



HAL
open science

Wind-Optimal Path Planning: Application to Aircraft Trajectories

Brunilde Girardet, Laurent Lapasset, Daniel Delahaye, Christophe Rabut

► **To cite this version:**

Brunilde Girardet, Laurent Lapasset, Daniel Delahaye, Christophe Rabut. Wind-Optimal Path Planning: Application to Aircraft Trajectories. ICARCV'14, 13th International Conference on Control, Automation, Robotics and Vision, Dec 2014, Singapore, Singapore. hal-01129968

HAL Id: hal-01129968

<https://enac.hal.science/hal-01129968>

Submitted on 13 Mar 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Wind-Optimal Path Planning: Application to Aircraft Trajectories

Brunilde Girardet
and Laurent Lapasset

Capgemini Technology Services F-31036 Toulouse
Email: girardet@recherche.enac.fr
llapasse@capgemini.com

Daniel Delahaye
ENAC-MAIAA

F-31055 Toulouse
Email: delahaye@recherche.enac.fr

Christophe Rabut
Université de Toulouse
(INSA,IMT,MAIAA)

F-31077 Toulouse
Email: christophe.rabut@insa-toulouse.fr

Abstract—In this paper, an algorithm to plan a continuous wind-optimal path is proposed, and simulations are made for aircraft trajectories. We consider a mobile which can move in a two dimensional space. The mobile is controlled only by the heading direction, the speed of the mobile is assumed to be constant. The objective is to plan the optimal path avoiding obstacles and taking into account wind currents. The algorithm is based on Ordered Upwind Method which gives an optimality proof for the solution. The algorithm is then extended to spherical coordinates in order to be able to handle long paths.

Index Terms—Obstacle avoidance, path planning, wind, aircraft trajectory.

I. INTRODUCTION

Planning optimal trajectories is a rich and dynamic research domain with many application areas like robotics, space or aviation. Depending on the problems needs, the issues are different in nature and so are the techniques used to solve them. Here, we are interested in finding the global optimal path in presence of currents in a two dimensional space. Several methods, as Dijkstra algorithm [1] or A* algorithm [2], discretize the domain and work on the generated network to find the optimal path. These algorithms are very efficient but the computed solution is restricted to the network. Thus, the quality of the solution depends on it. With the proposed algorithm, the solution is not constrained to follow a network and is computed on the continuous space. The algorithm is based on front propagation methods as Level Set methods, Fast Marching methods and Ordered Upwind methods. These different algorithms are developed by Sethian in [3].

In [4], Pêtrès adapts the Fast Marching Method to path planning for Autonomous Underwater Vehicles taking into account underwater currents. However, his algorithm cannot be applied to vehicles featuring behaviours more complex than a linear reaction to currents. In [5], Alton uses the Ordered Upwind algorithm with the Semi-Lagrangian method to generate optimal trajectories. The drawback of this approach is the need of a local minimization at each mesh point used to discretize the space, which increases the computation time. Here, we base our algorithm on the Ordered Upwind algorithm with the Eulerian method.

The main contribution of this paper is, first, to model the optimal path planning in order to use the Eulerian discretization in the Ordered Upwind Method. Usually, the drawback

of this discretization is the need to find the roots of a non-linear equation. In our case, we will show that the resolution is easy since the equation is reduced to a quadratic equation. The second contribution of the paper is to adapt the algorithm to the spherical space to be able to handle long paths around the Earth.

The paper is organized as follows. Section II presents the modelling of the problem. In Section III, we present the algorithm to plan wind-optimal path. Simulations and experiments are performed to validate the proposed approach in Section IV. Finally, conclusions of the work are given in Section V.

II. MODELLING PATH PLANNING PROBLEM

In this section, we present the mathematical formulation of the problem. The goal is to drive the mobile to reach a goal position (x_f, y_f) from a start position (x_0, y_0) in a minimum time while avoiding obstacles. The speed of the mobile is constant. The research area is a two dimensional space with the presence of currents. In this paper, we talk about wind currents since we are interested in aircraft trajectories but it can be also valid for other types of currents such as underwater currents.

