Trajectory clustering of air traffic flows around airports

Xavier Olive, Jérôme Morio

Aerospace Science and Technology ••• (••••) •••–•••

Contents lists available at ScienceDirect

Aerospace Science and Technology

www.elsevier.com/locate/aescte

Trajectory clustering of air traffic flows around airports

Xavier Olive *, Jérôme Morio

ONERA – Université fédérale de Toulouse, 2 avenue Édouard Belin, 31055 Toulouse cedex 4, France

ARTICLE INFO

Article history:
Received 21 June 2017
Received in revised form 15 June 2018
Accepted 16 November 2018
Available online xxxx

Keywords:
ADS-B
Air traffic flows
Trajectory clustering
Outlier trajectory analysis

ABSTRACT

We present a new approach to separate air traffic trajectories in an area constrained by operational procedures. This technique is applied on a set of real trajectories in Toulouse terminal manoeuvring area (TMA). The resulting clusters foster good understanding of the structure of traffic and of how controllers schedule landings at Toulouse–Blagnac airport; on the other hand, a group of peculiar trajectories emerge with useful information calling for further analysis and paving the way for a probabilistic approach to risk assessment in air traffic safety.

1. Introduction

Identifying air traffic flows is a topic of interest [1–3] with room for improvement in regard to several applications such as traffic analysis [4], load balancing on sectors monitored by air traffic controllers [5]; anomaly detection of events [6] or diverging behaviours that have a higher risk of leading to traffic incidents (loss of separation, runway excursion, etc.).

A way to assess rare aircraft incidents (e.g., runway excursion) is to identify contributing factors (e.g., late braking, long landing, inappropriate flare, unstable approach) and to build a dependency tree (e.g., long landing may be the result of an unstable approach not followed by a go around) that describes the causality between these factors. Probabilities are then fed into such models in order to evaluate the assessed risk. Identifying air traffic flows is a key issue in such treatment as it contributes to identify and estimate the probability to observe an aircraft with a non standard behaviour. In other words, we aim at detecting trajectories that do no fall in a common flow of trajectories. We can then place them in context in order to understand and model what makes them outliers. The understanding we get from this kind of study should provide insight and improve our models of contributing factors.

Trajectories are mathematical objects used to describe the evolution of a moving object. They are described by a state vector with parameters $(x(t), y(t), \cdots)$ that evolve in time. In practice, this state vector is only known at some sampled times. The term $(x_i, y_i, \cdots)$ represents the state vector at time t_i. For clarity concerns, we will name trajectory a sequence of recordings associated to a moving object. The explosion of recorded (and sometimes, available) data makes the study of trajectories a popular topic and opens new fields of research in data mining common patterns and identifying outliers from recorded trajectories [7].

The literature in trajectory clustering addresses the issue in various ways; the relevancy of each method is highly dependent on the nature of the moving objects tracked and on the kind of information we want to extract from the data. Lee et al. [8] present a clustering method based on the identification of common sub-trajectories and apply their techniques to the study of hurricanes. Gariel et al. [9] address the specificity of aircraft trajectories around San Francisco airport: they perform a clustering on turning points of trajectories followed by a sequence mining on these clusters. However, the spatial distribution of all turning points on our set of data does not separate well into clusters. Puechmorel et al. [10] manipulate trajectories as purely mathematical functional objects: they warp trajectories so as to highlight more intelligible tracks and define distances as a minimum energy necessary to bundle trajectories. However as their approach modifies the data records, it may not be appropriate for safety analysis.

The key to a proper clustering of trajectories lies in the proper definition of a distance. Jeung et al. [11] present different popular metrics, like the Hausdorff distance but it does not apply well on converging air traffic flows. Li [12] or Conde [13] consider resampling trajectories so as to transform them into n-dimensional vectors on which Euclidean distances are computed. This approach is relevant in some applications but can be problematic as an aircraft being put on holding stacks before landing may see this part

* Corresponding author.
E-mail addresses: xavier.olive@onera.fr (X. Olive), jerome.morio@onera.fr (J. Morio).
URLs: http://www.onera.fr/en/staff/xavier-olive (X. Olive), http://www.onera.fr/en/staff/jerome-morio (J. Morio).

