Aircraft Trajectory Planning under Wind Uncertainties

K.Legrand, ENAC, 7 Avenue Edouard Belin, 31055 Toulouse France (klegrand@recherche.enac.fr)
S.Puechmorel, ENAC, 7 Avenue Edouard Belin, 31055 Toulouse France (stephane.puechmorel@enac.fr)
Daniel DELAHAYE, ENAC, 7 Avenue Edouard Belin, 31055 Toulouse France (delahaye@recherche.enac.fr)
Y.Zhu, ENAC, 7 Avenue Edouard Belin, 31055 Toulouse France (yaozhu537@gmail.com)

***Abstract*—Wind optimal trajectory planning is a critical issue for airlines in order to save fuel for all their flights. This planning is difficult due to the uncertainties linked to wind data. Based on the current weather situation, weather forecast institutes compute wind maps prediction with a given level of confidence. Usually, 30-50 wind maps prediction can be produced. Based on those predictions, airlines have to compute trajectory planning for their aircraft in an efficient way. Such planning has to propose robust solutions which take into account wind variability for which average and standard deviation have to be taken into account. It is then better to plan trajectories in areas where wind has low standard deviation even if some other plannings induce less fuel consumption but with a higher degree of uncertainty.**

In this paper, we propose an efficient wind optimal algorithm based on two phases. The first phase considers the wind map predictions and computes for each of them the associated wind optimal trajectory also called geodesic. Such geodesics are computed with a classical Bellman algorithm on a grid covering an elliptical shape projected on the sphere. This last point enable the algorithm to address long range flights which are the most sensitive to wind direction. At the end of this first phase, we get a set of wind optimal trajectories.

The second phase of the algorithm extract the most robust geodesic trajectories by the mean of a new trajectory clustering algorithm. This clustering algorithm is based on a new mathematical distance involving continuous deformation approach. In order to measure this mathematical distance between two trajectories, a continuous deformation between them is first built. This continuous deformation is called homotopy. For any homotopy, one can measure the associated energy used to shift from the first trajectory to the second one. The homotopy with the minimum energy is then computed, for which the associated energy measure the mathematical distance between trajectories. Based on this new distance, an EM clustering algorithm has been used in order to identify the larger clusters which correspond to the most robust wind optimal trajectories. This new approach avoids the main drawback of the classical approach which uses the mean of the trajectories issued from the first phase. This algorithm has been successfully applied to north Atlantic flights.

I. Introduction

With the development of aviation industry and the improvement of the environmental awareness, more and more airlines have paid attention to reduce fuel consumption during daily flight operations. Airlines pursue to minimize the adverse effects of headwinds, or maximize the beneficial effect of tailwinds when planning flight trajectories. According to the jet stream profile of the world shown in figure 1, it is easy to notice that the jet streams are in the east-west direction instead of the north-south direction. Therefore, the flights of east-west routes are affected more significantly by en-route winds than the flights of north-south routes. In order to achieve the best flight performance in terms of the flight time and the fuel consumption, airlines may adjust the flight trajectories based on en-route wind profiles. Consequently, it is necessary to consider en-route wind effects when planning flight trajectories. However, it is difficult to identify the most suitable trajectory in a complex wind field. The wind directions and strength are varying in different regions, at different altitudes and different times. Even though the problem is complex to solve, it may benefit airlines in terms of fuel cost and on-time performance if the optimal long-haul routes are able to be planned.

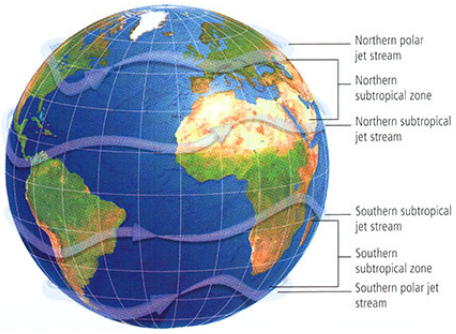


Fig. 1. Jet locations

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, such as Dijkstra algorithm [5] or A* algorithm [14], 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. Some others algorithms work on the continuous space. Those algorithms are based on front propagation methods such as Level Set methods, Fast Marching methods and Ordered Upwind methods. These different algorithms are developed by Sethian in [8]. In [3], Petres 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 behaviors more complex than a linear reaction to currents. In [10], Alton uses the Ordered Upwind algorithm with the Semi-Lagrangian method to generate optimal trajectories.

However, uncertainties related to the trajectory such as those in the weather conditions, cannot be fully eliminated; therefore, deviations between the actual and predicted trajectories are unavoidable. Wind is one of the most critical issue in the dispersion linked to predicted trajectory.

Usually, aircraft are optimizing trajectory in order to minimize some criteria : fuel, time, etc... When such planning is done in presence of wind one must take also into account the robustness of the planned trajectory. As a matter of fact airlines prefer to fly less efficient trajectories in terms of fuel but with a higher robustness.

In this paper, we address this robust trajectory planning in presence of wind with some uncertainties.

