}) = (\zeta_1, \ldots, \zeta_{N-1}, \phi_0, \phi_f, \mathbf{h}_1, \ldots, \mathbf{h}_N),\\
&c_L = (0, \ldots, 0, -\infty, \ldots, -\infty) \text{ and}\\
&c_U = (0, \ldots, 0).
\end{aligned}$$

We can notice here that the derived boundary conditions and path constraints are imposed at every time step, while dynamic constraints can be enforced at grid points as well as at mid points of the partition, depending on the chosen integration scheme. Below, we report the most commonly used definition of ζ_t, according to the corresponding integration scheme.

Euler

$$\zeta_t = \mathbf{y}_{t+1} - \mathbf{y}_t - (\tau_{t+1} - \tau_t)\mathbf{f}_t$$

Runge-Kutta

$$\zeta_\mathbf{t} = \mathbf{y}_{t+1} - \mathbf{y}_t - \frac{\tau_{t+1} - \tau_t}{6}(\mathbf{k}_1 + \mathbf{k}_2 + \mathbf{k}_3 + \mathbf{k}_4),$$

with

$$\begin{aligned}
\mathbf{k}_1 &= \mathbf{f}_t\\
\mathbf{k}_2 &= \mathbf{f}\left(\mathbf{y}_t + \frac{\mathbf{k}_1}{2}, \mathbf{u}\left(\frac{\tau_{t+1} + \tau_t}{2}\right), \frac{\tau_{t+1} + \tau_t}{2}, \mathbf{p}\right)\\
\mathbf{k}_3 &= \mathbf{f}\left(\mathbf{y}_t + \frac{\mathbf{k}_2}{2}, \mathbf{u}\left(\frac{\tau_{t+1} + \tau_t}{2}\right), \frac{\tau_{t+1} + \tau_t}{2}, \mathbf{p}\right)\\
\mathbf{k}_4 &= \mathbf{f}\left(\mathbf{y}_t + \mathbf{k}_3, \mathbf{u}_{t+1}, \tau_{t+1}, \mathbf{p}\right)
\end{aligned}$$

Trapezoidal

$$\zeta_{\mathbf{t}} = \mathbf{y}_{t+1} - \mathbf{y}_t - \frac{\tau_{t+1} - \tau_t}{2}(\mathbf{f}_t + \mathbf{f}_{t+1})$$

Hermite–Simpson

$$\zeta_{\mathbf{t}} = \mathbf{y}_{t+1} - \mathbf{y}_t - \frac{\tau_{t+1} - \tau_t}{6}(\mathbf{f}_t + 4\bar{\mathbf{f}}_{t+1} + \mathbf{f}_{t+1}),$$

with

$$\begin{aligned}\bar{\mathbf{y}}_{t+1} &= \frac{\mathbf{y}_{t+1} + \mathbf{y}_t}{2} + \frac{\tau_{t+1} - \tau_t}{8}(\mathbf{f}_t - \mathbf{f}_{t+1})\\ \bar{\mathbf{f}}_{t+1} &= \mathbf{f}\left(\bar{\mathbf{y}}_{t+1}, \mathbf{u}\left(\frac{\tau_{t+1} + \tau_t}{2}\right), \frac{\tau_{t+1} + \tau_t}{2}, \mathbf{p}\right).\end{aligned}$$

The design of an appropriate performance index/ objective function J is then necessary to solve the OC problem and it is usually tailored to the specific problem at hand. It might be nonetheless useful to write a generic problem formulation in a compact form as

$$\begin{gathered}\min\ J(\mathbf{x})\\ \text{subject to } c_L \leq \mathbf{c}(\mathbf{x}) \leq c_U,\end{gathered}$$

with

$$\begin{aligned}&\mathbf{c}(\mathbf{x}) = (\zeta_1, \ldots, \zeta_{N-1}, \phi_0, \phi_f, \mathbf{h}_1, \ldots, \mathbf{h}_N),\\ &c_L = (0, \ldots, 0, -\infty, \ldots, -\infty) \text{ and}\\ &c_U = (0, \ldots, 0).\end{aligned}$$

A deeper study of the direct collocation methods is not the main aim of this research, therefore we refer the interested reader to Betts (2009) and Böhme and Frank (2017) for a more detailed study. Then, having discussed the basic principle of both the vehicle routing and trajectory optimisation problem, we can now proceed to define the generic Vehicle Routing and Trajectory Optimisation Problem (VRTOP), integrating aspects stemming from both of the problems described above.

2.3 The VRTOP

In this section, we aim at giving a generic formulation of the Vehicle Routing and Trajectory Optimisation Problem (VRTOP) by presenting a problem where a fleet of vehicles is required to survey a set of point of interest, known as waypoints, also respecting kinematics and dynamics constraints. Consequently, recent developments in similar problem and solution approaches are discussed, highlighting the additional features of the work presented in the following chapters.

2.3.1 A General Mathematical Formulation

Let Υ be a set representing a fleet vehicles available at a starting point or launching site d. Let also $G = (V, A)$ be a graph, whose set of vertices V stand for the waypoints and A is the set of arcs between them. Moreover, let d' be the arrival or landing site. If we decide not to fix the number of vehicle used, similarly to before, it is convenient to introduce a cost F_g for using vehicle $g \in \Upsilon$. At this point, it is worth noting that if $d = d'$ and we neglect the dynamic constraints, the problem can be described as a VRP where the number of used vehicles is not fixed. In fact, we can utilise a notation similar to the corresponding model presented in section 2.1.1 and, while defining the variables

$$x_{gij} = \begin{cases} 1, & \text{if vehicle } g \text{ travels on arc } (i, j) \\ 0, & \text{otherwise} \end{cases}$$

$$\mathcal{Y}_g = \begin{cases} 1, & \text{if vehicle } g \text{ is deployed} \\ 0, & \text{otherwise} \end{cases}$$

we can also denote the position of node i in a tour with O_i and use the value q representing the maximum number of waypoints that can be visited by any vehicle. In all the examples that we consider, q will be equal to the total number of waypoints. We can then introduce TO features defining

- state variables $\mathbf{y}_g(\tau) \in \mathscr{Y}_g \subseteq \mathbb{R}^{n_y}$, $n_y \in \mathbb{N}$, describing the complete status (position, orientation and velocity) of vehicle g travelling at time $\tau \in \mathbb{R}_0^+$; and
- control variables $\mathbf{u}_g(\tau) \in \mathscr{U}_g \subseteq \mathbb{R}^{n_u}$, $n_u \in \mathbb{N}$, modelling, for each vehicle g, the inputs to the system.

In the following chapters, we will see that typical inputs for controlling UAVs can be the *bank angle* (needed to change the horizontal flight direction), the *angle-of-attack* (AoA) (needed to change the ascending or descending rate), which is directly related with the *coefficient of lift* of an aircraft, as well as the *thrust* when the UAV have an engine. Depending on the type of vehicle used, inputs can also vary in accordance with the source generating changes in the external force (for example, a quadcopter will not require the angle of attack, but on the other hand the thrust can be generated in different directions). In addition to the above, in order to describe the dynamic equations, we need to define parameters $\mathbf{p}_g \in \mathbb{R}^{n_p}$, $n_p \in \mathbb{N}$, relative to vehicle g. Typical examples from the UAV world are the mass, wing area and aerodynamics coefficients.

Having this in mind, we can set the objective of minimising the total costs of travel. Travel costs typically include time requirements, vehicle performance index, monetary costs and many others, depending on the problem at end. In particular, we can consider VR objectives

$$\sum_{g \in \Upsilon} \left(F_g \mathcal{Y}_g + \sum_{(i,j) \in A} c_{ij} x_{gij} \right),$$

as well as TO ones

$$\sum_{g \in \Upsilon} J_g(\mathbf{y}_g(\tau), \mathbf{u}_g(\tau)) \mathcal{Y}_g.$$

The objective function will then try to minimise all of them at once, summing together the fixed costs of using a vehicle, additional costs of going from waypoint i to waypoint j, as well as vehicle-related performance costs described through indices J_g. As previously mentioned, each performance index J_g can be written as the sum of a boundary penalisation index

$$\Phi_g(\mathbf{y}_g(\tau_0), \tau_0, \mathbf{y_g}(\tau_f), \tau_f, \mathbf{p}_g),$$

discouraging non desirable features at the end points of the UAVs' trajectories, and a routing cost index

$$\int_{\tau_0}^{\tau_f} w_g(\mathbf{y}_g(\tau), \mathbf{u}_g(\tau), \tau, \mathbf{p}_g)\ dt.$$

Undesirable features may include sharp flight angles, prohibited flight speeds and noise levels, see Vanderbei (2001) and Zhang et al. (2012) among others. It is worth noting here that the exploration of the full Pareto front for such a multi-objective problem might be of high interest, possibly giving different weights to the objectives. However, for simplicity of exposition and for a first comprehension of the problem, we decided to just sum the different components with equal weight.

Having defined the optimisation goals, we want also to impose requirements in such a way that

- all vehicle routes start at the launching site $d \in V$;
- selected vehicle routes terminate at landing site $d' \in V$;
- each location in $V \setminus \{d, d'\}$ is visited exactly once by exactly one vehicle.

In a similar fashion of section 2.1.1, we can write these constraints as

$$\sum_{(i,j)\in A} x_{gij} \leq M\mathcal{Y}_g \tag{2.25}$$

$$\sum_{g\in\Upsilon} \mathcal{Y}_g \geq m_L \tag{2.26}$$

$$\sum_{i\in V:(i,j)\in A} \sum_{g\in\Upsilon} x_{gij} = 1,\ \forall j \in V \setminus \{d\} \tag{2.27}$$

$$\sum_{j\in V:(i,j)\in A} \sum_{g\in\Upsilon} x_{gij} = 1,\ \forall i \in V \setminus \{d\} \tag{2.28}$$

$$\sum_{j\in V:(d,j)\in A} x_{gdj} = \mathcal{Y}_g,\ \forall g \in \Upsilon \tag{2.29}$$

$$\sum_{i\in V:(i,d')\in A} x_{gid'} = \mathcal{Y}_g,\ \forall g \in \Upsilon \tag{2.30}$$

$$\sum_{i\in V:(i,j)\in A} x_{gij} = \sum_{i\in V:(j,i)\in A} x_{gji},\ \forall j \in V, \forall g \in \Upsilon \tag{2.31}$$

$$O_i - O_j + q\sum_{g\in\Upsilon} x_{gij} \leq q - 1,\ \forall (i,j) \in A, i, j \neq d, d'. \tag{2.32}$$

Consequently, we want to impose trajectory constraints. The physical system governing the flying vehicle g is generally described by a system of Ordinary Differential Equations (ODE), called Equations of Motion (EoMs). As previously introduced, we can then enforce *dynamic constraints* requiring that

$$\dot{\mathbf{y}}_g = \mathbf{f}_g(\mathbf{y}_g(\tau), \mathbf{y}_g(\tau), \tau, \mathbf{p}_g)$$

for all $g \in \Upsilon$, where functions $\mathbf{f}_g$ represent then the relation of variables and parameters with the changes in state of the UAV over time. In addition, we consider the *path constraints*

$$\mathbf{h}_g(\mathbf{y}_g(\tau), \mathbf{u}_g(\tau), \tau, \mathbf{p}_g) \leq 0.$$

These constraints usually model obstacles in the space, collisions avoidance between UAVs, non desirable flying dynamics and many others, including variable bounds. Finally, we want to consider *boundary conditions*

$$\phi_g(\mathbf{y}_g(\tau_0), \tau_0, \mathbf{y}_g(\tau_f), \tau_f, \mathbf{p}_g) \leq 0.$$

We can finally write the problem in a complete and compact form as

$$
\begin{aligned}
\min \quad & \sum_{g\in\Upsilon}\left(F_g\mathcal{Y}_g + \sum_{(i,j)\in A} c_{ij}x_{gij} + J_g(\mathbf{y}_g(\tau),\mathbf{u}_g(\tau))\ \mathcal{Y}_g\right) && (2.33)\\
\text{subject to} \quad & \sum_{(i,j)\in A} x_{gij} \le M\mathcal{Y}_g && (2.34)\\
& \sum_{g\in\Upsilon}\mathcal{Y}_g \ge m_L && (2.35)\\
& \sum_{i\in V:(i,j)\in A}\sum_{g\in\Upsilon} x_{gij} = 1,\ \forall j\in V\setminus\{d\} && (2.36)\\
& \sum_{j\in V:(i,j)\in A}\sum_{g\in\Upsilon} x_{gij} = 1,\ \forall i\in V\setminus\{d\} && (2.37)\\
& \sum_{j\in V:(d,j)\in A} x_{gdj} = \mathcal{Y}_g,\ \forall g\in\Upsilon && (2.38)\\
& \sum_{i\in V:(i,d')\in A} x_{gid'} = \mathcal{Y}_g,\ \forall g\in\Upsilon && (2.39)\\
& \sum_{i\in V:(i,j)\in A} x_{gij} = \sum_{i\in V:(j,i)\in A} x_{gji},\ \forall j\in V, \forall g\in\Upsilon && (2.40)\\
& O_i - O_j + q\sum_{g\in\Upsilon} x_{gij} \le q-1,\ \forall (i,j)\in A, i,j\ne d,d' && (2.41)\\
& \dot{\mathbf{y}}_g = \mathbf{f}_g(\mathbf{y}_g(\tau),\mathbf{y}_g(\tau),\tau,\mathbf{p}_g)\ \mathcal{Y}_g,\ \forall g\in\Upsilon,\ \forall\tau && (2.42)\\
& \mathbf{h}_g(\mathbf{y}_g(\tau),\mathbf{u}_g(\tau),\tau,\mathbf{p}_g)\ \mathcal{Y}_g \le 0,\ \forall g\in\Upsilon,\ \forall\tau && (2.43)\\
& \phi_g(\mathbf{y}_g(\tau_0),\tau_0,\mathbf{y}_g(\tau_f),\tau_f,\mathbf{p}_g)\ \mathcal{Y}_g \le 0,\ \forall g\in\Upsilon,\ \forall\tau, && (2.44)
\end{aligned}
$$

with each variable and parameter living in the appropriate domain. Sometimes it is also convenient to formulate the problem as a collection of phases, each one representing the flight of a UAV between two different point of interest, and all linked to guarantee continuity. Defining variables

- $\tau^0_{gij}, \tau^f_{gij} \in \mathbb{R}^+_0$ the times when vehicle g can respectively start and end phase ij;
- $\mathbf{y}_{gij}(\tau) \in \mathscr{Y}_g \subseteq \mathbb{R}^{n_y}$, $n_y \in \mathbb{N}$, describing the complete status (position, orientation and velocity) of vehicle g travelling from waypoint i to waypoint j at time $\tau \in [\tau^0_{gij}, \tau^f_{gij}]$;
- $\mathbf{u}_{gij}(\tau) \in \mathscr{U}_g \subseteq \mathbb{R}^{n_u}$, $n_u \in \mathbb{N}$, modelling the inputs to the system between waypoint i and j at time $\tau \in [\tau^0_{gij}, \tau^f_{gij}]$;

as well as the parameters, $\mathbf{p}_{gij}$, and functions according to the corresponding phase, we can modify the previous formulation as follows

$$\min \sum_{g\in\Upsilon}\left(F_g\mathcal{Y}_g + \sum_{(i,j)\in A}\left(c_{ij}x_{gij} + J_{gij}(\mathbf{y}_{gij}(\tau),\mathbf{u}_{gij}(\tau))\ \mathcal{Y}_{gij}\right)\right) \tag{2.45}$$

$$\text{subject to} \sum_{(i,j)\in A} x_{gij} \le M\mathcal{Y}_g \tag{2.46}$$

$$\sum_{g\in\Upsilon}\mathcal{Y}_g \ge m_L \tag{2.47}$$

$$\sum_{i\in V:(i,j)\in A}\ \sum_{g\in\Upsilon} x_{gij} = 1,\ \forall j\in V\setminus\{d\} \tag{2.48}$$

$$\sum_{j\in V:(i,j)\in A}\ \sum_{g\in\Upsilon} x_{gij} = 1,\ \forall i\in V\setminus\{d\} \tag{2.49}$$

$$\sum_{j\in V:(d,j)\in A} x_{gdj} = \mathcal{Y}_g,\ \forall g\in\Upsilon \tag{2.50}$$

$$\sum_{i\in V:(i,d')\in A} x_{gid'} = \mathcal{Y}_g,\ \forall g\in\Upsilon \tag{2.51}$$

$$\sum_{i\in V:(i,j)\in A} x_{gij} = \sum_{i\in V:(j,i)\in A} x_{gji},\ \forall j\in V, \forall g\in\Upsilon \tag{2.52}$$

$$O_i - O_j + q\sum_{g\in\Upsilon} x_{gij} \le q-1,\ \forall (i,j)\in A, i,j\neq d,d' \tag{2.53}$$

$$\dot{\mathbf{y}}_{gij} = \mathbf{f}_g(\mathbf{y}_{gij}(\tau),\mathbf{y}_{gij}(\tau),\tau,\mathbf{p}_{gij})\ \mathcal{Y}_g,\ \forall g\in\Upsilon,\ \forall\tau\in[\tau^0_{gij},\tau^f_{gij}] \tag{2.54}$$

$$\mathbf{h}_g(\mathbf{y}_{gij}(\tau),\mathbf{u}_{gij}(\tau),\tau,\mathbf{p}_{gij})\ \mathcal{Y}_g \le 0,\ \forall g\in\Upsilon,\ \forall\tau\in[\tau^0_{gij},\tau^f_{gij}] \tag{2.55}$$

$$\phi_{gij}(\mathbf{y}_{gij}(\tau_0),\tau_0,\mathbf{y}_{gij}(\tau_f),\tau_f,\mathbf{p}_{gij})\ \mathcal{Y}_g \le 0,\ \forall g\in\Upsilon,\ \forall\tau\in[\tau^0_{gij},\tau^f_{gij}] \tag{2.56}$$

$$(\tau^0_{gjl} - \tau^f_{gij})x_{gij}x_{gjl} = 0,\ \forall i,j,l\in V,\ \forall g\in\Upsilon \tag{2.57}$$

$$\left(\mathbf{y}_g(\tau^0_{gjl}) - \mathbf{y}_g(\tau^f_{gij})\right)x_{gij}x_{gjl} = 0,\ \forall i,j,l\in V,\ \forall g\in\Upsilon \tag{2.58}$$

$$\left(\mathbf{u}_g(\tau^0_{gjl}) - \mathbf{u}_g(\tau^f_{gij})\right)x_{gij}x_{gjl} = 0,\ \forall i,j,l\in V,\ \forall g\in\Upsilon, \tag{2.59}$$

where objective and constraints (2.45)-(2.56) are as in the previously defined models, while constraints (2.57)-(2.59) ensure continuity of time, states and controls among the different phases. In light of this, having understood the VRTOP and its possible mathematical formulations, we present here some problems existent in the literature and focus on UAVs deployment, along with the methodology used to solve them. We also highlight here from the very beginning that the above model is just an illustration of how a generic nonlinear VRTOP model can be described. Our models and approaches presented along the manuscript differ from this and the other model presented in the literature for different aspects, of which the main is the way used to handle dynamic constraints (2.54)-(2.56). While a nonlinear description of the vehicles' dynamic might be more precise, it might lead to computational complexities and undesirable solving times. Therefore, accurate approximations of the dynamic constraints are derived, while avoiding oversimplification, allowing us to achieve reliable solutions within a reasonable time frame.

2.3.2 Research Developments

Interest in the autonomous use of UAVs, and consequently in the vehicle routing and trajectory optimisation problem, is exceptionally increased in the recent literature, where different and creative approaches and optimisation techniques have emerged to solve the VRTOP. However, due to the complications arising from embedding flight dynamic conditions into the planning process, the calculation of the related paths and trajectories is extremely simplified or neglected. The complex nature of the flight dynamic, see Stengel (2004), leads to nonconvex models and hence several optimisation techniques based on Optimal Control (OC) and Nonlinear Programming (NLP) have been developed to incorporate trajectory optimisation aspects in the problem formulation. As stated in Betts (2009), Rao (2010) and Rodrigues et al. (2014), NLP-based solutions are known to be sensitive to initial guesses, i.e. convergence can be only ensured if a proper initialisation is provided Zhao (2004), and providing those for flying vehicles requires extremely deep knowledge of flight dynamics. Sometimes, nonlinear dynamics of UAVs can be linearised and a Mixed-Integer Programming (MIP) formulation can be derived, overcoming such difficulties, see Stengel (2004), Hajiyev et al. (2015), How et al. (2015) and Harris and Acikmese (2013). Often however, some prior knowledge is required to perform such linearisation. Alternatively, Dubins curves, see Boissonnat et al. (1992), and Bézier curves, see Bertka (2008), are also utilized in the literature to represent the UAV dynamics. For example, Zhen et al. (2018) assign missions to a set of homogeneous UAVs using ant colony algorithms and generate UAVs trajectories through Dubins curves. In the authors approach, technical parameters of the UAVs, collision avoidance constraints and restricted airspaces influence the resulting paths.

Deploying such a plethora of techniques, quite a few VRTOP problems have been tackled in the literature. Keviczky et al. (2008) propose a decentralized approach to controlling the formation of autonomous vehicles based on receding horizon control. The approach relies on a combination of local sensor measurements and inter-vehicle communication to achieve a desired formation. On each vehicle, information about neighbors is used to predict their behavior and plan conflict-free trajectories that maintain coordination and achieve team objectives. Soler et al. (2014) present an approach for designing flight trajectories that minimise contrail formation. The approach is based on a mixed-integer nonlinear programming (MINLP) framework, which includes multiple phases to capture the different stages of a flight. The paper includes a detailed model of contrail formation and a comprehensive optimisation formulation that takes into account various constraints, such as air traffic control requirements, aircraft performance limits, and weather conditions. Fügenschuh and Müllenstedt (2015) use a mixed-integer linear programming (MILP) formulation to optimise the flight plans of UAVs. The MILP formulation includes variables to describe UAV's dynamic, such as their position, speed, and altitude at each time step. Constraints accounts for various factors affecting UAV's performance, including safety and airspace restrictions. Maolaaisha (2015) develops an optimisation framework for determining optimal aircraft trajectories from takeoff to landing in a free-flight environment. The approach uses mixed-integer programming (MIP) to consider discrete decisions, such as throttle setting and flap deployment, as well as continuous variables, such as altitude and airspeed, in the trajectory optimisation process. In Coutinho et al. (2019) a Mixed-Integer Programming (MIP) formulation is then proposed to solve a special case of the vehicle routing and trajectory optimisation problem where a fleet of homogeneous gliders is deployed, also known as GRTOP. The new formulation simultaneously optimises routes and flight trajectories along these routes. The flight dynamics of the fleet of gliders are modelled with constraints derived by time discretisation and integration schemes, while distance restrictions are modelled as second-order cone constraints. Highly nonlinear, nonconvex dynamic constraints are avoided by linearising the UAVs Equations of Motions (EOMs) around equilibrium flight conditions or previous trajectory guess, leading to a Mixed-Integer Second-Order Cone Programming (MISOCP) problem. To allow for a more tractable formulation while keeping high quality solutions, a relaxation of the resulting linear dynamical constraints and a corresponding addition of a penalisation term to the objective function has been made. Nonetheless, a single equilibrium condition is analysed, leading to a limited view on gliders dynamic considerations, especially when other steady-flight conditions have to be considered.

Ramirez-Atencia et al. (2017) deploy a genetic algorithm to assign tasks to a fleet of inhomogeneous UAVs considering the technical parameters of the different types of UAVs, the possibility of having restricted airspaces and also time windows. Ribeiro et al. (2019) consider a Mixed Integer Linear Program (MILP) to observe conveyor belts in a mining site. Authors' combinatorial approach focuses on energy consumption and the possibility to place and use charging stations, neglecting the construction of exact trajectories. A combination of an ant colony algorithm with metropolis criterion was applied by Li et al. (2020) to find optimal trajectories for a fleet of homogeneous UAVs on a grid map, including given obstacles and considering safety distance restrictions. Glock and Meyer (2020) design mission plans to maximise the collection of information after fire or chemical incidents. They propose a neighborhood search algorithm, translating flight dynamics, i.e. maximum velocity, acceleration and battery level, into travel times between target locations. Cheng et al. (2020) consider the problem of multi-trip parcel delivery and derive a MINLP formulation where the energy consumption is modeled by a mass-dependent, nonlinear function, and trajectories are given by distances and travel times. Thibbotuwawa et al. (2020) plan the supply of a set of customers with an inhomogeneous fleet of UAVs under weather uncertainty, deploying a MILP formulation. The authors distinguish between several wind zones, each related with corresponding energy consumption and collision avoidance. They then compute the sequence of visited customers, neglecting however the construction of the exact trajectory. Kai et al. (2019) decide the trajectory of a UAV required to visit a given set of points of interest by means of a MILP formulation, where a linearised flight dynamic as well as obstacles or restricted areas are taken into account. The interesting problem of controlling moving ships using a fleet of UAVs stationed at the coast is investigated by Xia et al. (2019). The authors propose a comparison between a MILP formulation and its Lagrangian relaxation. The UAVs are then supposed to meet vessels in the assigned waypoints. Though, besides the battery level, no flight dynamics are properly taken into account. Albert et al. (2017) use a fleet of homogeneous UAVs to observe drifting icebergs in support of shipping. They consider an integer linear optimisation model, with the utilisation of Dubins curves to compute trajectories and updating the present path during the mission. However, their approach neglects most of the flight dynamics for the benefit of the solution process speed. Chen et al. (2016) plan trajectories for a UAV considered as a particle affected by Newton laws. They use optimal control theory to solve the problem and, although given obstacles are considered, authors' approach overlook a detailed flight dynamics and weather conditions. Schmidt and Fügenschuh (2023) consider the planning problem for an inhomogeneous fleet of unmanned aerial vehicles (UAVs). The authors present a two-level time grid approach for a mixed integer program formulation. They also consider a three-dimensional mission area, containing convex-shaped restricted airspaces and convex subareas where the flight trajectories are affected by the wind. Flight dynamics are described by the equations governing the uniformly accelerated motion, where acceleration and velocity are treated as decision variables. Collision avoidance is also included in the model.

The above list of works can not be considered exhaustive and a more complete review on UAVs routing and Trajectory Optimisation (TO) problems is provided in Coutinho et al. (2018), where the authors introduce an additional Multi-phase Mixed Integer Optimal Control formulation for the UAV Routing and Trajectory Optimisation Problem (UAVRTOP) and a taxonomy devoted to UAV routing, task assignment, path planning and trajectory optimisation of UAVs. Several directions for future research are also identified therein, showing that research integrating vehicle routing and trajectory optimisation is still in its dawn.

We contribute to the state-of-the-art by considering the mission and flight planning problem for a fleet of UAVs, specifically gliders, in a mixed-integer linear programming framework, including detailed flight dynamics, the effect of wind, and extending several of vehicle routing and trajectory optimisation aspects to enhance the quality of the computed solution. The novelty of our approach is the combination of assigning waypoints to the participating UAVs and calculating detailed flight trajectories at the same time by means of a two-stage algorithm, also when prior knowledge of equilibrium or initial (steady-state) trajectory guesses are absent or very limited. None of these would be possible without considering mathematical techniques used to solve (robust) optimisation problems. Nocedal and Wright (1999) present several techniques used to exactly solve linear and general nonlinear optimisation problems. On the other hand, the proposed techniques can be exploited for solving multi-objectives problems, see Miettinen (1999), as well as optimisation problems with uncertainty, see Ben-Tal et al. (2001), typically reducing them into a single-objective and deterministic formulation, also known as robust counterpart.

Having presented a comprehensive formulation of the VRTOP, emphasizing the fundamental features and characteristics of the problem derived from both vehicle routing and trajectory optimisation aspects, in the subsequent chapter, we proceed by describing the basics of aircraft and gliders dynamics, introducing the necessary tools to understand and solve the VRTOP considered within this manuscript.

Chapter 3

The Dynamic of Gliders

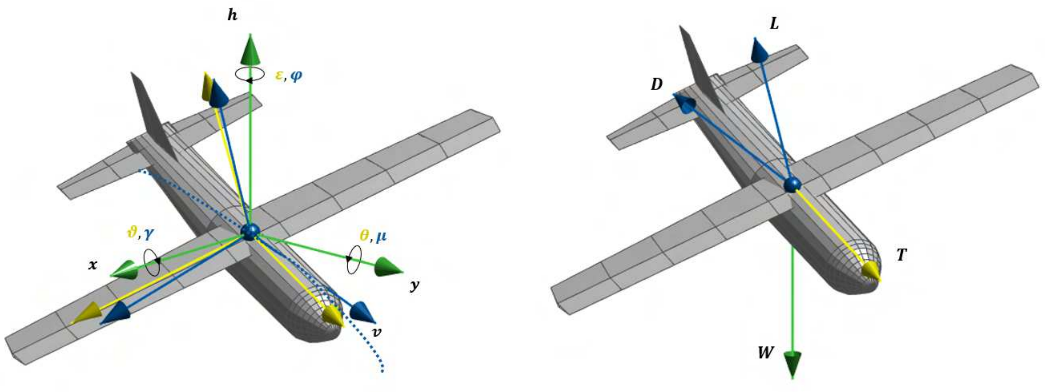
In this chapter, we explore the domain of gliders dynamics and delve into the fundamental equations that dictate the flight of this type of aircraft. Gliders, also known as sailplanes, offer a unique experience by harnessing the forces of nature to maintain extended periods of flight without the aid of onboard propulsion or engine utilisation. Understanding their dynamic is crucial for designing efficient control systems, optimising path planning and flight performances. This chapter begins by exploring the fundamental principles of flight's dynamic, including the concepts of forces like thrust, lift, drag, and weight, continues stating the importance of special aircraft like gliders, and introduces the equations that describe gliders' behavior, also know as the equations of motion (EoMs). In it, we provide the mathematical foundation for analysing and predicting gliders' flight characteristics and we guide the readers through possible techniques for solving the EoMs, with particular focus on the ones used later in thesis. Some example of flight dynamic and optimal control is finally furnished, comprehensive of numerical outcomes.

3.1 Basics of Aircraft Motion and the Flight Envelope

The range of static and dynamic conditions within which an aircraft operates is called its *flight envelope*. To explore the flight envelope, we first must establish frames of reference and define basic elements of position, velocity, force, moment, and equilibrium. The defined elements will also be useful to describe aircraft's equations of motion, presented in the following section. Whatever the frame of reference, we assume that interactions between forces, moments and the aircraft can be described by equations of motion (EoMs) stemming from Newton's laws. In what follows the aircraft is modelled as a point-mass moving over a flat Earth. Among different flight conditions, favourable attention is usually deserved by steady and quasi-steady flight. The first is characterized by constant velocity and altitude, with unchanging aircraft mass, as in wings-level cruise, where there is a balance between forces of lift, weight, thrust, and drag that will be accurately defined later. Being strictly precise, steady flight is possible only if thrust is produced without changing the aircraft's mass. Quasi-steady flight, instead, occurs when time variations in mass or flight condition are slow enough to produce negligible dynamic effects. As the reader might already point out, due to the absence of onboard propulsion, gliders cannot generate thrust and therefore cannot be characterised by (quasi) steady flight conditions. Nevertheless, analogous flight conditions can be considered, characterised by a constant velocity and constant descent rate, and will also be referred to as (quasi) steady flight conditions. Along the chapter, we consider a flat-earth assumption that allows us to neglect the variation of gravitational acceleration with distance from the centre of the Earth, and to disregard relatively subtle Coriolis and centrifugal effects. In this way, it is possible to exploit a clearer description of the glider's motions, useful for investigating the dynamics and control of flight within the atmosphere at suborbital speeds and within short-medium distances and periods of time. Four reference frames can then be considered to describe the aerodynamic forces acting on an aircraft: a basic inertial[1] frame of reference whose origin is fixed at a point on the Earth's surface and axes pointing towards the east (x), north (y), up (h) directions. A body-fixed frame where the origin is at the plane's center of mass (c.m.) and its axes are aligned with the nose, the right wing tip, and the upward perpendicular to the plane defined by the previous two axes. Alternatively, a velocity-axes frame can be considered and it is obtained by translating the origin of the inertial frame to the c.m. while the axis pointing towards north is then rotated and aligned to the direction of motion. Ultimately, a wind-axes frame can be obtained from the body-axes frame aligning the axis from the aircraft's nose towards the direction of motion. The latter, can also be obtained from the velocity-axes frame though a rotation about the velocity vector.

Aircraft's angular orientation with respect to the inertial frame can be expressed in terms of Euler angles, namely the yaw (ε), pitch (ϑ) and roll (θ) angles. In other words, yaw, pitch and roll are the angles needed to convert the inertial frame of reference into the body frame. Analogously, the inertial frame of reference can be converted into the wind-axes frame by means of rotations described by the heading angle (φ), flight path angle (γ) and bank angle (μ). While for a more complete description of the reference frames the reader can refer to Stengel (2004), we depict the just described transformations in Figure 3.1(a). Subsequently, four basic forces acting on the aircraft can be described adopting the above reference frames: thrust, drag, lift and weight. Thrust is the force that propels an aircraft forward through the air. It is generated by by the on-board propulsion systems of the aircraft itself (for a glider, it is absent since there is no engine on board). Drag is the resistance encountered by the aircraft as it moves through the air. Acting upon the airplane's fuselage and being parallel and opposed to the direction of motion, drag is mainly caused by air friction, typically reducing the aircraft's speed and

[1] Most "inertial" frames are not rigorously inertial, but the degree of error in making that assumption is, in some sense, acceptable



(a) Reference frames

(b) Forces Acting on the Aircraft

Figure 3.1: Reference Frames and Forces. In (a), the inertial frame (green), the body-axes frame (yellow) and wind-axes frame (blue) are represented. Angles for transformations from the inertial frame are depicted using the corresponding colours. In (b), thrust (T), drag (D), lift (L) and weight (W) forces are reported with the colour of the reference frame they belong to.

efficiency. Lift is the force generated by airflow through the vehicle's surfaces, primarily wings, enabling an aircraft to overcome gravity and stay airborne. Wings' shape and configuration determine the pressure difference producing the lift. Being perpendicular to the velocity vector and to the "wings axis", this upward force can be thought as pertaining to the wind-axes frame, together with the drag. Ultimately, weight is the force exerted by gravity, pulling the aircraft to the centre of the Earth. In Figure 3.1(b), all the described forces are depicted, each one within the reference frame they naturally belongs to.

By activating the control surfaces, pilots (physical, remote or autonomous) can carefully manage these forces, changing aircraft's orientation and determining safe and efficient flight paths. One of the most important features that characterise the control of an aircraft's horizontal orientation is the Angle-of-Attack (AoA), see Figure 3.2. The AoA denotes the angle formed between the wing's chord line, a straight line connecting the head and tail edges of the wing, and the vector representing the relative motion between the aircraft and the surrounding air. In other words, the AoA represents the difference between aircraft's pitch angle and the flight path angle and it plays a crucial role in determining the lift coefficient and hence in controlling the lift force generated by the aircraft's wings. It is worth mentioning here that a relation similar to the the pitch and flight path angle one exists between the yaw and heading angle, while a less straightforward one describes the transformation between the roll and the bank angle. However, in typical flight conditions and for the purpose of this document the bank angle μ can be considered very nearly similar to the roll angle. Once more we refer the interested reader to the books of Russell (1996) and Stengel (2004) for a more detailed understanding of the relations between reference frames and their transformations.

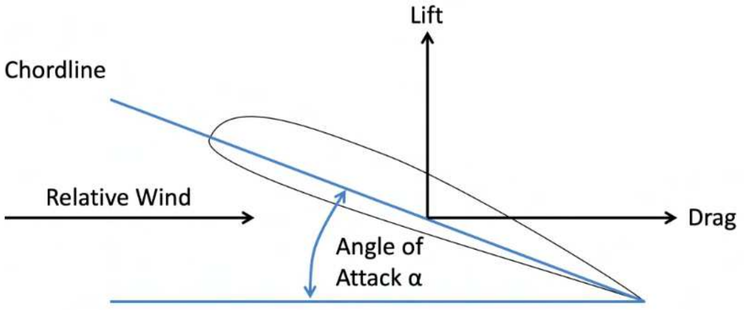


FIGURE 3.2: Wing section and the angle of attack (AoA).

Other essential primary control inputs for the aircraft dynamic are represented by ailerons, elevator and rudder, see Figure 3.3. Ailerons are control surfaces located on the wings that can be moved up or down independently on each wing. Their primary function is to control the rolling motion of the aircraft. By managing the ailerons, it is possible to maintain precise command over the lateral movement and attitude of the aircraft and controlled banked turns can be performed. The elevator instead is a primary control surface located on the horizontal stabilizer of an aircraft, typically at the rear, and it is utilised to control the aircraft's pitch during different flight phases. Similarly, the rudder is located on the vertical stabilizer at the rear of the aircraft. Its primary function is to control the aircraft's yaw during flight and it is particularly important in situations requiring coordinated turns when the aircraft exhibit the tendency to yaw in the opposite direction of a roll. In addition to its primary function, the rudder also plays a role in maintaining directional stability, helping in keeping the aircraft aligned with the desired flight path. All the just described control surfaces operate together to exercise a decisive influence in determining the flight direction and provide precise manoeuvrability of the aircraft.

With the above excursion on the fundamental principles of aircraft motion, we are now prepared to introduce glider's characteristics and peculiarities with respect to other aircraft and to then illustrate the equations that govern their flight dynamic.

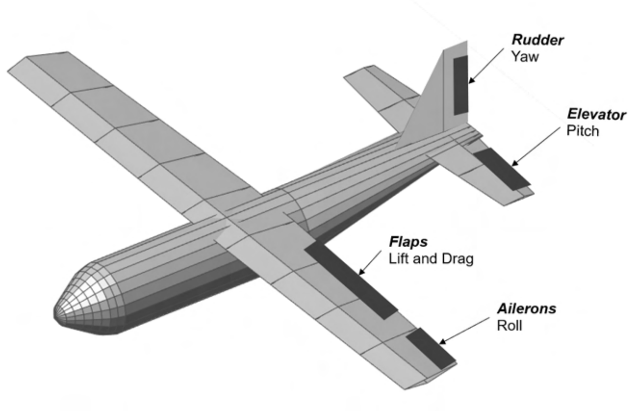


Figure 3.3: Aircraft Control Inputs (excluding engines). Ailerons are controlling surfaces helping changes in the roll angle, elevators help changes in the pitch angle and rudder helps changes in the yaw angle. Flaps helps in changing the AoA.

3.2 Gliders Motion

In this section, we delve into the distinctive features of gliders and we introduce the reader to the the mathematical framework that dictates their dynamics. By doing so, we aim at providing an essential understanding of the glider motion and the equations that shape their trajectory.

3.2.1 Gliders Importance and Characteristics

Gliders are aircraft designed for unpowered flight, meaning that, unlike traditional airplanes, they do not have engines or propulsion onboard and they solely rely on natural forces to fly. Due to this, gliders are typically designed to have long, slender wings that provide lift generation and excellent aerodynamic properties, allowing them to minimize drag, maintain efficiency and stay airborne. They can be launched into the air through various methods, from being towed by a powered aircraft or air-balloons to being launched from the ground using winches or bungee cords. Then gliders take advantage of air currents, known as thermals, and other atmospheric phenomena, to gain altitude and extend their flight time by means of the so-called *soaring techniques*. Historically, the literature distinguishes between *dynamic* and *static* soaring. The former is a nature-inspired powerless flight technique that exploits wind gradients and it has been first studied by Rayleigh (1883) as an explanation for the flight of pelicans and other large birds. With the recent developments in UAV technology, this technique has also become popular for autonomous gliding flight. Among others, Zhao (2004) studies optimal patterns for dynamic soaring, with a particular focus on traveling and loitering patterns.

On the other hand, static soaring relates with methods for extending gliders' flight by extracting energy from thermals. In Kagabo (2010) we can find an example of static soaring and, in addition, the authors show that flight's duration of multiple gliding UAVs could be extended through a combination of both dynamic and static soaring techniques. Thus, glider' performances are often measured by the glide ratio, which represents the distance the glider can travel forward compared to the altitude it loses. Gliders with higher glide ratios are able to cover longer distances during their flights. Due to their manoeuvrability, they can be used for various purposes including civil surveillance, military operations, pilots' training, medicine or goods delivery, as well as recreational flights and competitive sports. For example, Wu et al. (2012) study the optimisation of commercial aircraft flight under engine failure and emergency flight. Authors aim at finding flyable descent trajectory to achieve safe landing, having speed and flight attitude constraints. Coutinho et al. (2019) utilise a fleet of gliders in the context of surveillance and disaster assessment. The authors assume unique (quasi) steady flight conditions to linearise the aircraft dynamic and plan gliders' routes.

In addition, the utilisation of gliders offers several advantages. First of all they are environmentally friendly: gliders are silent and emission-free since they do not require any engine to stay airborne. Second, without the need for fuel or engine maintenance, the ongoing expenses can be significantly reduced, making gliders a cost-efficient option compared to powered aircraft. Then, gliders ability to fly quietly and for extended periods makes them ideal for gathering data without disturbing the natural environment. This makes them a valuable alternative in the context of scientific research, especially when dealing with atmospheric and environmental studies. Lastly, gliders do not require long runways or complex airport infrastructure. They can take-off and land on grass airstrips or specialized glider launch areas, reducing the need for extensive ground facilities. Crispin (2016) proposes the use of balloon-launched autonomous gliders in the context of atmospheric research. The author's goal is to optimise the trajectory of multiple gliders in order to achieve uniform air sampling within a specified airspace and to efficiently map the essential physical and chemical properties of the air.

Having understood the importance of gliders, as well as their advantages and their broad application range, in the following sections, we introduce the equations governing gliders' flight dynamics and we propose techniques to solve them that will turn to be useful for the overall mathematical formulation of the problem considered in this thesis.

3.2.2 Gliders Equations of Motion (EoMs)

The equations of motion (EoMs) play a vital role in capturing the intricate relationship between aerodynamic forces and in describing the behavior of an aircraft during its flight. By comprehending these equations, we gain a deeper understanding of how these forces, control inputs and external influences interact and contribute to the overall dynamic as well as the final path planning. As previously stated, influences on the aircraft dynamic are derived from Newton's laws. In accounting for weather dependencies when analysing the performance and behavior of an aircraft, especially a glider, it is often convenient to express these laws and the resulting equations adopting a wind-axes reference frame. Gliders' physical elements employed in this thesis combine aspects derived from the complementary designs suggested by Bower (2010) and Flanzer (2012). More precisely, most of the parameters in Table 3.1 are gathered from Bower's work, whereas reasonable lower and upper bounds for state and control variables are established based on Flanzer's research. Both authors rigorously tested and validated the models in diverse real flight conditions. Finally, following Zhao (2004), we can write the system of ordinary differential equations (ODEs) modelling the motion of aerial gliders based on the following assumptions:

1. the range of flight is negligible compared to Earth's circumference (a flat Earth);
2. the air density is constant;
3. the wind is stationary, blowing horizontally and can be described by a linear profile as a function of the flight altitude;
4. the mass of the glider does not change during the flight.

Then, let us write $\mathbf{y}(\tau) = (x(\tau), y(\tau), h(\tau), v(\tau), \gamma(\tau), \varphi(\tau))$ to define the variables representing the status (or state) of a glider at time $\tau \in [\tau_0, \tau_f]$, with the first three components $x(\tau), y(\tau)$ and $h(\tau)$ indicating the position of the UAV with respect to the inertial reference frame, whereas $v(\tau), \gamma(\tau)$ and $\varphi(\tau)$ are quantities referred to the wind-axes frame representing the velocity of the aircraft with respect to the air (airspeed module), flight path angle (from horizontal) and the heading angle (from north) respectively. Given that it is not unrealistic to assume that the primary controls previously described can arbitrarily influence the aircraft's lift and banking angle, we consider the control variables (or input) to the system as described by the vector $\mathbf{u}(\tau) = (C_L(\tau), \mu(\tau))$, where $C_L(\tau)$ is the lift coefficient, while $\mu(\tau)$ is indeed the bank angle.

With this in mind, omitting the explicit time dependence to simplify the notation, the EoMs describing the dynamic of a glider can finally be expressed by the following system of equations

$$\dot{x} = vcos(\gamma)sin(\varphi) + U \tag{3.1}$$
$$\dot{y} = vcos(\gamma)cos(\varphi) \tag{3.2}$$
$$\dot{h} = vsin(\gamma) \tag{3.3}$$
$$m\dot{v} = -D - mgsin(\gamma) - m\dot{U}cos(\gamma)sin(\varphi) \tag{3.4}$$
$$mv\dot{\gamma} = Lcos(\mu) - mgcos(\gamma) + m\dot{U}sin(\gamma)sin(\varphi) \tag{3.5}$$
$$mvcos(\gamma)\dot{\varphi} = Lsin(\mu) - m\dot{U}cos(\varphi), \tag{3.6}$$

with

$$L = \frac{1}{2}\rho S_w C_L v^2 \tag{3.7}$$
$$D = \frac{1}{2}\rho S_w C_D v^2 \tag{3.8}$$
$$C_D = C_{D_0} + k_A C_L^2 \tag{3.9}$$
$$k_A = \frac{1}{\pi \frac{b^2}{S_w} O_f} \tag{3.10}$$
$$U = U(\beta, h) = \beta h. \tag{3.11}$$

Here m represents the mass of the UAV, g is the gravity constant and U is the wind speed, which is assumed to increase linearly with the height, see Zhao (2004). Without loss of generality (W.L.O.G.), the wind is blowing towards east (x direction) and its strength coefficient β, sometimes referred to as *wind intensity* along the manuscript, can be chosen to linearly approximate the average wind gradient profile in UK, as suggested in Drew et al. (2013), for a reference altitude of $\simeq$ 500 meters. L and D represents the lift and drag forces respectively and can be expressed better by equations (3.7)-(3.8), being C_{D_0} the drag coefficient at *zero-lift*, ρ the air density and S_w the glider total wings' area. Finally, b expresses glider's wing span, while O_f is the *Oswald efficiency factor* estimated using the Matlab function available in Sartorius (2023). We report in Table 3.1 the values of the EoMs' constants that will be used in this manuscript.

