Glider Routing and Trajectory Optimisation in Disaster Assessment

Walton P. Coutinho[a], Jörg Fliege[a], Maria Battarra[b]

[a] *University of Southampton, University Road, Southampton, SO17 1BJ, United Kingdom*
{*w.p.coutinho, j.fliege*}*@soton.ac.uk*
[b] *University of Bath, Claverton Down, Bath, BA2 7AY, United Kingdom*
m.battarra@bath.ac.uk

Abstract

In this paper, we introduce the Glider Routing and Trajectory Optimisation Problem (GRTOP), the problem of finding optimal routes and trajectories for a fleet of gliders with the mission of surveying a set of locations. We propose a novel MINLP formulation for the GRTOP. In our approach, we consider the gliders' flight dynamics during the definition of the routes. In order to achieve better convergence, we linearise the gliders' dynamics and relax the dynamic constraints of our model, converting the proposed MINLP into a MISOCP. Several different discretisation techniques and solvers are compared. The formulation is tested on 180 randomly generated instances. In addition, we solve instances inspired by risk maps of flooding-prone cities across the UK.

Keywords: OR in disaster relief, unmanned gliders, routing, trajectory optimisation

1. Introduction

The Glider Routing and Trajectory Optimisation Problem (GRTOP) consists of finding optimal routes and trajectories for a fleet of unmanned aerial gliders. Gliders are Unmanned Aerial Vehicles (UAVs) without on-board propulsion. The gliders are due to visit a set of locations. This problem arises from disaster assessment applications, in which camera-equipped gliders survey a number of risky locations in a post-disaster situation. The collected information can be used to assess the severity of the effects in the aftermath of a disaster. This problem was motivated by discussions with the Royal National Lifeboat Institution, the UK leading charity providing flood rescue response, among other services.

Controlled powered drones have been used for raising emergency response in several scenarios, see for example, Chowdhury et al. (2017) and the recent Grenfell Tower disaster (Laville et al., 2017, June 15). However, these drones are expensive and often require experienced pilots to be operated. In aerial survey operations, a fleet of low cost ballon-launched autonomous gliders (Crispin, 2016) can be 3D printed (Keane et al., 2017) and do not require a specialised team to be operated. We believe this solution concept allows for rapid response to disasters.

In Coutinho et al. (2017), a review on UAV routing and UAV Trajectory Optimisation (TO) problems is provided. Moreover, the authors introduce a 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. The authors showed that there is a lack of research integrating UAV routing and trajectory optimisation.

Many Trajectory Optimisation Problems (TOPs) are non-convex in nature, e.g., most of the aerospace engineering-related problems (Conway, 2010; Shaw-Cortez & Frew, 2015). Several optimisation techniques based on Optimal Control (OC) and Non-linear Programming (NLP) have been developed to tackle TOPs. We refer the interested reader to Betts (2001) for a complete overview of those methods. NLP-based solution methods are known to be sensitive to initial guesses, i.e., convergence can be only ensured provided a proper initialisation (Zhao, 2004). Constructing a good set of initial guesses for flying vehicles often requires expertise on flight dynamics. In order to overcome such difficulties, the non-linear dynamics of UAVs is often linearised, e.g., Hajiyev et al. (2015), How et al. (2015) and Harris & Acikmese (2013).

In this paper, we propose a single-phase Mixed-Integer Non-linear Programming (MINLP) formulation for the GRTOP. Mixed-Integer Programming (MIP) has been already applied for solving TOPs and OC problems, see e.g., Keviczky et al. (2008), Soler et al. (2014), Fügenschuh & Müllenstedt (2015), Yuan et al. (2015) and Maolaaisha (2015). Our formulation takes into account the flight dynamics of the fleet of gliders during the definition of the routes. For this, we consider a linearisation of the gliders' Equations of Motions (EOMs) under special flight conditions. The resulting constraints are then relaxed and a penalisation term is added to the objective function. This allows for a more tractable formulation while keeping high quality solutions, i.e., with small error magnitudes. Next, we study several integration methods for solving the EOMs and test their performance when embedded in the MINLP formulation for the GRTOP.

We test our formulation with different integration methods. Moreover, we test alternative commercial Mixed-Integer Second-Order Cone Programming (MISOCP) solvers. In addition, we generate a number of real-life instances based on flood risk maps of cities in the UK, motivated by our application.

The remainder of this paper is organised as follows. In Section 2, we provide a brief introduction to flight dynamics and present the glider's EOMs. A MINLP for the GRTOP is introduced on Section 3. In Section 4, we linearise the glider's EOMs and present different integration methods for the linear dynamics. In Section 5, the resulting MISOCP formulation is then tested on several randomly generated and real-life-based instances. In Section 6, we conclude this work and highlight future research venues.

2. Gliders' Flight Dynamics

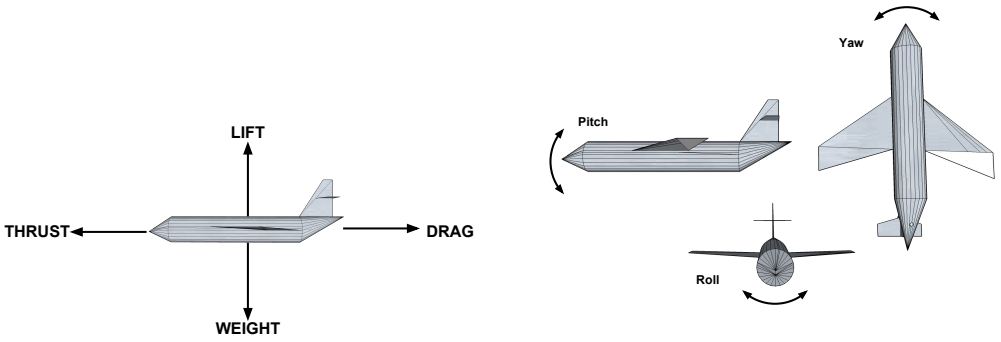
The four basic forces acting on an aircraft during flight are *thrust, drag, lift* and *weight*, these are depicted in Figure 1a. Thrust is the force generated by the on board propulsion of the aircraft itself. Drag is the air resistance acting upon the airplane's fuselage. Lift is the force generated by airflow through the control surfaces (flaps, ailerons, elevators and rudders), allowing the aircraft to fly. The weight models the

force pulling the aircraft to the centre of the Earth. For a glider, the thrust is absent, since there is no engine on board.

An aircraft is said to fly in *equilibrium* (a.k.a. steady-state flight) when the basic forces balance each other out. For instance, a powered steady-level flight can be achieved when lift equals weight and thrust equals drag. In simple terms, two types of equilibrium can be described, *static* and *dynamic*. Static equilibrium is related to the absence of velocity (static position). Dynamic equilibrium is related to the absence of acceleration (e.g., an object moving at constant velocity). In order to find steady-state conditions, we assume that gliders fly in dynamic equilibrium.

By activating the control surfaces of the aircraft, one can change its angular orientation. Angular orientation can be defined in terms of Euler angles, namely *pitch, yaw* and *roll* angles (here denoted by γ, φ and μ), with respect to a North-East-Down reference frame. Alternative representations can also be applied, e.g., *quartenions* and *rotation matrices*, but they will not be considered in this paper. Figure 1b depicts the planes of actuation of each Euler angle.

The control of an aircraft's horizontal orientation is usually described in terms of the Angle-of-attack (AoA). The AoA represents the difference between the angle of an aircraft's velocity vector and its flight path angle. We assume that the AoA can be written as a function of the *lift coefficient* (Cl). This is a common assumption in the flight dynamics literature (Stengel, 2004). In simple terms, the lift coefficient describes the amount of lift generated by an aircraft's wings. We refer the interested reader to the books by Russell (1996) and Stengel (2004) for a more detailed understanding of aircraft flight dynamics.



(a) Aerodynamic forces. (b) Planes of actuation of the Euler angles.

Figure 1: Relevant forces and angles in an aircraft's flight

2.1. Gliders' Equations of Motion

Dynamic soaring is a nature-inspired powerless flight technique that takes advantage of wind gradients. This technique 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 become popular for

autonomous gliding flight. Several studies have acknowledged the use of autonomous gliders for different purposes, see, e.g., Langelaan (2007), Chakrabarty & Langelaan (2011), and Crispin (2016).

The motion of an aerial glider can be modelled by a set of Ordinary Differential Equations (ODEs). One popular approach is employing a set of three-dimensional kinematics and dynamics EOMs for rigid bodies, like the ones presented by Zhao (2004). Their model includes the following assumptions: the circumference of the earth is negligible compared to the range of flight, the density of the air can be considered constant, the wind is stationary and the mass of the glider does not change during the flight. Without loss of generality, one can assume that only the horizontal component of the wind is present and it can be described by a linear profile as a function of the flight altitude.

Let us define the *state* of a glider at time $\tau \in \mathbb{R}_{\geq 0}$ as a *state vector* $\mathbf{y}(\tau) = (x(\tau), y(\tau), h(\tau), v(\tau), \gamma(\tau), \varphi(\tau))^\top$, where the first three components $x(t), y(t)$ and $z(t)$ of $\mathbf{y}(\tau)$ are the position of the glider in an Euclidean space and $v(\tau) \in \mathbb{R}_{\geq 0}$ is the relative velocity of the aircraft with respect to the wind velocity (*airspeed*). Finally, $\gamma(\tau) \in \mathbb{R}$ is the pitch angle and $\varphi(\tau) \in \mathbb{R}$ is the yaw angle. The *controls* (or input) to the system are represented by the *control vector* $\mathbf{u}(\tau) = (Cl(\tau), \mu(\tau))^\top$, where $Cl(\tau) \in \mathbb{R}$ is the lift coefficient and $\mu(\tau) \in \mathbb{R}$ the roll angle. All angles are defined over the *aerodynamic* (a.k.a *relative*) frame, .i.e., a system of geographical coordinates commonly used in aviation for representing states, see Fisch (2011). In the following, the notation "$\dot{}$" is used to represent time derivatives of the variables.

The glider model used in this paper is based on the designs proposed by Bower (2010) and Flanzer (2012). We computed the *aerodynamic coefficient* by using Equation (1), see Kroo (2001). The *Oswald factor* e is computed using the Matlab function available at Sartorius (2013). Finally, $b = 2.49$ is the glider's wing span and S the wing area.

$$k_A = \frac{1}{\pi e \frac{b^2}{S}} \tag{1}$$

The *wind strength* coefficient β has been chosen so as to linearly approximate the average wind gradient profile for the UK, as suggested by Drew et al. (2013), for a reference altitude of ≈ 500 metres. The remaining model parameters and their meaning are summarised in Table 1.