A. Equation of Motion

Considering wind currents, the motion of the mobile is defined as follows:

$$\begin{cases} \dot{x}(t) = V_a \sin(\theta(t)) + W_x(x, y) \\ \dot{y}(t) = V_a \cos(\theta(t)) + W_y(x, y) \end{cases}$$

where (x, y) is the mobile position, θ is the heading angle relative to North direction, V_a is the constant speed of the mobile, $W_x(x, y)$ is the east component of the wind, and $W_y(x, y)$ is the north component of the wind.

B. Optimization Problem

The control parameter of the optimal path is the heading angle θ . The optimization problem is then to find the best control θ over time to minimize the total travel time. The

optimization problem can be written as:

$$\left\{ \begin{array}{l} \min_{\theta} \quad (t_f - t_0) \\ \text{s.t.} \quad \dot{x}(t) = V_a \sin(\theta(t)) + W_x \\ \quad \quad \dot{y}(t) = V_a \cos(\theta(t)) + W_y \\ \quad \quad (x(t_0), y(t_0)) = (x_0, y_0) \\ \quad \quad (x(t_f), y(t_f)) = (x_f, y_f) \end{array} \right.$$

It is convenient to let the control variable be $\mathbf{a}(t) = (\sin(\theta(t)), \cos(\theta(t)))$, a unit vector in the direction of motion of the mobile. In this case, the optimization problem becomes:

$$\left\{ \begin{array}{l} \min_{\mathbf{a} \in A} \quad (t_f - t_0) \\ \text{s.t.} \quad \dot{\mathbf{X}} = f(\mathbf{X}(t), \mathbf{a}(t)) \\ \quad \quad \mathbf{X}(t_0) = \mathbf{X}_0 \text{ and } \mathbf{X}(t_f) = \mathbf{X}_f, \end{array} \right. \quad (1)$$

where \mathbf{X} is the mobile position, and $f(\mathbf{X}(t), \mathbf{a}(t))$ is the real speed of the mobile taking into account wind currents at time t .

III. PATH PLANNING BASED ON ORDERED UPWIND METHOD

The problem (1) is a standard formulation of optimal control problems. One resolution method is to write this problem as an Hamilton-Jacobi equation and to solve it. The form of the Hamilton-Jacobi equation is based on Bellman's optimality principle in discrete case [6]. The details of the calculations can be found in [7].

A. The Hamilton-Jacobi Equation

In our problem, the Hamilton-Jacobi equation corresponding to the optimal control problem (1) is given by:

$$\max_{\mathbf{a} \in A} \{-\nabla u(\mathbf{X}) \cdot f(\mathbf{X}, \mathbf{a})\} = 1 \quad (2)$$

The resolution of this Partial Differential Equation (PDE) on the whole space gives a necessary and sufficient condition for optimality. The principle of the resolution is to decompose recursively the problem into linked sub-problems as in Dynamic Programming for a discrete problem. The resolution can be seen as a front expansion problem where the wavefront represents the minimum time to reach the arrival point. The wavefront is computed gradually from the previous front positions. It is based on the Huygen's principle which states every point reached by a wavefront becomes a source of a spherical wavefront; the sum of these secondary waves determines the form of the wavefront at any subsequent time. The evolution of the wavefront is described by (3):

$$\|\nabla u(\mathbf{X})\| F(\mathbf{X}, \frac{\nabla u(\mathbf{X})}{\|\nabla u(\mathbf{X})\|}) = 1 \quad (3)$$

where: $F(\mathbf{X}, \mathbf{n})$ is the front speed in the direction \mathbf{n} , with \mathbf{n} the outward unit vector normal to the front at point \mathbf{X} , and $u(\mathbf{X})$ represents the minimum time to reach the end point starting from the point \mathbf{X} .

With (3) and by rewriting (2) as (4), we obtain a relation between the speed of the wavefront, $F(\mathbf{X}, \mathbf{n})$, and the speed of the mobile, $f(\mathbf{X}, \mathbf{a})$.

$$\|\nabla u(\mathbf{X})\| \max_{\mathbf{a} \in A} \left\{ -\frac{\nabla u(\mathbf{X})}{\|\nabla u(\mathbf{X})\|} \cdot f(\mathbf{X}, \mathbf{a}) \right\} = 1 \quad (4)$$

The optimal path problem can then be written as a front expansion problem where the speed of the wavefront, $F(\mathbf{X}, \mathbf{n})$, depends on the speed of the mobile, $f(\mathbf{X}, \mathbf{a})$.

$$F(\mathbf{X}, \mathbf{n}) = \max_{\mathbf{a} \in A} \{-\mathbf{n} \cdot f(\mathbf{X}, \mathbf{a})\} \quad (5)$$

To design the optimal path between the departure point and the arrival point, we have to follow the characteristics of the Hamilton-Jacobi PDE from the departure point until we reach the arrival point since we know that these characteristics are the optimal trajectories of the corresponding control problem.

