Numerical Techniques for Minimum-Time Routing on a Sphere with Realistic Winds

Andrei Marchidan* and Efstathios Bakolas†

The University of Texas at Austin, Austin, TX, 78712-1221, USA

I. Introduction

Flight time plays an important role in developing guidance and navigation planners for aircraft. It is one of the key factors affecting the direct operating costs and passenger experience. As a consequence, the airline industry aims to find better ways to generate minimum-time[1] and minimum-fuel[2,3] flight routes. In the latter case, fuel-optimal navigation can be performed by optimizing vertical profiles for climb, cruise and descent operations using a higher-dimensional model that includes gravitational effects,[3] while in the former case, minimum-time trajectories can be generated for a horizontal (constant altitude) profile where only a kinematic model is used.[1] While other methods can be employed to minimize both fuel and time,[4,5] this Note focuses only on a kinematic model for generating time-optimal routes.

The minimum-time trajectories of the aircraft navigation problem exploit the effect of wind in order to maximize ground speed. The problem of finding such trajectories can be formulated as a two-point boundary value problem,[6] for which many numerical solution techniques exist[7] and are already utilized by several navigation planners. These techniques can be classified as either direct or indirect methods for trajectory optimization.[8] On the one hand, direct methods are based on transcribing the optimal control problem into a nonlinear optimization problem whose solution is often subjected to numerical difficulties. On the other hand, indirect methods employ multiple shooting or collocation techniques, which typically require good initial approximations for the optimal trajectory.[7] Other navigation planners use algorithms that rely on the concept of the minimum-time function which provides the means to finding feasible and optimal trajectories. The computation of this function is usually performed with standard graph-search techniques, such as Dijkstra's algorithm or the A-star (A*) algorithm, with sampling-based search techniques such as

*Graduate Student, Department of Aerospace Engineering and Engineering Mechanics, andrei.marchidan@utexas.edu

†Assistant Professor, Department of Aerospace Engineering and Engineering Mechanics, AIAA Member, bakolas@austin.utexas.edu

RRTs (rapidly-exploring random trees), or with wavefront expansion techniques that interpolate solutions to linearized equations of motion or simple ordinary differential equations.[9] The reader may refer to articles by Eichhorn,[10] Lolla *et al.*,[11] Soulignac,[12] Chakrabarty and Langelaan,[13] Elston and Frew,[14] for examples and implementation of such methods in navigation applications.

The navigation problem can be formulated as a minimum-time problem for aircraft motion under the effects of a wind field. The goal of this problem is to analyze the wind effects on flight time for long distance routes. As such, the aircraft dynamics are not included in the analysis and only a kinematic model is used to describe the motion of the aircraft. This approach, which has been used previously in the literature,[1,15] is motivated by the large scale of time and distance required for medium and long-haul flights. The first one to consider and solve this problem for motion on a two-dimensional plane was Ernst Zermelo. The original solution can be found in Ref. [16] and its extension in the case when the vehicle moves on a sphere that is embedded in a three-dimensional space, was first presented by Jardin and Bryson.[1] In particular, the authors of Ref. [1] gave the necessary conditions for optimality, which allowed them to subsequently find nearly optimal solutions around a nominal path.

Both solutions from Refs. [1] and [16], can be used to generate optimal trajectories from one destination to another. However, by parameterizing these solutions throughout the state space of the problem, one is able to characterize the optimal synthesis and obtain a more general study of the effects of wind fields on flight time. In other words, one may look for minimum-time regions or surfaces rather than minimum-time trajectories in order to develop better planning strategies. The idea of computing the minimum-time function of the Zermelo navigation problem through propagation of the extremal (or candidate minimum-time) front for a planar surface was proposed by Bakolas[17,18] and Rhoads *et al.*[19] This extremal front expansion can be computed easily by exploiting the structure of the optimal control in order to parameterize the state space.

In this Note, a numerical technique is presented for the computation of the minimum-time function and for the generation of globally optimal trajectories. The technique uses the solution of the Zermelo navigation problem on a sphere in a wind field. While this solution alone assumes a candidate optimal trajectory, it does not guarantee global optimality. One of the challenges in generating globally optimal trajectories is the complex behavior of the wind fields, which may lead, in some cases, to the existence of multiple candidate optimal trajectories for the same target destination. The proposed algorithm guarantees globally optimal solutions by virtue of a simple, yet efficient and systematic method of recording the temporal and spatial evolution of the extremal front. Moreover, after computing the minimum-time function, minimum-time trajectories can be generated for any destination inside the reachable region with minimum computational effort. Previous works by Bijlsma[15] and Rhoads *et al*[19] deal with a similar problem on a planar surface. Specifically, Bijlsma presents a method to determine families of minimum-time routes. This method, however,

cannot handle satisfactorily cases in which discontinuities of the minimum-time function or singularities of the optimal synthesis appear. Rhoads *et al* provide a numerical study of the problem on the plane and an adaptive sampling and re-meshing procedure to deal with suboptimal solutions. While this algorithm achieves its goals for a planar surface, this Note presents an algorithm that is simpler to implement and that provides optimal solutions for a spherical surface, where the equations of motion are more complex.