The EOMs of a glider can be expressed by Equations (2 - 12). For the sake of simplicity of notation, we have omitted the dependence on time τ from state, control and auxiliary variables.

$$m_g \dot{v} = -D - m_g g_e \sin\gamma - m\dot{U} \cos\gamma \sin\varphi \tag{2}$$

$$m_g v \dot{\gamma} = L \cos\mu - m_g g_e \cos\gamma + m_g \dot{U} \sin\gamma \sin\varphi \tag{3}$$

$$m_g v \cos\gamma \dot{\varphi} = L \sin\mu - m_g \dot{U} \cos\varphi \tag{4}$$

$$\dot{x} = v \cos\gamma \sin\varphi + U(h) \tag{5}$$

$$\dot{y} = v \cos\gamma \cos\varphi \tag{6}$$

$$\dot{h} = v \sin\gamma, \tag{7}$$

Table 1: Environmental and glider constants

Symbol	Value	Description	Unity (IS)
ρ	1.22543	Density of the air at sea level	(kg/m^3)
g_e	9.80665	Gravity of Earth at sea level	(m/s^2)
β	0.02500	Wind strength	(s^{-1})
C_{D0}	0.01730	Coefficient of drag at zero-lift	(dimensionless)
k_A	0.03200	Aerodynamic coefficient	(dimensionless)
m_g	1.99000	Mass of the glider	(kg)
S	0.48500	Glider's total wing area	(m^2)

where

$$D = \frac{1}{2}\rho S_w C_D v^2 \tag{8}$$

$$L = \frac{1}{2}\rho S_w C l v^2 \tag{9}$$

$$C_D = C_{D0} + k_A C l^2 \tag{10}$$

$$U(h) = \beta h \tag{11}$$

$$\dot{U} = \frac{dU(h)}{dt} = \beta v \sin\gamma. \tag{12}$$

In Equations (2 - 7), the wind's velocity is $U(h)$. The auxiliary variables D and L (Equations (8) and (9)) represent the drag and lift forces, respectively, acting on the glider.

By re-writing Equations (2 - 7) so that the time derivatives are isolated and by grouping the equations, one can obtain a compact representation of the system dynamics as follows:

$$\dot{\mathbf{y}}(\tau) = f(\mathbf{y}(\tau), \mathbf{u}(\tau), \tau), \tag{13}$$

where $f(\mathbf{y}(\tau), \mathbf{u}(\tau), \tau)$ corresponds to the right-hand-side of the system of ODEs (2 - 7).

3. Problem Definition

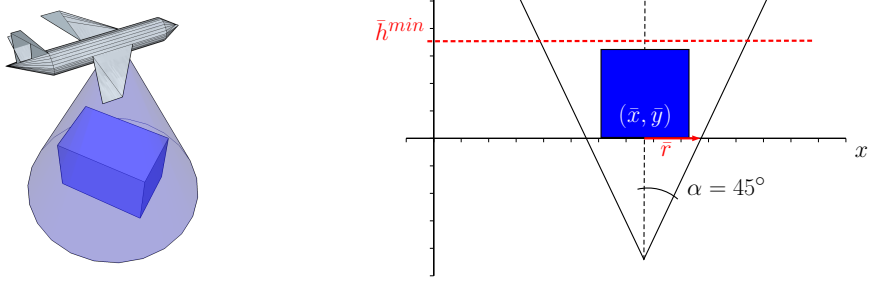
In the GRTOP, a fleet of balloon-lifted gliders is required to survey a number of points of interest, such as hospitals, schools and residential areas, in order to assess possible damages and people at risk in the aftermath of a disaster. Gliders are launched and are expected to land in one of the predetermined landing zones. The position of launch sites can be estimated using the tool "ASTRA High Altitude Balloon Flight Planner" proposed by Sobester et al. (2013), available at Zapponi (2013).

Each glider is equipped with a remote camera able to survey objects positioned within relative ranges. An inverted conic shape is adopted in order to model a cameras' field of view (Figure 2a). This type of geometry has also been used for UAV-camera systems in Ariyur & Fregene (2008), Roelofsen et al. (2016) and Nedjati et al. (2016). We assume that cameras are fixed to the body of the gliders, and we enforce the

gliders to fly in level-flight (or "flat") over a waypoint in order to properly photograph the desired object. For the sake of simplicity, we assume that the fleet of gliders is homogeneous and cameras have the same specifications. For each waypoint, the cameras' field of view allows us to define conic-like regions that must be visited by the gliders.

Figure 2b illustrates the geometric representation of a waypoint. In this picture, the object of interest corresponds to the blue box. Each waypoint is entirely described by $(\bar{x}_i, \bar{y}_i, \bar{r}_i, \underline{h}_i, \overline{h}_i), i \in V$, where $(\bar{x}_i, \bar{y}_i)$ represents the position of the object i in the xy plane. Parameter $\bar{r}_i > 0$ represents the radius of a circle in the xy plane enclosing the footprint of the object i. Parameters $\underline{h}_i$ and $\overline{h}_i$ denote the minimum and maximum heights in which object i can be photographed, respectively. The last two components define and constrain the quality of the pictures. Without loss of generality, we set the cameras' opening angle α to 45° while the points of interest are assumed to lie in the same xy plane (and therefore, their altitude $\bar{h}_i$ is neglected). Provided these assumptions, a glider flying at an altitude h can visit a waypoint i and take a good picture if it touches or enters the truncated cone i, i.e., $(x - \bar{x}_i)^2 + (y - \bar{y}_i)^2 \leq (h + \bar{r}_i) \tan \alpha = h + \bar{r}_i$.

Landing zones can be defined in a similar way. The tuple $(\tilde{x}_i, \tilde{y}_i, \tilde{r}_i)$ describes the geometry of a landing zone $i \in L$. The first three components define position on the xy plane and $\tilde{r}_i$ the radius of the landing site. Without loss of generality we will assume that $\tilde{h}_i$ equals 0 for all $i \in L$. The shape of landing zones consists of half-spheres with centres in the xy plane.



(a) Camera model. (b) Waypoint geometry.

Figure 2: Relevant forces and angles in an aircraft's flight

3.1. A Mixed-Integer Non-Linear Programming Formulation

In this section, a mathematical formulation for the GRTOP is proposed. In the following, we assume a fleet G of gliders is available at a known launching point 0. Let V represent a set of waypoints that have to be visited and L a set of possible landing sites. We are asked to find optimal routes and trajectories for the gliders in G such that the total mission time is minimised.

The motion of each glider is constrained by the system of ODEs (13). We assume that the initial position, denoted by $\mathbf{x}_o$, of each glider is known in advance, i.e., $\mathbf{x}_g(\tau = 0) = \mathbf{x}_o, \forall g \in G$, where the subvector $\mathbf{x}_g(\tau) = (x, y, h)$ represents the position of glider $g \in G$ at time τ, $\tau \in [\tau_o, \tau_f]$, $\tau_o = 0$ is the initial mission time and τ_f the maximum final mission time. We refer to the set of EOMs and initial conditions of each glider as their *dynamical system*.

The continuous dynamical system of each glider must be discretised in order to be used as constraints in an finite-dimensional optimisation problem. We define a time index set T by splitting the continuous time interval $[\tau_o, \tau_f]$ into $N-1$ time intervals of size η, where $N \in \mathbb{Z}$. Let $\mathbf{y}_{gt}$ and $\mathbf{u}_{gt}$ approximate the continuous state and control vectors $\mathbf{y}_g(\tau)$ and $\mathbf{u}_g(\tau)$, respectively, of glider $g \in G$ at the time instant t. A simple method for approximating the continuous dynamic system of glider g at the discrete time instants t with an associated error ε can be written as in Equation (14), based on Euler's method,

$$\mathbf{y}_{g(t+1)} = \mathbf{y}_{gt} + \eta f(\mathbf{y}_{gt}, \mathbf{u}_{gt}, t), t \in T \setminus \{N-1\}. \tag{14}$$

Other discretisation strategies will be discussed in Section 4. In our formulation, Constraints (14) are relaxed around the error term ε. Next, ε is added as a penalty to the objective function. This allows for a more tractable formulation, while maintaining the accuracy of trajectories. In Section 5, we show that the values of ε are very small, depending on the discretisation method that is employed.

We define the following binary decision variables:

$$a_{git} = \begin{cases} 1, & \text{if glider } g \text{ visits waypoint } i \text{ at time step } t \\ 0, & \text{otherwise.} \end{cases}$$

$$b_{git} = \begin{cases} 1, & \text{if glider } g \text{ lands in the landing site } i \text{ at time step } t \\ 0, & \text{otherwise.} \end{cases}$$

The GRTOP can be formally defined by the non-convex MINLP defined by Equations (15-38).

$$\min \quad \sum_{g \in G} \sum_{i \in L} \sum_{t \in T} t b_{git} + \varepsilon \tag{15}$$

$$\text{s.t.} \quad \sum_{\substack{g \in G \\ g \leq i}} \sum_{t \in T} a_{git} \geq 1, \forall i \in V \tag{16}$$

$$d_{git}^2 \geq (\bar{x}_i - x_{gt})^2 + (\bar{y}_i - y_{gt})^2, \forall g \in G, \forall i \in V, \forall t \in T \tag{17}$$

$$d_{git} \leq (h_{gt} + \bar{r}_i) + M(1 - a_{git}), \forall g \in G, \forall i \in V, \forall t \in T \tag{18}$$

$$h_{gt} \leq \bar{h}_i a_{git} + h_{ub}(1 - a_{git}), \forall g \in G, \forall i \in V, \forall t \in T \tag{19}$$

$$h_{g\tilde{t}} \geq h^{\min} a_{git} + h_{lb}(1 - a_{git}), \forall g \in G, \forall i \in V, \forall t \in T, \forall \tilde{t} \leq t \tag{20}$$

$$\gamma_{gt} \leq \hat{\gamma} a_{git} + \gamma_{ub}(1 - a_{git}), \forall g \in G, \forall i \in V, \forall t \in T \tag{21}$$

$$\gamma_{gt} \geq -\hat{\gamma} a_{git} + \gamma_{lb}(1 - a_{git}), \forall g \in G, \forall i \in V, \forall t \in T \tag{22}$$

$$\mu_{gt} \leq \hat{\mu} a_{git} + \mu_{ub}(1 - a_{git}), \forall g \in G, \forall i \in V, \forall t \in T \tag{23}$$

$$\mu_{gt} \geq -\hat{\mu} a_{git} + \mu_{lb}(1 - a_{git}), \forall g \in G, \forall i \in V, \forall t \in T \tag{24}$$

$$\sum_{i \in L} \sum_{t \in T} b_{git} = 1, \forall g \in G \tag{25}$$

$$\sum_{\substack{\tilde{t} \in T \\ \tilde{t} \leq t}} b_{gj\tilde{t}} \leq 1 - a_{git}, \forall g \in G, j \in L, i \in V, t \in T \tag{26}$$

