

Identifying Orthodromy and Loxodromy Segments in ADS-B Aircraft Trajectory Data

Richard Alligier,^{*,1,2} Kim Gaume,^{1,3} David Gianazza,^{1,2} Xavier Olive,^{1,3} and Nicolas Durand^{1,2}

¹Fédération ENAC ISAE-SUPAERO ONERA, Université de Toulouse, France

²ENAC, Université de Toulouse, France

³ONERA DTIS, Université de Toulouse, France

*Corresponding author: richard.alligier@enac.fr

(Received: 1 Nov 2025; Revised: 6 Feb 2026; Accepted: 10 Feb 2026; Published: 17 Feb 2026)

(Editor: Junzi Sun; Reviewers: Ramon Dalmau and Yutong Chen)

Abstract

Aircraft trajectories can often be interpreted in terms of two fundamental path types: orthodromies (great-circle routes, representing the shortest path on the globe) and loxodromies (constant track-angle paths). While great circles typically define the reality of intercontinental flight planning, operational constraints, such as waypoint routing and air traffic control instructions, mean that actual trajectories are composed of multiple segments, not always conforming neatly to one type or the other. Recognizing which segments correspond to orthodromies or loxodromies can provide semantic meaning to flight data, transforming raw positional information into interpretable patterns of pilot intent or controller intervention.

We present a method to automatically identify these trajectory segments without any prior knowledge about waypoints. The approach leverages map projections: with the Mercator projection, by construction, loxodromies appear as straight lines, while orthodromies do so in gnomonic projections. By detecting straight-line portions in these transformed trajectories, we can classify corresponding segments of the original flight path. This may enable the identification of even short segments, offering insights into flight planning and deconfliction manoeuvres—for example, detecting when an aircraft temporarily deviates from its planned orthodromic route under controller instruction. Using an open trajectories dataset where 13 % of the total duration is labeled as deconfliction maneuvers, the identified loxodromies have a duration of 2 % of the total duration while 39 % of these 2 % can be matched with deconfliction maneuvers, higher than the 13 % prior.

Keywords: aircraft trajectory labelling; deconfliction extraction; ADS-B data

1. Introduction

Automatic Dependent Surveillance-Broadcast (ADS-B) is basically a system where aircraft broadcast their positions approximately each second, and can be received by ground stations and other aircraft equipped with ADS-B receivers. The development of crowd-sourced networks of receivers such as the OpenSky Network [1] has popularized the analysis of ADS-B trajectory among the research communities and even a larger audience. Post-analysis of ADS-B trajectory data often raises a number of questions about what occurs to a flight along its trajectory. Typical questions that arise are: Is the aircraft climbing, descending, or flying at a constant flight level? Is it turning, or flying straight towards a navaid? Is it following its flight plan, or is it deviated by an air traffic controller

so as to avoid a meteorological event or another aircraft? Answers to these questions are not readily available in raw, unlabelled ADS-B data.

Previous works have studied how to detect flight phases [2, 3, 4], turns [5], holding patterns [6], deconfliction actions [7], and more patterns [8] in ADS-B trajectories. Using additional data from the AIP (Aeronautical Information Publication), [9] identifies the route being followed by the aircraft. In previous works [10, 11], lateral deviations were extracted from ADS-B and flight plan data in order to detect deconfliction events. These two papers rely on the method `aligned_on_navpoint` in the traffic library, which requires flight plan data or at least a set of existing beacons/navaids.

In this paper, we propose a method to identify orthodromy and loxodromy segments in ADS-B trajectories without any use of flight plan data nor beacons/navaids data. When applicable, this approach can detect when an aircraft follows a fixed heading (loxodromy) or when it follows a geodesic route (orthodromy) towards a navaid, without the help of any additional flight plan data.

The rest of this paper is organized as follows. Section 2 describes the proposed method, and gives an illustration on a well-chosen example. In section 3, we describe the dataset used for a larger experiment, as well as a baseline method and the metrics used for comparison. Section 4 provides some results, and section 5 concludes the paper.

2. Method

Our method relies on two projections of the latitude and longitude positional data in an (x, y) space: the gnomonic projection that transforms great circles into straight lines in the projected space, and the Mercator projection, in which any course at constant bearing is a straight segment. Due to the properties of our two projections, trajectory segments exhibiting a constant track angle in a gnomonic (resp. Mercator) projection are likely orthodromic (resp. loxodromic) segments. The final decision is based on the maximum distances between the actual trajectory, the orthodromy, and the loxodromy.

2.1 Identifying constant segments

After projection (gnomonic or Mercator), each considered trajectory is smoothed and the track angle is computed for each point. The trajectory segments having a constant track angle (see Figure 1) are then extracted by applying two successive filters to the smoothed trajectory.

The first filter uses a maximum error threshold `max_error_thresh` to detect time intervals during which the track angle remains approximately constant. This filter selects all the segments (i, j) such that $\max_{i \leq k \leq j} \text{angles}[k] - \min_{i \leq l \leq j} \text{angles}[l]$ is inferior to `max_error_thresh`. Only the maximal valid segments are considered *i.e.* valid segments that are not included in any other valid segment. In this paper, the function generating such segments is named `ConstantSegments(angles, max_error_thresh)`. It uses a sliding window and its worst case complexity is $O(n)$ where n is the number of points inside the array `angles` containing the successive track angles of the trajectory. The `max_error_thresh` threshold in this filter is supposed to account for small variations around a constant track angle value. However, in some cases, this filter might select trajectory segments where the track angle is slowly increasing (or decreasing) while staying within the bounds defined by this threshold. Also, even for segments that are truly of constant track angle, there may remain a few points at the extremities that actually belong to the end or the beginning of a turn.

In order to filter out all these cases, we fit a linear model to the track angle measurements in each temporal segment selected by the first filter, and then consider both the deviation from the linear model at the extremities of the segment and the slope of the fitted line. The points at the extremities of the segment are recursively removed when the gap with the linear model is beyond

a threshold *thresh_border*. This recursive process stops when both extremities have values closer than *thresh_border* from the linear model. The resulting segment (if any) is then identified as being of constant track angle if the slope of the fitted line is strictly smaller than a chosen threshold *thresh_slope*.