Weather forecast usually propose several possible situation by producing Ensemble Prediction. Ensemble Prediction Systems (EPS) are an approach to weather forecasting that has been adopted by the Numerical Weather Prediction centers in order to characterize and quantify the uncertainty inherent to prediction [9], a concept that cannot be captured with deterministic forecasts. This prediction technique involves generating a representative sample of the possible future states of the atmosphere. This collection of individual forecasts, called members, is generated by modifying the initial conditions and/or the meteorological model equations or parameters [1], [2].

The paper is organized as follow. The first part describes the algorithm used to compute wind optimal trajectories with a focus building of the network used by such algorithm. The second part presents the clustering algorithm and introduces a new mathematical distance between trajectories. The third part gives some results of the application of this new concept to weather data with different dispersion over the Atlantic Ocean and it shows how such algorithm can identify robust wind optimal trajectories.

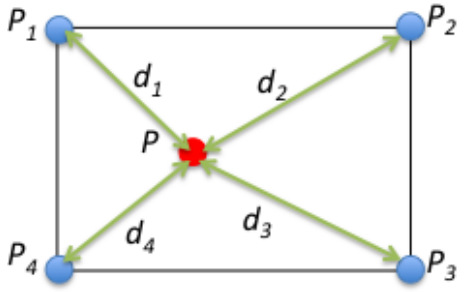


Fig. 2. Metric interpolation

II. Wind Optimal Trajectory Computation

A. Wind Grid Computation and Interpolation

We consider a 3-degree of freedom point-mass model of a fixed-wing aircraft flying though the North Atlantic Ocean. As an assumption, only cruise part of the flight is considered not only for the simplicity purposes but also because the cruise part is the majority of the flight. Additionally, we assume that the aircraft is flying at constant flight level. In this paper, we do not take temperature issue into account. As a result, we note that based on those assumptions, more complex problems can be simulated by applying this methodology easily.

We compute the optimal trajectory based on the wind predictions with a classical Bellman algorithm. In order to use Bellman algorithm to solve the problem, we need first to build a wind grid which stores wind data information.

1) Generate the wind grid: We generate a grid of size $N \times M$ nodes on he North Atlantic Ocean. The area from latitude 30 to latitude 70 and from longitude -90 to longitude 10 is taken into account. Each integer latitude and longitude point is regarded as a node. In order to generate smooth trajectories, we divide each latitude and longitude into 10 boxes. As a result, a 400 $\times$ 1000 grid table is generated.

2) Wind data interpolation: Note that, the wind data only contain the information at integer latitude and longitude node and we need to have the information at all nodes. We use Shepard's Method[4] to do such interpolation.
Let $F(P)$ be a function of the point $P = (x, y)$ defined for all P in the real plane $\mathbb{R}^2$, the value at point P is the weighted average of the values at nearby 4 data point P_1, P_2, P_3 and P_4(integer node). Denote the value of F at P_i by F_i and d_i be the distance between P_i and the generic point P in $\mathbb{R}^2$(See figure 2). The result was established by the function:

$$F(P) = \left[\sum_{i=1}^{4} F_i \prod_{j \neq i} d_j\right] \Bigg/ \left[\sum_{i=1}^{4} \prod_{j \neq i} d_j\right]$$

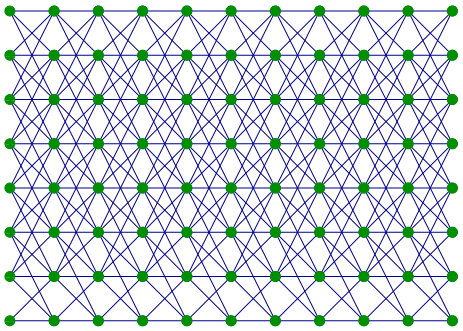


Fig. 3. Graph used for the wind optimal trajectory design.

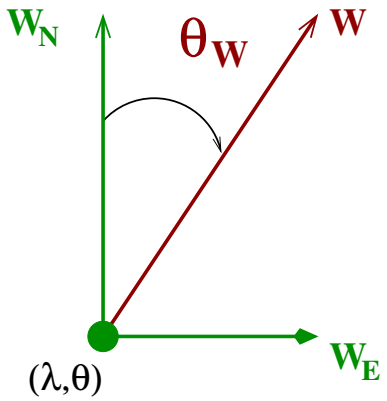


Fig. 4. Information contained in each node

B. Bellman Algorithm

In order to generate wind optimal trajectories, we start building a graph $G = \{\mathcal{N}, \mathcal{L}\}$ based on the wind grid (see figure 3), for which the set $\mathcal{N}$ represents the nodes and $\mathcal{L}$ the links.

Each node stores the following information : Latitude ϕ, longitude λ, altitude z, the east wind component W_E and the north wind component W_N. Based on those initial data coming from the wind grid, we compute also the wind norm $\|\vec{W}\| = \sqrt{W_E^2 + W_N^2}$ at each node and the associated wind bearing θ_W (see figure 4).