Constant	Value	Description	Unit
ρ	1.22543	Density of the air at sea level	kg/m^3
g	9.80665	Gravity of Earth at sea level	m/s^2
C_{D_0}	0.01730	Coefficient of drag at zero-lift	dimensionless
k_A	0.03200	Aerodynamic coefficient	dimensionless
m	1.99000	Mass of the UAV	kg
S_w	0.48500	UAV's total wing area	m^2

Table 3.1: Environmental and UAV constants

The Equations of Motion can be expressed in a more concise and compact form using the following notation

$$\dot{\mathbf{y}}(\tau) = \mathbf{f}(\mathbf{y}(\tau), \mathbf{u}(\tau), \mathbf{w}(\tau)),$$

where $\mathbf{f} = (f_1, \ldots, f_6)$ represents the right hand side (RHS) of the equations (3.1)-(3.6), eventually divided by the terms on the LHS that do not express time derivatives, while $\mathbf{w}(\tau) = \beta(\tau)$ can be seen as a term describing the external influences on the glider's behaviour. In addition, it is worth noting that the system of equations under consideration is autonomous, meaning that it does not explicitly depend on time. This implies that the dynamics of the system remain unchanged regardless of the specific moment at which they are observed and become solely determined by the initial conditions and the system state, simplifying the analysis and enabling a deeper understanding of its properties.

3.3 A Direct Collocation Method to Solve Gliders EoMs

As an illustration of how to effectively integrate and solve the aforementioned system of equations, we will consider it embedded within a specific trajectory optimisation problem where we require a glider to be guided from an initial state $\mathbf{y}_0$ to a desired final state $\mathbf{y}_f$ in the minimum amount of time possible. Including the exemplification within a TO problem will also enable us to observe the corresponding methodology used in the subsequent chapters, gaining therefore a clearer understanding. It is then worth mentioning that, despite the great variety of direct methods in the literature, we chose to deploy a direct collocation method to determine the control inputs allowing the glider to traverse the desired trajectory efficiently. This example will enable us to witness practical means to solve complex trajectory optimisation problems, providing useful insights and a better understanding of glider dynamics.

3.3.1 Nonlinear Programming Problem Derivation

Considering the optimal control problem requiring a glider fly from $\mathbf{y}_0$ to $\mathbf{y}_f$ in the shortest time:

$$\min_{\mathbf{y}(\tau),\mathbf{u}(\tau)} \quad J(\mathbf{y}(\tau), \mathbf{u}(\tau), \tau, \mathbf{p}) = \tau_f$$

$$\begin{aligned} \text{subject to } \dot{\mathbf{y}} &= \mathbf{f}(\mathbf{y}(\tau), \mathbf{u}(\tau), \mathbf{w}(\tau)) \\ \mathbf{h}(\mathbf{y}(\tau), \mathbf{u}(\tau), \mathbf{p}) &\leq 0 \\ \phi(\mathbf{y}(\tau_0), \mathbf{y}(\tau_f), \mathbf{p}) &\leq 0, \end{aligned}$$

with $\mathbf{f}$ representing the just introduced EoMs, $\mathbf{h}$ describing limitation of state and control variables of the type

$$\mathbf{y}_{lb} \leq \mathbf{y}(\tau) \leq \mathbf{y}_{ub}$$
$$\mathbf{u}_{lb} \leq \mathbf{u}(\tau) \leq \mathbf{u}_{ub}$$

and ϕ determining the initial and final conditions

$$\mathbf{y}(\tau_0) = \mathbf{y}_0$$
$$\mathbf{y}(\tau_f) = \mathbf{y}_f$$

as previously stated, we aim at transforming it into a nonlinear programming problem (NLP) by applying some direct collocation method. To do so, we rely on the discretisation of the solution time space as well as on an integration scheme to numerically solve the dynamic constraints described by the above presented system of ODEs

$$\dot{\mathbf{y}} = \mathbf{f}(\mathbf{y}, \mathbf{u}, \mathbf{w}),$$

where we omit for a moment the time dependency for simplicity of notation. Henceforth, we start dividing the time interval of interest, $[\tau_0, \tau_f]$, into N smaller intervals or time steps, such that

$$\tau_0 < \tau_1 < \cdots < \tau_N = \tau_f,$$

where we chose for our example an equally spaced time partition, that is a partition where every time step has the same length equal to

$$\eta = (\tau_{t+1} - \tau_t) = \frac{\tau_f - \tau_0}{N}$$

for every $t = 0, \ldots, N-1$. Subsequently, we want to determine the value of the state and control variables $(\mathbf{y}_0, \mathbf{u}_0, \ldots, \mathbf{y}_N, \mathbf{u}_N)$ at each time step (collocation point). Note once again that we write $\mathbf{y}_t$ for $\mathbf{y}(\tau_t)$, $\mathbf{u}_t$ in place of $\mathbf{u}(\tau_t)$, $\mathbf{f}_t$ for $\mathbf{f}(\mathbf{y}(\tau_t), \mathbf{u}(\tau_t), \mathbf{w}(\tau_t))$ and similarly for the other time-dependent quantities of interest. It is then necessary to numerically integrate the ODEs and to do so, we decide to deploy an Euler scheme of integration, where we estimate the next state $\mathbf{y}_{t+1}$ by applying the following approximation

$$\frac{\mathbf{y}_{t+1} - \mathbf{y}_t}{\eta} \simeq \dot{\mathbf{y}}_t = \mathbf{f}_t,$$

or equivalently, writing $\mathbf{y}_{t+1}$ based on the current state $\mathbf{y}_t$ and the current derivative function $\mathbf{f}_t$

$$\mathbf{y}_{t+1} \simeq \mathbf{y}_t + \eta \mathbf{f}_t. \tag{3.12}$$

Then, if we define the *defects* ζ_t to be

$$\zeta_t := \mathbf{y}_{t+1} - \mathbf{y}_t - \eta \mathbf{f}_t,$$

we can specify the aerodynamic constraints relative to the glider motion by simply equating the defects to zero. In other words, we can write

$$\zeta_t = 0 \qquad \forall t = 0, \ldots, N-1.$$

It is worth mentioning here that, as seen in the previous chapter, other numerical methods and defects definitions can be used. However, their investigation falls outside the scope of this manuscript and we refer the interested reader to the books of Stengel (2004), Betts (2009) and Hairer et al. (2011), or to the article of Coutinho et al. (2019) where detailed analyses and comparison of the different integration schemes applied to a glider routing and trajectory optimisation problem are discussed. Sometimes, it is desirable to avoid wobbling trajectories and sudden changes in manoeuvrability decisions. For this reason, it might be convenient to add penalty terms to the objective aiming at reducing the amount of changes in decision making as well as avoiding sharp turns and direction adjustments. An example of penalty term can be

$$p_s = \sum_{t=0}^{N-1} (x_{t+1} - x_t)^2. \tag{3.13}$$

Indicating with S the number of penalty terms, we can finally write the resulting problem in a compact form as

$$\begin{aligned}
\text{minimize } J' &= \tau_f + \sum_{s=1}^{S} p_s && (3.14)\\
\text{subject to } \zeta_t &= 0 \qquad \forall k = 0, \ldots, N-1 && (3.15)\\
\mathbf{y}_t &= \mathbf{y}_0 \qquad t = 0 && (3.16)\\
\mathbf{y}_t &= \mathbf{y}_f \qquad t = N && (3.17)\\
(\mathbf{y}_t, \mathbf{u}_t) &\in \mathscr{Y} \times \mathscr{U} \qquad \forall t = 0, \ldots, N, && (3.18)
\end{aligned}$$

where $\mathscr{Y} \times \mathscr{U} \subseteq \mathbb{R}^{6\times 2}$ is the set of admissible states and controls defined by $\mathbf{h}$. Given the above NLP formulation, it is possible to solve the considered trajectory optimisation problem by leveraging on state-of-the-art nonlinear solver, like SNOPT (see Gill et al.). Nonetheless, it is crucial noticing here that solving an optimal control problem, might lead to further challenges. In particular, the existent nonlinearities can cause the problem to be potentially nonconvex, high-dimensional, unstable and computationally complex. The nonconvexity of the problem poses significant difficulties in the identification of global optimal solutions. On the other hand, collocation methods require a discretisation of the time domain and choices of collocation points can impact the accuracy of the solution as well as the computational complexity. Typically it is possible to observe a reduction of the discretisation errors and growing computational times as we increase the number of collocation points and consequently the dimension of the problem. In addition, depending on the specific solver and the adopted solution technique, it is possible to encounter convergence and stability issues, including sensitivity to initial conditions where small changes of $\mathbf{y}_0$, initial trajectory guesses or system parameters can result in significant variations in the optimised trajectory, sometimes leaving the solver unable to provide a solution. Addressing these difficulties carefully is essential to the successful construction of the optimal trajectory.

In the following section and in the rest of the manuscript, we propose different techniques helping in (i) overcoming some of the above challenges and (ii) inserting vehicle routing characteristics to the problem, to produce sensible and reliable path optimising the given objectives.

3.3.2 Linearisation of the EoMs

Nonlinear programming formulation of an optimal control problem might lead to several challenges, some of which have been introduced in the previous section. Linearising the equations of motion might help overcoming some of them, improving computational efficiency, reducing the instability in the solutions, while at the same time making easier to analyse the behavior of the glider. To do so, prior knowledge on the glider and its flight conditions is usually required. Thus, lets assume first that an initial estimate of the optimal trajectory $(\mathbf{y}^o(\tau), \mathbf{u}^o(\tau))$, also known as *nominal trajectory*, is provided. It is then possible to express all the variables as resulting from small perturbation $\Delta(.)$ of the initial trajectory. More precisely, we can write

$$\begin{aligned} \mathbf{y}(\tau) &= \mathbf{y}^o(\tau) + \Delta\mathbf{y}(\tau) \\ \mathbf{u}(\tau) &= \mathbf{u}^o(\tau) + \Delta\mathbf{u}(\tau), \end{aligned}$$

or alternatively

$$\begin{aligned} \Delta\mathbf{y}(\tau) &= \mathbf{y}(\tau) - \mathbf{y}^o(\tau) \\ \Delta\mathbf{u}(\tau) &= \mathbf{u}(\tau) - \mathbf{u}^o(\tau). \end{aligned}$$

Discretising time as before and recalling then that constraints (3.15) are based on an Euler scheme of integration and are equivalent to the equations

$$\mathbf{y}_{t+1} = \mathbf{y}_t + \eta\ \mathbf{f}_t,$$

a common practice to linearise the glider's dynamic is to utilise the first-order Taylor expansion of $\mathbf{f}_t$, enabling to approximate it as follows

$$\mathbf{f}_t = \mathbf{f}(\mathbf{y}_t, \mathbf{u}_t) = \mathbf{f}(\mathbf{y}_t^o, \mathbf{u}_t^o) + \frac{\partial \mathbf{f}}{\partial \mathbf{y}}\Big|_{(\mathbf{y}_t^o, \mathbf{u}_t^o)} \Delta \mathbf{y}_t + \frac{\partial \mathbf{f}}{\partial \mathbf{u}}\Big|_{(\mathbf{y}_t^o, \mathbf{u}_t^o)} \Delta \mathbf{u}_t + o(\|(\Delta \mathbf{y}_t, \Delta \mathbf{u}_t)\|),$$

where we omit $\mathbf{f}_t$ dependency on external disturbances for simplicity in the notation. Therefore, if we define

$$\tilde{\mathbf{f}}_t = \tilde{\mathbf{f}}(\mathbf{y}_t, \mathbf{u}_t) := \mathbf{f}(\mathbf{y}_t^o, \mathbf{u}_t^o) + \frac{\partial \mathbf{f}}{\partial \mathbf{y}}\Big|_{(\mathbf{y}_t^o, \mathbf{u}_t^o)} \Delta \mathbf{y}_t + \frac{\partial \mathbf{f}}{\partial \mathbf{u}}\Big|_{(\mathbf{y}_t^o, \mathbf{u}_t^o)} \Delta \mathbf{u}_t,$$

we have that

$$\mathbf{f}_t \simeq \tilde{\mathbf{f}}_t$$

and we can write the approximated aerodynamic constraints as

$$\mathbf{y}_{t+1} = \mathbf{y}_t + \eta\, \tilde{\mathbf{f}}_t, \tag{3.19}$$

where $\tilde{\mathbf{f}}_t$ is finally a linear function of $(\mathbf{y}_t, \mathbf{u}_t)$. It is worth noticing here that also $(\mathbf{y}^o, \mathbf{u}^o)$ can depend on time.

Remark 3.1. Strictly speaking, the linearity of constraints (3.19) is reached when also the time partition length is fixed and $\eta = (\tau_f - \tau_0)/N$ does not depend on the arrival time τ_f, now considered as a variable. In the following chapter, we will see how the introduction of binary variables and upper bounds on the flight time can serve the objective of keeping the time partition length fixed, avoiding the need of fixing the arrival time τ_f.

Unfortunately, in facing planning problems such the vehicle routing and trajectory optimisation ones, it is often the case that information about an initial (optimal) trajectory are limited or absent. Nonetheless, with little prior information about the initial and final conditions and within certain settings of the point of interest, it is possible to construct reasonable nominal trajectories taking advantages of equilibria or (quasi) steady flight conditions, which are situations characterised by the glider flying at a constant speed, direction and descent rate. In particular, we can have equilibrium flight conditions when $\dot{\mathbf{y}} = \mathbf{f}(\mathbf{y}, \mathbf{u}) = \mathbf{c}$, with $\mathbf{c} = (c_x, c_y, c_h, c_v, c_\gamma, c_\varphi) \in \mathbb{R}^6$ and

$$\dot{v} = c_v = 0 \tag{3.20}$$

$$\dot{\gamma} = c_\gamma = 0 \tag{3.21}$$

$$\dot{\varphi} = c_\varphi = 0. \tag{3.22}$$

Considering then a stable air mass, the absence of wind ($\beta = 0$) and assuming moreover that

- the glider is flying flat, or better stated, the flight path angle γ is reasonably small and the UAV is not banking, i.e. $\mu = 0$;
- the glider heading angle is aligned with the vector pointing from the starting point $\mathbf{y}_0$ to the final point $\mathbf{y}_f$;
- the value of the lift coefficient C_L is fixed;

it is possible to solve the system $\mathbf{f}(\mathbf{y}, \mathbf{u}) = \mathbf{c}$ obtaining that

$$\begin{aligned} \dot{x} = f_1 = &\quad c_x, & \dot{v} = f_4 = &\quad 0, \\ \dot{y} = f_2 = &\quad c_y, & \dot{\gamma} = f_5 = &\quad 0, \\ \dot{h} = f_3 = &\quad c_h, & \dot{\varphi} = f_6 = &\quad 0. \end{aligned}$$

In particular, given the small angle assumptions, we have that $sin(\gamma) \simeq \gamma$, $cos(\gamma) \simeq 1$, $sin(\mu) = 0$ and $cos(\mu) = 1$. Then, denoting with φ_{eq} the angle such that φ is aligned with $(x_f, y_f) - (x_0, y_0)$, and with $C_{L_{eq}}$ the chosen value for the coefficient of lift, from equalities (3.4)-(3.5) we can derive the system of equations

$$\begin{aligned} \frac{1}{2m}\rho S_w(C_{D_0} + k_A C_{L_{eq}}^2)v^2 + g\gamma = 0 \\ \frac{1}{2m}\rho S_w C_{L_{eq}} v - \frac{g}{v} = 0 \end{aligned}$$

and obtain the following initial trajectory

$$x^o(\tau) = x_0 + v^o sin(\varphi^o)\tau \tag{3.23}$$

$$y^o(\tau) = y_0 + v^o cos(\varphi^o)\tau \tag{3.24}$$

$$h^o(\tau) = h_0 + v^o sin(\gamma^o)\tau \tag{3.25}$$

$$v^o(\tau) = v_{eq} = \sqrt{\frac{2mg}{\rho S_w C_{L_{eq}}}} \tag{3.26}$$

$$\gamma^o(\tau) = \gamma_{eq} = -\frac{C_{D_0} + k_A C_{L_{eq}}^2}{C_{L_{eq}}} \tag{3.27}$$

$$\varphi^o(\tau) = \varphi_{eq} \tag{3.28}$$

$$C_L^o(\tau) = C_{L_{eq}} \tag{3.29}$$

$$\mu^o(\tau) = \mu_{eq} = 0. \tag{3.30}$$

Applying the same principles seen above when prior knowledge on the initial trajectory is available, we can then linearise the aerodynamic constraints and solve the approximated trajectory optimisation problem.

Remark 3.2. Among all the equilibrium flight conditions, it is usually convenient to exploit a preferred one that, keeping all the other assumptions the same, optimises the drag to lift ratio (D/L). In particular, equating the derivative $\frac{\partial(D/L)}{\partial C_L}$ to zero, we have that

$$\frac{\partial(D/L)}{\partial C_L} = \frac{C_{D_0}}{C_L^2} + k_A = 0,$$

and solving for C_L, we can find the value $C_{L_{eq}} = \sqrt{\frac{C_{D_0}}{k_A}}$.

Finally, many situations occur where neither prior knowledge on the initial trajectory is available, nor clear equilibrium flight conditions can be derived, and addressing the linearisation challenge necessitates additional efforts and the need for novel tailored strategies representing one of the main contribution of this work. Therefore, the exploration of alternative approaches that do not rely on prior initial trajectory information will be undertaken at a later stage in the thesis, particularly in Chapter 5.

In the following section, we provide numerical examples for solving the EoMs by using instances of the illustrative trajectory optimisation problem described above, also highlighting results for the glider flight conditions as well as the (dis)advantages presented by a nonlinear numerical approach.

3.4 Glider Flight Example

In this section, we present numerical results obtained from solving exemplary instances of the afore-mentioned trajectory optimisation problem, offering valuable insights into the performance, stability and control characteristics of the glider. While analysing the positive outcomes of the solutions obtained via the deployment of the above-mentioned direct method, we also acknowledge and discuss the limitations and drawbacks that emerged during the analysis. By providing a comprehensive assessment of both the benefits and complications of the approach adopted, we aim at gaining deeper understanding of the glider's behavior as well as an enhanced comprehension of the challenges arising from the inclusion of flight dynamics in practical applications.

3.4.1 NLP solution

At first, we propose a solution to the NLP formulation of the glider trajectory optimisation problem (3.14)-(3.18) described before. In our simulations, we consider a glider with initial state

$$\mathbf{y}_0 = (x_0, y_0, z_0, v_0, \gamma_0, \varphi_0) = (0, 0, 261, 7, -0.0274, 0),$$

Variable	Lower Bound	Upper Bound	Unit
x	-5000	5000	m
y	-5000	5000	m
z	0	1500	m
v	5	40	m/s
γ	$-\frac{\pi}{3}$	$\frac{\pi}{3}$	$radians$
φ	-4π	4π	$radians$
C_L	0.1	1.17	$dimensionless$
μ	$-\frac{\pi}{3}$	$\frac{\pi}{3}$	$radians$

Table 3.2: Glider flight envelope. For each variable, we report here the upper and lower bounds as well as the unit of measure used.

that is required to reach a final location

$$(x_f, y_f, z_f) = (1550, 0, 0),$$

with a final moderate speed $v_f \leq 5.5$ m/s and within a time span of $[t_0, t_f] = [0, 295]$ seconds. It is then straightforward to understand that $\tau_f \leq t_f$. Following Flanzer (2012), it is possible to determine the glider flight envelope, used for the rest of the manuscript unless otherwise stated, as in Table 3.2.

Furthermore, as previously mentioned, it is often convenient to require smoothness of the trajectory. For this reason, we insert some penalty terms: first, we limit abrupt decisions' changes, possibly avoiding extremely steep turns, by penalising control variables adjustments

$$p_1 = \sum_{t=0}^{N-1} (C_{L_{t+1}} - C_{L_t})^2$$
$$p_2 = \sum_{t=0}^{N} \mu_t^2.$$

Consequently, we penalise changes in altitude, that will also help the glider to fly sufficiently far from the ground

$$p_3 = \sum_{t=0}^{N-1} (z_{t+1} - z_t)^2.$$

We also ask to avoid sudden accelerations

$$p_4 = \sum_{t=0}^{N-1} (v_{t+1} - v_t)^2,$$

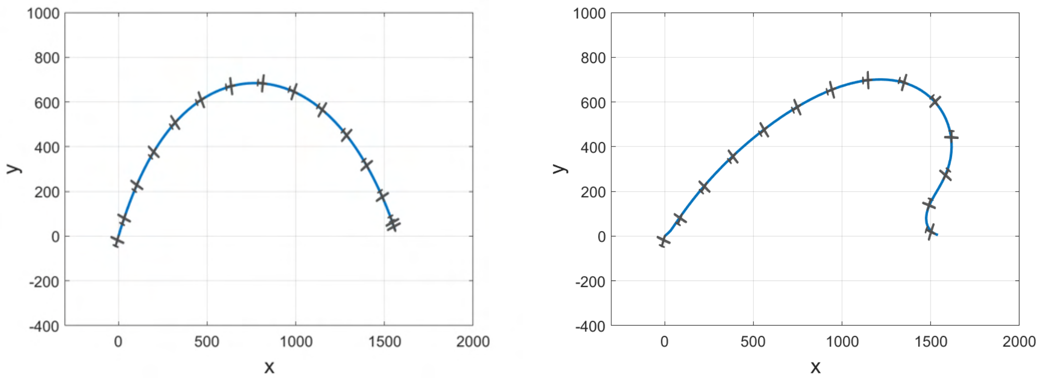
as well as sudden deep dives

$$p_5 = \sum_{t=0}^{N} (\gamma_{t+1} - \gamma_t)^2.$$

Finally, we want to avoid wobbling trajectories that exhibit continuous heading direction changes

$$p_6 = \sum_{t=0}^{N-1} (\varphi_{t+1} - \varphi_t)^2.$$

The other parameters and inputs to the EoMs or their nonlinear programming representation are given in Table 3.1. Deploying then a uniform time partition with $N = 94$ collocation points, it is possible to solve the glider trajectory optimisation problem using a state-of-the-art nonlinear solver like SNOPT on a Viglen desktop, Intel(R) Core(TM) $\mathrm{i}5-7500$ CPU with 3.40GHz and a RAM of 8 GB, running under Windows 10. Figure 3.4 illustrate the optimal trajectory found first considering a scenario in absence of wind ($\beta = 0$), Figure 3.4(a), and then a scenario with typical wind conditions, Figure 3.4(b). In the latter, the wind strength coefficient $\beta = 0.025$ is such that it linearly approximates the average wind gradient profile for the UK, as suggested by Drew et al. (2013), for a reference altitude of nearly 500 meters.

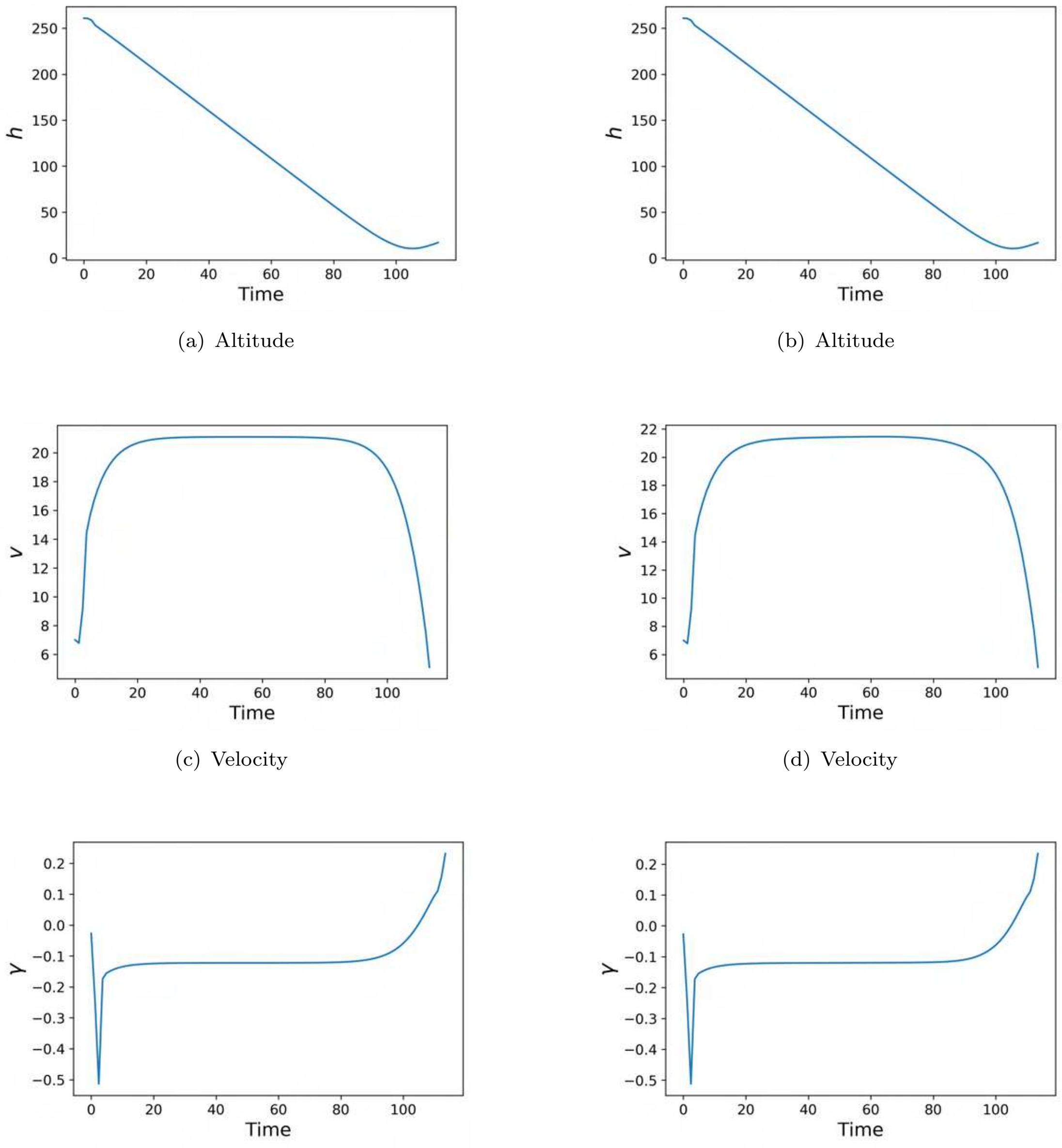


(a) $\beta = 0$

(b) $\beta = 0.025$

FIGURE 3.4: Optimal glider trajectories seen from above both in absence (a) and in presence of wind (b).

As expected, it is possible to notice that the wind's presence forces the glider to fly more towards east (x direction) and then approach the arrival point from another direction, being able to adjust its trajectory at lower altitudes. Furthermore, a better understanding of glider performances and its flight conditions can be obtained examining Figure 3.5, where the altitude (h), velocity (v) and flight path angle (γ) are illustrated again for the case with absence and presence of wind, and Figure 3.6, where the decisions made in terms of control variables are also examined. As it is possible to notice, the glider necessitate of an initial adjustment due to the fact that the glider starts with a very



(a) Altitude

(b) Altitude

(c) Velocity

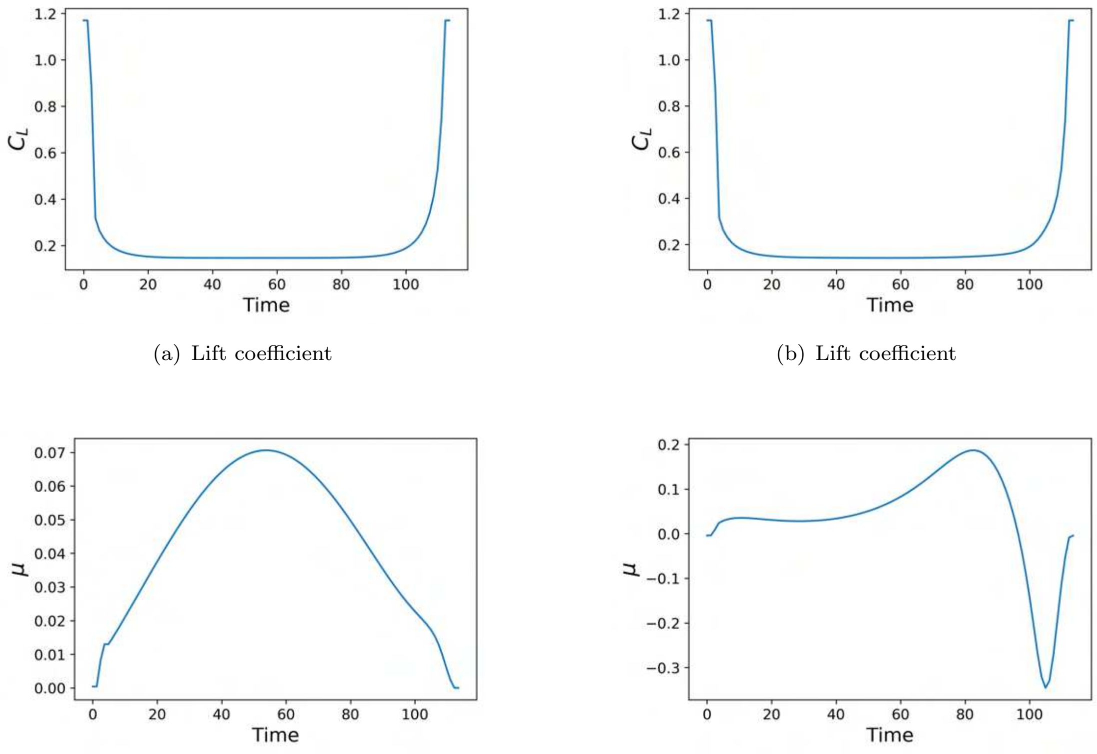
(d) Velocity

(e) Flight path angle

(f) Flight path angle

FIGURE 3.5: Some state variables profiles obtained by solving the NLP. Solutions attained in absence of wind are shown on the left, whereas the ones attained in presence of wind on the right.

low speed and an attitude close to stalling conditions. Consequently, a sudden change in γ is required, after which the glider gains a sufficient velocity and tends to descent at a regular pace, keeping its speed and flight path angle as constant as possible during the flight and then changing them for the landing period. Nonetheless, although gliders altitude, speed and flight path angle profiles show a similar behaviour regardless the presence of wind, it is possible to clearly observe the necessity to adjust the direction of flight according to it, leading to a significantly different solution in terms of bank angle choices.



(a) Lift coefficient

(b) Lift coefficient

(c) Bank angle

(d) Bank angle

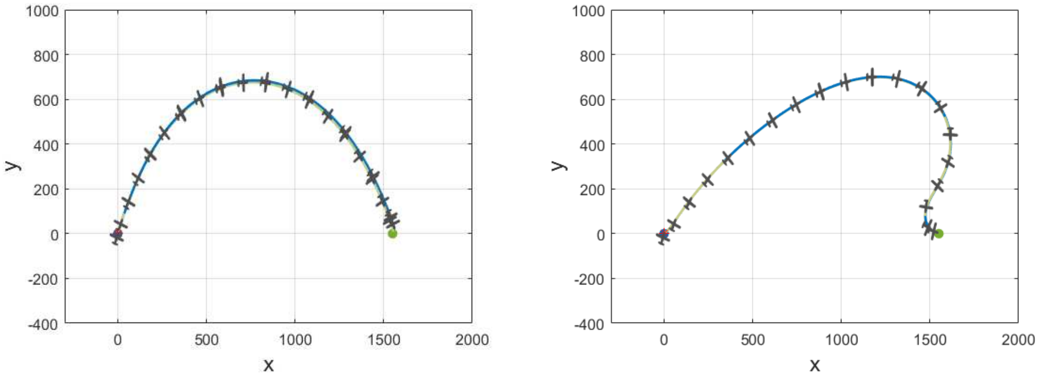
Figure 3.6: Control variables profiles obtained by solving the NLP. Decisions attained in absence of wind are shown on the left, whereas the ones attained in presence of wind on the right.

It is worth mentioning once more here that NLP-based solutions are sensitive to model parameters, time partition and initial guesses, with cases where convergence can be only ensured if a proper initialisation is provided Zhao (2004). Even in the example just shown, changing the time partition might lead to significant change in the landing time of the solution just shown by up to 40%. With this in mind, we provide now the reader with an example of the same problem where the trajectory is linearised around the found optimal solution.

3.4.2 Linearised Dynamic and Comparision with NLP

Analogously to the above, in this section we propose a solution to the same instance of the glider trajectory optimisation problem (3.14)-(3.18) this time considering a linear formulation where an initial optimal trajectory is provided. We compare then the solution found with the nonlinear solver, with the one obtained using the linear formulation. This illustration aims at validating the utilised linearisation method as well as showing its reliability and potential advantages. We solve the derived glider trajectory optimisation problem using a state-of-the-art linear solver like Gurobi 10.0.1, deployed again on a Viglen desktop, Intel(R) Core(TM) $i5 - 7500$ CPU with 3.40GHz and a RAM of 8

GB, running under Windows 10. Similarly to the previous case, Figure 3.7 illustrate an example of the optimal trajectories found.



(a) $\beta = 0$

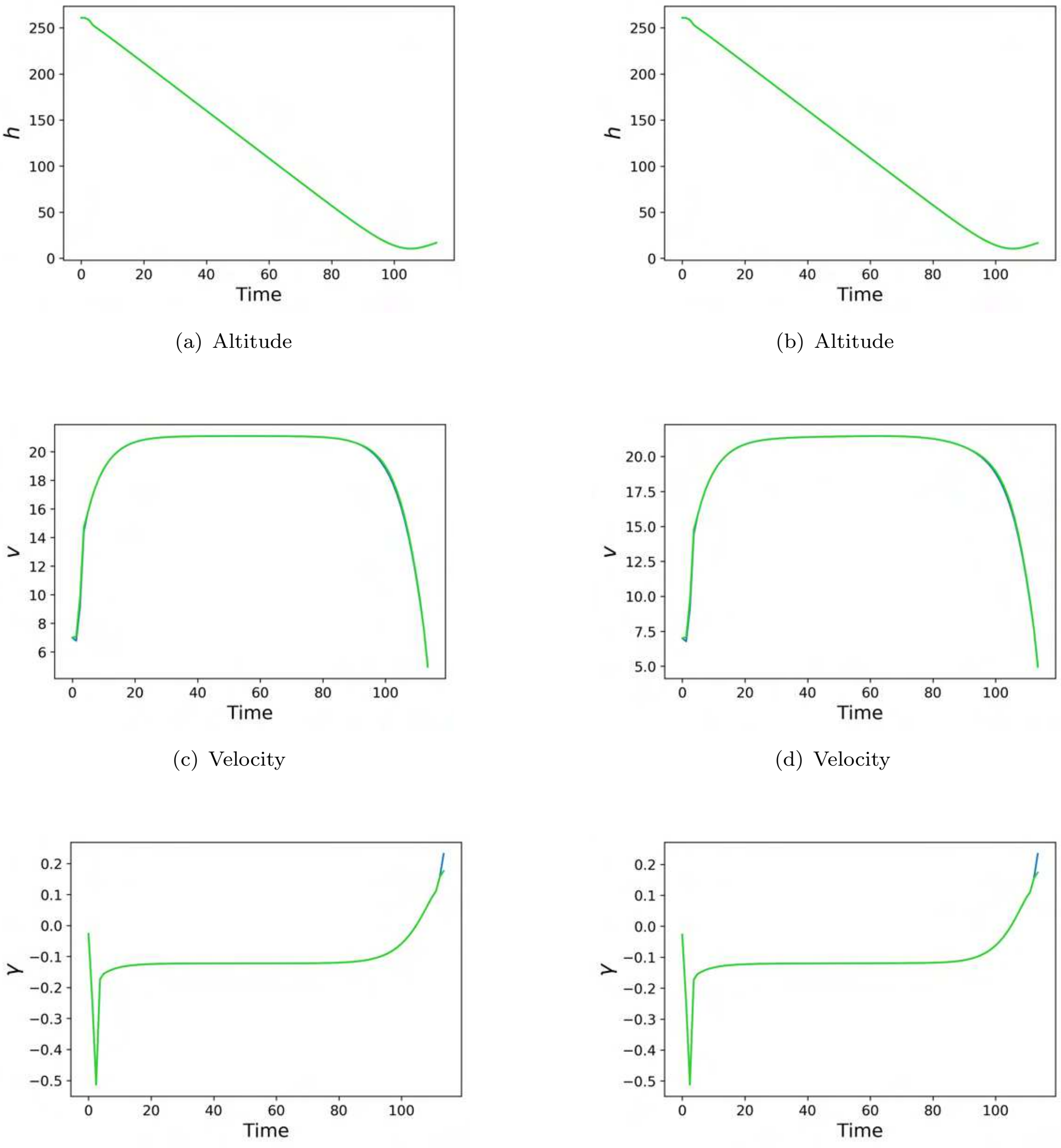
(b) $\beta = 0.025$

FIGURE 3.7: Comparison of the optimal solutions obtained from the NLP (blue) and the linearised dynamic (green), both in absence (a) and in presence of wind (b).

Again, a better understanding of glider performances and its flight conditions can be obtained by examining Figure 3.8 and Figure 3.9, where profiles of states and control variables stemming from the linearised solution are compared with the ones found by solving the NLP.

As desired, trajectories stemming from the solutions of the problems with linearised EoMs, as well as the corresponding variables profiles, result to be close to the ones found by solving the NLP.

In light of this, it is clear that the inclusion of flight dynamics in practical applications, as well as its integration within vehicle routing and trajectory optimisation problems, presents specific challenges that are critical to achieve efficient and reliable operations, some of which have been just presented. The high dimensionality of the solution space renders it difficult to exhaustively explore the flight envelope, with nonlinear dynamics increasing computational complexity and solutions instability. Linearisation of the EoMs helps in reducing those challenges, but maintain the difficulty of finding sound flyable trajectories in absence of good initial solutions or insights. Hence, it is indisputable that a thoughtful approach is required to address these challenges and reach a reliable and tractable integration of trajectory optimisation within vehicle routing problems. In the following chapters, we present first a method to solve the VRTOP when some prior knowledge about the initial trajectory or equilibrium conditions is available. Then, we discuss the intricate problem of finding optimal trajectories and reliable paths for the VRTOP, also in cases when prior knowledge is limited or absent.



(a) Altitude

(b) Altitude

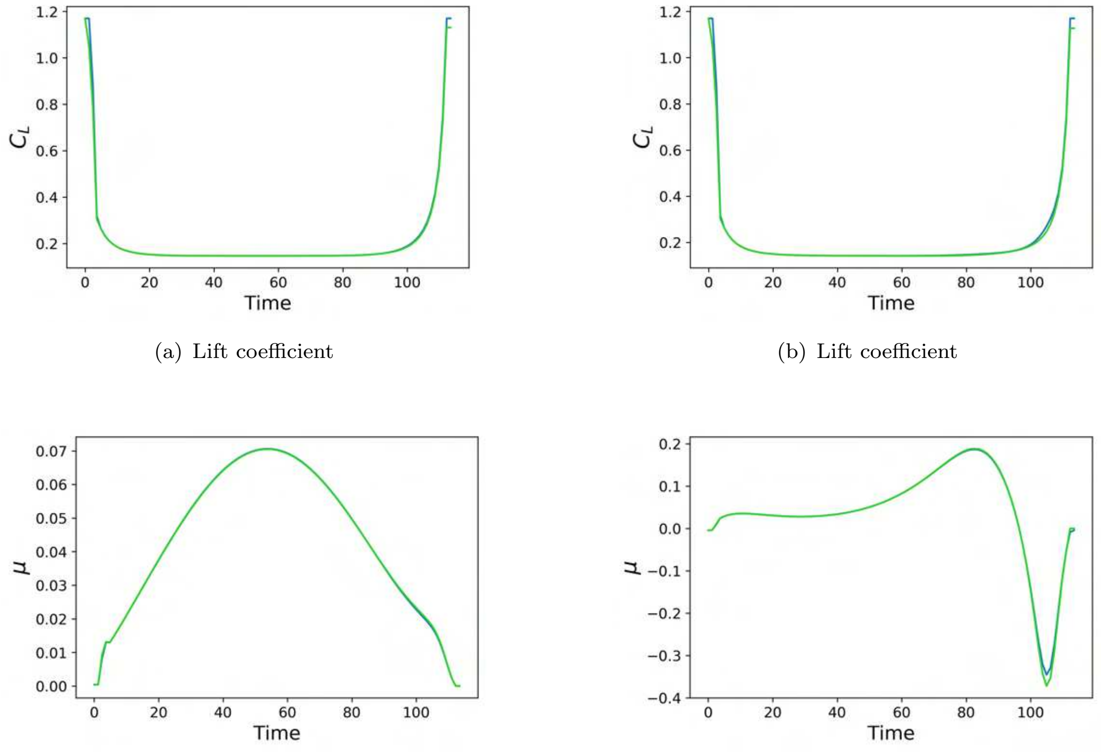
(c) Velocity

(d) Velocity

(e) Flight path angle

(f) Flight path angle

Figure 3.8: State variables profiles obtained by solving the linearised EoMs. Solutions attained in absence of wind (left) and in presence of wind (right) are compared to the ones found by solving the NLP.



(a) Lift coefficient

(b) Lift coefficient

(c) Bank angle

(d) Bank angle

FIGURE 3.9: Control variables profiles obtained by solving the linearised EoMs. Decisions made in absence of wind (left) and in presence of wind (right) are compared to the ones found by solving the NLP.

Chapter 4

The Glider Routing and Trajectory Optimisation Problem

The primary objective of this chapter is to present the Glider Routing and Trajectory optimisation Problem (GRTOP) and establish its mathematical formulations. Following initially Coutinho et al. (2019), we begin by describing the problem, introducing a Mixed-Integer Nonlinear Programming (MINLP) formulation, followed by a progression towards a Mixed-Integer Second Order Cone Programming (MISOCP) description, where dynamic constraints are linearised assuming prior knowledge of the initial trajectory or of well-defined equilibrium flight conditions. Subsequently, we present a novel Mixed-Integer Linear Programming (MILP) formulation obtained by changing or approximating the metric used to compute distances. At the end of the chapter, examples of glider's missions are also given, showing the possibility of improving computational times by means of a two level time partitioning. Glider routing and trajectory optimisation problems in cases where prior information on the initial trajectory or equilibrium conditions is absent are comprehensively treated through a detailed discussion in the subsequent chapter.

4.1 Problem Description

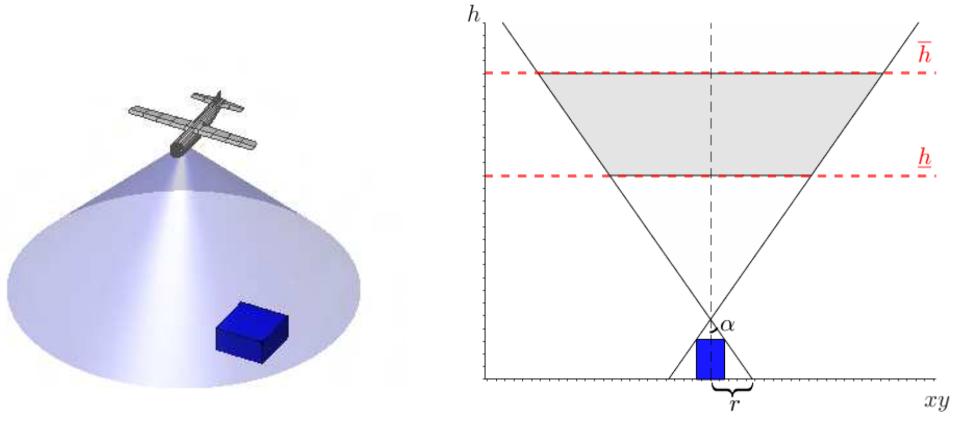
The problem under investigation involves the management of a fleet of balloon-lifted gliders, which are required to navigate the airspace while visiting specific points of interest, referred to as *waypoints*, capture images of objects located at these points and subsequently land safely at designated landing sites. In order to monitor and survey the objects in the locations of interest, each glider is equipped with a remote camera fixed to its body and able to take pictures within relative distant ranges. The fleet of gliders is assumed to be homogeneous and the cameras to have the same specifications. The primary objective of this mission is to optimise the mission duration, aiming at completing the required tasks in the shortest amount of time. At the same time, it is crucial to ensure that the flight dynamics constraints are respected throughout the entire mission. The problem just described is called the *the Glider Routing and Trajectory optimisation Problem* (GRTOP). To provide a precise and exhaustive mathematical description of the GRTOP, other than gliders' characteristics and dynamic described in the previous chapter, it is necessary to further describe its key components, including:

- balloon-lifting techniques;
- camera and photograph geometry;
- waypoints characterisation;
- landing sites description.