The function *IsConstant* implementing this second filter is described at the beginning of Algorithm 1.

The above two-step filtering process might end up with many segments (i, j) satisfying the various threshold conditions. The filtering procedure prevents from selecting segments included in other, larger segments. However, it does not prevent from extracting overlapping segments.

In order to extract the largest possible non-overlapping segments, we prioritize segments by their lengths and prune overlapping segments according to an Intersection over Union (IoU) threshold. This Intersection over Union metric is described in SubSection 3.3.2. The algorithm first sorts all segments, then iteratively selects the longest segment and discards others that overlap beyond the allowed threshold *thresh_iou*. This ensures a non-redundant set of segments.

The Table 1 summarizes the thresholds and their values of the presented method in this paper, see Algorithm 1 and Algorithm 2. They were found using a grid search to maximize a matching criteria as described in Sub-Section 4.1.1.

The extraction process described above is summarized in the pseudo-codes Algorithm 1 and Algorithm 2.

Table 1. Summarizes the thresholds and their values of the method identifying constant angle segments, see Algorithm 1 and Algorithm 2.

threshold parameter	<i>max_error_thresh</i> [°]	<i>thresh_border</i> [°]	<i>thresh_slope</i> [°/s]	<i>thresh_iou</i> [-]
value	0.5	0.1	0.001	0.1

Algorithm 2 Sort by length and filter by IoU

```

1: function SORTBYLENGTHTHENFILTERBYIOU(L, thresh_iou)
2:   L ← sort(L, by = interval length)
3:   R ← [] ▷ Initialize result set
4:   while L ≠ ∅ do
5:     x ← pop_last(L) ▷ Take best element
6:     append x to R
7:     L ← [y ∈ L | IoU(x.interval, y.interval) ≤ thresh_iou, ]
8:   end while
9:   return R
10: end function

```

2.2 Discriminating between an Orthodromy and a Loxodromy

Before discussing the criteria for discriminating between orthodromy and loxodromy, let us remind that there are cases when this is impossible. When an aircraft flies towards the North or towards the South, it follows both an orthodromy (a meridian) and a loxodromy (constant track angle 0 or 180 degrees) and there is no way to assign the trajectory exclusively to one category. The same problem occurs for flights following the equator.

However, in most cases, the orthodromy and loxodromy on the globe are distinct trajectories. Likewise, the farther from the equator and the more in a general East-West or West-East direction, the more distinct they are. Finally, the longer the trajectory segment, the easier it is to discriminate

Algorithm 1 Extract constant angle segments

```

1: function ISCONSTANT( $i, j, t, angles, thresh\_border, thresh\_slope$ )
2:    $linear\_model \leftarrow FitLinearModelonMedian(t, angles)$   $\triangleright$  a linear model fitting the median, obtained with an absolute
   value loss
3:    $newi \leftarrow$  increase  $i$  till  $|angles[i] - linear\_model.predict(t[i])| < thresh\_border$ 
4:    $newj \leftarrow$  decrease  $j$  till  $|angles[j] - linear\_model.predict(t[j])| < thresh\_border$ 
5:   if  $newi == i$  and  $newj == j$  then
6:     if  $abs(linear\_model.slope) < thresh\_slope$  then
7:       return ( $i, j$ )
8:     else
9:       return None
10:    end if
11:  else
12:    return IsConstant( $newi, newj, t, angles, thresh\_border, thresh\_slope$ )
13:  end if
14: end function
15: function EXTRACTCONSTANTAFTERPROJECTION( $proj, t, lats, lons, max\_error\_thresh, thresh\_border, thresh\_slope, thresh\_iou$ )
16:    $xs, ys \leftarrow Projection(proj, lats, lons)$   $\triangleright$  apply projection, gnomonic or Mercator
17:    $spline \leftarrow CubicSmoothingSpline(t, xs, ys)$   $\triangleright$  fit a spline on position measurements
18:    $dx\_dt, dy\_dt \leftarrow EvaluateFirstDerivative(spline, t)$   $\triangleright$  compute derivative using the fitted spline
19:    $angles \leftarrow arctan2(dy\_dt, dx\_dt)$   $\triangleright$  compute angles using the speed along the x and y axis
20:    $res \leftarrow empty$ 
21:   for ( $i, j$ ) in ConstantSegments( $angles, max\_error\_thresh$ ) do
22:      $seg \leftarrow IsConstant(i, j, t, angles, thresh\_border, thresh\_slope)$   $\triangleright$  refine and decide whether the segment is constant
   or not
23:     if  $seg$  is not None then
24:       add  $seg$  to  $res$ 
25:     end if
26:   end for
27:    $res \leftarrow SortByLengthThenFilterByIoU(res, thresh\_iou)$   $\triangleright$  merge segments that are close to each other in terms of
   intersection over union metrics, considering the indices ( $i, j$ ) as intervals to be compared
28:   return  $res$ 
29: end function

```

between orthodromy and loxodromy. To account for these limitations and identify when we cannot discriminate between orthodromy and loxodromy, the maximum distance between the two curves is computed. If this distance is large enough (larger than a threshold thresh_loxo_ortho), the curves are deemed discriminable. When this is the case, we still need to evaluate how close the actual trajectory is to the two curves in order to assign it to the corresponding category.

Formally, a trajectory segment is confirmed as an orthodromy if $\text{dist}(\text{traj}, \text{ortho}) < r \times \text{dist}(\text{loxo}, \text{traj})$ and $\text{dist}(\text{traj}, \text{ortho}) < r \times \text{dist}(\text{loxo}, \text{ortho})$, and $\text{dist}(\text{loxo}, \text{ortho}) > \text{thresh_loxo_ortho}$, where r is a factor smaller than 1 that guarantees that the trajectory is sufficiently close from one of the two curves while being far enough from the other. The logic is similar for the loxodromy. For the example in Figure 1, and all the results in this paper, we chose $r = 0.5$ and $\text{thresh_loxo_ortho} = 30 \text{ m}$. The obtained functions `IsOrthodromyOnly` and `IsLoxodromyOnly` are described in Algorithm 3.