B. Ordered Upwind Algorithm

The Ordered Upwind algorithm was developed by Sethian and Valdirimsky to approximate the solution of the Hamilton-Jacobi equations in [8]. In [9], Sethian proved that the algorithm converges to the viscosity solution of the Hamilton-Jacobi equations, a weak solution of the PDE (3).

The principle of the Ordered Upwind algorithm is to avoid iterations thanks to a careful use of the information about the characteristic directions of the PDE. It uses the fact that the value function, u , is strictly increasing along the characteristics and then verifies the causality property. The value function $u(\mathbf{X})$ is then constructed gradually using the previous smaller values $u(\mathbf{X})$ along the characteristic. This principle makes the algorithm highly efficient. In general, characteristic directions of the PDE are not known in advance. The strength of the Ordered Upwind method is the ability to compute information about the characteristics as the solution is constructed. In our problem, the characteristics of (2) represent the optimal trajectories.

To compute the value function, u , we consider a 2D non-regular triangular mesh. Let $(\mathbf{X}, \mathbf{X}_j, \mathbf{X}_k)$ be a simplex, the value of $u(\mathbf{X})$ is computed from $u(\mathbf{X}_j)$ and $u(\mathbf{X}_k)$ if the characteristic direction at mesh point \mathbf{X} lies inside the simplex $(\mathbf{X}, \mathbf{X}_j, \mathbf{X}_k)$. Let h be the maximum distance between two adjacent mesh points (i.e. if the mesh points \mathbf{X}_j and \mathbf{X}_k are adjacent, then $\|\mathbf{X}_j - \mathbf{X}_k\| \leq h$). All mesh points belong to one of the following classes (see Fig.1):

- *Accepted* is the set of mesh points where the function u has been computed. Such points are considered frozen.
- *Considered* is the set of mesh points where an estimate v of u has been computed but not frozen.
- *Far* is the set of all other mesh points where an estimate v of u has not been computed yet.

Two other sets are also defined:

- *AcceptedFront* is the subset of *Accepted* mesh points that are adjacent to some not-yet-accepted (i.e. *Considered*) mesh points.

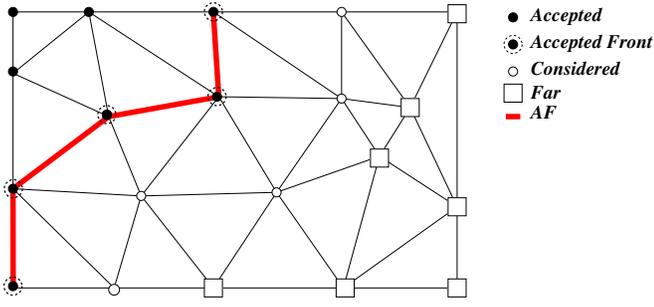


Fig. 1. All the mesh points are assigned to three different sets: Accepted, Considered, and Far. AcceptedFront is a subset of the Accepted set. The AF set describes the front.

- *AF* is the set of line segments $[X_j, X_k]$, such that X_j and X_k are adjacent mesh points on the *AcceptedFront*, and X_j and X_k are adjacent to a *Considered* mesh point X .

For each *Considered* mesh point X , we define a new set called *NearFront* (*NF*). It is the subset of *AF* segments that are close to the *Considered* mesh point X :

$$NF(\mathbf{x}) = \left\{ (\mathbf{x}_j, \mathbf{x}_k) \in AF \mid \exists \tilde{\mathbf{x}} \text{ on } (\mathbf{x}_j, \mathbf{x}_k) \text{ s.t. } \|\tilde{\mathbf{x}} - \mathbf{x}\| \leq h \frac{f_2}{f_1} \right\},$$

where f_1 and f_2 are respectively the minimum and the maximum speed of the mobile taking wind into account.