By carefully defining and elaborating on these key components, we can then establish a solid foundation for providing mathematical models, and consequently solutions algorithms, to the problem.

In the considered GRTOP, gliders can be balloon-lifted and launched from launching sites estimated using the tool "ASTRA High Altitude Balloon Flight Planner" proposed by Sobester et al. (2013) and available at Zapponi (2013). The provided tool enables users to simulate the trajectory of a balloon transporting a payload. Users have the flexibility to choose parameters such as payload weight and balloon type. Utilizing relevant weather forecasts, the ASTRA tool identifies optimal releasing points and generates potential flight trajectories for the balloon, allowing to take the gliders to an advantageous position relative to the waypoints and ensuring a more efficient planning aligned with the mission objectives. Then, a typical geometry describing the cameras' field of view is derived by adopting an inverted conic shape, where we consider its opening angle α equal to 45^o (0.7854 radians). This type of geometry, depicted in Figure 4.1 and adopted also in Ariyur and Fregene (2008), Roelofsen et al. (2016) and Nedjati et al. (2016) for UAV-camera systems, enables the definition of conical-section regions that the gliders must visit in order to reach the corresponding waypoints.

After defining the camera geometry, we can notice that each waypoint $i \in W$ can be entirely described by the tuple $(\bar{x}_i, \bar{y}_i, \bar{r}_i, \underline{h}_i, \bar{h}_i)$. Assuming that all the points of interest lie on the ground, $(\bar{x}_i, \bar{y}_i, 0)$ represents the position of the $i-th$ waypoint in the space, while $\bar{r}_i \geq 0$ represents the radius of a circle in the xy plane that encloses the footprint of the object to be photographed at i. At the same time, the parameters $\underline{h}_i$ and $\bar{h}_i$, define and constrain the quality of the pictures, representing respectively the minimum and maximum heights from which the object can be recorded. Given these assumptions, a glider g flying at an altitude h_g can be deemed to visit waypoint i and take a good quality picture of it, only if it enters the conic-like region above i at an altitude between



(a) (b)

Figure 4.1: Representation of gliders' camera field of view and its possibility of taking a good quality picture of an object on the ground. In (b), the grey conical-section from where the glider is able to take good quality pictures.

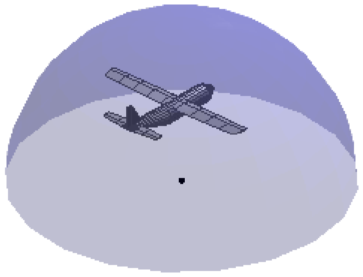


Figure 4.2: Representation of a landing zone geometry.

$\underline{h}_i$ and $\bar{h}_i$, see Figure 4.1(b). More precisely, the UAV g can visit waypoint i if its height is between the defined limits, $\underline{h}_i \leq h_g \leq \bar{h}_i$, and its position satisfies

$$(x_g - \bar{x}_i)^2 + (y_g - \bar{y}_i)^2 \leq (h_g tg(\alpha) - \bar{r}_i)^2 = (h_g - \bar{r}_i)^2,$$

where the last equality stems from the assumption that $\alpha = 45^o$. Analogously, we can completely determine the geometry of the landing zone $l \in L$ via the tuple $(\tilde{x}_l, \tilde{y}_l, \tilde{r}_l)$, with the shape of landing site that is supposed to be a half-sphere with center lying on the ground. Therefore, $(\tilde{x}_l, \tilde{y}_l, 0)$ defines the center's position on the xy plane, while $\tilde{r}_l$ represents the radius of the half-sphere enclosing the landing region, see Figure 4.2.

With this in mind, we are able to proceed and provide a first problem formulation for the GRTOP. In the following section, we introduce the definition of the variables required to write the GRTOP as a MINLP and we provide a mathematical description of the latter, also considering the non-linear dynamic constraint derived in the previous chapter utilising the Euler scheme to integrate the EoMs.

4.2 A MINLP Formulation

As introduced above, using the ASTRA tool we can assume a fleet of gliders, G, to be equipped with the remote cameras previously described and available at a known launching point and known initial state. Let then W be a given set of waypoints that have to be visited and L be a set of possible landing sites. The mission goal is then to find optimal routes and trajectories $\forall g \in G$ such that the whole set of waypoints is surveyed and the total mission time is minimised. To do so and simultaneously solve the (combinatorial) problem of routing while exploiting the previously described direct collocation method for optimising the trajectory, we start considering a time interval of interest, $[\tau_0, \tau_f]$. As noted in Remark 3.1, in order to derive a linear formulation of the problem, it is convenient to deal with constant time partitions. Therefore, we define a fixed time interval $[t_0, t_f]$ such that $[\tau_0, \tau_f] \subseteq [t_0, t_f]$. In particular, we can set t_0 to be equal to τ_0, which is fixed, whereas t_f is defined as the time required to travel along the most distant points in the space at a moderate speed. More precisely, if we consider B to be the smallest box containing all the point of interest (launching site, waypoints and landing site) and if we consider a space buffer enlarging B of about 5% in all the space directions, we can determine t_f as the time required to fly along the diagonal of B, d_B, at the preferred equilibrium velocity, v_{eq}, obtained by solving equations (3.23)-(3.30) while selecting the equilibrium lift coefficient by optimising the drag to lift ratio as in Remark 3.2. In other words,

$$t_f = \frac{d_B}{v_{eq}}. \tag{4.1}$$

Partitioning then the time interval $[t_0, t_f]$ into N smaller time steps, such that

$$t_0 = \tau_0 < \tau_1 < \cdots < \tau_N = t_f,$$

and indicating each time step index with $t \in T = \{0, \ldots, N\}$ corresponding to a time instant $\tau_t \in [t_0, t_f]$, it is then possible to determine the time in which a glider g is visiting a waypoint i or landing at landing site l by defining the following binary variables

$$a_{g,i,t} = \begin{cases} 1, & \text{if the glider } g \text{ is visiting waypoint } i \text{ at time step } t \\ 0, & \text{otherwise} \end{cases} \tag{4.2}$$

and

$$b_{g,l,t} = \begin{cases} 1, & \text{if the glider } g \text{ is in the landing site } l \text{ at time step } t \\ 0, & \text{otherwise} \end{cases}. \tag{4.3}$$

In this way, it is possible to determine the flight time for each glider g as the

$$\min_{t \in T}\{\tau_t \mid b_{g,l,t} = 1 \text{ for some } l \in L\},$$

and we can also determine a-posteriori the mission's end time, τ_f, as

$$\tau_f = \max_{g \in G}\left\{\min_{t \in T}\{\tau_t \mid b_{g,l,t} = 1 \text{ for some } l \in L\}\right\}. \tag{4.4}$$

With these definitions in mind, we can now proceed and formally write the objective function, which is the minimisation of the total flight time, described as the sum of gliders' flight times. Along the manuscript, we will use equally spaced time partitions such that each time step length is constant. Other partition techniques might be used, but they are out of this manuscript's scope and can the subject for future research investigations. Then, utilising such a uniform time partition, with time step length equal to

$$\eta = (\tau_{t+1} - \tau_t) = \frac{t_f - t_0}{N},$$

we require that the number of time steps t in which each glider g is airborne, i.e. $b_{g,l,t} = 0$ $\forall l$, is minimum. Given that each glider is balloon-launched and starts the mission already airborne, this is equivalent to require that the number of time step in which the glider is at a landing site is maximum, leading us to defining the objective function as

$$\min \quad -\sum_{g \in G, l \in L, t \in T \setminus \{0\}} \eta \; b_{g,l,t} = -\max \sum_{g \in G, l \in L, t \in T \setminus \{0\}} \eta \; b_{g,l,t}. \tag{4.5}$$

After defining the mission goal, we can start defining mission's constraints. First of all, we require that all the waypoints must be visited and this involves a series of constraints commencing with

$$\sum_{g \in G, t \in T} a_{g,i,t} \geq 1, \;\; \forall i \in W.$$

These indicate that at some point in time t a glider g is actually visiting waypoint i, i.e. $a_{g,i,t} = 1$. As such, the visiting glider g must be close enough to the waypoint and able to take a good quality picture of it. Therefore the distance of g from i on the xy plane at the time of visit, $d_{g,i,t}$, must be such that the cone describing the camera's field of view is including the object to be photographed. Then, a good quality picture can be only taken when g falls within the altitude range $[\underline{h}_i, \bar{h}_i]$ describing the waypoint. More precisely, we require that

$$\begin{aligned}
d^2_{g,i,t} &\geq (x_{g,t} - \bar{x}_i))^2 + (y_{g,t} - \bar{y}_i)^2, \ \forall g \in G, \forall i \in W, \forall t \in T \\
d_{g,i,t} &\leq (h_{g,t} - \bar{r}_i) + M(1 - a_{g,i,t}), \ \forall g \in G, \forall i \in W, \forall t \in T \\
h_{g,t} &\leq \bar{h}_i a_{g,i,t} + h_{ub}(1 - a_{g,i,t}), \ \forall g \in G, \forall i \in W, \forall t \in T \\
h_{g,t} &\geq \underline{h}_i a_{g,i,t} + h_{lb}(1 - a_{g,i,t}), \ \forall g \in G, \forall i \in W, \forall t \in T.
\end{aligned}$$

Here, a comment is required about the “big-M” term in the model definition. The constant M is a number big enough to make the logic implication

$$\begin{aligned}
a_{g,i,t} = 1 &\implies d_{g,i,t} \leq (h_{g,t} - \bar{r}_i) \\
a_{g,i,t} = 0 &\implies d_{g,i,t} \in \mathbb{R}_+
\end{aligned}$$

always valid and it has been computed as the space diagonal of the smallest cuboid containing the waypoints, landing sites and launching point, plus buffer. In other words, $M = d_B$. A similar role is then played by the lower and upper bounds h_{ub} and h_{lb}, being respectively the minimum and maximum flight altitude. In addition to these constraints, given that the camera is attached to the underside of the glider' fuselage or its *belly*, we require the UAV to fly "flat" with small bank, $\hat{\mu}$, and flight path, $\hat{\gamma}$, angles by writing

$$\begin{aligned}
\gamma_{g,t} &\leq \hat{\gamma} a_{g,i,t} + \gamma_{ub}(1 - a_{g,i,t}), \ \forall g \in G, \forall i \in W, \forall t \in T \\
\gamma_{g,t} &\geq -\hat{\gamma} a_{g,i,t} + \gamma_{lb}(1 - a_{g,i,t}), \ \forall g \in G, \forall i \in W, \forall t \in T \\
\mu_{g,t} &\leq \hat{\mu} a_{g,i,t} + \mu_{ub}(1 - a_{g,i,t}), \ \forall g \in G, \forall i \in W, \forall t \in T \\
\mu_{g,t} &\geq -\hat{\mu} a_{g,i,t} + \mu_{lb}(1 - a_{g,i,t}), \ \forall g \in G, \forall i \in W, \forall t \in T.
\end{aligned}$$

Ultimately, the glider needs to to avoid blurred pictures by flying at a moderate speed at the time of visit:

$$v_{g,t} \leq \hat{v} a_{g,i,t} + v_{ub}(1 - a_{g,i,t}), \ \forall g \in G, \forall i \in W, \forall t \in T.$$

In a similar manner, each glider g is then required to land before the final mission time t_f. In other words, we request that $b_{g,l,t} = 1$ for some time step $t \leq N$ and landing site l. Consequently, modelling again the logical implication through "big-M" constraints, at time of landing we ask the glider g to be within the half sphere determined by l:

$$\begin{aligned}
&\sum_{l \in L, t \in T} b_{g,l,t} \geq 1, \ \forall g \in G \\
&r^2_{g,l,t} \geq (x_{g,t} - \tilde{x}_l)^2 + (y_{g,t} - \tilde{y}_l)^2 + h^2_{g,t}, \ \forall g \in G, \forall l \in L, \forall t \in T \\
&r_{g,l,t} \leq \tilde{r}_l + M(1 - b_{g,l,t}), \ \forall g \in G, \forall l \in L, \forall t \in T,
\end{aligned}$$

with $r_{g,l,t}$ representing the distance of the glider from the landing site's center. Furthermore, we require that each glider lands only after its visit (if any) to the waypoints writing constraints

$$\sum_{\tilde{t} \leq t} b_{g,l,\tilde{t}} \leq N(1 - a_{g,i,t}), \ \forall g \in G, \forall l \in L, \forall i \in W, \forall t \in T.$$

Finally, after defining mission's objectives and targets, we proceed by considering UAVs' dynamic. Recalling that the motion of each glider is described by the system of ODEs

$$\dot{\mathbf{y}}_g(\tau) = \mathbf{f}(\mathbf{y}_g(\tau), \mathbf{u}_g(\tau), \mathbf{w}_g(\tau), \tau),$$

where $\mathbf{y}_g(\tau), \mathbf{u}_g(\tau), \mathbf{w}_g(\tau)$ represent respectively the six state variables, the control variables and external disturbances (i.e. the wind strength parameter β in our case) for glider g at time τ, and deploying the forward Euler integration scheme described in the previous section to solve the EoMs, it is possible to write dynamic constraints

$$\mathbf{y}_{g,t+1} = \mathbf{y}_{g,t} + \eta \mathbf{f}(\mathbf{y}_{g,t}, \mathbf{u}_{g,t}, \mathbf{w}_{g,t}), \ \forall g \in G, \forall t \in T \setminus \{N\} \tag{4.6}$$

$$\mathbf{y}_{g,0} = \mathbf{y}_0, \ \forall g \in G \tag{4.7}$$

$$\mathbf{y}_{lb} \leq \mathbf{y}_{g,t} \leq \mathbf{y}_{ub}, \ \forall g \in G, \forall t \in T \tag{4.8}$$

$$\mathbf{u}_{lb} \leq \mathbf{u}_{g,t} \leq \mathbf{u}_{ub}, \ \forall g \in G, \forall t \in T \tag{4.9}$$

determining, at each collocation point τ_t, the decisions and state of the glider during the flight, (4.6), indicating its initial conditions, (4.7), and defining its flight envelope bounds, (4.8)-(4.9). It is worth noting again that other integration schemes are also possible, leading to different equations describing constraints (4.6). We refer the interested reader to Coutinho et al. (2019) where a detailed analyses and comparison of the different methods is discussed in the context of the GRTOP. Within our formulation, it is also possible to contemplate an extended set of possible solutions, by enlarging the trajectories satisfying dynamic constraints (4.6). More precisely, we admit some deviations from the original equations penalising deviation from them in the objective. Consequently, instead of considering constrains (4.6), we can write the following inequalities

$$\mathbf{y}_{g,t+1} \leq \mathbf{y}_{g,t} + \eta \mathbf{f}(\mathbf{y}_{g,t}, \mathbf{u}_{g,t}, \mathbf{w}_{g,t}) + \boldsymbol{\varepsilon}_{g,t}, \ \forall g \in G, \forall t \in T \setminus \{N\} \tag{4.10}$$

$$\mathbf{y}_{g,t+1} \geq \mathbf{y}_{g,t} + \eta \mathbf{f}(\mathbf{y}_{g,t}, \mathbf{u}_{g,t}, \mathbf{w}_{g,t}) - \boldsymbol{\varepsilon}_{g,t}, \ \forall g \in G, \forall t \in T \setminus \{N\}, \tag{4.11}$$

where $\boldsymbol{\varepsilon}_{g,t} = (\varepsilon_x, \varepsilon_y, \varepsilon_h, \varepsilon_v, \varepsilon_\gamma, \varepsilon_\varphi)_{g,t} \geq \mathbf{0}$ represents the deviation from the equation describing the dynamic of g at time step t.Thus, as just mentioned, we consider inserting these terms $\boldsymbol{\varepsilon}_{g,t}$ in the objective function with the intention of minimising such divergences. Indicating the set of state variables with $S = \{x, y, h, v, \gamma, \varphi\}$, we can then write a new objective as

$$\begin{aligned}\min \quad & - w_1 * \sum_{g\in G, l\in L, t\in T\setminus\{0\}} \eta \; b_{g,l,t} \\ & + w_2 \sum_{s\in S, g\in G, t\in T} \eta \; \frac{\varepsilon_{s,g,t}}{\kappa_s} \\ & + w_3 \sum_{s\in S, g\in G} t_f \; \frac{\epsilon_{s,g}}{\kappa_s}\end{aligned} \tag{4.12}$$

where we specify convenient weights for the objectives, w_1, w_2, w_3 (see also Miettinen (1999) for weights role in multiobjective optimisation problems), and where we consider a further penalisation term to minimise the maximum deviation from the dynamic constraints (4.6) by defining variables $\boldsymbol{\epsilon_g} = (\epsilon_x, \epsilon_y, \epsilon_h, \epsilon_v, \epsilon_\gamma, \epsilon_\varphi)_g \geq \mathbf{0}$ and inserting constraints

$$\boldsymbol{\varepsilon_{g,t}} \leq \boldsymbol{\epsilon_g}, \;\; \forall g \in G, \forall t \in T. \tag{4.13}$$

Each deviation term is then appropriately scaled via the factors κ_s depending on the state variable's magnitude. With this in mind, a complete mixed-integer nonlinear programming (MINLP) formulation of the problem can then be derived as in (4.14)-(4.39).

$$\begin{aligned}\min \quad & - w_1 * \sum_{g\in G, l\in L, t\in T\setminus\{0\}} \eta \; b_{g,l,t} \\ & + w_2 \sum_{s\in S, g\in G, t\in T} \eta \; \frac{\varepsilon_{s,g,t}}{\kappa_s} \\ & + w_3 \sum_{s\in S, g\in G} t_f \; \frac{\epsilon_{s,g}}{\kappa_s}\end{aligned} \tag{4.14}$$

$$\text{subject to} \quad \sum_{g\leq i, t\in T} a_{g,i,t} \geq 1, \;\; \forall i \in W \tag{4.15}$$

$$d^2_{g,i,t} \geq (x_{g,t} - \bar{x}_i))^2 + (y_{g,t} - \bar{y}_i)^2, \;\; \forall g \in G, \forall i \in W, \forall t \in T \tag{4.16}$$

$$d_{g,i,t} \leq (h_{g,t} - \bar{r}_i) + M(1 - a_{g,i,t}), \;\; \forall g \in G, \forall i \in W, \forall t \in T \tag{4.17}$$

$$h_{g,t} \leq \bar{h}_i a_{g,i,t} + h_{ub}(1 - a_{g,i,t}), \;\; \forall g \in G, \forall i \in W, \forall t \in T \tag{4.18}$$

$$h_{g,t} \geq \underline{h}_i a_{g,i,t} + h_{lb}(1 - a_{g,i,t}), \;\; \forall g \in G, \forall i \in W, \forall t \in T \tag{4.19}$$

$$\gamma_{g,t} \leq \hat{\gamma} a_{g,i,t} + \gamma_{ub}(1 - a_{g,i,t}), \;\; \forall g \in G, \forall i \in W, \forall t \in T \tag{4.20}$$

$$\gamma_{g,t} \geq -\hat{\gamma} a_{g,i,t} + \gamma_{lb}(1 - a_{g,i,t}), \;\; \forall g \in G, \forall i \in W, \forall t \in T \tag{4.21}$$

$$\mu_{g,t} \leq \hat{\mu} a_{g,i,t} + \mu_{ub}(1 - a_{g,i,t}), \;\; \forall g \in G, \forall i \in W, \forall t \in T \tag{4.22}$$

$$\mu_{g,t} \geq -\hat{\mu} a_{g,i,t} + \mu_{lb}(1 - a_{g,i,t}), \;\; \forall g \in G, \forall i \in W, \forall t \in T \tag{4.23}$$

$$v_{g,t} \leq \hat{v} a_{g,i,t} + v_{ub}(1 - a_{g,i,t}), \;\; \forall g \in G, \forall i \in W, \forall t \in T \tag{4.24}$$

$$\sum_{l\in L, t\in T} b_{g,l,t} \geq 1, \ \forall g \in G \tag{4.25}$$

$$r^2_{g,l,t} \geq (x_{g,t} - \tilde{x}_l)^2 + (y_{g,t} - \tilde{y}_l)^2 + h^2_{g,t}, \ \forall g \in G, \forall l \in L, \forall t \in T \tag{4.26}$$

$$r_{g,l,t} \leq \tilde{r}_l + M(1 - b_{g,l,t}), \ \forall g \in G, \forall l \in L, \forall t \in T \tag{4.27}$$

$$\sum_{\tilde{t} \leq t} b_{g,l,\tilde{t}} \leq N(1 - a_{g,i,t}), \ \forall g \in G, \forall l \in L, \forall i \in W, \forall t \in T \tag{4.28}$$

$$\mathbf{y}_{g,t+1} \leq \mathbf{y}_{g,t} + \eta \mathbf{f}(\mathbf{y}_{g,t}, \mathbf{u}_{g,t}, \mathbf{w}_{g,t}) + \boldsymbol{\varepsilon}_{g,t}, \ \forall g \in G, \forall t \in T \setminus \{N\} \tag{4.29}$$

$$\mathbf{y}_{g,t+1} \geq \mathbf{y}_{g,t} + \eta \mathbf{f}(\mathbf{y}_{g,t}, \mathbf{u}_{g,t}, \mathbf{w}_{g,t}) - \boldsymbol{\varepsilon}_{g,t}, \ \forall g \in G, \forall t \in T \setminus \{N\} \tag{4.30}$$

$$\mathbf{y}_{g,0} = \mathbf{y}_0, \ \forall g \in G \tag{4.31}$$

$$\mathbf{y}_{lb} \leq \mathbf{y}_{g,t} \leq \mathbf{y}_{ub}, \ \forall g \in G, \forall t \in T \tag{4.32}$$

$$\mathbf{u}_{lb} \leq \mathbf{u}_{g,t} \leq \mathbf{u}_{ub}, \ \forall g \in G, \forall t \in T \tag{4.33}$$

$$\boldsymbol{\varepsilon_{g,t}} \leq \boldsymbol{\epsilon_g}, \ \forall g \in G, \forall t \in T \tag{4.34}$$

$$a_{g,i,t} \in \{0,1\}, \ \forall g \in G, \forall i \in W, \forall t \in T \tag{4.35}$$

$$b_{g,l,t} \in \{0,1\}, \ \forall g \in G, \forall l \in L, \forall t \in T \tag{4.36}$$

$$d_{g,i,t}, r_{g,i,t} \in \mathbb{R}_+, \ \forall g \in G, \forall i \in W \cup L, \forall t \in T \tag{4.37}$$

$$\mathbf{y}_{g,t} \in \mathbb{R}^6, \mathbf{u}_{g,t} \in \mathbb{R}^2, \ \forall g \in G, \forall t \in T \tag{4.38}$$

$$\boldsymbol{\varepsilon}_{g,t}, \boldsymbol{\epsilon}_g \in \mathbb{R}^6_+, \ \forall g \in G, \forall t \in T, \tag{4.39}$$

where, as previously described, objective (4.14) requires to minimise the time of flight as well as the scaled deviations from the linearised dynamic. It is worth mentioning here that, although a similar goal as in Coutinho et al. (2019) is presented, the different formulation of the objective as the maximisation of landing time, combined with constraint (4.25), the separation of penalty terms by state as well as their scaling allow solvers for a more reliable selection of the trajectory and provide more meaning to the dynamic after landing. Then, constraints (4.15)-(4.24) ensure the correct visit of the waypoints, also respecting camera's and glider dynamic limitations, while constraints (4.25)-(4.28) ensure a correct glider landing. Finally, constraints (4.29)-(4.34) describe gliders' dynamic restrictions throughout the flight, whereas the rest of the constraints define variables domains.

With this in mind, if on one side a MINLP formulation is extremely useful to accurately describe the glider routing and trajectory optimisation problem, on the other hand finding solutions to such problem formulation can be computationally expensive and, in the context of dealing with UAVs' dynamic constraints, can carry all the disadvantages highlighted in the previous chapter. Therefore, in the following sections, we devote our efforts in deriving approximation of nonlinear constraints in order to overcome such difficulties and to boost computational efficiency.

4.3 Approximation Techniques for a MILP Formulation

In the previous section, we derive a nonlinear model formulation for the glider routing and trajectory optimisation problem. However, this model might be difficult to handle and carries in it all the difficulties stemming from the nonlinearities of a trajectory optimisation problem, such as instability of the optimal solution, high dependence on initial conditions and high computational times. When some prior knowledge on an initial trajectory, equilibria or (quasi) steady-state flight conditions is available, it is always advisable to follow the techniques proposed in the previous chapter and reach a linearised formulation of the glider dynamic. Doing so, leveraging on the approximation $\tilde{\mathbf{f}}$ of $\mathbf{f}$, it is possible to replace constraints (4.10) and (4.11) with the linear ones

$$\mathbf{y}_{g,t+1} \leq \mathbf{y}_{g,t} + \eta\tilde{\mathbf{f}}(\mathbf{y}_{g,t}, \mathbf{u}_{g,t}, \mathbf{w}_{g,t}) + \boldsymbol{\varepsilon}_{g,t},\ \ \forall g \in G, \forall t \in T \setminus \{N\} \tag{4.40}$$

$$\mathbf{y}_{g,t+1} \geq \mathbf{y}_{g,t} + \eta\tilde{\mathbf{f}}(\mathbf{y}_{g,t}, \mathbf{u}_{g,t}, \mathbf{w}_{g,t}) - \boldsymbol{\varepsilon}_{g,t},\ \ \forall g \in G, \forall t \in T \setminus \{N\}. \tag{4.41}$$

In addition to this, to increase the reliability of the flight dynamic and ensure the representativeness of the dynamic constraints, also overcoming the local nature of Taylor's approximation, we decide to require the glider to fly as close as possible to the initial trajectory or equilibrium points. More precisely, we define further variables $\boldsymbol{\lambda}_{g,t} = (\lambda_x, \lambda_y, \lambda_h, \lambda_v, \lambda_\gamma, \lambda_\varphi)_{g,t} \geq \mathbf{0}$, $\boldsymbol{\Lambda}_g = (\Lambda_x, \Lambda_y, \Lambda_h, \Lambda_v, \Lambda_\gamma, \Lambda_\varphi)_g \geq \mathbf{0}$ as well as $\boldsymbol{\Lambda}^u_g = (\Lambda^u_{C_L}, \Lambda^u_\mu)_g \geq \mathbf{0}$, helping in penalising such deviations from the initial nominal trajectories $(\mathbf{y}^o_{g,t}, \mathbf{u}^o_{g,t})$. We then insert the following penalties in the objective function

$$w_4 \sum_{s\in S, g\in G, t\in T} \eta\ \frac{\lambda_{s,g,t}}{\kappa_s} \tag{4.42}$$

$$w_5 \sum_{s\in S, g\in G} t_f\ \frac{\Lambda_{s,g}}{\kappa_s} \tag{4.43}$$

$$w_6 \sum_{c\in C, g\in G} t_f\ \Lambda^u_{c,g}, \tag{4.44}$$

with $C = \{C_L, \mu\}$ indicating the set of control variables, as well as constraints

$$(\mathbf{y}_{g,t} - \mathbf{y}^o_{g,t}) \leq \boldsymbol{\lambda_{g,t}},\ \ \forall g \in G, \forall t \in T \tag{4.45}$$

$$(\mathbf{y}^o_{g,t} - \mathbf{y}_{g,t}) \leq \boldsymbol{\lambda_{g,t}},\ \ \forall g \in G, \forall t \in T \tag{4.46}$$

$$\boldsymbol{\lambda_{g,t}} \leq \boldsymbol{\Lambda_g},\ \ \forall g \in G, \forall t \in T \tag{4.47}$$

$$(\mathbf{u}_{g,t} - \mathbf{u}^o_{g,t}) \leq \boldsymbol{\Lambda^u_g},\ \ \forall g \in G, \forall t \in T \tag{4.48}$$

$$(\mathbf{u}^o_{g,t} - \mathbf{u}_{g,t}) \leq \boldsymbol{\Lambda^u_g},\ \ \forall g \in G, \forall t \in T. \tag{4.49}$$

In this way, by transforming the equations of motion into their linearised form, and inserting proper penalties to ensure model reliability, we reach a formulation of the

GRTOP as a mixed-integer second-order cone programming (MISOCP), which help us handling the problem with linear and convex quadratic constraints efficiently. However, while the current formulation can already serve our purpose of solving the GRTOP with prior information, in order to further boost efficiency and computational speed and render the problem more scalable, we aim at enhancing our approach by transitioning to a completely linear formulation, improving even further Coutinho et al. (2019) approach to the GRTOP with prior information.

4.3.1 Linear Approximation for Distance Constraints

After a closer examination, we can identify that the solely remaining source of nonlinearity emerges from Euclidean norms and distance constraints (4.16)-(4.17) and (4.26)-(4.27). Thus, to render the optimisation problem more menageable and exploit the potentialities of the state-of-the-art optimisation algorithms, we decide to linearise distance constraints by deploying a different norm that can be indeed easily linearised. More precisely, we can define a p-ball of radius r and centered in $\mathbf{x}_0$ as

$$B_p(\mathbf{x}_0, r) = \{x \in \mathbb{R}^d \mid \|\mathbf{x} - \mathbf{x}_0\|_p \leq r\},$$

where $\|\mathbf{x}\|_p = \sqrt[p]{\sum_{i=1}^{d} |x_i|^p}$ is the L^p norm in $\mathbb{R}^d$ for $1 \leq p < \infty$, while $\|\mathbf{x}\|_\infty = \max_{i=1,\dots,d} |x_i|$ represents the L^∞ norm when $p = \infty$. An illustration of the unitary balls ($r = 1$) for $p = 1, 2$ and ∞ is depicted in Figure 4.3.

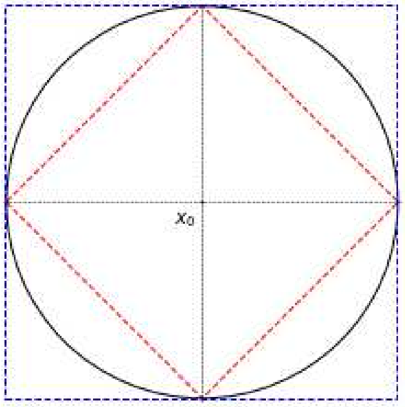


FIGURE 4.3: Unitary balls defined using different norms. In black we see the Euclidean ball, $B_2(\mathbf{x}_0, r)$, in red we depict the unitary ball determined by the L^1 norm, $B_1(\mathbf{x}_0, r)$, while the ball stemming from the L^∞ norm, $B_\infty(\mathbf{x}_0, r)$, is represented in blue.

It is then possible to observe that $B_1(\mathbf{x}_0, r)$ can be described by $2d+1$ linear inequalities (actually $2d$ given that it is the intersection of $2d$ half-spaces) and the simplest approach to do so is to introduce new variables $y_i \in \mathbb{R}_+$, $i = 1, \dots, d$ and write the following system of inequalities

$$\begin{aligned} &x_{0,i} - x_i \leq y_i \\ &x_i - x_{0,i} \leq y_i \\ &\sum_{i=1}^{d} y_i \leq r. \end{aligned}$$

We can then rewrite inequalities (4.16)-(4.17) as

$$d^x_{g,i,t} \geq x_{g,t} - \bar{x}_i, \;\; \forall g \in G, \forall i \in W, \forall t \in T \tag{4.50}$$
$$d^x_{g,i,t} \geq \bar{x}_i - x_{g,t}, \;\; \forall g \in G, \forall i \in W, \forall t \in T \tag{4.51}$$
$$d^y_{g,i,t} \geq y_{g,t} - \bar{y}_i, \;\; \forall g \in G, \forall i \in W, \forall t \in T \tag{4.52}$$
$$d^y_{g,i,t} \geq \bar{y}_i - y_{g,t}, \;\; \forall g \in G, \forall i \in W, \forall t \in T \tag{4.53}$$
$$\sum_{j \in \{x,y\}} d^j_{g,i,t} \leq (h_{g,t} - \bar{r}_i) + M(1 - a_{g,i,t}), \;\; \forall g \in G, \forall i \in W, \forall t \in T, \tag{4.54}$$

and obtain a linear formulation of the constraints. Noting that $B_1(\mathbf{x}_0, r) \subseteq B_2(\mathbf{x}_0, r)$, we can state that these inequalities produce a set of more conservative constraints, meaning that if (4.50)-(4.54) are valid inequalities for the current solution, then (4.16)-(4.17) are also valid inequalities for it, whereas the converse is not necessarily true. Analogously, we can rewrite constraints (4.26)-(4.27) as

$$r^x_{g,i,t} \geq x_{g,t} - \tilde{x}_l, \;\; \forall g \in G, \forall l \in L, \forall t \in T \tag{4.55}$$
$$r^x_{g,i,t} \geq \tilde{x}_l - x_{g,t}, \;\; \forall g \in G, \forall l \in L, \forall t \in T \tag{4.56}$$
$$r^y_{g,i,t} \geq y_{g,t} - \tilde{y}_l, \;\; \forall g \in G, \forall l \in L, \forall t \in T \tag{4.57}$$
$$r^y_{g,i,t} \geq \tilde{y}_l - y_{g,t}, \;\; \forall g \in G, \forall l \in L, \forall t \in T \tag{4.58}$$
$$r^h_{g,i,t} \geq h_{g,t}, \;\; \forall g \in G, \forall l \in L, \forall t \in T \tag{4.59}$$
$$r^h_{g,i,t} \geq -h_{g,t}, \;\; \forall g \in G, \forall l \in L, \forall t \in T \tag{4.60}$$
$$\sum_{j \in \{x,y,h\}} r^j_{g,l,t} \leq \tilde{r}_l + M(1 - b_{g,l,t}), \;\; \forall g \in G, \forall l \in L, \forall t \in T, \tag{4.61}$$

reaching a linear and more conservative version of the GRTOP.

As an alternative and less conservative approach, it is possible to leverage on an approximation of the Euclidean norm proposed by Rhodes (1995). In particular, the L^2 norm can be approximated by a convex combination of the L^1 norm and L^∞ norm by writing

$$\|\mathbf{x}\|_2 \simeq \lambda_d \|\mathbf{x}\|_1 + (1 - \lambda_d)\|\mathbf{x}\|_\infty \tag{4.62}$$

for $d = 2, 3$, $\mathbf{x} \in \mathbb{R}^d$ and constants $\lambda_2 = 0.3364$ or $\lambda_3 = 0.2980$. A comparison between the Euclidean norm and its just presented approximation in two dimensions is shown in Figure 4.4.

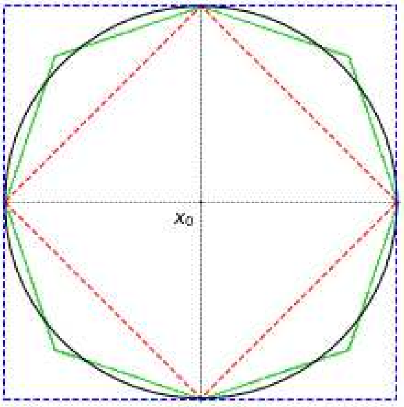


Figure 4.4: Euclidean ball approximation. The Euclidean ball is depicted in black, $B_2(\mathbf{x}_0, r)$, in red the unitary ball determined by the L^1 norm, $B_1(\mathbf{x}_0, r)$, while the ball stemming from the L^∞ norm, $B_\infty(\mathbf{x}_0, r)$, is represented in blue. Ultimately, in green is shown the convex approximation of the Euclidean norm determined by equation (4.62).

To determine the corresponding linear constraints, similarly to the above, we can define auxiliary variables y_i, $i = 1, \ldots, d$ such that

$$\begin{aligned} x_{0,i} - x_i &\leq y_i \\ x_i - x_{0,i} &\leq y_i. \end{aligned}$$

Then, recalling that $\|\mathbf{x}\|_\infty = \max_{i=1,\ldots,d} |x_i|$, we can define $\bar{y}$ such that

$$y_i \leq \bar{y}, \ \forall i = 1, \ldots, d,$$

and we can consider the Euclidean norm approximation $\xi_d(\mathbf{x})$ as

$$\xi_d(\mathbf{x}) = \lambda_d \sum_{i=1}^{d} y_i + (1 - \lambda_d)\bar{y}, \text{ for } d = 2, 3.$$

Consequently, $B_2(\mathbf{x}_0, r)$ can be approximated as

$$B_2(\mathbf{x}_0, r) = \{x \in \mathbb{R}^d \mid \xi_d(\mathbf{x}) \leq r\},$$

where $\xi_d(\mathbf{x}) \leq r$ is now a linear inequality.

It is worth mentioning here that despite we decide not to use such an approximation at this stage, and to deploy a more conservative description of distance constraints, equation (4.62) approximating the Euclidean norm will be extremely useful in the following chapter, helping us to find valuable initial trajectory guesses without the need of solving the nonlinear glider's dynamic.

4.4 A Two Time Level Approach to Solve the MILP

Although we have already achieved a linear formulation for the GRTOP with prior information, in our continuous pursuit of efficiency and improvement of computational times, we enhance the model further by reducing the number of binary variables and defining a two time step level approach following Schmidt and Fügenschuh (2023). This approach suggests dividing the optimisation procedure into two distinct time partitions. Each time partition may be used to address specific aspects of the problem, allowing for a more organised and efficient computation, while a mapping between the two partition is then defined to unify the problem requirements. More precisely, considering the time interval $[t_0, t_f]$ we can define the two partition as follows. Let N denote the number of time steps of a first uniform partition and η the length of each time step, such that $[t_0, t_f] = [t_0, \eta N]$. Then, we can define a second refined partition by adding n_f points in between two adjacent time steps $t_i, t_j \in T = \{0, \dots, N\}$. To easily define the mapping between the two time partitions it might be useful to redefine the index set for them. In particular, $T_c = \{0, n_f+1, \dots, N(n_f+1)\}$ is the set of indices corresponding to the first and coarse time partition, whereas $T_f = \{0, 1, \dots, N(n_f+1)\} \supseteq T_c$ is relative to the finer time partition. It is also convenient to remember that $|T_c| = |T|$ does not depend on n_f and therefore the time step length of the first partition is equal to η, whereas the length of the time step relative to the finer time partition is given by $\eta_f = \frac{\eta}{n_f+1}$. In Figure 4.5, we can see a succinct illustration of the two time level partition just described.

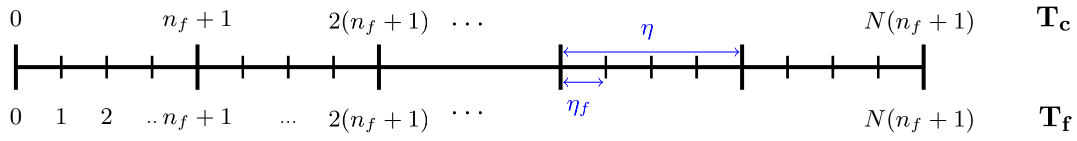


Figure 4.5: Illustration of the two time level partition.

In light of this, the idea is then to consider the continuous variables of the problem as defined on the finer time partition, allowing a better precision of the computed trajectories by the control of n_f, while the binary variables are only defined on the coarse one, so that their number is limited and hence the computational speed is enhanced. To describe constraints established on the finer time partition T_f and including both continuous and binary variables in the model, a function between the two partition partition indices can be introduced. More precisely, we define the function

$$\begin{aligned} \pi \colon T_f &\longrightarrow T_c \\ t &\longmapsto \left\lfloor \frac{t}{n_f + 1} \right\rfloor (n_f + 1) \end{aligned}$$

to map the time indices belonging to the finest partition to indices belonging to the coarse counterpart, always rounding down. It is worth noting here that the just-described approach will be used to boost the efficiency of all the models and algorithms presented further in the thesis. With all this in mind, it is then possible to rewrite the full model for the GRTOP when prior information is available as

Model I - GRTOP with Prior Information

$$\begin{aligned} \min \ & - w_1 * \sum_{g \in G, l \in L, t \in T_c \setminus \{0\}} \eta \ b_{g,l,t} \\ & + w_2 \sum_{s \in S, g \in G, t \in T_f} \eta_f \ \frac{\varepsilon_{s,g,t}}{\kappa_s} \\ & + w_3 \sum_{s \in S, g \in G} t_f \ \frac{\epsilon_{s,g}}{\kappa_s} \\ & + w_4 \sum_{s \in S, g \in G, t \in T_f} \eta_f \ \frac{\lambda_{s,g,t}}{\kappa_s} \\ & + w_5 \sum_{s \in S, g \in G} t_f \ \frac{\Lambda_{s,g}}{\kappa_s} \\ & + w_6 \sum_{c \in C, g \in G} t_f \ \Lambda^u_{c,g} \end{aligned} \tag{4.63}$$

$$\text{subject to} \sum_{g \leq i, t \in T_c} a_{g,i,t} \geq 1, \ \forall i \in W \tag{4.64}$$

$$d^x_{g,i,t} \geq x_{g,t} - \bar{x}_i, \ \forall g \in G, \forall i \in W, \forall t \in T_c \tag{4.65}$$

$$d^x_{g,i,t} \geq \bar{x}_i - x_{g,t}, \ \forall g \in G, \forall i \in W, \forall t \in T_c \tag{4.66}$$

$$d^y_{g,i,t} \geq y_{g,t} - \bar{y}_i, \ \forall g \in G, \forall i \in W, \forall t \in T_c \tag{4.67}$$

$$d^y_{g,i,t} \geq \bar{y}_i - y_{g,t}, \ \forall g \in G, \forall i \in W, \forall t \in T_c \tag{4.68}$$

$$\sum_{j \in \{x,y\}} d^j_{g,i,t} \leq (h_{g,t} - \bar{r}_i) + M(1 - a_{g,i,t}), \ \forall g \in G, \forall i \in W, \forall t \in T_c \tag{4.69}$$

$$h_{g,t} \leq \bar{h}_i a_{g,i,t} + h_{ub}(1 - a_{g,i,t}),\ \forall g \in G, \forall i \in W, \forall t \in T_c \tag{4.70}$$

$$h_{g,t} \geq \underline{h}_i a_{g,i,t} + h_{lb}(1 - a_{g,i,t}),\ \forall g \in G, \forall i \in W, \forall t \in T_c \tag{4.71}$$

$$\gamma_{g,t} \leq \hat{\gamma} a_{g,i,t} + \gamma_{ub}(1 - a_{g,i,t}),\ \forall g \in G, \forall i \in W, \forall t \in T_c \tag{4.72}$$

$$\gamma_{g,t} \geq -\hat{\gamma} a_{g,i,t} + \gamma_{lb}(1 - a_{g,i,t}),\ \forall g \in G, \forall i \in W, \forall t \in T_c \tag{4.73}$$

$$\mu_{g,t} \leq \hat{\mu} a_{g,i,t} + \mu_{ub}(1 - a_{g,i,t}),\ \forall g \in G, \forall i \in W, \forall t \in T_c \tag{4.74}$$

$$\mu_{g,t} \geq -\hat{\mu} a_{g,i,t} + \mu_{lb}(1 - a_{g,i,t}),\ \forall g \in G, \forall i \in W, \forall t \in T_c \tag{4.75}$$

$$v_{g,t} \leq \hat{v} a_{g,i,t} + v_{ub}(1 - a_{g,i,t}),\ \forall g \in G, \forall i \in W, \forall t \in T_c \tag{4.76}$$

$$\sum_{l \in L, t \in T_c} b_{g,l,t} \geq 1,\ \forall g \in G \tag{4.77}$$

$$r^x_{g,i,t} \geq x_{g,t} - \tilde{x}_l,\ \forall g \in G, \forall l \in L, \forall t \in T_c \tag{4.78}$$

$$r^x_{g,i,t} \geq \tilde{x}_l - x_{g,t},\ \forall g \in G, \forall l \in L, \forall t \in T_c \tag{4.79}$$

$$r^y_{g,i,t} \geq y_{g,t} - \tilde{y}_l,\ \forall g \in G, \forall l \in L, \forall t \in T_c \tag{4.80}$$

$$r^y_{g,i,t} \geq \tilde{y}_l - y_{g,t},\ \forall g \in G, \forall l \in L, \forall t \in T_c \tag{4.81}$$

$$r^h_{g,i,t} \geq h_{g,t},\ \forall g \in G, \forall l \in L, \forall t \in T_c \tag{4.82}$$

$$r^h_{g,i,t} \geq -h_{g,t},\ \forall g \in G, \forall l \in L, \forall t \in T_c \tag{4.83}$$

$$\sum_{j \in \{x,y,h\}} r^j_{g,l,t} \leq \tilde{r}_l + M(1 - b_{g,l,t}),\ \forall g \in G, \forall l \in L, \forall t \in T_c \tag{4.84}$$

$$\sum_{\tilde{t} \leq t} b_{g,l,\tilde{t}} \leq N(1 - a_{g,i,t}),\ \forall g \in G, \forall l \in L, \forall i \in W, \forall t \in T_c \tag{4.85}$$

$$\mathbf{y}_{g,t+1} \leq \mathbf{y}_{g,t} + \eta_f \tilde{\mathbf{f}}(\mathbf{y}_{g,t}, \mathbf{u}_{g,t}, \mathbf{w}_{g,t}) + \boldsymbol{\varepsilon}_{g,t},\ \forall g \in G, \forall t \in T_f \setminus \{N(n_f + 1)\} \tag{4.86}$$

$$\mathbf{y}_{g,t+1} \geq \mathbf{y}_{g,t} + \eta_f \tilde{\mathbf{f}}(\mathbf{y}_{g,t}, \mathbf{u}_{g,t}, \mathbf{w}_{g,t}) - \boldsymbol{\varepsilon}_{g,t},\ \forall g \in G, \forall t \in T_f \setminus \{N(n_f + 1)\} \tag{4.87}$$

$$\mathbf{y}_{g,0} = \mathbf{y}_0,\ \forall g \in G \tag{4.88}$$

$$\mathbf{y}_{lb} \leq \mathbf{y}_{g,t} \leq \mathbf{y}_{ub},\ \forall g \in G, \forall t \in T_f \tag{4.89}$$

$$\mathbf{u}_{lb} \leq \mathbf{u}_{g,t} \leq \mathbf{u}_{ub},\ \forall g \in G, \forall t \in T_f \tag{4.90}$$

$$\boldsymbol{\varepsilon_{g,t}} \leq \boldsymbol{\epsilon_g},\ \forall g \in G, \forall t \in T_f \tag{4.91}$$

$$(\mathbf{y}_{g,t} - \mathbf{y}^o_{g,t}) \leq \boldsymbol{\lambda_{g,t}},\ \forall g \in G, \forall t \in T_f \tag{4.92}$$

$$(\mathbf{y}^o_{g,t} - \mathbf{y}_{g,t}) \leq \boldsymbol{\lambda_{g,t}},\ \forall g \in G, \forall t \in T_f \tag{4.93}$$

$$\boldsymbol{\lambda_{g,t}} \leq \boldsymbol{\Lambda_g},\ \forall g \in G, \forall t \in T_f \tag{4.94}$$

$$(\mathbf{u}_{g,t} - \mathbf{u}^o_{g,t}) \leq \boldsymbol{\Lambda^u_g},\ \forall g \in G, \forall t \in T_f \tag{4.95}$$

$$(\mathbf{u}^o_{g,t} - \mathbf{u}_{g,t}) \leq \boldsymbol{\Lambda^u_g},\ \forall g \in G, \forall t \in T_f \tag{4.96}$$

$$a_{g,i,t} \in \{0,1\},\ \forall g \in G, \forall i \in W, \forall t \in T_c \tag{4.97}$$

$$b_{g,l,t} \in \{0,1\},\ \forall g \in G, \forall l \in L, \forall t \in T_c \tag{4.98}$$

$$d^j_{g,i,t} \in \mathbb{R}_+,\ \forall g \in G, \forall i \in W \cup L, \forall j \in \{x,y\}, \forall t \in T_c \tag{4.99}$$

$$r^j_{g,i,t} \in \mathbb{R}_+,\ \forall g \in G, \forall i \in W \cup L, \forall j \in \{x,y,h\}, \forall t \in T_c \tag{4.100}$$

$$\mathbf{y}_{g,t} \in \mathbb{R}^6, \mathbf{u}_{g,t} \in \mathbb{R}^2,\ \forall g \in G, \forall t \in T_f \tag{4.101}$$

$$\boldsymbol{\varepsilon}_{g,t}, \boldsymbol{\epsilon}_g, \boldsymbol{\lambda}_{g,t}, \boldsymbol{\Lambda}_g \in \mathbb{R}^6_+, \boldsymbol{\Lambda}^{\boldsymbol{u}}{}_g \in \mathbb{R}^2_+,\ \forall g \in G, \forall t \in T_f. \tag{4.102}$$

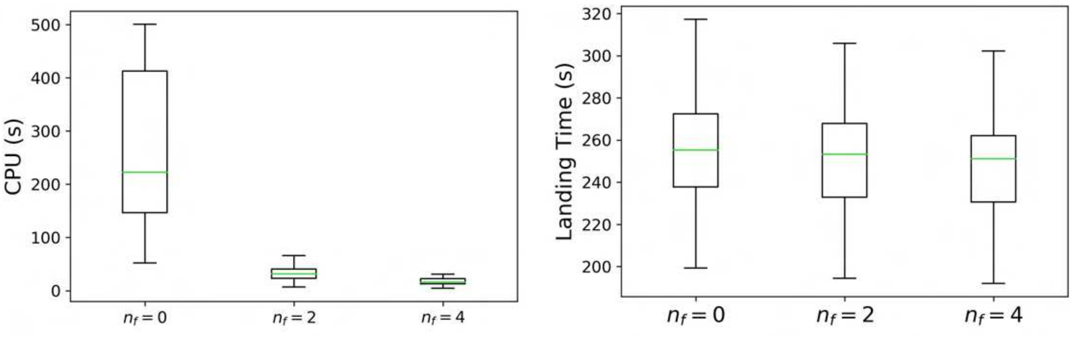
It is worth saying here that, objective (4.63) requires to minimise the time of flight as well as the scaled deviations from the linearised dynamic, and it also includes scaled penalty terms improving the stability of states as well as control variables. At the same time, the now linear constraints (4.64)-(4.76) ensure the correct visit of the waypoints, also respecting camera's and glider dynamic limitations, while linear constraints (4.77)-(4.85) ensure a correct glider landing. Finally, constraints (4.86)-(4.96) describe the linearised dynamic of each glider throughout its flight, as well as its stability limitations, whereas the rest of the constraints define variables domains.