As it can be seen on figure 3, we have structured our graph into layers in order to speed up the Bellman algorithm. As a matter of fact, thanks to this structure, only one Bellman algorithm iteration is requested to find the minimum path.

Each node has also a list of successive neighbors which are represented by the blue links on figure 3. Each node (except the extreme north and extreme south) has some neighbors in the north direction and in the south direction (in our case, two in the north and two in the south). Such limitation will ensure smooth trajectory, avoiding sharp turns. Each link $l \in \mathcal{L} = (\mathcal{N}_i, \mathcal{N}_j)$ connects one origin node

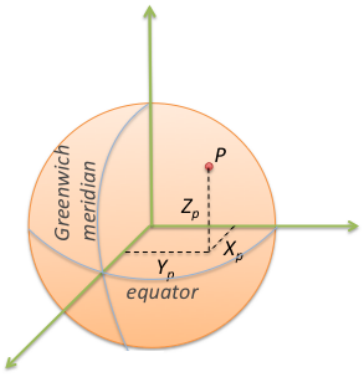


Fig. 5. The Cartesian coordinates

N_o and one destination node N_d. The grand circle distance of link l, d_l is given by the following formula :

$$d_l = R.\arcsin\left\{\|\vec{P}_d \wedge \vec{P}_o\|\right\}$$

where $\vec{P}_o = (x_o, y_o, z_o)^T$ $\vec{P}_d = (x_d, y_d, z_d)^T$ are the Cartesian coordinates of the nodes N_o and N_d, $\wedge$ is the vector product and R is the radius of the earth. For a given node P (see figure 5), the Cartesian coordinates are given by the following formula :

$$\vec{P} = \begin{cases} x = R.\cos(\phi) * \cos(\lambda) \\ y = R.\cos(\phi) * \sin(\lambda) \\ z = R.\sin(\lambda) \end{cases}$$

Each link contains also its associated bearing (see figure 6) θ_l which is given by the following formula :

$$\begin{cases} \theta_l(N_o, N_d) = \arctan\left(\frac{y}{x}\right) \\ y = \sin(\Delta_\lambda).\cos(\phi_d) \\ x = \cos(\phi_o).\sin(\phi_d) - \sin(\phi_o).\cos(\phi_d).\cos(\Delta_\lambda) \\ \Delta_\lambda = \lambda_d - \lambda_o \end{cases}$$

Based on the previous equation, one can now compute the tail wind on each extremities of the link l (TW_{lo} and TW_{ld}) :

$$TW_o = \|\vec{W}_o\|.\cos(\theta_l - \theta_{W_o})$$

$$TW_d = \|\vec{W}_d\|.\cos(\theta_l - \theta_{W_d})$$

Those two tail winds are then averaged and associated to each link :

$$TW_l = \frac{TW_O + TW_d}{2}$$

This last tail wind will be used for the cost associated to each link in the shortest path computation.

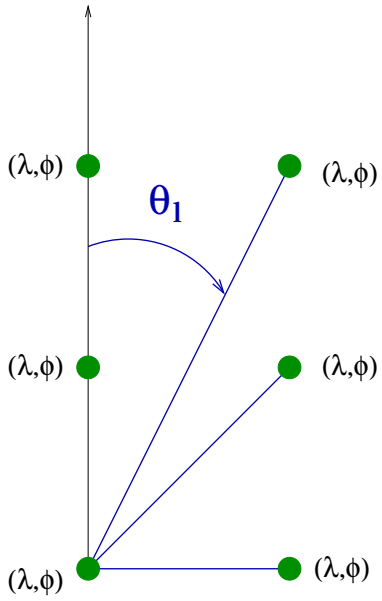


Fig. 6. Information contained in links

To compute the wind optimal trajectory, we will consider for each link, the time needed by aircraft to connect node N_o to node N_d. This time t_l is given by :

$$t_l = \frac{d_l}{T_a + TW_l}$$

where T_a is the true airspeed of the aircraft.

Having a graph with layer structure, we have implemented a Bellman-Ford algorithm for finding the shortest between a node at the extreme left (N_o) and all the nodes at the extreme right (N_d). The algorithm is organized into three steps :

STEP 1: Initialization This step initializes distances (dist) from source to all vertices as infinite and distance to source itself as 0.

STEP 2 : Propagation The source node is first considered and its associated neighboring links.

Starting from the source node $N_O = src$ (first column) and for each link associated to N_o the algorithm marks the neighboring nodes of $N_O = src$ with the following rule ;

- if $dist[N_d] > dist[N_o] + d_l$ then update $dist[N_d] \Rightarrow dist[N_d] = dist[N_o] + d_l$ (keep in node N_d the node N_o which has been use for this update

Shift to the next column (column 2) and apply the same rule to all nodes which have been updated in order to propagate the distance update to the third column. This process is repeated until the propagation reach the last column (on the right).

STEP 3 : Path building If one want to compute the shortest path for the source node *src* to any destination nodes on the right (*dest*), one first select a destination node among the nodes belonging to the last column (column number K). Select the node N_{K-1} in column $K-1$ which has updated the *dest* node in column K. Then, select the node N_{K-2} in column $K-2$ which has updated the node N_{K-1} in column $K-1$ and so on until the source node is reached in this back propagation process.

