

A Gaussian process based decision support tool for air-traffic management

Willem J. Eerland^{1,*}, Simon Box^{1,†}, Hans Fangohr^{1,2,‡},
and András Sóbester^{1,§}

¹ *University of Southampton, Southampton, England SO17 1BJ, United Kingdom*

² *European XFEL GmbH, Holzkoppel 4, 22869 Schenefeld, Germany*

Technological developments in the last decade have shifted challenges in traffic flow management from obtaining and storing data, to analysing and presenting the enormous amount of available trajectory data in a comprehensible manner. This paper introduces a novel approach to visualising air-traffic, shifting the focus from displaying traffic density, towards directly visualising the flight corridors used by air-traffic. Such an approach is suitable for visualising air-traffic in three dimensions, which is particularly helpful in the vicinity of an airport where the air-traffic often changes level. Furthermore, the approach is data-driven, allowing the comparison of multiple trajectory datasets in order to identify changes in traffic corridors related to changing air-traffic and weather conditions. Finally, by using the probabilistic nature of the approach, it is possible to quantify the air-traffic complexity in terms of the traffic structure. The results presented in this paper show the approach applied to a trajectory dataset as measured by ground-radar near Denver airport (DEN).

*Ph.D. Candidate. Aeronautics, Astronautics, and Computational Engineering. Member AIAA. w.j.eerland@soton.ac.uk

†Visiting Research Fellow. Transportation Research Group.

‡Professor of Computational Modelling. Aeronautics, Astronautics, and Computational Engineering. Senior Data Analysis Scientist at XFEL.

§Associate Professor. Aeronautics, Astronautics, and Computational Engineering. Senior Member AIAA.

Nomenclature

$\boldsymbol{\mu}$	= mean vector
$\boldsymbol{\Sigma}$	= covariance matrix
σ	= standard deviation
τ	= normalised time [-]
$\boldsymbol{\tau}$	= normalised time vector [-]
\mathcal{C}	= complexity measure
\mathbb{E}	= expectation
\mathcal{GP}	= Gaussian process
k	= order of complexity
\mathbf{k}	= covariance kernel
M	= number of data-points
\mathbf{m}	= mean function
N	= number of trajectories
\mathcal{S}	= set of corridors
\mathbf{u}	= vector holding a three-dimensional coordinate [m]
\mathbf{v}	= column vector holding three-dimensional coordinates [m]
$\hat{\mathbf{v}}$	= estimated column vector holding three-dimensional coordinates [m]
x	= eastings [m]
\mathbf{x}	= eastings vector [m]
y	= northings [m]
\mathbf{y}	= northings vector [m]
z	= altitude [m]
\mathbf{z}	= altitude vector [m]

I. Introduction

As passenger demand for air travel and demand for air freight continues to increase, air-traffic grows globally, and forecasts indicate that this trend will continue over the next two decades. Therefore, in order to handle the increased air-traffic, the Air Traffic Management (ATM) system needs to increase its capacity. Currently, ATM systems operate on two different time scales. The Air Traffic Control (ATC) function operates on a short/mid term horizon, with the goal of maintaining the appropriate separation between aircraft in the different stages of their flights, from departure to arrival. The Traffic Flow Management (TFM) function operates on a long term horizon by defining the flow patterns to ensure an efficient organization of the overall air-traffic [1]. This paper is concerned with the long term horizon planning required in the TFM function. The goal is to provide decision support by visualising the flight corridors and produce air-traffic complexity maps as detected in a trajectory dataset.

Previous work focuses on clustering the trajectory data based on similar weather conditions [2] and similar flight paths [3], which has been proven effective at clustering and classifying air-traffic flows [4]. However, in order to provide decision support to TFM, the clustered trajectory data needs to be presented in a comprehensible manner.

Modelling air-traffic as if it moved along a limited set of predefined air routes is unrealistic. Therefore, the air-traffic flow needs to be modelled from the trajectory data as measured (e.g., by radar) [5]. An effective approach to visualising the airspace traffic flow is graph bundling by density estimation [6]. By applying graph bundling by density estimation, it is possible to gain an understanding of the traffic flow based on the trajectory data [7]. This approach has been included in an interactive program, capable of visualising traffic information with the focus on air-traffic flow analysis [8]. However, the aforementioned methods focus on visualising the traffic flow in two dimensions.

Visualising large datasets in three dimensions brings its own challenges, including the occlusion of data [9]. Technically, it is possible to process the data real-time by using a GPU-based visualisation pipeline [10], but the occlusion of information remains a problem. The use of flight corridors to visualise the air-traffic is seen in work by Salaun et al. [11], replacing the raw trajectory data with flight corridors. Here the flight corridor is represented via window frames, each defined by a vertical and a horizontal line that encloses the air-traffic. The frames also captures the dispersion by fitting a univariate normal distribution in the vertical and horizontal direction independently, where the dispersion is considered proportional to the standard deviation. We described a framework for modelling ensembles of flight trajectories as instances of a Gaussian process [12]. By doing so, missing data can be estimated, and the flight corridors are shaped by a continuous merger of ellipsoids. In these

flight corridors each individual ellipsoid represents the three-dimensional normal distribution with a constant standard deviation at each time-step. The application of Gaussian processes to model the uncertainties in aircraft trajectories has also been applied to conformance monitoring [13]. In this work, the Gaussian process model captures the behaviour of the trajectory data measured by a surveillance system and integrates the result in a conformance monitoring scheme [14].

Ellipsoids have previously been used as a measure for air-traffic complexity in three-dimensional airspace [15]. The focus of the air-traffic complexity here is on the *traffic structure*. When the focus is *air-traffic controller workload*, there is evidence that including factors such as heading, speed, and altitude change is a better measure than traffic density only [16, 17]. In neither is the air-traffic complexity directly linked to the traffic density, however it is possible to estimate a density at which the airspace becomes saturated [18].

