Trajectory Clustering, Modelling and Selection with the focus on Airspace Protection

Willem J. Eerland * and Simon Box †

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

February 5, 2016

Abstract

Take-off and landing are the periods of a flight where aircraft are most vulnerable to a ground based rocket attack by terrorists. While aircraft approach and depart from airports on pre-defined flight paths, there is a degree of uncertainty in the trajectory of each individual aircraft. Capturing and characterizing these deviations is important for accurate strategic planning for the defence of airports against terrorist attack. A methodology is demonstrated whereby approach and departure trajectories to a given airport are characterized statistically from historical data. It uses a two-step process of first clustering to extract the common trend, and then modelling uncertainty using Gaussian Processes (GPs). Furthermore it is shown that this approach can be used to either select probabilistic regions of airspace where trajectories are likely and - if required - can automatically generate a set of representative trajectories, or select key trajectories that are both likely and critically vulnerable. An evaluation of the methodology is demonstrated on an example data-set collected by the ground radar at an airport. The evaluation indicates that 99.8% of the calculated footprint underestimates less than 5% when replacing the original trajectory data with a set of representative trajectories.

1 Introduction

Extracting patterns from data is an active field for both research and industry, ranging from tracking traffic to making predictions on the financial market. It allows for objects to be clustered when following a similar trend. And by identifying the generic response, an attempt can be made to explain these reactions. When it comes to air traffic, the reason is well known. As an aircraft enters controlled airspace, such as near an airport, they follow the instructions of air traffic control, who guide them along pre-designated paths.

In the real-world, these pre-designated paths, also called flight paths, have more resemblance to corridors. For this reason, the current methodology on calculating noise contours around civil airports in Europe, uses several sub-tracks to model the dispersion along a single flight path [3]. When using the Integrated Noise Model (INM) [2] to calculate the noise contours, either the sub-tracks or dispersion will have to be supplied by the user. Nowadays, with the air traffic increasing in volume, the introduction of new technologies and procedures being developed under the names *NextGen* and *SESAR*, the dispersion along the flight paths is more susceptible to change than ever before. A step-by-step guide to obtain a number of representative trajectories from historical data (over any given time-frame or conditions), is able to reduce the computational load in any subsequent analysis, without losing integrity and with limited effort for the user.

While noise can literally keep people awake at night, it is nowhere near as vital as securing the infrastructure. More specifically, protecting aircraft against the threat from Rocket-Propelled Grenades (RPGs), that can hit a moving target up to a distance of 300 metres [1]. The threat is real, as actions in the past, such as the heightened security around Heathrow back in 2003 have shown [8]. This is one of the reasons governments, airports and airliners alike perform much strategic planning to defend aircraft from terrorists attacks. In the scenario where the computational budget is limited, possibly due to the multitude of scenarios to evaluate or a time restriction, evaluating all the trajectories is not always a viable option. Having a method to determine, in a robust statistical manner, where the aircraft are most likely to be, is the first step in efficiently identifying high-risk launch sites.

Focussing on the work done in the aerospace sector, there have already been great advances in automatically clustering of aircraft trajectories on a common flightpath. One clustering method re-samples

* Postgraduate researcher, Transportation Research Group, w.j.eerland@soton.ac.uk

† Lecturer, Transportation Research Group

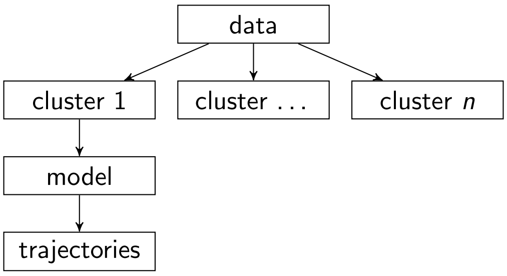


Figure 1: Procedure to obtain representative trajectories based on data.

the trajectories to fit in a vector of fixed size, after which the size of the vector is reduced using Principle Component Analysis (PCA). The data are then clustered using Density-Based Spatial Clustering of Applications (DBSCAN) [7], or *k-means* clustering [4]. Here the DBSCAN [6] allows for the filtering of outliers, resulting in a more robust clustering method compared to using *k-means* clustering. Another interesting method used to cluster the aircraft trajectories is based on Fourier coefficients [9]. The major difference here is that the trajectories are not merely re-sampled, but represented as Fourier-coefficients that effectively parametrises the aircraft trajectories. However, it should be said that the parametrisation is limited here to two spatial dimensions, ignoring the vertical variation in the flight-paths.