For each weather sample, such minimum time path algorithm is computed in order to create a set of wind optimal trajectories that has to be clustered.

III. Trajectory Clustering Algorithm

A. Mathematical Distance Between Trajectories

The question of gathering mobile trajectories into clusters of similar shapes first arose in the context of shape recognition. Within this frame, contours of objects are closed curves that describe the most important visual feature that one wants to classify. A very important constraint to take into account is the parametrization invariance: the shape of an object is independent on the way its contour is followed. In its seminal paper, Kendall introduced the notion of shape manifold [11]: the originality of its work was the use of a differential geometry setting to implicitly enforce the invariance with respect to shape-preserving transformations. Curves were represented as finite sequences of distinguished points, called landmarks. Some related algorithms were eventually designed for air traffic analysis applications. In a study conducted by the Mitre corporation on behalf of the Federal Aviation Authority (FAA) [6], a spectral clustering algorithm was applied to sampled trajectories. Only the distance between landmarks was used, no invariance under euclidean transformations were imposed. Due to the high computational complexity, a random projection was first applied to the data in order to reduce the dimension of the samples. The most important limitation of this approach is that the shape of the trajectories is not taken into account when applying the clustering procedure unless a re-sampling procedure based on arc-length is applied: changing the time parametrization of the flight paths will induce a change in the classification. Methods based on times series as surveyed in [12], [17] are appealing, but turn out to be inadequate for the present application. Finally, functional data statistics [7], [15] provides a powerful framework, still lacking the re-parametrization invariance. In this section, flight paths will be modeled as points in an infinite dimensional riemanian manifold. An intrinsic notion of distance exists in this setting and is defined as the infimum of the length of the paths connecting two points. Having this at hand allows the use of standard, distance based algorithms like k-means, k-mediods or hierarchical clustering.

B. Trajectories registration

A flight path may be modeled as a smooth curve $\gamma\colon [a,b] \to \mathbb{R}^3$ that maps a time to a position. Two distinct trajectories γ_1, γ_2 are most of the time defined on different time intervals, say $[a_1,b_1]$ (resp. $[a_2,b_2]$) for γ_1 (resp. γ_2), making the comparison between them quite awkward. This issue is well known in the field of functional data statistics as the registration problem. In a formal sense, it amounts to find a pair (ϕ_1,ϕ_2) of strictly increasing diffeomorphisms $\phi_1\colon [0,1] \to [a_1,b_1]$, $\phi_2\colon [0,1] \to [a_2,b_2]$ such that the transformed curves $\gamma_1 \circ \phi_1$, $\gamma_2 \circ \phi_2$, defined on the common interval $[0,1]$, are as similar as possible. The special problem instance:

$$\min_{\phi_1,\phi_2} \int_0^1 \|\gamma_1 \circ \phi_1(t) - \gamma_2 \circ \phi_2(t)\|^2 \, dt$$

gives the Frchet distance between γ_1, γ_2. Computing the optimal ϕ_1, ϕ_2 is a difficult task, unless the curves are assumed to be polygonal. Furthermore, as mentioned in [15], the registration procedure may remove some important features from the data: the extra degree of freedom provided by the so-called warping functions ϕ_1, ϕ_2 may have the detrimental effect of registering curves that does not need it [16]. A discrete relative to the Frchet distance is known as dynamic time warping and may be used to compare sampled sequences. Nevertheless, it suffers from the same drawback.

On the end of the other scale, a much simple procedure is to select only affine transformations for the warping functions. Given a trajectory $\gamma\colon [a,b] \to \mathbb{R}^3$, the affine registration is $\gamma \circ \phi$ with:

$$\phi\colon t \in [0,1] \mapsto a + (b-a)t$$

It amounts to shift the time origin so as to make it coincident with 0, then to scale by the length $b-a$ of the time interval.

In between, registration procedures based on time landmarks or monotonic polynomial approximation may be used [16]. Most of the time, a penalty criterion must be added to the similarity measure in order to avoid the over-registration phenomenon. It worth mentioning a special procedure, that will be used in the sequel, that is more in line with geometry. Given a smooth curve $\gamma\colon [a,b] \to \mathbb{R}^3$, its arclength is the smooth mapping:

$$s\colon t \in [a,b] \mapsto \int_a^t \|\gamma'(u)\| du$$

The length l_γ of the curve is just $s(b)$. Assuming that γ' never vanishes, s is strictly increasing, thus invertible. It induces a warping function:

$$\xi\colon t \in [0,1] \mapsto s^{-1}(t l_\gamma) \in [a,b]$$

that is characterized by the property:

$$\forall t \in]0,1[, \|D_t \gamma \circ \xi(t)\| = l_\gamma$$

where D_t stands for the derivative with respect to t. This warping function is intimately related to the landmarks approach of [11], as sampling evenly in the interval $[0,1]$ will result in a geometric even sampling on the curve itself (with respect to arclength). It will be denoted as the arclength warping in the sequel.