4.5 Flight Examples

To show the efficacy of the proposed MILP formulation and examine the influence of fine time steps on the solution quality and computational times, we decided to solve several instance of the GRTOP problem using Gurobi 10.0.1 as a solver on a Viglen desktop, Intel(R) Core(TM) $i5-7500$ CPU with 3.40GHz and a RAM of 8 GB, running under Windows 10, with all the parameters set to their standard configurations. More precisely, we created 50 instances of the problem, where one glider is required to survey 4 waypoints randomly located in a space of $2 \times 2\ km^2$ and land in a predefined landing site, in absence of wind ($\beta = 0$). Then, assuming to know an initial trajectory for each of the instances, we compare the computational times for a given level of detail in the absence and presence of fine time steps. Therefore, each instance was solved utilising a fine partition with $|T_f| = 90$ time steps and a time step length of $\eta_f \simeq 3$s, whereas the coarse partitions possess $|T_c| = \left\lfloor \frac{|T_f|}{(n_f+1)} \right\rfloor$ time steps, where the number of fine time steps may vary, $n_f = 0, 2, 4$, as well as the time step length $\eta = (n_f + 1)\eta_f$. Other parameters used in this section are reported in Table 4.1, and we recall also that the flight envelope for the variables not specified here is defined in Table 3.2.

Parameter	Value	Unit
w_1	0.6	-
w_2	0.7	-
w_3	0.9	-
w_4	1.1	-
w_5	1.1	-
w_6	1.1	-
$\kappa_x, \kappa_y, \kappa_h$	100	m
κ_v	$\frac{3}{5}(v_{ub} - v_{lb})$	m/s
κ_γ	$\frac{3}{5}\gamma_{ub}$	radians
κ_φ	$\frac{1}{6}\pi$	radians
$\hat{\gamma}$	0.03	radians
$\hat{\mu}$	0.03	radians
$\hat{v}$	14.75	m/s

TABLE 4.1: Objective function weights and other model's parameters.

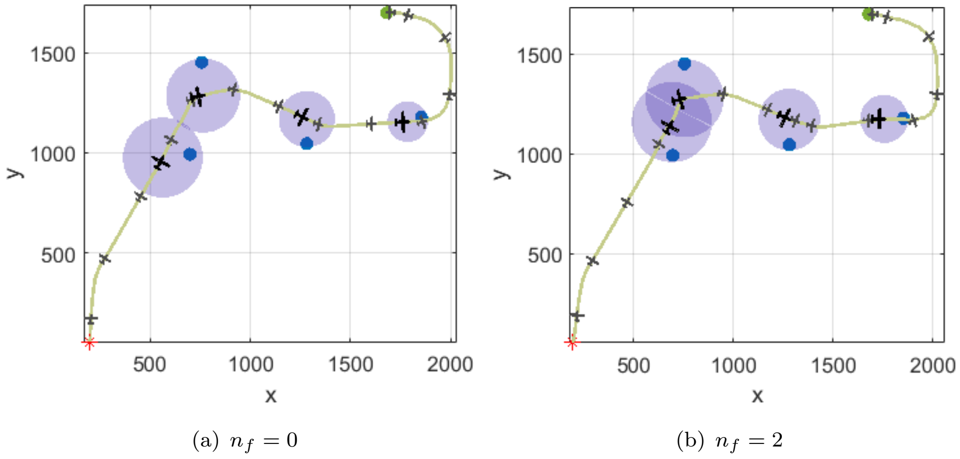


(a) CPU Usage

(b) Landing Times

FIGURE 4.6: Distribution of computational times (a) and landing times (b), in seconds, needed for solving the same problems while considering different number of fine time steps.

As it is possible to notice in Figure 4.6, increasing the number of fine time steps brings a clear benefit in reducing computational times required to solve the GRTOP, while keeping landing times substantially unchanged. Nonetheless, the advantage it is less pronounced after a certain number of fine time steps, $n_f = 2$ in the examined cases. Other than the overall mission and computational times, we examine the quality of the solution found with the insertion of fine time steps by comparing the trajectories obtained solving only one of the instances of the GRTOP first as is, i.e. with $n_f = 0$, and then utilising $n_f = 2$ fine time steps. To keep the same resolution quality we consider for the first case, $n_f = 0$, a time partition $T_c = T_f$ with 90 time steps and a time step length of $\eta_f = 2.97$s. Instead, in the case where we insert fine time steps, $n_f = 2$, we consider T_f to be the same as before, 90, whereas we reduce the coarse time partition to obtain $|T_c| = \left\lfloor \frac{|T_f|}{(n_f+1)} \right\rfloor$ and $\eta = (n_f + 1)\eta_f$. The results obtained with these settings are shown in Figure 4.7, Figure 4.8 and Figure 4.9, where first the trajectory and second the state and control variables profiles are compared.



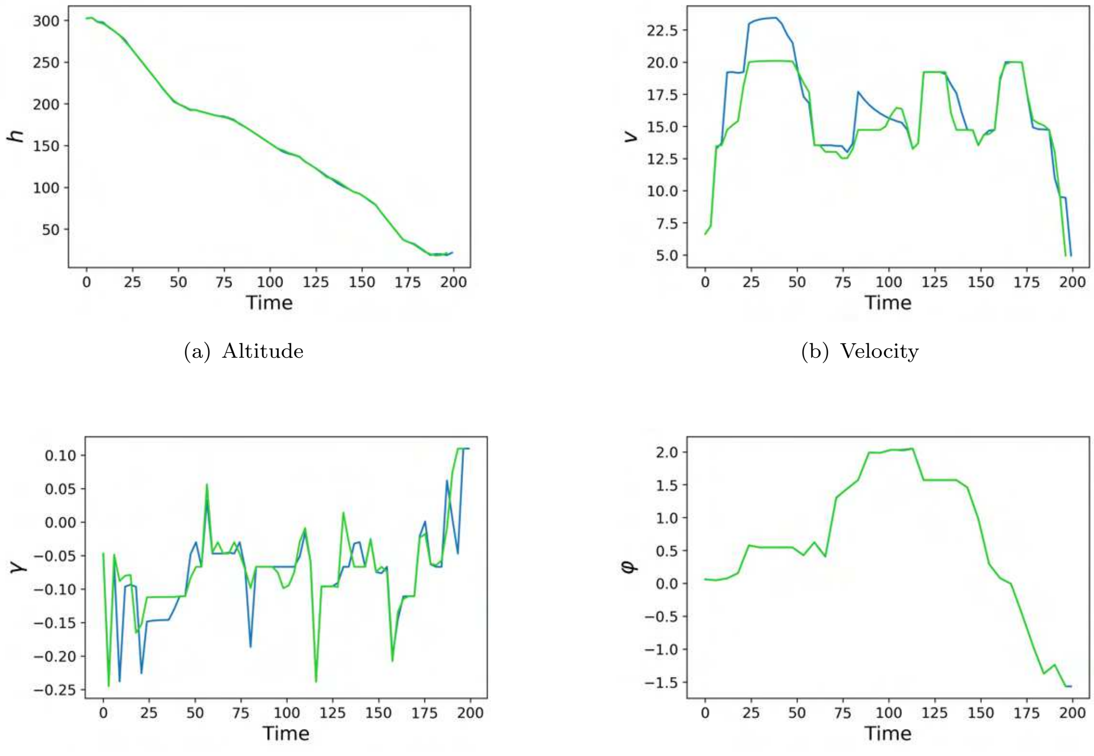
(a) $n_f = 0$

(b) $n_f = 2$

Figure 4.7: Glider trajectories, fine time step partition comparison. Green lines show gliders' trajectories seen from above for the solutions without fine time steps (a) and with $n_f = 2$ fine time steps (b). Red stars indicate the launching point, while blue dots represent the waypoints and green dots show the landing site's location. Camera's field of view while visiting the waypoints is also shown.

It is possible to observe that, even though both trajectories show a similar landing time, the solution is found utilising much more computational time, 76.3s for the case with $n_f = 0$ while 8.6s for the case with $n_f = 2$. In addition, we can observe that the two solutions exhibit almost identical trajectories, altitude profiles and heading angles, see Figure 4.7, Figure 4.8(a) and Figure 4.8(d) respectively. They also show similar patterns in terms of bank angles see Figure 4.9(b), with some differences in the times in which they visit the waypoints due to the definition of variables $a_{g,i,t}$ on a different time partition. Due to constraints (4.70)-(4.76), this is reflected into differences in velocity and flight path angle profiles, see Figure 4.8(b) and Figure 4.8(c), as well as in the lift coefficient, see Figure 4.9(a). Nonetheless, we can consider those discrepancies negligible, especially considering the benefit obtained in the time required to find the solution. In light of this, in the rest of the discussion we prefer to consider GRTOP models that account for fine time steps.

Up to this point, our focus has been on scenarios where some knowledge about an initial trajectory, equilibrium or (quasi) steady-state flight conditions is assumed to exist and be available. However, a central innovation and primary contribution of our research is the exploration and analysis of situations where this prior knowledge is absent. By exploring these scenarios, our work aims to advance the field by providing insights and solutions for cases where conventional trajectory knowledge may not be sufficient, thus expanding the scope and applicability of our research.



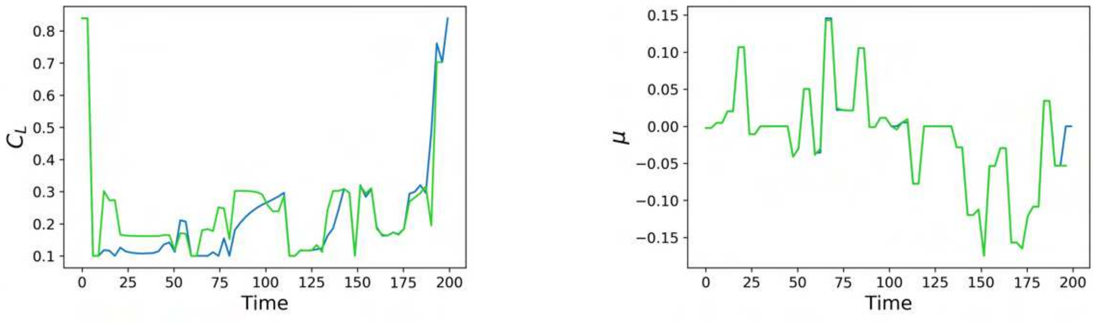
(a) Altitude

(b) Velocity

(c) Flight path angle

(d) Heading angle

Figure 4.8: State variables profiles obtained by solving the GRTOP without fine time steps (blue) and utilising 2 fine time steps (green).



(a) Lift coefficient

(b) Bank angle

Figure 4.9: Control variables profiles obtained by solving the GRTOP without fine time steps (blue) and utilising 2 fine time steps (green).

Chapter 5

Two-Step Algorithm for the GRTOP

Having prior information regarding an initial trajectory, equilibrium or (quasi) steady flight conditions is not guaranteed in many practical scenarios. Consequently, the investigation and scrutiny of situations where such knowledge is limited or absent hold primary importance and constitute a pivotal aspect of our research. In this chapter, we start by introducing techniques to linearly approximate the Pythagorean trigonometric identity, as well as bilinear terms, that allow us to handle nonlinearities stemming from the EoMs and to gain valuable insights useful for the construction of initial trajectory guesses. Subsequently, we provide ways to integrate these methodologies and insights into the previously presented GRTOP framework, with the purpose of finding potential solutions and flyable paths in a wider array of scenarios by means of a two-step solution algorithm. Finally, we provide the reader with examples of glider missions, illustrating algorithm's ability to find flyable trajectories in absence of prior information.

5.1 Construction of initial trajectory

Complex aerodynamic interactions lead to the highly nonlinear equations (3.1)-(3.6) governing gliders' dynamics which are difficult to solve, especially when prior knowledge is unavailable. Therefore, a simplification strategy for obtaining some primary information about an initial trajectory or equilibrium conditions is deemed to be necessary. To do so, the first step involves the identification and scrutiny of the nonlinear elements in the EoMs, which can be problematic to directly estimate or approximate, such as

- trigonometric functions; and
- bilinear and polynomial terms.

Thus, in the rest of this section we discuss how to manage trigonometric functions when a small angle approximation is not possible, as well as how to handle bilinear and polynomial terms, rendering all elements linear in our model formulation. Note that in what follows, even if all the existing and newly defined variables refer to a specific glider at a specific time step, for simplicity of notation we omit the subscripts g and t, unless it is strictly necessary for the discussion and the reader's understanding.

5.1.1 A Linear Model for Trigonometric Functions

In order to eliminate trigonometric functions from the model constraints, we can observe first that normal flight conditions implies that the flight path angle γ and the bank angle μ are relatively small. Therefore, we can directly adopt a small angle approximation and for each small angle $\omega = \gamma, \mu$ write

$$\begin{aligned} sin(\omega) &\simeq \omega \\ cos(\omega) &\simeq 1. \end{aligned}$$

However, such an assumption turns out to be false for the heading angle φ which can assume any value within 0 and 2π. Thus, an alternative approach becomes essential to overcome this challenge and handle trigonometric functions of φ. The main idea is then to define new variables representing the $sin(\varphi)$ and $cos(\varphi)$, that respect the Pythagorean trigonometric identity expressing the fundamental relation between sine and cosine, i.e.

$$sin^2(\varphi) + cos^2(\varphi) = 1. \tag{5.1}$$

Then, approximating equation (5.1) through linear constraints, it is possible to remove trigonometric functions from the model formulation for the price of introducing new variables and constraints. To do so, we first define

$$\begin{aligned} \psi_x &:= sin(\varphi) \\ \psi_y &:= cos(\varphi) \end{aligned}$$

and, for reasons that will be clearer later in the discussion, we decompose them into their positive and negative parts, introducing the variables

$$\psi_x^+ := max\{\psi_x, 0\} \tag{5.2}$$

$$\psi_x^- := max\{-\psi_x, 0\} \tag{5.3}$$

$$\psi_y^+ := max\{\psi_y, 0\} \tag{5.4}$$

$$\psi_y^- := max\{-\psi_y, 0\}, \tag{5.5}$$

so that

$$\psi_x = \psi_x^+ - \psi_x^-$$
$$\psi_y = \psi_y^+ - \psi_y^-$$

Substituting then the $sin(\varphi)$ and $cos(\varphi)$ with the just-introduced variables into the EoMs, we are able us to avoid trigonometric functions in the model formulation. However, a direct insertion of such variables is not sufficient, given that at this stage all of them can be chosen independently and their dynamic laws stemming from equation (3.6) are not taken into account. Therefore, we first ensure that only one between ψ_x^+ and ψ_x^-, and similarly only one between ψ_y^+ and ψ_y^-, can be greater than 0 by defining the binary variables

$$\phi_x = \begin{cases} 1, & \text{if } \psi_x^+ \geq 0 \\ 0, & \text{otherwise} \end{cases} \tag{5.6}$$

$$\phi_y = \begin{cases} 1, & \text{if } \psi_y^+ \geq 0 \\ 0, & \text{otherwise} \end{cases} \tag{5.7}$$

and constraining $\psi_x^+, \psi_x^-, \psi_y^+, \psi_y^-$ via the inequalities

$$\psi_x^+ \leq \phi_x \tag{5.8}$$
$$\psi_x^- \leq 1 - \phi_x \tag{5.9}$$
$$\psi_y^+ \leq \phi_y \tag{5.10}$$
$$\psi_y^- \leq 1 - \phi_y \tag{5.11}$$

It is worth noting that in this way we also ensure that the variable range is respected, i.e. $sin(\varphi), cos(\varphi) \in [-1, 1]$. Sine and cosine, and consequently the new variables, are then tied together by their fundamental relation (5.1), and hence we can direct our attention to model a linear description of the latter. To do so, we can observe that equation (5.1) is nothing other than requiring that $\|\boldsymbol{\psi}\|_2^2 = 1$, where $\boldsymbol{\psi} = (\psi_x, \psi_y)$. We can then think of taking advantage of the Euclidean norm approximation as derived before in equation (4.62), i.e.

$$\|\boldsymbol{\psi}\|_2 \simeq \lambda_2 \|\boldsymbol{\psi}\|_1 + (1 - \lambda_2) \|\boldsymbol{\psi}\|_\infty. \tag{5.12}$$

Yet, when dealing with equality constraints of the type $\|\boldsymbol{\psi}\|_2^2 = 1$, upper bound strategies adopted in section 4.3.1 are not sufficient to approximate the border of the unit ball, and hence the need of a dedicated approach become apparent. Therefore, we first have to find a way to correctly retrieve the value for the module and maximum function. For the former, we can observe that the absolute value of a number x can be obtained by leveraging on its decomposition into his positive and negative parts as $|x| = x^+ + x^-$. Consequently, we can write

$$|\psi_x| = \psi_x^+ + \psi_x^- \tag{5.13}$$

$$|\psi_y| = \psi_y^+ + \psi_y^-, \tag{5.14}$$

obtaining that

$$\|\boldsymbol{\psi}\|_1 = |\psi_x| + |\psi_y| = \psi_x^+ + \psi_x^- + \psi_y^+ + \psi_y^-. \tag{5.15}$$

On the other hand, $\|\boldsymbol{\psi}\|_\infty = max\{|\psi_x|, |\psi_y|\}$ can be described by first defining a variable ψ^{max} representing it, also noting that

$$\psi^{max} \geq |\psi_x| \tag{5.16}$$

$$\psi^{max} \geq |\psi_y|, \tag{5.17}$$

and second by establishing a further binary variable

$$\phi^{max} = \begin{cases} 1, & \text{if } |\psi_x| \geq |\psi_y| \\ 0, & \text{otherwise} \end{cases}, \tag{5.18}$$

as well as constraints

$$\psi^{max} \leq |\psi_x| + (1 - \phi^{max}) \tag{5.19}$$

$$\psi^{max} \leq |\psi_y| + \phi^{max}. \tag{5.20}$$

Once a complete representation of $\|\boldsymbol{\psi}\|_1$ and $\|\boldsymbol{\psi}\|_\infty$ is given, we are ready to linearly approximate the fundamental trigonometric relation (5.1) by defining the variable ξ, constrained by

$$\xi = \lambda_2(|\psi_x| + |\psi_y|) + (1 - \lambda_2)\psi^{max} \tag{5.21}$$

$$\xi = 1, \tag{5.22}$$

and to find sensible values for ψ_x, ψ_y, and hence for $sin(\varphi)$ and $cos(\varphi)$. Although, without further requirements, at this point these values result to be independent at each time step, given the absence of their time law description. To limit then the nonphysical behaviours of the gliders, as a final passage for getting rid of the trigonometric functions and obtain a reasonable basic approximation of the corresponding EoMs, we decide to define empirical bounds and restrict the changes on ψ_x, ψ_y. In particular, we aim at finding an upper bound $\overline{\psi} \geq 0$ on $|\dot{\psi}_x|$ and $|\dot{\psi}_y|$ such that

(i) the implied rate of heading angle changes, $\dot{\varphi}$, allows for appreciable direction changes and feasible fast turns;

(ii) the bank angle μ required for such turns is generally not too steep;

(iii) starting the turn with such a bank angle results in a reasonable sink rate without significant changes in flight path angle and in sudden losses of altitude. In other words, with all the other variables being the same, we require that $\dot{\gamma}_{\mu}$, the infinitesimal changes of flight path angle when deploying bank angle μ, does not differ too much from $\dot{\gamma}_0$, the infinitesimal changes of flight path angle without turning, i.e. when $\mu = 0$.

To do so, we start noting that

$$-\overline{\psi} \leq \dot{\psi}_x = \frac{dsin(\varphi)}{d\tau} = cos(\varphi)\dot{\varphi} = \psi_y\dot{\varphi} \leq \overline{\psi} \tag{5.23}$$

$$-\overline{\psi} \leq \dot{\psi}_y = \frac{dcos(\varphi)}{d\tau} = -sin(\varphi)\dot{\varphi} = -\psi_x\dot{\varphi} \leq \overline{\psi}. \tag{5.24}$$

Then, we utilise a simulation approach to find values of $\overline{\psi}$ that respect the conditions (i)-(iii). More precisely, with a fixed value for $\overline{\psi}$, we repeat the following steps 300000 times

1. we generate glider's quasi-steady flight conditions by generating random values for the lift coefficient and direction of flight, $C_{L_{eq}} \sim U(0.1, 1.17)$ and $\varphi_{eq} \sim U(0, 2\pi)$;

2. we compute the sine and cosine of φ_{eq} to find ψ_x and ψ_y, and we apply equations (3.26)-(3.27) to find the equilibrium velocity and flight path angle, v_{eq} and γ_{eq};

3. considering a glider flying within the so-found quasi-steady flight conditions and given the fixed value of $\overline{\psi}$, we observe that the maximum change of φ is limited by

$$\dot{\varphi} \leq \frac{\overline{\psi}}{|\psi_y|}$$

$$\dot{\varphi} \leq \frac{\overline{\psi}}{|\psi_x|}$$

therefore, we set a target turning rate at its maximum, i.e. $\dot{\varphi}_{max} = min\{\frac{\overline{\psi}}{|\psi_y|}, \frac{\overline{\psi}}{|\psi_x|}\}$;

4. inverting equation (3.6), we can find the corresponding bank angle μ_{max} required to start the turn at the specified rate $\dot{\varphi}_{max}$;

5. with all the other variables being the same, we evaluate $\dot{\gamma}_{max}$ and $\dot{\gamma}_0$, being respectively the variation of flight path angle obtained from equation (3.5) when the glider starts turning at $\mu = \mu_{max}$, and when it is not turning, i.e. $\mu = 0$;

6. we collect the values of $\dot{\varphi}_{max}$, μ_{max}, $\dot{\gamma}_{max}$ and $\dot{\gamma}_0$.

For each fixed value of $\overline{\psi}$, it is then possible to construct empirical distributions for the collected values. Noting then that the higher the value of $\overline{\psi}$ the higher the bank angle required for the maximum turn and hence the higher the sudden loss of altitude, we decide to start focusing on condition (iii). Knowing that a typical safe descent rate is reached with flight path angles between 3^o and 5^o, see International Civil Aviation Organisation (2018), we require that an acceptable value for $\overline{\psi}$ is such that 90% of the maximum turns are obtained with a sink rate that does not increase more than 5^o per second with respect to the non-turning flight conditions. In other words, the 10^{th} percentile of $\dot{\gamma}_{max} - \dot{\gamma}_0$ distribution, called p_{10}^{γ}, should not be less than -5^o. Noting that p_{10}^{γ} is a decreasing function of $\overline{\psi}$, as we can observe in Figure 5.1, we can make $\overline{\psi}$ vary and find the

$$sup\{\overline{\psi} \mid p_{10}^{\gamma}(\overline{\psi}) \geq -5^o\}. \tag{5.25}$$

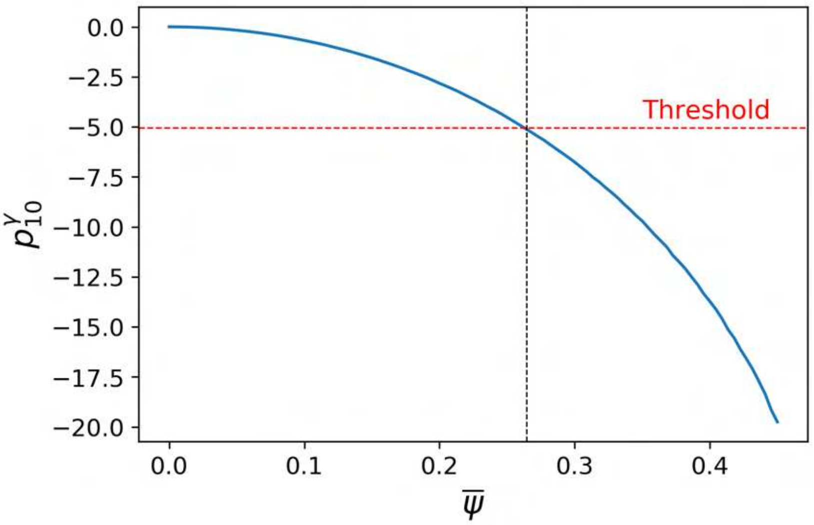
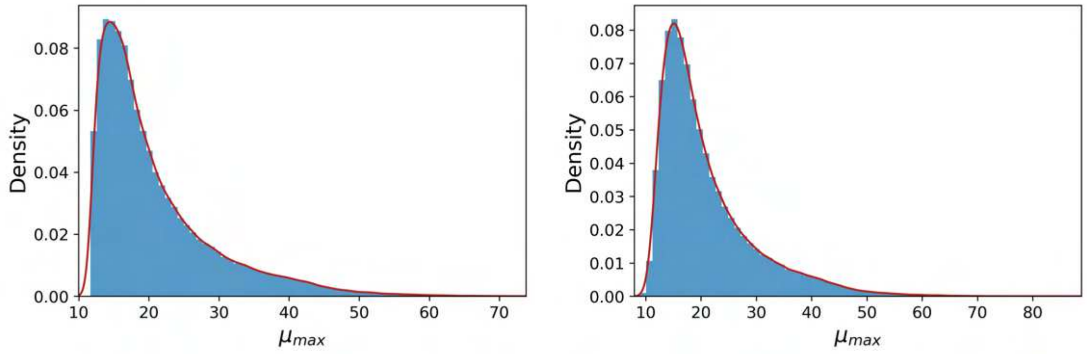


Figure 5.1: 10^{th} percentile of the sink rate difference as a function of $\overline{\psi}$. Threshold level at -5^o is also shown as well as the corresponding $\overline{\psi}$ value.

Doing so, we obtain that a value of $\overline{\psi} \simeq 0.265$ serves the objective. We would like now to observe if conditions (i) and (ii) are respected for such a value of $\overline{\psi}$, otherwise we would need to find a trade-off between those. Luckily, we can observe that such $\overline{\psi}$ results in an average target turn rate of 17^o per second, which would allow the glider to reach a 180 degrees turn in roughly 10.5 seconds, and in a maximum target turn rate of 21.5^o which would allow the glider to reach a 180 degrees turn in roughly 8 seconds. This rate of turn can be considered acceptable when observing flight data available at Skylines Flight Data, where many manned aerobatic glider flights are recorded. Consequently, we consider a value of $\overline{\psi}$ equal to 0.265 acceptable for attaining condition (i). Finally, in Figure 5.2(a), we can observe the bank angle μ_{max} distribution corresponding to the same value of $\overline{\psi}$.



(a) Turns start from steady flight conditions

(b) Turns start from general flight conditions

FIGURE 5.2: Target bank angle distribution when the glider is requested to turn at a rate of change corresponding to $\dot{\varphi}_{max}$ for a value of $\overline{\psi} = 0.265$. In (a) the distribution of μ_{max} when the turn starts from some (quasi) steady flight conditions. In (b) its distribution when the turn starts from more generic flight conditions.

It is possible to observe that all the simulated turns happen within a bank angle μ_{max} of roughly 70 degrees and that 90% of them falls within 32.8 degrees. Knowing that typical airline operations require a bank angle of 25 degrees, or bank angle giving a rate of turn of 3 degrees per second, see International Civil Aviation Organisation (2018), and that the gliding community considers steep bank angles as the ones above 30^o or 40^o, see British Gliding Association (2017) and Aviation Security Service, we consider those values acceptable for respecting the required conditions. Nonetheless, it is worth noting here that by increasing the above thresholds, it is possible to determine more aggressive values of $\overline{\psi}$ and pushing glider performances to the limit. Conversely, it is also possible to obtain more mild turns by reducing threshold values, reaching then safer flight decisions.

In addition to the above, knowing that a glider is not always flying at equilibrium, we further verify the validity of our limitations by perturbing the (quasi) steady-state flight conditions. More precisely, keeping $\overline{\psi} = 0.265$, in step 2 of our simulation approach we decided to use velocity and flight path angle values equal to $v' = v_{eq} + \nu$ and $\gamma' = \gamma_{eq} + \upsilon$ respectively, where $\nu \sim N(0, 0.75)$ and $\upsilon \sim N(0, 0.017)$. As before, we then evaluate the bank angle μ_{max} attained to perform the targeted maximum turn starting from the so-obtained flight conditions, as well as the changes in sink rate and altitude losses. First, it is possible to observe that sometimes a greater bank angle is necessary to perform the required turns, as we can see from Figure 5.2(b) where we depict the new bank angle distribution, and second it is also possible to notice that now 90% of altitude losses happen within a slightly bigger change of flight path angle, that is 5.2^o instead of the 5^o threshold of before. Nonetheless, these changes in bank angle distribution and increase of sink rate changes are not deemed to be significant. Once again we remind he reader that it is always possible to tighten or loose the threshold on p_{10}^{γ} to reach more conservative or aggressive flight turns.

In light of this, we are now able to reasonably limit heading angle variation by writing the following constraints

$$\psi_{xt+1} - \psi_{xt} = \psi^{+}_{x\ t+1} - \psi^{-}_{x\ t+1} - \psi^{+}_{x\ t} + \psi^{-}_{x\ t} \leq \eta\overline{\psi},\ \forall t \tag{5.26}$$

$$\psi_{xt} - \psi_{xt+1} = \psi^{+}_{x\ t} - \psi^{-}_{x\ t} - \psi^{+}_{x\ t+1} + \psi^{-}_{x\ t+1} \leq \eta\overline{\psi},\ \forall t \tag{5.27}$$

$$\psi_{yt+1} - \psi_{yt} = \psi^{+}_{y\ t+1} - \psi^{-}_{y\ t+1} - \psi^{+}_{y\ t} + \psi^{-}_{y\ t} \leq \eta\overline{\psi},\ \forall t \tag{5.28}$$

$$\psi_{yt} - \psi_{yt+1} = \psi^{+}_{y\ t} - \psi^{-}_{y\ t} - \psi^{+}_{y\ t+1} + \psi^{-}_{y\ t+1} \leq \eta\overline{\psi},\ \forall t. \tag{5.29}$$

Utilising the just described approach and techniques, we are then able to get rid of the trigonometric functions present in the nonlinear constraints stemming from the EoMs. Nonetheless, nonlinearities still occur in the EoMs, requiring further approximation efforts.

5.1.2 Approximation Strategies for Bilinear and Power Terms

Although in the previous section we free the equations of motion from trigonometric terms, nonlinearities are still present in the newly approximated EoMs. More precisely, excluding the wind effect on $\dot{v}$ since its values are generally negligible unless the wind is blowing at an extremely high gradient, and ignoring the equation governing the changes of flight path angle γ, we currently have that

$$\dot{x} = v\psi_x^{+} - v\psi_x^{-} + \beta h \tag{5.30}$$

$$\dot{y} = v\psi_y^{+} - v\psi_y^{-} \tag{5.31}$$

$$\dot{h} = v\gamma \tag{5.32}$$

$$\dot{v} = -\frac{1}{2m}\rho S_w (C_{D_0} + k_A C_L^2) v^2 - g\gamma. \tag{5.33}$$

Focusing first on equations (5.30) and (5.31), each bilinear term can be linearly approximated following techniques similar to the construction of McCormick envelopes, see McCormick (1976). More precisely, for bounded real variables $x \in [x_L, x_U]$ and $y \in [y_L, y_U]$, it is possible to linearly approximate their product xy by defining a new variable $w = xy$ and noting that it must satisfy the following inequalities

$$w \geq x_L y + x y_L - x_L y_L \tag{5.34}$$

$$w \geq x_U y + x y_U - x_U y_U \tag{5.35}$$

$$w \leq x_U y + x y_L - x_U y_L \tag{5.36}$$

$$w \leq x_L y + x y_U - x_L y_U. \tag{5.37}$$

Considering now for example the product $v\psi_x^+$, and noting that $v \in [v_{lb}, v_{ub}]$, while $\psi_x^+ \in [0, 1]$, we can define a new variable $v_x^+ = v\psi_x^+$ and readjust McCormick inequalities (5.34)-(5.37) to write the following constraints

$$v_{lb}\psi_x^+ \leq \quad v_x^+ \quad \leq v_{ub}\psi_x^+ \tag{5.38}$$

$$v_{ub}(\psi_x^+ - 1) \leq v_x^+ - v \leq v_{lb}(\psi_x^+ - 1). \tag{5.39}$$

Similarly, we can determine new variables and constraints for the other bilinear terms. It is worth noting here that the decomposition of sine and cosine variables into their positive and negative parts helps also here in reaching better approximations, since tighter bounds can be obtained.

On the other hand, although it is possible to apply the McCormick envelope technique to equation (5.32), we decide to appriximate the product by utilising the linearisation technique introduced in Chapter 3. This will help in avoiding the definition of new binary variables to create tighter bounds, here necessary to evaluate the climbing or descending conditions of the glider. Thus, we decide to linearise $v\gamma$ using a Taylor expansion around points (v^o, γ^o), obtaining the approximation

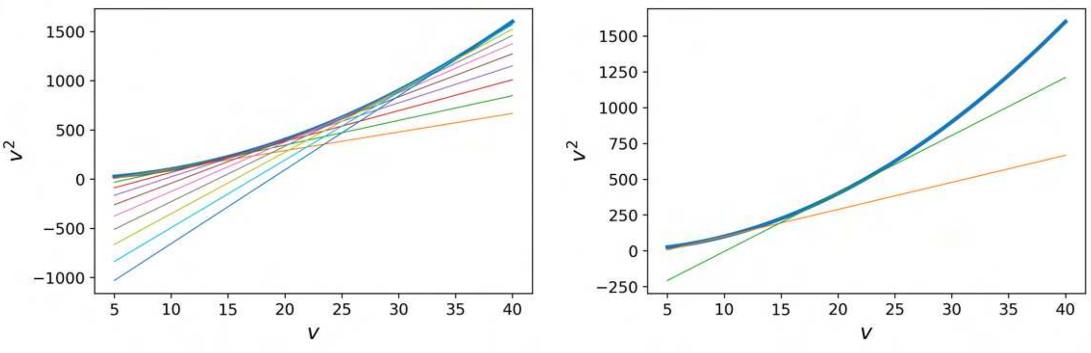
$$v\gamma \simeq v^o\gamma^o + v^o(\gamma - \gamma^o) + \gamma^o(v - v^o). \tag{5.40}$$

Now, in order to choose sensible values for (v^o, γ^o), we turn our attention to equation (5.33) first, with a special focus on the nonlinear term v^2. Following the same Taylor expansion, it can also be linearised as

$$v^2 \simeq (v^o)^2 + 2v^o(v - v^o). \tag{5.41}$$

From this, and by choosing different values v_i^o, it is possible to construct a linear envelope for v^2, as it is depicted in Figure 5.3(a).

If on one hand the linear envelope approximates well v^2, on the other hand, the utilisation of several values v_i^o would lead to an increased problem complexity due to the necessity of introducing different binary variables to help the selection of the right line that approximate v^2 at a specific time step. Hence, we decide that two velocity regimes

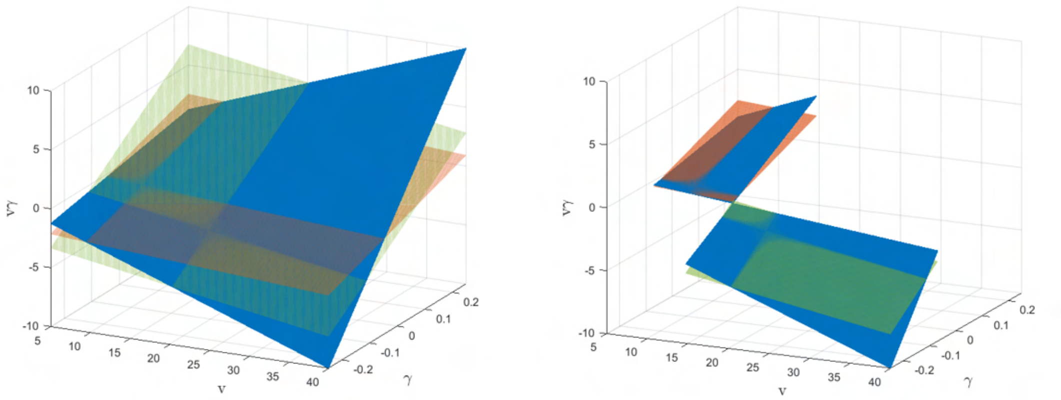


(a) Linear envelope with 10 lines

(b) Linear envelope with 2 lines

FIGURE 5.3: Linear approximation of v^2 (thick blue curve). In (a), 10 lines are forming a linear envelope for v^2, while in (b) only 2 lines are chosen to form the linear envelope.

might suffice to find reasonable approximation for v^2 and consequently help the linearisation of equations (5.32) and (5.33). A first possible choice for values of v_i^o is using the preferred equilibrium flight conditions of Remark 3.2, where the glider is flying at a moderate speed of $v^{\downarrow} = 9.45$m/s. Consequently, as second point must be such that the glider is flying at a higher velocity. Considering then that a steep descend rate of -15^o and a minimum lift coefficient of 0.1 lead to a maximum velocity of around 31m/s, and having selected as a first equilibrium point the one at roughly 9.5m/s, we decided to set the second equilibrium velocity at an intermediate value of $v^{\uparrow} = 20.25$m/s. Utilising these values for v_i^o, it is possible to approximate v^2 as depicted in Figure 5.3(b). We can notice that, although we have reduced the goodness of the approximation, we start having a significant divergence for limit flight conditions with velocity higher than 30m/s. Moreover, we reduced to the bare minimum the number of binary variables necessary for a good v^2 approximation. Now, knowing the equilibrium velocities $v^{\downarrow}$ and $v^{\uparrow}$, it is possible to invert equation (3.26), find the corresponding value of the lift coefficient at equilibrium, $C_L^{\downarrow}$ and $C_L^{\uparrow}$, and then, using equality (3.27), derive the relative value for $\gamma^{\downarrow}$ and $\gamma^{\uparrow}$. With this in mind, we are able to reasonably approximate the bilinear term $v\gamma$ by using only two hyperplane defined by the pairs $(v^{\downarrow}, \gamma^{\downarrow})$, $(v^{\uparrow}, \gamma^{\uparrow})$ through Taylor expansion (5.40), as shown in Figure 5.4(a). The idea is then that these hyperplane are sufficient for a good approximation because, a part from very edge cases, a glider necessitate lower flight path angles ($\gamma \simeq \gamma^{\uparrow}$) to fly fast ($v \simeq v^{\uparrow}$), while higher angles ($\gamma \simeq \gamma^{\downarrow}$) to fly slow ($v \simeq v^{\downarrow}$). Conversely, if a glider modify its flight path angle, it is possible to show that it reaches the corresponding equilibrium speed fairly quickly. Therefore, it is rarely the case where we observe fast velocities ($v \simeq v^{\uparrow}$) and small descending angles ($\gamma \simeq \gamma^{\downarrow}$), or slow speed ($v \simeq v^{\downarrow}$) and big descending angles ($\gamma \simeq \gamma^{\uparrow}$) for long time periods. In light of this, as it is possible to observe in Figure 5.4(b), we can choose the



(a) Full flight envelope for $v\gamma$

(b) Two regimes flight envelope for $v\gamma$

FIGURE 5.4: Linearisation of $v\gamma$ (blue surface). In (a) we observe the approximating hyperplane corresponding to $(v^{\downarrow}, \gamma^{\downarrow})$ (orange), and $(v^{\uparrow}, \gamma^{\uparrow})$ (green) on the entire flight envelope for $v\gamma$. In (b) it is possible to see the same hyperplane depicted within typical flight domain, excluding edge cases where $v \simeq v^{\uparrow}$ and $\gamma \simeq \gamma^{\downarrow}$, or $v \simeq v^{\downarrow}$ and $\gamma \simeq \gamma^{\uparrow}$.

best approximation of $v\gamma$ in most of the flight conditions, by defining binary variables

$$\chi^{\uparrow} = \begin{cases} 1, & \text{if the glider is flying in a fast regime} \\ 0, & \text{otherwise} \end{cases} \tag{5.42}$$

and constraints

$$v \leq \frac{v^{\downarrow} + v^{\uparrow}}{2} + v_{ub}\chi^{\uparrow} \tag{5.43}$$

$$v \geq \frac{v^{\downarrow} + v^{\uparrow}}{2} + v_{ub}(1 - \chi^{\uparrow}) \tag{5.44}$$

$$\gamma \leq \gamma^{\downarrow} + \gamma_{ub}(1 - \chi^{\uparrow}), \tag{5.45}$$

with the latter constraint (5.45) helping in avoiding edge cases by ensuring that a glider cannot fly at a high speed without a sufficiently steep flight path angle. It is also worth mentioning here that, while other variables such directional changes might require responsive adjustments, the range of different velocity regimes is wide enough to permit us to define the binary variables $\chi^{\uparrow}$ on a coarse time partition, so improving computational speed for the initial trajectory problem.