Automatic clustering (i.e. discovery of flight-paths) is an important first step, however, it holds little information about the level of dispersion of the trajectories within a given cluster, i.e. around the nominal flight-path. A method has been proposed by Salaun et al.[11] to calculate the probabilities by re-sampling, and fitting a univariate Gaussian in both the lateral and vertical direction perpendicular to the mean trajectory. Using this approach, they were successful in creating a tunnel through which a percentage of aircraft trajectories manoeuvre. More recently, a similar approach of modelling trajectories as Gaussian Processes (GPs) has been developed by the authors[5]. This allows the aircraft trajectories to be treated in a continuous manner and also models the covariance between the lateral and vertical direction.

In this paper, we present a step-by-step method to replace a large data-set of historical trajectories, with a number of representative trajectories that have none-the-less captured the dispersion in the original large data-set. The trajectory data are clustered, after which a probabilistic model is generated for each individual cluster. This probabilistic model is then used to generate weighted representative trajectories, each capturing a fraction of the whole cluster. The aim here is to reduce the computational cost of calculations that are done in sub-sequential steps. Such calculations can be either focussed on calculating noise footprints, or be aimed at performing a strategic analysis with the focus on safeguarding the airspace infrastructure. The latter is evaluated to demonstrate the effectiveness of the proposed method when generating a footprint on ground level.

This paper is organised as follows. In section 2, each step is explained, resulting in a guide how to transform the original trajectory data into a set of weighted representative trajectories. Next, in section 3, the methods are applied in a case study. This includes in an evaluation to compare the original trajectory data with a set of representative trajectories. The set of representative trajectories are generated in two ways, one takes into account the dispersion in lateral direction only, while the other also includes the vertical dispersion. Finally, in section 4, we conclude the paper with remarks and recommendations for future work.

2 Methods

This section reviews the techniques used to replace historical data with representative trajectories. The first part focuses on clustering the trajectory data by identifying groups of trajectories with similar flight-path. The second part focuses on estimating the dispersion of each individual cluster along the flight-path. The third and final part is aimed at creating weighted trajectories that represent the aircraft trajectories flying through the airspace. The complete procedure is seen in figure 1, where the data are first clustered, then modelled and finally representative trajectories are generated (leaving the possibility of selection).

2.1 Clustering the Trajectories

In order to cluster the trajectories based on a common flightpath, it is assumed that in the reference frame, the location of the runways are known and there is a human-in-the-loop, i.e. the process is not fully automated. The clustering technique takes the shape of the following three-step approach:

1. clustering trajectories as *approach* or *departure*
2. clustering the trajectories by run-way
3. re-sampling of trajectories
4. dimension reduction with PCA
5. DBSCAN clustering

The first step distinguishes the trajectory data between approach and departure. In the case where these meta-data are not included, the location of the airport is sufficient to identify whether a trajectory either ends at the airport (approach), or is leaving (departure). E.g. if the Euclidean distance between the airport and the first point of the trajectory is smaller than the Euclidean distance between the airport and the last point of the trajectory, it is likely to be a departure. The second step requires information about the location of the runway in a similar reference frame as the trajectories. While it reduces the generality, it does allow for a clean separation per runway.

Steps $3-5$ are mostly similar to the procedure as presented in Gariel et al.[7]. In step 3 the trajectories are re-sampled to a vector of a fixed size using uniform spacing based on the index number. Note that re-sampling over 30 steps result in a $[1 \times 90]$ vector as aircraft trajectories have 3 dimensions, and the dimensions are concatenated. Furthermore, the re-sampling is done per individual trajectory, while the next step, the reduction of the vector size, considers the entire data-set. Next, in step 4, these vectors are reduced in size using PCA. Here the principle components with the largest variance are kept, while the components with little variation are ignored. In this step it is assumed that the components with the largest variation are most important for the clustering. The clustering occurs in the final step, where the trajectories are clustered using DBSCAN. Here, the ϵ-neighbourhood parameter ϵ of the DBSCAN algorithm needs to be set on a case-by-case basis, a smaller ϵ will result in more clusters with fewer trajectories, whereas with a larger ϵ results in less clusters with more trajectories. In the final step all clusters with less than a user-specified number of trajectories will be ignored, effectively removing the outliers.