Ordered Upwind Algorithm

- 1) Start with all the mesh points in Far ($u = +\infty$);
- 2) Move the initial point X_0 to Accepted ($u(X_0) = 0$);
- 3) Move all the mesh points X adjacent to the initial point into Considered and evaluate the trial value $v(X)$ as:

$$v(X) := \min_{\mathbf{x}_i \in NF(\mathbf{x})} v_{\mathbf{x}_i}(X)$$

- 4) Find the mesh point \tilde{X} with the smallest value of v among all Considered;
- 5) Move \tilde{X} to Accepted ($u(\tilde{X}) = v(\tilde{X})$), and update the AcceptedFront;
- 6) Move the Far mesh points X adjacent to \tilde{X} in Considered, and compute their trial values by:

$$v(X) := \min_{\mathbf{x}_j, \mathbf{x}_k \in NF(\mathbf{x})} v_{\mathbf{x}_j, \mathbf{x}_k}(X) \quad (6)$$

- 7) Recompute the values for all the other Considered X such that $\tilde{X}X_i \in NF(X)$ by:

$$v(X) = \min \left\{ v(X), \min_{\tilde{X}X_i \in NF(X)} v_{\tilde{X}, X_i}(X) \right\}$$

- 8) If Considered is not empty, then go to step 4.

Two equivalent methods are presented in [9] to compute the trial value $v(X)$ from a simplex (X, X_j, X_k) . The Semi-Lagrangian method requires performing a local minimization at each mesh point, whereas the finite-differences upwind update formula (Eulerian discretization) requires finding the roots of a non-linear equation. In our case, the speed of the wavefront F has a closed form, and the value $v_{\mathbf{x}_j, \mathbf{x}_k}(X)$ in (6) can be computed using a finite-differences upwind formula of

the Hamilton-Jacobi equation (4). The discretized equation is then a quadratic equation.

By modelling the mobile speed as $f(X, \mathbf{a}) = V_a \mathbf{a} + W$ in (5), the speed of the wavefront is equal to:

$$F(X, \mathbf{n}) = V_a - \langle \mathbf{n}, W \rangle \quad (7)$$

More details about the calculus of the speed of the wavefront are given in [10]. From this definition of the wavefront speed, (3) is discretized using an upwind finite-difference discretization on the simplex (X, X_j, X_k) to compute the value $v_{X_j, X_k}(X)$. Equation (3) becomes:

$$\|P^{-1}w(X)\|^2 V_a^2 = (1 + \langle P^{-1}w(X), W \rangle)^2 \quad (8)$$

The vector $P^{-1}w(X)$ is the discretization of $\nabla u(X)$ from the directional derivatives of u in the directions defined by the edges of the simplex $(\mathbf{x}, \mathbf{x}_j, \mathbf{x}_k)$ (see [10] for the details). We note that (8) is a quadratic equation of the form:

$$Av_{\mathbf{x}_j, \mathbf{x}_k}^2(X) + Bv_{\mathbf{x}_j, \mathbf{x}_k}(X) + C = 0 \quad (9)$$

with :

$$\begin{aligned} A &= V_a^2 \langle P^{-1}\alpha, P^{-1}\alpha \rangle - \langle P^{-1}\alpha, W \rangle^2 \\ B &= 2V_a^2 \langle P^{-1}\alpha, P^{-1}\beta \rangle - 2\langle P^{-1}\alpha, W \rangle (\langle P^{-1}\beta, W \rangle + 1) \\ C &= V_a^2 \langle P^{-1}\beta, P^{-1}\beta \rangle - [\langle P^{-1}\beta, W \rangle + 1]^2 \end{aligned}$$

The value $v_{\mathbf{x}_j, \mathbf{x}_k}$ is then computed by solving (9). To ensure that $v_{\mathbf{x}_j, \mathbf{x}_k}$ is a good approximation of the value function, u , at the point X , the characteristic direction for the mesh point X needs to lie inside the simplex (X, X_j, X_k) . It is shown in [10] how to check this property.

The optimal trajectory is built by moving from the initial point to the destination point along the characteristic direction determined by:

$$\frac{dX}{dt} = -V_a \frac{\nabla u(X)}{\|\nabla u(X)\|} + W(X)$$

The computational complexity of this algorithm is $O(\Upsilon N \log N)$, where N is the number of mesh points and Υ the anisotropy ratio: the ratio of the fastest to slowest propagation speed for each point (see [9] for more details).