Algorithm 3 Algorithms to discriminate between loxodromy and orthodromy

```

1: function ISORHTODROMYONLY(segment, traj_adsb,  $r = 0.5$ ,  $\text{thresh\_loxo\_ortho} = 30 \text{ m}$ )
2:   (i, j)  $\leftarrow$  segment                                      $\triangleright$  indexes of the starting and ending point
3:   loxo  $\leftarrow$  generating loxodromy connecting point i to point j of traj_adsb
4:   ortho  $\leftarrow$  generating orthodromy connecting point i to point j of traj_adsb
5:   return  $\text{max\_dist}(\text{traj\_adsb}, \text{ortho}) < r \cdot \text{max\_dist}(\text{traj\_adsb}, \text{loxo})$ 
         $\wedge \text{max\_dist}(\text{traj\_adsb}, \text{ortho}) < r \cdot \text{max\_dist}(\text{ortho}, \text{loxo})$ 
         $\wedge \text{max\_dist}(\text{ortho}, \text{loxo}) > \text{thresh\_loxo\_ortho}$ 
6: end function
7: function ISLOXODROMYONLY(segment, traj_adsb,  $r = 0.5$ ,  $\text{thresh\_loxo\_ortho} = 30 \text{ m}$ )
8:   (i, j)  $\leftarrow$  segment                                      $\triangleright$  indexes of the starting and ending point
9:   loxo  $\leftarrow$  generating loxodromy connecting point i to point j of traj_adsb
10:  ortho  $\leftarrow$  generating orthodromy connecting point i to point j of traj_adsb
11:  return  $\text{max\_dist}(\text{traj\_adsb}, \text{loxo}) < r \cdot \text{max\_dist}(\text{traj\_adsb}, \text{ortho})$ 
         $\wedge \text{max\_dist}(\text{traj\_adsb}, \text{loxo}) < r \cdot \text{max\_dist}(\text{ortho}, \text{loxo})$ 
         $\wedge \text{max\_dist}(\text{ortho}, \text{loxo}) > \text{thresh\_loxo\_ortho}$ 
12: end function

```

2.3 Illustration of the method on one example

Figures 1 and 2 illustrate the method on a selected example. Table 2 shows the pairwise maximum distances between the actual, loxodromic, and orthodromic trajectories for the segments identified in Figure 1. The segment just before the loxodromy segment was discarded and does not appear in Table 2 because it did not meet our distance criteria. The loxodromy is most likely a "turn left" deconfliction manoeuvre instructed by an air traffic controller, with a heading that resulted in a constant 80° track angle (see Figure 2).

Table 2. Flight duration, and pairwise maximum distances between loxodromy, orthodromy and actual ADS-B trajectory, for each identified segment.

identified	segment number	duration [s]	ortho-loxo [m]	adsb-loxo [m]	ortho-adsb [m]
loxodromy	1	157	34.70	13.72	42.63
orthodromy	1	304	114.82	107.64	13.68
orthodromy	2	507	375.17	383.71	31.60
orthodromy	4	387	207.00	199.88	27.34

On this example, we see that it is possible to identify and discriminate loxodromy and orthodromy segments that are fairly short (a few minutes in Table 2). A more comprehensive study is required to explore the limits of this method regarding the possibility to isolate constant track angle segments in noisy data, and the ability to discriminate between loxodromy and orthodromy for trajectories close to a meridian direction.

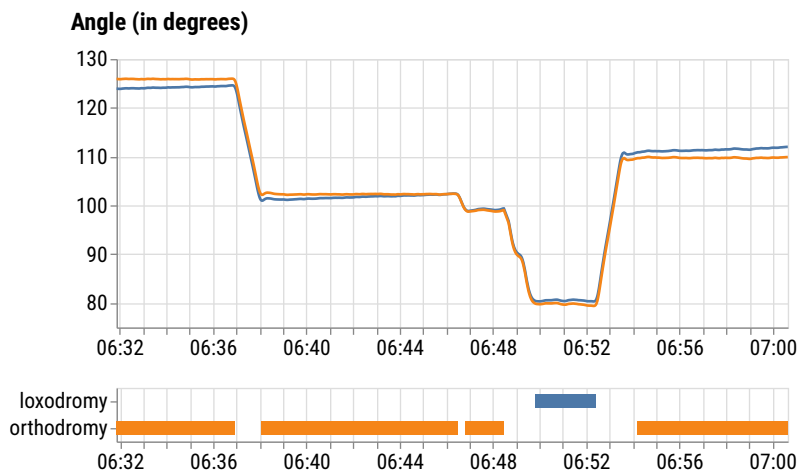


Figure 1. Illustration on how the method works on a well-selected example.

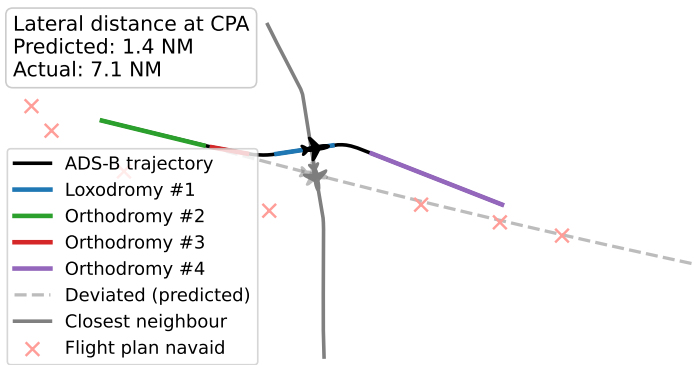


Figure 2. Loxodromy is likely a deconfliction manoeuvre

3. Experiment setup

To our knowledge, there is no dataset that contains the ground truth on whether the aircraft flies an orthodromy or a loxodromy. However, we can compare the method presented in this paper with a baseline method of the Python traffic library, on a dataset of traffic samples. The baseline used for comparison is the method `aligned_on_navpoint` of the library `traffic`. It identifies orthodromic segments only, by testing the alignment to the orthodromic routes leading to candidate nav aids. This method requires knowing the set of nav aids that the flight might align to.