2.2 Modelling the Spatial Distribution

In this section a short overview of the modelling technique is provided. For a complete description, please see Eerland and Box[5]. Essentially, there are two steps in modelling the spatial distribution:

- normalising the trajectory data (per cluster, see previous section)
- learn model parameters via maximum likelihood estimation

The first step assures that the dimensions are of the same scale. In particular for aircraft trajectories this is an important step, as usually the distance covered horizontally is much larger than the distance travelled vertically. When estimating the parameters, this can lead to computational problems, as such, the trajectory data are normalised such that each dimension fits on a $[0, 1]$ range. This transformation can be reverted once the model has been created. Furthermore, normalised time τ is introduced to align all points of the trajectories at the start at the end. The start is set to be $\tau = 0$ and the end is $\tau = 1$, where the points in between are set proportionally. E.g. if the first point occurs at 0 seconds and the last point occurs at 20 seconds, the point at 12 seconds will have the normalised time $\tau = 12/20 = 0.6$. As such, each point in the trajectory is described as a 3×1 vector $\mathbf{y}(\tau)$, holding eastings, northings and altitude at the normalised time τ.

In the second step the model parameters are estimated. Here the parameters consist of the mean function $\mathbf{m}(\tau)$, covariance kernel $\mathbf{k}(\tau, \tau')$ and noise precision term β. These parameters capture the underlying function $\mathbf{y}(\tau)$ according to the following relation:

$$\mathbf{y}(\tau) \sim \mathcal{GP}(\mathbf{m}(\tau), \mathbf{k}(\tau, \tau')) \tag{1}$$

where

$$\mathbf{m}(\tau) = \mathbb{E}[\mathbf{y}] = \boldsymbol{\phi}(\tau)\boldsymbol{\mu} \tag{2}$$

$$\mathbf{k}(\tau, \tau') = \mathbb{E}[(\mathbf{y}(\tau) - \mathbf{m}(\tau))(\mathbf{y}(\tau') - \mathbf{m}(\tau'))] = \boldsymbol{\phi}(\tau)\boldsymbol{\Sigma}\boldsymbol{\phi}^{\intercal}(\tau') + \beta^{-1}\mathbf{I} \tag{3}$$

In these equations the mean function $\mathbf{m}(\tau)$ and the covariance kernel $\mathbf{k}(\tau, \tau')$ are captured using J basis functions. Here the discrete number of parameters captured in the $3J \times 1$ vector $\boldsymbol{\mu}$ and $3J \times 3J$ matrix $\boldsymbol{\Sigma}$, are converted from the discrete domain to the continuous domain via the block-diagonal $3 \times J$ matrix

$\boldsymbol{\phi}(\tau)$. The basis functions $\boldsymbol{\phi}(\tau)$ consists of 3 blocks, corresponding with the number of dimensions found in aircraft trajectories.

Next, for the estimation the Expectation-Maximisation (EM) algorithm is applied, this deals with the chicken-egg paradigm. More specifically, β is needed to estimate $\mathbf{m}(\tau)$ and $\mathbf{k}(\tau, \tau')$, and $\mathbf{m}(\tau)$ and $\mathbf{k}(\tau, \tau')$ are needed to estimate β. Basically, each individual trajectory is captured in the model described by the couple $\mathbf{m}(\tau)$ and $\mathbf{k}(\tau, \tau')$, however not perfectly, thus the remaining error is captured in β. And by doing so, maximizing the likelihood of these three terms using EM, prevents the probabilistic model to over-fit on the data (assuming the remaining error is Gaussian distributed).