https://doi.org/10.1016/j.ast.2018.11.031
1270-9638/

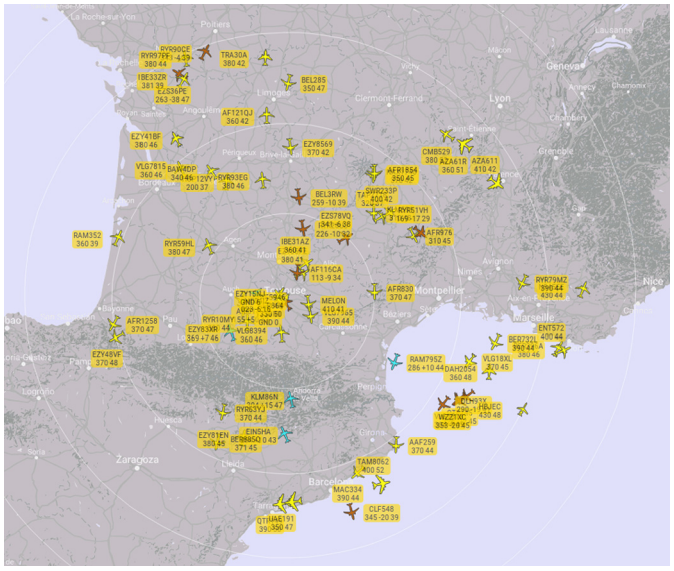


Fig. 1. An instant view of traffic as seen by a Radarcape antenna.

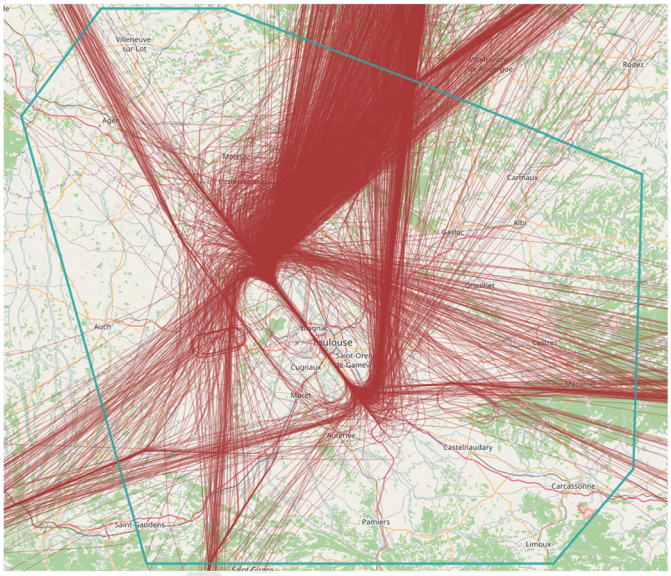


Fig. 2. The full dataset: three weeks of recorded trajectories. The border delimits the convex hull of the TMA.

of the trajectory trimmed out after resampling. In general, it is rather difficult to define a meaningful distance that separates trajectories well, generalises on full trajectories and computes in a reasonable time.

We present a different approach to identify patterns in landing with two objectives: understand how controllers schedule landing on a specific airport, then analyse outlier trajectories with respect to the risk of runway excursion. The proposed algorithm computes a clustering on subsets of significant points of trajectories while keeping a dependency tree of their temporal chaining (Section 3); then associates trajectories to root-to-leaf paths in the dependency tree based on the clusters they cross (Section 4). The full process is rather quick, efficient and robust, and detects outlier trajectories calling for further analysis (Section 5).

2. The dataset

ADS-B is a cooperative surveillance technology for air traffic control. An aircraft determines its position via satellite, inertial and radio navigation and periodically emits it (roughly 1 Hz) with other relevant parameters to ground stations and other equipped aircraft. A decent ADS-B receiver antenna can receive messages from cruising aircraft located up to 400 km far away (Fig. 1); the range is lower for aircraft flying in low altitude.

We present in this paper the result of three weeks of ADS-B recording from 5 to 30 December 2016.[1] Apart from positional messages, aircraft also emit a callsign, i.e. an 8-character identifier (like `"EZY62FN"`) for a flight (the EasyJet flight from Berlin–Schönefeld and scheduled to land in Toulouse–Blagnac at 19:30). After filtering out all flights not landing in Toulouse, we simplified trajectories as they enter the TMA through a Douglas–Peucker algorithm [14,15] as on Fig. 3, with $\varepsilon = 1$ km (i.e. roughly: remove points less than ε far from a straight line) and reduced about 5 GB worth of data to less than 200 kB for 1991 trajectories. Most trajectories now consist of 5 to 7 points, but can reach 20 points for aircraft being put on holding stacks.

Fig. 2 displays all trajectories we consider in this paper, as they are trimmed within the TMA. We can already remark that the traffic bound for Toulouse airport is heavily unbalanced, with its major part coming from the North-East side of the map. Unbalanced density is a common difficult pattern in clustering that we propose to address with our algorithm.