$$r_{git}^2 \geq (\tilde{x}_i - x_{gt})^2 + (\tilde{y}_i - y_{gt})^2 + h_{gt}^2, \forall g \in G, i \in L, t \in T \tag{27}$$

$$r_{git} \leq \tilde{r}_i + M(1 - b_{git}), \forall g \in G, i \in L, t \in T \tag{28}$$

$$\mathbf{y}_{g,t+1} \leq \mathbf{y}_{gt} + \eta f(\mathbf{y}_{gt}, \mathbf{u}_{gt}, t) + \mathbf{1}\varepsilon, \forall g \in G, \forall t \in T \setminus \{N-1\} \tag{29}$$

$$\mathbf{y}_{g,t+1} \geq \mathbf{y}_{gt} + \eta f(\mathbf{y}_{gt}, \mathbf{u}_{gt}, t) - \mathbf{1}\varepsilon, \forall g \in G, \forall t \in T \setminus \{N-1\} \tag{30}$$

$$\mathbf{x}_{g0} = \mathbf{x}_o, \forall g \in G \tag{31}$$

$$\mathbf{y}_{lb} \leq \mathbf{y}_{gt} \leq \mathbf{y}_{ub}, \forall g \in G, \forall t \in T \tag{32}$$

$$\mathbf{u}_{lb} \leq \mathbf{u}_{gt} \leq \mathbf{u}_{ub}, \forall g \in G, \forall t \in T \tag{33}$$

$$a_{git} \in \{0, 1\}, \forall g \in G, \forall i \in V, \forall t \in T \tag{34}$$

$$b_{git} \in \{0, 1\}, \forall g \in G, \forall i \in L, \forall t \in T \tag{35}$$

$$d_{git}, r_{git} \in \mathbb{R}, \forall g \in G, \forall i \in V, \forall t \in T \tag{36}$$

$$\mathbf{y}_{gt} \in \mathbb{R}^6, \mathbf{u}_{gt} \in \mathbb{R}^2, \forall g \in G, \forall t \in T \tag{37}$$

$$\varepsilon \in \mathbb{R}_{\geq 0}. \tag{38}$$

The constants and "big-M" terms in the model have been defined as follows. The M constant has been computed as the space diagonal of the smallest cuboid containing the waypoints, landing sites and launching point. This is an upper bound on the distance between a glider and any waypoint and landing site at any time. The value $h^{\min}$ is defined as the minimum allowed flight altitude before landing, i.e., $h^{\min} = \max_i \{\underline{h}_i \mid i \in V\}$, assuming that $h^{\min} < \min_i \{\overline{h}_i \mid i \in V\} + C$, whith $C \in \mathbb{R}$ properly chosen. The values $\hat{\gamma} > 0$ and $\hat{\mu} > 0$ are small pitch and row values forcing the glider to fly "flat" in the cone covering an object. Finally, we denote by $\mathbf{1}$ a vector of ones with the same length as $\mathbf{y}_{gt}$.

The objective function (15) minimises a linear combination of the mission time and the discretisation error. The minimisation of the mission time forces the gliders to land as soon as possible. The second term of the objective function minimises the discretisation error that could be increased by landing too early. Constraints (16) state that every waypoint should be visited at least once. Constraints (17) and (18) make sure that gliders fly within the cone above each waypoint in order to take pictures. Constraints

(19) and (20) ensure that the gliders respect minimum and maximum surveying heights. Constraints (21) to (24) enforce gliders to be "flat" when taking pictures. Constraints (25) ensure that each glider lands in exactly one landing site and constraints (26) make sure that gliders do not land before all waypoints are visited. Constraints (27) and (28) guarantee that gliders land within pre-assigned regions. The dynamics of each glider are taken into account in Constraints (29) and (30). These constraints can be seen as a relaxation of the Equations (14). These equations allow for a more tractable optimisation problem and for a representation of the discretisation errors due to numerical integration methods. Constraints (29) and (30) are non-convex and therefore they make the model a MINLP formulation.

Equations (31) define the initial positions of each glider and Equations (32) and (33) define bounds on the state and control variables. Finally, Expressions (34) to (38) define the domain of the variables.

4. Linearisation and Discretisation of the Glider's Dynamics

The model (15-38) combines routing and trajectory optimisation decisions in a non-convex formulation. Local optimisation software for non-linear optimisation often requires high quality initial guesses. In order to avoid this issue, we transform the MINLP into a more tractable convex model by linearising the gliders' EOMs. This simplification is usually preferred in the literature when the dynamics are very non-linear (Ahmed et al., 2015; Hajiyev et al., 2015; How et al., 2015). In the following sections, we present the procedure for linearising the gliders' flight dynamics and the discretisation methods we applied for solving the resulting linear system.

4.1. Equilibrium flight and linearisation

A classic approach for linearising a system of ODEs consists of assuming the system operates in a *steady-state* condition, a.k.a. in equilibrium conditions. The equivalent linear system is then modelled assuming perturbations from this steady-state. Alternative techniques involve, for example, sequential and input-output linearisation. An interested reader can refer the books by Russell (1996) and Stengel (2004) for more methods of finding steady-state conditions.

We denote by $\mathbf{y}_{eq}$ and $\mathbf{u}_{eq}$ the steady-states and their respective controls of the glider dynamics. In a steady flight, the resultant forces and moments acting on the vehicle are zero. In other words, let us define $\mathbf{y}_{eq} = [x_{eq}, y_{eq}, z_{eq}, v_{eq}, \gamma_{eq}, \varphi_{eq}]^\top$ and $\mathbf{u}_{eq} = [Cl_{eq}, \mu_{eq}]^\top$ be the state and control variables such that

$$\dot{\mathbf{y}} = f(\mathbf{y}_{eq}, \mathbf{u}_{eq}, t) = \mathbf{0}. \tag{39}$$

In order to find an analytic solution to the Equation (39), the following assumptions are made as in Stengel (2004) and Langelaan (2007):

- Steady gliding flight, i.e., the equilibrium is achieved by matching the wind force with the drag, and the lift with the weight.

- The flight path and roll angles, γ and μ, respectively, are very small. Therefore, $\sin\gamma \approx \gamma$, $\cos\gamma \approx 1$, $\sin\mu \approx \mu$ and $\cos\mu \approx 1$. It is also assumed that $\varphi = 0$.
- The air mass is stable and the wind velocity is constant.
- The lift coefficient is constant.

From these assumptions, the EOMs (2-12) can be simplified to the following equations (40 - 42).

$$\dot{v} = -D/m_g - g\gamma = 0 \tag{40}$$

$$\dot{\varphi} = -L\mu/m_g v = 0 \tag{41}$$

$$\dot{\gamma} = L/m_g v - g/v = 0. \tag{42}$$

The optimal static lift coefficient is expected to minimise the drag-to-lift (D/L) ratio, therefore:

$$\frac{\partial(D/L)}{\partial Cl_{eq}} = -\frac{C_{D0}}{Cl_{eq}^2} + k_A = 0 \implies Cl_{eq} = \sqrt{\frac{C_{D0}}{k_A}}. \tag{43}$$

From Equations (40 - 42), the expressions of the remaining equilibrium states are found:

$$v_{eq} = \sqrt{\frac{2m_g g}{\rho S Cl_{eq}}}, \tag{44}$$

$$\gamma_{eq} = -2\sqrt{kC_{D0}}. \tag{45}$$

Let us define the following new variables as perturbations around state and control variables as $\delta\mathbf{y}(\tau) = \mathbf{y}(\tau) - \mathbf{y}_{eq}$ and $\delta\mathbf{u}(\tau) = \mathbf{u}(\tau) - \mathbf{u}_{eq}$, respectively. Applying first order Taylor's expansion to the system (13) around the steady-state conditions gives:

$$T(\mathbf{y}_{eq}, \mathbf{u}_{eq}, \delta\mathbf{y}, \delta\mathbf{u}, \tau) = f(\mathbf{y}_{eq}, \mathbf{u}_{eq}, \tau) + \frac{\partial f(\mathbf{y}_{eq}, \mathbf{u}_{eq}, \tau)}{\partial \mathbf{y}}\delta\mathbf{y}(\tau) + \frac{\partial f(\mathbf{y}_{eq}, \mathbf{u}_{eq}, \tau)}{\partial \mathbf{u}}\delta\mathbf{u}(\tau) + \dots. \tag{46}$$

By definition, the first term of equation (46) equals zero for the equilibrium condition. In this work, we discard the higher order terms of the Taylor's expansion. Matrices $A = \frac{\partial f(\mathbf{y}_{eq}, \mathbf{u}_{eq}, \tau)}{\partial \mathbf{y}}$ and $B = \frac{\partial f(\mathbf{y}_{eq}, \mathbf{u}_{eq}, \tau)}{\partial \mathbf{u}}$ denote the Jacobians of the dynamics (13) with respect to state and control variables. This linear system of ODEs can be re-written in a state-space form as in Equation (47)

$$\dot{\mathbf{y}} = A\delta\mathbf{y}(t) + B\delta\mathbf{u}(t), \tag{47}$$

where the system's matrices A and B have been found by computing the derivatives of the glider's EOM (2 - 12) at the steady-state conditions $\mathbf{y}_{eq}$ and $\mathbf{u}_{eq}$:

$$A = \begin{bmatrix} 0 & 0 & 0.025 & 0 & 0 & 9.44023 \\ 0 & 0 & 0 & 0.99889 & 0.44456 & 0 \\ 0 & 0 & 0 & -0.04704 & 9.44023 & 0 \\ 0 & 0 & 0 & -0.09766 & -9.79579 & 0.01110 \\ 0 & 0 & 0 & 0.21947 & -0.04881 & 0.00006 \\ 0 & 0 & 0 & 0 & -0.02506 & 0 \end{bmatrix}, \mathbf{y}_{eq} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 9.45068 \\ -0.04705 \\ 0 \end{bmatrix}, B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ -0.62763 & 0 \\ 1.41127 & 0 \\ 0 & 1.03882 \end{bmatrix}, \mathbf{u}_{eq} = \begin{bmatrix} 0.73527 \\ 0 \end{bmatrix}.$$

4.2. Discretisation methods

In this paper, we solve the gliders' EOMs by means of a *direct collocation* method (Betts, 2001). This is accomplished by discretising the linear EOMs represented by Equations (47) and adding the resulting expressions as constraints in the GRTOP.

In a direct collocation method, a continuous optimal control problem is discretised into a NLP by defining a grid of collocation points over a time interval $[\tau_o, \tau_f]$. Let us define N as the number of collocation points, where each time point t represents a time instant $\tau \in [\tau_o, \tau_f]$. In this paper, a uniform grid is adopted, as represented in the Equation $\tau = \tau_o + \eta t, t \in T, \eta = \frac{\tau_f - \tau_o}{N}$, where $T = \{0, \ldots, N-1\}$ is a set of collocation points. The value of η denotes the step size, which is constant in a uniform grid. Without loss of generality, we assume that $\tau_o = 0$.