Initially $\boldsymbol{\mu}$ (a $3J \times 1$ vector) is assumed $\mathbf{0}$, and $\boldsymbol{\Sigma}$ (a $3J \times 3J$ matrix) is assumed $a\mathbf{I}$, where $\mathbf{I}$ is the identity matrix and a an arbitrarily large number. This represents a *not very informative* prior and reflects the concept that no initial knowledge is available. The noise precision term β can be set high (e.g. in the order of magnitude of 10^3), to reflect that the measurements are exact and the remaining error low.

Using these initial parameters, the expected parameter values $\mathbf{w}$ (the E-step of the EM algorithm) are:

$$\mathbb{E}[\mathbf{w}_n] = \mathbf{S}_n(\beta \boldsymbol{\phi}_n^\intercal \mathbf{y}_n + \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}) \tag{4}$$

$$\mathbb{E}[\mathbf{w}_n \mathbf{w}_n^\intercal] = \mathbf{S}_n + \mathbb{E}[\mathbf{w}_n]\mathbb{E}[\mathbf{w}_n^\intercal] \tag{5}$$

where

$$\mathbf{S}_n^{-1} = \boldsymbol{\Sigma}^{-1} + \beta \boldsymbol{\phi}_n^\intercal \boldsymbol{\phi}_n \tag{6}$$

In these equations n represents the individual trajectory, and N equates the total number of trajectories found in the cluster.

Next, the likelihood is maximized with respect to the model parameters (the M-step of the EM algorithm) using:

$$\hat{\boldsymbol{\mu}} = \frac{1}{N}\sum_{n=1}^{N}\{\mathbb{E}[\mathbf{w}_n]\} \tag{7}$$

$$\hat{\boldsymbol{\Sigma}} = \frac{1}{N}\sum_{n=1}^{N}\{\mathbb{E}[\mathbf{w}_n\mathbf{w}_n^\intercal] - 2\mathbb{E}[\mathbf{w}_n^\intercal]\boldsymbol{\mu} + \boldsymbol{\mu}\boldsymbol{\mu}^\intercal\} \tag{8}$$

$$\frac{1}{\hat{\beta}} = \frac{1}{3M^*}\sum_{n=1}^{N}\{\mathbf{y}_n^\intercal\mathbf{y}_n - 2\mathbf{y}_n^\intercal(\boldsymbol{\phi}_n\mathbb{E}[\mathbf{w}_n]) + \mathrm{Tr}(\boldsymbol{\phi}_n^\intercal\boldsymbol{\phi}_n\mathbb{E}[\mathbf{w}_n\mathbf{w}_n^\intercal])\} \tag{9}$$

where the hat seen in $\hat{\boldsymbol{\mu}}$, $\hat{\boldsymbol{\Sigma}}$ and $\hat{\beta}$ signifies an approximation. These two steps in the EM algorithm are repeated until the likelihood is converged, where the negative log-likelihood itself can be evaluated using:

$$\begin{aligned} -\ln\mathcal{L} = -\frac{3M^*}{2}\ln(\beta) + \frac{\beta}{2}\sum_{n=1}^{N}\{\mathbf{y}_n^\intercal\mathbf{y}_n - 2\mathbf{y}_n^\intercal(\boldsymbol{\phi}_n\mathbf{w}_n) + \mathrm{Tr}(\boldsymbol{\phi}_n^\intercal\boldsymbol{\phi}_n\mathbf{w}_n\mathbf{w}_n^\intercal)\} \\ + \frac{N}{2}\ln(|\boldsymbol{\Sigma}|) + \frac{1}{2}\sum_{n=1}^{N}\{\mathrm{Tr}(\boldsymbol{\Sigma}^{-1}(\mathbf{w}_n\mathbf{w}_n^\intercal - 2\mathbf{w}_n^\intercal\boldsymbol{\mu} + \boldsymbol{\mu}\boldsymbol{\mu}^\intercal))\} \end{aligned} \tag{10}$$

where

$$M^* = \sum_{n=1}^{N}\{M_n\} \tag{11}$$

and M_n represents the total number of points in $\mathbf{y}_n$, thus M^* embodies the total number of points in the entire cluster.