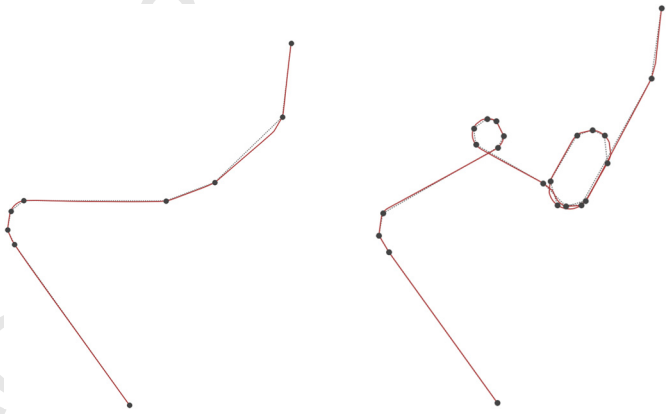


Fig. 3. Douglas–Peucker simplification of two sample trajectories (in red) with $\varepsilon = 1$ km. The dashed lines connect the points selected by the algorithm. A trajectory put on holding stack yields more points than a regular approach. (For interpretation of the colours in the figure(s), the reader is referred to the web version of this article.)

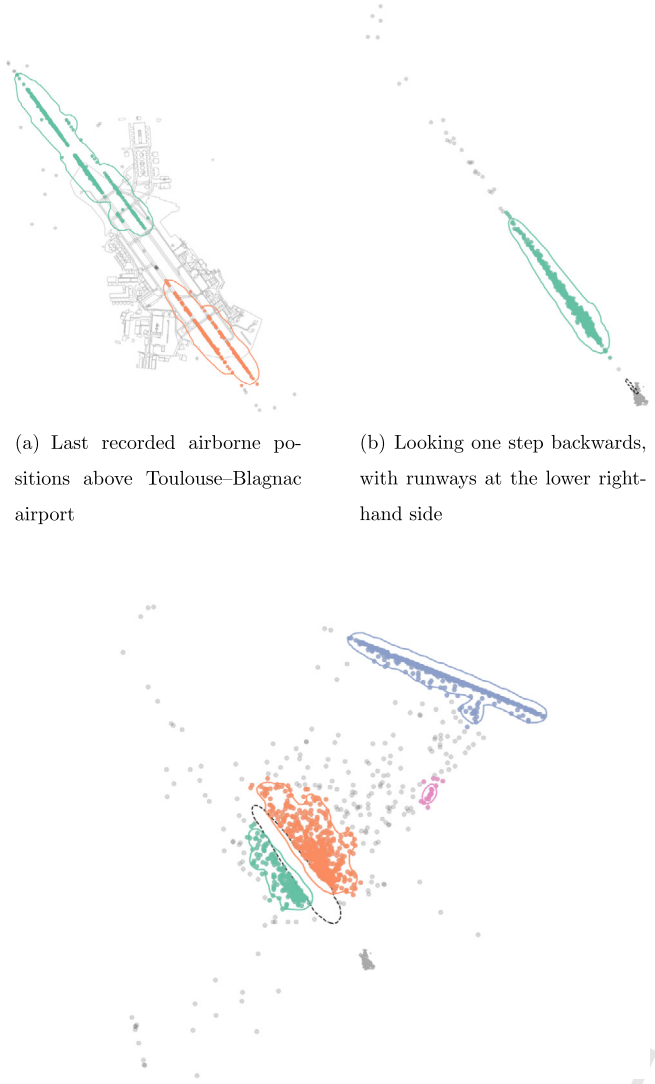
Aircraft may land from the South-East (QFU32, or 323°) or from the North-West (QFU14, or 143°). QFU is a Q-letter code naming the magnetic direction (or number) of the runway to be used. Air traffic management concertedly determines a QFU based on wind.

3. Clustering and sequencing significant points

Fig. 4(a) represents Toulouse–Blagnac airport with the last airborne positions recorded for each flight. The quality of the reception is variable; yet, we can distinguish two main clusters according to the direction of landing. Fig. 4(b) looks one step backward and displays all one-before-last stored points for aircraft landing on QFU14. The cluster that pops out corresponds to the moment aircraft aligned with the runway, also called *roll-out point*.

Again, we select the points in all trajectories that fall in the area delimited by our cluster, look one step backward and filter out points that are in the current cluster. Fig. 4(c) links the new resulting clusters (full lines) to previous cluster (dashed lines) in our backward-looking process. A basic recursion (to be formalised

[1] The dataset has been made available at the following link: https://doi.org/10.5281/zenodo.891468.



(a) Last recorded airborne positions above Toulouse–Blagnac airport

(b) Looking one step backwards, with runways at the lower right-hand side

(c) Looking two steps backwards. New resulting clusters are in full lines; all trajectories passing by these clusters will change directions in the cluster from previous iteration, delimited by dashed lines.

Fig. 4. Recursive clustering of significant points on recorded trajectories.

further) builds a full dependency tree of common trajectory segments as reflected on Fig. 5.