This paper applies the approach of modelling the trajectory data as a Gaussian process to the visualisation of air-traffic trajectory data as flight corridors. This shifts the focus from density-based methods to probabilistic analysis. As a result, less frequently used flight paths are not automatically ignored, but included in the analysis. A key benefit of our approach is obtaining situational awareness from large historic datasets, otherwise difficult to visualise. Furthermore, by focusing on identifying and visualising flight corridors used by dominant flows of air-traffic, we can identify and visualise areas that have a structural presence of high air-traffic complexity in terms of traffic structure. The approach is data-driven, and requires minimum human interaction to construct the flight corridors and complexity maps from the trajectory data, save selecting the resolution of the evaluation. The flight corridors are generated via an open-source Python toolbox, developed specifically for generating corridors in two and three dimensions [19]. We provided evidence of the effectiveness of this toolbox in the analysis of rocket trajectories generated by a stochastic rocket simulator [20, 21]. The aircraft trajectory data is provided as input via a Comma Separated Values (CSV) file. The data and code required to generate the figures seen in this paper are available online [22]. This includes a Jupyter notebook which was used to produce all the figures seen in this paper.

The rest of the paper is organised as follows. Section II elaborates on how the flight corridors are formed, and how the complexity maps are generated from these flight corridors. Results using trajectory data gathered at Denver airport (DEN) are presented in section III. Finally, conclusions and future work are provided in section IV.

II. Methodology

Section A begins with an overview of our proposed method for generating a probabilistic model from clustered trajectory data. Next, section B elaborates on how a flight corridor is created from a probabilistic model. Finally, section C presents an approach for producing a complexity map from the flight corridors.

A. Probabilistic modelling of trajectories

The task at hand is to translate the trajectory data into a mean trajectory and capture the dispersion from the mean trajectory. This section contains a review of the method, specifically adapted to three-dimensional aircraft trajectory data. We make two key assumptions: the data describing all trajectories are complete and they are not corrupted by measurement noise (see [12] for a more general treatment). The trajectory data referred to in this section are assumed to be clustered, therefore all trajectories contained in a cluster belong to aircraft flying the same nominal flight path.

Given the trajectory data \mathbf{v} in the following format:

$$\mathbf{v} = \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \\ \mathbf{z} \end{bmatrix} \quad (1)$$

where \mathbf{v} is a column vector and \mathbf{x} , \mathbf{y} , and \mathbf{z} are vectors containing the coordinates easting, northing, and altitude respectively, as a function of time. Therefore, when a trajectory has M coordinates, \mathbf{v} is a $3M \times 1$ vector. As the interest lies in finding the spatial distribution of trajectories and we are not concerned with their variation in time, each trajectory is normalised in the interval $\tau = [0, 1]$, in which τ indicates normalised time. With the assumption that the measurements are made at regular intervals, and all trajectories in a cluster have an identical, constant velocity, it is valid to state that the trajectories start at the normalised time $\tau = 0$, and end at $\tau = 1$. The effect of normalising time has been investigated in previous work [12]. Following this procedure, the normalised time vector $\boldsymbol{\tau}$ to accompany the coordinate vector \mathbf{v} is equal to:

$$\boldsymbol{\tau} = \begin{bmatrix} 0 & \frac{1}{M} & \dots & \frac{M}{M} \end{bmatrix}^T \quad (2)$$

Furthermore, with the additional assumption that the aircraft travels linearly between each data point, the coordinate vector \mathbf{u} can be expressed as a function of the normalised time τ :

$$\mathbf{u}(\tau) = [x \ y \ z]^T \quad (3)$$

where x , y , and z are the coordinates from \mathbf{x} , \mathbf{y} , and \mathbf{z} , linearly interpolated as a function of τ . All trajectories can now be resampled to the column vector $\hat{\mathbf{v}}$. The structure of $\hat{\mathbf{v}}$ is identical to \mathbf{v} , seen in Equation (1), however, the size is identical across all trajectories. In the experiments described in this paper 100 steps are taken, thus $\hat{\mathbf{v}}$ is a 300×1 vector for all N trajectories found in the data. The mean vector $\boldsymbol{\mu}$ and the covariance matrix $\boldsymbol{\Sigma}$ of the multivariate normal distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ are calculated following:

$$\boldsymbol{\mu} = \frac{1}{N} \sum_{n=1}^N \{\mathbb{E}[\hat{\mathbf{v}}_n]\} \quad (4)$$

$$\boldsymbol{\Sigma} = \frac{1}{N} \sum_{n=1}^N \{\hat{\mathbf{v}}_n \hat{\mathbf{v}}_n^\top - 2\hat{\mathbf{v}}_n^\top \boldsymbol{\mu} + \boldsymbol{\mu} \boldsymbol{\mu}^\top\} \quad (5)$$

This solution fits into a Gaussian Process (GP) framework, where a function is defined as a Gaussian process \mathcal{GP} by a mean function $\mathbf{m}(\tau)$ and a covariance kernel $\mathbf{k}(\tau, \tau')$:

$$\mathbf{u}(\tau) \sim \mathcal{GP}(\mathbf{m}(\tau), \mathbf{k}(\tau, \tau')) \quad (6)$$

where

$$\mathbf{m}(\tau) = \mathbb{E}[\mathbf{u}] = \boldsymbol{\mu} \quad (7)$$

$$\mathbf{k}(\tau, \tau') = \mathbb{E}[(\mathbf{u}(\tau) - \mathbf{m}(\tau))(\mathbf{u}(\tau') - \mathbf{m}(\tau'))] = \boldsymbol{\Sigma} \quad (8)$$

A more intuitive representation of the mean function $\mathbf{m}(\tau)$, given by Equation (7), is a point that moves along a three-dimensional trajectory as a function of τ . In turn, the covariance kernel $\mathbf{k}(\tau, \tau')$, given by Equation (8), can be represented as a three-dimensional ellipsoid at a constant probability. The reference to the GP framework is included here, as it provides the mathematical tools necessary to handle noise measurements and missing data. However, the dataset used in this paper has neither, and therefore the trajectory data can be modelled via the computationally less taxing approach described in this section.