3.1 Data set

In order to have a good chance to actually observe both orthodromic and loxodromic segments in aircraft trajectories that are usually made mostly of orthodromic segments, we shall consider traffic samples where lateral deconfliction actions were taken.

Many of the deconfliction instructions issued to pilots by air traffic controllers are "turn left or right" lateral maneuvers enforcing a relative heading change (e.g. "turn left 10 degrees") or an absolute one (e.g. "turn left, heading 260 degrees). The new trajectory resulting from such a heading change is a loxodromy.

The data set used in this section is the catalogue of deconfliction cases extracted in [12], using a data-driven heuristic method. This dataset contains traffic samples where some flights are tagged by the heuristic as being deviated due to a conflict resolution. Each sample contains the deviated aircraft and the surrounding traffic, as well as additional information concerning the ATC sector geometry and the flight plans of the involved aircraft. These flight plans are the last ones filled before flight, and they contain the waypoints of the planned route of each aircraft. In the following results, these waypoints are used as candidates for alignment by the baseline method `aligned_on_navpoint` to detect orthodromic trajectory segments.

The dataset of deconfliction cases in [12] considers only lateral deviations occurring at a constant altitude, above 20,000 *ft* and that last at least 30 s. To be able to compare our method to the baseline on the maneuvered aircraft identified in this dataset, a similar filter is applied to the trajectory segments identified by both methods. We select from the dataset of deconfliction cases a subset from the AIRAC cycle 2207, with 3,816 deviated trajectories where the lateral deviation lasts at least 120 s.

3.2 Baseline method

The method `aligned_on_navpoint` of the Python library `traffic` takes a trajectory as input and returns the list of trajectory segments that are aligned to a waypoint of the flight plan. An aircraft is said to be aligned to a given waypoint if its track angle is close to the one required to follow an orthodromic (great-circle) trajectory toward this waypoint. A threshold of 0.5° is used for our experiments. This threshold is consistent with the 1° threshold for `max_error_thresh` of the method presented in the paper. Two segments may have a non-empty intersection: at a given time, the aircraft may be aligned on two waypoints simultaneously. To avoid this, we apply the method to the return list of segments.

3.3 Metrics used to compare our method with the baseline

The output of both the baseline and the proposed method are sets of temporal segments. The baseline produces segments tagged as orthodromy, whereas our method generates two sets: one tagged as orthodromy and the other as loxodromy. Comparing two sets of interval segments is not straightforward. We use a metric used in object detection in Machine Learning: the intersection over Union (IoU), it looks for exact match between the predicted segment and the ground truth segments. In Section 4.3, we compare the predicted segments to deconfliction segments. Each deconfliction segment

may contain several heading change, and hence several loxodromy/orthodromy. As a consequence when evaluating the predicted segments we look for inclusion in the deconfliction segments, leading us to use the Intersection over Length (IoL) metric.

3.3.1 Intersection over Union (IoU)

A common metric used to evaluate the performance of object detection and segmentation models is the *Intersection over Union* (IoU). Given a predicted region \hat{B} and the corresponding ground-truth region B , the IoU is defined as the ratio between the area of overlap and the area of union:

$$\text{IoU}(B, \hat{B}) = \frac{|B \cap \hat{B}|}{|B \cup \hat{B}|}, \quad (1)$$

where $|B \cap \hat{B}|$ denotes the area of intersection between the predicted and ground-truth regions, and $|B \cup \hat{B}|$ denotes the area of their union. The IoU value ranges from 0 to 1, where $\text{IoU} = 1$ indicates perfect alignment and $\text{IoU} = 0$ indicates no overlap.

In practice, a prediction is often considered a correct detection if its IoU with the ground truth exceeds a predefined threshold (e.g., 0.5 or 0.75). This makes IoU a crucial evaluation criterion in benchmarks for object detection and semantic segmentation.

In order to quantify how well a specific trajectory segment A identified by a method matches one of the segments in the set IntervalSet returned by the other method, we define the following metric:

$$\text{MaxIoU}(A, \text{IntervalSet}) = \max_{X \in \text{IntervalSet}} \text{IoU}(A, X), \quad (2)$$

Note that the roles of the sets being compared with the MaxIoU metric are not symmetric. In Eq. 2, the set IntervalSet is considered as the ground truth and we are trying to assess how close a segment A predicted by another method is from this ground truth. In the following of this paper, when comparing two methods X and Y (our method and the baseline, typically), we can assume either one or the other to be the ground truth. We can plot the distribution of $\{\text{MaxIoU}(x, S_Y) \mid x \in S_X\}$ to assess how the individual segments of S_X returned by method X match the segments S_Y found by method Y . In this case S_Y is considered as the ground truth. We can also plot the distribution of $\{\text{MaxIoU}(y, S_X) \mid y \in S_Y\}$ if we consider S_X as the ground truth. In the following we shall plot both distributions (see Figure 6 for example).

3.3.2 Metric quantifying whether the segments are included in another segments set

The metrics based on the the intersection over union (IoU) are not well suited to test if a segment A is included in another segment B . To quantify the inclusion, we define the Intersection over Length (IoL) metric:

$$\text{IoL}(A, B) = \frac{|A \cap B|}{|A|}, \quad (3)$$

The $\text{IoL}(A, B)$ metric quantifies the proportion of A included in B .

Finally, we define the following metric $\text{MaxIoL}(A, \text{IntervalSet})$ that quantifies the maximum inclusion of a segment A in any of the segments of set of intervals IntervalSet

$$\text{MaxIoL}(A, \text{IntervalSet}) = \max_{X \in \text{IntervalSet}} \text{IoL}(A, X), \quad (4)$$

4. Results

Let us denote respectively `SegmentsProjOrtho` and `SegmentsProjLoxo` the sets of orthodromic and loxodromic segments identified by our method, and `SegmentsBaseline` the set of orthodromic segments identified by the baseline method. In this section, we study segments with a duration superior to 30 s.

