OR58 Annual Conference – Keynote Papers and Extended Abstracts
University of Portsmouth, Portsmouth, 6 – 8 September 2016
Transport and Mobility

A Conic-Programing-Based Approach for Trajectory Optimisation of Unmanned Gliders

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

[a] University of Southampton, Department of Mathematical Sciences, Southampton, United Kingdom
[b] University of Bath, School of Management, Bath, United Kingdom
w.p.coutinho@soton.ac.uk, j.fliege@soton.ac.uk, m.battarra@soton.ac.uk

Abstract

In recent years, employing Unmanned Aerial Vehicles (UAV) to collect data and make measurements has gained momentum. Often, the use of UAVs allows for a reduction in costs and improvements of other performance criteria. Those characteristics make UAVs suitable for disaster assessment, response and management. While the utilisation of powered UAVs has been broadly investigated in the literature, the employment of unpowered UAVs such as gliders has not been well explored. In fact, specialised control systems based on optimisation must be developed in order to guide such vehicles during their operations. In this paper we consider the problem of guiding a glider, along predetermined waypoints, in a wind field. We propose a Conic Programming Glider Trajectory Optimisation Problem, motivated by disaster assessment applications, and a solution framework. Some preliminary computational results are presented at the end of this work.

Keywords: Trajectory Optimisation, Unmanned Gliders, Quadratic Programming

1. Introduction

Unmanned Aerial Vehicles (UAV), a.k.a. drones, are aircrafts that do not need a human pilot on board. In general, these vehicles need to be controlled either by automated commands from an embedded computer and/or by a pilot operating a remote control. Unmanned aerial systems have been first used in special operations in which the presence of human pilots was either infeasible or life threatening (Beard and McLain, 2012). However, the popularity of UAVs in civil and commercial applications, e.g. in aerial reconnaissance, aerial forest fire detection, ship tracking, hazard management, disaster assessment and response, has recently increased (Ruzgienė et al. 2015).

The development of aerial survey systems based on powered UAVs is well advanced. However, unpowered UAV-based systems have not been extensively investigated in the literature. Gliders possess some advantages over other fixed-wing powered UAVs, the most significant one is unit cost, allowing the employment of larger and more flexible fleets. By using fleets of aerial gliders, disaster response teams would be able to quickly acquire data that would otherwise only be possible to gather *in situ*. This data would then help teams to focus their response efforts and apply their resources more efficiently and accurately. However, due to the absence of thrust, gliders are more difficult to control. Therefore, trajectory optimisation for such vehicles is more challenging.

This work aims to develop a general framework for coupling glider trajectory optimisation and routing problems in disaster assessment and response applications. We will refer to this general concept as Glider Routing and Trajectory Optimisation Problem. In this paper, we will focus on the trajectory optimisation aspects of the problem.

2. Glider Kinematics and Dynamics

In this section we present a linearisation of equations of motion based on Newton's second law for the motion of the glider under a horizontal wind shear around steady-state conditions.

2.1 Glider's equations of motion

Let $Y(t) = \big(x(t), y(t), z(t), v(t), \gamma(t), \varphi(t)\big)$ represent the state of the glider at time $t \in [t_0, t_f]$, where the first triple (x, y, z) denotes the glider's position, and the last three elements (v, γ, φ) denote its velocity, relative pitch and relative yaw angles, respectively. Let also $U(t) = \big(CL(t), \mu(t)\big)$ represent the glider's controls, where the first component denotes the lift-coefficient and the second element the relative bank-angle.

In this work, we use the set of Ordinary Differential Equations (ODEs) described in (Flanzer 2012) to model the motion of the glider. In general, these equations take the form $\dot{Y}(t) = f(Y(t), U(t), t)$, where the LHS corresponds to the time derivative of the state variable. Using this set of equations directly in an optimisation problem generally leads to non-convex formulations. In addition, very good initial guesses should be provided if one intends to solve NLP problems using a nonlinear optimisation solver (Zhao 2004). Hence, we will linearise these equations in order to make this problem more tractable.