The difference between two sequential log-likelihood evaluations is used as a stopping criteria, at this point the model approximation is assumed sufficient. Due to the nature of the EM algorithm, it will always be considered an approximation.

The approximated parameters can now be interested in the model, seen in equation (1), to estimate the probabilistic model at any τ in the domain $\tau = [0, 1]$. This model allows itself to be expressed in a multivariate Gaussian distribution as a function of τ, which will be used to generate *weighted* representative trajectories in the next section.

2.3 Generating representative trajectories

The previous section described how to estimate the probabilistic model. This section provides a method to convert this model to *weighted* trajectories that represent the entire cluster.

For a 3-dimensional vector $\mathbf{y}(\tau)$, the multivariate Gaussian distribution takes the form:

$$\mathcal{N}(\mathbf{m}, \mathbf{k}) = \frac{1}{\sqrt{(2\pi)^{3/2}}} \frac{1}{|\mathbf{k}|^{1/2}} \exp\left\{-\frac{1}{2}\mathrm{MD}(\mathbf{y}, \mathbf{m}, \mathbf{k})\right\} \tag{12}$$

where MD represents the Mahalanobis distance:

$$\mathrm{MD}(\mathbf{y}, \mathbf{m}, \mathbf{k})) = (\mathbf{y} - \mathbf{m})^\intercal \mathbf{k}^{-1} (\mathbf{y} - \mathbf{m}) \tag{13}$$

And note that the dependence on τ has been dropped for readability.

Furthermore, at a constant Mahalanobis distance, this equation takes the form of an ellipsoid described by $|\mathbf{k}|^{-1/2}$, centred at $\mathbf{m}$. In this scenario the axes of the covariance ellipse are given by the eigenvectors $\mathbf{r}$ of the covariance kernel $\mathbf{k}$. The corresponding lengths, for an ellipse with unit Mahalanobis radius, are given by the square roots of the corresponding eigenvalues λ. Both can be found using the eigenvalue decomposition of the matrix $\mathbf{k}$.

$$\mathbf{R}\mathbf{\Lambda}\mathbf{R}^{-1} = \mathbf{k} \tag{14}$$

where

$$\mathbf{R} = [\mathbf{r}_1\ \mathbf{r}_2\ \mathbf{r}_3] \tag{15}$$

and

$$\mathbf{\Lambda} = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \tag{16}$$

In short, the shape of the ellipsoid is described by the eigenvalues found in $\mathbf{\Lambda}$, where $\mathbf{m}$ and $\mathbf{R}$ are merely a translation and rotation respectively. The equation for the ellipsoid is given by:

$$\frac{y_1^2}{\lambda_1} + \frac{y_2^2}{\lambda_2} + \frac{y_3^2}{\lambda_3} = 1 \tag{17}$$

The plane perpendicular to the mean function at time τ, can be described with a unit normal vector $\mathbf{n}$. As $\mathbf{m}(\tau)$ is continuous, this unit normal vector $\mathbf{n}$ can be both derived, or calculated numerically:

$$\mathbf{n} = \frac{\mathbf{m}(\tau + d\tau/2) - \mathbf{m}(\tau - d\tau/2)}{d\tau} \tag{18}$$

where $d\tau$ is an arbitrarily small number. The mathematics required to calculate the intersection between the plane and ellipsoid is given in Klein [10]. By doing so, the *two-dimensional* ellipse can be evaluated at any angle. However, it's important to note that the plane generated at time τ intersects multiple ellipsoids, it's therefore necessary to evaluate all those that intersect and store the point corresponding with the largest deviation from the centre point (provided by $\mathbf{m}(\tau)$).