B. Flight corridor

This section introduces the algorithm for generating the flight corridors, starting from the mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ found through the methodology described in the previous section.

As mentioned before, at each time-step for a given standard deviation σ , the three-dimensional normal distribution can be expressed as an ellipsoid. For two sequential time-steps, and the corresponding two ellipsoids, the volume of these ellipsoids, and the volume

between the two ellipsoids is assumed to be part of the flight corridor. The necessary resolution of the discretization depends on the geometry of the trajectory; for the experiments described here we used 100 steps.

Next, in order to evaluate whether a point is within the ellipsoids or the volume between the ellipsoids, the Quickhull algorithm [23] is applied. Essentially, the points belonging to the ellipsoids are taken as a point-cloud, thereafter the Quickhull algorithm efficiently converts this to a hull constructed by the outer points. Using the hull, it is possible to quickly evaluate which points lie inside the flight corridor, and which lie outside. The figures seen in this paper are produced by evaluating a grid of points, however there is no such restriction. For example, the same approach can evaluate whether a flight corridor conflicts with restricted airspace.

C. Complexity map

This section describes the method implemented to evaluate air-traffic complexity. This method works via the probabilistic models as described in section A, hence the air-traffic complexity as discussed here only relates to the *traffic structure*.

The complexity measure used in the complexity map has been inspired by Prandini, Putta and Hu [15], and allows for the identification of conflict regions. This approach allows the visualisation of different orders of complexity, where the first order complexity \mathcal{C}^1 represents the probability of at least one flight corridor intersecting with a selected point in space. The second order complexity \mathcal{C}^2 represents the probability of at least two flight corridors, and so forth. The mathematical description of the k -th order complexity \mathcal{C}^k evaluated at point x is equal to Equation (9):

$$\mathcal{C}^k(x) = \sum_{\mathcal{S}} \binom{\mathcal{S}}{k} \left(\prod_n^T (p(x|n)) \right) \quad (9)$$

where $\binom{\mathcal{S}}{k}$ is a k -th combination of the set of all corridors \mathcal{S} , and $p(x|n)$ represents the probability of corridor n intersecting with point x . Note that following the method from section A, each corridor is described by 100 normal distributions $\mathcal{N}(\mu, \Sigma)$. Therefore, there are 100 probabilities corresponding to corridor n intersecting with point x . However, as we are not concerned with their variation in time (each normal distribution represents a time-step), $p(x|n)$ is taken equal to the highest probability. This corresponds to the largest chance of point x intersecting with corridor n at any given time.

Finally, a demonstration of Equation (9) is provided here. A schematic overview of the example is shown in Figure 1, where point x is surrounded by the set of corridors $\mathcal{S} = \{A, B, C\}$. Following Equation (9), the corresponding complexity measures \mathcal{C}^1 , \mathcal{C}^2 , and

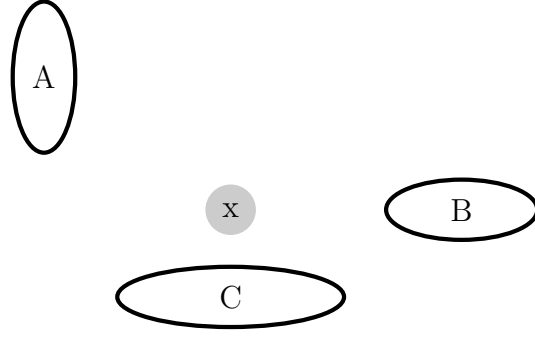


Figure 1: An example where point x is surrounded by the corridors A , B , and C .

\mathcal{C}^3 are then equal to Equations (10) to (12):

$$\mathcal{C}^1(x) = p(x|A) + p(x|B) + p(x|C) \quad (10)$$

$$\mathcal{C}^2(x) = p(x|A) \cdot p(x|B) + p(x|A) \cdot p(x|C) + p(x|B) \cdot p(x|C) \quad (11)$$

$$\mathcal{C}^3(x) = p(x|A) \cdot p(x|B) \cdot p(x|C) \quad (12)$$

This example confirms that \mathcal{C}^1 is indeed the sum of probability of each individual corridor intersecting with point x , \mathcal{C}^2 the sum of probability for all combinations of two corridors intersecting with point x , and \mathcal{C}^3 the probability of all three corridors intersecting with point x .

III. Results

In this section, the methods described in section II are applied to a trajectory dataset measured by ground-radar at Denver airport (DEN). First, using surveillance meta-data, the trajectories from aircraft that are landing at the airport are extracted from the trajectory data. The aircraft trajectory data of *approaching* aircraft holds 909 trajectories. Secondly, trajectory data are cut-off at an altitude of 800 metres. Finally, these trajectories are clustered according to their nominal flight paths via the method presented by Gariel, Srivastava and Feron [3], where the epsilon parameter used in the clustering is 0.3, with a minimum number of trajectories per cluster of 10. This method identified 10 distinct nominal flight paths, placing 909 trajectories in 10 clusters, where 339 trajectories are considered outliers. A summary of the clustered trajectory data is available in Table 1.