We compute clusters with a standard DBSCAN algorithm and select points that lie within a cluster through Kernel Density Estimation. Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is a common density-based clustering algorithm [16,17]. It finds clusters based on a density criterion ε and a minimum sample value n: the cluster forms from core elements having at least n neighbours within an ε distance, and grows by finding new neighbours to the clusters until no other element can be added. Outliers are elements that do not belong to any cluster.

DBSCAN could predict whether a new point falls inside an existing cluster, but as we will need more than a yes/no answer further in our presentation, we use Kernel Density Estimation to evaluate the distribution of samples inside a cluster. Let $\vec{u_1}, \cdots \vec{u_n}$ be a set of independent and identically distributed random samples in dimension d with unknown Probability Density Function (PDF) f. We can estimate f with a Kernel Density Estimator (KDE) as:

$$\widehat{f}_h(\vec{u}) = \frac{1}{Nh^d} \sum_{i=1}^{N} \mathcal{K}\left(\frac{\vec{u} - \vec{u_i}}{h}\right), \tag{1}$$

where $\mathcal{K}$ is a multivariate kernel (a non-negative symmetric function that integrates to one) and h a parameter called bandwidth. We set the same bandwidth parameter for each of the two dimensions of $\vec{u}$, namely the x, y coordinates (conformal conic projections from latitudes and longitudes). There are many available kernels (e.g., Gaussian, uniform, sigmoid, etc.); we prefer here the Epanechnikov kernel that is appropriate with bounded support distributions:

$$\mathcal{K}(\vec{u}) \propto \left(1 - \vec{u}^T \cdot \vec{u}\right) \mathbb{1}_{(\vec{u}^T \cdot \vec{u} \leq 1)} \tag{2}$$

We fit a PDF on all core elements from each cluster, with $h = \varepsilon$ (that is the density criterion from DBSCAN). Since all elements of a cluster fall within distance ε of one of the core elements, the domain delimited by this cluster is defined as:

$$\left\{(x, y) \in \mathbb{R}^2 \text{ s.t. } \widehat{f}_\varepsilon(x, y) > 0\right\} \tag{3}$$

We may now formalise the proposed clustering method of this article in Algorithm 1. We name Ω the set of all (x, y)-coordinates that constitute our full set of trajectories; and $\widehat{\Phi}$ a sequence of PDF $\widehat{f}_i$ built through recursion.

Algorithm 1 Dependency tree of trajectory segments

```
function GROW_TREE(clusters, Φ̂)
  for all c ∈ clusters do
    fit density f̂ to all (x_i, y_i) ∈ core elements of c
    add f̂ as a new child node to the tree
    S' ← {(x_i, y_i) ∈ Ω s.t. f̂(x_i, y_i) > 0}
    S ← {(x_i, y_i) ∈ Ω s.t. (x_{i+1}, y_{i+1}) ∈ S'}        ▷ look backward
    for all f̂* ∈ Φ̂ do                                       ▷ see dashed lines on Fig. 4(c)
      S ← S − {(x_i, y_i) ∈ S s.t. f̂*(x_i, y_i) > 0}
    end for
    if card S > threshold then
      GROW_TREE(DBSCAN(S), Φ̂ ∪ {f̂})
    end if
  end for
end function
▷ Now we start the recursion
GROW_TREE(DBSCAN(last positions), Φ̂ = {})
```

4. Labelling trajectories

Algorithm 1 produces a tree whose root node is the airport and all paths to leaf nodes describe a potential flow of aircraft following landing procedures. Fig. 5 plots a geographical representation of this tree. A potential pattern p is therefore described as a suite of PDF $\left(\widehat{f}_{p_i}\right), i \in \{0, \cdots n_p\}$, n_p being the depth of the leaf node associated to p.

As our algorithm considers segments of trajectory, we shall keep in mind that some dense segments may emerge as part of a set of flights that gets more sparse in later parts of the trajectory. We insist on this aspect as we study the outliers in Section 5. Therefore, there may be a number of leaf-to-root paths in our tree that do not result in clusters.