C. Obstacles Avoidance

We propose to modify the speed of propagation. We slow down the wavefront in parts of the environment that have to be avoided. Thus, the value function u is increased, penalizing the passage through these areas. We create a map of values ξ as a function of the obstacles. The values ξ are between 0 and a maximum value ξ_{max} . The scaled values ξ are then used to slow down the wavefront speed as follows: $(1 - \xi)F(X, \mathbf{n})$. The maximum value ξ_{max} needs to be less than 1 to keep the wavefront speed strictly positive.

D. Adaptation to Spherical Coordinates

The algorithm presented above works on a Cartesian grid; it therefore relies on the assumption that geodesics are straight lines. Recall that geodesics are shortest paths in terms of distance between two points in a given space. In order to work with long paths for which the wind-optimal trajectories offer the most benefits, we have to take into account the sphericity of Earth where geodesics are then segments of great circles. We approximate the form of Earth by a sphere, and we adapt the algorithm to spherical coordinates. The idea is to project the spherical coordinates on a plane with Cartesian coordinates via a map projection.

The map projection, ψ , used in this work is a transformation of latitudes, ϕ , and longitudes, θ , of points on the surface of a sphere into points on a plane:

$$\begin{aligned} \psi : [-180; 180] \times [-90; 90] &\rightarrow \mathbb{R} \times \mathbb{R} \\ (\theta, \phi) &\rightarrow (x, y) \end{aligned}$$

However, projecting points on a sphere into a plane is not sufficient to use the Ordered Upwind Method since all map projections cause distortion in the shapes, areas, or distances. Different map projections exist in order to preserve some of these properties. Earth being represented by a sphere, any flat representation generates distortions such that angles and areas cannot both be maintained simultaneously, and distances can never be preserved. This last point explains why the Ordered Upwind Method needs adaptations to be used on a sphere.

The principle of the Ordered Upwind Method is to compute the minimum time to reach a point of the space from the propagation of the front by solving (3). To keep the equation on the map, the projection needs to preserve the relation between vectors. The projection of a vector \vec{v} on a sphere in spherical coordinates into a vector \vec{w} on a plane in Cartesian coordinates by an application ψ can be done as follows.

Let γ be a path on the sphere and x be a point on this path: $\gamma :]-\varepsilon, \varepsilon[\rightarrow S \subset \mathbb{R}^3$ and $\gamma(0) = x$. The speed vector in spherical coordinates \vec{v} at the point x is defined by $\gamma'(0)$. By the application ψ , the speed vector \vec{w} on the map projection is equal to:

$$\vec{w} = (\psi \circ \gamma)'(0) = \underbrace{\psi'(\gamma(0))}_{J_\psi(x)} \cdot \gamma'(0) = J_\psi(x) \cdot \vec{v} \quad (10)$$

Thus, the inner product of two projected vectors w_1 and w_2 is equal to:

$$\langle J_\psi(\theta, \phi) v_1, J_\psi(\theta, \phi) v_2 \rangle = \langle J_\psi^T(\theta, \phi) J_\psi(\theta, \phi) v_1, v_2 \rangle$$

For the Ordered Upwind algorithm, the inner product has to be preserved up to a positive multiplicative constant. To meet this requirement, the projection has to be conformal meaning that the matrix $J_\psi(\theta, \phi)$ is a positive multiple of a rotation matrix. Equivalently, the matrix $J_\psi^T(\theta, \phi) J_\psi(\theta, \phi)$ is a positive multiple of the identity. Under this assumption, the distortion induced by the projection is isotropic and can be corrected by scaling. However, note that the scaling factor depends on the point where the correction has to be applied.

In the presented algorithm, we first project the points into the plane, and then adapt the Ordered Upwind Method by correcting the distance distortions at the neighbourhood of each point on the plane. The optimal trajectory will be computed on the plane and then projected back on the sphere by the inverse projection.

1) *Lambert Conformal Conic Projection*: The map projection, ψ , used here is a Lambert conformal conic projection. It is a conformal map. The general principle of the projection construction is as follows (see [11] for more details on different map projections). The idea is to seat a cone over the sphere of Earth and projects conformally onto the cone. The cone is unrolled to give a plane, and the parallel touching the sphere is assigned unitary scale and is called the reference parallel or standard parallel. The transformation is based on two key points: first, as the projection is conform, the angles are preserved; secondly, the distance between points on the standard parallel is preserved on the plane. The scale factor m is then equal to 1 for the latitude of the reference parallel. The equations to project a point on the surface of the sphere into a point on a plane by the Lambert conformal conic projection are presented in [11].