We are now able to approximate equations (5.32)-(5.33) as

$$v_h := \dot{h} \leq v^{\uparrow}\gamma^{\uparrow} + v^{\uparrow}(\gamma - \gamma^{\uparrow}) + \gamma^{\uparrow}(v - v^{\uparrow}) + M(1 - \chi^{\uparrow}) \tag{5.46}$$

$$v_h := \dot{h} \geq v^{\uparrow}\gamma^{\uparrow} + v^{\uparrow}(\gamma - \gamma^{\uparrow}) + \gamma^{\uparrow}(v - v^{\uparrow}) - M(1 - \chi^{\uparrow}) \tag{5.47}$$

$$
\begin{aligned}
v_h &:= \dot{h} \leq v^{\downarrow}\gamma^{\downarrow} + v^{\downarrow}(\gamma - \gamma^{\downarrow}) + \gamma^{\downarrow}(v - v^{\downarrow}) + M\chi^{\uparrow} && (5.48)\\
v_h &:= \dot{h} \geq v^{\downarrow}\gamma^{\downarrow} + v^{\downarrow}(\gamma - \gamma^{\downarrow}) + \gamma^{\downarrow}(v - v^{\downarrow}) - M\chi^{\uparrow} && (5.49)\\
\mathfrak{z} &:= \dot{v} \leq -\frac{1}{2m}\rho S_w(C_{D_0} + k_A {C_L^{\uparrow}}^2)\left({v^{\uparrow}}^2 + 2v^{\uparrow}(v - v^{\uparrow})\right) - g\gamma + M(1 - \chi^{\uparrow}) && (5.50)\\
\mathfrak{z} &:= \dot{v} \geq -\frac{1}{2m}\rho S_w(C_{D_0} + k_A {C_L^{\uparrow}}^2)\left({v^{\uparrow}}^2 + 2v^{\uparrow}(v - v^{\uparrow})\right) - g\gamma - M(1 - \chi^{\uparrow}) && (5.51)\\
\mathfrak{z} &:= \dot{v} \leq -\frac{1}{2m}\rho S_w(C_{D_0} + k_A {C_L^{\downarrow}}^2)\left({v^{\downarrow}}^2 + 2v^{\downarrow}(v - v^{\downarrow})\right) - g\gamma + M\chi^{\uparrow} && (5.52)\\
\mathfrak{z} &:= \dot{v} \geq -\frac{1}{2m}\rho S_w(C_{D_0} + k_A {C^{\downarrow}}^2)\left({v^{\downarrow}}^2 + 2v^{\downarrow}(v - v^{\downarrow})\right) - g\gamma - M\chi^{\uparrow}, && (5.53)
\end{aligned}
$$

where tighter bounds can be found by choosing $M = v_{ub}$. Deploying then the usual Euler scheme, it is possible to translate the inequalities above into valid linear constraints.

Finally, we decide to ignore in this phase constraints stemming from the equation of motion relative to $\dot{\gamma}$ because it is unreasonable to impose rough, worst case limitations on its changes since they can be very rapid also in normal flight conditions, and also because a more detailed inclusion might complicate the problem and its solving times substantially. For example, considering a glider flying at the preferred (quasi) steady flight conditions of Remark 3.2, it is sufficient to choose a lift coefficient equal to 0.1 that in one second it is possible to experience a variation of γ of $\simeq 0.9$ radians which is very close to the flight envelope limits. Worse changes are possible at slightly lower speeds. Strictly speaking, we are actually allowed to insert such worse case limitation on changes of flight path angle, however, in all gliders' flights that we show during the manuscript, they will never be active constraints due to the fact that our time partitions generally present a length η greater than 1s. Nonetheless, in this phase we restrict bounds on γ to ensure a reduction of extreme flight conditions. Then, ignoring also control variables, assuming that with the new given constraints there exist always C_L and μ such that the glider can fly close to the newly found initial trajectory, we are now ready to write a complete formulation of the problem and determine such initial trajectory.

Remark 5.1. It is also possible to consider in the model an average wind effect on $\dot{v}$ and on the changes of ψ_x (respectively ψ_y), consequently on $\dot{\varphi}$, by looking at equations (3.4) and (3.6). However, the effect is often negligible with respect to the other terms, and therefore we omit its estimate for simplicity and brevity of the model and manuscript.

5.1.3 A MILP Formulation for the Initial Trajectory Problem

Deciding to deploy the same logic and notation adopted before in Chapter 4 when defining a two time level partition, and smoothing the trajectory by inserting penalties for limiting its sudden changes[1], we can provide the reader with a complete compact formulation for the problem of finding the initial trajectory as follows

[1] We used the same variable terminology and variables' notation for the penalties on state changes as we used for the penalties of deviations from the initial trajectory in Chapter 4.

Model 0 - Initial Trajectory for the GRTOP

$$\begin{aligned}\min \ & - w_1 * \sum_{g\in G, l\in L, t\in T_c\setminus\{0\}} \eta \ b_{g,l,t} \\ & + w_2 \sum_{s\in S, g\in G, t\in T_f} \eta_f \ \frac{\varepsilon_{s,g,t}}{\kappa_s} \\ & + w_3 \sum_{s\in S, g\in G} t_f \ \frac{\epsilon_{s,g}}{\kappa_s} \\ & + w_4 \sum_{s\in S, g\in G, t\in T_f} \eta_f \ \frac{\lambda_{s,g,t}}{\kappa_s} \\ & + w_5 \sum_{s\in S, g\in G} t_f \ \frac{\Lambda_{s,g}}{\kappa_s}\end{aligned} \tag{5.54}$$

$$\text{subject to} \sum_{g\leq i, t\in T_c} a_{g,i,t} \geq 1, \ \forall i \in W \tag{5.55}$$

$$d^x_{g,i,t} \geq x_{g,t} - \bar{x}_i, \ \forall g \in G, \forall i \in W, \forall t \in T_c \tag{5.56}$$

$$d^x_{g,i,t} \geq \bar{x}_i - x_{g,t}, \ \forall g \in G, \forall i \in W, \forall t \in T_c \tag{5.57}$$

$$d^y_{g,i,t} \geq y_{g,t} - \bar{y}_i, \ \forall g \in G, \forall i \in W, \forall t \in T_c \tag{5.58}$$

$$d^y_{g,i,t} \geq \bar{y}_i - y_{g,t}, \ \forall g \in G, \forall i \in W, \forall t \in T_c \tag{5.59}$$

$$\sum_{j\in\{x,y\}} d^j_{g,i,t} \leq (h_{g,t} - \bar{r}_i) + M(1 - a_{g,i,t}), \ \forall g \in G, \forall i \in W, \forall t \in T_c \tag{5.60}$$

$$h_{g,t} \leq \bar{h}_i a_{g,i,t} + h_{ub}(1 - a_{g,i,t}), \ \forall g \in G, \forall i \in W, \forall t \in T_c \tag{5.61}$$

$$h_{g,t} \geq \underline{h}_i a_{g,i,t} + h_{lb}(1 - a_{g,i,t}), \ \forall g \in G, \forall i \in W, \forall t \in T_c \tag{5.62}$$

$$\gamma_{g,t} \leq \hat{\gamma} a_{g,i,t} + \gamma_{ub}(1 - a_{g,i,t}), \ \forall g \in G, \forall i \in W, \forall t \in T_c \tag{5.63}$$

$$\gamma_{g,t} \geq -\hat{\gamma} a_{g,i,t} + \gamma_{lb}(1 - a_{g,i,t}), \ \forall g \in G, \forall i \in W, \forall t \in T_c \tag{5.64}$$

$$v_{g,t} \leq \hat{v} a_{g,i,t} + v_{ub}(1 - a_{g,i,t}), \ \forall g \in G, \forall i \in W, \forall t \in T_c \tag{5.65}$$

$$\sum_{l\in L, t\in T_c} b_{g,l,t} \geq 1, \ \forall g \in G \tag{5.66}$$

$$r^x_{g,i,t} \geq x_{g,t} - \tilde{x}_l, \ \forall g \in G, \forall l \in L, \forall t \in T_c \tag{5.67}$$

$$r^x_{g,i,t} \geq \tilde{x}_l - x_{g,t}, \ \forall g \in G, \forall l \in L, \forall t \in T_c \tag{5.68}$$

$$r^y_{g,i,t} \geq y_{g,t} - \tilde{y}_l, \ \forall g \in G, \forall l \in L, \forall t \in T_c \tag{5.69}$$

$$r^y_{g,i,t} \geq \tilde{y}_l - y_{g,t}, \ \forall g \in G, \forall l \in L, \forall t \in T_c \tag{5.70}$$

$$r^h_{g,i,t} \geq h_{g,t}, \ \forall g \in G, \forall l \in L, \forall t \in T_c \tag{5.71}$$

$$r^h_{g,i,t} \geq -h_{g,t}, \ \forall g \in G, \forall l \in L, \forall t \in T_c \tag{5.72}$$

$$\sum_{j\in\{x,y,h\}} r^j_{g,l,t} \leq \tilde{r}_l + M(1 - b_{g,l,t}), \ \forall g \in G, \forall l \in L, \forall t \in T_c \tag{5.73}$$

$$\sum_{\tilde{t}\leq t} b_{g,l,\tilde{t}} \leq N(1 - a_{g,i,t}), \ \forall g \in G, \forall l \in L, \forall i \in W, \forall t \in T_c \tag{5.74}$$

$$\psi^{+}_{x\ g,t} \leq \phi_{x g,t},\ \forall g \in G, \forall t \in T_f \tag{5.75}$$

$$\psi^{-}_{x\ g,t} \leq 1 - \phi_{x g,t},\ \forall g \in G, \forall t \in T_f \tag{5.76}$$

$$\psi^{+}_{y\ g,t} \leq \phi_{y g,t},\ \forall g \in G, \forall t \in T_f \tag{5.77}$$

$$\psi^{-}_{y\ g,t} \leq 1 - \phi_{y g,t},\ \forall g \in G, \forall t \in T_f \tag{5.78}$$

$$\psi^{max}_{g,t} \geq \psi^{+}_{x\ g,t} + \psi^{-}_{x\ g,t},\ \forall g \in G, \forall t \in T_f \tag{5.79}$$

$$\psi^{max}_{g,t} \geq \psi^{+}_{y\ g,t} + \psi^{-}_{y\ g,t},\ \forall g \in G, \forall t \in T_f \tag{5.80}$$

$$\psi^{max}_{g,t} \leq \psi^{+}_{x\ g,t} + \psi^{-}_{x\ g,t} + (1 - \phi^{max}_{g,t}),\ \forall g \in G, \forall t \in T_f \tag{5.81}$$

$$\psi^{max}_{g,t} \leq \psi^{+}_{y\ g,t} + \psi^{-}_{y\ g,t} + \phi^{max}_{g,t},\ \forall g \in G, \forall t \in T_f \tag{5.82}$$

$$\xi_{g,t} = \lambda_2(\psi^{+}_{x\ g,t} + \psi^{-}_{x\ g,t} + \psi^{+}_{y\ g,t} + \psi^{-}_{y\ g,t}) + (1 - \lambda_2)\psi^{max}_{g,t},\ \forall g \in G, \forall t \in T_f \tag{5.83}$$

$$\xi_{g,t} = 1,\ \forall g \in G, \forall t \in T_f \tag{5.84}$$

$$\psi^{+}_{x\ g,t+1} - \psi^{-}_{x\ g,t+1} - \psi^{+}_{x\ g,t} + \psi^{-}_{x\ g,t} \leq \eta_f \overline{\psi},\ \forall g \in G, \forall t \in T_f \setminus \{N(n_f + 1)\} \tag{5.85}$$

$$\psi^{-}_{x\ g,t+1} - \psi^{+}_{x\ g,t+1} - \psi^{-}_{x\ g,t} + \psi^{+}_{x\ g,t} \leq \eta_f \overline{\psi},\ \forall g \in G, \forall t \in T_f \setminus \{N(n_f + 1)\} \tag{5.86}$$

$$\psi^{+}_{y\ g,t+1} - \psi^{-}_{y\ g,t+1} - \psi^{+}_{y\ g,t} + \psi^{-}_{y\ g,t} \leq \eta_f \overline{\psi},\ \forall g \in G, \forall t \in T_f \setminus \{N(n_f + 1)\} \tag{5.87}$$

$$\psi^{-}_{y\ g,t+1} - \psi^{+}_{y\ g,t+1} - \psi^{-}_{y\ g,t} + \psi^{+}_{y\ g,t} \leq \eta_f \overline{\psi},\ \forall g \in G, \forall t \in T_f \setminus \{N(n_f + 1)\} \tag{5.88}$$

$$v_{g,t} \leq \frac{v^{\downarrow} + v^{\uparrow}}{2} + v_{ub}\chi^{\uparrow}_{g,\pi(t)},\ \forall g \in G, \forall t \in T_f \tag{5.89}$$

$$v_{g,t} \geq \frac{v^{\downarrow} + v^{\uparrow}}{2} + v_{ub}\left(1 - \chi^{\uparrow}_{g,\pi(t)}\right),\ \forall g \in G, \forall t \in T_f \tag{5.90}$$

$$\gamma_{g,t} \leq \gamma^{\downarrow} + \gamma_{ub}\left(1 - \chi^{\uparrow}_{g,\pi(t+1)}\right),\ \forall g \in G, \forall t \in T_f \setminus \{N(n_f + 1)\} \tag{5.91}$$

$$v^{+}_{i\ g,t} \geq v_{lb}\psi^{+}_{i\ g,t},\ \forall g \in G, \forall t \in T_f, \forall i \in \{x, y\} \tag{5.92}$$

$$v^{+}_{i\ g,t} \leq v_{ub}\psi^{+}_{i\ g,t},\ \forall g \in G, \forall t \in T_f, \forall i \in \{x, y\} \tag{5.93}$$

$$v^{+}_{i\ g,t} - v_{g,t} \geq v_{ub}(\psi^{+}_{i\ g,t} - 1),\ \forall g \in G, \forall t \in T_f, \forall i \in \{x, y\} \tag{5.94}$$

$$v^{+}_{i\ g,t} - v_{g,t} \leq v_{lb}(\psi^{+}_{i\ g,t} - 1),\ \forall g \in G, \forall t \in T_f, \forall i \in \{x, y\} \tag{5.95}$$

$$v^{-}_{i\ g,t} \geq v_{lb}\psi^{-}_{i\ g,t},\ \forall g \in G, \forall t \in T_f, \forall i \in \{x, y\} \tag{5.96}$$

$$v^{-}_{i\ g,t} \leq v_{ub}\psi^{-}_{i\ g,t},\ \forall g \in G, \forall t \in T_f, \forall i \in \{x, y\} \tag{5.97}$$

$$v^{-}_{i\ g,t} - v_{g,t} \geq v_{ub}(\psi^{-}_{i\ g,t} - 1),\ \forall g \in G, \forall t \in T_f, \forall i \in \{x, y\} \tag{5.98}$$

$$v^{-}_{i\ g,t} - v_{g,t} \leq v_{lb}(\psi^{-}_{i\ g,t} - 1),\ \forall g \in G, \forall t \in T_f, \forall i \in \{x, y\} \tag{5.99}$$

$$x_{g,t+1} \leq x_{g,t} + \eta_f(v^{+}_{x\ g,t} - v^{-}_{x\ g,t} + \beta h_{g,t}) + \varepsilon_{x g,t}, \forall g \in G, \forall t \in T_f \setminus \{N(n_f + 1)\} \tag{5.100}$$

$$x_{g,t+1} \geq x_{g,t} + \eta_f(v^{+}_{x\ g,t} - v^{-}_{x\ g,t} + \beta h_{g,t}) - \varepsilon_{x g,t}, \forall g \in G, \forall t \in T_f \setminus \{N(n_f + 1)\} \tag{5.101}$$

$$y_{g,t+1} \leq y_{g,t} + \eta_f(v^{+}_{y\ g,t} - v^{-}_{y\ g,t}) + \varepsilon_{y g,t},\ \forall g \in G, \forall t \in T_f \setminus \{N(n_f + 1)\} \tag{5.102}$$

$$y_{g,t+1} \geq y_{g,t} + \eta_f(v^{+}_{y\ g,t} - v^{-}_{y\ g,t}) - \varepsilon_{y g,t},\ \forall g \in G, \forall t \in T_f \setminus \{N(n_f + 1)\} \tag{5.103}$$

$$v_{h g,t} \leq v^{\uparrow}\gamma^{\uparrow} + v^{\uparrow}(\gamma_{g,t} - \gamma^{\uparrow}) + \gamma^{\uparrow}(v_{g,t} - v^{\uparrow}) + v_{ub}\left(1 - \chi^{\uparrow}_{g,\pi(t)}\right), \forall g \in G, \forall t \in T_f \tag{5.104}$$

$$v_{h g,t} \geq v^{\uparrow}\gamma^{\uparrow} + v^{\uparrow}(\gamma_{g,t} - \gamma^{\uparrow}) + \gamma^{\uparrow}(v_{g,t} - v^{\uparrow}) - v_{ub}\left(1 - \chi^{\uparrow}_{g,\pi(t)}\right), \forall g \in G, \forall t \in T_f \tag{5.105}$$

$$v_{hg,t} \leq v^{\downarrow}\gamma^{\downarrow} + v^{\downarrow}(\gamma_{g,t} - \gamma^{\downarrow}) + \gamma^{\downarrow}(v_{g,t} - v^{\downarrow}) + v_{ub}\chi^{\uparrow}_{g,\pi(t)}, \ \forall g \in G, \forall t \in T_f \tag{5.106}$$

$$v_{hg,t} \geq v^{\downarrow}\gamma^{\downarrow} + v^{\downarrow}(\gamma_{g,t} - \gamma^{\downarrow}) + \gamma^{\downarrow}(v_{g,t} - v^{\downarrow}) - v_{ub}\chi^{\uparrow}_{g,\pi(t)}, \ \forall g \in G, \forall t \in T_f \tag{5.107}$$

$$h_{g,t+1} \leq h_{g,t} + \eta_f v_{hg,t} + \varepsilon_{hg,t}, \ \forall g \in G, \forall t \in T_f \setminus \{N(n_f+1)\} \tag{5.108}$$

$$h_{g,t+1} \leq h_{g,t} + \eta_f v_{hg,t} - \varepsilon_{hg,t}, \ \forall g \in G, \forall t \in T_f \setminus \{N(n_f+1)\} \tag{5.109}$$

$$\mathfrak{z}_{g,t} \leq C_v^{\uparrow}\left({v^{\uparrow}}^2 + 2v^{\uparrow}(v_{g,t} - v^{\uparrow})\right) - g\gamma_{g,t} + v_{ub}\left(1 - \chi^{\uparrow}_{g,\pi(t)}\right), \ \forall g \in G, \forall t \in T_f \tag{5.110}$$

$$\mathfrak{z}_{g,t} \geq C_v^{\uparrow}\left({v^{\uparrow}}^2 + 2v^{\uparrow}(v_{g,t} - v^{\uparrow})\right) - g\gamma_{g,t} - v_{ub}\left(1 - \chi^{\uparrow}_{g,\pi(t)}\right), \ \forall g \in G, \forall t \in T_f \tag{5.111}$$

$$\mathfrak{z}_{g,t} \leq C_v^{\downarrow}\left({v^{\downarrow}}^2 + 2v^{\downarrow}(v_{g,t} - v^{\downarrow})\right) - g\gamma_{g,t} + v_{ub}\chi^{\uparrow}_{g,\pi(t)}, \ \forall g \in G, \forall t \in T_f \tag{5.112}$$

$$\mathfrak{z}_{g,t} \geq C_v^{\downarrow}\left({v^{\downarrow}}^2 + 2v^{\downarrow}(v_{g,t} - v^{\downarrow})\right) - g\gamma_{g,t} - v_{ub}\chi^{\uparrow}_{g,\pi(t)}, \ \forall g \in G, \forall t \in T_f \tag{5.113}$$

$$v_{g,t+1} \leq v_{g,t} + \eta_f \mathfrak{z}_{g,t} + \varepsilon_{vg,t}, \ \forall g \in G, \forall t \in T_f \setminus \{N(n_f+1)\} \tag{5.114}$$

$$v_{g,t+1} \geq v_{g,t} + \eta_f \mathfrak{z}_{g,t} - \varepsilon_{vg,t}, \ \forall g \in G, \forall t \in T_f \setminus \{N(n_f+1)\} \tag{5.115}$$

$$\mathbf{y}_{g,0} = \mathbf{y}_0, \ \forall g \in G \tag{5.116}$$

$$\mathbf{y}_{lb} \leq \mathbf{y}_{g,t} \leq \mathbf{y}_{ub}, \ \forall g \in G, \forall t \in T_f \tag{5.117}$$

$$\boldsymbol{\varepsilon_{g,t}} \leq \boldsymbol{\epsilon_g}, \ \forall g \in G, \forall t \in T_f \tag{5.118}$$

$$(\mathbf{y}_{g,t} - \mathbf{y}_{g,t-1}) \leq \boldsymbol{\lambda_{g,t}}, \ \forall g \in G, \forall t \in T_f \setminus \{0\} \tag{5.119}$$

$$(\mathbf{y}_{g,t-1} - \mathbf{y}_{g,t}) \leq \boldsymbol{\lambda_{g,t}}, \ \forall g \in G, \forall t \in T_f \setminus \{0\} \tag{5.120}$$

$$\boldsymbol{\lambda_{g,t}} \leq \boldsymbol{\Lambda_g}, \ \forall g \in G, \forall t \in T \tag{5.121}$$

$$a_{g,i,t} \in \{0,1\}, \ \forall g \in G, \forall i \in W, \forall t \in T_c \tag{5.122}$$

$$b_{g,l,t} \in \{0,1\}, \ \forall g \in G, \forall l \in L, \forall t \in T_c \tag{5.123}$$

$$d^j_{g,i,t} \in \mathbb{R}_+, \ \forall g \in G, \forall i \in W \cup L, \forall j \in \{x,y\}, \forall t \in T_c \tag{5.124}$$

$$r^j_{g,i,t} \in \mathbb{R}_+, \ \forall g \in G, \forall i \in W \cup L, \forall j \in \{x,y,h\}, \forall t \in T_c \tag{5.125}$$

$$\psi^j_{i\,g,t} \in [0,1], \ \forall g \in G, \forall t \in T_f, \forall i \in \{x,y\}, \forall j \in \{+,-\} \tag{5.126}$$

$$\psi^{max}_{g,t} \in [0,1], \ \forall g \in G, \forall t \in T_f \tag{5.127}$$

$$\phi_{ig,t}, \ \phi^{max}_{g,t} \in \{0,1\}, \ \forall g \in G, \forall t \in T_f \tag{5.128}$$

$$\xi_{g,t}, \mathfrak{z}_{g,t} \in \mathbb{R}_+, \ \forall g \in G, \forall t \in T_f \tag{5.129}$$

$$\chi^{\uparrow}_{g,t} \in \{0,1\}, \ \forall g \in G, \forall t \in T_c \tag{5.130}$$

$$v^j_{i\,g,t} \in [0, v_{ub}], \ \forall g \in G, \forall t \in T_f, \forall i \in \{x,y\}, \forall j \in \{+,-\} \tag{5.131}$$

$$v_{hg,t} \in [-v_{ub}, v_{ub}], \ \forall g \in G, \forall t \in T_f \tag{5.132}$$

$$\mathbf{y}_{g,t} \in \mathbb{R}^6, \ \forall g \in G, \forall t \in T_f \tag{5.133}$$

$$\boldsymbol{\varepsilon}_{g,t}, \boldsymbol{\epsilon}_g, \boldsymbol{\lambda}_{g,t}, \boldsymbol{\Lambda}_g \in \mathbb{R}^6_+, \ \forall g \in G, \forall t \in T_f, \tag{5.134}$$

where $C_v^{\uparrow} = -\frac{1}{2m}\rho S_w\left(C_{D_0} + k_A {C_L^{\uparrow}}^2\right)$ and analogously $C_v^{\downarrow} = -\frac{1}{2m}\rho S_w\left(C_{D_0} + k_A {C_L^{\downarrow}}^2\right)$, also recalling that $\pi\colon T_f \longrightarrow T_c$ is the map from the finer time partition to the coarse one.

It is worth recalling here that, objective (5.54) requires to minimise the time of flight as well as the scaled deviations from the approximated gliders' dynamic, and it also includes scaled penalty terms improving the stability of states variables. At the same time, linear constraints (5.55)-(5.65) ensure the correct visit of the waypoints, always respecting camera's and glider dynamic limitations, while linear constraints (5.66)-(5.74) ensure a correct glider landing. Then, as previously introduced during the chapter, constraints (5.75)-(5.121) ensure that each glider respects the approximated flight's dynamic. More precisely, constraints (5.75)-(5.88) define the horizontal direction of flight and constraints (5.89)-(5.91) determine the velocity regime of each glider. At the same time, constraints (5.92)-(5.103) help identifying a correct horizontal velocity and consequent positional changes and similarly, constraints (5.104)-(5.109) describe the vertical movement. Finally, while constraints (5.110)-(5.115) determine the acceleration magnitude and the consequent velocity changes, constraints (5.116)-(5.121) ensure that initial position, variable bounds and penalties' ranges are respected. The rest of the constraints are used to define variables domains.

5.2 Integration within the GRTOP

Having defined a model for finding initial insights and trajectories for the GRTOP, we examine here the integration of its solution into our well-established model's framework for the GRTOP with prior information described in Chapter 4. By incorporating the so-found initial trajectory into the existing model, we consolidate the reliability of the final solution and we unleash the potential of solving the GRTOP without the necessity of some form of prior knowledge.

5.2.1 Linear Interpolation and Nominal Trajectory Definition

After having constructed an initial trajectory by solving model (5.54)-(5.134), our next step involves the integration of such a trajectory within the existing GRTOP formulation (4.63)-(4.102). To do so, we start by noting that in the well-established Model I we tend to use a finer time partition, due to the fact that we require more precision and the presence of binary variables is more limited, making the problem ”easier” to solve. Thus, our next effort involves the implementation of a linear interpolation technique to synchronize the initial trajectory found via Model 0 within the finer time partition deployed for the existing GRTOP model. Let then $\{\mathbf{y}_t\}_{t\in T_f} = (x_t, y_t, h_t, v, \gamma_t, \phi_t)_{t\in T_f}$ be the state variables of the initial trajectory of some glider obtained as a solution of problem (5.54)-(5.134), with $T_f = \{0, \dots, N_f\}$ be the set of fine time steps used and $\mathcal{T}_f = \{\tau_0, \dots, \tau_{N_f}\}$ the ordered set of the corresponding times. Consider now a finer set of time steps $T'_f = \{0, \dots, N'_f\}$, with $N'_f \geq N_f$, and $\mathcal{T}'_f = \{\tau'_0, \dots, \tau'_{N'_f}\} \supseteq \mathcal{T}_f$, with $\tau_0 = \tau'_0$ and $\tau_{N_f} = \tau'_{N'_f}$, the ordered set of times corresponding to the finer partition.

Indicating with $|T_f|$ and $|T'_f|$ the respective cardinality of the two sets, we then construct a map

$$\begin{aligned} \Lambda : \quad & \mathbb{R}^{6|T_f|} \to \mathbb{R}^{6|T'_f|} \\ & \{\mathbf{y}_t\}_{t\in T_f} \mapsto \{\mathbf{y}'_s\}_{s\in T'_f} \end{aligned}$$

that extend the first initial trajectory into a corresponding trajectory defined on the finer time partition T'_f and determined by

$$\mathbf{y}'_s = \begin{cases} \mathbf{y}_t, & \text{if } \tau'_s = \tau_t \text{ for some } t \in T_f \\ \mathbf{y}_t + \frac{\tau'_s - \tau_t}{\tau_{t+1} - \tau_t}(\mathbf{y}_{t+1} - \mathbf{y}_t), & \text{if } \tau_t \leq \tau'_s \leq \tau_{t+1} \text{ for some } t \in T_f \setminus \{N_f\} \end{cases} . \tag{5.135}$$

At this point we have defined the states variables describing the refined initial trajectory, but what can the control variables be? Recalling that we did not use them to find the initial solution, we must follow a different road. Therefore, we decide to utilise the most up to date information as well as equilibrium flight conditions to determine a complete initial trajectory $\{\mathbf{y}^o_s, \mathbf{u}^o_s\}_{s\in T'_f}$ that can be deployed in the existing GRTOP Model I. More precisely, at time step s, we consider the glider flying with a bank angle of $\mu^o_s = 0$, and to determine the nominal lift coefficient C^o_{Ls} we decide to invert equation (3.26). To be precise, to ensure that the so found nominal lift coefficient is within the admissible flight envelope, we specify it as

$$C^o_{Ls} = \begin{cases} 0.1, & \text{if } \frac{2mg}{v'^2_s \rho S_w} \leq 0.1 \\ \frac{2mg}{v'^2_s \rho S_w}, & \text{if } 0.1 \leq \frac{2mg}{v'^2_s \rho S_w} \leq 1.17 \\ 1.17, & \text{if } 1.17 \leq \frac{2mg}{v'^2_s \rho S_w} \end{cases} . \tag{5.136}$$

With these definitions, we can finally write the ultimate initial nominal trajectory $(\mathbf{y}^o_s, \mathbf{u}^o_s)$ at time step s as

$$x^o_s = x'_s \tag{5.137}$$

$$y^o_s = y'_s \tag{5.138}$$

$$h^o_s = h'_s \tag{5.139}$$

$$v^o_s = v'_s \tag{5.140}$$

$$\gamma^o_s = \gamma'_s \tag{5.141}$$

$$\varphi^o_s = \varphi'_s \tag{5.142}$$

$$C^o_{Ls} = \begin{cases} 0.1, & \text{if } \frac{2mg}{v'^2_s \rho S_w} \leq 0.1 \\ \frac{2mg}{v'^2_s \rho S_w}, & \text{if } 0.1 \leq \frac{2mg}{v'^2_s \rho S_w} \leq 1.17 \\ 1.17, & \text{if } 1.17 \leq \frac{2mg}{v'^2_s \rho S_w} \end{cases} \tag{5.143}$$

$$\mu^o_s = 0. \tag{5.144}$$

It is worth mentioning here that we decided v'_s to be the most influential factor to determine the initial lift coefficient at time step s, also because the flight path angle γ' was found ignoring constraints stemming from dynamic's equation (3.5).

Interpolating and constructing the initial trajectory as just described, we are now ready to deploy Model I and ultimately solve the GRTOP in absence of prior knowledge on gliders' flight. In particular, in the following section we define an integrated algorithm to solve the GRTOP that not only refines our initial trajectory and facilitates a smoother transition between the different time scales, but also provides a more detailed view of the underlying dynamics within the model, consequently leading to more reliable solutions.

5.2.2 A Two-Step Solution Algorithm

In the endeavor of solving the GRTOP and finding flyable trajectories, we acknowledge that while our initial trajectory served as a valuable starting point, we can reach a more accurate representation of the glider's dynamic by incorporating the full set of linearised EoMs. Thus, enhanced and reliable trajectories solving the GRTOP can be found by adopting a *Two Step Solution Algorithm*, as defined in algorithm 1.

The idea of the framework for solving the GRTOP in absence of prior information is to proceed by steps. In the first step, we define a possibly coarse initial time partition $\mathcal{T}_f$ with n_f number of fine time steps as in Step 1-(i) and, in Step 1-(ii), we solve a simpler approximated problem as the one described by Model 0, so constructing an initial trajectory defined on $\mathcal{T}_f$ for each glider. Consequently, in the second step, we improve the initial trajectory solution solving a more precise, though still linear, GRTOP model, encoded in Model I. Here we improve the precision by defining a finer time partition $\mathcal{T}'_f$, Step 2-(i), and extending the initial trajectories to it via linear interpolation, Step 2-(ii) and Step 2-(iii). Dynamic constraints are then linearised around this enlarged initial trajectory, Step 2-(iv), and the GRTOP defined by Model I is solved, Step 2-(v), providing the final (sub) optimal solution. Deploying the Two Step Solution Algorithm, it is therefore possible to find solutions to the GRTOP that are sound, reliable and that can guarantee (sub) optimal path planning for gliders' missions. In the rest of the chapter, we provide first methodologies to assess the quality of the resulting solution and then an example of mission, helping the reader in getting valuable understanding of algorithm1 and its effectiveness.

Algorithm 1 - *Two Step Algorithm for the GRTOP*

STEP 1:

(i) - Select a time partition $\mathcal{T}_f = \{\tau_0, \dots, \tau_{N_f}\}$ with n_f fine time steps

(ii) - For each glider, construct initial state variables' trajectories, $\{\mathbf{y}_t\}_{t \in T_f}$, by solving Model 0 (5.54)-(5.134)

STEP 2:

(i) - Select a finer time partition $\mathcal{T}'_f = \{\tau'_0, \dots, \tau'_{N'_f}\} \supseteq \mathcal{T}_f$ with n'_f fine time steps

(ii) - For each glider, linearly interpolate the corresponding initial states $\{\mathbf{y}_t\}_{t \in T_f}$ to find $\{\mathbf{y'}_s\}_{s \in T'_f}$ following definition (5.135)

(iii) - For each glider, construct complete initial trajectories $\{\mathbf{y}^o_s, \mathbf{u}^o_s\}_{s \in T'_f}$ by exploiting equations (5.137)-(5.144)

(iv) - Linearise the EoMs and dynamic constraints around $\{\mathbf{y}^o_s, \mathbf{u}^o_s\}_{s \in T'_f}$ as described in Chapter 3 and equation (3.19)

(v) - Given the newly linearised dynamic's constraints, solve Model I (4.63)-(4.102)

5.3 Error Analysis and Validation Methods

One key question to be addressed in order to validate our model and Two Step Algorithm, is to understand how to measure the error committed while deploying the linearisation approach and the subsequent approximations. To do so, two ways can be followed:

1. solve the NLP-based GRTOP and compare the trajectories found;
2. define (local) error measures to evaluate the difference with respect to the nonlinear dynamic in a single or limited amount of time steps.

Following the first approach, the problem is solved by deploying algorithm 1. Consequently, when possible, the NLP-based GRTOP is solved, utilising the two-step algorithm solution as initial guess, and possibly penalising the deviation from the initial trajectory. Finally, it is possible to measure:

- the ability to find an optimal solution to the NLP, i.e. if the solver is able to return an optimal solution; as well as
- the relative distance between the NLP-based solution and the trajectory found deploying algorithm 1.

More precisely, given an optimal solution to the GRTOP $\{\mathbf{y}_t^*, \mathbf{u}_t^*\}$ stemming from algorithm 1, we can observe if we are able to find an optimal solution stemming from the NLP problem with initial guesses and penalties, and consequently, given a solution of the latter $\{\hat{\mathbf{y}}_t, \hat{\mathbf{u}}_t\}$, we can measure the *relative distance*, $\boldsymbol{\delta}_t$, between the trajectories at time step t as

$$\boldsymbol{\delta}_t = \frac{|(\mathbf{y}_t^*, \mathbf{u}_t^*) - (\hat{\mathbf{y}}_t, \hat{\mathbf{u}}_t)|}{\eta \mathscr{D}}, \tag{5.145}$$

where $\mathscr{D} = (\mathscr{D}(x), \mathscr{D}(y), \mathscr{D}(h), \mathscr{D}(v), \mathscr{D}(\gamma), \mathscr{D}(\varphi), \mathscr{D}(C_L), \mathscr{D}(\mu))$ represents the range (or domain) of the corresponding variables, η is the time partition length, and the differences and ratio have to be intended variable wise. To express a global performance measure, it is also possible to define an *average relative distance*, $\overline{\boldsymbol{\delta}}$, as the average of the relative distances across all the time steps $\boldsymbol{\delta}_t$.

On the other hand, computing NLP-based solutions for multiple gliders' flights in a complete mission setting is not always possible or practical. Nonetheless, it is possible to define measures of trajectories' quality by observing first the penalty terms $\boldsymbol{\varepsilon}$ in the objective function of Model I, indicating the deviations from the linearised dynamic, and second by defining a local error measure to capture "errors" committed in the linearisation process, by comparison with the nonlinear dynamic. More precisely, we can introduce the following

Definition 5.2. Given an optimal solution to the GRTOP $\{\mathbf{y}_t^*, \mathbf{u}_t^*\}_{t \in T_f'}$ obtained via algorithm 1, we can define the *Local Error Measure* at time step $t \geq 1$ as

$$\mathscr{E}_t = |\mathbf{y}_t^* - \hat{\mathbf{y}}_t|,$$

where

$$\hat{\mathbf{y}}_t = \mathbf{y}_{t-1}^* + \int_{\tau_{t-1}}^{\tau_t} \mathbf{f}(\mathbf{y}^*(s), \mathbf{u}^*(s)) ds \simeq \mathbf{y}_{t-1}^* + \eta \mathbf{f}(\mathbf{y}_{t-1}^*, \mathbf{u}_{t-1}^*),$$

with $\eta = (\tau_t - \tau_{t-1})$, $\mathbf{f}$ representing the EoMs (3.1)-(3.6), and the last approximation obtained by deploying an Euler scheme of integration, imposing that $\mathbf{y}^*(s) = \mathbf{y}_{t-1}^*$ and $\mathbf{u}(s) = \mathbf{u}_{t-1}^*$ $\forall s \in [t-1, t]$.

In other words, the local error measure represents the local deviation of the optimal solution $\{(\mathbf{y}_t^*, \mathbf{u}_t^*)\}_{t \in T}$ from the trajectory obtained by applying the true dynamic equations in the time period between τ_{t-1} and τ_t, keeping variables constant.

To have a better comparative measure, it is often convenient to consider a relative error measure. More precisely, the local error measure might be related to the length of the time step as well as to the magnitude of the corresponding variables. This leads to the following

Definition 5.3. Given an optimal solution to the GRTOP $\{\mathbf{y}_t^*, \mathbf{u}_t^*\}_{t\in T_f'}$ obtained via algorithm 1, we can define the *Relative Local Error Measure* at time step $t \geq 1$ as

$$\mathscr{R}_t = \frac{\mathscr{E}_t}{\eta \mathscr{D}_S},$$

where $\mathscr{D}_S = (\mathscr{D}(x), \mathscr{D}(y), \mathscr{D}(h), \mathscr{D}(v), \mathscr{D}(\gamma), \mathscr{D}(\phi))$ represents the range (or domain) of the corresponding state variables, and the ratio has to be intended variable wise.

In this way, we are able to get a more comprehensible and significant measure of the error we commit by making linear approximations and deploying the Two Step Algorithm for the GRTOP, while evaluating the validity of our methodology.

5.4 Flight Examples

In this section we present flight examples showing the efficacy of our Two Step Algorithm and demonstrating how the just presented techniques can help in finding flyable paths in absence of prior information. Once more, models and solutions are implemented utilising the AMPL modelling language and solved using Gurobi 10.0.1 on a Viglen desktop, Intel(R) Core(TM) $\mathrm{i}5-7500$ CPU with 3.40GHz and a RAM of 8 GB, running under Windows 10. As illustrative examples, we require a glider to start at the launching point $(x_0, y_0, h_0) = (1967, 430, 1010)$ and initial state

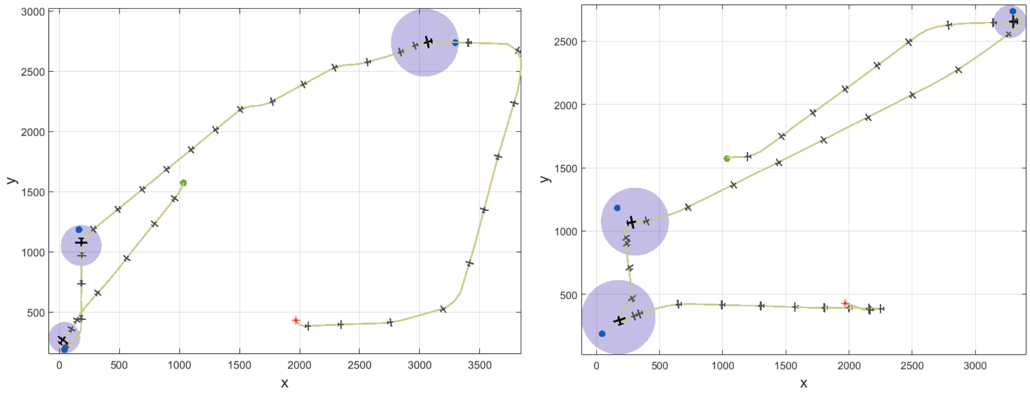
$$\mathbf{y}_0 = (x_0, y_0, h_0, v_0, \gamma_0, \varphi_0) = (1967, 430, 1010, 7.1, -0.07, \pi),$$

to visit the three waypoints at sparse locations and then land in the predefined landing site in $(x_f, y_f, h_f) = (1034, 1572, 0)$, in scenarios where no wind is blowing as well as in ones where it blows towards east (x direction) at its average intensity in UK ($\beta = 0.025$). Applying the two step algorithm 1 with $N = 18$ coarse time steps in Model 0, $N = 54$ coarse time steps in Model I and $n_f = n_f' = 2$, and utilising the other model parameters as determined by Table 5.1, within the flight envelope described by Table 3.2, we can obtain the following results.

As reported in Figure 5.5, we are able to find reasonable trajectories and complete the mission in both scenarios. Nonetheless, it is possible to notice that the resulting trajectories can be very different. In fact, in presence of wind ($\beta = 0.025$), the glider decides to start visiting a different waypoint to prevent to be pushed too much towards east and so that the greatest distance, i.e. going from western waypoints to the eastern one, can be travelled in favour of wind. In addition, although the glider is flying to the landing point against the wind, it is doing so at a lower altitude, so that the wind speed, $U = \beta h$, is less pronounced. In fact, after the last waypoint visit, it tries to

Parameter	Value Step 1	Value Step 2	Unit
N	18	54	-
n_f	2	2	-
w_1	0.6	0.6	-
w_2	0.7	0.7	-
w_3	0.9	0.9	-
w_4	0.1	1.1	-
w_5	0.07	1.1	-
w_6	-	1.1	-
κ_x, κ_y	100	100	m
κ_h	15	100	m
κ_v	$\frac{3}{5}(v_{ub} - v_{lb})$	$\frac{3}{5}(v_{ub} - v_{lb})$	m/s
κ_γ	$\frac{3}{5}\gamma_{ub}$	$\frac{3}{5}\gamma_{ub}$	radians
κ_φ	$\frac{1}{6}\pi$	$\frac{1}{6}\pi$	radians
$\hat{\gamma}$	0.03	0.03	radians
$\hat{\mu}$	0.03	0.03	radians
$\hat{v}$	14.75	14.75	m/s

TABLE 5.1: Parameters of the models for the Two Step Algorithm.

