DOI. No. 10.1109/MAES.2018.170050

Feature Article:

Robust Aircraft Optimal Trajectory in the Presence of Wind

***Karim Legrand, Stephane Puechmorel, Daniel Delahaye, Yao Zhu,* ENAC, Toulouse, France**

INTRODUCTION

As environmental awareness has increased, airlines are paying attention to fuel consumption during daily flight operations with the hope of reducing consumption. Because the direction and strength of the wind affects the fuel consumption of an aircraft, airlines are seeking 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, it is easy to notice that the jet streams flow in the east-west direction instead of the north-south direction. Therefore, the flights on east-west routes are affected more by en route winds than the flights on north-south routes. In order to achieve the best flight performance in terms of flight time and 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 because such field is not easy to estimate [17], [25], [8], [9], [26]. The wind directions and strength are varying in different regions, at different altitudes, and different times. Even though the problem is complex, solving it may benefit airlines in terms of fuel cost and on-time performance.

Planning optimal trajectories is a rich and dynamic research domain with many application areas like robotics, space, and aviation [6], [2], [3]. Depending on the problem, 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 the presence of currents in a two-dimensional space. Several methods, such as Dijkstra algorithm [10] or A* algorithm [20], 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. Other 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 [13]. In [5], 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 [15], Alton uses the Ordered Upwind algorithm with the Semi-Lagrangian method to generate optimal trajectories. In [3], a straightforward wind computation approach is presented, which relies on the difference between the predicted motion of the aircraft and the real motion measured by sensors. This wind estimation is fully tactical.

However, trajectory uncertainties (related to weather conditions) cannot be completely eliminated; therefore, there are discrepancies between actual and projected trajectories.

Usually, aircraft trajectories are optimized 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 article, we address this robust trajectory planning in the presence of wind with some uncertainties.

Weather forecasts usually propose several possible situations by producing an ensemble prediction. Ensemble prediction systems 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 [14]. 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], [4].

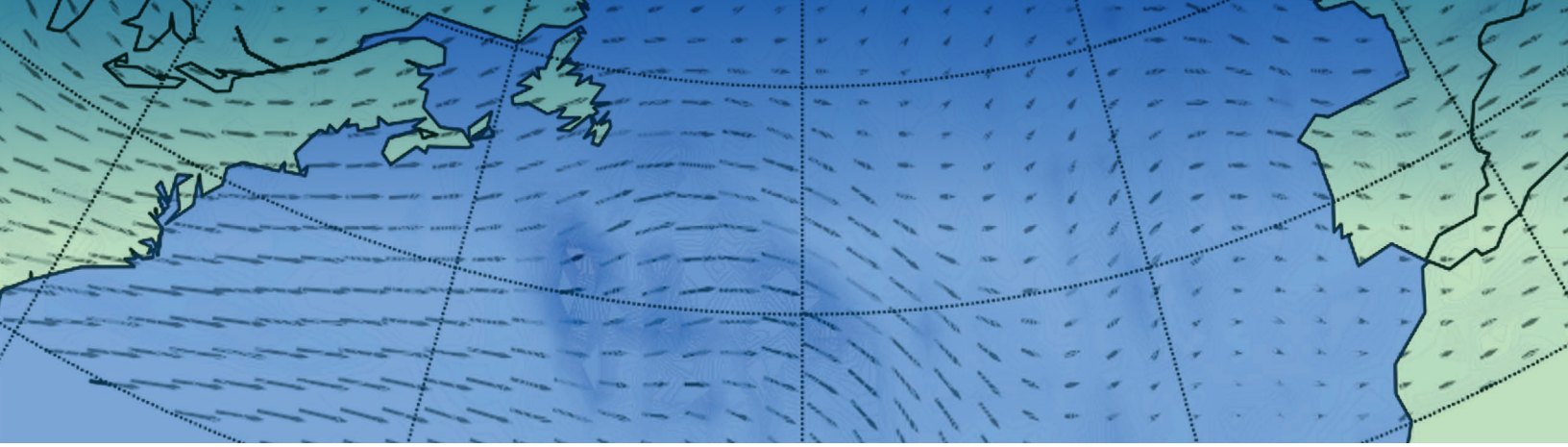
This article is organized as follows. The first part describes the algorithm to design aircraft optimal trajectory in the presence of wind. The second part presents the clustering algorithm with a focus on a new mathematical distance between trajectories. The third part gives some results of the application of our algorithm to the oceanic airspace in order to generate robust aircraft optimal trajectories in the presence of wind.

WIND OPTIMAL TRAJECTORY COMPUTATION

WIND GRID COMPUTATION AND INTERPOLATION

We consider a three-degree-of-freedom point-mass model of a fixed-wing aircraft flying over the North Atlantic Ocean. As an

Authors' current address: ENAC, 7 Avenue Edouard Belin, 31055 Toulouse France, E-mail: (klegrand@recherche.enac.fr).
This paper is an invited paper as a DASC best student paper.
Manuscript received February 9, 2017, revised July 11, 2017, November 30, 2017, March 12, 2018, and ready for publication March 13, 2018.
Review handled by M. Greco.
0885/8985/18/$26.00 © 2018 IEEE



assumption, only the cruise part of the flight is considered, not only to simplify the problem but also because the cruise part of the flight is the majority of the flight. Additionally, we assume that the aircraft is flying at a constant flight level. In this article, we do not take temperature 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 the 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 the 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 × 1,000 grid table is generated.