4.1 Results on the identified orthodromies

Let us first compare our method with the baseline on the detection of orthodromies only.

4.1.1 Parameters tuning

The algorithm presented in this paper has four parameters to be tuned described in Section 2.1 (*max_error_thresh*, *thresh_border*, *thresh_slope* and *thresh_iou*). As stated in Sub-Section 3.2, choosing *max_error_thresh* = 0.5° is consistent with the parameters of the baseline algorithm. This leaves us with only three parameters to tune. To tune these parameters, the orthodromies set `SegmentsBaseline` given by the baseline algorithm, described in Sub-Section 3.2, will be used as ground truth. To quantify how much a set of segments `IntervalSet1` is included in an other `IntervalSet2`, we define the metric below:

$$\text{IsIncluded}(\text{IntervalSet1}, \text{IntervalSet2}) = \frac{\text{Average } \text{MaxIoU}(X, \text{IntervalSet2})}{X \in \text{IntervalSet1}}$$

The `IsIncluded`(`IntervalSet1`, `IntervalSet2`) value will be 1 if any segment in `IntervalSet1` can be found in `IntervalSet2`, whereas this value will be 0 if each segment in `IntervalSet1` has an empty intersection with all segments in `IntervalSet2`. As can be seen, this metric is asymmetric, and using a grid search, the selected parameters are the ones generating a set of orthodromies `SegmentsProjOrtho` that maximizes the criteria below:

$$\text{criteria}(\text{SegmentsProjOrtho}) = \min(\text{IsIncluded}(\text{SegmentsBaseline}, \text{SegmentsProjOrtho}), \text{IsIncluded}(\text{SegmentsProjOrtho}, \text{SegmentsBaseline})) \quad (5)$$

The baseline and our algorithm can both contains false positive and false negative. In order to avoid adjusting parameters over noise, we use only segments with a duration superior to 120s that are empirically more reliable segments. As can be seen in Figure 3, this criteria seems to be in local maxima when using *thresh_border* = 0.15°, *thresh_slope* = 0.0005°/s and *thresh_iou* = 0. These will be the values used in this study.

4.1.2 Descriptive statistics

The `SegmentsBaseline` set contains 10191 segments whereas `SegmentsProjOrtho` contains 8706 segments. Our method seems to detect more orthodromies than the baseline method. This might come from the fact that the baseline method considers only waypoints from the flight plan to test for alignment. Consequently the trajectory of an aircraft flying toward a navaid that is not in its flight plan will not be identified as an orthodromy by the baseline method.

The distribution of the maximum distance between the orthodromy and the trajectory on the identified segments is plotted on Figure 4. The maximum distance obtained with the method presented in this paper is smaller, suggesting a better segment delimitation than the baseline method.

The distribution of the duration of the orthodromies is plotted on Figure 5. The baseline method identifies more very short segments than the projection method developed in this paper. Conversely, the projection method identifies more segments in the 3-10 minutes range.

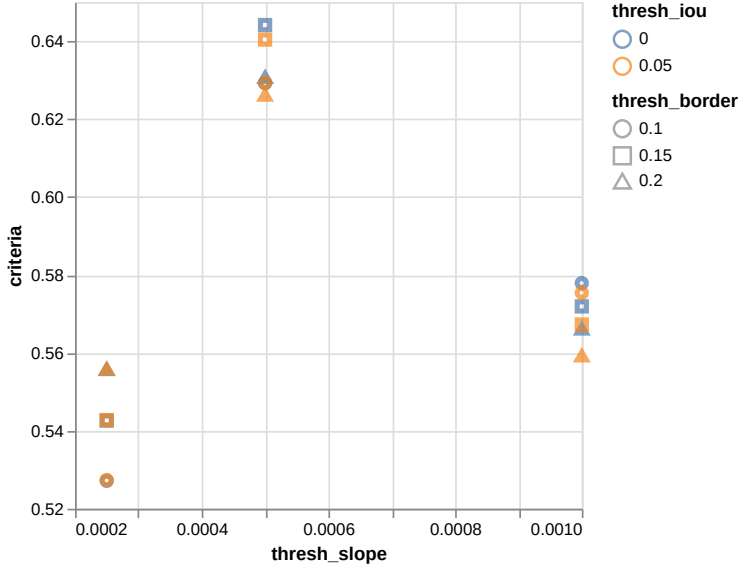


Figure 3. Plot of criteria (SegmentsProjOrtho) for different combination of values for the parameters *max_error_thresh*, *thresh_border*, *thresh_slope* and *thresh_iou*.

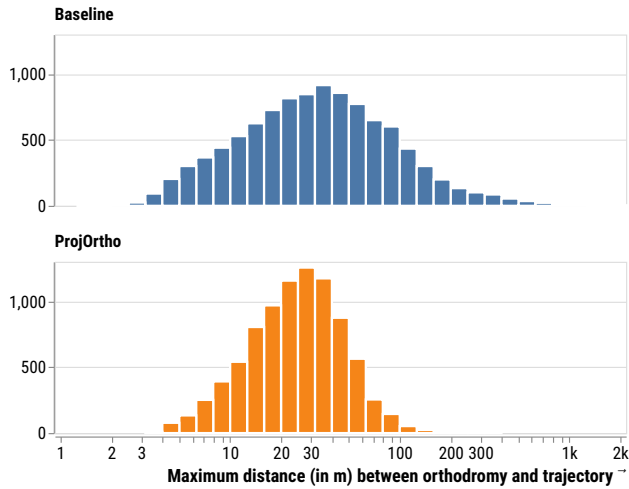


Figure 4. Comparison of the “quality” of the orthodromies identified by the baseline method and the segments SegmentsProjOrtho obtained with the method described in this paper.