We represent all trajectories t in Ω as a sequence of $\left(\vec{u_j}\right) = \left(x_j, y_j\right)$, $j \in \{0, \cdots n\}$. In addition, we describe every possible path p by a sequence of $\left(\widehat{f}_{p_i}\right), i \in \{0, \cdots n_p\}$. We define as m_t^p the following (n_p, n)-matrix:

$$\left(m_t^p\right)_{ij} = \left(\widehat{f}_{p_i}\left(\vec{u_j}\right)\right)_{ij} \tag{4}$$

that is, a probability for each point of the trajectory to fall within one of our cluster. A square and diagonal matrix m_t^p with

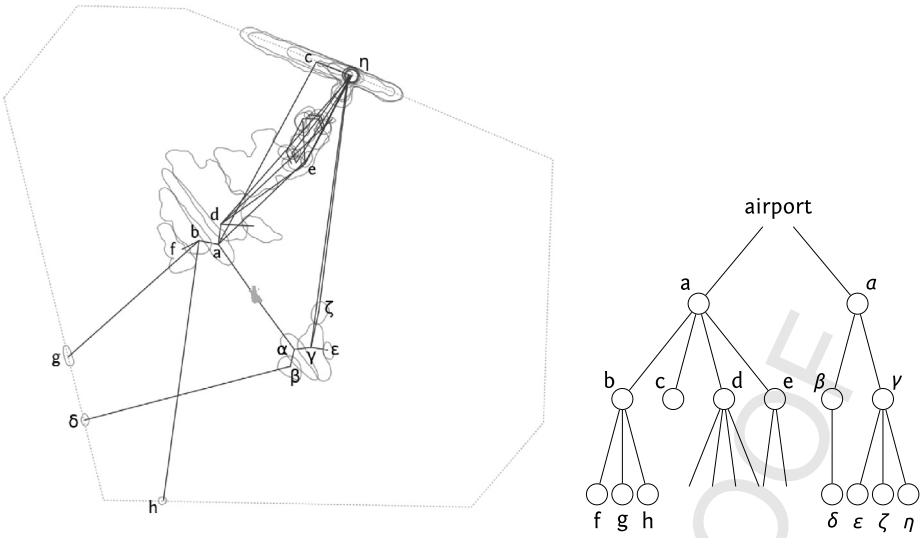


Fig. 5. Dependency tree (map view to the left, unfolded view to the right) as produced by GROW_TREE algorithm of Algorithm 1. Latin (resp. Greek) letters label nodes for aircraft landing on QFU 14 (resp. QFU 32).

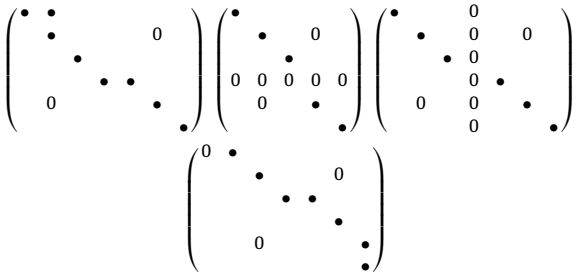


Fig. 6. Patterns for matrix m_t^p; a • means $\widehat{f}_{p_i}(\vec{u}_j) > 0$.

$\forall i, \widehat{f}_{p_i}(\vec{u}_i) > 0$ and $\forall i \neq j, \widehat{f}_{p_i}(\vec{u}_j) = 0$ would mean that our candidate trajectory passes through all *gates* of the candidate path, and that for each gate, exactly one point of the trajectory falls into that gate.

A *gate* is a nice metaphor to describe a node of our tree as part of a procedure, i.e. a candidate leaf-to-root path. We say a point of a trajectory falls into a gate if the evaluation of its associated PDF is strictly positive, that is the trajectory goes through one of the *bubbles* of Fig. 5. A gate can be compared to a distribution probability centred around a procedure point that the aircraft must fly over before landing.

In practice, those matrices are not square, so it may happen that a point falls into two consecutive gates, or that several consecutive points of a trajectory fall into the same gate. Yet if all points of the trajectory fall in the right order into each gate of a possible leaf-to-root path, we can assert that the trajectory flies according to this path and may form a cluster with other trajectories meeting this criterion.

Fig. 6 shows possible patterns for matrix m_t^p. Pattern a) is an extension of the diagonal matrix described here above: some consecutive points of the trajectory fall in the same *gate*. Pattern b) is representative of trajectories that fly only through gates of a path, yet miss one of its gates p_i. Pattern c) is a trajectory which follows a path, yet has one point outside the gates. It cannot match the path as the trajectory is too distorted compared to the set of other trajectories matching this path.