C. The manifold of paths

The idea of representing curves as point on an infinite dimensional manifold arises in the field of pattern recognition as an answer to the problem of assessing a degree of similarity between two shapes [13]. Within this frame, only closed curves were considered as they represent objects contours. In the context of air traffic, flight paths are never closed, unless the aircraft take off and land at the same airport, which is a quite uncommon for airliners. The initial mathematical model must be adapted to cope this specificity. For the sake of simplicity, all trajectories are assumed to be defined on the time interval $[0,1]$.

Definition 1: The space of immersions $\mathbf{Imm}([0,1],\mathbb{R}^3)$ is the set of smooth curves $\gamma\colon [0,1] \to \mathbb{R}^3$ with nowhere vanishing derivative in the interval $]0,1[$.

Generally speaking, an immersion will be a curve with nowhere vanishing derivative in the interior of its domain. It is clear that for such a curve the arclength is well defined and strictly increasing thus the geometric warping function exists. It may be used to perform a registration step to ensure that all curves are defined on $[0,1]$.

Given γ in $\mathbf{Imm}([0,1],\mathbb{R}^3)$, its derivative norm $\|D_t\gamma\|$ is a continuous mapping on the compact interval $[0,1]$ and thus has a non-zero minimum value m. If $\varepsilon\colon [0,1] \to \mathbb{R}^3$ is a smooth mapping such that $\sup_{[0,1]} \|D_t\varepsilon\| < m$, then $\gamma + \varepsilon$ will have a nowhere vanishing derivative and thus still belongs to $\mathbf{Imm}([0,1],\mathbb{R}^3)$. This indicates that this space has locally the structure of a vector space (in fact a Banach space) and globally the one of a differentiable manifold. To get rid of the influence of parametrization, the shape space is defined as a quotient with respect to all increasing diffeomorphisms of the interval $[0,1]$:

$$\mathcal{E} = \mathbf{Imm}([0,1],\mathbb{R}^3)/\mathbf{Diff}^+([0,1])$$

$\mathcal{E}$ inherits the manifold structure from $\mathbf{Imm}([0,1],\mathbb{R}^3)$. A point in $\mathcal{E}$ will be denoted by $[\gamma]$ and is an equivalence class of mappings $\gamma \circ \phi$ with $\phi \in \mathbf{Diff}^+([0,1])$. A tangent vector at $[\gamma]$ is a couple $([\gamma], v)$ where v is a smooth mapping from $[0,1]$ to $\mathbb{R}^3$. This mapping must be understood as an infinitesimal displacement field on the base curve γ. As usual, the set of tangent vector is called the tangent

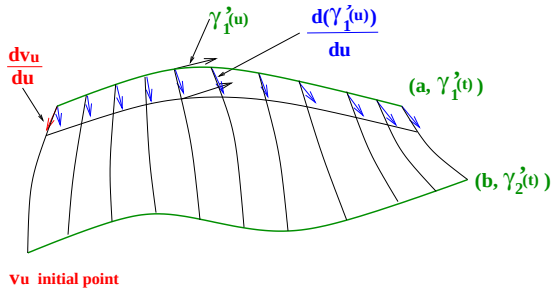


Fig. 7. Smooth path between two curves

bundle of $\mathcal{E}$, denoted by $T\mathcal{E}$. An riemanian metric can be introduced on $\mathcal{E}$, in the spirit of [13]:

$$g_{[\gamma]}(u,v) = \int_0^1 \langle u(t), v(t)\rangle (1 + A\kappa^2(t)) \|D_t\gamma(t)\| dt \quad (1)$$

$$+\mu\langle u(1), v(1)\rangle - \mu\langle u(0), v(0)\rangle \quad (2)$$

where γ is a representative curve of $[\gamma]$ and κ is the curvature of γ at t. The parameters A, μ are strictly positive real numbers that tune the respective importance of the curvature and the endpoints. Please note the difference with the original metric for closed curves that appears in the endpoints term. The riemanian metric is invariant under a changer of parametrization and thus does not depend on the particular choice of γ in the equivalence class $[\gamma]$.

A smooth path between two points $[\gamma_1], [\gamma_2]$ in $\mathcal{E}$ is represented by a smooth homotopy $\Phi \in \mathbf{Imm}([0,1], \mathbb{R}^3)$, that is a smooth mapping from $[0,1]^2$ to $\mathbb{R}^3$ such that:

- $\Phi(0,\bullet) = \gamma_1(\bullet), \Phi(1,\bullet) = \gamma_2(\bullet)$
- $\forall (s,t) \in [0,1]^2, D_t\Phi(s,t) \neq 0$

The derivative of Φ with respect to the homotopy parameter s, denoted by $D_s\Phi$n is a smooth curve on $[0,1]$, so that for a given s, the couple $([\Phi(s,\bullet)], D_s\Phi(s,\bullet))$ is a tangent vector in $T\mathcal{E}_{[phi(s,\bullet)]}$. An visual representation of an smooth homotopy along with the associated tangent vectors is given in figure (7).