The rest of this Note is organized as follows. In Section II the problem description and the solution of the Zermelo navigation problem on the sphere are presented. Section III describes the numerical algorithm for the computation of the minimum-time function. Numerical simulations for realistic wind fields are presented in Section IV, along with an investigation for the accuracy of these solutions. Section V concludes the Note with a summary of remarks.

II. Problem Description

Consider an aircraft flying at constant high altitude in a realistic wind field assuming a spherical model of the Earth. In order to account for the curvature of the Earth and for the effects of its shape on the motion of the aircraft, traveling speeds and times need to be comparable to those of medium and long-haul flights. Also, realistic flow fields are represented by spatially varying winds based on real data. These winds are only spatially varying since it is assumed that meteorological data is not updated continuously but rather at various time cycles, such as 1, 3 or 6 hours.[20]

Let time be denoted by $t \in \mathbb{R}$, the heading angle by $\psi \in [0, 2\pi]$, and the position of the aircraft by the vector $\boldsymbol{q} = [\phi, \theta]^T$, where $\phi \in [0, 2\pi]$ and $\theta \in [0, \pi]$ represent the spherical coordinates. Furthermore, consider $u(\phi, \theta)$ and $v(\phi, \theta)$ to be the components of the wind velocity in the corresponding spherical coordinates, at position $\boldsymbol{q}$ on the sphere. The kinematic model describing the motion of an aircraft on a sphere is represented by the following equations:

$$\dot{\phi} = \frac{1}{r \sin\theta}(u(\phi, \theta) + \bar{u}\cos\psi), \qquad \dot{\theta} = \frac{1}{r}(v(\phi, \theta) + \bar{u}\sin\psi), \tag{1}$$

where $\bar{u}$ and ψ are the airspeed and the heading angle of the aircraft, respectively, and r is the Earth's radius ($3,959$ miles). The control input is represented here by ψ, while the initial and terminal conditions are represented by the following variables:

$$\phi(0) = \phi_0, \quad \phi(t_f) = \phi_f, \qquad \theta(0) = \theta_0, \quad \theta(t_f) = \theta_f, \tag{2}$$

where t_f is the (free) final time of arrival.

The problem previously described is set up as a minimum-time control problem, which will be henceforth

called the Zermelo navigation problem on a sphere (ZNPS). Specifically, the problem can be stated as follows: Given an aircraft traveling with constant airspeed in a flow field on the surface of a sphere, determine the time history of all heading angles such that traveling time is minimized. A solution to this problem that exploits the structure of the optimal control for the kinematic model described by (1) is determined using a systematic numerical procedure. This allows one to generate the family of globally optimal trajectories and the minimum-time function.

A. Structure of the Optimal Heading Control

It can be shown by using standard optimal control techniques such as those used in Ref. [1] that the candidate optimal control that solves the ZNPS satisfies the following equation:

$$\dot{\psi} = \frac{1}{r}\left[\frac{\partial v}{\partial \phi}\csc\theta\sin^2\psi + \csc\theta\Big(v\cos\theta + \frac{\partial u}{\partial \phi} - \frac{\partial v}{\partial \theta}\sin\theta\Big)\sin\psi\cos\psi + \bar{u}\cot\theta\cos\psi + \cos^2\psi\Big(u\cot\theta - \frac{\partial u}{\partial \theta}\Big)\right]. \quad (3)$$

The initial condition for the above equation, $\psi(0) = \psi_0 \in [0, 2\pi]$, corresponds to the control input of the system described by (1) at time $t = 0$. It is interesting to note that equation (3) reduces to the planar heading equation obtained by Zermelo,[16] if θ is set to $\frac{\pi}{2}$ (known as the "navigation formula"). Also, a syntactically different result was obtained by Jardin and Bryson,[1] by considering θ to be measured from the line of the Equator rather than the South pole. One possible issue worth mentioning here is the existence of singularities at the spherical poles. These appear at points where equation (3) is undefined. This issue can be resolved with a coordinate rotation that shifts the position of the poles when solutions are needed in their proximity. Since the method used for this rotation has already been presented in Ref. [1], it will not be further investigated here.

B. Formulation of the Extremal Front and the Minimum-Time Function

Next, the fact that the structure of the candidate optimal input ψ satisfies (3) will be used to determine the extremal trajectories of the optimal control problem. Specifically, the extremal trajectories are computed by solving simultaneously the control input differential equation (3) and the kinematic differential equations (1) with initial conditions $\psi_0 \in [0, 2\pi]$, $t \in [0, t_f]$ and boundary conditions (2). As such, the trajectory starting from initial position $\boldsymbol{q}_0 = [\phi_0, \theta_0]^T$ at time $t = 0$, is generated by the kinematic model with the application of a control input that solves (3) for a particular value of ψ_0 and $t \in [0, t_f]$. This control input is called an extremal control input and is denoted by $\psi^\star(t; \psi_0)$. Next, consider a mapping $\varphi(\cdot; \boldsymbol{q}_0, \psi(\cdot)) : [0, t_f] \to \mathbb{S} := [0, 2\pi] \times [0, \pi]$ that maps the flight time, $t \in [0, t_f]$, to a reachable position, $\boldsymbol{q} \in \mathbb{S}$, where $\boldsymbol{q} = \varphi(t; \boldsymbol{q}_0, \psi(\cdot))$, for a given time history of the heading angle $\psi(\cdot)$, and a given initial state $\boldsymbol{q}_0$. Then, the trace of the extremal trajectory emanating from $\boldsymbol{q}_0$ and generated with the application of the candidate optimal input, $\psi^\star(\cdot; \psi_0)$, in the time

interval $[0, t_f]$ is the set $\{\boldsymbol{q} \in \mathbb{S} : \ \boldsymbol{q} = \varphi(t; \boldsymbol{q}_0, \psi^\star(\cdot; \psi_0)), t \in [0, t_f]\}$.