Pattern d) is similar to c) but must be considered as matching. From the structure of the tree represented on Fig. 5, some leaf nodes do not reach the border of the TMA: the origin of the trajectories converging to this node is not dense enough to form

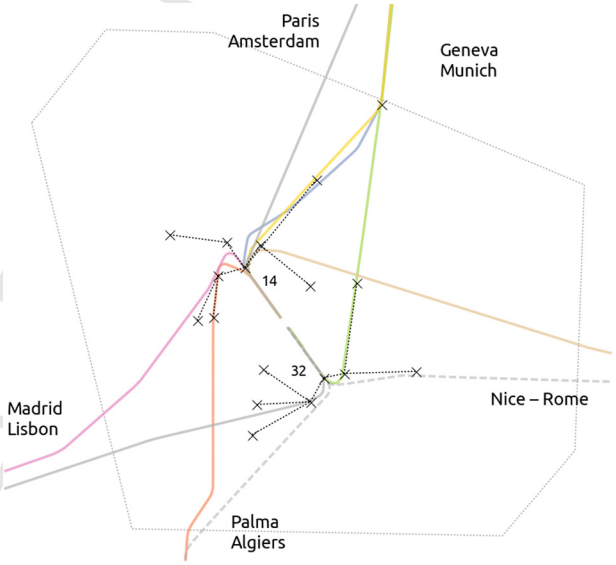


Fig. 7. Resulting clusters for traffic in Toulouse TMA; significant trajectories for every cluster are in full lines, two outliers complete the picture in dashed lines. Crosses and dotted lines reflect official procedures currently implemented.

clusters. However, we can bundle all these trajectories as long as they fly through all the gates of the path.

Based on this analysis, we associate a numerical score s_t^p from m_t^p:

$$s_t^p = \begin{cases} 0 & \text{if } \forall i \geq 0, \quad \sum_j \left(m_t^p\right)_{i,j} = 0 \\ 0 & \text{if } \exists t \geq 0 \text{ s.t. } \forall j \geq t, \quad \sum_i \left(m_t^p\right)_{i,j} = 0 \\ \sum_i \max_j \left(\left(m_t^p\right)_{i,j}\right) & \text{otherwise} \end{cases} \tag{5}$$

Trajectory t matches pattern p if $\forall p' \neq p, s_t^p > s_t^{p'}$. If $s_t^p = 0$ for all patterns p, trajectory t is flagged as outlier. Optionally, we can set a threshold number of flights matching a path to consider them as a cluster. Potential clusters with a number of flights below the threshold may redistribute the trajectories to other candidate patterns or to the pool of outliers.

Fig. 7 plots one flight (the one with the highest score) for each cluster determined by our method (with a threshold number of

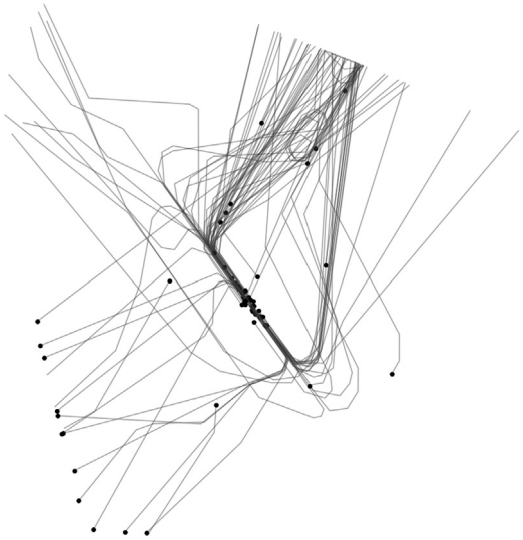


Fig. 8. Outliers with respect to our clustering of trajectories: incomplete trajectories.

Fig. 9. Outliers with respect to our clustering of trajectories: trajectories on holding stacks.

flights of 5). The structure of traffic becomes clear, with several directions of incoming flights and different ways to align on both QFU. We added to this plot two types of information: two outlier flights in dashed lines complete the picture for flights coming from the South; crosses and dotted lines reflect standard arrival procedures (STARs) ruling approaches at Toulouse airport.

The dense traffic from the North and bound for QFU 14 splits into several clusters showing how aircraft are scheduled according to a pattern of *linear hold*: aircraft are scheduled by timing their turn into final approach so as to land no less than two minutes apart. One of the clusters is formed from trajectories being warped even further in order to respect tight scheduling constraints; on the other hand, a cluster forms from flights from the North which get clearance from the ATC to shortcut the entry point to the TMA and align directly into final approach.

These two emerging clusters show how STAR procedures published by eAIP[2] are not sufficient to describe traffic flows around airports. To sum up, it is interesting to see how official procedures somehow emerge from the clustering we produce, and how the picture is also completed with common practices used to regulate the traffic, i.e. giving aircraft the best approaches for their direction of travel according to the current traffic and weather conditions.

From a performance point of view, the building of the dependency tree described in Section 3 took few seconds to compute with the implementation of DBSCAN provided by scikit-learn (Intel Core i5 2.40 GHz). The labelling was the most intensive part as it took about 6 minutes with our pure Python implementation to label all trajectories.

5. A study of the outliers

Our clustering of trajectories yields about 30% of outliers. The study of the outliers is interesting as these trajectories may contain information about the circumstances under which a landing procedure may differ from the norm and result in dangerous behaviour, or worse, traffic incidents.