2) Wind data interpolation: Note that the wind data only contain the information at the integer latitude and longitude node, and we need to have the information at all nodes. We use Shepard's Method [7] 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 four data points 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 1). The result was established by the function:

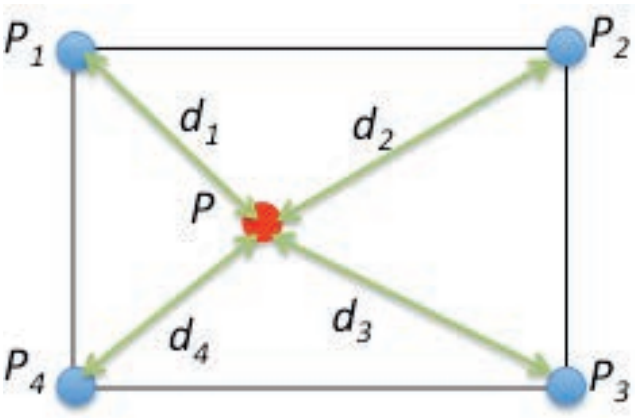


Figure 1.
Metric interpolation.

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

BELLMAN ALGORITHM

In order to generate wind optimal trajectories, we start building a graph based on the wind grid (see Figure 2), 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 the initial data coming from the wind grid, we compute also the wind norm

$$\left\|\vec{W}\right\| = \sqrt{W_E^2 + W_N^2}$$

at each node and the associated wind bearing θ_W (see Figure 3a).

The graph is structured into layers (see Figure 2) to improve the performance of the Bellman algorithm. As a matter of fact, thanks to this structure, only one Bellman algorithm iteration is needed to find the minimum path.

Each node has also a list of successive neighbors which are represented by the blue links on Figure 2. 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 $l \in \mathcal{L} = (\mathcal{N}_o, \mathcal{N}_d)$ connects one

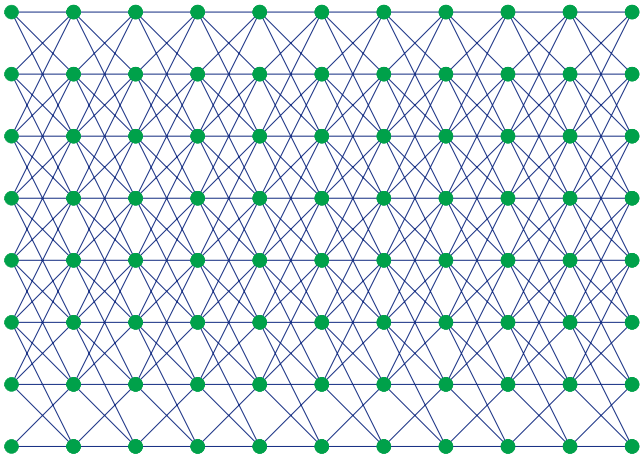


Figure 2.
Graph used for the wind optimal trajectory design.

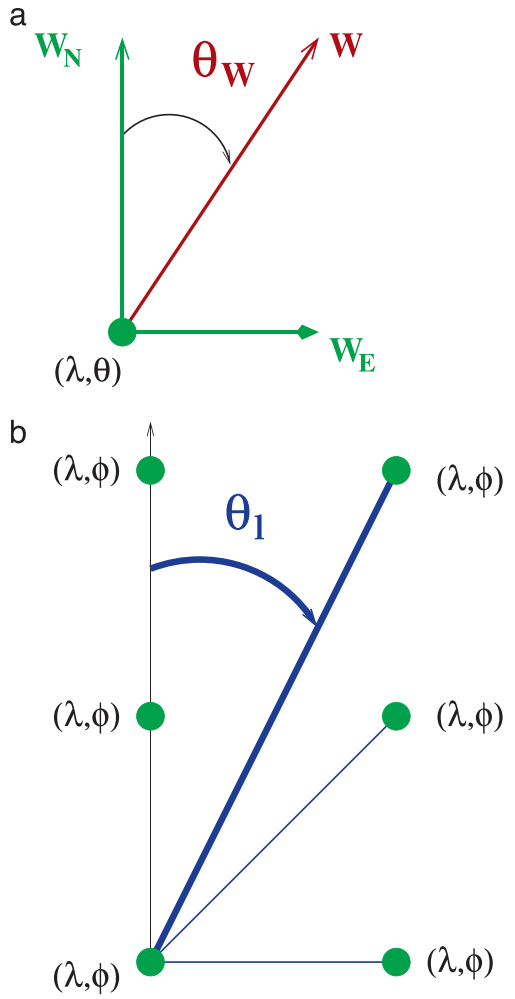


Figure 3.
Information contained in links. (a) Wind information contained in nodes. (b) Information contained in links.