Several integration schemes have been employed in order to discretise the linear system dynamics. All of them are defined over the same time grid. More specifically, we have applied a forward Euler method, a Trapezoidal method, two Adams-Bashforth methods and two Runge-Kutta methods (differing only by the controls interpolation). These approaches will be detailed in the next sections. More information about numerical methods for solving ODEs can be found, e.g., in the books by Betts (2001) and Butcher (2008).

The flight time interval $[\tau_o, \tau_f]$ and step size η will be fixed for all methods presented here, therefore the linearity of the EOMs is maintained. Due to the linearity of the EOMs, the MINLP formulation for the GRTOP, defined by Equations (15-38), becomes a MISOCP formulation that can be solved by commercial optimisation software without the need of initial guesses to converge.

4.2.1. Euler method

The forward Euler method is a first-order explicit numerical approach for solving ODEs (Butcher, 2008). Let $\mathbf{y}_t$ and $\mathbf{u}_t$ approximate $\mathbf{y}(\tau)$ and $\mathbf{u}(\tau)$, respectively, at time $\tau \in [\tau_o, \tau_f]$ with an associated error ε such that:

$$\dot{\mathbf{y}} \approx \frac{\mathbf{y}_{t+1} - \mathbf{y}_t}{\eta}.$$

By using this approximation on Equation (47) we can write:

$$\mathbf{y}_{t+1} = \mathbf{y}_t + \eta(A\delta\mathbf{y}(t) + B\delta\mathbf{u}(t)). \tag{48}$$

This dynamical system can be re-written in terms of the discretised state and control variables in Equation (49).

$$\mathbf{y}_{t+1} = (\eta A + I)\mathbf{y}_t + \eta B\mathbf{u}_t - \eta(A\mathbf{y}_{eq} + B\mathbf{u}_{eq}). \tag{49}$$

4.2.2. Trapezoidal method

The Trapezoidal method is a second-order implicit approach for solving ODEs based on the trapezoidal rule for computing integrals (Butcher, 2008). The equation defining the trapezoidal method can be derived

from both Runge-Kutta and Adams-Bashforth methods, and for the GRTOP it can be defined as follows:

$$\mathbf{y}_{t+1} = \mathbf{y}_t + \frac{1}{2}\eta(A\delta\mathbf{y}_t + B\delta\mathbf{u}_t + A\delta\mathbf{y}_{t+1} + B\delta\mathbf{u}_{t+1}). \tag{50}$$

By re-writing this system in terms of the original variables we find the discrete linear system (51).

$$\mathbf{y}_{t+1} = \mathbf{y}_t + \frac{1}{2}\eta(A(\mathbf{y}_{t+1} + \mathbf{y}_t) + B(\mathbf{u}_{t+1} + \mathbf{u}_t)) - \eta(A\mathbf{y}_{eq} + B\mathbf{u}_{eq}). \tag{51}$$

4.2.3. Runge-Kutta methods

The Runge-Kutta methods are a family of numerical methods for solving initial value problems. It consists of sampling intermediate values between subsequent time steps in order to cancel out lower order error terms (Butcher, 2008). In this paper, we apply a fourth-order Runge-Kutta method ($RK4$) in order to discretise the glider dynamics. The coefficients of the $RK4$ method for the linear glider dynamics are defined by Equations (52-55) in terms of the original discrete state and control variables.

$$k_1 = A(\mathbf{y}_t - \mathbf{y}_{eq}) + B(\mathbf{u}_t - \mathbf{u}_{eq}) \tag{52}$$

$$k_2 = A(\mathbf{y}_t + \frac{1}{2}\eta k_1 - \mathbf{y}_{eq}) + B(\hat{\mathbf{u}} - \mathbf{u}_{eq}) \tag{53}$$

$$k_3 = A(\mathbf{y}_t + \frac{1}{2}\eta k_2 - \mathbf{y}_{eq}) + B(\hat{\mathbf{u}} - \mathbf{u}_{eq}) \tag{54}$$

$$k_4 = A(\mathbf{y}_t + \eta k_3 - \mathbf{y}_{eq}) + B(\mathbf{u}_{t+1} - \mathbf{u}_{eq}), \tag{55}$$

where the auxiliary variable $\hat{\mathbf{u}}$ represents an interpolation of the control variables between time steps t and $t+1$. Here, two interpolation methods have been used. The first one consists of a linear interpolation (Equation (56)) and the second one consist of an exponential smoothing (Equation (57)), which weights the control history up to time step t.

$$\hat{\mathbf{u}} = \frac{\mathbf{u}_{t+1} + \mathbf{u}_t}{2} \tag{56}$$

$$\hat{\mathbf{u}} = \frac{1}{2}\sum_{k=0}^{k\leq t}\frac{1}{2^k}\mathbf{u}_{t-k} \tag{57}$$

Discretised EOMs can then be defined by Equation (58)

$$\mathbf{y}_{t+1} = \mathbf{y}_t + \frac{\eta}{6}(k_1 + 2k_2 + 2k_3 + k_4). \tag{58}$$

4.2.4. Adams-Bashforth methods

Linear multistep methods use information from previous steps to determine the current values of the state vector (Butcher, 2008). Unlike Runge-Kutta methods, multistep methods do not required interpolation of the control variables at intermediate steps since calculations are based on predetermined collocation points (in case of a direct collocation method). The Adams-Bashforth (AB) methods are a family of explicit integrators that compute the value of the current state from a linear combination of the values of previous

states. In this article, a third-order Adams-Bashforth method ($AB3$) and a fourth-order Adams-Bashforth method ($AB4$) are presented, in terms of the original state and control vectors, in the form of Equations (59) and (60).

$$\begin{aligned}\mathbf{y}_{t+3} = \mathbf{y}_{t+2} + \frac{1}{12}\eta(&23(A(\mathbf{y}_{t+2} - \mathbf{y}_{eq}) + B(\mathbf{u}_{t+2} - \mathbf{u}_{eq}))\\ &-16(A(\mathbf{y}_{t+1} - \mathbf{y}_{eq}) + B(\mathbf{u}_{t+1} - \mathbf{u}_{eq}))\\ &+5(A(\mathbf{y}_{t} - \mathbf{y}_{eq}) + B(\mathbf{u}_{t} - \mathbf{u}_{eq}))\end{aligned} \tag{59}$$

$$\begin{aligned}\mathbf{y}_{t+4} = \mathbf{y}_{t+3} + \frac{1}{24}\eta(&55(A(\mathbf{y}_{t+3} - \mathbf{y}_{eq}) + B(\mathbf{u}_{t+3} - \mathbf{u}_{eq}))\\ &-59(A(\mathbf{y}_{t+2} - \mathbf{y}_{eq}) + B(\mathbf{u}_{t+2} - \mathbf{u}_{eq}))\\ &+37(A(\mathbf{y}_{t+1} - \mathbf{y}_{eq}) + B(\mathbf{u}_{t+1} - \mathbf{u}_{eq}))\\ &-9(A(\mathbf{y}_{t} - \mathbf{y}_{eq}) + B(\mathbf{u}_{t} - \mathbf{u}_{eq}))\end{aligned} \tag{60}$$

Unlike single step methods, the third- and fourth-order AB methods require 3 and 4 initial values at the first iteration, respectively. This can be accomplished by running a single step method in order to find these initial values and then continuing the solution process with a multistep AB method. In this paper, we apply the Euler method in order to compute initial values.

5. Computational Experiments

The computational experiments described in the next sections have been implemented in the AMPL modelling language (version 20150516) and solved with CPLEX 12.7, Gurobi 7.0 and Xpress 8.0 in an Intel Core i7-4770 CPU with 3.40GHz and 16GB of RAM running under Linux Mint 17 64bits (kernel 3.13.0-24). The solvers were set to their standard configurations, with a time limit of 1 hour of execution each.

5.1. Generation of test instances

A number of test instances have been generated in the following way. First of all, we have considered instances having $n \in \{2, \ldots, 10\}$ waypoints, $m \in \{1, 2\}$ landing zones and $\lfloor n/2 \rfloor$ gliders. Two classes of instances have been created. The so-called *small ranged* instances (represented by "S" in the instances' name) have been defined over an area of $1km^2$ and the so-called *large range* instances (represented by "L" in the instances' name) over an area of $25km^2$. We assume a square shape for each area. In addition, each combination of number of waypoints and landing zones received 5 different random instances, so to have a diversity of test cases. These instances are grouped in our tables by the number of waypoints, e.g., the group GRTOP-S10 represents all small-ranged instances with 10 waypoints.

The geometry of waypoints and landing zones has been defined by the parameters in Table 2. Let $\mathcal{U}[a, b]$ denote the continuous uniform distribution from a to b. The launching altitude $\bar{z}_0$ has been chosen from

$\mathcal{U}[500, 600]$ for the small range instances and $\mathcal{U}[1000, 2000]$ for the large range instances, with the values of the limits given in metres. In Table 2, the value of R represents the square's side of the area, $R = 1km$ for the small-ranged instances and $R = 5km$ for the large range ones. All the other values in Table 2 are given in metres.

The positions of waypoints were not constrained, overlaps were allowed except when they generate duplicates and were assigned randomly within the boundaries of the area. However, landing zones were not allowed to overlap with waypoints.

Table 2: Limits of parameters defining the geometry of waypoints and landing zones in the generated instances.

Parameter	a	b	Parameter	a	b
$\bar{x}$	0	R	$\tilde{x}$	0	R
$\bar{y}$	0	R	$\tilde{y}$	0	R
$\bar{z}$	0	0	$\tilde{z}$	0	0
$\bar{r}$	10	25	$\tilde{r}$	10	25
$h^{\bar{m}in}$	50	100	$\bar{x}_0$	0	R
$h^{\bar{m}ax}$	200	300	$\bar{y}_0$	0	R

5.2. Comparing the performance of different solvers

Table 3 shows a summary of the results for the small range instances. In this table, *Group* denotes the nine groups of 10 instances (organised according to the number of waypoints in each instance). The performance of each solver is shown on columns *CPLEX*, *Gurobi* and *Xpress*. These solvers were chosen due to availability of licenses and their popularity in the research community (Mittelmann, 2017).