Using the riemanian metric (1) on $T\mathcal{E}$, the energy of a path Φ can be defined in the usual way:

$$E(\Phi) = \int_0^1 g_{[\phi(s,\bullet)]}(D_s\Phi(s,\bullet), D_s\Phi(s,\bullet))\, ds \quad (3)$$

It is equivalent for a path to minimize the energy or the length, the former is preferred as it saves a square root in the expression. The critical points of E are called geodesic paths. Since it is only a local condition (vanishing derivative), it may not correspond to a minimum of E. If such a global minimum exists, a path realizing it is called a minimizing geodesic. In the finite dimensional setting, the Hopf-Rinov theorem may be invoked to prove the existence of a minimizing geodesic between arbitrary points. Unfortunately, it doesn't hold generally for infinite

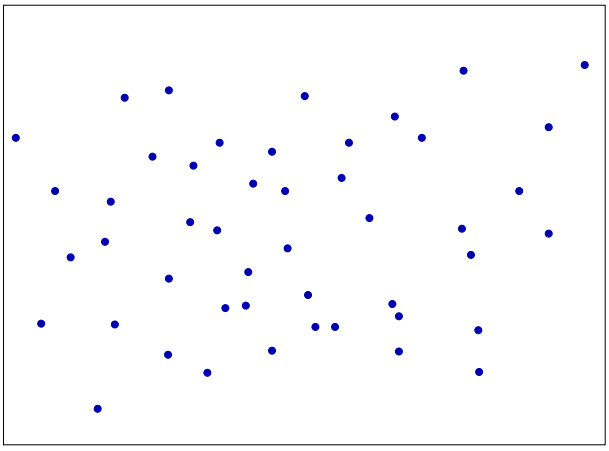


Fig. 8. On this metric space each trajectory is represented by a point (blue point)..

dimensional manifolds. It turns out that in the framework defined above, a minimizing geodesic exits between any two curves, thus making possible the definition of a distance on $\mathcal{E}$:

For any couple $([\gamma_1], [\gamma_2])$ in $\mathcal{E}^2$, the **distance between** $[\gamma_1]$ **and** $[\gamma_2]$ is given by:

$$\int_0^1 \sqrt{g_{[\phi(s,\bullet)]}(D_s\Phi(s,\bullet), D_s\Phi(s,\bullet))}\, ds$$

where Φ is any homotopy between γ_1, γ_2 realizing the minimum of E.

The distance d turns $\mathcal{E}$ into a metric space and can be used in any distance-based clustering algorithm.

D. Trajectory Clustering Algorithm

We consider a set of trajectories which has been build thanks to the Bellman algorithm and the trajectory distance defined in the previous section. Having such distance definition, one can gather together such trajectories in order to create clusters by using an adaptive clustering algorithm (hierachical clustering). Based on a set of N trajectories, clustering algorithm aims to partition such a set into K clusters. To reach this goal, the trajectories are consider as points in the associated metric space (see figure 8).

This algorithm uses two parameters, d_{min} and d_{max}, to respectively fuse clusters and create new clusters. Initially, each trajectory is considered as the centroid of a cluster. We then apply the three following principles one after the other:

- if two centroids are at a distance lower than d_{min}, we fuse them into a single cluster, of which the resulting centroid is the barycenter of the two initial centroids.

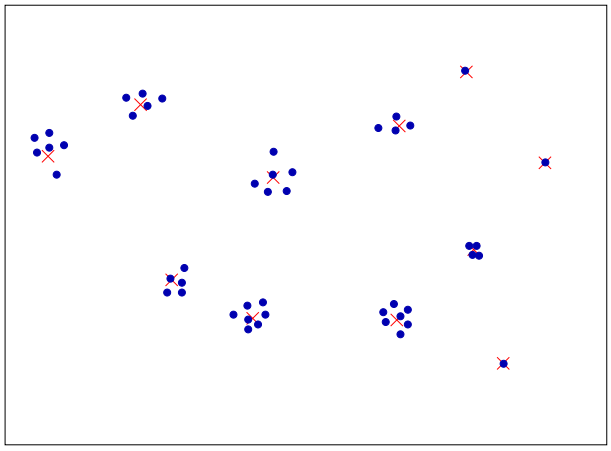


Fig. 9. In this example the algorithm find eleven clusters with different features.

The barycenter is computed the following way :

$$\mu_i = \frac{1}{N} \sum_{i=1}^{i=N} \gamma_i$$

- a new individual is aggregated to a cluster if its distance from the closest centroid is lower than d_{max} and in this case we compute the new global centroid.
- Otherwise, we create a new cluster containing the single trajectory.

The number of cluster is also a result of the algorithm. An example of clustering resust is given on figure 9.

for each cluster c, one can compute the following features :

- Number of trajectories in the cluster N_c
- Mean trajectory which is the cluster centroid (γ_c
- Dispersion of the cluster

$$\sum_{i=1}^{N_c} \left\| \gamma_j - \gamma_c \right\|^2$$

where $\|.\|$ is the norm in the trajectory metric space.