The glider's equations of motion can be linearised using the first order Taylor's expansion around certain equilibrium conditions. These equilibrium conditions are defined under the assumption that all forces acting upon the glider are summing up to zero, thus causing the equilibrium state. After applying Taylor, we define a linear system of differential equations that describe the movement of the glider. Assuming that the linearisation error ε can be neglected and applying Euler's method to integrate the resulting linear equations of motion, we write this linear system in the form of Equation (1).

Let $t \in [t_0, t_f]$ be a time interval and $h = \big(t_f - t_0\big)/T$ be a predefined integration time step, where T is the number of time steps (so-called discretisation size). Let also Y_t and U_t represent approximations to $Y(t)$ and $U(t)$at time t. We can then replace the system of ODEs $\dot{Y}(t) = f(Y(t), U(t), t)$ with

$$Y_{t+1} = (hA + I)Y_t + hBU_t - h\big(AY_{eq} + BU_{eq}\big), \tag{1}$$

where the matrices A and B represent the Jacobians of this system of ODEs with respect to the state and control variables, respectively, and $I$is the identity matrix. In this work, Equation

(1) will be used as a constraint in order to to derive a conic programming problem to optimise the trajectory of a glider.

3. Glider Trajectory Optimisation Problem

In this section, we define a quadratic programming problem and an optimisation framework for computing trajectories over a set of targets.

3.1 Conically constrained model

Given a set of targets to be visited $V = \{0, \dots, n\}$, described by their position vector $(\bar{x}, \bar{y}, \bar{z})$, and a sequence $S = \left(i_{0,\dots}, i_q\right), i_q \in V, q < n$, we divide the trajectory into $q - 1$ paths according to each arc of the route S, such that in each path the distance between the glider and the final target of this arc at the last time step d_T is minimised. For each arc, $a = 1, \dots, q - 1$, the initial conditions at that arc are set equal to the final conditions of the previous arc.

In the model described by Equations (2), given a fixed flight time $\Delta t^a = t_f^a - t_0^a$ for the arc a, we aim to minimise the distance from the glider to the target at the final time step T. The second term in the objective function penalises unstable solutions, i.e., solutions that present a ringing effect (Vanderbei, 2001). The first constraint is a conic constraint that measures the distance from the glider to the final target at time step T. The second constraint accounts for the glider's dynamics. The set of inequalities involving $\pi_{1,}\pi_{2,}$ and π_3, and the subsequent two inequalities define a penalisation over solutions associated to harsh manoeuvres. The equality constraints involving Y^a and U^a represent the link between subsequent arcs of a sequence. Finally, the last four equations define the variables bounds and domains.

3.2 Flight time optimisation

Section 3.1 only considers to optimise the trajectories of the glider over a fixed time interval. We now consider an approach for finding a minimum flight time between two targets. For this, we use a bisection algorithm that solves in each arc a the trajectory optimisation problem given by Equations (2) iteratively for different values of Δt^a.

Equation (3) defines a single variable optimisation problem to minimise the flight time between every pair of subsequent targets in a sequence $S = \left(i_{0,\dots}, i_q\right), i_q \in V, q < n$, where Δt^a denotes the flight time of arc a and $f(\Delta t^a)$ denotes the objective function value of the problem defined by (2) at Δt^a. The bounds on the flight time that are necessary to solve this problem were calculated using the lower and upper bounds on the velocity of the glider and the Euclidean distances between the targets.