In Table 3, the *Status* column shows the tuple $(a, b, c) \in \mathbb{Z}^3$, representing the possible output statuses from AMPL as explained on AMPL (1998), where a denotes the number of `solved` instances, b represents the number of instances finished with status `solved?` and `limit`, and c represents the number of instances finished with status `failure`. Column *Error* corresponds to the normalised average discretisation error for each group. This has been calculated by averaging the error in each group and then dividing this average by the smallest error among the solvers for the same group. We use normalised average errors in order to make the comparison between different strategies easier. Column *Gap(%)* shows the average optimality gap at the end of the optimisation. Column *CPU(s)* represents the average computing time in seconds. Finally, column *Tree Size* shows the average number of explored branch-and-bound nodes for each group.

In order to test the solvers, we have chosen the Euler discretisation method. The number of collocation points N has been set to 30. The flight time horizon of each instance has been estimated by using the steady-state velocity and the largest range of that instance, i.e., $\tau_f = \max_{i \in V}\{\bar{x}_i, \bar{y}_i\}/v_{eq}$.

The solver Xpress outperforms the other solvers in most aspects. Best results overall in each column (except from the second) have been highlighted in boldface. Averages are shown for each solver. The relationship between CPU times and tree sizes indicates that Xpress is noticeably faster on processing the

second-order cone relaxations during the branch-and-bound search. Due to a large number of failures and worse performance for computing Second-Order Cone Programming (SOCP) relaxations, CPLEX presents smaller average tree sizes on most cases. For our problem, Xpress has been capable of solving 76% of the instances to optimality. As opposed to 32% and 24% of the instances that have been solved by CPLEX and Gurobi, respectively. One can notice that CPLEX has failed to find solutions to 15 problems in total. For the reasons exposed above, the solver Xpress has been chosen for the computational experiments presented in the next sections.

Table 3: Summary of the results for different solvers

Group	Status	Error	Gap(%)	CPU(s)	Tree
			CPLEX		
GRTOP-S2	(10,0,0)	**1.000**	**0.00%**	22.348	1274.30
GRTOP-S3	(8,0,2)	1.007	**0.00%**	99.081	2428.75
GRTOP-S4	(2,0,8)	1.034	**0.00%**	1199.889	26176.00
GRTOP-S5	(3,4,3)	**1.000**	18.57%	2171.628	12652.43
GRTOP-S6	(3,6,1)	1.403	32.33%	2997.753	20993.56
GRTOP-S7	(1,8,1)	1.736	35.78%	3175.324	13763.67
GRTOP-S8	(2,8,0)	4.017	51.30%	3397.745	12404.38
GRTOP-S9	(0,10,0)	3.285	54.50%	3600.500	11439.10
GRTOP-S10	(0,10,0)	3.230	65.90%	3600.589	12096.90
avg.	-	1.97	28.71%	2251.65	12581.01
			Gurobi		
GRTOP-S2	(10,0,0)	**1.000**	**0.00%**	44.125	**2831.40**
GRTOP-S3	(9,1,0)	1.028	2.70%	492.016	**7698.60**
GRTOP-S4	(1,9,0)	1.681	22.10%	3351.221	**40894.40**
GRTOP-S5	(1,9,0)	2.256	26.00%	3250.099	40628.50
GRTOP-S6	(1,9,0)	2.763	44.00%	3379.685	42566.10
GRTOP-S7	(0,10,0)	2.731	39.40%	3600.206	48218.50
GRTOP-S8	(0,10,0)	2.736	42.90%	3600.184	39544.80
GRTOP-S9	(0,10,0)	2.946	39.60%	3600.166	35009.00
GRTOP-S10	(0,10,0)	4.482	56.90%	3600.204	31395.40
avg.	-	2.40	30.40%	2768.66	32087.41
			Xpress		
GRTOP-S2	(10,0,0)	**1.000**	**0.00%**	**7.124**	800.60
GRTOP-S3	(10,0,0)	**1.000**	**0.00%**	**14.024**	3250.30
GRTOP-S4	(10,0,0)	**1.000**	**0.00%**	**104.985**	18475.50
GRTOP-S5	(9,1,0)	1.020	**0.10%**	**727.232**	**186538.20**
GRTOP-S6	(7,3,0)	**1.000**	**2.40%**	**1294.716**	**140235.50**
GRTOP-S7	(9,1,0)	**1.000**	**0.60%**	**1207.613**	**138074.90**
GRTOP-S8	(5,5,0)	**1.000**	**3.40%**	**2226.280**	**200746.40**
GRTOP-S9	(7,3,0)	**1.000**	**5.20%**	**2157.578**	**173651.90**
GRTOP-S10	(1,9,0)	**1.000**	**12.30%**	**3256.597**	**201550.40**
avg.	-	**1.00**	**2.67%**	**1221.79**	**118147.08**

5.3. Comparing the performance of different discretisation methods

In this section we compare the performance of the numerical integration methods presented in Section 4. Table 4 summarises the results for the small range instances. The remaining columns refer to the aforementioned discretisation methods, namely the *Euler* method, Trapezoidal method (TRP), the third- and fourth-order Adams-Bashforth methods, $AB3$ and $AB4$, respectively, and both versions of the fourth-order Runge-Kutta method $1RK4$ and $2RK4$, where the former refers to the RK4 method with linear control interpolation of Equation (56) and the latter to the RK4 method with the interpolation described in Equation (57). Table 4 has been subdivided for each algorithm performance measure, namely, Status, Error, Gap(%), CPU(s) and Tree, as in Table 3. Overall averages are shown at the end of each subdivision. The discretisation size and flight time horizon estimation have been kept the same as in the previous section.

From the results in Table 4, one can notice that the Euler and Trapezoidal methods are the most effective in solving instances to optimality, finding 68 (75.6%) and 69 (76.7%) optimal solutions, respectively, against 62 (68.9%) and 61 (67.8%) optimal solutions found by using the 1RK4 and 2RK4 methods. Together, they also produce smaller gaps for the instances that were not solved within the provided time limit. One can also verify that the ratio between the average number of branch-and-bound nodes and average CPU times is larger for these methods. This fact indicates that the Euler and Trapezoidal methods generate relaxations that are easier to solve during the tree search. On the other hand, Runge-Kutta methods outperform all the others in terms of discretisation error. The Runge-Kutta methods present error magnitudes that are up to 16 times smaller on average than the largest errors, at the expense of presenting higher average gaps for the instances that were not solved to optimality. There is a clear trade-off between computational performance and solution accuracy among the lower and higher order integration methods. Nonetheless, the third- and fourth-order Adams-Bashforth methods perform quite poorly for our problem, given the large error values and considerable gaps compared to the lower order methods.

We have extended our computational results for the smallest and largest instances of type "S" in Table 5. In this table, the first column shows the instance names and remaining columns present the error and CPU times for each discretisation methods presented in Section 4. The settings for the experiments shown in Table 5 remain the same as in the ones shown in Table 4.

From the results presented in Table 5, it can be seen that the magnitudes of the discretisation errors can be considered acceptable even for the lower order methods. To support our claim we have further investigated which state variables are most affected by the errors when using the Euler's method. This has been accomplished by performing two modifications in our formulation. The first one consists of using a vector $\boldsymbol{\varepsilon} \in \mathbb{R}^6$ to represent the error for each state variable in the dynamic equations. For example, the generic discretisation method presented in Constraints (29) and (30) can be re-written in the form of Constraints (61) and (62), respectively. It means that more variables will be added to the MISOCP

Table 4: Comparing discretisations

Group	Euler	TRP	AB3	AB4	1RK4	2RK4
			Status			
GRTOP-S2	(10,0,0)	(10,0,0)	(10,0,0)	(10,0,0)	(10,0,0)	(10,0,0)
GRTOP-S3	(10,0,0)	(10,0,0)	(10,0,0)	(10,0,0)	(10,0,0)	(10,0,0)
GRTOP-S4	(10,0,0)	(10,0,0)	(10,0,0)	(10,0,0)	(10,0,0)	(10,0,0)
GRTOP-S5	(9,1,0)	(10,0,0)	(10,0,0)	(9,1,0)	(8,2,0)	(7,3,0)
GRTOP-S6	(7,3,0)	(9,1,0)	(5,5,0)	(4,6,0)	(7,3,0)	(6,4,0)
GRTOP-S7	(9,1,0)	(8,2,0)	(5,5,0)	(3,7,0)	(6,4,0)	(6,4,0)
GRTOP-S8	(5,5,0)	(5,5,0)	(5,5,0)	(3,7,0)	(6,4,0)	(6,4,0)
GRTOP-S9	(7,3,0)	(6,4,0)	(2,8,0)	(1,9,0)	(3,7,0)	(3,7,0)
GRTOP-S10	(1,9,0)	(1,9,0)	(2,8,0)	(2,8,0)	(2,8,0)	(3,7,0)
			Error			
GRTOP-S2	6.076	6.071	6.040	6.079	**1.000**	1.077
GRTOP-S3	4.438	4.565	4.633	4.565	**1.000**	1.007
GRTOP-S4	6.825	6.963	7.134	7.207	**1.000**	1.045
GRTOP-S5	3.866	4.015	3.928	3.951	**1.000**	1.088
GRTOP-S6	9.525	9.454	9.518	9.789	**1.000**	1.309
GRTOP-S7	5.854	5.659	6.264	6.164	1.033	**1.000**
GRTOP-S8	14.396	14.608	15.477	16.025	1.028	**1.000**
GRTOP-S9	8.114	8.194	9.018	8.868	**1.000**	1.108
GRTOP-S10	12.478	13.235	12.937	12.141	**1.000**	1.382
avg.	7.952	8.085	8.328	8.310	**1.007**	1.113
			Gap(%)			
GRTOP-S2	**0.00%**	**0.00%**	**0.00%**	**0.00%**	**0.00%**	**0.00%**
GRTOP-S3	**0.00%**	**0.00%**	**0.00%**	**0.00%**	**0.00%**	**0.00%**
GRTOP-S4	**0.00%**	**0.00%**	**0.00%**	**0.00%**	**0.00%**	**0.00%**
GRTOP-S5	0.10%	**0.00%**	**0.00%**	0.90%	3.00%	0.40%
GRTOP-S6	2.40%	**0.50%**	4.10%	3.00%	6.50%	9.20%
GRTOP-S7	**0.60%**	1.30%	5.10%	9.50%	10.20%	3.90%
GRTOP-S8	3.40%	**3.20%**	6.20%	13.50%	9.80%	8.30%
GRTOP-S9	5.20%	**4.80%**	15.40%	12.30%	17.00%	15.60%
GRTOP-S10	**12.30%**	16.10%	16.50%	20.50%	34.20%	34.40%
avg.	**2.67%**	2.88%	5.26%	6.63%	8.97%	7.98%
			CPU(s)			
GRTOP-S2	7.12	9.09	13.62	20.55	6.96	**6.14**
GRTOP-S3	**14.02**	17.10	38.15	39.37	16.93	49.12
GRTOP-S4	**104.99**	151.17	338.46	714.07	306.09	530.00
GRTOP-S5	727.23	**146.84**	328.67	919.91	859.51	1282.27
GRTOP-S6	1294.72	**939.52**	2423.59	2778.01	1602.54	1878.44
GRTOP-S7	**1207.61**	1693.21	2517.73	3157.09	1952.83	1774.54
GRTOP-S8	2226.28	2159.28	2571.27	3071.74	**1813.62**	1958.09
GRTOP-S9	**2157.58**	2789.33	3276.83	3386.65	2959.64	2660.28
GRTOP-S10	3256.60	3277.13	3216.06	3274.96	3089.89	**2938.90**
avg.	**1221.79**	1242.52	1636.04	1929.15	1400.89	1453.09
			Tree			
GRTOP-S2	800.60	**491.50**	1536.60	2208.70	1349.80	605.00
GRTOP-S3	3250.30	**2772.60**	9364.90	6327.00	2937.30	13045.80
GRTOP-S4	**18475.50**	24006.80	47125.80	87958.70	74628.00	88431.30
GRTOP-S5	186538.20	**17292.70**	36597.40	85667.00	104206.00	197635.70
GRTOP-S6	140235.50	**88382.10**	165818.60	140593.00	114531.70	98954.10
GRTOP-S7	138074.90	157636.90	146485.20	133274.00	126049.60	**89096.20**
GRTOP-S8	200746.40	144065.70	122573.20	100995.50	95772.00	**95153.00**
GRTOP-S9	173651.90	174789.50	114985.30	**98945.70**	127546.60	100475.20
GRTOP-S10	201550.40	174476.00	110871.60	**82600.00**	112147.00	113462.60
avg.	**118147.08**	87101.53	83928.73	82063.29	84352.00	88539.88