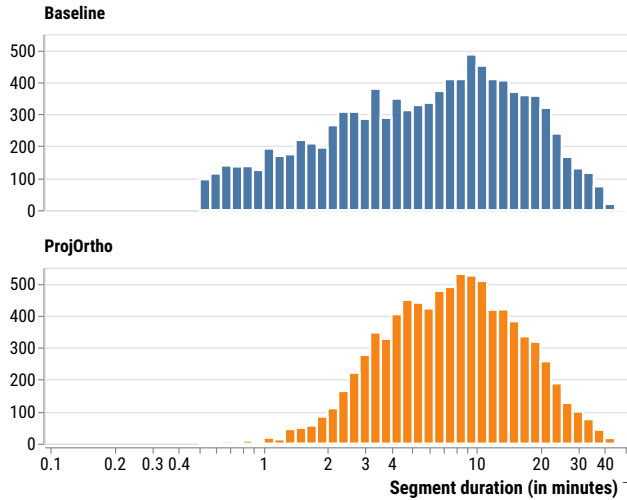


Figure 5. Comparison of the duration of the orthodromies identified by the baseline method and the segments SegmentsProjOrtho obtained with the method described in this paper.

4.1.3 Segments sets comparisons

Figure 6 plots the histograms of MaxIoU values (see section 3.3 for a definition of this metric) when comparing the two segments sets SegmentsBaseline and SegmentsProjOrtho obtained with the baseline method and our method, respectively. The histogram at the top of Fig. 6 shows how the individual segments obtained with the baseline method match the set of segments (here considered as the ground truth) obtained with our method. In the histogram at the bottom, the perspective is reversed (the baseline segments are considered as the ground truth) and the histogram shows how the segments obtained with our method match the set obtained with the baseline. In both histograms, the transparent bars show the MaxIoU values obtained when considering all segments in SegmentsBaseline and SegmentsProjOrtho, whereas the plain bars account only for the segments with a duration superior to 300 s.

Looking at the transparent bars in the histograms of Fig. 6, we can observe a peak at MaxIoU = 1 on the right of each histogram, indicating a number of approximately 3400 perfect matches between the predictions of both trajectories. However, we also observe a non-negligible peak at MaxIoU = 0 on the left, indicating that each method detects orthodromic segments that were not detected by the other. When looking at segments with a duration superior to 300 s (plain bars in Fig. 6) we can see that this discrepancy is drastically reduced, suggesting that both methods detect mostly the same orthodromic segments in that case.

4.2 Results on the discrimination between loxodromy and orthodromy

Let us now consider the set SegmentsProjLoxo of loxodromic trajectory segments found by our method. There is no baseline method to compare with, for the loxodromy detection. However, we can search for matches with SegmentsProjOrtho to see when trajectories are tagged both as loxodromies and orthodromies.

As stated in section 2.2, some overlaps are expected between the loxodromies in SegmentsProjLoxo and the orthodromies in SegmentsProjOrtho. Considering the latitudes at which our traffic samples are located (Bordeaux ATC center in France) these overlaps are mostly expected for aircraft flying either toward the North or the South, following a meridian.

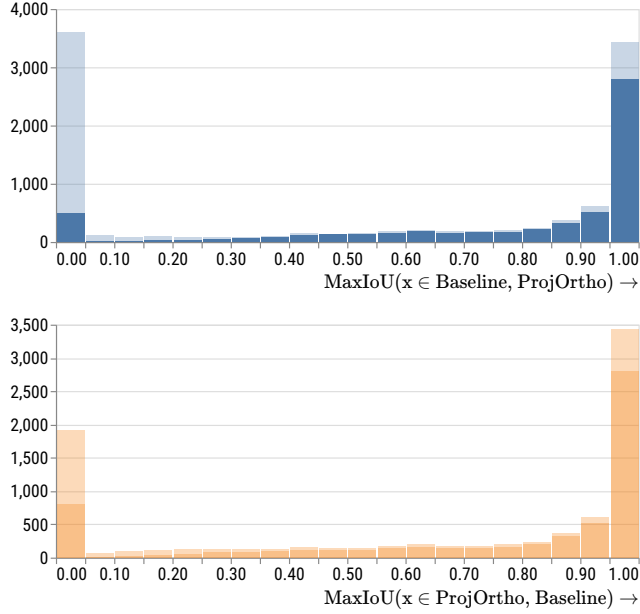


Figure 6. Histogram of the MaxIoU values obtained when matching the segments found by the baseline with the segments found by our method (top), and when matching the segments found by our method with the segments found by the baseline (bottom). Transparent bars account for all segments whereas plain bars account for segments with a duration above 300 s.

The proportion of theoretically indistinguishable orthodromy and loxodromy may vary according: 1) the latitude of the airspace, 2) the noisiness of the data, and 3) the track angle of the considered flight. In our experiments, the considered sector is located at a latitude of approximately 44° . This latitude is intermediate within the European airspace, being neither the closest to the equator nor the most northerly. The data used are real ADS-B data which are somewhat noisy data, we did not quantify this noise nor tried to lower or increase it as it is not a simple thing to do. Lastly, concerning the track angle, the Figure 7 plots the average MaxIoU at different track angles for SegmentsProjLoxo and SegmentsProjOrtho. This allows to quantify how much orthodromies overlaps with at least one loxodromy. This overlap is almost complete with a MaxIoU around 0.9 for track angles in the North-South part $\pm 10^\circ$. This overlap falls drastically to a MaxIoU below 0.3 when considering track angles in the East-West part $\pm 60^\circ$. The overlap is even lower when considering long segment with a duration superior to 300 s. All this suggests that in this East-West track angle interval, the identified orthodromy and loxodromy segments do not overlap, they are mostly distinct parts of the trajectories and hence orthodromies and loxodromies are distinguishable inside this track angle interval, in the considered sector's latitude.

As expected, many loxodromic segments oriented in a direction close to a meridian are also identified as orthodromic, and we have much less overlaps between loxodromies and orthodromies for aircraft flying East or West.

Applying the rules described in Algorithm 3 in SubSection 2.2, we obtain two other sets of segments noted SegmentsProjLoxoNotOrtho containing 741 segments and SegmentsProjOrthoNotLoxo containing 4822 segments. This rule successfully separates loxodromies and orthodromies when comparing these sets using the MaxIoU metric (Figure 8).