With our procedure for attributing a label to a trajectory, we outcast by default those that lack data around landing. Fig. 8 plots all trajectories for which the last recorded points do not end in one of our original clusters above the runways (refer to Fig. 4(a)). We find on the lower left-hand side a series of trajectories which do not seem to land at Toulouse airport. After investigating this anomaly, we found that the callsigns associated to these trajectories correspond to flights from Nantes or Bordeaux, which land in Toulouse before taking off again and flying further to their final destination (here Tenerife, Canaries and Agadir) with the same callsign. As we made this mistake in the preprocessing of our trajectories (the trajectories should have been trimmed after the aircraft landed) in good faith,[3] we mention them here as they demonstrate that the method we present was robust enough to keep them out of any cluster.

Fig. 9 plots trajectories being put on holding stacks of various orientations and positions. We remark that the northernmost hippodrome could be dense enough to form some kind of cluster. However, as no clear way out of this pattern emerges (*linear holds* are pushed to extremes in such saturated situation), our algorithm will not make a pattern out of these trajectories. It could be interesting to study their distance to statistically standard landings once they are back on a regular track. More generally, some trajectories enter the TMA from areas of lesser density, yet still match existing clusters on the latter part of the landing; we could craft a refined way to match a trajectory to a cluster and use this degree of matching to evaluate some degree of risk in an approach.

The outlier trajectory on Fig. 10 is associated to callsign `AF526KB` on December 26. This flight started its approach for QFU 32 but changed halfway and aligned on QFU 14. Indeed, a QFU change at the turning point of this trajectory is consistent with the landing history of that day. If we look more closely on Fig. 11, we can see that the altitude profile (pale dashed lines) is not surprising, yet there seems to be a sudden change in ground speed (full line) and vertical speed (dotted-dash) at 20:52 on that day, about 50 seconds before the aircraft changes its trajectory. As sudden as the change may look, the aircraft managed to align properly on QFU 14.

A late QFU change may lead to unstable approaches, which may lead to a late touchdown on the runway (long landing), which is a potential risk of runway excursion. In practice, the estimation of the different conditional probabilities involved in the estimation of the risk of runway excursion (like the probability of an unstable approach for flights subject to a late QFU change), may only be

[2] https://www.sia.aviation-civile.gouv.fr/.

[3] Institutions like Eurocontrol try to enforce using a different Globally Unique Flight Identifier (GUFI) for each leg of this kind of flights rather than callsigns that are subject to such misinterpretations. GUFIs would still be helpless with rerouted flights.

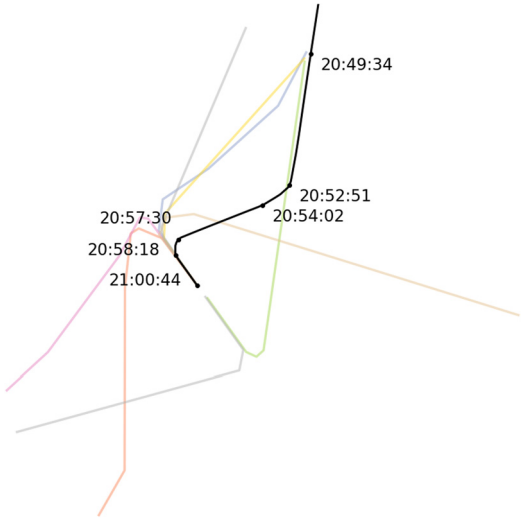


Fig. 10. Trajectory clusters and outlier flight AF526KB (bold) on December 26.

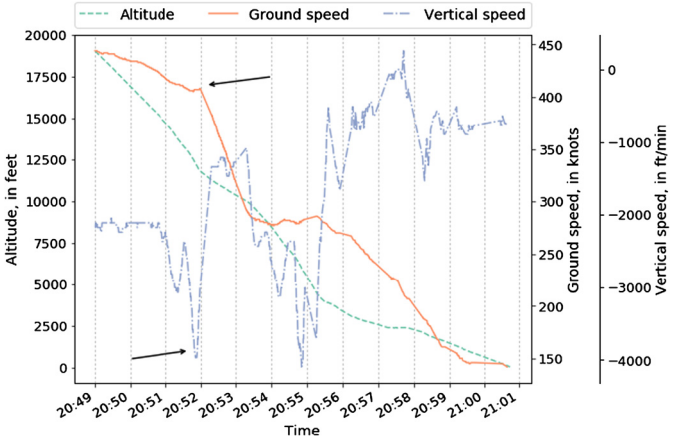


Fig. 11. QFU change appears in speed plots on outlier flight AF526KB on December 26, 20:52.

roughly evaluated by experts since those events are not properly recorded or reported by airlines. A good analysis of outlier trajectories yielded from a bigger amount of data could improve the estimation of such conditional probabilities, hence improve the estimation of such risks.