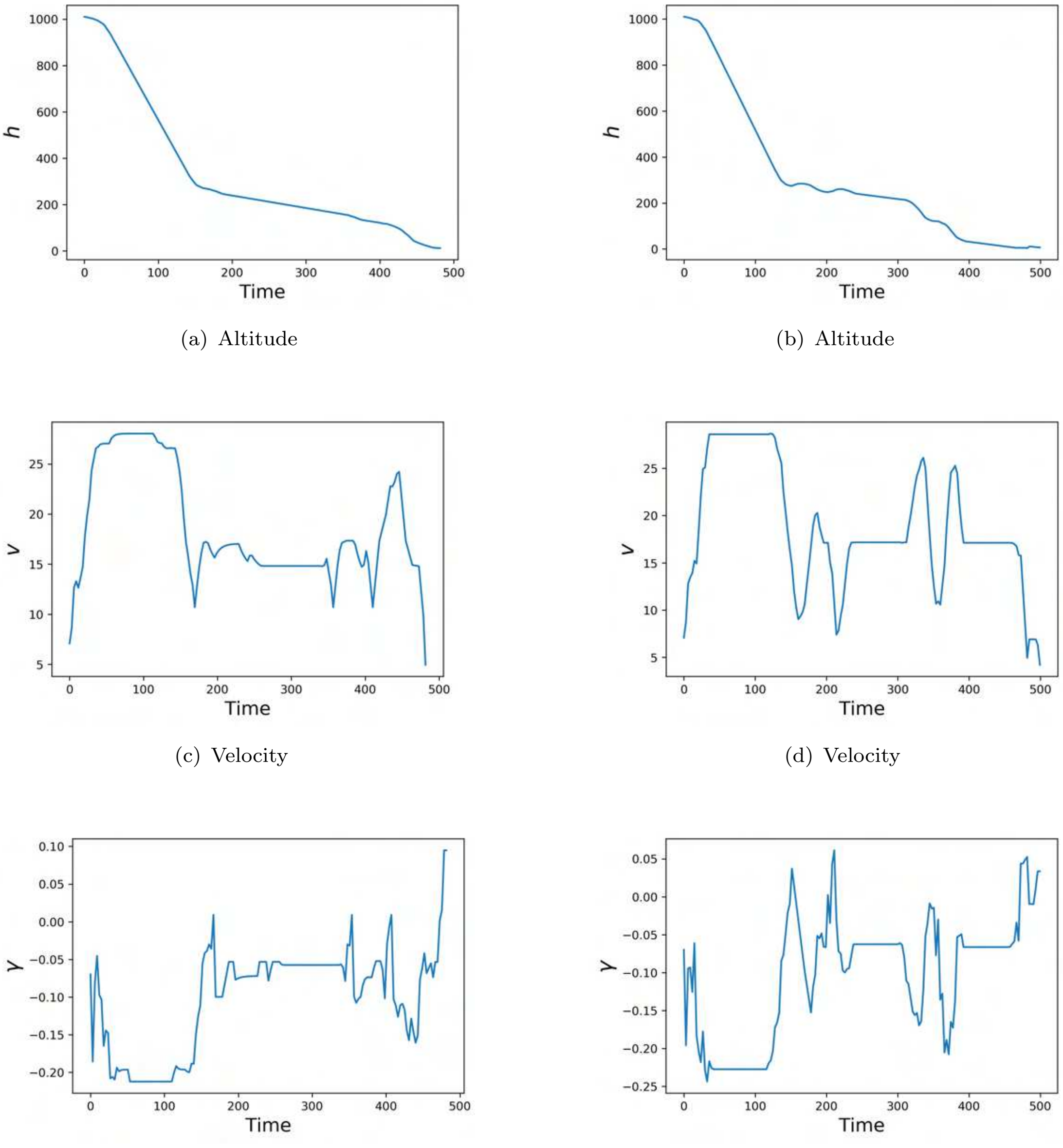


(a) $\beta = 0$

(b) $\beta = 0.025$

FIGURE 5.5: Example of glider's optimal trajectories in absence of wind (a) and in the presence of wind (b).

reduce its altitude faster to diminish the resistance of the wind, as it is possible to observe by comparing Figure 5.6(a) and Figure 5.6(b). Analogously, looking at Figure 5.6 and Figure 5.7 where we report the obtained state and control variables profiles, we can observe that in presence of wind, the glider is flying against the latter at the beginning of the flight, while trying to loose altitude (steeper values of γ are reached) and increase its velocity, in order to overcome wind effects. Nevertheless, in both situations, the glider tries to fly as fast as possible between waypoints' visits, while it reduces its speed for landing. Moreover, time steps where the glider is visiting a waypoint are clearly recognisable by the fact that, by regulating its lift coefficient, it adjust its velocity and flight path angle according to the requirements of flying flat at a moderate speed. Conditions on the bank angle are also respected, although less evident given that the glider is trying to fly straight most of the time. At the same time, turns are



(a) Altitude

(b) Altitude

(c) Velocity

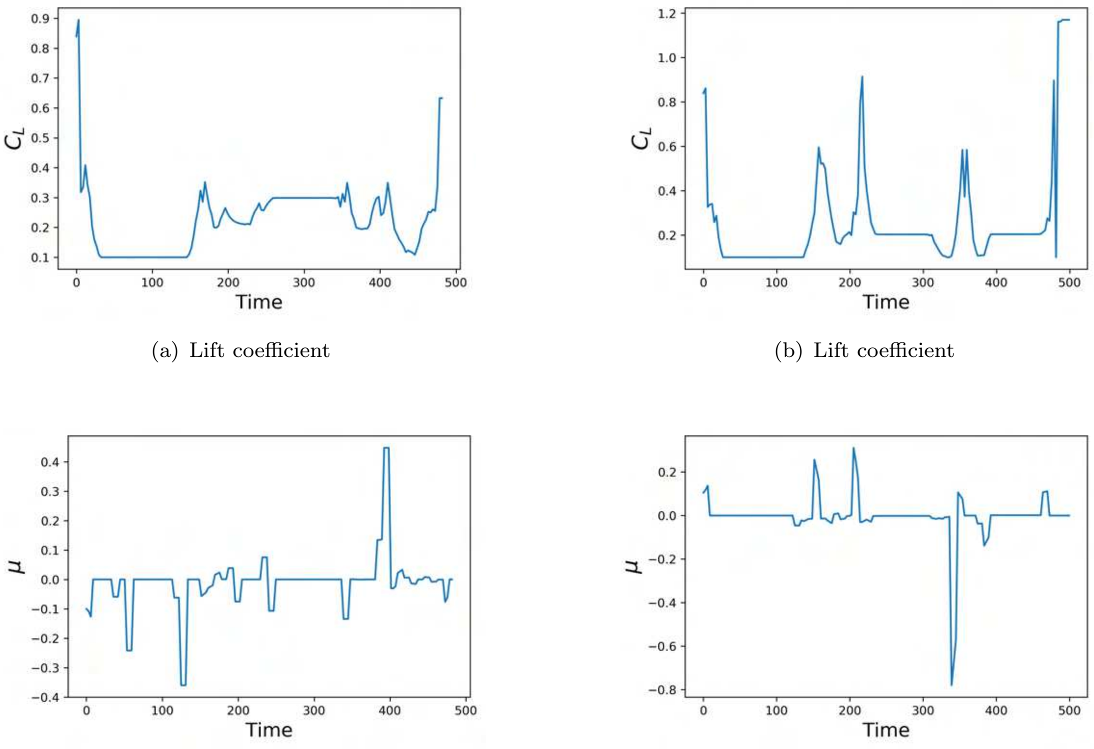
(d) Velocity

(e) Flight path angle

(f) Flight path angle

Figure 5.6: Optimal state variables profiles in absence of wind (left) and in the presence of wind (right).

also completed within reasonable states and controls, with the steepest one executed with an average bank angle of around 0.6 radians, corresponding to roughly 35^o, in absence of wind. Once again, it is possible to observe that with blowing wind the glider counterbalance its effects by executing a steeper turn at around 0.8 radians, roughly 45^o, when it is close to the last waypoint visit, before directing itself toward the landing point. Finally, we note that this bank angle attitude is not kept for extended periods of time and it falls well within glider's flight envelope. It is worth recalling here that whenever milder bank angles are required, it is always possible to increase the limitations, or penalties, for changes of heading angle.



(a) Lift coefficient

(b) Lift coefficient

(c) Bank angle

(d) Bank angle

FIGURE 5.7: Optimal control variables profiles in absence of wind (left) and in the presence of wind (right).

With this in mind, we are able to examine more numerical results, also validating our two step algorithmic approach by a thorough error analysis and showing its efficacy.

Chapter 6

Numerical Results

In this chapter, we present some numerical results obtained by solving the GRTOP problem utilising the Two Step Algorithm presented in the previous chapter. We start by examining trajectory optimisation problems, also comparing the solutions with trajectories found solving the NLP counterparts. In this way, we show the validity of our approach in finding reliable and feasible trajectories to accomplish the mission at hand. We also show the robustness of the model by showing the sensitivity of the solutions with respect to some of the model's parameters. It is possible to see that, even though trajectories can be highly affected by the latter, our approach is able to return flyable trajectories most of the times. Then we continue our analyses by investigating cases of multiple gliders flights, also introducing new methods for the evaluation of trajectories' goodness when a direct comparison with their NLP counterpart is not possible or not practical. Finally, we investigate algorithm 1 scalability, first by comparing the performances of different commercial solvers, and second by varying the size of the inputs to the GRTOP problem.

Unless stated otherwise, along the chapter, models and solutions are again implemented utilising the AMPL modelling language and solved using Gurobi 10.0.1 as a solver on a Viglen desktop, Intel(R) Core(TM) $i5-7500$ CPU with 3.40GHz and a RAM of 8 GB, running under Windows 10. We also remind the reader that we decided to use a two time level partition, as described in Chapter 4, with parameters stated in Table 6.1 or specified in due course, also respecting glider's flight envelope as per Table 3.2. With this in mind, we can then proceed by examining the trajectories found within several flight conditions.

Parameter	Value Step 1	Value Step 2	Unit
N	18	54	-
n_f	2	2	-
w_1	0.6	0.6	-
w_2	0.7	0.7	-
w_3	0.9	0.9	-
w_4	0.1	1.1	-
w_5	0.07	1.1	-
w_6	-	1.1	-
κ_x, κ_y	100	100	m
κ_h	15	100	m
κ_v	$\frac{3}{5}(v_{ub} - v_{lb})$	$\frac{3}{5}(v_{ub} - v_{lb})$	m/s
κ_γ	$\frac{3}{5}\gamma_{ub}$	$\frac{3}{5}\gamma_{ub}$	radians
κ_φ	$\frac{1}{6}\pi$	$\frac{1}{6}\pi$	radians
$\hat{\gamma}$	0.03	0.03	radians
$\hat{\mu}$	0.03	0.03	radians
$\hat{v}$	14.75	14.75	m/s

Table 6.1: Parameters of the models used along the chapter, unless otherwise stated.

6.1 Glider Flight Validation

In this section we aim at validating our Two Step Algorithm reliability and efficacy. Through illustrative examples, we explore how the just presented techniques can help in finding flyable paths, possibly close to optimal, in absence of further prior information. To do so, we examine the trajectory optimisation problem illustrated in Chapter 3 comparing then the results obtained by deploying algorithm 1, with the ones obtained by solving the NLP formulation of the problem utilising SNOPT as nonlinear solver. More precisely, we consider a glider with initial state

$$\mathbf{y}_0 = (x_0, y_0, h_0, v_0, \gamma_0, \varphi_0) = (0, 0, 261, 7, -0.0274, 0),$$

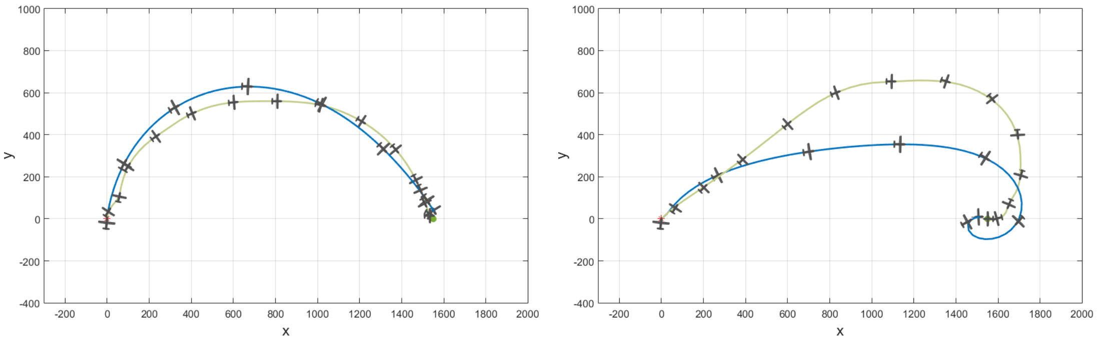
that is required to reach a final location

$$(x_f, y_f, h_f) = (1550, 0, 0),$$

with a moderate speed $v_f \leq 5.5$ m/s and within a time span of $[t_0, t_f] = [0, 290]$ seconds, also considering two different scenarios: one in absence of wind ($\beta = 0$) and one with typical wind conditions with the wind strength coefficient approximating the average wind gradient profile for the UK ($\beta = 0.025$). Solving the problem using algorithm 1, we find a trajectory $\{\mathbf{y}_t^*, \mathbf{u}_t^*\}$, that we can compare to the solution obtained by solving its NLP counterpart defined by problem (3.14)-(3.18). To be reasonably comparable though, this time the latter is solved by deploying the same uniform time partition used within Model I of algorithm 1 until the landing time, so that decisions are made within the same time frame. Moreover, we provide to the nonlinear solver $\{\mathbf{y}_t^*, \mathbf{u}_t^*\}$ as an initial trajectory guess. Results are reported in Table 6.2, where we compare the flight and computational times, in Figure 6.1, where we illustrate the found optimal trajectories, and in Figure 6.2 and Figure 6.3, where a comparison between state and control variables profiles is also investigated.

Type	**Flight Time (s)**		**Total CPU (s)**	
	No Wind	Wind	No Wind	Wind
Algorithm 1	114.10	114.10	17.53	17.22
NLP Solution	108.4	108.4	2.98	4.21

Table 6.2: Flight and computational times comparison between the solution attained via algorithm 1 and the one attained solving the corresponding NLP with initial guess.

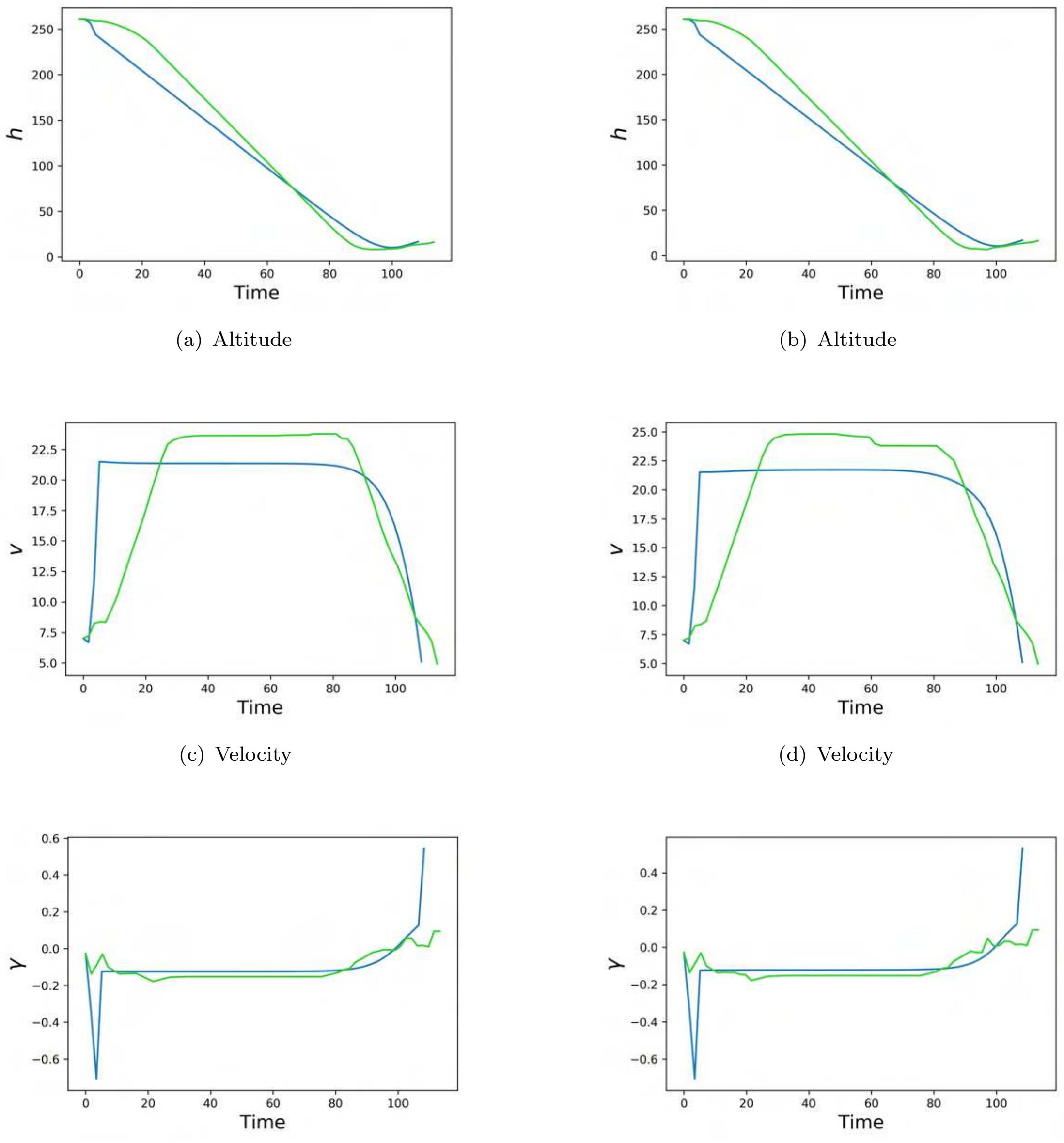


(a) In absence of wind

(b) In presence of wind

Figure 6.1: Comparison of the optimal trajectories found via algorithm 1 (in green) and by solving the corresponding NLP with initial guess (in blue), in scenarios with (a) and without (b) the presence of wind.

As before, in both cases the wind's presence forces the glider to fly more towards east (x direction) and then approach the landing point from a different direction. However, we can notice that trajectories can be quite different. In particular, in both solutions the glider tends to descent at a regular pace adjusting its velocity and attitude from the beginning, keeping it as constant as possible during the flight and then changing it for the landing period. Although, NLP solutions tend to be more responsive, enabling the glider to adjust its state and decision to face difficult situations like at the beginning of the flight when low speed oblige the glider to dramatically reduce its flight path angle.



(a) Altitude

(b) Altitude

(c) Velocity

(d) Velocity

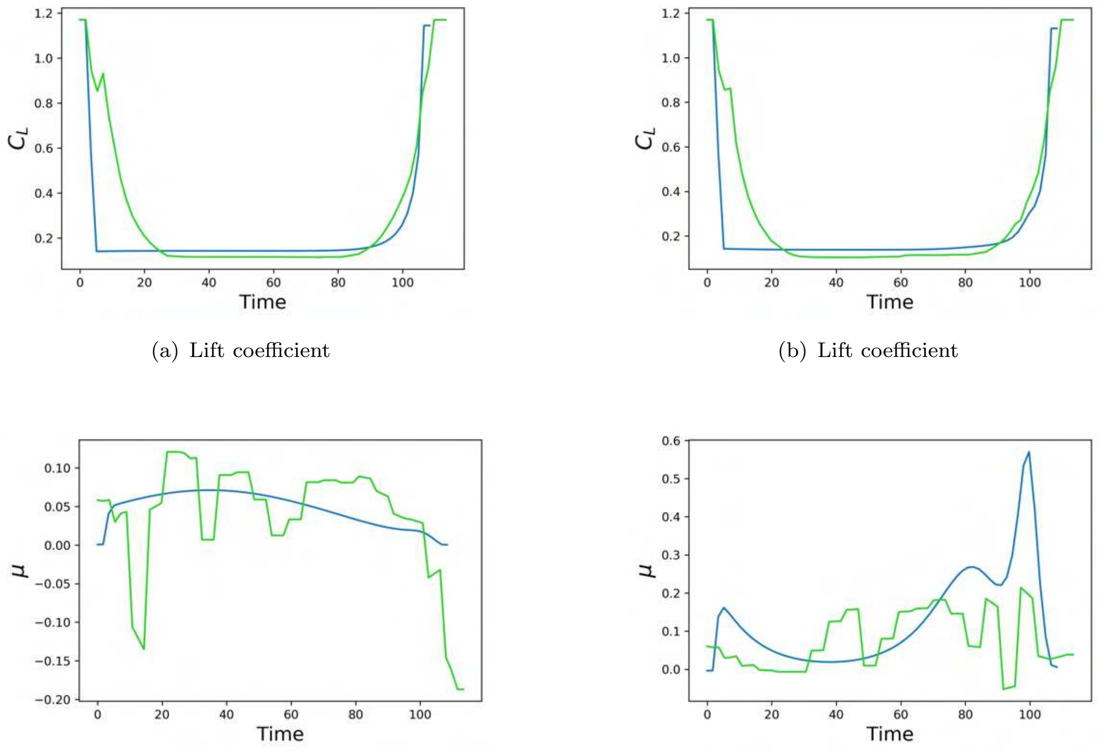
(e) Flight path angle

(f) Flight path angle

FIGURE 6.2: Comparison of the state variables profiles found via algorithm 1 (in green) and by solving the corresponding NLP with initial guess (in blue), in scenarios with (a) and without (b) the presence of wind.

In addition, as one can expect, NLP solutions tends to be smoother, see especially the different choices of bank angle μ.

As mentioned in Chapter 3, in general it is difficult to guarantee the optimality of the solution, given that we are in presence of nonconvex problems with the resulting trajectories and flight times being affected by the chosen time partition as well as model parameters and initial guesses, see Zhao (2004), Betts (2009), Rao (2010) and Rodrigues et al. (2014). Nonetheless, we can ask if the (sub) optimal trajectories stemming from algorithm 1 are actually flyable. If so, we have found another possible path from the



(a) Lift coefficient

(b) Lift coefficient

(c) Bank angle

(d) Bank angle

FIGURE 6.3: Comparison of the control variables profiles found via algorithm 1 (in green) and by solving the corresponding NLP with initial guess (in blue), in scenarios with (a) and without (b) the presence of wind.

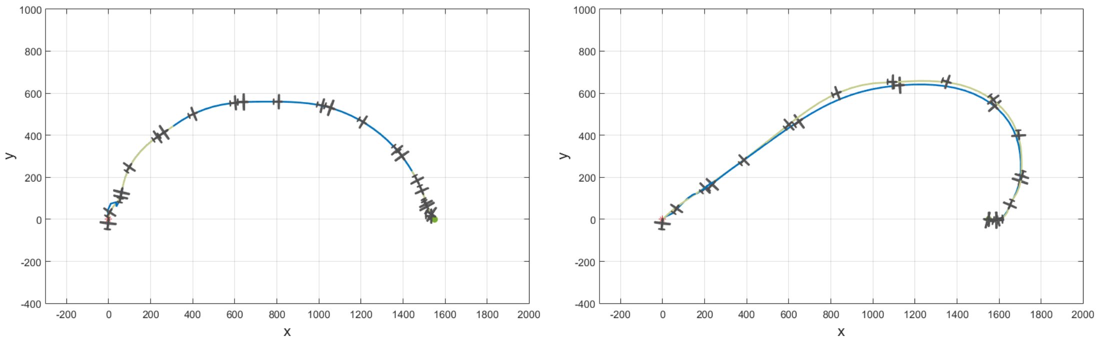
initial state to the final one that can serve our purposes. To answer such a question, in the NLP model and solution, we decide to penalise deviations from position variables of the initial trajectory guess by inserting the following terms in the objective function

$$
\begin{aligned}
p_x &= \sum_{k=0}^{N_L} (x_k - x_k^*)^2 \\
p_y &= \sum_{k=0}^{N_L} (y_k - y_k^*)^2 \\
p_h &= \sum_{k=0}^{N_L} (h_k - h_k^*)^2.
\end{aligned}
$$

As hoped, we can see that we are able to find trajectories very close to the one found with algorithm 1, both in presence and absence of wind, see Table 6.3 and Figure 6.4. Once again, control variables, and consequently altitude, velocity and flight path angle profiles are examined. A part from an initial adjustment due to the fact that the glider starts with an attitude close to stalling conditions, which are difficult to prevent by our linear approximations, this time states and control variables result to be much closer to the one stemming from algorithm 1 as it is possible to see in Figure 6.5 and Figure 6.6.

Type	**Flight Time (s)**		**Total CPU (s)**	
	No Wind	Wind	No Wind	Wind
Algorithm 1	114.10	114.10	17.53	17.22
NLP Solution	114.10	114.10	0.57	2.2

Table 6.3: Flight and computational times comparison between the solution attained via algorithm 1 and via the corresponding NLP with initial guess and penalties.



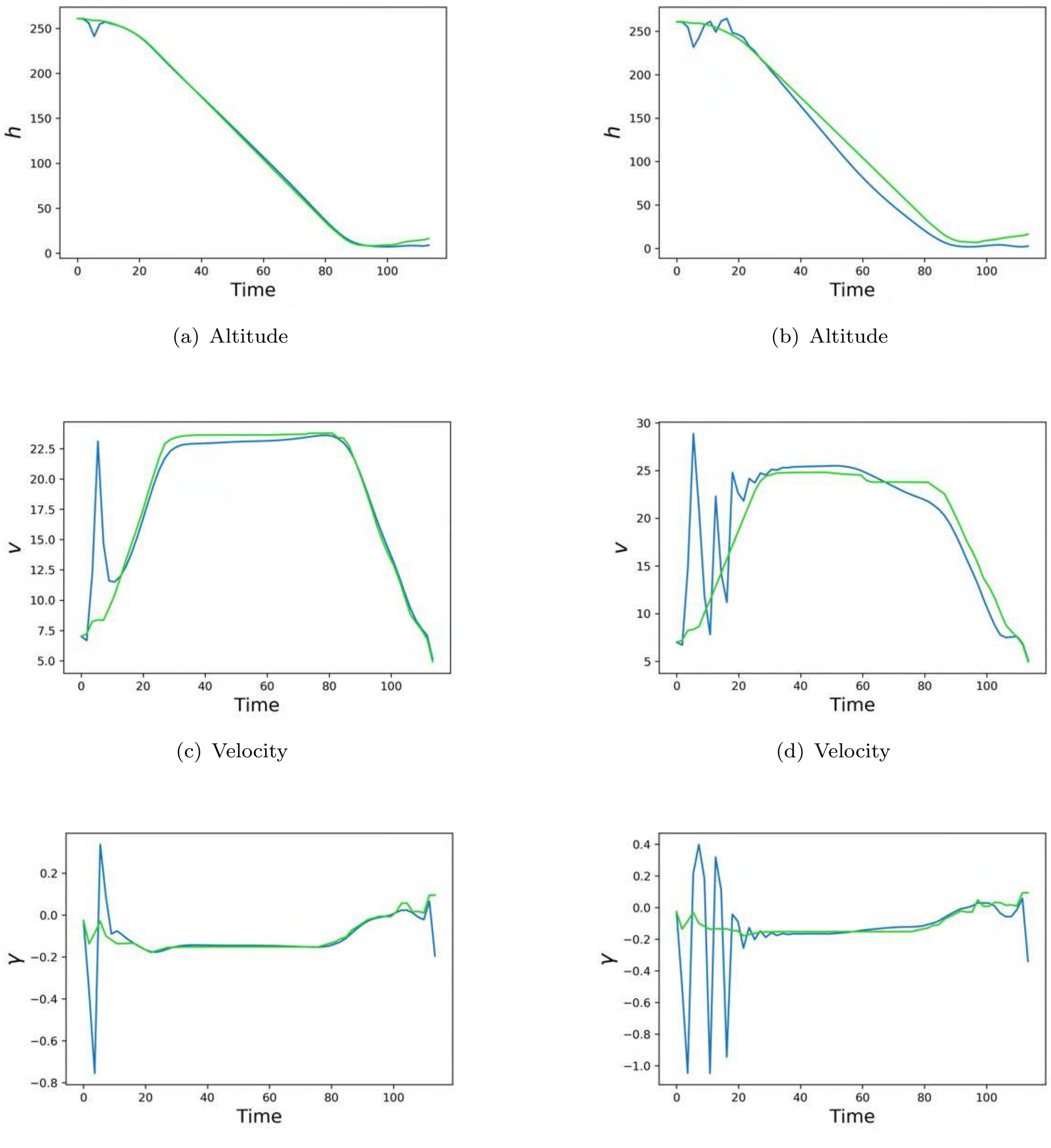
(a) $\beta = 0$

(b) $\beta = 0.025$

Figure 6.4: Comparison of the optimal trajectories found via algorithm 1 (in green) and by solving the corresponding NLP (in blue) with initial guess and penalties, in scenarios with (b) and without (a) the presence of wind.

With this example, we show our ability to find flyable trajectories as well as some model limitations, where the linearised dynamic and initial trajectory approximations are not able to perfectly handle extreme flight conditions. Nonetheless, it is possible to show that with different initial conditions closer to an equilibrium point, such adjustments are not required. For instance, if the glider starts with an attitude close to the preferred equilibrium conditions of Remark 3.2, we obtained similar trajectories between algorithm 1 solution and the NLP-based solution, with variables' profiles comparison shown in Figure 6.7 and Figure 6.8. Here it is possible to observe that such initial adjustments are indeed no more required and even more similar decisions as well as variable trajectories are established. It is worth recalling here that thanks to the approach "ASTRA High Altitude Balloon Flight Planner" proposed by Sobester et al. (2013) and available at Zapponi (2013), it is possible to lift gliders to optimal releasing points, possibly close to equilibrium flight conditions, allowing to take them to advantageous positions that ensure a more efficient trajectory planning, also avoiding the necessity of managing extreme flight conditions.

As a further validation of our solutions and dynamic approximations, we decide to repeat the process and generate 50 instances where one glider is required to go from an initial state A to a final landing position B. The position of A is a point $(0, 0, h_A)$, with h_A chosen randomly between 100m and 300m, whereas point B lays on the ground with its xy coordinates that can vary between 0 and 1500m. An initial velocity between 7.5m/s and 11m/s and orientation are also chosen randomly, instead the lift coefficient is chosen



(a) Altitude

(b) Altitude

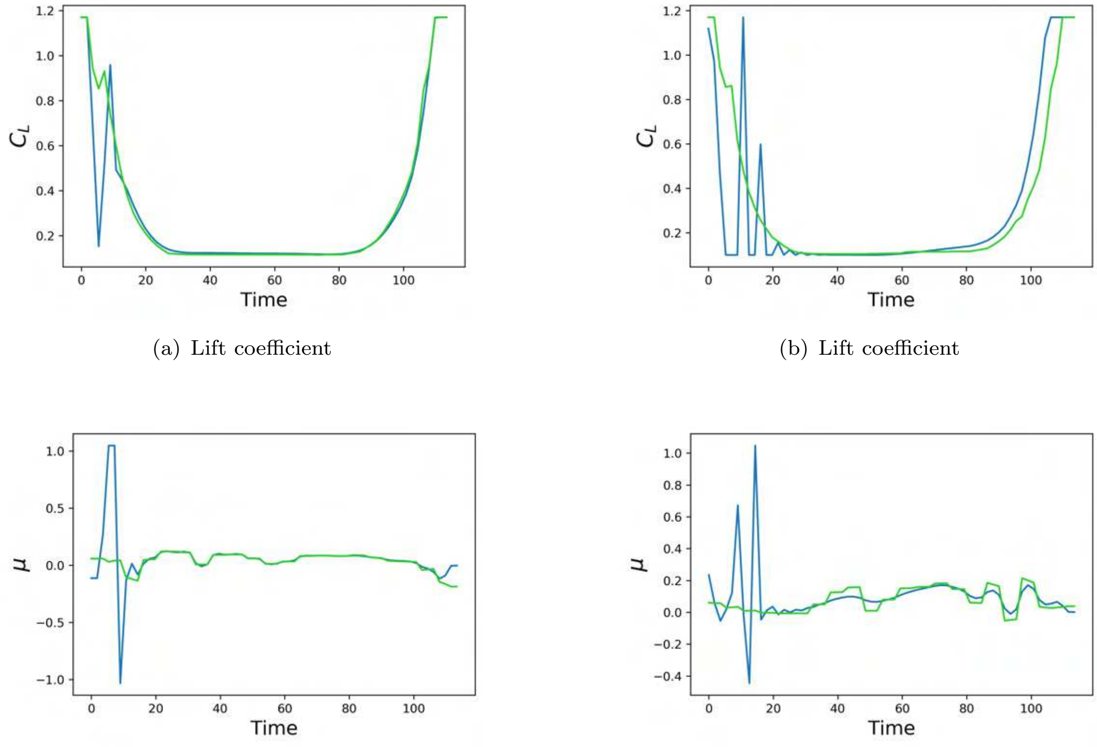
(c) Velocity

(d) Velocity

(e) Flight path angle

(f) Flight path angle

FIGURE 6.5: Comparison of state variable profiles found via algorithm 1 (in green) and by solving the corresponding NLP (in blue) with initial guess and penalties, in scenarios with (right) and without (left) the presence of wind.



(a) Lift coefficient

(b) Lift coefficient

(c) Bank angle

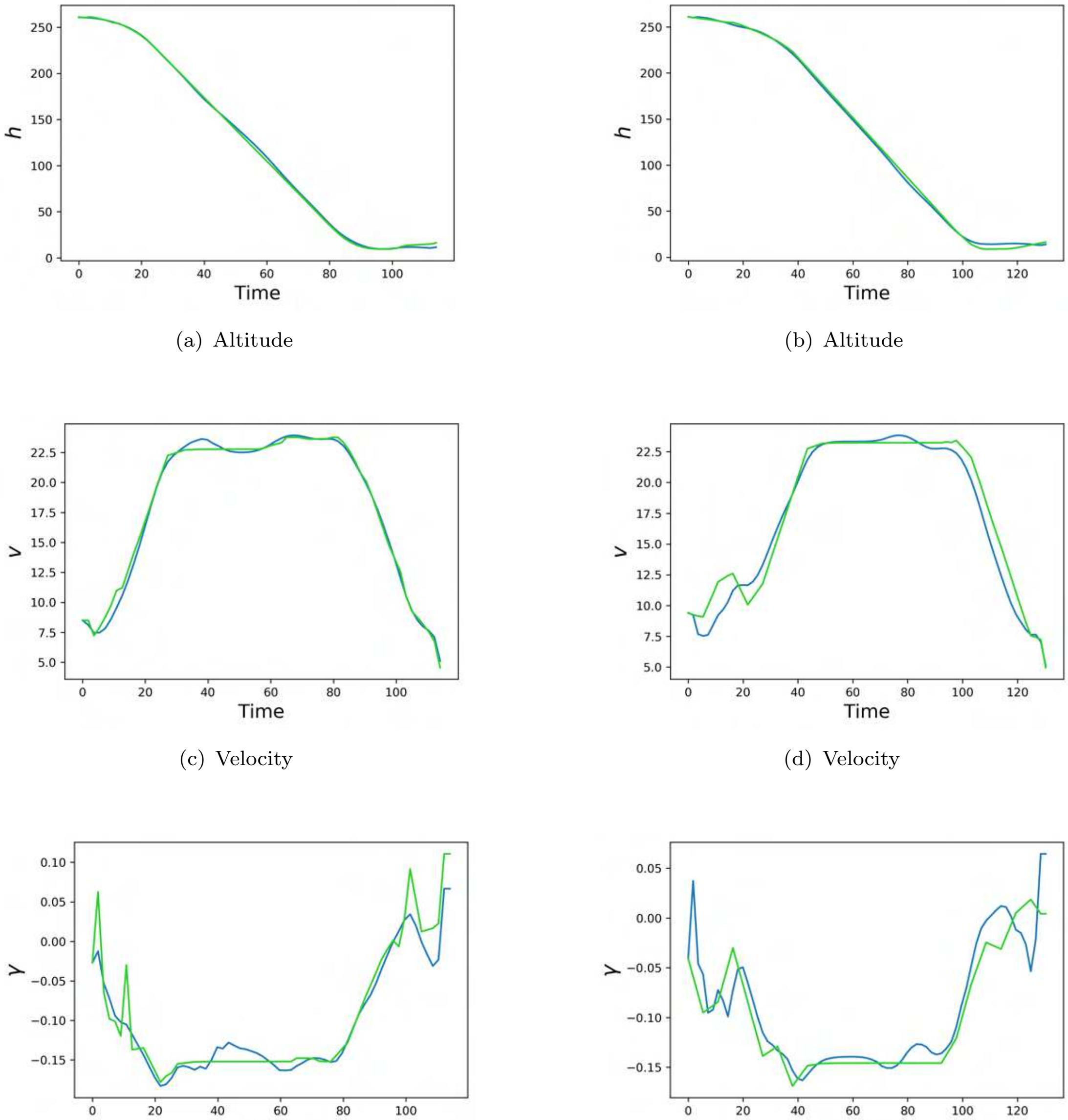
(d) Bank angle

Figure 6.6: Comparison of control variable profiles found via algorithm 1 (in green) and by solving the corresponding NLP (in blue) with initial guess and penalties, in scenarios with (right) and without (left) the presence of wind.

by inverting equation (3.26). Then, to validate our model and Two Step Algorithm, we solve first the problem deploying algorithm 1 and consequently we solve the NLP with the so-found solution as initial guess, in scenarios of absent wind as well as the wind blowing at the average UK speed. We again penalise the deviation from the initial trajectory and we measure

- the frequency with which we are able to find an optimal solution to the NLP; and
- the average relative distance between the NLP solution and the trajectory found deploying algorithm 1.

More precisely, given an optimal solution to the GRTOP $\{\mathbf{y}_t^*, \mathbf{u}_t^*\}$ stemming from algorithm 1, we can count the number of times in which we are able to find an optimal solution stemming from the NLP with initial guesses and penalties, and consequently, given a solution of the latter $\{\hat{\mathbf{y}}_t, \hat{\mathbf{u}}_t\}$, we can measure the *relative distance*, $\boldsymbol{\delta}_t$, between the trajectories at time step t as defined in equation (5.145). Then, we also measure the *average relative distance*, $\overline{\boldsymbol{\delta}}$, as the average of $\boldsymbol{\delta}_t$ by time steps and instances. Results are then reported in Table 6.4, where it is possible to see that we are always able to solve the NLP problem to optimality and within an average relative distance of less than 1.5% for



(a) Altitude

(b) Altitude

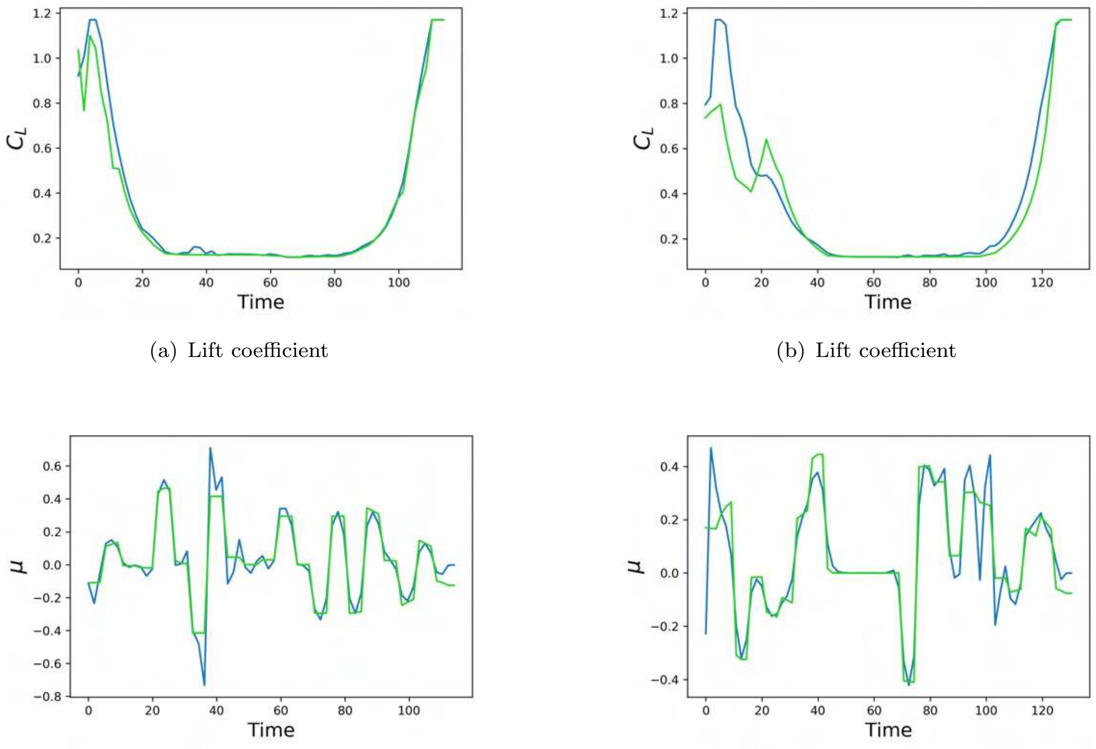
(c) Velocity

(d) Velocity

(e) Flight path angle

(f) Flight path angle

Figure 6.7: Comparison of state variable profiles found via algorithm 1 (in green) and by solving the corresponding NLP (in blue) with initial guess and penalties, when the glider starts with flight conditions close to an equilibrium point, in scenarios with (right) and without (left) the presence of wind.



(a) Lift coefficient

(b) Lift coefficient

(c) Bank angle

(d) Bank angle

Figure 6.8: Comparison of control variable profiles found via algorithm 1 (in green) and by solving the corresponding NLP (in blue) with initial guess and penalties, when the glider starts with flight conditions close to an equilibrium point, in scenarios with (right) and without (left) the presence of wind.

Type	**NLP Optimality**	**Average Relative Distance (%)**							
		$\overline{\delta}_x$	$\overline{\delta}_y$	$\overline{\delta}_h$	$\overline{\delta}_v$	$\overline{\delta}_\gamma$	$\overline{\delta}_\varphi$	$\overline{\delta}_{C_L}$	$\overline{\delta}_\mu$
$\beta = 0$	100%	0.1	0.1	0.4	1.4	1.1	1.0	5.0	10.6
$\beta = 0.025$	100%	0.1	0.1	0.5	1.4	1.3	1.2	5.8	10.4

Table 6.4: Frequency of initial trajectory stemming from algorithm 1 leading to an optimal solution to the corresponding NLP and average relative distance between the trajectories. Quantities are measured in scenarios of presence and absence of wind.

all the states variables, and of less than 10.5% for all the controls. Despite bank angle decisions exhibit more pronounced discrepancies of about 10%, we consider the model valid since in the majority of the cases these differences are not dramatic in terms of flight impact, i.e. glider position and attitude are not significantly affected.

Having shown our ability to find flyable trajectories for several different flight conditions, we proceed here with the investigation of the dependency of model outcomes from our models' parameters.

6.2 Sensitivity to Model Parameters

In this section, we examine the impact of some parameters on the optimal solutions, especially in terms of trajectory reliability, trajectory shape and landing times. We focus especially on the selection of time partition, penalty weights in the initial trajectory model, as well as the effect of wind strength on the final path. For the former cases, unless otherwise stated, we will consider the same trajectory optimisations flights used at the end of previous section, in absence of wind.

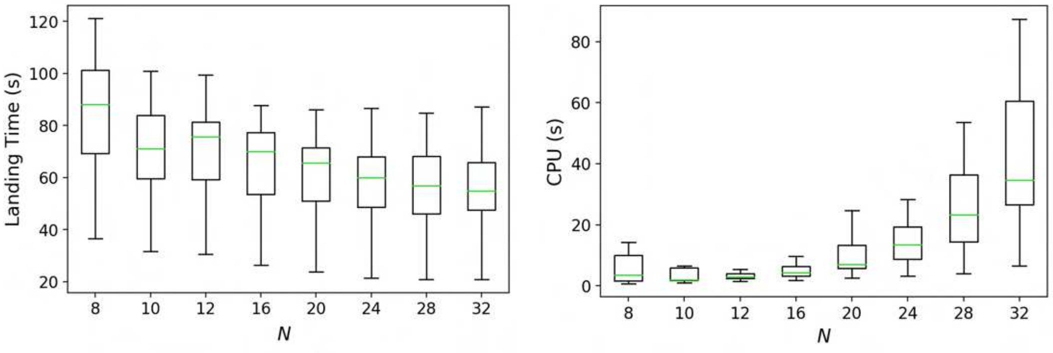
6.2.1 Time Partition Sensitivity

To test the influence of the time partition on the optimal solution, we decide to vary the coarse time steps N between 8 and 18 for the first step of algorithm 1, whereas in the second step we use triple the number of time steps, with a minimum of 33 coarse time steps. We then solve the 50 trajectory optimisation instances defined at the end of section 6.1 deploying our Two Step Algorithm and we compare the so-found trajectories with the NLP solutions with initial guess and penalties for deviating from the initial trajectory, to evaluate path feasibility. It is possible to observe in Table 6.5 that almost regardless the number of time step chosen, we are able to solve to optimality the NLP with initial guess and, looking at the relative distances, find an optimal trajectory fairly close to the initial guess. This indicates the ability of our model to produce (sub) optimal flyable trajectories.

On the other hand, in Figure 6.9 we report the results relative to the changes in landing time distribution as well as CPU usage to solve Model 0 of our algorithm (it is possible to show that similar CPU behaviour, although of different magnitude, can be observed in solving Model I). Landing and computational times can be indeed significantly affected by the number of time step used. As expected, computational times can worsen significantly as we increase the number of time steps whilst the solution quality remains under investigation: if on one side we start observing a possible improvement and convergence of landing times, with the landing time distribution that seems to start stabilising after $N = 24$, on the other side we have a significant increase of the average relative distance for decisions made about the bank angle. All the other average relative distances stay fairly stable.

N	**NLP Optimality**	**Average Relative Distance (%)**							
		$\overline{\delta}_x$	$\overline{\delta}_y$	$\overline{\delta}_h$	$\overline{\delta}_v$	$\overline{\delta}_\gamma$	$\overline{\delta}_\varphi$	$\overline{\delta}_{C_L}$	$\overline{\delta}_\mu$
8	96.7%	0.1	0.2	0.7	1.9	1.1	1.7	5.9	13.3
10	100%	0.1	0.2	0.7	2.0	1.2	1.4	5.7	12.7
12	100%	0.2	0.2	0.7	2.0	1.5	1.8	6.1	14.2
16	100%	0.1	0.1	0.6	2.2	1.5	1.0	4.6	14.9
20	100%	0.1	0.1	0.5	3.0	1.6	1.2	4.3	17.6
24	100%	0.1	0.1	0.6	2.9	1.8	1.3	4.3	20.3
28	100%	0.1	0.1	0.6	2.9	2.1	1.3	4.6	23.1
32	100%	0.1	0.2	0.7	2.8	2.4	1.7	4.7	25.1

TABLE 6.5: Sensitivity of the solution with respect to time step partition. The ability to solve the corresponding NLP problems and the average relative distance from the initial trajectory is shown as the number of coarse time steps of the first step of algorithm 1 (N) varies.