Remark [Solution Domain] The domain of the ZNPS is $\mathbb{S} = [0, 2\pi] \times [0, \pi]$ which only implies that solutions are represented by spherical coordinates. Therefore, the boundaries are identified symmetrically. By this, it is meant that, if $\boldsymbol{q} = [\phi, \theta]^T$ is a position in $\mathbb{S}$, then $[0, \theta]^T \sim [2\pi, \theta]^T$ for all $\theta \in [0, \pi]$, and $[\phi, 0]^T \sim [\phi, \pi]^T$ for all $\phi \in [0, 2\pi]$. (The identification operator $\sim$ has the same effect as the equivalence relation in topology; in other words, it declares that two points are identical and that the interval defined by them forms a closed loop.)[9]

The extremal front can now be defined as the set of points which can be reached in a given time by extremal trajectories of the ZNPS. As such, the extremal front is generated by the end points of the extremal trajectories at time t, that originate from the initial condition $\boldsymbol{q}_0$, and is denoted by $\mathcal{F}_t(\boldsymbol{q}_0)$, where $\mathcal{F}_t(\boldsymbol{q}_0) := \{\boldsymbol{q} \in \mathbb{S} : \boldsymbol{q} = \varphi(t; \boldsymbol{q}_0, \psi^\star(\cdot; \psi_0)), \ \psi_0 \in [0, 2\pi]\}$. Thus, as time progresses, this front expands in a ripple-like behavior, covering regions in space that are reachable by the aircraft.

Next, the minimum-time function of the ZNPS is defined by $F : \mathbb{S} \to \mathbb{R}_{\geq 0}$, where $\mathbb{R}_{\geq 0}$ denotes the set of non-negative real numbers (here, representing travel times). In particular, $F(\boldsymbol{q}; \boldsymbol{q}_0)$ gives the minimum time that the aircraft takes to reach destination $\boldsymbol{q} \in \mathbb{S}$ from a given point of departure $\boldsymbol{q}_0 \in \mathbb{S}$; that is,

$$F(\boldsymbol{q}; \boldsymbol{q}_0) := \min_{\psi_0 \in [0, 2\pi]} \{t \geq 0 : \quad \boldsymbol{q} = \varphi(t; \boldsymbol{q}_0, \psi^\star(\cdot; \psi_0))\}. \tag{4}$$

III. Numerical Method for the Computation of the Minimum-Time Function

In this section, a systematic method for the numerical solution of the minimum-time function is presented. Following this result, (approximated) minimum-time trajectories can be determined. The backbone of this method is the extremal front expansion. Using the definition given in Section II, the extremal front consists of an infinite number of states at each time $t > t_0$. Therefore, an approximation with an appropriate finite set is required for practical reasons. To this end, a finite discretization of the unit circle is introduced. This uses a partition $\mathcal{C}$ of the compact interval $[0, 2\pi - \epsilon]$, where $\mathcal{C} := \{\psi_0^{(1)}, \psi_0^{(2)}, \ldots, \psi_0^{(H)}\}$, such that $\psi_0^{(1)} = 0$ and $\psi_0^{(H)} = 2\pi - \epsilon$, for $0 < \epsilon \ll 1$. This implies that $\mathcal{C} = \{0, \epsilon, 2\epsilon, \ldots, H\epsilon\}$, where ϵ is the heading angle step size. Note that the superscript k, where $k \in \{1, 2, \ldots, H\}$, denotes the index of the discretized initial condition for the control input. Consequently, the extremal front is approximated by the set $\mathcal{F}_t(\boldsymbol{q}_0; \mathcal{C}) := \{\boldsymbol{q} \in \mathbb{S} : \boldsymbol{q} = \varphi(t; \boldsymbol{q}_0, \psi^\star(\cdot; \psi_0)), \psi_0 \in \mathcal{C}\}$, for any $t > t_0$.

It is well known that dynamical systems can exhibit unpredictable behavior far away from initial conditions. As such, the resolution of the heading angle step size may cause large separations between the end-points of trajectories, which may lead to poor approximations of the extremal front. This implies that any computational constants, such as ϵ or H, cannot be optimally set a priori. To resolve this issue, an adap-

tive bisection algorithm is used whenever the distance between the end-points of two trajectories, obtained from consecutive initial headings, is larger than a predefined upper bound $\bar{d}$. The distance is measured by computing the Euclidean norm for the two end-points. In this way, the extremal front takes into account the sensitivity of extremal trajectories to initial heading angles. A similar method was also used by Bijlsma.[15]

Next, the expansion of the extremal front is performed by integrating forward the extremal trajectories over a predefined time step t_s, using as initial conditions the values of their current end-points. For computational purposes, this expansion may stop when a specific destination or a time limit is reached. In the proposed algorithm, the latter condition will be used for the termination of the extremal front expansion procedure. The time limit is defined by the user and is denoted here by $T_{\max}$.