2) *Scale Factor*: The scale factor m gives the distortion of the projection in the area of interest. It is the ratio of the scale at a given point to the true scale. Scale distortions exist at locations where the scale factor is smaller or larger than 1. Distortions increase as the distance from the standard point (tangent plane) increases. The scale factor is not constant, changing with direction and location. It depends on the properties of the projection, and is defined at a given point by:

$$m = \frac{\text{Distance on the map projection}}{\text{Distance on the sphere}}$$

The scale factor is a key point in the adaptation of the Ordered Upwind Method since we will use it to compute the value function $u(X)$. As we saw in Section III-B, the time at each point is determined using two neighbouring points. We use the distance between points and the speed of the propagation to compute a time. Thus, if the distance between projected points is not equal to the real distance on the sphere, the computed time in the flat map will not be correct. The distance between projected points has to be corrected using the scale factor. This correction is only valid as long as points are not too far from each other.

As we consider two neighbouring points X_j and X_k to compute the time at one point X , the distance between these three points is not large. Thus, we assume that on the triangle described by X , X_j and X_k , a straight line approximates the geodesic. With this reasonable hypothesis, the distance between X , X_j and X_k can be corrected using the computed scale factor. And based on this critical distance correction on the triangle X , X_j and X_k , we can use the same steps of the Ordered Upwind algorithm to compute the minimum time at the point X .

As in (7), the speed of propagation F depends on the aircraft speed, the wind speed and the normal vector of the

propagation front, all of these parameters have to be in the same coordinates systems. The normal of the propagation front is computed within the algorithm; this vector is already in Cartesian coordinates and takes distortions into account. Thus, instead of correcting the projected distance, we work with the projected speed of propagation F on the map using the scale factor. As the normal vector of the propagation front is already computed on the map, we only have to modify the aircraft speed and to project the wind speed on the map. The aircraft speed is multiplied by the scale factor to offset the distortions, and the wind speed is projected on the map. Finally, the propagation speed on the map is equal to:

$$F(X, \mathbf{n}) = m V_a - \langle \mathbf{n}, W_P \rangle$$

where W_P is the wind vector projected on the map and m is the scale factor at the point X .

3) *Algorithm:* Let us summarize the different steps of the adaptation of the algorithm.

- 1) Discretization of the workspace and transformation of the grid points in longitude/latitude coordinates into the plane by a Lambert conformal conic projection;
- 2) Transformation of winds data into the plane;
- 3) Application of the Ordered Upwind algorithm on the projected map using scale factor to correct distortions;
- 4) Computation of the trajectory. This step is detailed below.
- 5) Inverse transformation of the trajectory points from Cartesian coordinates into longitude/latitude coordinates.

The optimal trajectory is built by moving from the initial point to the destination point along the characteristic direction by solving the following equation:

$$\frac{dX}{dt} = -m V_a \frac{\nabla u(X)}{\|\nabla u(X)\|} + W_P(X)$$

To compute the gradient $\nabla u(X)$, we compute the gradient $\nabla u(\theta, \phi)$ using finite differences on the regular grid on the sphere. The gradient $\nabla u(\theta, \phi)$ can be written as a function of the gradient $\nabla u(x, y)$ and of the map projection ψ . Equation (11) gives the relation between the gradient in spherical coordinates and the gradient in Cartesian coordinates (with J_ψ the Jacobian matrix of ψ):

$$\begin{aligned} \langle \nabla u(x, y), \vec{w} \rangle &= \langle \nabla u(x, y), J_\psi(\theta, \phi) \vec{v} \rangle \\ &= \underbrace{\langle J_\psi^T(\theta, \phi) \nabla u(x, y), \vec{v} \rangle}_{\nabla u(\theta, \phi)} \\ &= \langle \nabla u(\theta, \phi), \vec{v} \rangle \end{aligned} \quad (11)$$