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

$$d_l = R.\arcsin\left\{\left\|\vec{P}_d \wedge \vec{P}_o\right\|\right\} \tag{2}$$

where

$$\vec{P}_o = (x_o, y_o, z_o)^T \quad \vec{P}_d = (x_d, y_d, z_d)^T$$

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 4), the Cartesian coordinates are given by the following formula:

$$\vec{P} = \begin{cases} x = R.\cos(\phi) * \cos(\lambda) \\ y = R.\cos(\phi) * \sin(\lambda) \\ z = R.\sin(\lambda) \end{cases} \tag{3}$$

where λ is the latitude and ϕ is the longitude.

Each link contains also its associated bearing (see Figure 3b) θ_l which is given by the following formula:

$$\begin{cases} \theta_l(N_o, N_d) = \arctan\left(\dfrac{y}{x}\right) \\ y = \sin(\Delta_\lambda).\cos(\phi_d) \\ x = \cos(\phi_o).\sin(\phi_d) - \sin(\phi_o).\cos(\phi_d).\cos(\Delta_\lambda) \\ \Delta_\lambda = \lambda_d - \lambda_o \end{cases} \tag{4}$$

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

$$TW_o = \left\|\overrightarrow{W_o}\right\|.\cos(\theta_l - \theta_{W_o}) \quad TW_d = \left\|\overrightarrow{W_d}\right\|.\cos(\theta_l - \theta_{W_d}) \tag{5}$$

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

$$TW_l = \frac{TW_O + TW_d}{2} \tag{6}$$

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

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:

$$t_l = \frac{d_l}{T_a + TW_l} \tag{7}$$

where T_a is the true airspeed of the aircraft.

Having a graph with layer structure, a Bellman-Ford algorithm has been implemented for finding the shortest path 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 reaches the last column (on the right).

STEP 3: Path building To compute the shortest path for the source node *src* to any destination nodes on the right (*dest*), 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

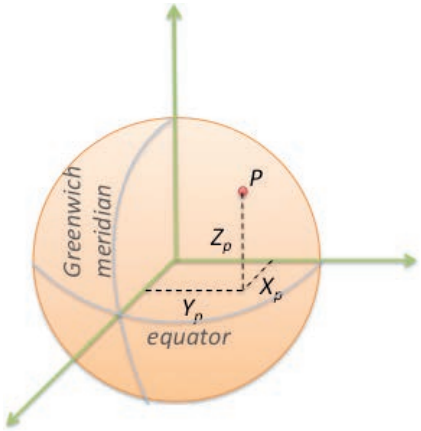


Figure 4.
The Cartesian coordinates.

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 trajectories that has to be clustered. To compute those trajectory clusters, a distance between trajectories has first to be defined.

TRAJECTORY CLUSTERING ALGORITHM

In order to group trajectories into clusters, one must first establish a mathematical distance between such mathematical objects.

MATHEMATICAL DISTANCE BETWEEN TRAJECTORIES

In a vector space, distances are very well defined.

For two points $\vec{P}_1 = (x_1, y_1)^T$ and $\vec{P}_2 = (x_2, y_2)^T$ in a plane (see Figure 5), the distance between them can be computed with the classic formula of the Euclidean distance (see Figure 5).

What is the distance if now the points $\vec{P}_1$ and $\vec{P}_2$ are replaced by two trajectories γ_1 and γ_2? Trajectories are infinite dimension mathematical objects which are not easy to manipulate.

One of the main results of this article is the establishment of such mathematical distance.

1) Current Trajectory Distances: An aircraft trajectory is a time sequence of coordinates representing the aircraft path over a period of time and may be represented by N-uple: $T = \{(x_1, y_1, z_1, t_1), (x_2, y_2, z_2, t_2), \ldots, (x_N, y_N, z_N, t_N)\}$ where N is the duration.

The simplest metric used for computing the distance between a pair of trajectories is the mean of coordinate distances.

Note that, the mean of distance metric makes three critical assumptions:

1. the durations of both trajectories are the same: $N^a = N^b = N$
2. the coordinates are synchronized $t_n^a = t_n^b$
3. the time sampling rate is constant $t_{n+1}^a - t_n^a = t_{m+1}^a - t_m^a$

It is evident that the mean of distance is very sensitive to the partial mismatches and cannot deal with the distortions in time.

To provide more descriptive information, the second order statistics such as median, variance, minimum and maximum distance may be incorporated.

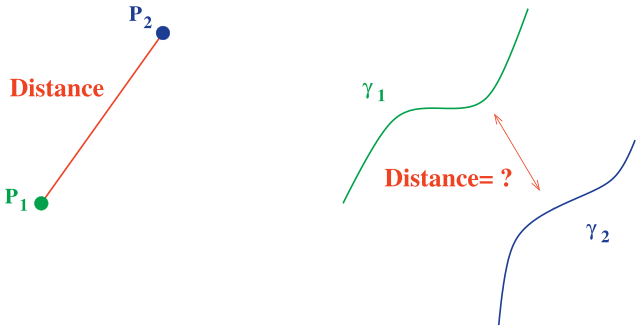


Figure 5.
On the left, two points $\vec{P}_1$ and $\vec{P}_2$ has been drawn for which the classical Euclidean distance is shown in red. On the right, two trajectories are drawn (γ_1, γ_2) for which a mathematical distance has to be determined.