Furthermore, in order to form the minimum-time function, global optimality of the obtained numerical solutions needs to be ensured. This is challenging because complex behaviors of dynamical systems can lead to the appearance of anomalies in the extremal front of the minimum-time function, as was noted in Refs. [15] and [19]. In other words, the dynamical system may generate extremal trajectories which reach the same position at different times and cause discontinuities in the extremal front. To avoid this issue, a criterion for "filtering out" sub-optimal points from the extremal front is proposed. Sub-optimality results from the fact that $\boldsymbol{q} \in \mathcal{F}_{t_1}(\boldsymbol{q}_0)$ does not necessarily imply $F(\boldsymbol{q};\boldsymbol{q}_0) = t_1$. This might happen, as stated above, due to the existence of another time instant $t_2 < t_1$ for which $\boldsymbol{q} \in \mathcal{F}_{t_2}(\boldsymbol{q}_0)$, as well. It is obvious that, in this case, $F(\boldsymbol{q};\boldsymbol{q}_0) \leq t_2$. However, if $\boldsymbol{q} \in \mathcal{F}_{t_1}(\boldsymbol{q}_0)$ and $\boldsymbol{q} \notin \mathcal{F}_{t_2}(\boldsymbol{q}_0)$, for any $t_2 < t_1$, then one can conclude that $F(\boldsymbol{q};\boldsymbol{q}_0) = t_1$.

Now that a procedure is given for determining which solutions are contained in the minimum-time function, consider the following method for computing the numerical approximation of the function. First, consider a uniform spherical mesh grid $\Sigma(\texttt{lon}, \texttt{lat})$, where `lon` and `lat` represent the number of longitudinal and latitudinal mesh levels. Mesh points from the spherical grid are denoted by $\boldsymbol{q}_{ij}$, where i and j are the longitudinal and the latitudinal levels, respectively. Each mesh point is assigned a pair of two variables specific to its position on the sphere with respect to the extremal front: $\boldsymbol{q}_{ij} \leftarrow (\mathbf{B}, \tau)$, where $\mathbf{B}$ represents a boolean variable that determines if the point is reachable within the time interval $[0, T_{\max}]$, and τ represents a time variable which is assigned the minimum time required by the aircraft to reach position $\boldsymbol{q}_{ij}$. The pair is initialized for all mesh points with $(0, \infty)$, where 0 suggests that the mesh points are unvisited. Throughout this Note, $\boldsymbol{q}_{ij} \leftarrow (\mathbf{B}, \tau)$ corresponds to assigning values, while $\boldsymbol{q}_{ij} \rightarrow (\mathbf{B}, \tau)$ corresponds to reading the values already assigned.

Next, as the extremal front expands, it is overlaid on top of the mesh. Each unvisited mesh point found inside the interior of the closed curve approximated by $\mathcal{F}_t(\boldsymbol{q}_0; \mathcal{C})$, denoted by $\mathcal{I}_t$, is assigned the pair $(1, t)$: $\forall\, \boldsymbol{q}_{ij} \in \mathcal{I}_t$ if $\boldsymbol{q}_{ij} \rightarrow (0, \infty)$ then $\boldsymbol{q}_{ij} \leftarrow (1, t)$. Note that $\mathcal{I}_t$ represents, in other words, the reachable region

enclosed by the extremal front. This process marks the mesh point as "visited" and assigns to it the time value at which the extremal front first passes over it. In this way, every mesh point is assigned only one time value that corresponds to the minimum time that the aircraft requires to reach the respective position. Optimality is guaranteed by considering only the first crossing and capture of the mesh point by the extremal front. After all mesh points in $\mathcal{I}_t$ are marked, the expansion of the extremal front continues and the algorithm is repeated until the time limit $T_{\max}$ is reached.

The minimum-time function is determined by all the mesh points found inside the region covered by the expansion of the extremal front. This region, which is denoted by $\mathcal{I}$, is taken to be the union of the interiors of the closed curves approximated by $\mathcal{F}_t(\boldsymbol{q}_0;\mathcal{C})$, that is, $\mathcal{I} := \bigcup_{t\in[0,T_{\max}]} \mathcal{I}_t$. If time values are required in between mesh points, standard bilinear interpolation is used to get the interpolated time value denoted by $t^\star$. This means that the precision of the approximation to the minimum-time function depends both on the size of the mesh and on the integration time step t_s. As a result, the approximation of the minimum-time function is determined by:

$$\hat{F}(\boldsymbol{q};\boldsymbol{q}_0,T_{\max}) := \begin{cases} t = F(\boldsymbol{q};\boldsymbol{q}_0), & \text{if } \exists\, i,j \text{ such that } \boldsymbol{q}=\boldsymbol{q}_{ij},\ \boldsymbol{q}_{ij}\to(1,t), \text{ and } F(\boldsymbol{q};\boldsymbol{q}_0)\le T_{\max} \\ t^\star, & \text{if } \exists\, t\le T_{\max} \text{ such that } \boldsymbol{q}\in\mathcal{I} \text{ and } \boldsymbol{q}\neq\boldsymbol{q}_{ij}, \forall\, i,j \\ \infty, & \text{otherwise} \end{cases} \tag{5}$$