Up to 50 trajectories per cluster are seen in Figures 2a and 3a, showing identical trajectory

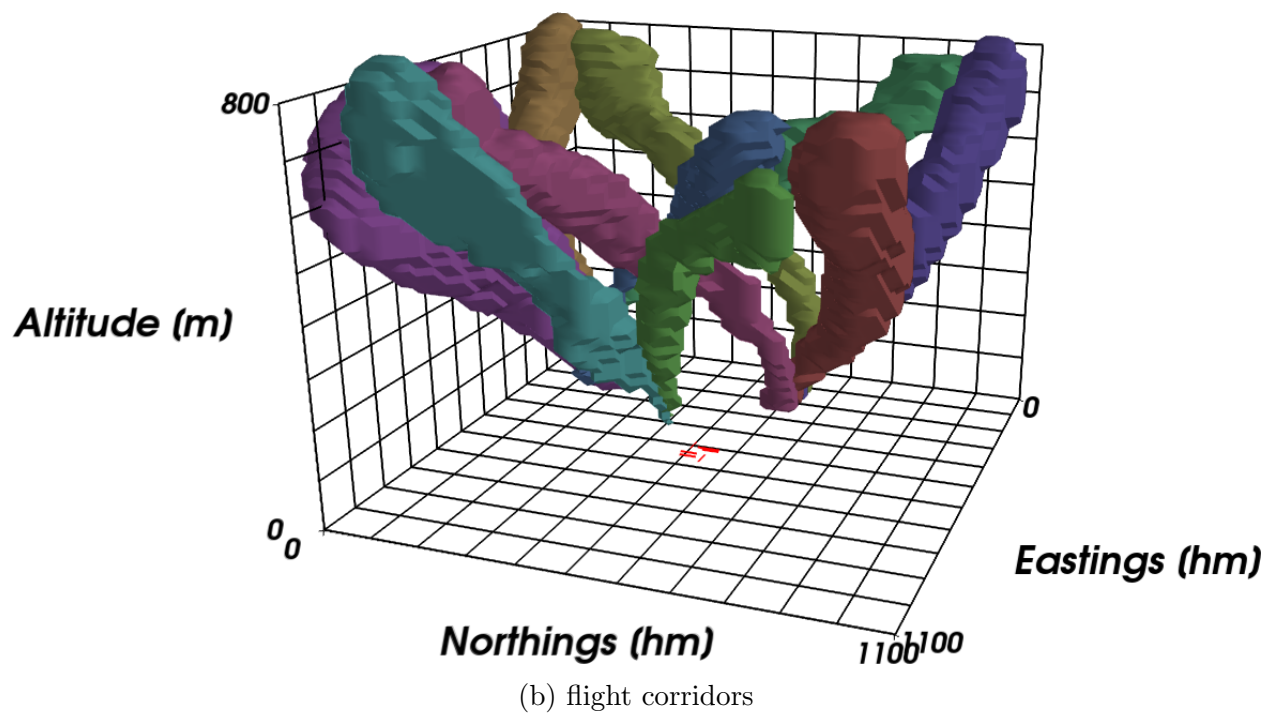
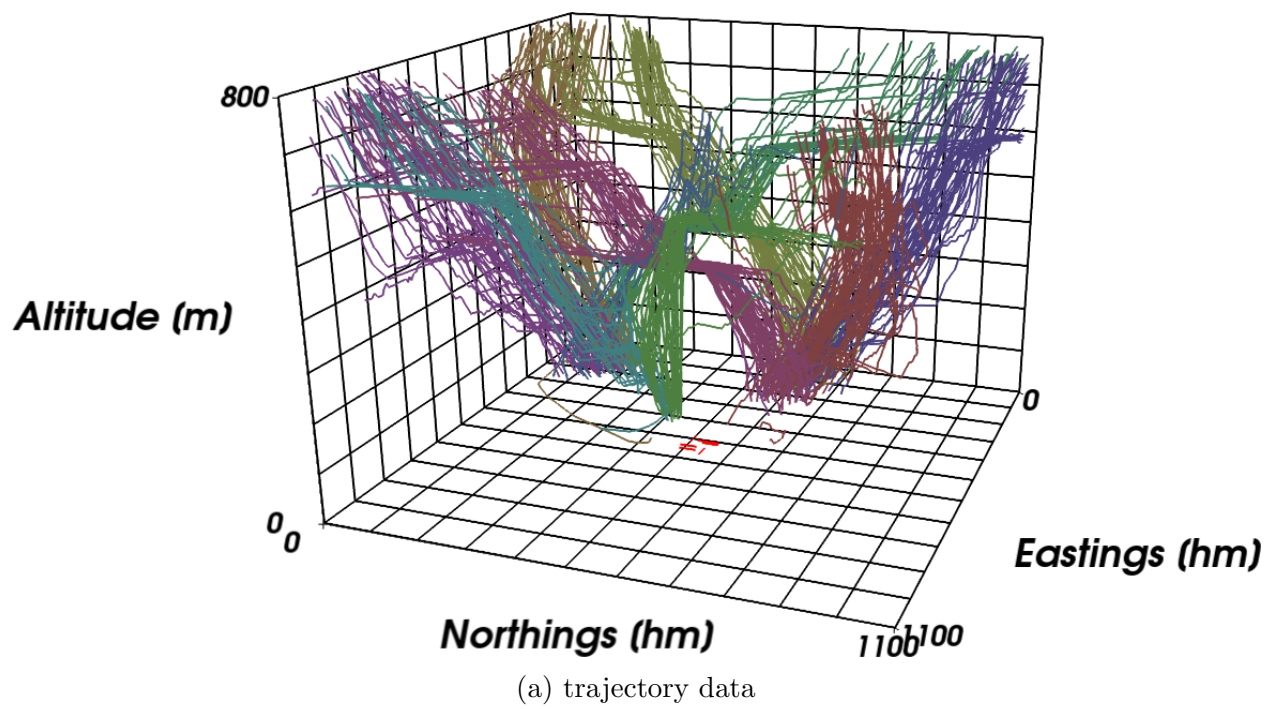


Figure 2: Comparing trajectory data and flight corridors, first viewpoint.