6. Conclusion and future works

We presented a new method to cluster aircraft trajectories before landing and compared the resulting clusters to official STAR procedures for Toulouse airport: the clustering complements the procedures well as it adds insight concerning common practices of the ATC in charge of approaches. Then we explained how the study of outliers may nurture a more precise risk assessment analysis. In the specific case of outlying trajectory we address, we find a late QFU change to a shorter STAR procedure: a further analysis on a larger time scale of trajectories subject to a late QFU change could be a valuable result. More generally, we believe that a source of data with a global view on traffic (such as ADS-B) in addition to other data from flight data managers (owned by airline operators) announces promising perspectives as it considers each aircraft in context, with information about how other aircraft fly in the same environment.

Future works may include a validation on other airports with various runway configurations, a better metric measuring a deviation from a trajectory cluster core. A study of the evolution of traffic patterns over the seasons, correlated with meteorological reports (METAR) would also be precious, as weather (gale, fog, etc.) has a strong impact on how traffic in the TMA is regulated and on how aircraft fly before landing. The use of other available data like vertical speed, indicated airspeed, roll, track, or heading angles would also be of interest; as well as a coupling with ACARS messages aircraft send shortly before landing with information about required maintenance operations.

Conflict of interest statement

None declared.

References

[1] S. Sidiropoulos, A. Majumdar, K. Han, W. Ochieng, Identifying significant traffic flow patterns in multi-airport systems terminal manoeuvring areas under uncertainty, in: 16th AIAA Aviation Technology, Integration, and Operations Conference, 2016, p. 3162.
[2] H. Chida, C. Zuniga, D. Delahaye, Topology design for integrating and sequencing flows in terminal maneuvering area, Proc. Inst. Mech. Eng., G J. Aerosp. Eng. (2016), https://doi.org/10.1177/0954410016636155.
[3] W. Jiening, H. Qizhen, L. Yongxin, Z. Chunfeng, Visualizing air traffic flow management alert information using squarified treemaps, in: 2009 Sixth International Conference on Computer Graphics, Imaging and Visualization, 2009, pp. 419–422.
[4] M.C. Rocha Murça, A robust optimization approach for airport departure metering under uncertain taxi-out time predictions, Aerosp. Sci. Technol. 68 (2017) 269–277.
[5] Y.-H. Chang, S. Solak, J.-P.B. Clarke, E.L. Johnson, Models for single-sector stochastic air traffic flow management under reduced airspace capacity, J. Oper. Res. Soc. 67 (1) (2016) 54–67.
[6] Spatial-temporal traffic flow pattern identification and anomaly detection with dictionary-based compression theory in a large-scale urban network, Transp. Res., Part C, Emerg. Technol. 71 (2016) 284–302.
[7] Y. Zheng, Trajectory data mining: an overview, ACM Trans. Intell. Syst. Technol. 6 (3) (2015) 29.
[8] J.-G. Lee, J. Han, K.-Y. Whang, Trajectory clustering: a partition-and-group framework, in: Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data, ACM, 2007, pp. 593–604.
[9] M. Gariel, A.N. Srivastava, E. Feron, Trajectory clustering and an application to airspace monitoring, IEEE Trans. Intell. Transp. Syst. 12 (4) (2011) 1511–1524.
[10] S. Puechmorel, F. Nicol, Entropy minimizing curves with application to flight path design and clustering, Entropy 18 (9) (2016) 337.
[11] Y. Zheng, X. Zhou, Computing with Spatial Trajectories, Springer Science & Business Media, 2011.
[12] L. Li, M. Gariel, R.J. Hansman, R. Palacios, Anomaly detection in onboard-recorded flight data using cluster analysis, in: Digital Avionics Systems Conference (DASC), 2011 IEEE/AIAA 30th, IEEE, 2011.
[13] M. Conde Rocha Murca, R. DeLaura, R.J. Hansman, R. Jordan, T. Reynolds, H. Balakrishnan, Trajectory clustering and classification for characterization of air traffic flows, in: 16th AIAA Aviation Technology, Integration, and Operations Conference, 2016.
[14] D.H. Douglas, T.K. Peucker, Algorithms for the reduction of the number of points required to represent a digitized line or its caricature, Cartographica 10 (2) (1973) 112–122.
[15] G. Sim, J. Chung, Y. Sung, 3D UAV flying path optimization method based on the Douglas–Peucker algorithm, Springer Singapore, Singapore, 2017, pp. 56–60.
[16] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, in: Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining, 1996, pp. 226–231.
[17] P. Berkhin, A survey of clustering data mining techniques, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 25–71.