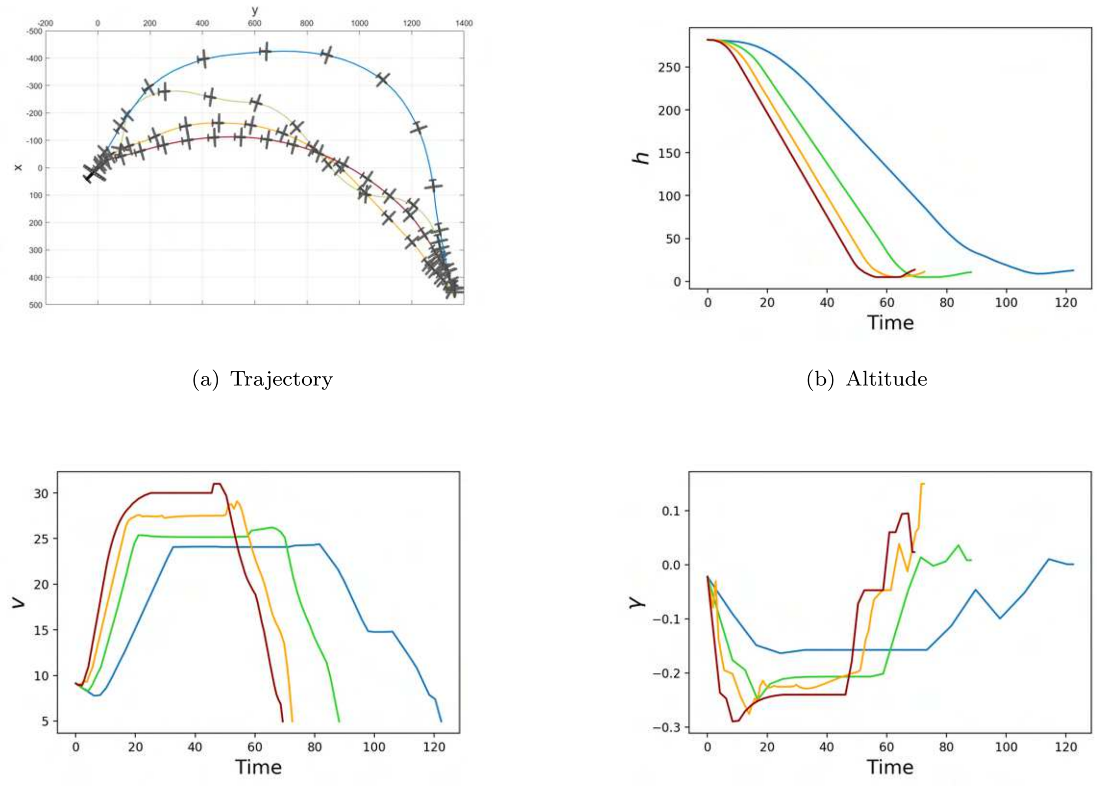


(a) Landing Time

(b) CPU(s) - Model 0

FIGURE 6.9: Distribution of landing and computational times for solving Model 0 as the number of coarse time steps for the model in the first step of algorithm 1 varies.

In addition to this, we show in Figure 6.10 the optimal trajectories and state variable profiles found solving a single instance and deploying algorithm 1 using different number of time steps. It is possible to observe that increasing the number of time steps, and consequently having the possibility to adjust the trajectory more frequently, the glider is able to better control its flight path angle and to reach higher velocities and descent rates. In this way, it is able to land earlier and complete the mission within less time and possibly with a smoother trajectory.



(a) Trajectory

(b) Altitude

(c) Velocity

(d) Flight path angle

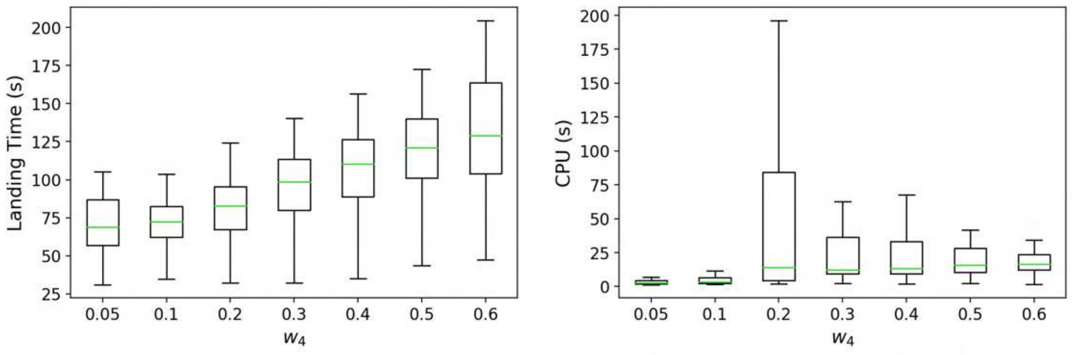
FIGURE 6.10: Trajectories and variables profiles as the number of coarse time steps for the model in the first step of algorithm 1 varies among $N = 8$ (blue), $N = 16$ (green), $N = 24$ (orange), $N = 32$ (red).

6.2.2 Weights Sensitivity

Similarly to the examination about the effects of the time partition on the solution, we want to evaluate the sensitivity of the latter with respect to the some of the models' weights. It is understandable that Model 0 and the first step of algorithm 1 have huge repercussions on the final optimal trajectory. Consequently, weights chosen in it play a vital role and one of the highest impact is due to the weight on the penalisation on state changes, w_4. Therefore, we decide to analyse consequences of changing it by making it vary between 0.05 and 0.6, and by solving the problems via our two step algorithm as well as their NLP counterparts. At this stage, we decide to utilise $N = 12$ coarse time steps, to maintain a balance between the quality of the trajectory and computational times. Analogously to what reported above, in Table 6.6 and in Figure 6.11 we represent the corresponding results where once again we can observe our ability in finding flyable trajectories for most of the instances. However, it is possible to observe higher average relative distance, lower ability in solving the corresponding NLP to optimality, and an increase in landing times as we require smaller state changes between time steps. The intuition behind the latter is that, increasing the importance of having smaller changes, the glider aims at flying at moderate velocities, limiting variations between v_0 and v_f, and therefore leading to a longer flight. It is also possible to observe that computational times can be significantly affected by the weight chosen, meaning that the solver might not know which objective must be prioritised most.

$\mathbf{w_4}$	**NLP**	**Average Relative Distance (%)**							
	Optimality	$\overline{\delta}_x$	$\overline{\delta}_y$	$\overline{\delta}_h$	$\overline{\delta}_v$	$\overline{\delta}_\gamma$	$\overline{\delta}_\varphi$	$\overline{\delta}_{C_L}$	$\overline{\delta}_\mu$
0.05	100.0%	0.1	0.1	0.6	2.1	1.5	1.2	5.4	13.0
0.1	100.0%	0.1	0.1	0.5	1.9	1.2	1.1	5.0	10.7
0.2	98.1%	0.1	0.2	0.5	1.8	1.0	1.7	5.9	13.6
0.3	100.0%	0.2	0.2	0.5	1.7	1.1	2.7	7.2	23.1
0.4	96.2%	0.2	0.2	0.5	1.6	1.1	3.3	7.4	23.7
0.5	92.3%	0.2	0.2	0.5	1.6	1.1	3.6	8.0	23.7
0.6	96.7%	0.2	0.2	0.6	1.6	1.2	3.1	6.6	19.2

Table 6.6: Sensitivity of the solution with respect to Model 0 weight w_4. The ability to solve the corresponding NLP problems and the average relative distance from the initial trajectory is shown as w_4 varies.

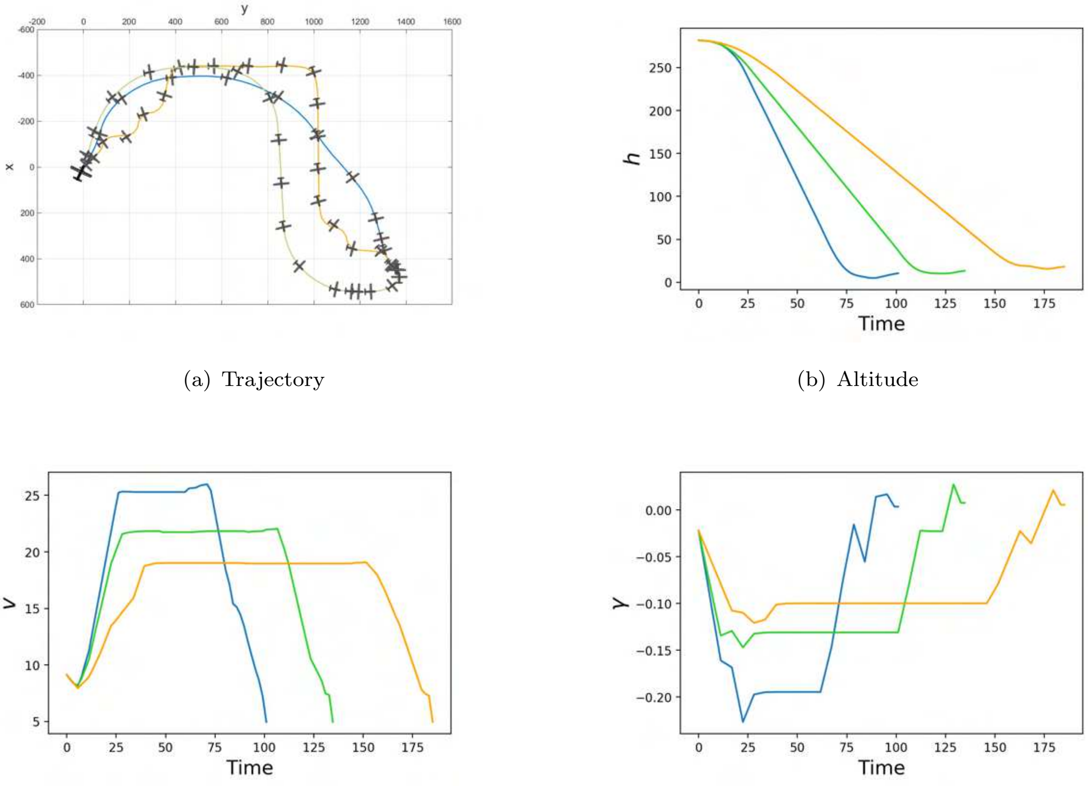


(a) Landing Time

(b) CPU(s) - Model 0

Figure 6.11: Distribution of landing and solving times as the weight relative to the penalty on state changes (w_4) varies for the first step of algorithm 1.

In addition, similarly to what previously done, in Figure 6.12 we show a comparison of the optimal trajectories and some state variable profiles found solving a single instance via algorithm 1 using different weights. It is possible to observe that a moderate weight let more freedom to the glider that in this way it is able to reach the target faster. On the other hand, higher weights penalising the changes of state lead to more rigid trajectories and prevent the glider to reach higher velocities and land faster.



(a) Trajectory

(b) Altitude

(c) Velocity

(d) Flight path angle

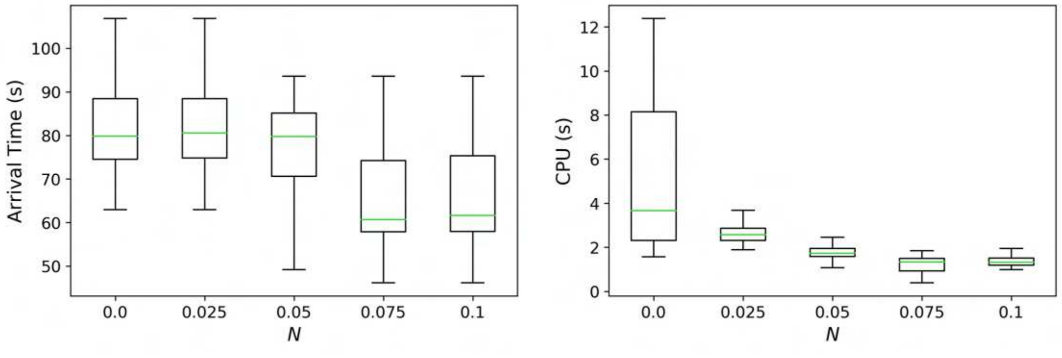
Figure 6.12: Trajectories and variables profiles as the weight of penalties on state changes varies among $w_4 = 0.1$ (blue), $w_4 = 0.3$ (green) and $w_4 = 0.5$ (orange).

6.2.3 Wind Sensitivity

Ultimately, we would like to examine the effect of the wind on the choice of trajectory. In this occasion however, given the assumption of an altitude dependent wind profile, we create 50 instance of a TO problem where a glider is required to fly at higher altitudes. More precisely, a glider is required to fly from a point $A = (0, 0, h_A)$ to a point $B = (x_B, y_B, h_B)$, where h_A is chosen randomly between 550m and 750m, whereas x_B, y_B vary between 0m and 1500m and $h_B = 450$m. Moreover, the glider starts with an initial velocity chosen randomly between 8.5m/s and 11m/s, orientation also chosen randomly within the flight envelope, while the lift coefficient is selected by inverting equation (3.26). With this in mind, keeping the same time partition with $N = 12$ coarse time steps, we solve the instance making the wind intensity β vary between 0.0 and 0.1 passing from its average value 0.025. The results are reported in Table 6.7 and in Figure 6.13.

β	NLP	Average Relative Distance (%)							
	Optimality	$\overline{\delta}_x$	$\overline{\delta}_y$	$\overline{\delta}_h$	$\overline{\delta}_v$	$\overline{\delta}_\gamma$	$\overline{\delta}_\varphi$	$\overline{\delta}_{C_L}$	$\overline{\delta}_\mu$
0.0	100.0%	0.1	0.1	0.3	2.4	1.1	1.3	5.5	7.4
0.025	100.0%	0.1	0.1	0.3	2.2	1.3	2.0	5.4	9.3
0.05	86.7%	0.5	0.3	3.0	3.4	4.0	1.9	6.2	12.0
0.075	56.7%	1.2	0.5	5.4	4.2	4.0	2.3	4.0	20.5
0.1	16.7%	2.0	0.6	8.1	4.9	3.8	2.4	3.2	20.9

TABLE 6.7: Sensitivity of the solution with respect to the wind strength β. The ability to solve the corresponding NLP problems and the average relative distance from the initial trajectory is shown as β varies.



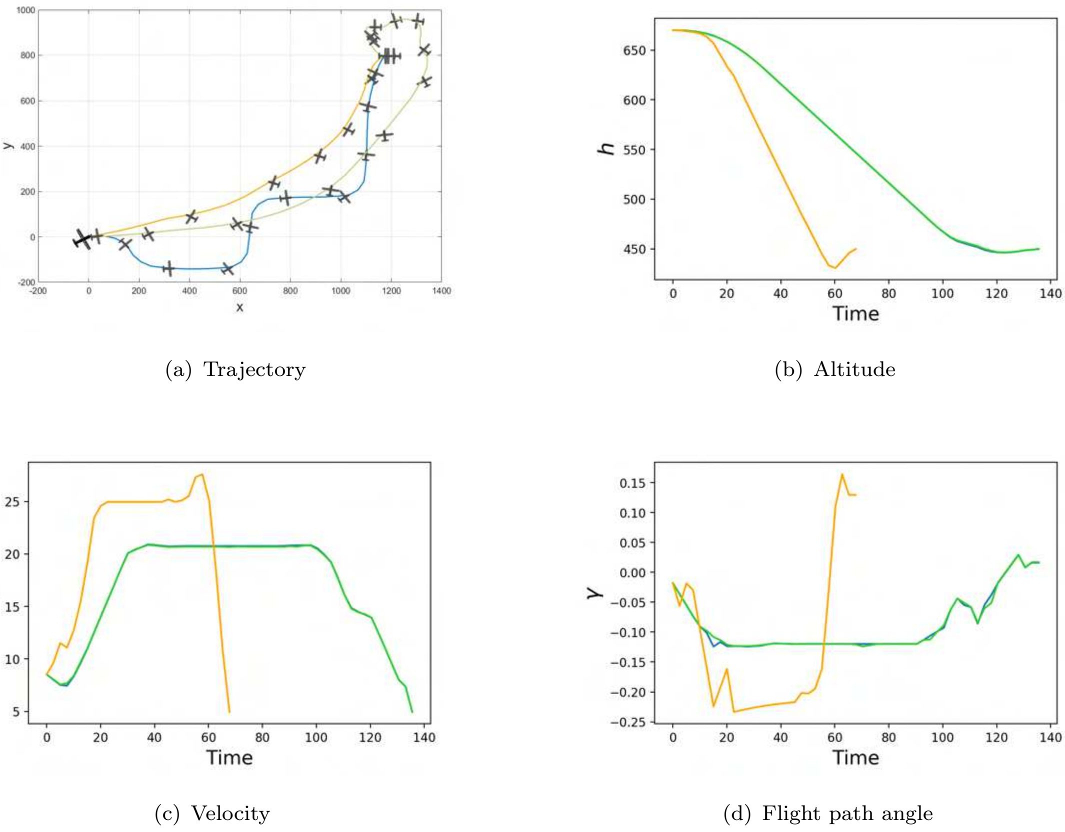
(a) Arrival Time

(b) CPU(s) - Model 0

FIGURE 6.13: Distribution of flight (arrival) and solving times as the wind strength (β) varies.

It is possible to observe that, despite the flight time and computational time are less affected by the wind (consider the different time scales of Figure 6.13 with respect to the other parameters' analyses), the quality of the solution can be strongly affected. In fact, a much stronger wind can lead to our inability to solve the NLP counterparts to optimality, also increasing the average relative distance especially for the height h, as reasonable, given the assumption on the wind profiles. However, the fact that we do not find NLP-based optimal solution, does not mean that this does not exist and that it is impossible for us to find it, but just that we are not sure that we can do it like for the other cases, since the NLP solver ends in a different status, meaning that it is not able to find an optimal solution. Anyway, extra care is required while planning missions when a strong wind is blowing.

As before, we show here examples of optimal trajectories that might help the reader to understand the wind effects on glider's dynamic. In Figure 6.14, it is possible to see that the wind plays a crucial role in determining the final trajectory. In particular, while absent or moderate wind can lead to similar variables profiles, glider positions can vary significantly with the UAV approaching the arrival point B from a different position and a different attitude. On the other hand, when the wind blows intensely, $\beta = 0.075$, the glider decides to loose altitude faster and to fly aggressively at higher speed against the wind, in order to counteract its strength and avoid being blown over the target point B.



(a) Trajectory

(b) Altitude

(c) Velocity

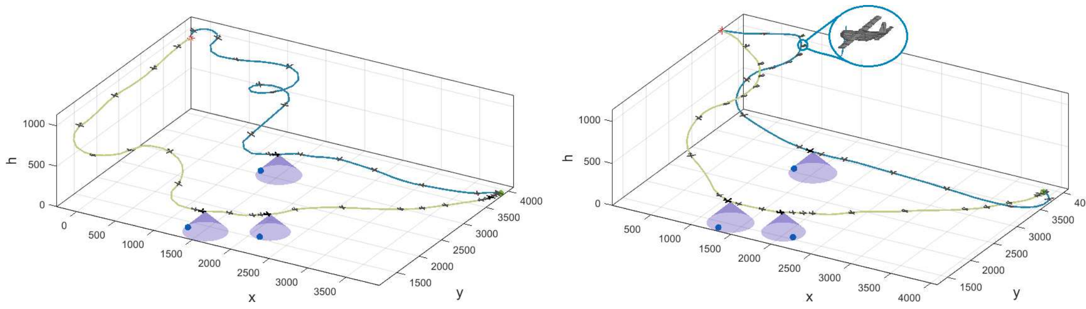
(d) Flight path angle

Figure 6.14: Trajectories and variables profiles as the wind strength varies between $\beta = 0.0$ (blue), $\beta = 0.025$ (green) and $\beta = 0.075$ (orange).

With all this in mind, we are ready to examine other flight missions involving actual waypoint vists in combination with the utilisation of multiple UAVs.

6.3 Multiple UAVs

Being able to find (sub) optimal flyable trajectories deploying our Two Step Algorithm for the GRTOP, and knowing the effect of (some of) the parameters involved as well as of the weather conditions, we can now proceed to investigate multiple gliders' flights. Similarly to what shown in the flight illustration at the end of the Chapter 5, we start by examining the flight of two gliders launched from the same site and required to visit three waypoints before landing in a predefined landing region. Deploying once again algorithm 1, with $N = 10$ and $w_4 = 0.1$, it is possible to obtain the following results in scenarios where no wind is blowing as well as in ones where it blows around its average intensity in UK.



(a) $\beta = 0$

(b) $\beta = 0.025$

FIGURE 6.15: Optimal trajectories of two gliders (blue and green) found via algorithm 1 and in scenarios of absence of wind (a) as well as in its presence (b).

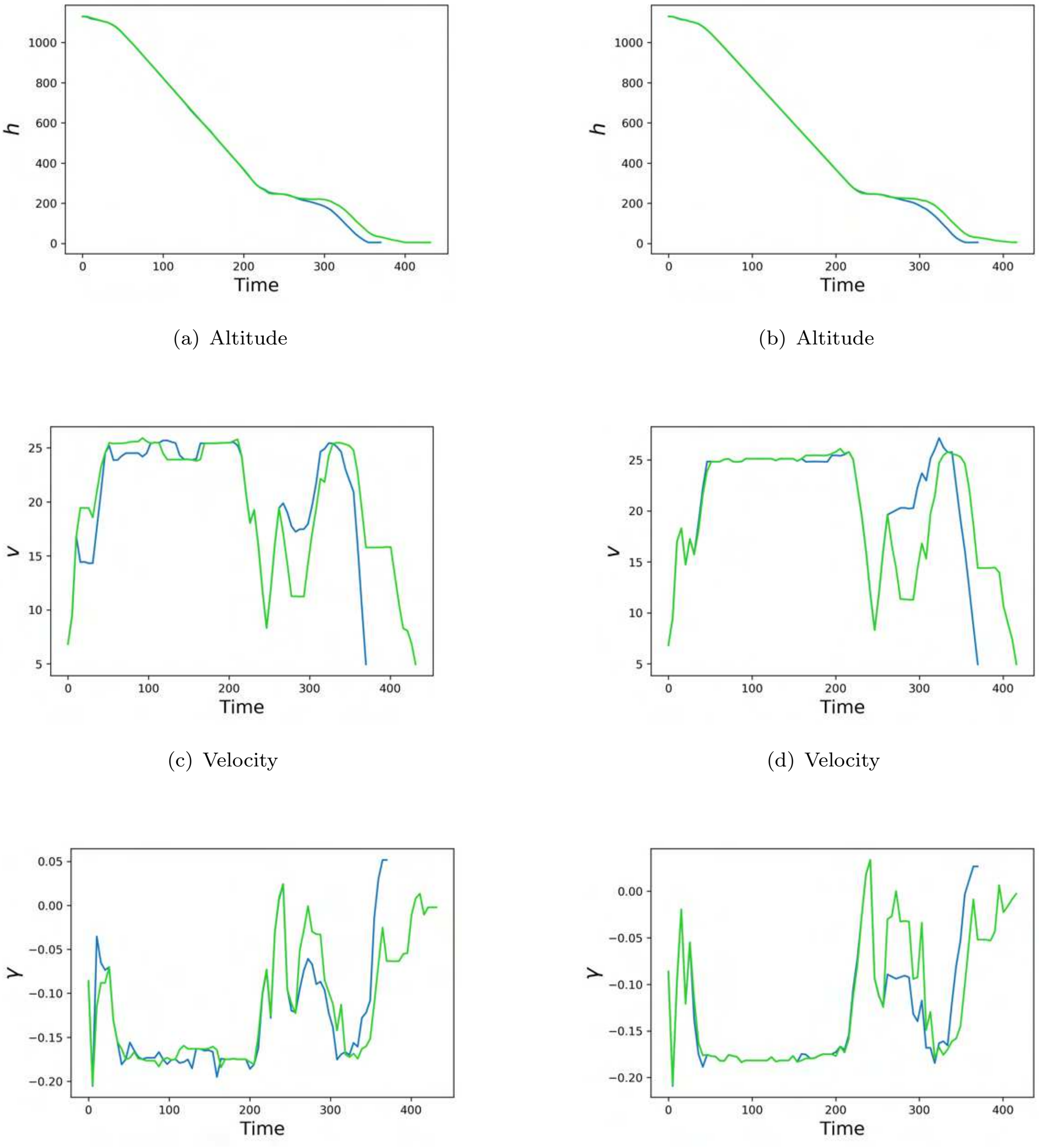
It is possible to notice from Figure 6.15 that in both scenarios we are able to find sound trajectories and complete the mission, with the wind being decisive in determining the final path. More precisely, it is possible to observe that each glider is forced to fly almost against it at the beginning when the altitude is still significant, see for example in the zoomed image of Figure 6.15(b). On the other hand, when the gliders reach more moderate altitudes, they can harness the wind by adjusting their flight attitude and reach faster the landing zone, thus reducing the total mission time, see Table 6.8.

Type	**Total CPU (s)**	**Flight Time (s)**	
		Green Glider	Blue Glider
$\beta = 0$	253.64	431.40	369.79
$\beta = 0.025$	490.81	415.99	369.79

TABLE 6.8: Flight and computational times for the solution attained in the presence and absence of wind.

In Figure 6.16, it is possible to observe some state variables profiles that help with the understanding of the overall mission. Gliders tend to move as fast as possible to the next target, whilst reducing their altitude if necessary and adjusting their attitude while visiting a waypoint or while landing, as can be clearly seen from velocity and flight path angle profiles. It is also possible to observe that these variables remain similar despite the different scenarios, adjusting solely the direction of flight to counterbalance the wind.

After a graphical verification of the sensibleness of the found optimal trajectories, we would like to investigate better measures to understand their quality and reliability.



(a) Altitude

(b) Altitude

(c) Velocity

(d) Velocity

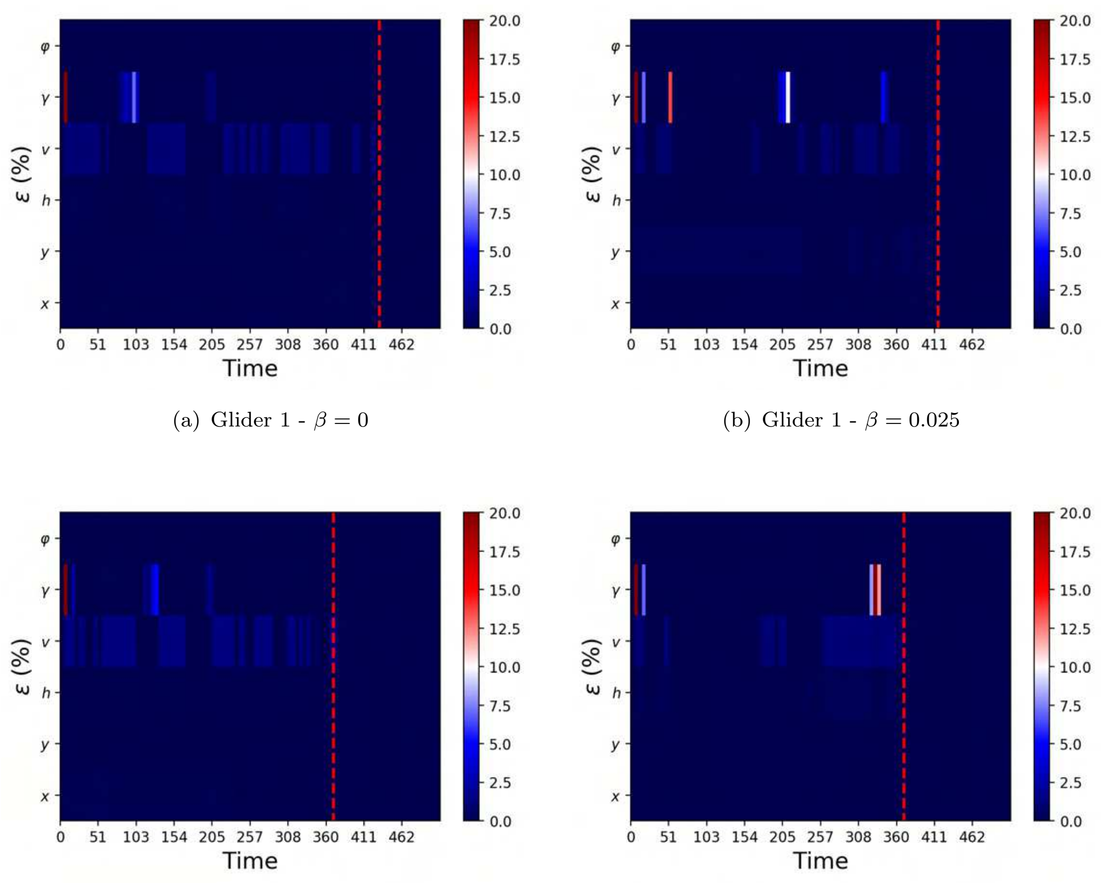
(e) Flight path angle

(f) Flight path angle

Figure 6.16: State variable profiles for the optimal trajectories found via algorithm 1, in scenarios with (right) and without (left) the presence of wind. Colours refer to the corresponding glider depicted in Figure 6.15.

6.3.1 Error Analysis

As previously stated, while examining multiple gliders' flights in a complete mission setting, it is not possible to have a complete evaluation of the goodness of the trajectory by comparing it with the corresponding NLP-based solution. Indeed the latter would present instabilities and in most of the instances would neither be able to return sensible trajectories nor to find them within a reasonable time. Nonetheless, it is possible to



(a) Glider 1 - $\beta = 0$

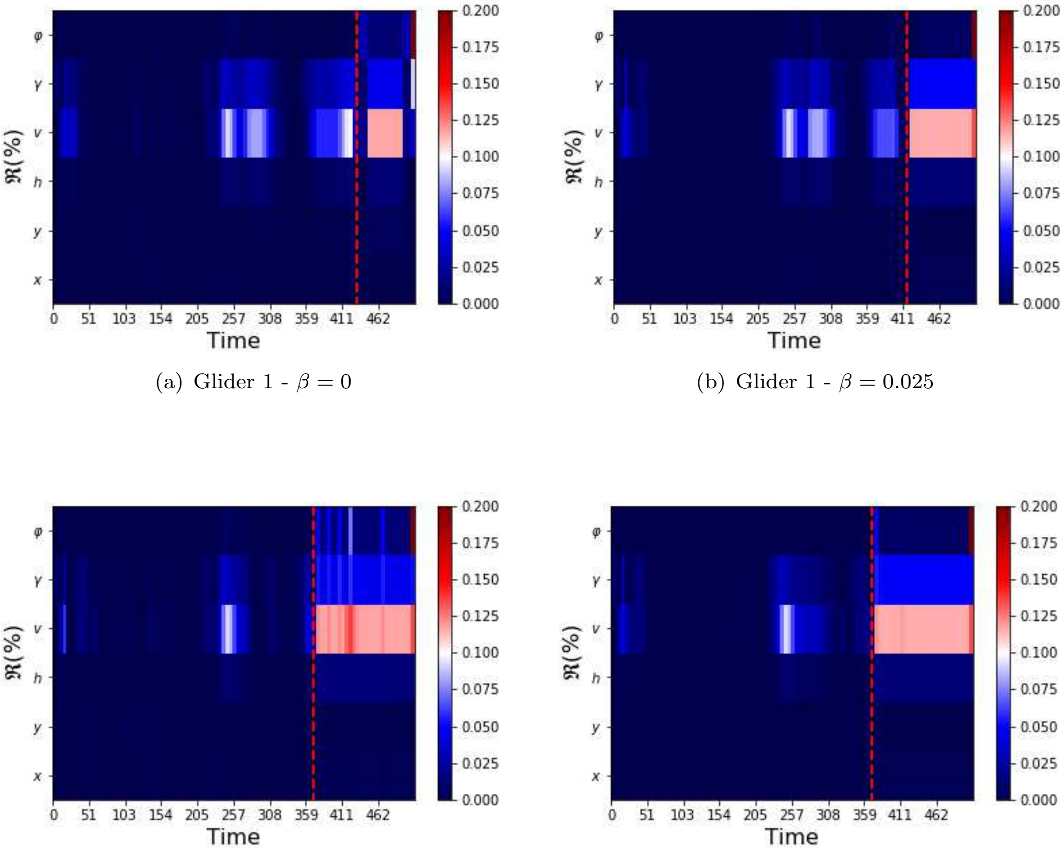
(b) Glider 1 - $\beta = 0.025$

(c) Glider 2 - $\beta = 0$

(d) Glider 2 - $\beta = 0.025$

Figure 6.17: Relative deviations from the linearised dynamic during the flight for each glider and in both wind scenarios. Vertical red dashed lines indicate the moment of landing for the respective glider.

measure the quality of the trajectory found via algorithm 1 by observing first the penalty terms $\boldsymbol{\varepsilon}$ in the objective function of Model I, indicating the deviations from the linearised dynamic, and second by deploying local error measures to capture ”errors” with respect to the nonlinear dynamic as described in definitions 5.2 and 5.3. In this way, we are able to get a more comprehensible and significant measure of the error we commit by making linear approximations and deploying the Two Step Algorithm for the GRTOP. If we apply these definitions to the flight example just shown, and if we get a relative measure, analogous to the one in definition 5.3, for the penalty term $\boldsymbol{\varepsilon}$ expressing the deviation from the linearised dynamic, it is possible to observe in Figure 6.17 that the latter is in general small, even close to zero for most of the time and for most of the variables. Only some adjustment on the flight path angle γ are required, but only for a short period of time, typically few time steps. At the same time, we can observe in Figure 6.18 that the relative local error measure is also small for the majority of the flight with some speed and flight path angle adjustments required especially during waypoint visits and while landing, but never exceeding 10%.



(a) Glider 1 - $\beta = 0$

(b) Glider 1 - $\beta = 0.025$

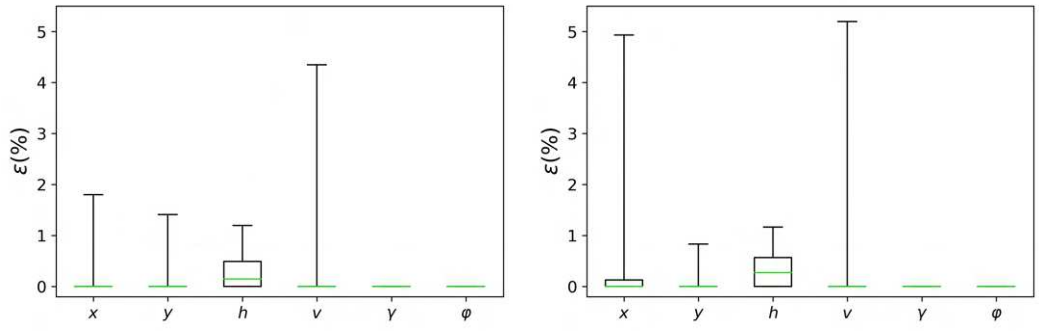
(c) Glider 2 - $\beta = 0$

(d) Glider 2 - $\beta = 0.025$

FIGURE 6.18: Relative local error measure for each glider and in both wind scenarios. Vertical red dashed lines indicate the moment of landing for the respective glider.

To further analyse the reliability of our model and the Two Step Algorithm for the GRTOP, we prepare more flight simulations and we measure the relative deviation from the linearised dynamic as well as the relative local errors in all of them. In particular, we consider 120 medium range instances where one to three gliders are required to visit some waypoints, from 2 to 6, and land in a predefined landing sites, maximum 2. As usual, we investigate scenarios where the wind is absent or blowing at the average UK speed at 500m. Gliders start at an altitude between 120m and 1500m, then the waypoints and the landing site are randomly located within an area of $5 \times 5km^2$ around the starting position. Collecting the interested quantities for each flight simulation, we can compute the distributions of the deviations from the linearised dynamic, as well as the distributions of the relative local errors. The results are then reported in Figure 6.19 and Figure 6.20 respectively.

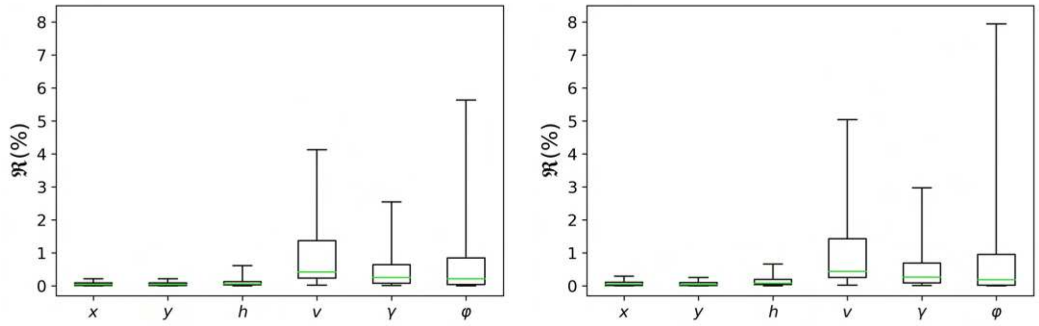
As it is possible to notice, the relative error term as well as the relative divergence from the linearised dynamic can be considered small, with some sporadic circumstance where deviations from both the linearised and/ or the true dynamic is deemed necessary by the solver. Nonetheless, these situations are limited and the vast majority of the relative



(a) $\beta = 0$

(b) $\beta = 0.025$

FIGURE 6.19: Distribution of relative deviations from the linearised dynamic for 120 flight simulations within different wind scenarios.



(a) $\beta = 0$

(b) $\beta = 0.025$

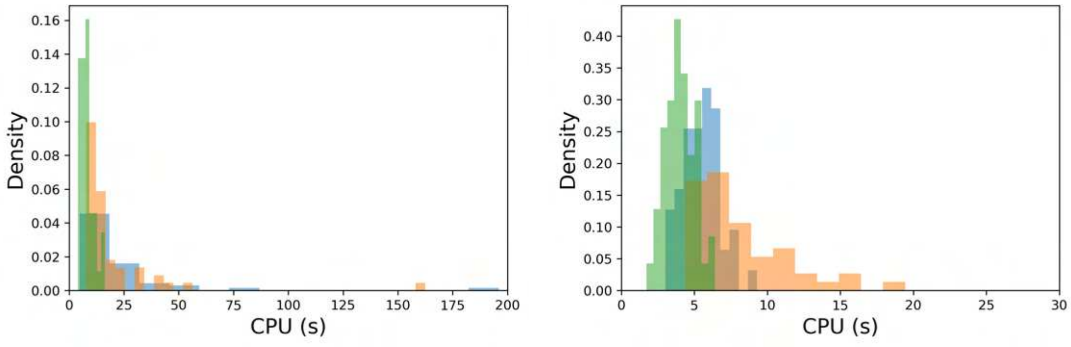
FIGURE 6.20: Distribution of the relative local errors for 120 flight simulations within different wind scenarios.

local errors and deviation from the linearised dynamic falls within 5%, reinforcing the validity of our model formulation.

After analysing multiple gliders' flights, we proceed by investigating first the alternative solvers available in the market, and second by examining the scalability of our Two Step Algorithm for the GRTOP when we deploy the best identified solver.

6.4 Comparison between solvers

Mixed-Integer Linear Programming (MILP) formulations of an optimisation problem offer the indisputable advantage of being solvable by leveraging on powerful and readily available commercial solvers. Therefore, common solvers among the research community, see Mittelmann (2017), like CPLEX, version 22.1.1.0, Xpress, version 9.0.3, and Gurobi, version 10.0.1, can be deployed to solve instances of Model 0 and Model I, and the stemming GRTOP solutions can be compared. Thus, we observe the performances obtained on 50 instances randomly generated where a glider starting at an altitude between 250m and 500m is required to visit 3 waypoints and land in a single predefined



(a) Model 0

(b) Model I

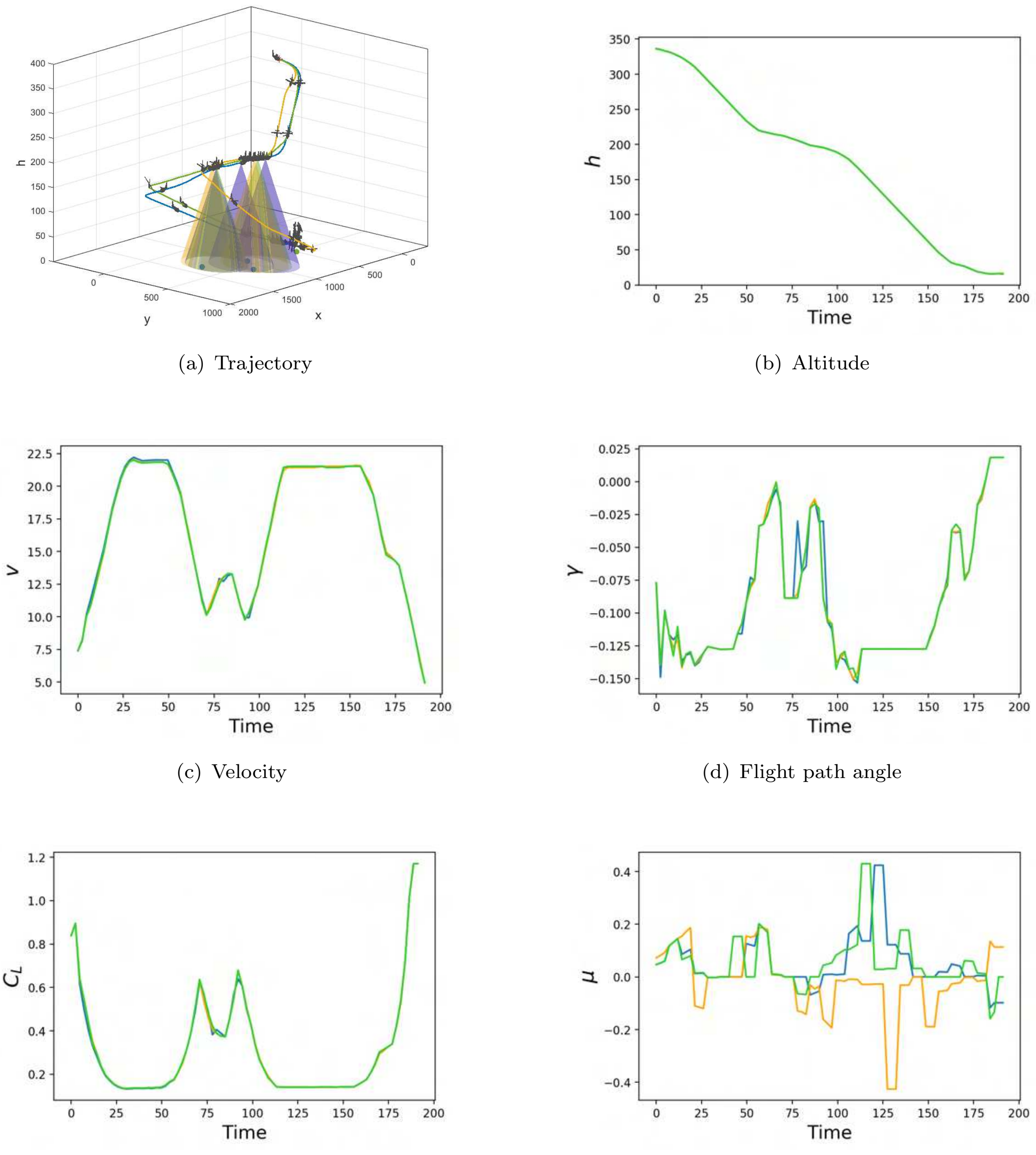
Figure 6.21: A comparison of computational times necessary to solve the corresponding instances of Model 0 (left) and Model I (right) within algorithm 1, using different commercial solvers. In blue the distribution of computational times relative to CPLEX, in orange the one corresponding to Xpress and in green the one for Gurobi.

landing zone, with the mission taking place in a radius of 1500m, this time only in absence of wind, $\beta = 0$. We solve each of the instances by deploying $N = 10$ coarse time steps in Model 0 of algorithm 1, and $N = 27$ coarse time steps for Model I, while choosing $w_4 = 0.1$. Other parameters are chosen like in Table 6.1. Results are reported in Figure 6.21, where computational times' distributions are compared for both the first step (Model 0) of the algorithm as well as the second one (Model I).

For the chosen instances, all the commercial solvers are able to find an optimal solution although with different solving times. As it is possible to observe, while CPLEX and Xpress show similar behaviour, especially in the first step of algorithm 1, Gurobi is able to outperform the other solvers significantly in each step. It is worth noting here that from this analysis it is also possible to deduce that generally it is easier to solve the second step of the problem, with all of the computational times necessary to solve it below 20s. This seems indeed reasonable, given that an initial trajectory is provided and deeper exploration is therefore rarely required. Finally, we report here one instance where the found optimal trajectories and variable profiles, Figure 6.22, mission's times and average deviation from the linearised dynamic, Table 6.9, and relative local error measures, Figure 6.23, are also compared for the three solvers.

Solver	**Flight Time (s)**	**Total CPU (s)**	$\bar{\varepsilon}_x$	$\bar{\varepsilon}_y$	$\bar{\varepsilon}_h$	$\bar{\varepsilon}_v$	$\bar{\varepsilon}_\gamma$	$\bar{\varepsilon}_\varphi$
CPLEX	191.26	9.79	0.1%	0.1%	0.0%	0.5%	3.2%	0.0%
Xpress	191.26	16.14	0.1%	0.1%	0.0%	0.5%	2.7%	0.0%
Gurobi	191.26	8.42	0.2%	0.1%	0.0%	0.5%	2.8%	0.0%

Table 6.9: Comparison of flight times, computational times and average deviations from the linearised dynamic for the solutions of the instance attained using different commercial solvers.



(a) Trajectory

(b) Altitude

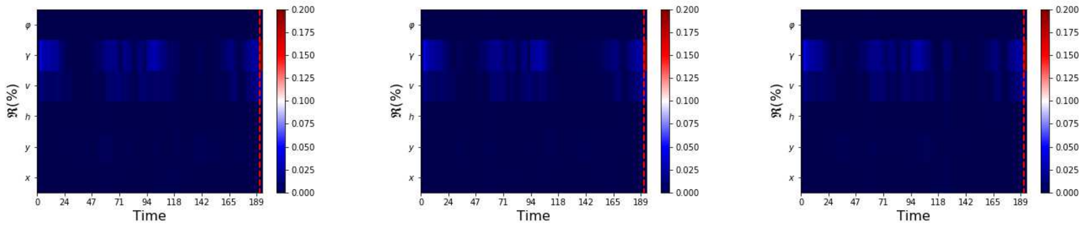
(c) Velocity

(d) Flight path angle

(e) Lift coefficient

(f) Bank angle

FIGURE 6.22: Comparison of trajectories and variables profiles obtained using CPLEX (blue), Xpress (orange) and Gurobi (green) as solver.