The main steps of the systematic procedure are summarized as follows:

Step 1: Compute the extremal front $\mathcal{F}_t(\boldsymbol{q}_0;\mathcal{C})$ by solving the ZNPS for the given time step and initial conditions;

Step 2: Check the distance between the end-points of two "consecutive" trajectories to determine if bisection is necessary:

$$\|\boldsymbol{q}^{(k)} - \boldsymbol{q}^{(k-1)}\| \ge \bar{d} \tag{6}$$

where $\boldsymbol{q}^{(k)} = \varphi(t;\boldsymbol{q}_0,\psi^\star(\cdot;\psi_0^{(k)}))$ and $\boldsymbol{q}^{(k-1)} = \varphi(t;\boldsymbol{q}_0,\psi^\star(\cdot;\psi_0^{(k-1)}))$ are the end-points of two trajectories obtained from "consecutive" initial conditions for the control heading, $\psi_0^{(k)}$ and $\psi_0^{(k-1)}$. If (6) is satisfied, perform bisection, otherwise go to step 3;

Bisection Step 2.1: Increase the number of initial heading angles, $H \leftarrow H+1$, through bisection: $\hat{\psi}_0^{(k)} \leftarrow (\psi_0^{(k)} - \psi_0^{(k-1)})/2$ and $\mathcal{C} \leftarrow \mathcal{C} \cup \{\hat{\psi}_0^{(k)}\}$;

Bisection Step 2.2: Compute the corresponding extremal trajectories $\boldsymbol{q}^{(k)} = \varphi(t;\boldsymbol{q}_0,\psi^\star(\cdot;\hat{\psi}_0^{(k)}))$ to

increase the fidelity of the extremal front for each time step until current time t is reached;

Step 3: Assign the current time value t to any unvisited mesh points inside the region enclosed by the extremal front : $\forall\, \boldsymbol{q}_{ij} \in \mathcal{I}_t$ for which $\boldsymbol{q}_{ij} \to (0, \infty)$, $\boldsymbol{q}_{ij} \leftarrow (1, t)$. In this way, the approximation of the minimum-time function, $\hat{F}$, is assigned the value t for all the mesh points that are contained in the region enclosed by the extremal front and have not been already assigned a value less than t;

Step 4: Repeat steps 1-3 for each time step t_s until $T_{\max}$ is reached, while updating every time the initial conditions for the integration of the ordinary differential equations (3) and (1) that lead to the extremal trajectory, to avoid redundancy in computation.

To illustrate more clearly the generation of the extremal front, consider Figure 1. Here it can be observed how the extremal front, which consists of the end points of the extremal trajectories, is expanded. The proposed method forms the minimum-time function incrementally, by considering all the mesh points enclosed by the extremal front. Mesh points or nodes that have not been visited yet are illustrated by simple dots while visited nodes are described by asterisks.

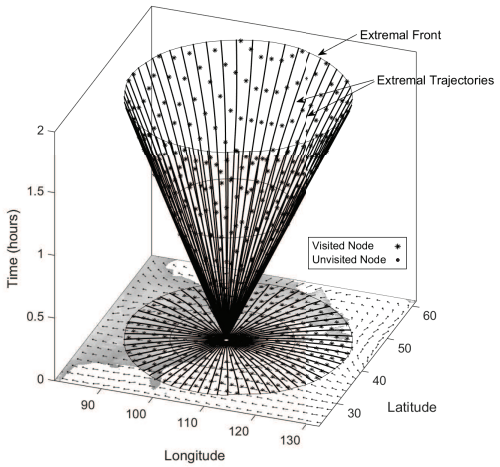

Figure 1. Visualization of the extremal front expansion in time with nodal partition for visited and unvisited mesh points.

Lastly, minimum-time trajectories can be obtained through a simple search on the extremal front. If the target destination, denoted by $\boldsymbol{q}_f$, does not correspond to a point on the extremal front, the initial condition for the optimal heading $\psi(0)$ is unknown, since it is not in the set $\mathcal{C}$ which defines the extremal front. However, knowing that the minimum time required to reach the destination is $t^\star = \hat{F}(\boldsymbol{q}_f; \boldsymbol{q}_0, T_{\max})$, one can look for the points near the target destination $\boldsymbol{q}_f$ only on the extremal front, $\mathcal{F}_t(\boldsymbol{q}_0; \mathcal{C})$, for which expansion time t is approximately equal to $t^\star$. The initial conditions used to obtain these extremal front points can be interpolated to approximate the initial condition $\psi(0)$ which gives $\varphi(t; \boldsymbol{q}_0, \psi^\star(\cdot; \psi(0))) \approx \boldsymbol{q}_f$. Thus, an approximation of the globally optimal trajectory to $\boldsymbol{q}_f$ can be determined by solving the ZNPS

with the interpolated initial condition $\psi(0)$. Furthermore, even though there might be different times when the extremal front encloses the destination, only the minimum time will be considered due to the search restriction presented herein. By doing so, any issues that may appear due to discontinuities of the minimum-time function or singularities of the optimal synthesis can be handled, and thus only optimal solutions are obtained.