$$
\begin{gathered}
\min d_{T-1} + \varepsilon(\pi_1 + \pi_2 + \pi_3) \\
s.t. d_{T-1}^2 \geq \left(\bar{x}_{i_q} - x_{T-1}\right)^2 + \left(\bar{y}_{i_q} - y_{T-1}\right)^2 + \left(\bar{z}_{i_q} - z_{T-1}\right)^2, i_q \in V \\
Y_{t+1} = (hA + I)Y_t + hBU_t - h\left(AY_{eq} + BU_{eq}\right), t = 0, \ldots, T-1 \\
\pi_1 \geq \sum_{t=0}^{T-1} (CL_{t+1} - CL_t)^2 \\
\pi_2 \geq \sum_{t=0}^{T-1} (\mu_{t+1} - \mu_t)^2 \\
\pi_3 \geq h^2 \sum_{t=0}^{T-1} (\varphi_{t+1} + \varphi_t)^2 \Big/ 4 \\
(CL_{t+1} - CL_t)^2 < \xi, t = 0, \ldots, T-1 \\
(\mu_{t+1} - \mu_t)^2 < \xi, t = 0, \ldots, T-1 \\
Y_0^a = Y_{T-1}^{a-1} \\
U_0^a = U_{T-1}^{a-1} \\
0 \leq d_T \leq d_T^{ub}, d_t \in \mathbb{R} \\
Y_{lb} \leq Y_t \leq Y_{ub}, t = 0, \ldots, T-1 \\
U_{lb} \leq U_t \leq U_{ub}, t = 0, \ldots, T-1 \\
Y_t \in \mathbb{R}^6, U_t \in \mathbb{R}^{2,} \, t = 0, \ldots, T
\end{gathered}
\tag{2}
$$

$$
\min_{\Delta t^a} \Delta t^a + f(\Delta t^a), a = 1, \ldots, q-1 \tag{3}
$$

Finally, the total flight time of a route is then calculated as the sum of all flight times of each arc of the given sequence.

3.3 Computational Experiments

The algorithm was coded in C++ and the tests were carried out in an Intel Core i7 with 3.40 GHz and 16 GB of RAM running under Linux Mint 17. The conic problems were solved using CPLEX 12.4. Only a single thread was used in our experiments.

Table 1 shows the results of our approach for several random example problems. In this table, the first column shows the name of the instance, which also specifies the number of targets. The second column shows the resulting flight time for each instance. The last two columns show the CPU time in seconds and the total number of iterations (total number of calls to the optimisation solver) required to compute the whole trajectory.

Table 1 Result of experiments

Inst. Name	Flight time (s)	CPU (s)	QP Iter.
2targets	71.541	1.409	35
3targets	127.971	2.602	29
4targets	290.154	6.221	26
6targets	323.878	8.408	30
7targets	389.301	10.572	28
8targets	454.000	13.846	25

4. Conclusion

In this paper, we present an optimisation framework based on Conic Programming to compute trajectories for a glider over a set of sequenced targets. The equations of motion of the glider take into account all the forces and velocities that are natural to the dynamics and kinematics of an unpowered flight under the influence of the wind. In our algorithm, these equations were linearised in order to make the problem more tractable. The main advantages of linearising the equations of motion are that the trajectory optimisation can be performed by any appropriate optimisation software and the user does not need to provide an initial guess to the final solution. The trajectories computed by this algorithm are, in general, sub-optimal, since the whole flight path is divided into arcs and the trajectories for each arc are computed individually. Even though this approach only yields sub-optimal solutions, the CPU times required to solve each instance are very promising. Promising further avenues of research include embedding this trajectory optimisation framework into an algorithm for the Vehicle Routing Problem, so that routes can be generated automatically from a given set of targets.

5. References

Beard R and McLain T (2012). 'Small Unmanned Aircraft: Theory and Practice'. Princenton Press, Princenton, NJ.

Flanzer T (2012). 'Robust Trajectory Optimisation and Control of a Dynamic Soaring Unmanned Aerial Vehicle.' PhD. Thesis, Stanford: Stanford University.

Ruzgienė B, Tautvydas B, Silvija G, Edita J and Vladislovas Č A (2015). 'The Surface Modelling Based on UAV Photogrammetry and Qualitative Estimation'. Measurement 73 (September): 619–27.

Vanderbei R J (2001). 'Case Studies in Trajectory Optimization: Trains, Planes, and Other Pastimes'. Optimization and Engineering 2 (2): 215–43.

Zhao Y J (2004). 'Optimal Patterns of Glider Dynamic Soaring'. Optimal Control Applications and Methods 25 (2): 67–89. doi:10.1002/oca.739.