The gradient $\nabla u(X)$ used to construct the optimal trajectory is then obtained in (12) from the gradient $\nabla u(\theta, \phi)$:

$$\nabla u(X) = (J_\psi^T(\theta, \phi))^{-1} \nabla u(\theta, \phi) \quad (12)$$

IV. SIMULATIONS AND RESULTS ANALYSIS

A. Trajectory Simulation with Obstacles

Our first test problem aims at emphasizing the impact of wind on trajectories while avoiding obstacles. From the point

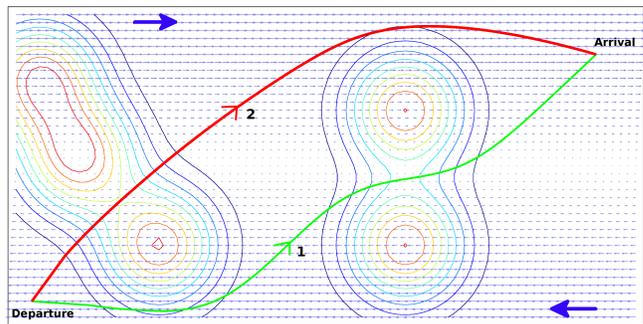


Fig. 2. Optimal trajectory with obstacles: 1. Without wind (green), 2. With wind (red)

of view of airlines' operations, these obstacles could model some adverse weather conditions such as storms or turbulences. In Fig. 2, the optimal trajectories without wind (1.) and with wind (2.) have been computed using the Ordered Upwind algorithm with the Eulerian discretization. The obstacles are described by coloured iso-contours, red represents impassable obstacles and the colour goes toward the blue as the importance of the obstacles decreases. In the north part, wind is in the east direction and increases northwards. In the south part, wind is in the west direction and increases southwards.

The trajectory taking into account the wind takes advantage of favourable winds. The two trajectories do not behave in the same way to get around the obstacles. This is attractive to take into account the wind to plan the aircraft trajectory. The wind-optimal trajectory can be far from the optimal trajectory computed without wind.

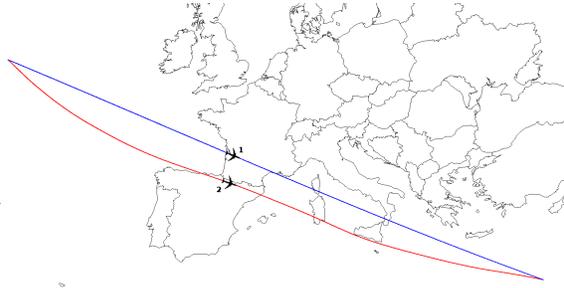
B. Trajectory Simulation in Spherical Coordinates

1) *Traffic Data for Simulations:* The data used comes from a simulation of one day of traffic in Europe. The traffic sample contains 28,278 flights with different types of aircraft flying at different speeds. We only select a part of the flights: the cruise phase. We assume that during this phase, the altitude remains constant. In the simulation, we first deal with all the flights. Then, we concentrate on the longest trajectories (>800 km) to study the benefit of wind-optimal routes. An analysis of the traffic data can be found in Table I.

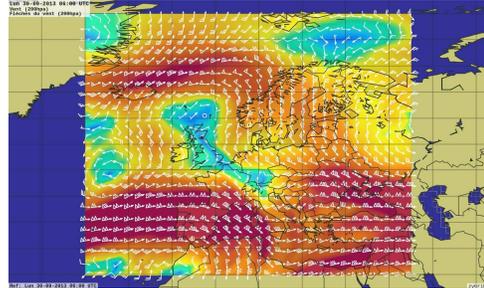
2) *Wind Data for Simulations:* Three days of wind forecast over Europe are considered: September 30th, October 2nd, and October 4th 2013. An analysis of the wind data for the three days is displayed in Table II. Figure 3b shows the wind forecast on September 30th 2013. Wind vectors (arrows) represent the direction and speed of the wind. Colours also show the

TABLE I
AVERAGE TRAFFIC DATA

	Aircraft number	Speed (kt)	Distance (km)	Flight time (min)
All traffic	28278	403	850	64
> 800km	10911	460	1700	120



(a) Trajectories: 1. Direct Route (blue), 2. Optimal Route (red)



(b) Wind forecast on September 30th 2013

Fig. 3. Simulation results for one flight: JWD020.

different speeds, red colour symbolizes the maximum wind speed and blue colour denotes the minimum wind speed. The optimal route of each flight is computed with the three different wind conditions, and is compared with the direct route.