IV. Numerical Simulations and Results

In this section, simulation results for the extremal front algorithm are presented. To demonstrate the procedure's applicability, realistic winds obtained from NOAA's Satellite and Information NOMADS[20] open database were used. The structure of the winds can be observed in Figure 2 as a direction vector field representing wind data extracted from `10/28/2014 0:00:00 AM`. The maximum wind speed from this data was about 130 mph. Aircraft airspeed was set constant at 560 mph. This causes a decrease in travel time, compared to the real travel time, since take-off and landing are not considered. Optimal trajectories were generated and their travel times were computed for the following flight routes: JFK to SFO in 4 hours and 50 minutes, JFK to MIA in 2 hours and 11 minutes, JFK to JNB in 13 hours and 25 minutes, and JFK to AUH in 11 hours and 27 minutes. These times show a decrease in total flight time by more than an hour compared to the actual flight times, which can be obtained from any airline website. This is partly the result of the simplified model which does not account for landing and take off and partly the result of more complex constraints that airlines try to satisfy. Considering these facts, it can be concluded that the presented algorithm is able to exploit the structure of winds in order to produce flight routes that are at least comparable if not faster than the ones currently used by airlines. The generated results are displayed in Figures 2 and 3.

To demonstrate the ability of the proposed algorithm to handle discontinuities of the minimum-time function or singularities of the optimal synthesis, a simulation which excludes the systematic method for eliminating sub-optimal solutions is performed. Furthermore, to increase the possibility of obtaining multiple trajectories to the same destination, the aircraft's airspeed is reduced to 90 mph such that it is smaller than the speed of the winds in which it travels. This forces the aircraft to fly only in the direction that the winds prescribe. As such, the result for an aircraft departing from Johannesburg and traveling to a destination in the Indian ocean, which is illustrated in Figures 4 and 5, indicates that overlapping extremal front instances lead to the existence of sub-optimal trajectories. Thus, for this example, two initial headings lead the aircraft to the same destination yielding two possible travel times: 5 hours and 54 minutes for the optimal trajectory, and 17 hours and 45 minutes for the suboptimal one. Figure 5 is a three-dimensional plot of the extremal front time evolution, which shows the existence of multiple solutions caused by the intersection

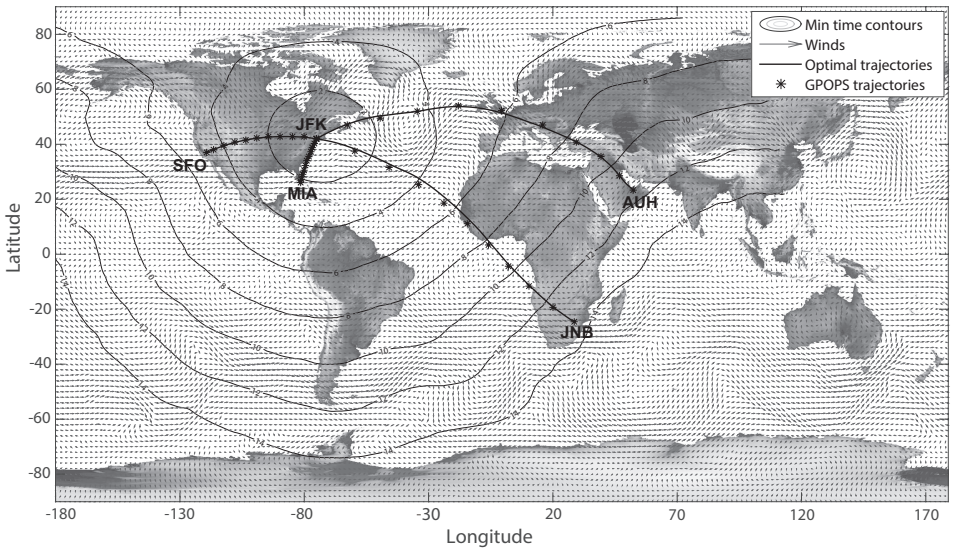

Figure 2. Minimum-time function contours on the mapped sphere with optimal trajectories.

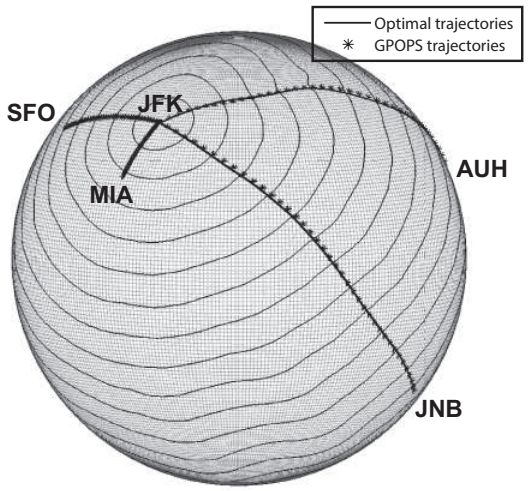

Figure 3. Extremal front expansion on the sphere with optimal trajectories for four different destinations.

of the extremal front expansion, represented by light grey circles, with the target destination, shown as the black vertical line. This is representative for the discussion on the possibility of obtaining suboptimal trajectories given in Section III. The systematic method proposed herein is able to handle such cases and to compute unique global minimum-time solutions.