Average MaxIoU($x \in \text{ProjOrtho}, \text{ProjLoxo}$) per track angle

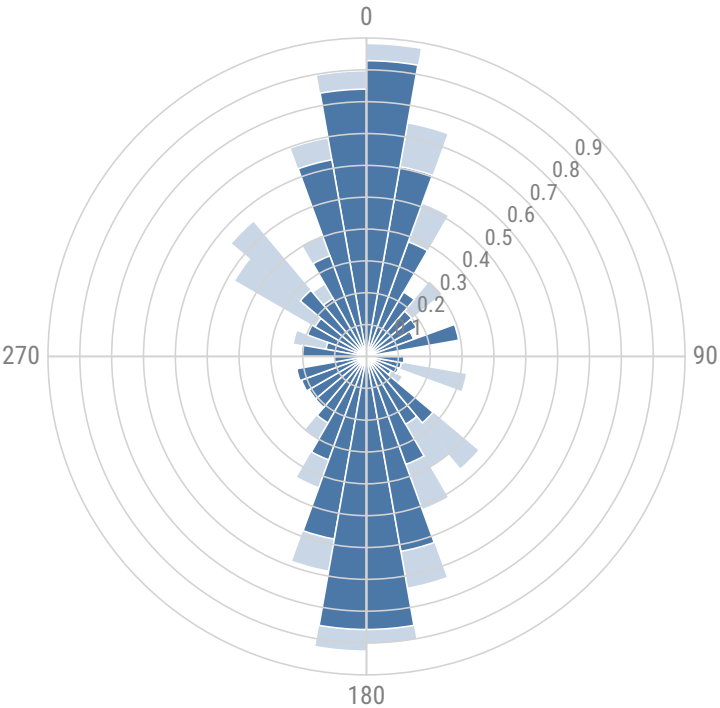


Figure 7. Average MaxIoU for different track angle when trying to match each segment in SegmentsProjLoxo with segments in SegmentsProjOrtho.

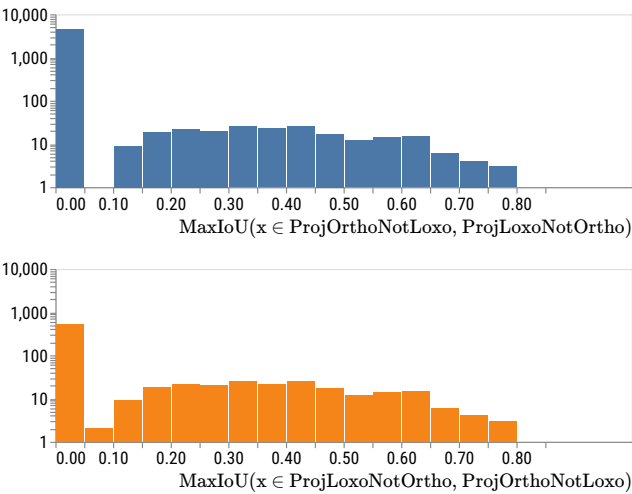


Figure 8. MaxIoU distributions when comparing segments in SegmentsProjLoxoNotOrtho with segments in SegmentsProjOrthoNotLoxo.

4.3 Results on whether the pure loxodromy can be matched with a deconfliction manoeuvrer

The dataset in [12] contains deviations that are tagged as deconfliction manoeuvrers by a data-driven heuristic method. Each lateral deconfliction manoeuvrer is associated with a temporal segment. Let us note `SegmentsDeconfliction` this set of segments. The duration of these conflict segments amounts to 15 days, 7:57:49 among the 114 days 12:53:34 total flights duration, so a ratio of 13 %.

To investigate whether the pure loxodromy in `SegmentsProjLoxoNotOrtho` can be matched with a deconfliction maneuver in `SegmentsDeconfliction`, we look at the `MaxIoL` metric. As the deconfliction temporal segment does not necessarily contain only a loxodromy, we prefer this inclusion metric to the `MaxIoU` metric that measures exact matching between segments. The `MaxIoL` metric value is 1 if the pure loxodromy is fully included in one deconfliction temporal segment, this value reduces if one part of the pure loxodromy is not included, it can reach the value 0 if the pure loxodromy have no intersection at all with any deconfliction temporal segments.

The bottom histogram in Figure 9 shows the `MaxIoL` values for the loxodromies, with a logarithmic scale, measuring their inclusion in the segments of the deconfliction maneuvers' set. The total duration of these pure loxodromies is 3 days, 2:48:33 , and the duration of these loxodromies inside the deconfliction maneuvers amounts to 1 day, 5:24:50 , so a ratio of 39 %, which is greater that the prior ratio 13 %. As a sanity check we have plotted a similar figure for the pure orthodromies (top histogram in Figure 9). The total duration of these pure orthodromies is 43 days, 23:14:47 , and the duration of these orthodromies inside the deconfliction maneuvers amounts to 22:58:24 , so a ratio of 2 %, which is way lower than the prior ratio 13 %. We can see that only a small proportion of the pure orthodromies is included in a deconfliction maneuver.

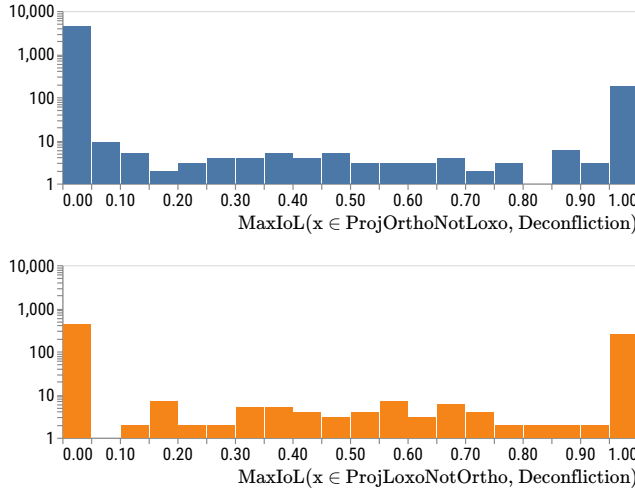


Figure 9. The distribution of the proportion included in segment in `SegmentsDeconfliction`.