formulation for the GRTOP.

$$\mathbf{y}_{g,t+1} \leq \mathbf{y}_{gt} + \eta f(\mathbf{y}_{gt}, \mathbf{u}_{gt}, t) + \boldsymbol{\varepsilon}, \forall g \in G, \forall t \in T \setminus \{N-1\} \tag{61}$$

$$\mathbf{y}_{g,t+1} \geq \mathbf{y}_{gt} + \eta f(\mathbf{y}_{gt}, \mathbf{u}_{gt}, t) - \boldsymbol{\varepsilon}, \forall g \in G, \forall t \in T \setminus \{N-1\} \tag{62}$$

The second modification follows from the first one as the objective function (15) needs to be re-written as in Equation (63). While the first term remains the same, the second term of the new objective function sums up the individual errors for each state variable.

$$\min \quad \sum_{g \in G} \sum_{i \in L} \sum_{t \in T} tb_{git} + \mathbf{1}^{\top} \boldsymbol{\varepsilon}. \tag{63}$$

Table 6 shows the results of this reformulation for a subset of instances (using Euler's method for discretising the dynamics). The main source of errors are the position components of the state vector. This can be explained by the fact that those components have the largest magnitude among all state variables, varying roughly between 0 and 1000. Even though the error associated to the translational dynamics is comparatively higher, they only represent a small fraction of the magnitudes of the position variables. For example, the error $\varepsilon_y = 11.58$ associated to the y variable for the small-ranged instance `grtopS_21_1` only represents 1.16% of the range of this state variable. The last two columns of Table 6 shows the results from the original formulation with a single error variable using Euler's method.

Table 5: Detailed discretisation error results

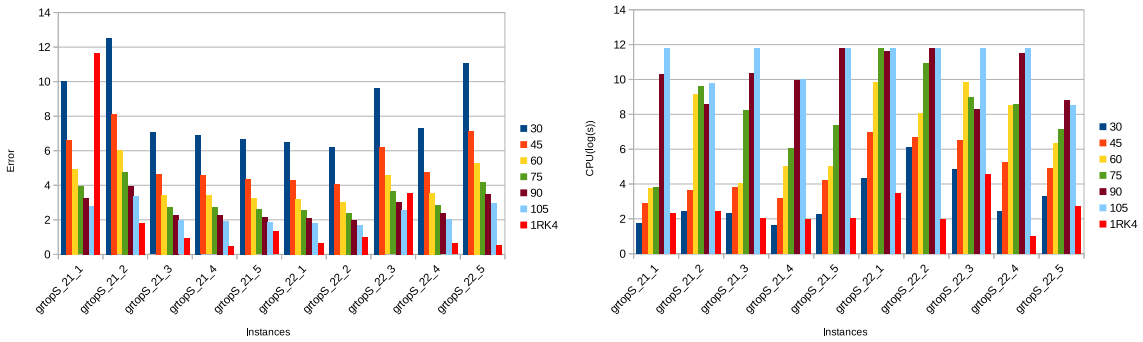
Name	Euler		TRP		AB3		AB4		1RK4		2RK4	
	error	CPU(s)	error	CPU(s)	error	CPU(s)	error	CPU(s)	error	CPU(s)	error	CPU(s)
grtopS_21_1	15.54	5.0	15.54	6.4	15.05	8.6	14.59	13.2	11.68	5.0	11.62	5.2
grtopS_21_2	18.63	6.4	16.57	8.5	15.46	12.4	16.53	19.0	1.83	5.3	1.66	4.7
grtopS_21_3	12.54	5.7	13.43	7.0	13.09	9.9	13.12	12.5	0.96	4.1	0.96	6.4
grtopS_21_4	12.02	3.7	12.96	6.6	12.74	8.4	13.76	12.1	0.50	3.9	0.56	3.0
grtopS_21_5	12.77	5.5	12.63	8.2	13.38	10.1	12.51	19.1	1.32	4.2	1.35	3.3
grtopS_22_1	12.19	7.9	14.23	13.3	12.95	23.8	12.83	22.8	0.68	11.0	1.48	12.8
grtopS_22_2	12.24	6.6	12.95	9.5	13.24	12.2	12.27	20.2	1.01	4.0	1.04	5.2
grtopS_22_3	13.07	11.5	13.12	12.5	14.64	23.9	13.72	41.3	3.55	23.6	4.54	9.9
grtopS_22_4	14.06	9.4	12.91	8.9	13.50	15.3	13.66	29.4	0.63	2.0	0.63	4.1
grtopS_22_5	14.95	9.6	13.58	10.0	13.15	11.6	15.10	15.9	0.56	6.5	0.63	7.0
avg.	**13.80**	**7.1**	**13.79**	**9.1**	**13.72**	**13.6**	**13.81**	**20.5**	**2.27**	**7.0**	**2.45**	**6.1**
grtopS_101_1	23.04	3601.6	26.40	3600.6	29.45	2603.9	25.66	2863.3	1.37	1732.4	1.24	2049.6
grtopS_101_2	23.17	3600.5	20.50	3600.4	20.78	3600.4	21.30	3600.3	1.24	3600.4	1.22	3600.4
grtopS_101_3	30.60	3600.5	25.93	3600.2	30.49	3600.5	23.68	3600.4	4.11	3600.5	3.50	3600.4
grtopS_101_4	23.39	160.3	23.27	367.2	24.80	753.3	24.36	1083.1	0.83	363.1	1.01	889.0
grtopS_101_5	24.94	3600.7	28.66	3600.5	26.31	3600.4	22.48	3600.4	2.22	3600.3	6.89	3600.3
grtopS_102_1	25.91	3600.5	22.28	3600.4	26.11	3600.5	24.67	3600.4	1.64	3600.3	0.78	3600.5
grtopS_102_2	24.60	3600.6	33.02	3600.6	27.00	3600.5	32.97	3600.4	2.85	3600.5	8.35	3600.7
grtopS_102_3	30.08	3600.5	26.27	3600.4	27.11	3600.2	25.29	3600.4	4.09	3600.4	1.69	1247.1
grtopS_102_4	22.01	3600.5	27.28	3600.5	21.67	3600.6	19.75	3600.4	1.07	3600.4	2.20	3600.6
grtopS_102_5	26.27	3600.3	35.81	3600.6	29.63	3600.3	26.99	3600.5	0.93	3600.6	1.26	3600.4
avg.	**25.40**	**3256.6**	**26.94**	**3277.1**	**26.33**	**3216.1**	**24.71**	**3275.0**	**2.04**	**3089.9**	**2.81**	**2938.9**

Finally, we analyse how the discretisation error and CPU times behave as the number of collocation points N increases. This results are shown in Figure 3. For this experiment, the following number of

Table 6: Analysis of the discretisation error for the individual components of the state vector.

Name	x	y	h	v	γ	φ	Sum	CPU(s)	ε	CPU(s)
grtopS_21_1	0.00	11.58	0.00	0.00	0.68	0.00	12.26	6.34	15.54	4.98
grtopS_21_2	0.00	0.01	0.00	0.00	1.22	0.14	1.37	10.67	18.63	6.36
grtopS_21_3	0.00	0.79	0.00	0.00	1.14	0.00	1.93	6.58	12.54	5.70
grtopS_21_4	0.00	0.38	0.00	0.00	0.66	0.00	1.04	2.77	12.02	3.68
grtopS_21_5	0.00	0.00	0.00	0.00	0.99	0.00	0.99	3.77	12.77	5.54
grtopS_22_1	0.00	0.00	0.00	0.00	0.75	0.00	0.75	6.47	12.19	7.85
grtopS_22_2	0.00	0.00	0.00	0.00	0.89	0.00	0.89	5.01	12.24	6.59
grtopS_22_3	0.00	0.59	0.00	0.00	1.20	0.04	1.83	12.80	13.07	11.50
grtopS_22_4	0.00	0.00	0.00	0.00	0.66	0.00	0.66	1.47	14.06	9.42
grtopS_22_5	0.00	0.47	0.00	0.00	0.59	0.35	1.40	5.97	14.95	9.62

collocation points have been adopted $N = \{30, 45, 60, 75, 90, 105\}$. The flight time interval has been computed as in the previous sections. Since the first term of the objective function (15) might affect the value of ε, we have eliminated this term from the objective function in order to properly assess the behaviour of the error term when varying the discretisation size. We highlight that these experiments were performed with the original MISOCP formulation (with a single error variable) and Euler's method. For the sake of comparison, we added a column (in red) representing the results from using the 1RK4 method with $N = 30$. Figure 3a shows the magnitude of the error for each number of collocation points on the small range instances of group GRTOP-S2 (expressed in the horizontal axis). One can notice that the discretisation error decreases as the number of collocation points increases, as expected. The opposite happens with the CPU times. In Figure 3b, these have been plotted in *log* scale.



(a) Magnitude of the error.

(b) CPU(s) in log scale.