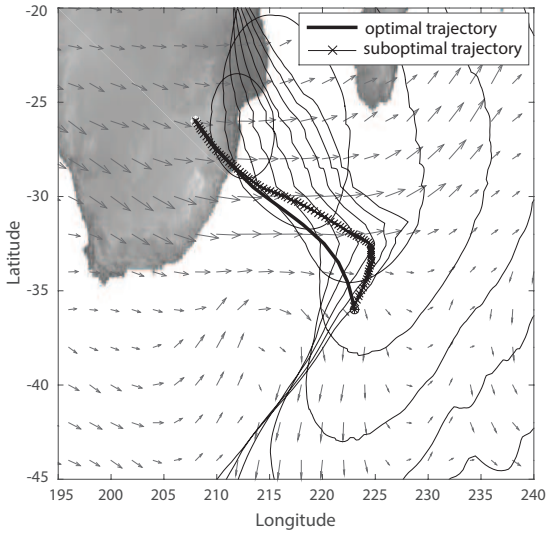

Figure 4. Trajectories for a destination that admits multiple candidate optimal solutions. Specifically, two different values of ψ_0 determine two candidate optimal trajectories which reach the same point at different times.

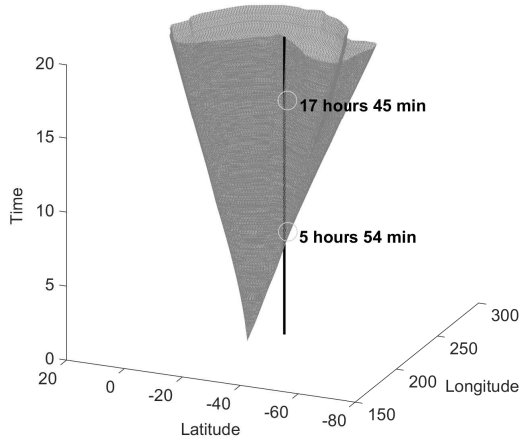

Figure 5. Time evolution of the extremal front presented in Figure 4, with two possible solutions for the time function.

Further, to test the accuracy of these solutions, the optimization software GPOPS[21] was used to generate minimum-time trajectories of the Zermelo navigation problem on the sphere. This software is able to compute accurate solutions to the optimal control problem. However, without sufficient effort on tuning the initial

guesses for each different scenario, GPOPS cannot easily compute globally the minimum-time function. In other words, GPOPS is more suitable for the computation of optimal trajectories for a given set of boundary conditions than the characterization of the optimal synthesis, which typically requires a dynamic programming approach. As such, only trajectories were compared for the first presented scenario and the results were plotted against the minimum-time trajectories obtained with the proposed algorithm in Figures 2 and 3. The mean squared error (MSE) between these trajectories is averaged to `0.005489`%. This close agreement demonstrates the ability of the proposed algorithm to produce accurate optimal solutions.

Lastly, to illustrate the efficiency of the presented algorithm, consider the times required for the computation of the paths presented in Figures 2 and 3. The minimum-time function was computed in 251 seconds. While this might appear as a long time, one must consider the fact that this function provides minimum time values for the entire reachable set within the specified time limit of $T_{\max} = 14$ hours. Having the minimum-time function computed, the 4 specific paths were obtained in times comparable to GPOPS. These times are presented in Table 1.

Table 1. Comparison for Algorithm Computational Times and Errors

Path	Presented Algorithm Times (sec)	GPOPS Times (sec)	MSE (km)	MSE (%)
JFK-SFO	0.240	0.632	0.1605	0.0043
JFK-MIA	0.244	0.259	0.0719	0.0039
JFK-JNB	0.239	0.716	0.5871	0.0055
JFK-AUH	0.470	0.667	1.0510	0.0083

V. Conclusions

This Note presents a new method for computing globally the minimum-time function of the Zermelo navigation problem on a sphere, in the presence of a spatially varying wind field. One of the distinctive features of the method is the ability to handle discontinuities in the minimum-time function and singularities of the optimal synthesis. The proposed algorithm exploits the structure of the solution to the optimal control problem and uses a systematic method to "filter out" sub-optimal solutions. This allows for the computation of the minimum-time function using a discretized mesh of points that are assigned minimum time values specific to their respective positions. In general, the use of a mesh may lead to slow computational times if the mesh size is very large. However, the mesh size depends on the desired application, which in this case does not require an extremely fine resolution. Furthermore, with the minimum-time function already generated, minimum-time trajectories can be obtained with minimal computational effort.

References

[1]Jardin, M. R. and Bryson, A. E., "Methods for Computing Minimum-Time Paths in Strong Winds," *Journal of Guidance, Control, and Dynamics*, Vol. 35, No. 1, 2012, pp. 165–171, doi:10.2514/1.53614.