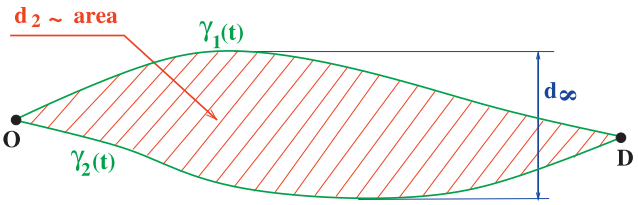


Figure 6.
Supremum norm distance.

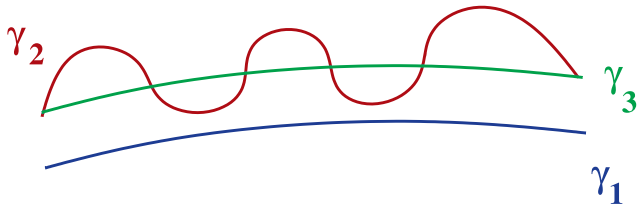


Figure 7.
Different trajectories with same sup distance. The sup norm between the green trajectory and the red trajectory is the same as the one computed between the green and the blue trajectories; meaning the such sup norm consider γ_1 and γ_2 at the same distance from γ_3 even if γ_1 and γ_2 are very different.

Although these statistics supply extra information, they inherit (even amplify) the shortcomings of the ordinary mean of distance metric m_1. None of the above metrics is sufficient enough by itself to make an accurate assessment of the similarity.

Another possible candidate for the distance between two trajectories γ_1 and γ_2 will simply be to take the supremum norm $d_\infty(\gamma_1, \gamma_2)$ (see Figure 6).

Since γ_1 and γ_2 are constant outside bounded intervals of $\mathbb{R}$, the supremum is well defined. However, this metric is not sensitive to global properties of curves. In the Figure 7, the curves γ_1 and γ_2 are at the same distance from γ_3 but have very different shapes.

From an operational point of view, γ_1 is just a shifted copy of γ_3 while γ_2 will probably not be realistic.

For trajectories γ_1, γ_2 with the same origin-destination pairs, $\gamma_1 - \gamma_2$ can be defined as a compactly supported mapping and an area distance between trajectories ($d_2(\gamma_1, \gamma_2)$) can be defined (see Figure 6).

This distance can handle more complex trajectories; however, it is sensitive to entanglements of the trajectory, discards the time continuity, and fails to distinguish two trajectories in opposite directions. Furthermore, it cannot be extended to the third dimension.

In order to introduce our new mathematical distance between trajectories, one must first give some representation definitions.

2) Representation: Since objects of interest are aircraft trajectories, we need to find an adapted framework in which computations may be made on trajectories as a whole. There are basically two ways of understanding what a trajectory is:

- The time/position approach, for which time is included in the representation. In this case, a trajectory can be represented as a mapping from a bounded interval of $\mathbb{R}$ (the life time of the trajectory) to $\mathbb{R}^3$ or $\mathbb{R}^6$ depending on whether speed is part of the data or not. Since there is an explicit dependence on time, there is a need to calibrate trajectories with time shifts for all applications involving trajectory comparison. We will see in the following that there is a mean of reducing the problem by calibrating automatically the origin of time.
- The shape approach, for which only the path is considered without time. Here, trajectories are understood as paths and time is not directly relevant (from a more formal point of view, we take the quotient of the trajectories understood as mappings by the group of diffeomorphisms acting on time), so that we may assume that the underlying lifetimes of trajectories are always the interval [0,1]. This framework is adapted for dealing with major flows estimation.

3) Trajectories as mappings: We will assume in the following that trajectories are given as mappings from a compact interval of $\mathbb{R}$ to $\mathbb{R}^3$. The case of mappings from $\mathbb{R}$ to $\mathbb{R}^6$ (that is with explicit speed, for example as given by radar tracking filter [8], [9]) can be derived with minor changes and thus will not be addressed here. Since physical trajectories are smooth unless there is a perturbing noise, we made the choice to take all trajectories as **s**mooth mappings from a compact interval of $\mathbb{R}$ to $\mathbb{R}^3$.

The first point to deal with is the necessary calibration of the origin of time for trajectories comparison. Remembering that there is an explicit dependence on time, one cannot just time shift one trajectory in time in order to make it coincident with another in order to compare them: this will result in forgetting distortions in time, that is trajectories with the same range (as mappings) but different positions at different times may become equal.

Since we choose to compare trajectories as mappings, a good candidate for computing the distance will be to integrate over time (like for the area distance) and to evaluate a mean error instead of the raw sum of squares:

$$d_T(\gamma_1,\gamma_2)^2 = \frac{1}{2T}\int_{-T}^{T}\left\|\gamma_1(t)-\gamma_2(t)\right\|^2 dt \tag{8}$$

with $T > 0$. This formula defines a semi-distance between trajectories γ_1 and γ_2.

The previous family of semidistances has nice features because of the scaling ability, but since it is not a single metric, it is difficult to use standard algorithms based on distances (for example, classification algorithms). There is thus a need for another definition of proximity between trajectories that will yield a single value while capturing interesting global characteristics.