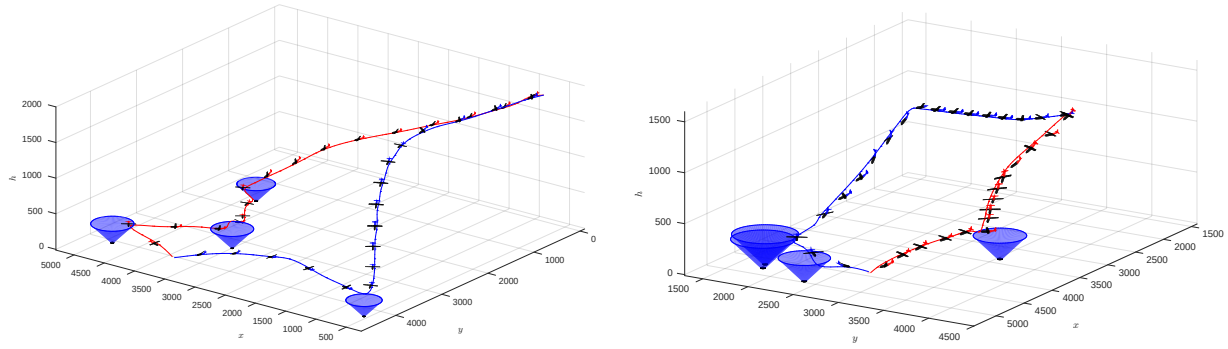
Figure 3: Behaviour of discretisation error and CPU times for several values of N.

5.4. Results for large range instances

In this section, we present the results of our model for a subset of large range instances. Here, we have chosen Euler's discretisation method due to its better average performance among the lower-order methods (Section 5.3). The number of collocation points N has been set to 60 and the flight time horizon τ_f has been chosen by the same procedure as described in Section 5.2.

In Table 7, we show the results of our model for instances with $3, 4, 5$ and 6 waypoints. One will notice that running times for the instances solved to optimality have substantially increased compared to the small range instances, as well as the gap for the instances were optimality could not be proved. This happens because of the larger discretisation size that we have applied. Our choice on larger discretisation sizes seeks to guarantee the convergence of the integration methods. We also highlight that discretisation errors remain small compared to the range of the instances. The largest error for the results in Table 7 (89.83) represents only 1.8% of the magnitude of position variables related to this instance.

Figure 4 depicts the optimal solution for two instances, `grtopL_41_5` and `grtopL_42_3`. For illustration purposes, we have approximated the trajectories between collocation points by natural cubic splines. In the solution of instance `grtopL_41_5`, the flight times are 156s and 152s, for the first and second gliders, respectively, and the step size equals 4.34s. In the solution of instance `grtopL_42_3`, the flight times are 139s and 85s, for the first and second gliders, respectively, and the step size equals 3.39s. We have also provided video animations of feasible and optimal solutions of several instances as supplementary material and through the website Coutinho (2017).



(a) Solution of `grtopL_41_5`.

(b) Solution of `grtopL_42_3`.

Figure 4: Depiction of the optimal solutions of two large range instances.

5.5. Routing and trajectory optimisation for disaster assessment

A number of instances based on UK cities prone to flooding has been created. They represent hypothetical flooding scenarios in the cities of Boston, Highbridge, London, Moore (Warrington) and Portsmouth. The so-called *UK instances* have been constructed as follows. We searched for flood risk-prone cities in the UK by using risk information maps provided at Government Digital Service (2017).

For each city, waypoints have been placed in locations with a high concentration of potential flooding victims, such as hospitals, schools, nurseries, residential areas, asylums and industries (which could become possible environmental hazards in a disaster situation), using Google maps. Next, the geographic coordinates of waypoints were collected and converted into Euclidean coordinates. The geometry of the waypoints and landing sites has been chosen in order to match the dimensions of the real locations.

Table 7: Results for a number of L instances

Name	Fleet	UB	LB	Error	CPU(s)	Tree	Gap(%)
grtopL_31_1	1	94.49	94.49	38.49	52.84	6559	0.00%
grtopL_31_2	1	83.43	83.43	28.43	99.77	10871	0.00%
grtopL_31_3	1	105.56	105.56	46.56	150.12	50459	0.00%
grtopL_31_4	1	135.85	135.85	76.85	35.71	4239	0.00%
grtopL_31_5	1	108.36	108.36	49.36	40.19	5323	0.00%
grtopL_32_1	1	73.59	73.59	24.59	167.19	15641	0.00%
grtopL_32_2	1	103.17	103.17	44.18	1098.67	262303	0.00%
grtopL_32_3	1	124.55	124.55	65.55	102.86	10618	0.00%
grtopL_32_4	1	95.88	75.81	37.88	3600.83	783809	21.00%
grtopL_32_5	1	91.79	91.79	33.80	260.57	38517	0.00%
grtopL_41_1	2	138.36	138.35	59.36	1943.31	115060	0.00%
grtopL_41_2	2	124.20	102.43	42.20	3600.47	185657	18.00%
grtopL_41_3	1	113.71	87.80	33.71	3600.79	334907	23.00%
grtopL_41_4	2	96.20	96.20	25.20	3080.92	306189	0.00%
grtopL_41_5	2	100.45	100.44	31.45	3174.82	192406	0.00%
grtopL_42_1	2	92.30	85.59	34.30	3600.32	122582	7.00%
grtopL_42_2	2	86.12	65.83	27.12	3600.28	99994	24.00%
grtopL_42_3	2	109.12	109.12	45.12	301.81	10467	0.00%
grtopL_42_4	2	153.41	97.07	71.41	3600.48	102879	37.00%
grtopL_42_5	1	129.94	129.94	54.94	2376.52	177920	0.00%
grtopL_51_1	2	160.34	99.19	82.34	3600.70	272174	38.00%
grtopL_51_2	2	122.69	107.50	43.69	3600.44	146891	12.00%
grtopL_51_3	2	154.80	146.37	72.80	3600.26	167945	5.00%
grtopL_51_4	2	116.08	105.20	28.08	3600.47	250171	9.00%
grtopL_51_5	2	137.60	88.60	58.60	3600.51	293565	36.00%
grtopL_52_1	2	148.99	148.99	59.99	1083.69	26234	0.00%
grtopL_52_2	2	93.96	79.15	41.96	3600.53	168447	16.00%
grtopL_52_3	2	119.81	85.83	36.81	3600.53	151592	28.00%
grtopL_52_4	2	141.57	95.62	51.57	3600.67	147272	32.00%
grtopL_52_5	2	159.38	91.63	75.38	3600.77	236883	43.00%
grtopL_61_1	3	168.25	107.26	37.25	3600.59	121569	36.00%
grtopL_61_2	3	126.14	112.61	49.14	3600.48	136656	11.00%
grtopL_61_3	3	182.83	79.91	89.83	3600.58	96375	56.00%
grtopL_61_4	2	159.76	112.63	73.76	3600.43	149115	30.00%
grtopL_61_5	2	121.34	80.89	43.34	3600.54	105711	33.00%
grtopL_62_1	2	141.84	81.54	50.84	3600.46	82085	43.00%
grtopL_62_2	2	167.52	78.92	88.52	3600.47	84507	53.00%
grtopL_62_3	3	193.56	154.27	78.56	3600.44	40665	20.00%
grtopL_62_4	3	163.16	78.92	40.16	3600.65	108157	52.00%
grtopL_62_5	2	153.53	91.17	50.53	3600.61	79393	41.00%

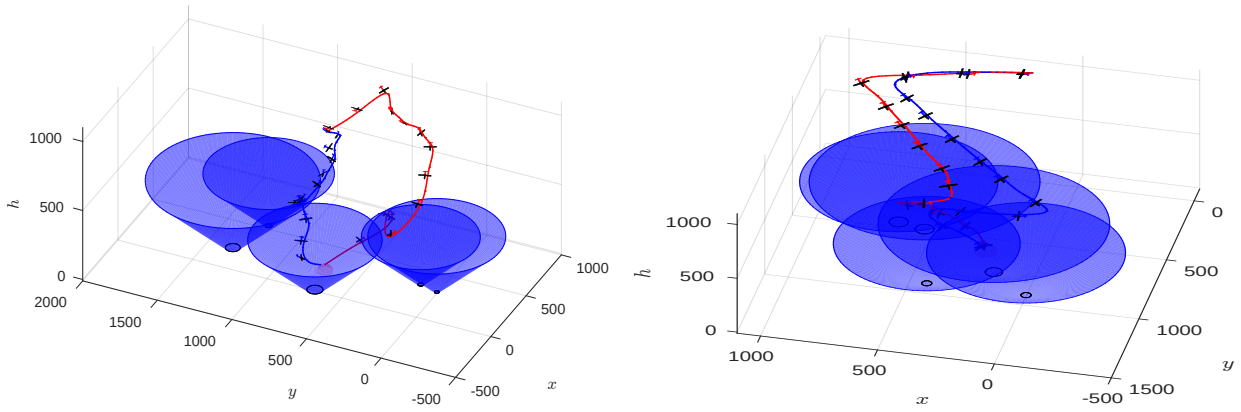
Table 8 shows the results of our experiments with the generated UK instances. In this table, column $\#W$ represents the number of waypoints in the instance. The remaining columns kept the same meaning as in Table 7. The number of landing sites for all instances was set to 1. The number of collocation points has been set to $N = 30$. Figure 5 depicts the solutions for 2 UK instances, namely, `grtopS_lond1` (London) and `grtopS_highb` (Highbridge).

6. Conclusion

In this paper we have tackled the GRTOP. This problem has been motivated by a disaster assessment application. In the GRTOP, we are asked to find optimal routes and trajectories for a fleet of unmanned gliders. The fleet of gliders is modelled by their EOMs, which consist of a set of ordinary differential equations. We propose a novel MINLP formulation for the GRTOP. In order to avoid a non-convex

Table 8: Results of the GRTOP formulation for the UK instances

Name	#W	Fleet	UB	LB	Error	CPU(s)	Tree	Gap(%)
grtop_bost1	7	3	73.03	73.03	34.03	263.95	9973	0.00%
grtop_bost2	7	2	75.38	75.15	31.38	3600.16	391381	0.00%
grtop_bost3	7	3	69.07	69.07	31.07	255.58	24063	0.00%
grtop_highb	5	2	57.08	57.08	23.08	34.23	830	0.00%
grtop_lond1	5	2	59.44	59.44	27.44	54.46	4923	0.00%
grtop_lond2	7	3	73.86	73.86	34.86	106.89	3059	0.00%
grtop_lond3	10	1	77.41	75.09	34.41	3600.42	241168	3.00%
grtop_lond4	7	3	75.48	75.48	30.48	142.17	5207	0.00%
grtop_moore	7	1	80.39	80.39	25.40	51.51	751	0.00%
grtop_prtsm	5	1	65.59	65.59	28.59	103.49	16355	0.00%



(a) Solution of `grtopS_lond1`.