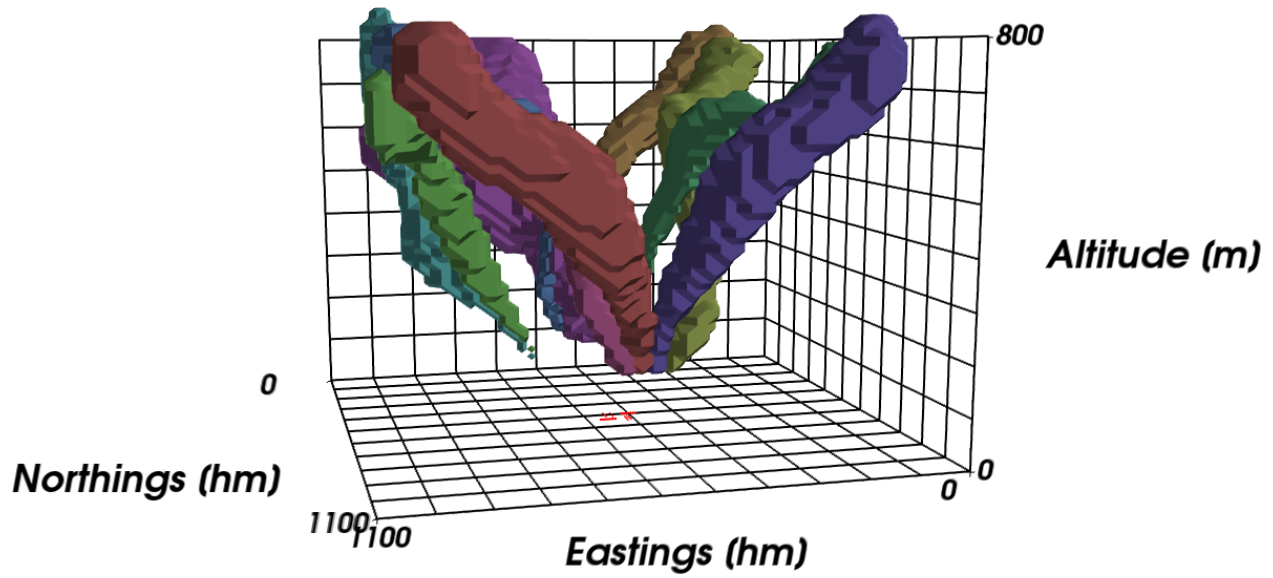
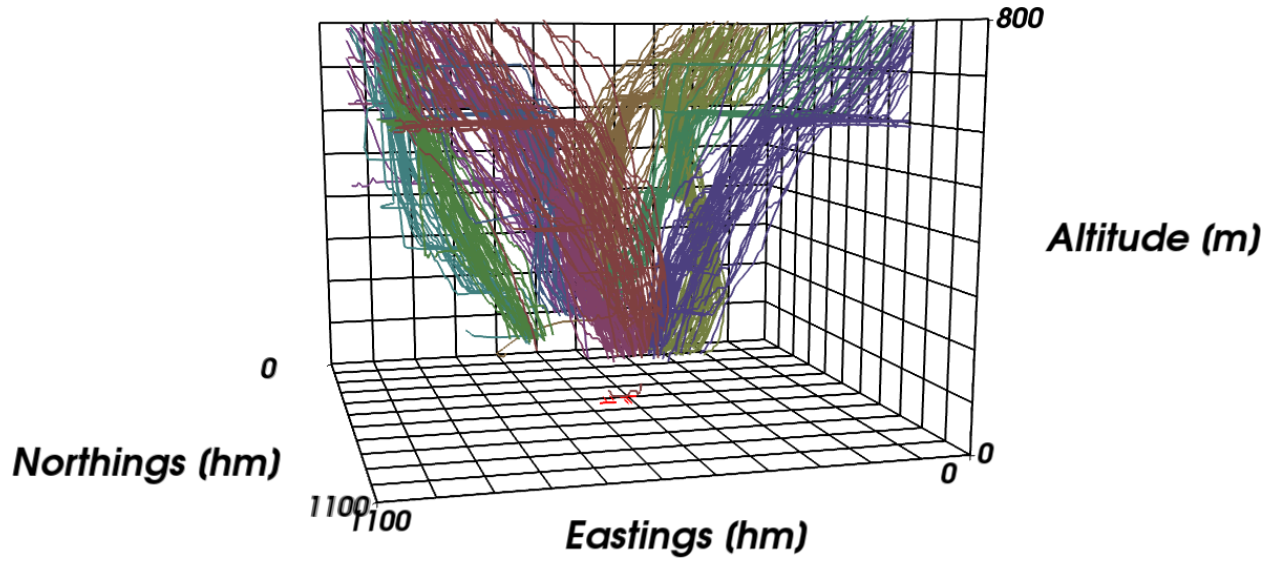
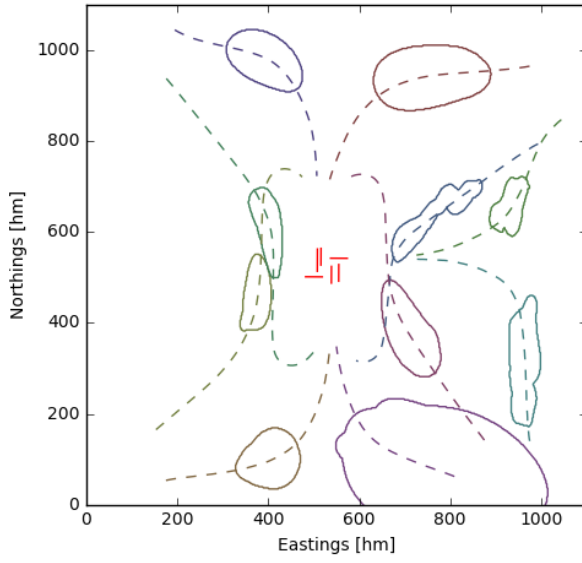