NEW DISTANCE DEFINITION BASED ON HOMOTOPY

Before introducing our homotopic distance between trajectories, one must introduce how we cope with time difference between trajectories.

1) Parametrization invariance: The parametrization invariance is a very important feature: the shape of an object is independent of the way its contour is followed. In a seminal paper, Kendall introduced the notion of shape manifold [16]: 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) [11], 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 resampling 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 [18], [24] are appealing, but are inadequate for the present application. Finally, functional data statistics [12], [22] provides a powerful framework, still lacking the reparameterization invariance. In this section, flight paths will be modeled as points in an infinite dimensional riemannian manifold. In such space, each point is considered as a full trajectory. 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.

2) Trajectories registration: A flight path may be modeled as a smooth curve $\gamma:[a,b]\to\mathbb{R}^3$ that maps a time to a position (at each time one can determine the associated 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 finding a pair (ϕ_1, ϕ_2) of strictly increasing diffeomorphisms $\phi_1 : [0,1] \to [a_1, b_1]$, $\phi_2 : [0,1] \to [a_2, b_2]$ 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:

$$\min_{\phi_1,\phi_2}\int_0^1\left\|\gamma_1\circ\phi_1(t)-\gamma_2\circ\phi_2(t)\right\|^2 dt \tag{9}$$

gives the Fréchet distance between γ_1, γ_2. Computing the optimal ϕ_1, ϕ_2 is a difficult task, unless the curves are assumed to be polygo-

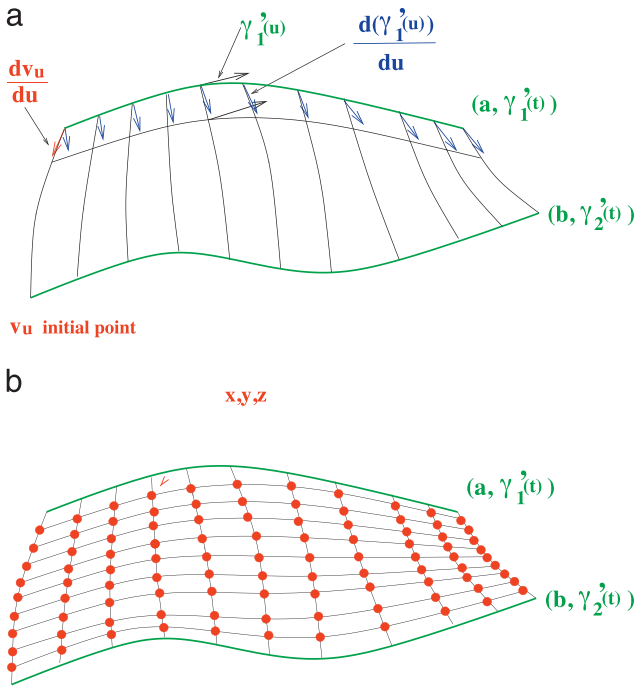


Figure 8.
(a) Smooth path between two curves. (b) Structure of the grid used for homotopy energy minimization. Each red point has two-dimensional coordinates (x, y) for which an optimization algorithm is used for searching the z coordinates which minimize the energy of the homotopy connecting γ_1 to γ_2.

nal. Furthermore, as mentioned in [22], 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 do not need it [23]. A discrete relative to the Fréchet distance is known as dynamic time warping and may be used to compare sampled sequences. Nevertheless, it suffers from the same drawback (two trajectories having different time sampling will be considered as different which is not relevant for our application).

3) Distance based on Homotopy between Trajectories: In order to compute the distance between two trajectories (γ_1, γ_2), a time regularization is first applied to trajectories. Then, a homotopy Φ between γ_1, γ_2 is built for which its associate energy is computed for extracting a distance metric (such distance has been developed by S. Puechmorel [21]).

Let a be the origin of the trajectory γ. We have: $\gamma(t) = a + \int_0^t \gamma'(s)ds$, so a couple ($a$, γ') ($\in \mathcal{W}$) with γ' compactly supported defines a trajectory.

A homotopy between (a, γ_1') and (b, γ_2') is a continuous mapping $\Phi : [0,1] \to \mathcal{W}$ such that $\Phi(0) = (a, \gamma_1'), \Phi(1) = (b, \gamma_2')$. Intuitively, a homotopy is a continuous deformation between two trajectories (see Figure 8a).

The deformation energy between γ_1 and γ_2 is linked to the distance between those trajectories and can be computed with the energy of the homotopy between γ_1 and γ_2:

$$E(\Phi) = \int_0^1 \left(\left\| \frac{\partial v_u}{\partial u} \right\|^2 + \int_{\mathbb{R}} \left\| \frac{\partial \gamma_u'(s)}{\partial u} \right\|^2 \cdot \left\| \gamma_u'(s) \right\| ds \right) du \qquad (10)$$