Table III presents the benefit in time if aircraft follow optimal routes rather than direct routes. Q1, Q2 and Q3 denote respectively the first, second and third quartile of the time benefit. Depending on the day, the average time benefit is different. For half of the flights, the direct route is the best solution; the time benefit is thereby null and the algorithm finds the direct route. In the three cases, it is interesting to note that optimizing the longest trajectories is most beneficial. About three-quarters of these flights get a time benefit.

The direct and optimal routes computed for one particular flight, JWD020, have been depicted in Fig. 3a in which the blue trajectory (aircraft 1) is the direct route and the red trajectory (aircraft 2) is the wind-optimal one. Table IV illustrates the difference between the optimal and the direct routes for the flight JWD020. The optimal route is deviated from the direct route to take advantage of tail winds. The travelled distance is then longer but the flight time is shorter. The obtained benefit is around 6 minutes.

V. CONCLUSION

We presented in this paper a new model for the optimal path planning problem of a single trajectory. We proposed an analytical solution of the discretized equation for the Ordered Upwind algorithm. Thus, this model has the advantage to avoid an iterative algorithm for the resolution of the equation. The results show the benefits of taking into account currents while avoiding obstacles to plan optimal path especially for long paths or for strong currents.

TABLE II
WIND DATA

	Average (kt)	Min (kt)	Max (kt)
2013/09/30	40	0.25	130
2013/10/02	48	0.58	114
2013/10/04	42	0.6	115

REFERENCES

- [1] E. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik I*, pp. 269–271, 1959.
- [2] P. Hart, N. Nilsson, and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, July 1968.
- [3] J. Sethian, *Level Set Methods and Fast Marching Methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science*. Cambridge University Press, 1999, no. 3.
- [4] C. Pêtrès, Y. Pailhas, P. Patron, Y. Petillot, J. Evans, and D. Lane, "Path Planning for Autonomous Underwater Vehicles," *IEEE Transactions on Robotics*, vol. 23, no. 2, pp. 331–341, 2007.
- [5] K. Alton, "Dijkstra-like Ordered Upwind Methods for Solving Static Hamilton-Jacobi Equations," Ph.D. dissertation, The University of British Columbia, 2010.
- [6] R. Bellman, *Dynamic Programming*. Princeton University Press, 1957.
- [7] D. Bertsekas, *Dynamic Programming and Optimal Control*. Athena Scientific Belmont, 1995, vol. 1, no. 2.
- [8] J. Sethian and A. Vladimirov, "Ordered Upwind Methods for Static Hamilton-Jacobi Equations," *Proceedings of the National Academy of Sciences*, vol. 98, no. 20, pp. 11 069–11 074, 2001.
- [9] —, "Ordered Upwind Methods for Static Hamilton-Jacobi Equations: Theory and Algorithms," *SIAM Journal on Numerical Analysis*, vol. 41, pp. 325–363, 2003.
- [10] B. Girardet, L. Lapasset, D. Delahaye, C. Rabut, and Y. Brenier, "Generating Optimal Aircraft Trajectories with respect to Weather Conditions," in *ISIATM 2013, 2nd International Conference on Interdisciplinary Science for Innovative Air Traffic Management*, 2013.
- [11] J. P. Snyder, *Map Projections-A Working Manual*. USGPO, 1987, no. 1395.

TABLE III
TIME BENEFIT OF OPTIMAL ROUTE COMPARED WITH DIRECT ROUTE

		Average (s)	Min (s)	Max (s)	Q1 (s)	Q2 (s)	Q3 (s)
2013/09/30	All traffic	10.24	0	1950	0	0	7
	> 800km	25.99	0	1950	3	10	25
2013/10/02	All traffic	30.48	0	3415	0	1	13
	> 800km	76.88	0	3415	7	21	79
2013/10/04	All traffic	18.55	0	5084	0	0	7
	> 800km	46.71	0	5084	3	12	49

TABLE IV
COMPARISON BETWEEN OPTIMAL AND DIRECT ROUTES FOR THE FLIGHT JWD020 ON 2013/09/30.

	Speed (kt)	Distance (km)	Time (min)	Benefit (min)
Direct	511	5346	315	6
Optimal		5403	309	