This clustering algorithm has been used to classify trajectories produced by the Bellman algorithm.

The overall processing can be summarized by the figure 10

IV. Results

This section presents the initial results that has been produced by this new algorithms. First we have consider two wind samples over the Atlantic ocean from two different day (January 09, 2016 and February, 14 2016). An example of such map is given on figure 11.

Those two days present different wind dispersion data with 34 wind samples at each point of the wind grid which (resolution : 1 degree).

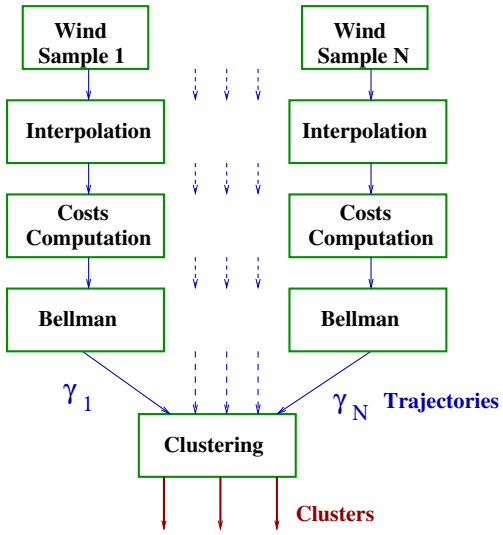


Fig. 10. Overall structure of the algorithm

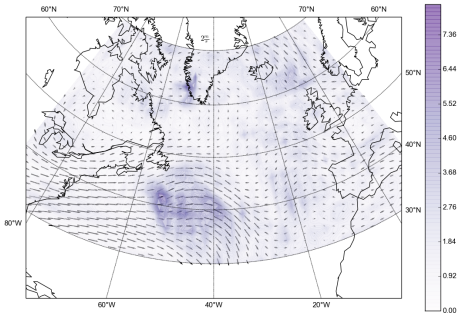


Fig. 11. Example of wind distribution over the Atlantic ocean

Based on those wind data, we have applied the algorithm in order to compute the wind optimal route between two points. The origin has been settled at $\vec{P}_o$=(lat, long): (λ=30, θ=-90) and the destination at $\vec{P}_d$(lat, long): (λ=60, θ=10). The first data sample presents less dispersion and may result in a better planing in terms of robustness. The Bellman algorithm has been applied 34 times between points $\vec{P}_o$ and $\vec{P}_d$ and has generated 34 trajectories that have been represented on figure 12.

Those trajectories have been clustered thanks to the new distance that has been developed. Two clusters have been extracted as it can be seen on figure 13. The first cluster gather together 30 trajectories and the second one 4 trajectories. The trajectories belonging to the first cluster are more robust and may be considered as the most robust wind optimal trajectories between $\vec{P}_o$ and $\vec{P}_d$. Fortunately, the best trajectory in terms of flight duration belongs also

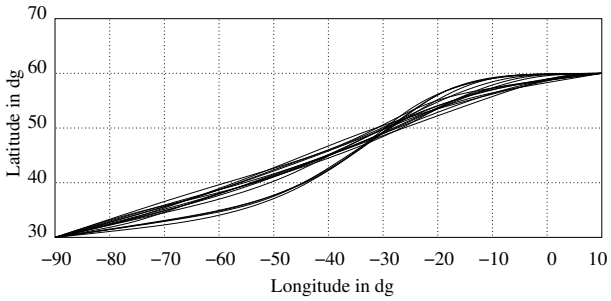


Fig. 12. Wind optimal trajectories for the first wind sample set (January 09, 2016)

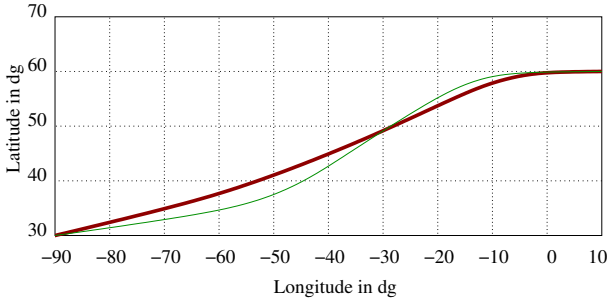


Fig. 13. Cluster produced for the first wind sample set. The cluster which has the most representatives is represented in red.

to cluster 1 (Flight time 11h34'; to compute this flight time, a True Air Speed of 450kts has been considered). The best trajectory in cluster 2 has a flight time of 11h47'. If the situation was opposite, one has to balance the associated robustness, which is linked to the number of representative in each cluster, with the associated flight duration.

The second wind sample data from February, 14 2016 is more critical in this sense. This day has much more dispersion in the wind data and the trajectories produced by the Bellman algorithm are also more spread as it can be seen on figure 14.