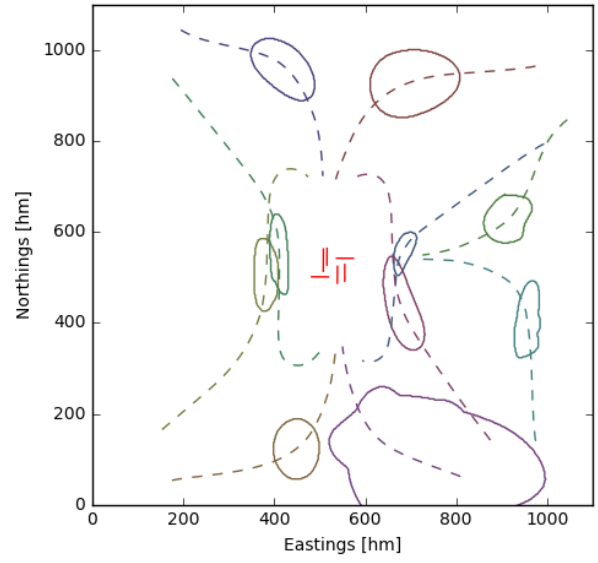
Figure 3: Comparing trajectory data and flight corridors, second viewpoint.

Table 1: An overview of the clustered trajectory data.

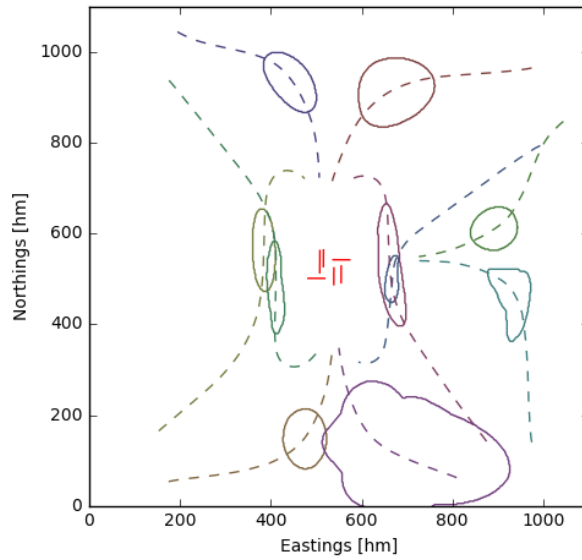
Cluster	1	2	3	4	5	6	7	8	9	10	<i>total</i>
Size	80	63	58	37	26	25	10	89	104	78	570



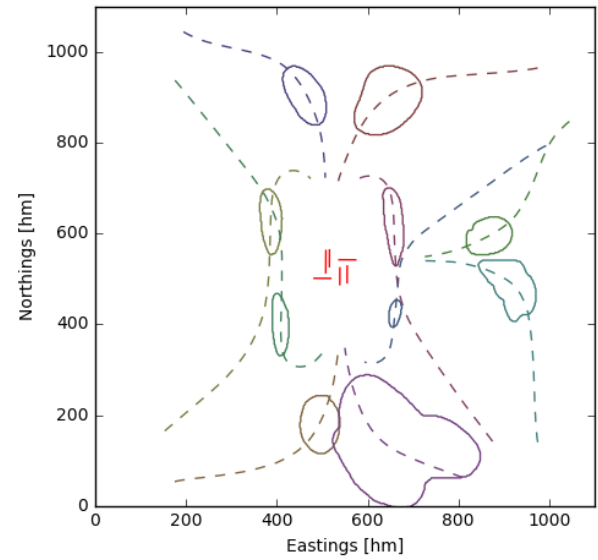
(a) 550 m



(b) 500 m



(c) 450 m



(d) 400 m

Figure 4: Flight corridors at a constant altitude.

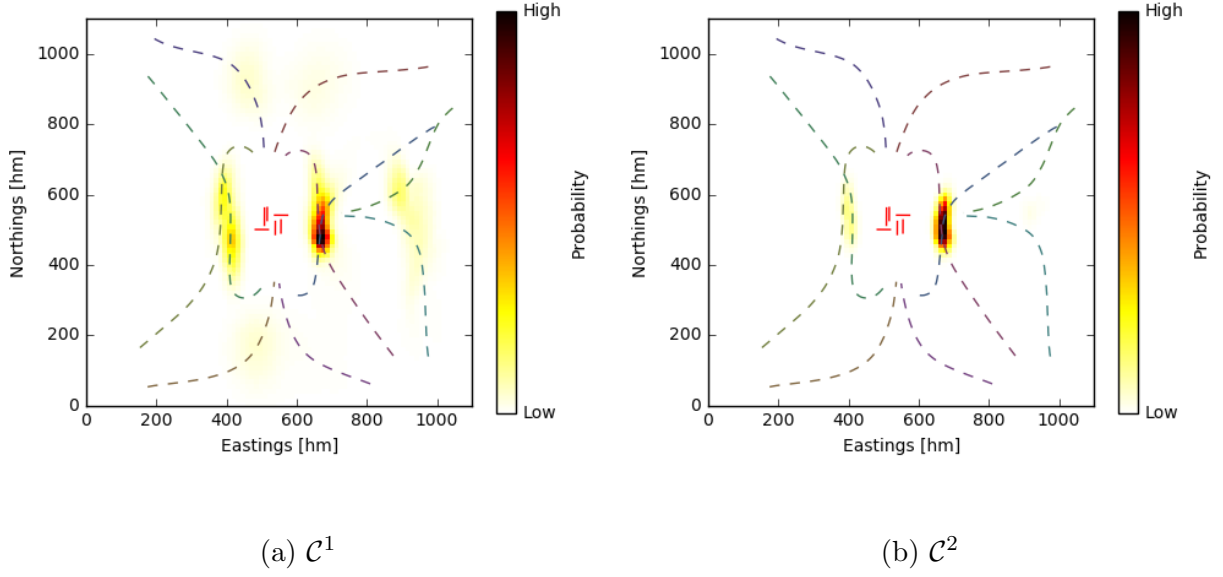


Figure 5: Complexity map of complexity measures \mathcal{C}^1 and \mathcal{C}^2 at an altitude of 450 metres.