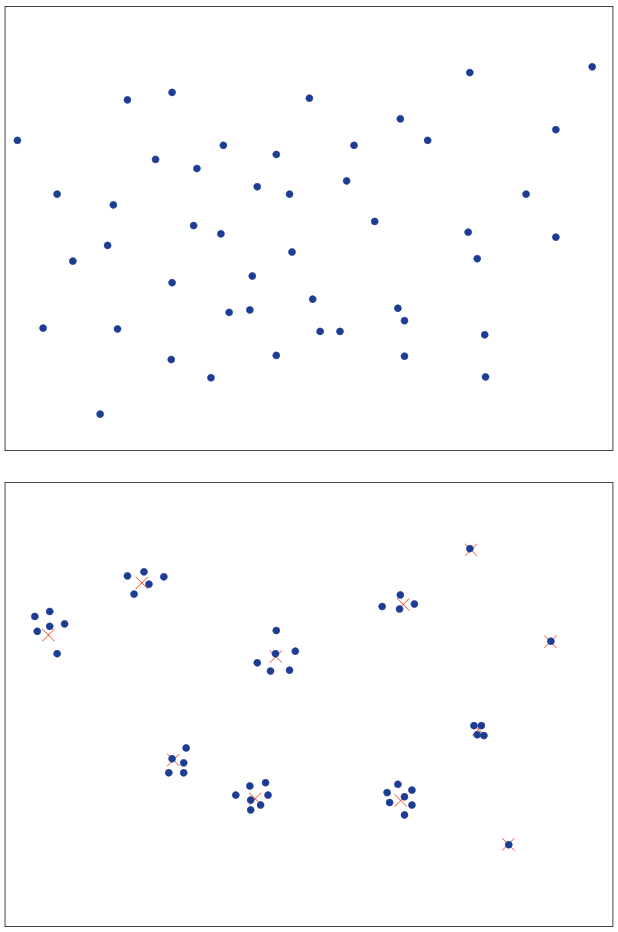


Figure 9.
Trajectory clustering algorithm representation. (a) On this metric space, each trajectory is represented by a point (blue point). (b) In this example, the algorithm finds eleven clusters with different features.

where $\frac{\partial v_u}{\partial u}$ is the normal vector (see Figure 8a).

In order to compute such energy, a grid on the homotopy connecting γ_1 to γ_2 is built, as shown on Figure 8b.

The optimization algorithm searches for the z coordinate of each grid point in order to minimize $E(\Phi)$. One can show that such problem is convex (from the optimization theory point of view) and a gradient-like method can be used to find the associated minimum (quadratic programming has been used to solve this problem efficiently).

This distance can now be used in any distance-based clustering algorithm.

CLUSTERING ALGORITHM

We consider a set of trajectories computed by the Bellman algorithm for each wind sample map. Having defined a distance between trajectories, one can gather together such trajectories in order to create clusters by using an adaptive clustering algorithm (hierarchical clustering). Such a clustering algorithm aims to partition the trajectory set into K clusters. To reach this goal, trajectories are considered as points in the associated metric space (see Figure 9a).

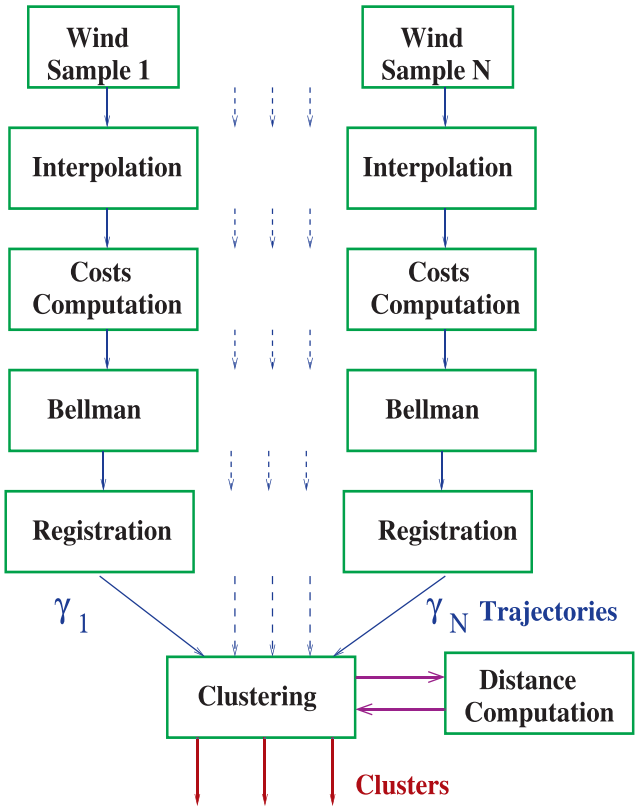


Figure 10.
Overall structure of the algorithm.

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.
- 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, create a new cluster containing the single trajectory.

The number of clusters is also a result of the algorithm. An example of clustering result is given on Figure 9b.

For each cluster c, one can compute also the following features:

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

The overall processing on the trajectory clustering algorithm is summarized by the Figure 10.

RESULTS

First, we have to consider two wind samples over the Atlantic Ocean from two different days (January 09, 2016 and February 14, 2016). The first sample has few wind dispersions compared to the second one. An example of such map is given on Figure 11.