(b) Solution of `grtopS_highb`.

Figure 5: Depiction of the optimal solutions of two UK instances.

formulation we linearise the gliders' EOM using a set of steady-state conditions. This reduces the MINLP into a more tractable MISOCP problem. In addition, we relax the resulting dynamic equations and penalise the corresponding error term in the objective function. We present several discretisation methods for the resulting linear dynamic equations.

In order to test our model, we have generated 180 random instances. We compared different commercial solvers on a subset of instances, namely, CPLEX, Gurobi and Xpress. Based on the results, Xpress was chosen for the next experiments. The second set of experiments was concerned about the discretisation methods and discretisation errors. Higher order integration methods were able of achieving smaller error magnitudes, but at the expense of CPU times. On the other hand, lower order methods typically reduced computation times, at the expense of solution accuracy. A detailed analysis has been carried out regarding the magnitude of the discretisation error. It was shown that the errors represent a small fraction of the magnitudes of the state variables and therefore are considered acceptable. Experiments on a subset of instances showed that the discretisation error is mostly due to the position variables, which have higher magnitude than the variables regarding angular orientation and airspeed. Finally, we studied the effect of

increasing the number of collocation points on the magnitudes of discretisation error and CPU times. In general, the trade-off between error magnitudes and CPU times becomes clear on the choice of N.

The results for large range instances showed that acceptable accuracy can also be achieved for long range flights even by employing lower-order discretisation methods. In order to guarantee the convergence of integration methods, the number of collocation points has to be increased. This has a direct influence on the number of large range instances solved to optimality. Finally, we present results for the so-called UK instances. These instances are created for disaster assessment in UK cities with high flooding risk. The model presents a good performance over instances from this group.

The accuracy of our solutions could be further improved by applying sequential linearisation at each node of the B&B tree, but this would dramatically increase the computation times. Further research could also focus on an adaptive integration method instead of the fixed time grid we have used. However, this improvements would also increase the CPU times.

Our formulation is capable to tackle a good number of test cases. However, for the instances with a larger number of waypoints and for higher discretisation sizes we were unable to prove optimality. This motivates the development of heuristic methods that should be able to find good solutions in small CPU times.

Acknowledgements

The authors would like to thank Jodie Walshe from RNLI and Dr. Andràs Sóbester from the Engineering Department at the University of Southampton for suggesting this problem. We also would like to thank Mrs. Fatine Mrabet for creating the UK instances. The first author received grants from CNPq [Grant no. 202241/2041-9].

References

Ahmed, E., Hafez, A., Ouda, A., Ahmed, H., & Abd-Elkader, H. (2015). Modelling of a Small Unmanned Aerial Vehicle. *Advances in Robotics & Automation*, *4*. doi:10.4172/2168-9695.1000126.

AMPL (1998). New in ampl: Statuses. http://www.ampl.com/NEW/statuses.html. Accessed: 25/05/2017.

Ariyur, K. B., & Fregene, K. O. (2008). Autonomous tracking of a ground vehicle by a UAV. In *2008 American Control Conference* (pp. 669–671). IEEE. doi:10.1109/ACC.2008.4586569.

Betts, J. T. (2001). *Practical methods for optimal control using nonlinear programming*. Advances in design and control. Philadelphia, PA: Society for Industrial and Applied Mathematics.

Bower, G. C. (2010). *Boundary Layer Dynamic Soaring for Autonomous Aircraft: Design and Validation*. Ph.D. Thesis Stanford University Stanford, California.

Butcher, J. C. (2008). *Numerical Methods for Ordinary Differential Equations*. Wiley.

Chakrabarty, A., & Langelaan, J. W. (2011). Energy-Based Long-Range Path Planning for Soaring-Capable Unmanned Aerial Vehicles. *Journal of Guidance, Control, and Dynamics*, *34*, 1002–1015. doi:10.2514/1.52738.

Chowdhury, S., Emelogu, A., Marufuzzaman, M., Nurre, S. G., & Bian, L. (2017). Drones for disaster response and relief operations: A continuous approximation model. *International Journal of Production Economics*, *188*, 167–184. doi:10.1016/j.ijpe.2017.03.024.

Conway, B. A. (2010). *Spacecraft Trajectory Optimization*. Cambridge University Press.

Coutinho, W. P. (2017). Glider routing. https://www.youtube.com/playlist?list=PL1mldBX67GxrUkuQZSzArTRLbYOowDD45. Accessed: 25/05/2017.

Coutinho, W. P., Battarra, M., & Fliege, J. (2017). *The Unmanned Aerial Vehicle Routing and Trajectory Optimisation Problem*. Technical Report. Available at http://www.optimization-online.org/DB_HTML/2017/06/6106.html.

Crispin, C. (2016). *Path Planning Algorithms for Atmospheric Science Applications of Autonomous Aircraft Systems*. Ph.D. Thesis University of Southampton Southampton, UK.

Drew, D. R., Barlow, J. F., & Lane, S. E. (2013). Observations of wind speed profiles over greater london, uk, using a doppler lidar. *Journal of Wind Engineering and Industrial Aerodynamics*, *121*, 98 – 105. doi:10.1016/j.jweia.2013.07.019.

Fisch, F. (2011). *Development of a Framework for the Solution of High-Fidelity Trajectory Optimization Problems and Bilevel Optimal Control Problems*. Ph.D. Thesis Technical University of Munich Munich, Germany.

Flanzer, T. (2012). *Robust Trajectory Optimisation and Control of a Dynamic Soaring Unmanned Aerial Vehicle.*. PhD. Thesis Stanford University Stanford.

Fügenschuh, A., & Müllenstedt, D. (2015). *Flight Planning for Unmanned Aerial Vehicles*. Technical Report AMOS #34(2015) Helmut Schmidt University / University of the Federal Armed Forces Hamburg. Available at `https://www.hsu-hh.de/download-1.5.1.php?brick_id=fyCYI55SQ6oFqe5e`.

Government Digital Service (2017). Long term flood risk information. `https://flood-warning-information.service.gov.uk/long-term-flood-risk/map?` Accessed: 20/05/2017.

Hajiyev, C., Soken, H. E., & Vural, S. Y. (2015). Equations of Motion for an Unmanned Aerial Vehicle. In *State Estimation and Control for Low-cost Unmanned Aerial Vehicles* (pp. 9–23). Springer International Publishing.

Harris, M. W., & Acikmese (2013). Maximum Divert for Planetary Landing Using Convex Optimization. *Journal of Optimization Theory and Applications*, *162*, 975–995. doi:`10.1007/s10957-013-0501-7`.

How, J. P., Frazzoli, E., & Chowdhary, G. V. (2015). Linear Flight Control Techniques for Unmanned Aerial Vehicles. In K. P. Valavanis, & G. J. Vachtsevanos (Eds.), *Handbook of Unmanned Aerial Vehicles* (pp. 529–576). Springer Netherlands.

Keane, A., Scanlan, J., Lock, A., Ferraro, M., Spillane, P., & Breen, J. (2017). Maritime flight trials of the southampton university laser sintered aircraft: Project albatross. *The Aeronautical Journal of the Royal Aeronautical Society*, (p. to appear). Available at `https://eprints.soton.ac.uk/411713/`.

Keviczky, T., Borrelli, F., Fregene, K., Godbole, D., & Balas, G. (2008). Decentralized Receding Horizon Control and Coordination of Autonomous Vehicle Formations. *IEEE Transactions on Control Systems Technology*, *16*, 19–33. doi:`10.1109/TCST.2007.903066`.

Kroo, I. (2001). *Aircraft Design: Synthesis and Analysis*. P.O. Box 20384, Stanford, CA 94309: Desktop Aeronautics. URL: `http://rahauav.com/Library/Design-performance/Aircraft%20Design,%20synthesis%20and%20analysis.pdf` version 0.99.

Langelaan, J. (2007). Long Distance/Duration trajectory optimization for small UAVs. In *AIAA Guidance, Navigation and Control Conference and Exhibit*. American Institute of Aeronautics and Astronautics.

Laville, S., Ross, A., Topping, A., Gayle, D., & Grierson, J. (2017, June 15). Grenfell tower: firefighters search overnight with toll expected to rise. https://www.theguardian.com/uk-news/2017/jun/14/fire-24-storey-grenfell-tower-block-white-city-latimer-road-london.

Maolaaisha, A. (2015). *Free-Flight Trajectory Optimization by Mixed Integer Programming*. Master Thesis University of Hamburg Hamburg.

Mittelmann, H. D. (2017). Mixed-integer socp benchmark. http://plato.asu.edu/ftp/misocp.html. Accessed: 14/07/2017.

Nedjati, A., Izbirak, G., Vizvari, B., & Arkat, J. (2016). Complete coverage path planning for a multi-uav response system in post-earthquake assessment. *Robotics*, *5*. URL: http://www.mdpi.com/2218-6581/5/4/26. doi:10.3390/robotics5040026.

Rayleigh, L. (1883). The Soaring of Birds. *Nature*, *27*, 534–535. doi:10.1038/028198a0.

Roelofsen, S., Martinoli, A., & Gillet, D. (2016). 3d collision avoidance algorithm for Unmanned Aerial Vehicles with limited field of view constraints. In *2016 IEEE 55th Conference on Decision and Control (CDC)* (pp. 2555–2560). doi:10.1109/CDC.2016.7798647.

Russell, J. (1996). *Performance and Stability of Aircraft*. Butterworth-Heinemann.

Sartorius, S. (2013). Oswald efficiency estimation function. https://uk.mathworks.com/matlabcentral/fileexchange/38800-oswald-efficiency-estimation-function?focused=3795877&tab=function.

Shaw-Cortez, W. E., & Frew, E. (2015). Efficient Trajectory Development for Small Unmanned Aircraft Dynamic Soaring Applications. *Journal of Guidance, Control, and Dynamics*, *38*, 519–523. doi:10.2514/1.G000543.

Sobester, A., Castro, I. P., Czerski, H., & Zapponi, N. (2013). Notes on Meteorological Balloon Mission Planning. In *AIAA Balloon Systems (BAL) Conference* (pp. 1–15). American Institute of Aeronautics and Astronautics.

Soler, M., Zou, B., & Hansen, M. (2014). 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. doi:10.1016/j.trc.2014.08.009.

Stengel, R. F. (2004). *Flight Dynamics*. Princeton University Press.

Yuan, C., Zhang, Y., & Liu, Z. (2015). 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. doi:10.1139/cjfr-2014-0347.

Zapponi, N. (2013). Astra high altitude balloon flight planner. http://astra-planner.soton.ac.uk/. Accessed: 21/02/2017.

Zhao, Y. J. (2004). Optimal patterns of glider dynamic soaring. *Optimal Control Applications and Methods*, *25*, 67–89. doi:10.1002/oca.739.