(a) CPLEX

(b) Xpress

(c) Gurobi

FIGURE 6.23: Comparison of the relative local error measure for the solution attained using different commercial solvers. Vertical red dashed lines indicate landing times.

It is possible to observe that all the three solvers return a similar solution, with moderate error terms. Different decisions are made regarding the bank angle which lead, in case of Gurobi and CPLEX to a similar solution, whereas Xpress decide for a opposite direction of landing, creating a symmetrical path. Due to the absence of wind, solvers decisions are completely legitimate, and in fact irrelevant for determining the flight time. On the other hand, CPLEX solution opts for a separate visit to the three waypoints, whereas Xpress and Gurobi decide to visit two waypoints simultaneously, photographing the remaining object in a separate moment. As before, this decision does not influence the overall mission times. In light of this, given the similarity of solution quality and the substantial differences in computational time performances, we consider Gurobi as the state-of-the-art solver for tackling this type of problems and it has indeed been chosen for the computational experiments presented in this manuscript. With this in mind, we can proceed by examining the computational efficiency and scalability of our GRTOP formulation in terms of number of gliders, number of waypoints, as well as number of landing points.

6.5 Problem Scalability

In this section we investigate one fundamental aspect of solving the GRTOP: its scalability, i.e. how efficiently we can compute optimal solutions as the problem size increases. To address this question, we explore the the performance of our numerical model and optimisation algorithm, with particular focus on its behaviour when the number of glider, waypoints and landing sites vary. Thus, a number of test instances are generated at first. The instances considered have between 2 and 7 waypoints and a maximum of 2 landing regions. The number of available gliders is chosen between 1 and 3, with special interest in situations where the number of waypoints is at least double the one of gliders. Target objects and landing areas are scattered over an area of $5 \times 5km^2$. Then, for each combination of number of gliders, waypoints and landing sites, we create 5 random instances. Finally, gliders can initiate the mission from a randomly generated launching site within the area, at an altitude varying between 100 and 1500m, and a moderate initial velocity. Then, we solve each GRTOP instance by deploying algorithm 1, with $N = 10$ coarse time steps in its first step and 54 in its second step, for the usual cases where the wind is respectively absent or blowing at average UK intensity. All the other parameters are once again specified like in Table 6.1 and a CPU limit of 1h (3600s) for solving the corresponding problem is imposed to both steps of the algorithm. Ultimately, results relative to interesting instances (instances of lower size were all solved to optimality) are reported in Table 6.10 and Table 6.11. It is possible to observe that for instances with less than 6 waypoints, we are generally able to solve the problem to optimality or we reach a small relative optimality gap, at least in one of the two steps of algorithm 1. The resulting relative local error is also small on average with higher maximum values,

Name	#G	#W	#L	CPU(s)		Gap (%)		$\mathscr{R}$(%)	
				Step 1	Step 2	Step 1	Step 2	Mean	Max
grtop_261_1	2	6	1	901.51	189.81	0.0%	0.0%	0.52%	17.72%
grtop_261_2	2	6	1	134.91	127.55	0.0%	0.0%	0.42%	16.47%
grtop_261_3	2	6	1	340.66	100.19	0.0%	0.0%	0.74%	17.22%
grtop_261_4	2	6	1	720.01	191.78	0.0%	0.0%	0.71%	13.78%
grtop_261_5	2	6	1	120.75	262.11	0.0%	0.0%	0.56%	18.75%
grtop_262_1	2	6	2	812.23	159.08	0.0%	0.0%	0.26%	4.59%
grtop_262_2	2	6	2	1599.2	275.89	0.0%	0.0%	0.28%	6.64%
grtop_262_3	2	6	2	144.0	211.55	0.0%	0.0%	0.35%	3.71%
grtop_262_4	2	6	2	530.39	133.28	0.0%	0.0%	0.29%	7.86%
grtop_262_5	2	6	2	625.91	122.53	0.0%	0.0%	0.38%	5.63%
grtop_351_1	3	5	1	1005.78	270.7	0.0%	0.0%	0.55%	13.28%
grtop_351_2	3	5	1	420.03	629.39	0.0%	0.0%	0.6%	20.51%
grtop_351_3	3	5	1	2354.16	197.69	0.0%	0.0%	0.94%	20.42%
grtop_351_4	3	5	1	3600	3433.81	3.0%	0.0%	0.47%	18.45%
grtop_351_5	3	5	1	3130.27	798.53	0.0%	0.0%	0.52%	11.8%
grtop_352_1	3	5	2	3600	1236.0	16.0%	0.0%	0.43%	15.62%
grtop_352_2	3	5	2	3600	361.03	21.0%	0.0%	0.27%	4.17%
grtop_352_3	3	5	2	3600	412.97	16.0%	0.0%	0.33%	11.02%
grtop_352_4	3	5	2	3600	522.89	17.0%	0.0%	0.35%	6.0%
grtop_352_5	3	5	2	3600	477.86	13.0%	0.0%	0.37%	5.8%
grtop_361_1	3	6	1	3600	691.42	25.0%	0.0%	0.53%	12.9%
grtop_361_2	3	6	1	3600	453.28	38.0%	0.0%	0.32%	17.93%
grtop_361_3	3	6	1	3600	1210.2	33.0%	0.0%	0.45%	19.3%
grtop_361_4	3	6	1	3177.49	579.61	0.0%	0.0%	0.5%	12.44%
grtop_361_5	3	6	1	1776.84	309.97	0.0%	0.0%	0.53%	14.4%
grtop_362_1	3	6	2	3600	2287.36	21.0%	0.0%	0.41%	4.77%
grtop_362_2	3	6	2	3600	1040.58	19.0%	0.0%	0.46%	6.5%
grtop_362_3	3	6	2	3470.95	1416.52	0.0%	0.0%	0.32%	4.68%
grtop_362_4	3	6	2	3600	1080.69	10.0%	0.0%	0.33%	6.77%
grtop_362_5	3	6	2	3600	1993.38	25.0%	0.0%	0.38%	7.82%
grtop_371_1	3	7	1	3600	3600	34.0%	1.0%	3.14%	18.35%
grtop_371_2	3	7	1	3600	3600	18.0%	5.0%	1.72%	30.05%
grtop_371_3	3	7	1	3600	3600	124.0%	3.0%	2.64%	29.9%
grtop_371_4	3	7	1	3600	3600	9.0%	2.0%	2.58%	30.11%
grtop_371_5	3	7	1	3600	1345.8	10.0%	0.0%	2.41%	25.13%
grtop_372_1	3	7	2	1758.28	1542.7	0.0%	0.0%	0.48%	9.32%
grtop_372_2	3	7	2	3600	3007.3	45.0%	0.0%	0.88%	15.53%
grtop_372_3	3	7	2	3600	1719.72	68.0%	0.0%	0.51%	11.48%
grtop_372_4	3	7	2	3600	3600	75.0%	5.0%	0.68%	35.24%
grtop_372_5	3	7	2	3600	3600	43.0%	3.0%	1.16%	22.71%

Table 6.10: Computational results for a number of instances with absent wind.

indicating that for short period of time an adjustment might be required for some of the variables. Furthermore, it is also possible to observe some effect of the wind, helping sometimes to improve the solvability in the first step of the algorithm, while increasing the difficulties in the second step, see for examples the instances "grtop_352". On the other hand, instances with 7 or more waypoints become complicated to solve, showing a higher optimality gap. At the same time, apart from very few instances, we observe higher average and maximum relative local errors, suggesting that feasibility of the trajectory remains to be proven.

We have seen here that our Two Step Algorithm can successfully can return sound mission plans and flyable trajectories for many reasonable size real-world instances of the GRTOP. Nevertheless, when dealing with larger instances, we are unable to always demonstrate optimality of the path chosen. Further improvements might then be necessary to continuously advance this challenging and fascinating research area.

Name	**#G**	**#W**	**#L**	**CPU(s)**		**Gap (%)**		$\mathscr{R}$(%)	
				Step 1	Step 2	Step 1	Step 2	Mean	Max
grtop_261_1	2	6	1	1168.78	354.97	0.0%	0.0%	0.51%	9.66%
grtop_261_2	2	6	1	289.17	619.14	0.0%	0.0%	0.37%	8.31%
grtop_261_3	2	6	1	90.09	327.38	0.0%	0.0%	0.42%	8.71%
grtop_261_4	2	6	1	896.62	324.56	0.0%	0.0%	0.82%	18.44%
grtop_261_5	2	6	1	278.75	182.89	0.0%	0.0%	0.51%	8.82%
grtop_262_1	2	6	2	593.59	937.64	0.0%	0.0%	0.25%	7.29%
grtop_262_2	2	6	2	1824.03	505.2	0.0%	0.0%	0.37%	7.62%
grtop_262_3	2	6	2	190.83	674.19	0.0%	0.0%	0.65%	11.64%
grtop_262_4	2	6	2	970.08	409.33	0.0%	0.0%	0.27%	4.06%
grtop_262_5	2	6	2	206.67	643.38	0.0%	0.0%	0.36%	7.61%
grtop_351_1	3	5	1	739.09	2723.27	0.0%	0.0%	0.81%	20.26%
grtop_351_2	3	5	1	541.31	1484.72	0.0%	0.0%	0.45%	17.4%
grtop_351_3	3	5	1	537.98	1525.56	0.0%	0.0%	0.61%	18.09%
grtop_351_4	3	5	1	1707.86	3600	0.0%	3.0%	0.63%	14.61%
grtop_351_5	3	5	1	1162.95	3600	0.0%	2.0%	0.43%	12.44%
grtop_352_1	3	5	2	3600	2129.09	22.0%	0.0%	0.42%	10.72%
grtop_352_2	3	5	2	1779.8	2236.69	0.0%	0.0%	0.33%	15.7%
grtop_352_3	3	5	2	1511.17	811.52	0.0%	0.0%	0.35%	12.58%
grtop_352_4	3	5	2	3515.55	913.44	0.0%	0.0%	0.41%	13.59%
grtop_352_5	3	5	2	1257.12	1242.12	0.0%	0.0%	0.48%	15.73%
grtop_361_1	3	6	1	3600	3600	24.0%	3.0%	0.56%	14.85%
grtop_361_2	3	6	1	3235.72	3600	0.0%	9.0%	0.69%	16.75%
grtop_361_3	3	6	1	3600	3507.8	36.0%	0.0%	0.46%	10.74%
grtop_361_4	3	6	1	2361.02	948.5	0.0%	0.0%	0.67%	14.9%
grtop_361_5	3	6	1	1421.59	2625.45	0.0%	0.0%	0.86%	17.01%
grtop_362_1	3	6	2	3600	1995.03	25.0%	0.0%	0.67%	14.64%
grtop_362_2	3	6	2	3600	2725.67	19.0%	0.0%	0.59%	12.57%
grtop_362_3	3	6	2	3600	2608.51	16.0%	0.0%	0.54%	14.1%
grtop_362_4	3	6	2	3600	1302.58	8.0%	0.0%	0.51%	17.06%
grtop_362_5	3	6	2	3600	3600	28.0%	2.0%	0.47%	10.64%
grtop_371_1	3	7	1	3600	3600	62.0%	5.0%	3.74%	33.95%
grtop_371_2	3	7	1	3600	3600	273.0%	3.0%	2.5%	35.69%
grtop_371_3	3	7	1	3600	3600	285.0%	9.0%	2.44%	37.21%
grtop_371_4	3	7	1	3600	3600	37.0%	3.0%	3.25%	41.71%
grtop_371_5	3	7	1	3600	3600	38.0%	12.0%	4.03%	36.42%
grtop_372_1	3	7	2	3600	3600	26.0%	20.0%	0.6%	22.4%
grtop_372_2	3	7	2	3600	3600	82.0%	4.0%	0.94%	37.83%
grtop_372_3	3	7	2	3600	3600	79.0%	9.0%	0.62%	36.49%
grtop_372_4	3	7	2	3600	3600	110.0%	16.0%	1.69%	35.79%
grtop_372_5	3	7	2	3600	3600	41.0%	14.0%	1.0%	21.88%

Table 6.11: Computational results for a number of instances with blowing wind.

Chapter 7

Conclusions and Directions for Future Research

This chapter marks the culmination of our journey into the optimal control and routing of autonomous flying vehicles. We describe here the core findings of our research and their implications. As we summarise these discoveries, we also look forward, outlining directions for further research and identifying unexplored avenues that promise further innovation and understanding of vehicle routing and trajectory optimisation problems.

7.1 Research Findings

In this thesis, motivated by several real-world applications like environmental control, distribution and delivery services, surveillance and security among many others, we address the intricate problem of Vehicle Routing and Trajectory Optimisation, also known as the VRTOP. This problem is notably complex due to the aggregation of combinatorial optimisation challenges, routing, and the involvement of nonlinearities and nonconvexities stemming from the equations governing vehicles dynamic that need to be solved to find the best trajectories and paths. During our literature review, it became apparent that definitions and solution techniques for the VRTOP are at their dawn, with previous research often making simplifying assumptions that did not capture the complications of the real world. In particular, when dealing with Unmanned Aerial Vehicles, several challenges emerge with the most prominent being the the development of models and scalable solution algorithms that are capable of solving the above problem within reasonable computing times, also taking into account the equations governing the dynamic of the UAVs within a certain degree of accuracy. These issues formed the core focus of our research efforts.

Considering special UAVs like gliders, i.e. aircraft without propulsion onboard, we develop techniques to simultaneously find optimal routes and trajectories for fleets of these vehicles, by first describing methodologies to produce fast and accurate solutions when prior knowledge of initial trajectories and/ or steady flight conditions is known, reducing the model to a Mixed-Integer Linear Programming Problem (MILP). More precisely, given the complexity of gliders' dynamic, we introduce techniques to linearise it as well as to reach the linearisation of distance constraints. We further enhance model's computational capabilities by means of a two time level partition, where binary variables are defined on a fewer number of time steps. Consequently, we presented the main contribution of our research that is a model enabling us to find initial trajectories and insights also in case of absent prior information on gliders dynamic. We introduce then a Two Step Solution Algorithm for the Glider Routing and Trajectory Optimisation Problem (GRTOP) able to integrate the initial trajectory model into the well-established MILP describing the problem when prior information is available.

With this novel methodology, we were able to find relatively quickly (sub) optimal reliable trajectories for small-medium sized problems by leveraging on state-of-the-art solvers like CPLEX, Gurobi or Xpress. We validate our results by comparing our solutions, where possible, with the ones obtained by solving a corresponding NLP where exact nonlinear glider's dynamic is taken into account. This shows our ability to find flyable paths in many different circumstances. On the other hand, our model presents some limitations that can be the base for future work. In particular, we show that optimality of the solutions in terms of mission's time might depend on the time partition as well as from other model parameters. In addition, despite our ability of finding flyable trajectories, extreme flight conditions might require adjustments that our model is not able to accurately identify. Nevertheless, we are able to suggest a viable planning that can be adjusted by the nonlinear onboard controls on the UAVs and classical control algorithms.

Other future directions are then discussed in the following section.

7.2 Future Research

We believe that this research project has contributed to the existing literature and could have a tremendous impact to future applications. Nonetheless, our methodology is dependent on (quasi) steady-state flight conditions and equilibria, hence suggesting that this research can be extended by enlarging these conditions for example considering situations of wind-driven (quasi) steady flight or situations of constant turn in order to perfect the choice of bank angles. In addition, extreme flight conditions can be investigated to accurately handle all flight situations. An even more complete exploration of the model's parameters space can be conducted along with wiser time partitioning that is denser close to adjustment points, i.e. the moment where a significant change of (quasi) steady-state flight is made.

Furthermore, alternative mission's goals and objectives can be considered such as the optimisation of the fleet size and mission costs as well as different type of functions like minimising the flight time of the latest landing glider. Thus, the launching position of the gliders can as well be optimised as it might affect decisions and the optimal plan. Further security requirements can also be inserted like collision avoidance for example. The GRTOP presented in this thesis can be easily adapted to other types of vehicles, such as a non homogeneous fleet of glider, powered UAVs, unmanned underwater vehicles and powered UAVs, by simply replacing gliders' dynamics with the appropriate equations and applying the presented methodology to the new class of autonomous vehicles. It can also be extended to be integrated as a part of broader missions, where for instance several GRTOP must be solved simultaneously and cooperating or competing fleet of gliders or UAVs are involved at different locations and can fight or collaborate to reach overall missions' goals. Finally, enlarging the set of weather's scenarios and inserting uncertainties in the problem formulation (for example regarding weather conditions, waypoint positions, etc.), we believe that the proposed model can have astounding implications in helping addressing many real-world problems and challenges.

Bibliography

E. Aarts and J. K. Lenstra. *Local search in combinatorial optimization.* Chichester, England: Wiley., 1997.

A. Albert, F.S. Leira, and L. Imsland. Uav path planning using milp with experiments. *Modeling, Identification and Control.*, 38(1):21–32, 2017.

D. Applegate, R.E. Bixby, V. Chvátal, and W. Cook. Tsp cuts which do not conform to the template paradigm. *Computational Combinatorial Optimization.*, 2241:261–304, 2000.

N. Aras, D. Aksen, and M. T. Tekin. Selective multi-depot vehicle routing problem with pricing. *Transportation Research Part C.*, 19(5):866–884, 2011.

K. B. Ariyur and K. O. Fregene. Autonomous tracking of a ground vehicle by a uav. *Proceedings of the 2008 American control conference. IEEE.*, page 669–671, 2008.

Aviation Security Service. Licensing and certification for pilots, steep-turns.

R. Baldacci, N. Christofides, and A. Mingozzi. An exact algorithm for the vehicle routing problem based on the set partitioning formulation with additional cuts. *Mathematical Programming.*, 2007.

R. Baldacci, E.A. Hadjiconstantinou, and A. Mingozzi. An exact algorithm for the capacitated vehicle routing problem based on a two-commodity network flow formulation. *Operations Research.*, 52:723–738, 2004.

R. Baldacci and A. Mingozzi. A unified exact method for solving different classes of vehicle routing problems. *Mathematical Programming.*, 120:347–380, 2009.

M. L. Balinski and R. E. Quandt. On an integer program for a delivery problem. *Operations Research.*, 12-2, 1964.

J. Beasley. Route first-cluster second methods for vrp. *Omega.*, 1:403–408, 1983.

T. Bektas. The multiple traveling salesman problem: an overview of formulations and solution procedures. *The International Journal of Management Science*, 34(3):209–219, 2006.

A. Ben-Tal, L. El Ghaoui, and A. Nemirovski. *Robust Optimization.* Philadelphia. PA: Society for Industrial and Applied Mathematics., 2001.

B. T. Bertka. An introduction to bezier curves, b-splines, and tensor product surfaces with history and applications., 2008. University of California Santa Cruz.

J. T. Betts. Survey of numerical methods for trajectory optimization. *Journal of Guidance, Control and Dynamics*, 21:193–207, 1998.

J. T. Betts. *Practical methods for optimal control using nonlinear programming. Advances in design and control.* Princeton and Oxford. Princeton University Press., 2009.

L.D. Bodin, B.L. Golden, A.A. Assad, and M.O. Ball. Routing and scheduling of vehicles and crews. the state of the art. *Computers and Operations Research*, 10:69–211, 1983.

T. J. Böhme and B. Frank. *Direct Methods for Optimal Control.* Springer., 2017.

J.D. Boissonnat, A. Cerezo, and J. Leblond. Shortest path of bounded curvature in the plane. *Proceedings of the IEEE International Conference on Robotics and Automation*, 3:2315–2320, 1992.

G.C. Bower. *Boundary Layer Dynamic Soaring for Autonomous Aircraft: Design and Validation.* PhD thesis, Stanford University, Stanford, California., 2010.

British Gliding Association. Turning guidelines, 2017.

A.E. Bryson. Optimal control-1950 to 1985. *IEEE Control Systems Magazine*, 16:26–33, 1996.

J. Buckley. *Air Power in the Age of Total War (Warfare and History).* Routledge., 1999.

Y. Chen, G. Luo, Y. Mei, J. Yu, and X. Su. Uav path planning using artificial potential field method updated by optimal control theory. *International Journal of Systems Science*, 47(6):1407–1420, 2016.

C. Cheng, Y. Adulyasak, and L.M. Rousseau. Drone routing with energy function: Formulation and exact algorithm. *Transportation Research Part B*, 139:364–387, 2020.

H. N. Chiu, Y. S. Lee, and J. H. Chang. Two approaches to solving the multi depot vehicle routing problem with time windows in a time-based logistics environment. *Production Planning and Control*, 17(5):480–493, 2006.

E. Choi and D. W. Tcha. A column generation approach to the heterogeneous fleet vehicle routing problem. *Computers & Operations Research*, 34(7):2080–2095, 2007.

S. Chowdhury, A. Emelogu, M. Marufuzzaman, S. G. Nurre, and L. Bian. Drones for disaster response and relief operations: A continuous approximation model. *International Journal of Production Economics*, 188:167–184, 2017.

N. Christofides. Vehicle routing. *The Traveling Salesman Problem. A Guided Tour of Combinatorial Optimization*, pages 431–448, 1985.

N. Christofides, A. Mingozzi, and P. Toth. State-space relaxation procedures for the computation of bounds to routing problems. *Wiley Periodicals*, pages 145–164, 1981.

G. Clarke and J. Wright. Scheduling of vehicles from a central depot a number of delivery points. *Operations Research.*, 12:568–581, 1964.

J. F. Cordeau. A branch-and-cut algorithm for the dial-a-ride problem. *Operations Research.*, 54:573–586, 2006.

J. F. Cordeau, M. Gendreau, and G. Laporte. A tabu search heuristic for periodic and multi-depot vehicle routing problems. *Networks*, 30(2):105–119, 1997.

J. F. Cordeau, M. Gendreau, G. Laporte, J.Y. Potvin, and F. Semet. A guide to vehicle routing heuristics. *Journal of the Operational Research Society*, 53(5):512–522, 2022.

J. F. Cordeau, G. Laporte, and A. Mercier. A unified tabu search heuristic for vehicle routing problems with time windows. *Journal of the Operational Research Society*, 52(8):928–936, 2001.

J.F. Cordeau, G. Laporte, M.W.P. Savelsbergh, and D. Vigo. Chapter 6 vehicle routing. *Handbooks in Operations Research and Management Science*, 14:367–428, 2007.

A. Cornell, B. Kloss, D.J. Presser, and R. Riedel. Drones take to the sky, potentially disrupting last-mile delivery, 2023.

J. A. Corona. Hiperheurísticas a través de programación genética para la resolución de problemas de ruteo de vehÌculos., 2005.

W. P. Coutinho, M. Battarra, and J. Fliege. The unmanned aerial vehicle routing and trajectory optimisation problem: a taxonomic review. *Computers and Industrial Engineering*, 120:116–128, 2018.

W. P. Coutinho, M. Battarra, and J. Fliege. Glider routing and trajectory optimisation in disaster assessment. *European Journal of Operational Research*, 274:1138–1154, 2019.

C. Crispin. *Path planning algorithms for atmospheric science applications of autonomous aircraft systems.* PhD thesis, Southampton, UK: University of Southampton, 2016.

G. Dantzig, R. Fulkerson, and S. Johnson. Solution of a large-scale travelling salesman problem. *Operations Research*, 2:393–410, 1954.

M. Desrochers, J.K. Lenstra, and M.W.P. Savelsbergh. A classification scheme for vehicle routing and scheduling problems. *European Journal of Operational Research*, 46:322–332, 1990.

D. R. Drew, J. F. Barlow, and S. E. Lane. Observations of wind speed profiles over greater london, uk, using a doppler lidar. *Journal of Wind Engineering and Industrial Aerodynamics*, 121:98–105, 2013.

DII Drone Industry Insights. Drone application report 2022., 2022.

S. Eilon, C. D. T. Watson-Gandy, and N. Christofides. *Distribution Management-Mathematical Modelling and Practical Analysis.* London, Griffin., 1971.

B. Eksioglu, A.V. Vural, and A. Reisman. The vehicle routing problem: A taxonomic review. *Computers & Industrial Engineering*, 57:1472–1483, 2009.

J. Elston, B. Argrow, M. Stachura, D. Weibel, D. Lawrence, and D. Pope. Overview of small fixed-wing unmanned aircraft for meteorological sampling. *Journal of Atmospheric and Oceanic Technology*, 32:97–115, 2015.

J. Escobar, R. Linfati, and P. Toth. A two-phase hybrid algorithm for the capacitated location-routing problem. *Computers and Operations Research*, 40(1):70–79, 2013.

M. Farhan, I. Ahmed, M. Nader, J. Al-Jaroodi, and J. Imad. Uavs for smart cities: Opportunities and challenges. *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2014.

M. Filipec, D. Skrlec, and S. Krajcar. New approach to multiple depot capacitated vehicle routing problem. *In Proceedings of the 1997 IEEE International conference on systems, man and cybernetics*, 1:421–426, 1997.

M. Fischetti, P. Toth, and D. Vigo. A branch-and-bound algorithm for the capacitated vehicle routing problem on directed graphs. *Operations Research*, 42-5:846–859, 1994.

T. Flanzer. *Robust Trajectory Optimisation and Control of a Dynamic Soaring Unmanned Aerial Vehicle.* PhD thesis, Stanford University, Stanford, 2012.

P. Flisberg, B. Liden, and M. Rönnqvist. A hybrid method based on linear programming and tabu search for routing of logging trucks. *Computers and Operations Research*, 36(4):1122–1144, 2009.

A. Fügenschuh and D. Müllenstedt. Flight planning for unmanned aerial vehicles., 2015. Helmut Schmidt University/University of the Federal Armed Forces Hamburg.

R. Fukasawa, H. Longo, J. Lysgaard, M. P. de Aragão, M. Reis, E. Uchoa, and R. F. Werneck. Robust branch-and-cut-and-price for the capacitated vehicle routing problem. *Mathematical Programming.*, 106, 2006.

A. Gasparetto, P. Boscariol, A. Lanzutti, and R. Vidoni. Path planning and trajectory planning algorithms: A general overview. *Motion and operation planning of robotic systems mechanisms and machine science.*, pages 3–27, 2015.

M. Gendreau, G. Laporte, and R. Seguin. Stochastic vehicle routing. *European Journal of Operational Research*, 88:3–12, 1996.

M. Gendreau and J.Y. Potvinm. Dynamic vehicle routing and dispatching. *Fleet management and logistic*, page 115–226, 1998.

P.E. Gill, W. Murray, and M.A. Saunders. User's guide for snopt version 7: Software for large-scale nonlinear programming. University of California, San Diego and Stanford University, Stanford.

K. Glock and A. Meyer. Mission planning for emergency rapid mapping with drones. *Transportation science*, 54(2):534–560, 2020.

B.S. Goh. Optimal singular rocket and aircraft trajectories. *IEEE, 2008 Chinese Control and Decision Conference*, 2008.

B. L. Golden, A. A. Assad, L. Levy, and F. G. Gheysens. The fleet size and mix vehicle routing problem. *Computers & Operations Research*, 11(1):49–66, 1984.

B. L. Golden, T. L. Magnanti, and H. Q. Nguyan. Implementing vehicle routing algorithms. *Networks*, 7(2):113–148, 1972.

B.L. Golden and A.A. Assad. *Vehicle Routing: Methods and Studies.* North-Holland, Amsterdam., 1988.

Gurobi. Gurobi optimizer reference manual. Version 10.0, Copyright © 2023, Gurobi Optimization, LLC.

E. Hadjiconstantinou and R. Baldacci. A multi-depot period vehicle routing problem arising in the utilities sector. *Journal of the Operational Research Society*, 49(12): 1239–1248, 1998.

E. Hairer, S. P. Norsett, and G. Wanner. *Solving ordinary differential equations I.* Springer. Springer series in computational mathematics., 2011.

C. Hajiyev, H. E. Soken, and S. Y. Vural. *State estimation and control for low-cost unmanned aerial vehicles.*, chapter Equations of motion for an unmanned aerial vehicle., pages 9–23. Springer International Publishing., 2015.

M. W. Harris and Acikmese. Maximum divert for planetary landing using convex optimization. *Journal of Optimization Theory and Applications*, 162:975–995, 2013.

S. Hayat, E. Yanmaz, and R. Muzaffar. Survey on unmanned aerial vehicle networks for civil applications: A communications viewpoint. *IEEE Communications Surveys and Tutorials*, 18(4):2624–2661, 2016.

K. Helsgaun. An effective implementation of the lin-kernighan traveling salesman heuristic. *European Journal of Operational Research*, 126:106–130, 2000.

W. Ho, G. T. S. Ho, P. Ji, and H. C. W. Lau. A hybrid genetic algorithm for the multi-depot vehicle routing problem. *Engineering Applications of Artificial Intelligence*, 21(4):548–557, 2008.

J. P. How, E. Frazzoli, and G. V. Chowdhary. *Handbook of unmanned aerial vehicles.*, chapter Linear flight control techniques for unmanned aerial vehicles., page 529–576. Springer Netherlands., 2015.

IBM-ILOG-CPLEX. Ibm ilog cplex optimization studiocplex user's manual. Version 12, Release 8.

International Civil Aviation Organisation. *Aircraft Operations.*, volume I. 2018.

H.R. Joshi. Optimal control of an hiv immunology model. *Optimal Control Appl. Methods*, 23:199–213, 2002.

W. Kagabo. Optimal trajectory planning for a uav glider using atmospheric thermals., 2010.

F. Kai, H. Songchen, L. Wenjing, and L. Binbin. 4d trajectory planning of unmanned aerial vehicle in trajectory based operation airspace. *2nd International Conference on Robot Systems and Applications*, page 51–58, 2019.

K. Kanistras, G. Martins, M. J. Rutherford, and K. P. Valavanis. A survey of unmanned aerial vehicles (uavs) for traffic monitoring. *International Conference on Unmanned Aircraft Systems (ICUAS)*, 2013.

A. Keane, J. Scanlan, A. Lock, M. Ferraro, P. Spillane, and J. Breen. Maritime flight trials of the southampton university laser sintered aircraft–project albatross. *The Aeronautical Journal*, 121:1502–1529, 2017.

T. Keviczky, F. Borrelli, K. Fregene, D. Godbole, and G. Balas. Decentralized receding horizon control and coordination of autonomous vehicle formations. *IEEE Transactions on Control Systems Technology*, 16:19–33, 2008.

D.E. Kirk. *Optimal Control Theory: An Introduction.* Dover Publications., 1998.

G. Laporte. Développements algorithmiques récents et perspectives de recherche en distributique. *Les Cahiers Scientifiques du Transport*, 21:61–84, 1990.

G. Laporte. The vehicle routing problem: An overview of exact and approximate algorithms. *European Journal of Operational Research*, 59:345–358, 1992.

G. Laporte, M. Gendreau, J. Potvin, and F. Semet. Classical and modern heuristics for the vehicle routing problem. *International Transactions in OR*, 7:285–300, 2002.

G. Laporte and Y. Nobert. Exact algorithms for the vehicle routing problem. *Surveys in Combinatorial Optimization, North-Holland, Amsterdam*, pages 147–184, 1987.

G. Laporte, Y. Nobert, and M. Desrochers. Optimal routing under capacity and distance restrictions. *Operations Research*, 33:1050–1073, 1985.

A. Larsen. *The dynamic vehicle routing problem.* PhD thesis, Department of Mathematical Modelling (IMM), Technical University of Denmark (DTU), Lyngby, Denmark, 2000.

S. Laville, A. Ross, A. Topping, D. Gayle, and J. Grierson. Grenfell tower: Firefighters search overnight with toll expected to rise, 2017.

B. Li, X. Qi, B. Yu, and L. Liu. Trajectory planning for uav based on improved aco algorithm. *IEEE Access*, 8:2995–3006, 2020.

Y. Li and X. Liu. Modelling and its ant colony algorithm for multi-depot open vehicle routing problem with replenishment on the way. *Computer Integrated Manufacturing Systems*, 2008.

A. Lim and W. Zhu. A fast and effective insertion algorithm for multi-depot vehicle routing problem and a new simulated annealing approach. *Advances in artificial intelligence. Lecture notes in computer science*, 4031:282–291, 2006.

S. Lin and W. Kernighan. An effective heuristic algorithm for the traveling-salesman problem. *Operations Research*, 21:498–516, 1973.

C. Liu and J. Yu. Multiple depots vehicle routing based on the ant colony with the genetic algorithm. *Journal of Industrial Engineering and Management*, 6(4):1013–1026, 2013.

C. Y. Liu. An improved adaptive genetic algorithm for the multi-depot vehicle routing problem with time windows. *Journal of Networks*, 8(5):1035–1042, 2013.

A. M. Low. The first guided missile. *Royal Air Force Flying Review*, 26-27:436–438, 1958.

A. Maolaaisha. Free-flight trajectory optimization by mixed integer programming. Master's thesis, University of Hamburg, Hamburg, 2015. Available at: epub.sub.uni-hamburg.de/epub/volltexte/2020/100621/pdf/828597510.pdf.

A. C. Matos and R. C. Oliveira. An experimental study of the ant colony system for the period vehicle routing problem. *Ant colony optimization and swarm intelligence. Lecture Notes in Computer Science*, 3172:1–29, 2004.

G. P. McCormick. Computability of global solutions to factorable nonconvex programs: part i - convex underestimating problems. *Mathematical Programming.*, 10:147–175, 1976.

A. McKenna. *The Future of Drone Use: Opportunities and Threats from Ethical and Legal Perspectives.* Springer., 2016.

K. Miettinen. *Nonlinear Multiobjective Optimization.* Boston. Kluwer Academic Publishers., 1999.

C.E. Miller, A.W. Tucker, and R.A. Zemlin. Integer programming formulation of traveling salesman problems. *Journal of the Association for Computing Machinery*, 7: 326–329, 1960.

H. D. Mittelmann. Mixed-integer socp benchmark, 2017.

R. Mole and S. S. Jameson. A sequential route-building algorithm employing a generalized saving criterion. *Operation Research*, 11:503–511, 1976.

G. Nagy and S. Salhi. Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries. *European Journal of Operational Research*, 162(1):126–141, 2005.

A. Nedjati, G. Izbirak, B. Vizvari, and J. Arkat. Complete coverage path planning for a multi-uav response system in post-earthquake assessment. *Robotics*, 5, 2016.

J. Nocedal and S. J. Wright. *Numerical Optimization.* Springer. Springer Series in Operations Research., 1999.

I. H. Osman and J. P. Kelly. Meta-heuristics: An overview. *Meta-heuristics: Theory and applications*, page 1–21, 1996.

A. Otto, N. Agatz, J. Campbell, B. Golden, and E. Pesch. Optimization approaches for civil applications of unmanned aerial vehicles (uavs) or aerial drones: A survey. *Networks.*, 72(4):411–458, 2018.

Z. Özyurt and D. Aksen. Solving the multi-depot location routing problem with lagrangian relaxation. *Operations Research. Computer Science Interfaces Series.*, 37: 125–144, 2007.

P. Parthanadee and R. Logendran. Periodic product distribution from multi depots under limited supplies. *IIE Transactions.*, 38(11):1009–1026, 2006.

A. Pessoa, M. P. de Aragão, and E. Uchoa. A robust branch-cut-and-price algorithm for the heterogeneous fleet vehicle routing problem. *Part of the Lecture Notes in Computer Science*, 4525, 2007.

M. Polacek, S. Benkner, K. F. Doerner, and R. F. Hartl. A cooperative and adaptive variable neighborhood search for the multi depot vehicle routing problem with time windows. *Business Research*, 1(2):207–218, 2008.

M. Polacek, R. F. Hartl, K. Doerner, and M. Reimann. A variable neighborhood search for the multi depot vehicle routing problem with time windows. *Journal of Heuristics*, 10(6):613–627, 2005.

L.S. Pontryagin, V.G. Boltyanskii, R.V. Gamkrelidze, and E.F. Mishchenko. *The mathematical theory of optimal processes.* A Pergamon Press Book, The Macmillan Co., 1964.

C. Ramirez-Atencia, G. Bello-Orgaz, M.D. R-Moreno, and D. Camacho. Solving complex multi-uav mission planning problems using multi-objective genetic algorithms. *Soft Computing.*, 21(17):4883–4900, 2017.

A. V. Rao. A survey of numerical methods for optimal control., 2010. In: *Advances in the Astronautical Sciences.*

A. V. Rao. Trajectory optimization: A survey. *Optimization and optimal control in automotive systems.*, page 3–21, 2014.

L. Rayleigh. The soaring of birds. *Springer Nature*, 27:534–535, 1883.

J. Renaud, G. Laporte, and F. F. Boctor. A tabu search heuristic for the multi depot vehicle routing problem. *Computers and Operations Research.*, 23(3):229–235, 1996.

F. Rhodes. On the metrics of chaudhuri, murthy and chaudhuri. *Pattern Recognition.*, 28:745–752, 1995.

R.G. Ribeiro, J.R.C. J´unior, L.P. Cota, T.A.M. Euz´ebio, and F.G. Guimar˜aes. Unmanned aerial vehicle location routing problem with charging stations for belt conveyor inspection system in the mining industry. *IEEE Transactions on Intelligent Transportation Systems.*, page 1–10, 2019.

H.S. Rodrigues, M.T.T. Monteiro, and D.F.M. Torres. Optimal control and numerical software: an overview., 2014. In: *Systems Theory: Perspectives, Applications and Developments.*

S. Roelofsen, A. Martinoli, and D. Gillet. 3d collision avoidance algorithm for unmanned aerial vehicles with limited field of view constraints. *Proceedings of the 55th IEEE conference on decision and control (CDC). IEEE.*, page 2555–2560, 2016.

J. Russell. *Performance and stability of aircraft.* Butterworth-Heinemann, 1996.

B. Ruzgiené, T. Berteška, S. Gečyte, E. Jakubauskiené, and V. Č. Aksamitauskas. The surface modelling based on uav photogrammetry and qualitative estimation. *Measurement.*, 73:619–627, 2015.

R. Rysdyk. Unmanned aerial vehicle path following for target observation in wind. *Journal of Guidance, Control and Dynamics.*, 29(5):1092–1100, 2006.

S. Salhi and G. Nagy. A cluster insertion heuristic for single and multiple depot vehicle routing problems with backhauling. *Journal of the Operational Research Society.*, 50(10):1034–1042, 1999.

S. Salhi and M. Sari. A multi-level composite heuristic for the multi-depot vehicle fleet mix problem. *European Journal of Operational Research.*, 103(1):95–112, 1997.

S. Salhi, S. R. Thangiah, and F. Rahman. A genetic clustering method of the multi-depot vehicle routing problem. *Artificial neural network and genetic algorithms.*, page 234–237, 1998.

Sky Sartorius. Oswald efficiency estimation function., 2023. MATLAB Central File Exchange. Retrieved July 9, 2023.

J. Schmidt and A. Fügenschuh. A two-time-level model for mission and flight planning of an inhomogeneous fleet of unmanned aerial vehicles. *Computational Optimization and Applications*, 85:293–335, 2023.

H. Shakhatreh, A. H. Sawalmeh, A. Al-Fuqaha, Z. Dou, E. Almaita, I. Khalil, N. S. Othman, A. Khreishah, and M. Guizani. Unmanned aerial vehicles (uavs): A survey on civil applications and key research challenges. *IEEE*, 2019.

M. Skok, D. Skrlec, and S. Krajcar. The non-fixed destination multiple depot capacitated vehicle routing problem and genetic algorithms. *In Proceedings of the 22nd international conference on information technology interfaces*, pages 403–408, 2000.

Skylines Flight Data. Flight data.

A. Sobester, I. P. Castro, H. Czerski, and N. Zapponi. Notes on meteorological balloon mission planning. *American Institute of Aeronautics and Astronautics (AIAA)*, 2013.

M. Soler, B. Zou, and M. Hansen. Flight trajectory design in the presence of contrails: Application of a multiphase mixed-integer optimal control approach. *Transportation Research Part C: Emerging Technologies*, 48:172–194, 2014.

R. F. Stengel. *Flight dynamics.* Princeton University Press, 2004.

C. Stöcker, A. Eltner, and P. Karrasch. Measuring gullies by synergetic application of uav and close range photogrammetry — a case study from andalusia, spain. *CATENA*, 132:1–11, 2015.

O. v. Stryk and R. Bulirsch. Direct and indirect methods for trajectory optimization. *Annals of Operations Research*, 37:357–373, 1992.

Z. Sun and S. Li. Optimal control of dynamic investment on inventory with stochastic demand. *IEEE, 2008 Chinese Control and Decision Conference*, 2008.

H.J. Sussmann and J.C. Willems. 300 years of optimal control: from the brachystochrone to the maximum principle. *IEEE Control Systems Magazine*, 17:32–44, 1997.

S. R. Thangiah and S. Salhi. Genetic clustering: An adaptive heuristic for the multi depot vehicle routing problem. *Applied Artificial Intelligence*, 15(4):361–383, 2001.

A. Thibbotuwawa, G. Bocewicz, G. Radzki, P. Nielsen, and Z. Banaszak. Uav mission planning resistant to weather uncertainty. *Sensors*, 20(2):515, 2020.

C. J. Ting and C. H. Chen. Combination of multiple ant colony system and simulated annealing for the multi depot vehicle routing problem with time windows. *Transportation Research Record: Journal of the Transportation Research Board*, 2089:85–92, 2009.

P. Toth and D. Vigo. An overview of vehicle routing problems. *The Vehicle Routing Problem. SIAM Monographs on Discrete Mathematics and Applications*, pages 1–26, 2002.

D. Tüzün and L. Burke. A two-phase tabu search approach to the location routing problem. *European Journal of Operational Research*, 116(1):87–99, 1999.

M. Uysal, A. S. Toprak, and N. Polat. Dem generation with uav photogrammetry and accuracy analysis in sahitler hill. *Measurement*, 73:539–543, 2015.

R.J. Vanderbei. Case studies in trajectory optimization: trains, planes, and other pastimes. *Optimization and Engineering*, 2(2):215–243, 2001.

D. S. Vianna, L. S. Ochi, and L. M. Drummond. A parallel hybrid evolutionary metaheuristic for the periodic vehicle routing problem. *Parallel and distributed processing. Lecture notes in computer science*, 1586:183–191, 1999.

T. Vidal, T. G. Crainic, M. Gendreau, N. Lahrichi, and W. Rei. A hybrid genetic algorithm for multidepot and periodic vehicle routing problems. *Operations Research*, 60:611–624, 2012.

T. Vidal, T. G. Crainic, M. Gendreau, and C. Prins. A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with timewindows. *Computers and Operations Research*, 40:475–489, 2013.

T. Vidal, T. G. Crainic, M. Gendreau, and C. Prins. Implicit depot assignments and rotations in vehicle routing heuristics. *European Journal of Operational Research*, 237(1):15–28, 2014.

Y. Wang, D. Zhang, Q. Liu, F. Shen, and L. H. Lee. Towards enhancing the last-mile delivery: An effective crowd-tasking model with scalable solutions. *Transportation Research Part E: Logistics and Transportation Review*, 93:279–293, 2016.

Z. Wang, J. Zhang, and X. Wang. A modified variable neighborhood search algorithm for the multi-depot vehicle routing problem with time windows. *Chinese Journal of Management Science*, 19(2):99–109, 2011.

H. Wu, N. Chio Cho, H. Bouadi, L. Zhong, and F. Mora-Camino. Dynamic programming for trajectory optimization of engine-out transportation aircraft. *In Control and Decision Conference (CCDC)*, page 98–103, 2012.

J. Xia, K. Wang, and S. Wang. Drone scheduling to monitor vessels in emission control areas. *Transportation Research Part B*, 119:174–196, 2019.

H. Yaman. Formulations and valid inequalities for the heterogeneous vehicle routing problem. *Mathematical Programming*, 106:365–390, 2006.

G. Yaohuang, X. Binglei, and G. Qiang. Overview of stochastic vehicle routing problems. *Journal of Southwest Jiaotong University (English Edition)*, 10(2):113–121, 2002.

C. Yuan, Y. Zhang, and Z. Liu. A survey on technologies for automatic forest fire monitoring, detection, and fighting using unmanned aerial vehicles and remote sensing techniques. *Canadian Journal of Forest Research*, 45:783–792, 2015.

G. N. Yücenur and N. C. Demirel. A new geometric shape-based genetic clustering algorithm for the multi-depot vehicle routing problem. *Expert Systems with Applications*, 38(9):11859–11865, 2011.

N. Zapponi. Astra high altitude balloon flight planner., 2013.

Y. Zhang, J. Chen, and L. Shen. Hybrid hierarchical trajectory planning for a fixed-wing ucav performing air-to-surface multi-target attack. *Journal of Systems Engineering and Electronics*, 23(4):536–552, 2012.

Y. J. Zhao. Optimal patterns of glider dynamic soaring. *Optimal Control Applications and Methods*, 25:67–89, 2004.

Z. Zhen, D. Xing, and C. Gao. Cooperative search-attack mission planning for multi-uav based on intelligent self-organized algorithm. *Aerospace Science and Technology*, 76: 402–411, 2018.