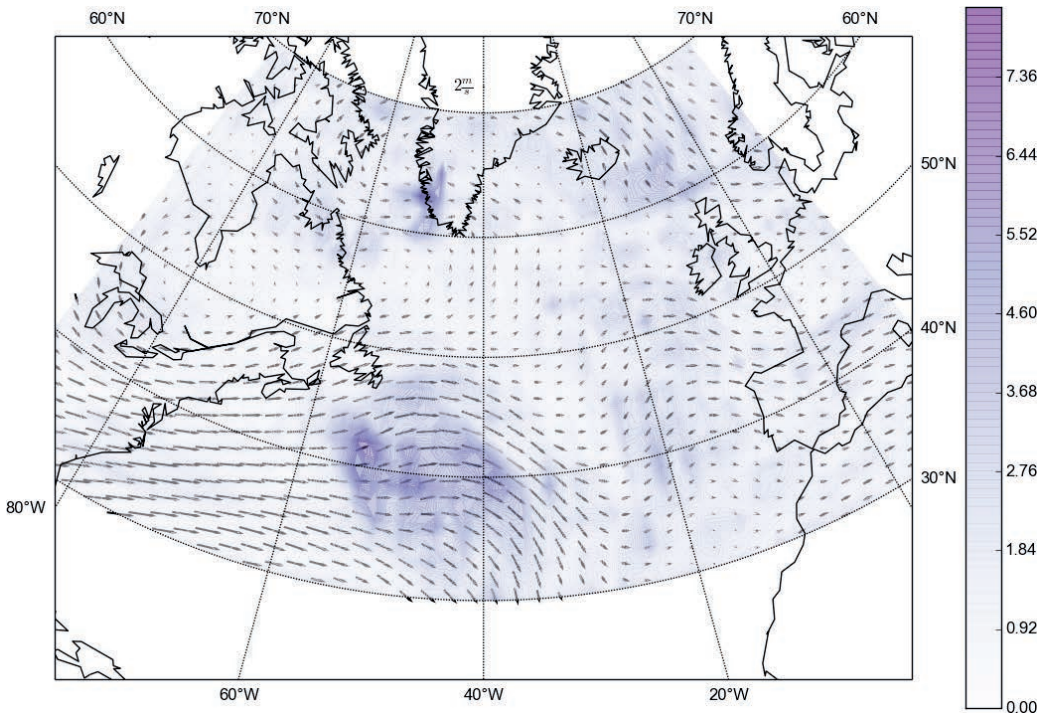


Figure 11.
Example of wind distribution over the Atlantic Ocean.