data as seen from a different viewpoint. A maximum of 50 trajectories per cluster are shown to prevent visual clutter. Furthermore, the airport runways have been included and are visible as red lines. By applying the method as presented in section B, the flight corridors are generated for one standard deviation. The results are visible in Figures 2b and 3b, with different viewpoints of the trajectory datasets. Again, the airport runways are visible as red lines. Each cluster of trajectory data and the corresponding flight corridor are linked via colour coding.

The same models capturing the flight corridors at one standard deviation are presented at a constant altitude in Figure 4. As the aircraft in the trajectory data are on approach, the altitude changes from high (550 m) to low (400 m). The airport runways are visible as red lines, and the average trajectory for each cluster is visible as a dashed line. The colour coding is consistent with Figures 2 and 3. Due to the overlap of the flight corridors at an altitude of 450 metres, both to the east and west of the airport, a high complexity in terms of traffic structure is expected.

The complexity map of the complexity measures \mathcal{C}^1 and \mathcal{C}^2 at an altitude of 450 metres is shown in Figure 5. These complexity maps are normalised, displaying only the relative complexity. Each complexity measure represent the *background noise* from the lower measure, for example \mathcal{C}^1 is the lower complexity measure of \mathcal{C}^2 . This follows from the definition, where \mathcal{C}^1 represents the probability of *at least 1* flight corridor intersecting – this includes the values of \mathcal{C}^2 , which represents the probability of *at least 2* flight corridors intersecting.

Both Figure 5a and Figure 4c visualise the probabilities of each individual cluster, where the complexity map shows the probability density, and the flight corridor displays a contour (chosen via the standard deviation) of that density. Figure 5b shows the most important complexity measure for identification of multi-aircraft conflict situations, the probability of two or more corridors intersecting. This figure highlights high complexity in the regions to the east and west of the airport, corresponding with the situation sketched as seen in Figure 4c.

IV. Conclusion

This paper introduces an approach of modelling the flight corridors found in aircraft trajectory data as Gaussian processes. Results reveal that the Python package *Teetool* is capable of generating flight corridors from aircraft trajectory data – effectively capturing the dominant flows found in the data. These results have been visualised in both two and three dimensions, supporting the Traffic Flow Management (TFM) in improving the situational awareness.

Furthermore, these flight-corridors supported by a probabilistic model have been applied to generate a complexity map, identifying the areas that have a structural presence of high air-traffic complexity in terms of traffic structure. Such a data-driven approach enables the use of optimisation schemes in the re-design of existing procedures and the addition of new arrival/departure procedures.

A limitation of generating the flight corridors via the method presented in this paper is that the results are *very* dependent on the clustering method. For example, no information on the weather conditions are available for the trajectory data used in this paper, therefore no actual conclusions can be made about the intersecting flight corridors. This is because changing wind conditions will trigger the use of different approach routes. While this is not important for the purpose of this paper, which is to demonstrate the workings of the method, it is important when providing decision support to the TFM and specific conditions are evaluated.

Future work involves analysing aircraft trajectory data that include information on the weather conditions, providing a case-study suitable for a more extensive analysis. Future work also includes the implementation of an optimisation algorithm that is capable of optimising the structure of the existing procedures and of introducing new paths that avoid existing flight corridors.

Acknowledgements

The authors gratefully acknowledge the funding provided under research grant EP/L505067/1 from the Engineering and Physical Sciences Research Council (EPSRC) and Cuning Running Software Ltd. Supplemental material supporting this study are openly available from the University of Southampton repository at <https://doi.org/10.5258/S0T0N/D0071>.

References

- [1] M. Prandini, L. Piroddi, S. Puechmorel and S. L. Brazdilova. ‘Toward Air Traffic Complexity Assessment in New Generation Air Traffic Management Systems’. In: *IEEE Transactions on Intelligent Transportation Systems* 12.3 (2011), pp. 809–818. ISSN: 1524-9050. DOI: 10.1109/TITS.2011.2113175.
- [2] S. Grabbe, B. Sridhar and A. Mukherjee. ‘Clustering Days and Hours with Similar Airport Traffic and Weather Conditions’. In: *Journal of Aerospace Information Systems* 11.11 (2014), pp. 751–763. DOI: 10.2514/1.I010212.
- [3] M. Gariel, A. N. Srivastava and E. Feron. ‘Trajectory Clustering and an Application to Airspace Monitoring’. In: *IEEE Transactions on Intelligent Transportation Systems* 12.4 (2011), pp. 1511–1524. ISSN: 1524-9050. DOI: 10.1109/TITS.2011.2160628.
- [4] M. C. R. Murca, R. DeLaura, R. J. Hansman, R. Jordan, T. Reynolds and H. Balakrishnan. ‘Trajectory Clustering and Classification for Characterization of Air Traffic Flows’. In: AIAA Aviation 2016. American Institute of Aeronautics and Astronautics, 2016. DOI: 10.2514/6.2016-3760.
- [5] G. Vourous. *White paper: Data-Driven Aircraft Trajectory Prediction Exploratory Research*. Tech. rep. University of Piraeus, Piraeus, Greece, 2017, p. 23. URL: <http://dart-research.eu/2017/04/03/dart-white-paper/>.
- [6] C. Hurter, O. Ersoy and A. Telea. ‘Graph Bundling by Kernel Density Estimation’. In: *Computer Graphics Forum* 31.3pt1 (2012), pp. 865–874. ISSN: 1467-8659. DOI: 10.1111/j.1467-8659.2012.03079.x.
- [7] A. Marzuoli, C. Hurter and E. Feron. ‘Data visualization techniques for airspace flow modeling’. In: *Intelligent Data Understanding (CIDU), 2012 Conference on*. 2012, pp. 79–86. DOI: 10.1109/CIDU.2012.6382187.
- [8] C. Hurter, S. Conversy, D. Gianazza and A. Telea. ‘Interactive image-based information visualization for aircraft trajectory analysis’. In: *Transportation Research Part C: Emerging Technologies* 47, Part 2 (2014), pp. 207–227. ISSN: 0968-090X. DOI: 10.1016/j.trc.2014.03.005.