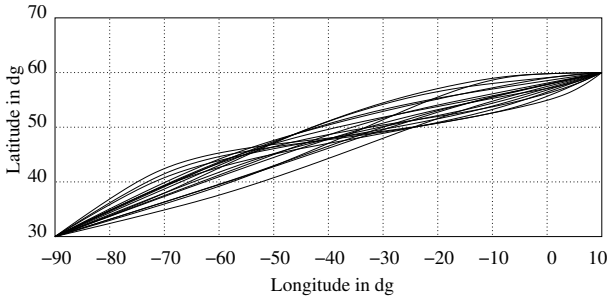


Fig. 14. Wind optimal trajectories for the second wind sample set (February, 14 2016)

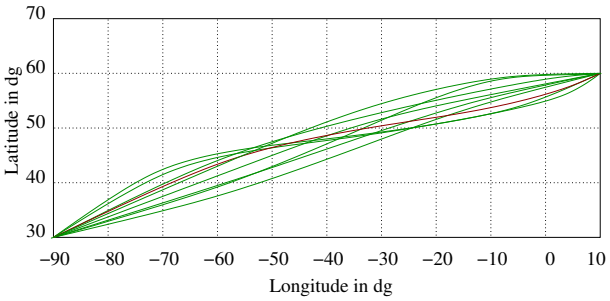


Fig. 15. Cluster produced for the second wind sample set.

Those trajectories have been also clustered and the associated cluster are represented on figure 15. In this case, nine clusters have been extracted with a maximum of five representative. In this case, there is not a big difference between cluster in terms of representative number, and we can say that the associated robustness is the same. In this case, one must select the one with the minimum flight duration.

V. Conclusion

This paper has introduced a new approach for designing robust wind optimal trajectory. A methodology for computing tail wind on each link on a grid network over the Atlantic ocean has been introduced. First, wind has been interpolated on a more accurate grid, then tailwind formula on each link has been established and tail wind on each link has been computed. Based on this network an efficient adaptation of the Bellman algorithm has been proposed thanks to the layers structure of the associated graph. In order to cluster trajectories produced by Bellman algorithm a hierarchical clustering algorithm has been developed and a new exact mathematical distance between trajectories has been introduced. This new methodology has been successfully applied to real wind data in order to identify robust wind optimal trajectories.

References

[1] Arribas A, Robertson K.B, and Mylne K.R. Test of a poor mans ensemble prediction system for short-range probability forecastingensemble experiments on numerical weather prediction error and uncertainty for a north pacific forecast failure. *Monthly Weather Review*, 133(7):1825–1839, 2005.

[2] Lu C, Yuan H, Schwartz B.E, and Benjamin S.G. Short-range numerical weather prediction using time-lagged ensembles. *Weather and Forecasting*, 22(3):580–595, 2007.

[3] Pêtrès C, Pailhas Y, Patron P, Petillot Y, Evans J, and D. Lane. Planning for autonomous underwater vehicles. *IEEE Transactions on Robotics*, 23(2):331–341, 2007.

[4] Shepard D. A two-dimensional interpolation function for irregularly-spaced data. In *ACM Conf.* ACM, 1968.

[5] Dijkstra E. A note on two problems in connexion with graph. *Numerische Mathematlk*, 1:269–271, 1959.

[6] M. Enriquez. Identifying temporally persistent flows in the terminal airspace via spectral clustering. In FAA-Eurocontrol, editor, *ATM Seminar 10*, 06 2013.

[7] F. Ferraty and P. Vieu. *Nonparametric Functional Data Analysis: Theory and Practice*. Springer Series in Statistics. Springer, 2006.

[8] Farin G. *Level Set Methods and Fast Marching Methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science*, volume 3. Cambridge University Press, 1999.

[9] Hacker J.P, Krayenhoff E.S, and Stull. R.B. Ensemble experiments on numerical weather prediction error and uncertainty for a north pacific forecast failurea note on two problems in connexion with graph. *Weather and forecasting*, 18(1):219–260, 2003.

[10] Alton K. *Dijkstra-like Ordered Upwind Methods for Solving Static Hamilton-Jacobi Equations*. PhD thesis, The University of British Columbia, 2010.

[11] David G. Kendall. Shape manifolds, procrustean metrics, and complex projective spaces. *Bulletin of the London Mathematical Society*, 16(2):81–121, 1984.

[12] T. Warren Liao. Clustering of time series data - a survey. *Pattern Recognition*, 38:1857–1874, 2005.

[13] Peter W Michor and David Mumford. Riemannian geometries on spaces of plane curves. *J. Eur. Math. Soc. (JEMS)*, 8:1–48, 2006.

[14] Hart P, Nilsson N, and Raphae B. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.

[15] J. Ramsay and B.W. Silverman. *Functional Data Analysis*. Springer Series in Statistics. Springer New York, 2006.

[16] J. O. Ramsay and Xiaochun Li. Curve registration. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 60(2):351–363, 1998.

[17] Sangeeta Rani and Geeta Sikka. Recent techniques of clustering of time series data: A survey. *International Journal of Computer Applications*, 52(15):1–9, August 2012. Full text available.