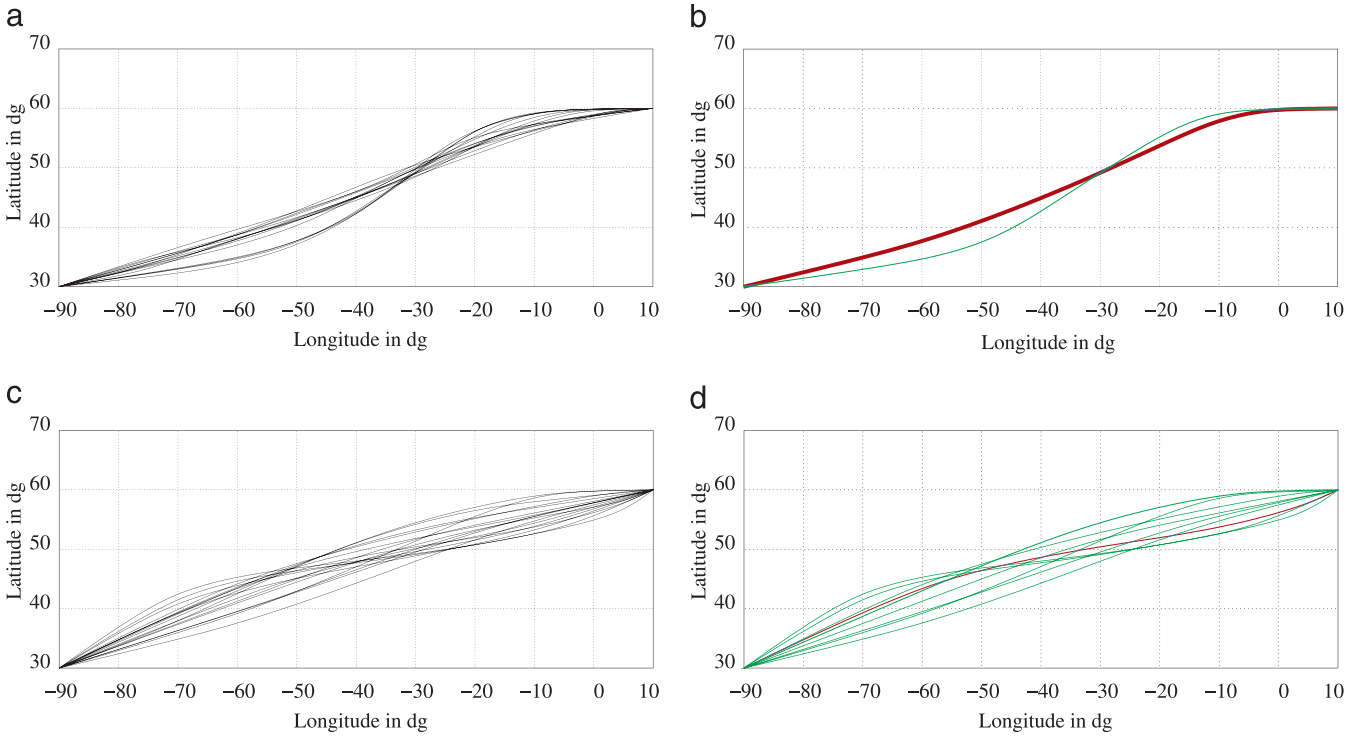


Figure 12.
(a) Wind optimal trajectories for the first wind sample set (January 09, 2016). Each black curve corresponds to an optimal trajectory computed by the Bellman algorithm for each wind sample. (b) Clusters produced for the first wind sample set. Two clusters have been extracted for which the most representative is represented in red. This cluster has 28 representatives and the other one has only six. This red cluster is then considered as the most robust trajectory. (c) Wind optimal trajectories for the second wind sample set (February 14, 2016). In this set, it can be noticed that optimal trajectories produced by the Bellman algorithm present more dispersion. (d) Clusters produced for the second wind sample set. In this case, due to trajectories dispersion, nine clusters have been identified with near the same number of representatives (between one and three). The cluster with three representatives (in red) may not be considered as robust and in this situation, it is very difficult to select one cluster among the others.

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

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): ($\lambda = 30$, $\theta = -90$) and the destination at $\vec{P}_d$(lat, long): ($\lambda = 60$, $\theta = 10$). The first data sample presents less dispersion and may result in more robust planning. 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 12a.

Those trajectories have been clustered by using the new distance that has been developed in the previous section. Two clusters have been extracted as it can be seen on Figure 12b. The first cluster has 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 to cluster 1 (Flight time 11h 34′; to compute this flight time, a True Air Speed of 450 kts has been considered). The best trajectory in cluster 2 has a flight time of 11h 47′.

The second wind sample data from February 14, 2016 is more critical with more dispersion in the wind data and the trajectories produced by the Bellman algorithm are also more spread (see Figure 12c).

Those trajectories have been also clustered and the associated clusters are represented on Figure 12d. In this case, nine clusters have been extracted with a maximum of five representatives. In this case, there is not a big difference between clusters in terms of representative number, and we can say that the associated robustness is the same.

CONCLUSION

This article 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. Those algorithms have been successfully applied to real wind data in order to identify robust wind optimal trajectories.

As a future work, we will compare the associated performances of trajectories in terms of flight time and fuel consumption. ◈

REFERENCES

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

[2] Arain, B., and Kendoul, F. Real-time wind speed estimation and compensation for improved flight. *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 50, 2 (2014), 1599–1606.

[3] Brezoescu, A., Castillo, P., and Lozano, R. Wind estimation for accurate airplane path following applications. In *Proceedings of the International Conference on Unmanned Aircraft Systems (ICUAS)*, 2013.

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

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

[6] Cho, A., Kim, J., Lee, S., and Kee, C. Wind estimation and airspeed calibration using a UAV with a single-antenna GPS receiver and pitot tube. *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 47, 1 (2011), 109–117.

[7] Shepard, D. A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the ACM Conference,* 1968.

[8] Delahaye, D., and Puechmorel, S. Tas and wind estimation from radar data. In *Proceedings of the IEEE/AIAA DASC*, 2009.

[9] Delahaye, D., Puechmorel, S., and Vacher, P. Windfield estimation by radar track Kalman filtering and vector spline extrapolation. In *Proceedings of the IEEE/AIAA DASC*, 2003.

[10] Dijkstra, E. A note on two problems in connexion with graph. *Numerische Mathematik*, Vol. 1, (1959), 269–271.

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

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

[13] Farin, G. *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*, Vol. 3. Cambridge University Press, 1999.

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

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

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

[17] Lefas, C. C. Real-time wind estimation and tracking with transponder downlinked airspeed and heading data. *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 23, 2 (1987), 169–174.

[18] Liao, T. W. Clustering of time series data—A survey. *Pattern Recognition*, Vol. 38, (2005), 1857–1874.

[19] Needham, C., and Boyle, R. Performance evaluation metrics and statistics for positional tracker evaluation. In *Proceedings of the ICVS,* Graz, Austria, Apr. 2003, 278–289.

[20] 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*, Vol. 4, 2 (1968), 100–107.

[21] Puechmorel, S. Geometry of curves with application to aircraft trajectories analysis. *Annales de la Faculté des Sciences de Toulouse, Mathématiques, Série 6*, Jul. 2015.

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

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

[24] Rani, S., and Sikka, G. Recent techniques of clustering of time series data: A survey. *International Journal of Computer Applications*, Vol. 52, 15 (Aug. 2012), 1–9.

[25] Weber, M. E., and Stone, M. L. Low altitude wind shear detection using airport surveillance radars. *IEEE Aerospace and Electronic Systems Magazine*, Vol. 10, 6 (1995), 3–9.

[26] Xiang, X., Wang, Z., Mo, Z., Chen, G., Blasch, E., and Pham, K. Wind field estimation through autonomous quadcopter avionics. In *Proceedings of the IEEE/AIAA DASC*, 2016.