- [9] N. Elmqvist and P. Tsigas. ‘A Taxonomy of 3D Occlusion Management for Visualization’. In: *IEEE Transactions on Visualization and Computer Graphics* 14.5 (2008), pp. 1095–1109. ISSN: 1077-2626. DOI: 10.1109/TVCG.2008.59.
- [10] S. Buschmann, M. Trapp and J. Döllner. ‘Real-Time Animated Visualization of Massive Air-Traffic Trajectories’. In: *Cyberworlds (CW), 2014 International Conference on*. 2014, pp. 174–181. DOI: 10.1109/CW.2014.32.
- [11] E. Salaun, M. Gariel, A. E. Vela and E. Feron. ‘Aircraft Proximity Maps Based on Data-Driven Flow Modeling’. In: *Journal of Guidance, Control, and Dynamics* 35.2 (2012), pp. 563–577. ISSN: 0731-5090. DOI: 10.2514/1.53859.
- [12] W. J. Eerland, S. Box and A. Sóbester. ‘Modeling the Dispersion of Aircraft Trajectories Using Gaussian Processes’. In: *Journal of Guidance, Control, and Dynamics* 39.12 (2016), pp. 2661–2672. DOI: 10.2514/1.G000537. eprint: <http://eprints.soton.ac.uk/399818/>.
- [13] H. Yan, B. Yang, H. Yang and R. Wang. ‘Probabilistic Approach to Conformance Monitoring Using Gaussian Processes’. In: *Journal of Guidance, Control, and Dynamics* (2017). DOI: 10.2514/1.G002383.
- [14] T. G. Reynolds and R. J. Hansman. ‘Conformance monitoring approaches in current and future air traffic control environments’. In: *Proceedings. The 21st Digital Avionics Systems Conference*. Vol. 2. 2002, pp. 7C1–1–7C1–12. DOI: 10.1109/DASC.2002.1052922.
- [15] M. Prandini, V. Putta and J. Hu. ‘A probabilistic measure of air traffic complexity in 3-D airspace’. In: *International Journal of Adaptive Control and Signal Processing* 24.10 (2010), pp. 813–829. ISSN: 1099-1115. DOI: 10.1002/acs.1192.
- [16] I. V. Laudeman, S. Shelden, R. Branstrom and C. Brasil. *Dynamic density: An air traffic management metric*. Monograph. National Aeronautics and Space Administration Moffett Field, CA 94035 USA: Ames Research Center, 1998. URL: <https://ntrs.nasa.gov/search.jsp?R=19980210764>.
- [17] B. Sridhar, K. Sheth Sheth and S. Grabbe. ‘Airspace complexity and its application in air traffic management’. In: *2nd USA/Europe Air Traffic Management R&D Seminar*. 1998, pp. 1–6.
- [18] M. R. Jardin. ‘Analytical Relationships Between Conflict Counts and Air-Traffic Density’. In: *Journal of Guidance, Control, and Dynamics* 28.6 (2005), pp. 1150–1156. DOI: 10.2514/1.12758.

- [19] W. J. Eerland, S. Box, H. Fangohr and A. Sóbester. ‘Teetool – a probabilistic trajectory analysis tool’. In: *Journal of Open Research Software* 5.1 (2017), p. 6. DOI: 10.5334/jors.163. eprint: <https://eprints.soton.ac.uk/408279/>.
- [20] W. J. Eerland, S. Box, H. Fangohr and A. Sóbester. ‘An open-source, stochastic, six-degrees-of-freedom rocket flight simulator, with a probabilistic trajectory analysis approach’. In: *AIAA Modeling and Simulation Technologies Conference*. Grapevine, Texas, USA: American Institute of Aeronautics and Astronautics, 2017. DOI: 10.2514/6.2017-1556. eprint: <http://eprints.soton.ac.uk/403364/>.
- [21] W. J. Eerland, S. Box and A. Sóbester. ‘Cambridge Rocketry Simulator – A Stochastic Six-Degrees-of-Freedom Rocket Flight Simulator’. In: *Journal of Open Research Software* 5.1 (2017). DOI: 10.5334/jors.137. eprint: <http://eprints.soton.ac.uk/405278/>.
- [22] W. J. Eerland, S. Box, H. Fangohr and A. Sóbester. *Supplemental material – A Gaussian process based decision support tool for air-traffic management*. 2017. URL: <https://doi.org/10.5258/SOTON/D0071>.
- [23] C. B. Barber, D. P. Dobkin and H. Huhdanpaa. ‘The Quickhull algorithm for convex hulls’. In: *ACM Transactions on Mathematical Software* 22.4 (1996), pp. 469–483. DOI: 10.1145/235815.235821.