[2]Dancila, B. D. and Botez, R. M., "Construction of an Aircraft's VNAV Flight Envelope for in-FMS Flight Trajectory Computation and Optimization: Application on the Airbus A310 Performance Data," *AIAA Aviation*, 2014, doi:10.2514/6.2014-2291.

[3]Ng, H. K., Sridhar, B., and Grabbe, S., "Optimizing Aircraft Trajectories with Multiple Cruise Altitudes in the Presence of Winds," *Journal of Aerospace Information Systems*, Vol. 11, No. 1, 2014, pp. 35–47, doi:10.2514/1.I010084.

[4]Murrieta-Mendoza, A. and Botez, R. M., "Lateral Navigation Optimization Considering Winds and Temperatures for Fixed Altitude Cruise using the Dijkstras Algorithm," *The ASME 2014 International Mechanical Engineering Congress & Exposition*, 2014, doi:10.1115/IMECE2014-37570.

[5]Felix Patron, R. S., Kessaci, A., and Botez, R. M., "Horizontal Flight Trajectories Optimization for Commercial Aircraft through a Flight Management System," *The Aeronautical Journal*, Vol. 118, No. 1210, 2014.

[6]Kirk, D., *Optimal Control Theory*, Dover Publications, Inc., 2004, pp. 240–259, 329-408.

[7]Betts, J., *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*, Society for Industrial and Applied Mathematics, 2nd ed., 2010, pp. 91–118.

[8]von Stryk, O. and Roland, B., "Direct and indirect methods for trajectory optimization," *Annals of operations research*, Vol. 37, No. 1, 1992, pp. 357–373, doi:10.1007/BF02071065.

[9]Lavalle, S., *Planning Algorithms*, Cambridge University Press, 2006, pp. 304–359, 590-650.

[10]Eichhorn, M., "Optimal Routing Strategies for Autonomous Underwater Vehicles in Time-Varying Environment," *Robotics and Autonomous Systems*, 2013, doi:10.1016/j.robot.2013.08.010.

[11]Lolla, T., Lermusiaux, P., Ueckermann, M., and Haley, P., "Time-Optimal Path Planning in Dynamic Flows using Level Set Equations: Theory and Schemes," *Ocean Dynamics*, Vol. 64, No. 10, 2014, pp. 1373–1397, doi:10.1007/s10236-014-0757-y.

[12]Soulignac, M., "Feasible and Optimal Path Planning in Strong Current Fields," *IEEE Transactions on Robotics*, Vol. 27, No. 1, 2011, pp. 89–98, doi:10.1109/TRO.2010.2085790.

[13]Chakrabarty, A. and Langelaan, J. W., "Energy-Based Long-Range Path Planning for Soaring-Capable Unmanned Aerial Vehicles," *Journal of Guidance, Control, and Dynamics*, Vol. 34, No. 4, 2013, pp. 1002–1015, doi: 10.2514/1.52738.

[14]Elston, J. and Frew, E., "Unmanned Aircraft Guidance for Penetration of Pretornadic Storms," *Journal of Guidance, Control, and Dynamics*, Vol. 33, No. 1, 2010, pp. 99–107, doi:10.2514/1.45195.

[15]Bijlsma, S., "Optimal Aircraft Routing in General Wind Fields," *Journal of Guidance, Control, and Dynamics*, Vol. 32, No. 3, 2009, pp. 1025–1029, doi:10.2514/1.42425.

[16]Zermelo, E., "Über das Navigationsproblem bei Ruhender oder Veranderlicher Windverteilung," *Zeitschrift für Angewandte Mathematik und Mechanik*, Vol. 11, No. 2, 1931, pp. 114–124.

[17]Bakolas, E. and Tsiotras, P., "Optimal Pursuit of Moving Targets using Dynamic Voronoi Diagrams," *Proceedings of IEEE International Conference on Decision and Control*, IEEE, Atlanta, GA, 2010, pp. 7431–7436, doi:10.1109/CDC.2010.5717963.

[18]Bakolas, E. and Tsiotras, P., "Optimal Partitioning for Spatiotemporal Coverage in a Drift Field," *Automatica*, Vol. 49, No. 7, 2013, pp. 2064–2073, doi:10.1016/j.automatica.2013.04.013.

[19]Rhoads, B., Mezic, I., and Poje, A., "Minimum Time Heading Control of Underpowered Vehicles in Time-Varying Ocean Currents," *Ocean Engineering*, Vol. 66, 2013, pp. 12–31, doi:10.1016/j.oceaneng.2013.03.012.

[20]Rutledge, G., Alpert, J., and Ebuisaki, W., "NOMADS: A Climate and Weather Model Archive at the National Oceanic and Atmospheric Administration," *Bull. Amer. Meteor. Soc.*, Vol. 87, 2006, pp. 327–341, doi:10.1175/BAMS-87-3-327.

[21]Rao, A., Benson, D., Darby, C., Mahon, B., Francolin, C., Patterson, M., Sanders, I., and Huntington, G., "Algorithm 902: GPOPS, A MATLAB Software for Solving Multiple-Phase Optimal Control Problems Using the Gauss Pseudospectral Method," *ACM Trans. Math. Softw.*, Vol. 37, No. 2, 2010, pp. 22:1 – 22:39, doi:10.1145/1731022.1731032.