5. Conclusion

This paper presented a geometric method for identifying orthodromic and loxodromic segments within ADS-B trajectory data, based solely on projection properties and without reliance on flight plan information. By combining gnomonic and Mercator projections with an algorithm for detecting constant track-angle segments, the approach provides a data-driven means to interpret the structure

of real-world flight trajectories.

The results demonstrate that orthodromic segments detected through this projection-based approach are consistent with those obtained from a baseline method relying on waypoint alignment, while the additional identification of loxodromic segments offers new insight into controller-induced heading maneuvers. Using an open trajectories dataset where 13 % of the total duration is labeled as deconfliction maneuvers, the identified pure loxodromies have a duration of 2 % of the total duration while 39 % of these 2 % can be matched with deconfliction maneuvers, higher than the 13 % prior. Contrastively, the pure orthodromies identified amounts to 38 % of total duration while only 2 % of these 38 % can be matched with deconfliction maneuvers.

Future work will investigate the robustness of the method under varying noise conditions, its integration with flight plan reconstruction techniques, and its application to large-scale datasets to better characterize navigation practices and air traffic management behavior across different airspaces. Overall, the proposed framework contributes to enriching the semantic interpretation of ADS-B data and paves the way for automated detection of tactical flight patterns.

Author contributions

Conceptualization (R.A), Methodology (R.A), Software (R.A, K.G, X.O), Validation (R.A, X.O, D.G), Formal analysis (*all*), Investigation (*all*), Data Curation (R.A, K.G), Writing – Original Draft (R.A, D.G), Writing – Review & Editing (D.G, X.O, N.D), Visualization (X.O)

Open data statement

All the data is available at a link that will be here eventually.

Reproducibility statement

All the figures and algorithms for this paper are available in the repository <https://github.com/richardalligier/identifyingortholoxo>.

References

- [1] Matthias Schäfer, Martin Strohmeier, Vincent Lenders, Ivan Martinovic, and Matthias Wilhelm. “Bringing up OpenSky: a large-scale ADS-B sensor network for research”. In: *Proceedings of the 13th International Symposium on Information Processing in Sensor Networks*. IPSN '14. Berlin, Germany: IEEE Press, 2014, pp. 83–94. ISBN: 9781479931460.
- [2] Junzi Sun, Joost Ellerbroek, and Jacco Hoekstra. “Large-scale flight phase identification from ADS-B data using machine learning methods”. In: *Proc. of the 7th International Conference on Research in Air Transportation*. June 2016.
- [3] Nataliia Kuzmenko, Ivan Ostroumov, Yurii Bezkorovainyi, Yuliya Averyanova, Vitalii Larin, Olha Sushchenko, Maksym Zaliskyi, and Oleksandr Solomentsev. “Airplane flight phase identification using maximum posterior probability method”. In: *Proc. of the 3rd International Conference on System Analysis & Intelligent Computing*. 2022. DOI: 10.1109/SAIC57818.2022.9922913.
- [4] Qilei Zhang, John H Mott, Mary E Johnson, and John A Springer. “Development of a reliable method for general aviation flight phase identification”. In: *IEEE Transactions on Intelligent Transportation Systems* 23.8 (2021). DOI: 10.1109/TITS.2021.3106774.

- [5] Junzi Sun, Joost Ellerbroek, and Jacco Hoekstra. “Reconstructing Aircraft Turn Manoeuvres for Trajectory Analyses Using ADS-B Data”. In: *Proc. of the 9th SESAR Innovation Days*. Athens, Greece, 2019.
- [6] Xavier Olive, Luis Basora, Junzi Sun, and Enrico Spinielli. “Training a Machine Learning Model to Detect Holding Patterns in Aircraft Trajectories”. In: *Journal of Open Aviation Science* 2.2 (2025). doi: [10.59490/joas.2024.7943](https://doi.org/10.59490/joas.2024.7943).
- [7] Xavier Olive and Luis Basora. “Detection and Identification of Significant Events in Historical Aircraft Trajectory Data”. In: *Transportation Research Part C: Emerging Technologies* 119 (2020). doi: [10.1016/j.trc.2020.102737](https://doi.org/10.1016/j.trc.2020.102737).
- [8] Xavier Olive, Junzi Sun, Adrien Lafage, and Luis Basora. “Detecting Events in Aircraft Trajectories: Rule-Based and Data-Driven Approaches”. In: *Proc. of the 8th OpenSky Symposium*. 2020. doi: [10.3390/proceedings2020059008](https://doi.org/10.3390/proceedings2020059008).
- [9] Hyeonwoong Lee and Hak-Tae Lee. “Extracting flight plans from recorded ADS-B trajectories”. In: *International Journal of Aeronautical and Space Sciences* 24.2 (2023). doi: [10.1007/s42405-022-00539-3](https://doi.org/10.1007/s42405-022-00539-3).
- [10] Kim Gaume, Xavier Olive, David Gianazza, Richard Alligier, and Nicolas Durand. “A Catalogue of Deconfliction Actions Extracted from Historical ADS-B Data”. In: *Journal of Open Aviation Science* 1.2 (Oct. 2023). doi: [10.59490/joas.2023.7222](https://doi.org/10.59490/joas.2023.7222).
- [11] Kim Gaume, Richard Alligier, Nicolas Durand, David Gianazza, and Xavier Olive. “Extracting Lateral Deconfliction Actions from Historical ADS-B Data with Median Regression”. In: *Proc. of the 11th International Conference for Research in Air Transportation*. Singapore, 2024.
- [12] Kim Gaume, Xavier Olive, David Gianazza, Richard Alligier, and Nicolas Durand. “Extracting aircraft conflict-resolution situations from historical ADS-B data”. In: *Transportation Research Interdisciplinary Perspectives* 34 (2025), p. 101669. doi: [10.1016/j.trip.2025.101669](https://doi.org/10.1016/j.trip.2025.101669).