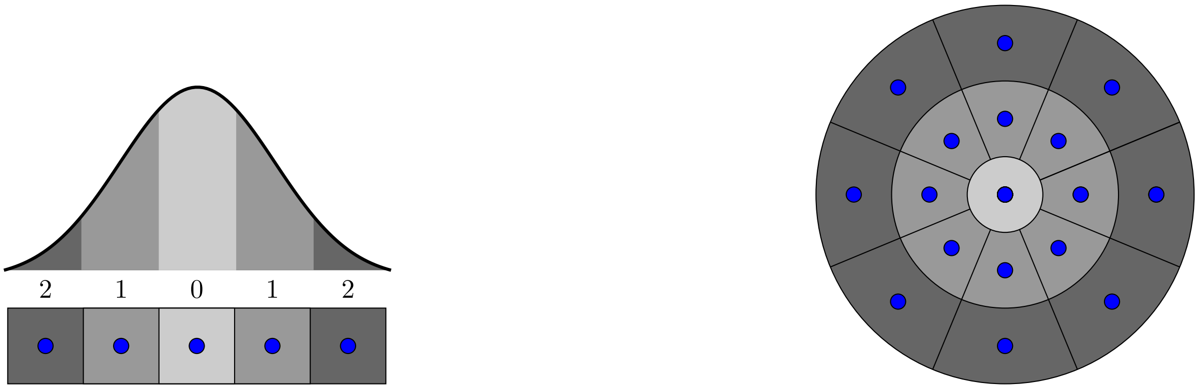
To obtain a representative trajectory, the ellipse at any specific angle, which is a single point in a three-dimensional space, can be evaluated over $\tau = [0, 1]$ in any number of steps. In this paper 100 steps are used. Thus the combination of a constant Mahalanobis distance (representing a confidence interval, to be discussed next) and a given angle provides one representative trajectory. However, when it comes to selecting the angles to obtain a selection of representative trajectories, there are an infinite number of options. In this paper two options are compared. In the handbook on generating noise contours [3], only the dispersion in lateral direction is taken into account. Here the cross-section containing the artificial trajectories appears like figure 2(a), where the Gaussian distribution is included as a reference. In this figure, 5 trajectories are shown to capture a given percentage (the area under the curve). This corresponds with the Cumulative Distribution Function (CDF), which is equal to chi-square with 1 degree of freedom. The resulting weight per trajectory is shown in table 1. The area under the curve described by the Gaussian corresponds with a confidence interval (thus capturing a certain percentage of the complete data) - and this percentage is divided over multiple trajectories due to symmetry. E.g. for the case of lateral dispersion only, in the range $[0.5^2, 1.5^2]$, the total percentage 48.3461% is divided over two trajectories centred at a standard deviation of one ($N = 2$), resulting in 24.1730% per trajectory. When including the vertical dispersion, the model show more similarity to figure 2(b). As it now encompasses two dimensions, the chi-square with 2 degrees of freedom is used. Combined with angles at various ranges, there are 17 representative trajectories. The weight per trajectory is shown in table 2. In the evaluations seen further on in this paper, these percentages will be multiplied with the total traffic to obtain representative number of trajectories. I.e. the percentages shown here, are the weights used in the calculations.

Table 1: Weights generated using the chi-square distribution table (1 degree of freedom).

range	**total percentage captured**	**percentage per trajectory**
$[0.0^2, 0.5^2]$	38.29%	38.29% *(N = 1)*
$[0.5^2, 1.5^2]$	48.35%	24.17% *(N = 2)*
$[1.5^2, 2.5^2]$	12.12%	6.06% *(N = 2)*
$[0.0^2, 2.5^2]$	98.76%	98.76% *(N = 5)*

Table 2: Weights generated using the chi-square distribution table (2 degrees of freedom).

range	**total percentage captured**	**percentage per trajectory**
$[0.0^2, 0.5^2]$	11.75%	11.75% *(N = 1)*
$[0.5^2, 1.5^2]$	55.78%	6.97% *(N = 8)*
$[1.5^2, 2.5^2]$	28.07%	3.51% *(N = 8)*
$[0.0^2, 2.5^2]$	95.61%	95.61% *(N = 17)*



(a) Cross-section of the representative trajectories (indicated by blue dots), when only taking the lateral dispersion into account. At the top is a schematic representation of the Gaussian distribution, placed as a reference.

(b) Cross-section of the representative trajectories (indicated by blue dots) when taking both the lateral and vertical dispersion into account.

Figure 2: Two options to obtain representative trajectories.

Table 3: Number of trajectories found in each cluster.

cluster	number of trajectories
1	267
2	194
3	178
4	176

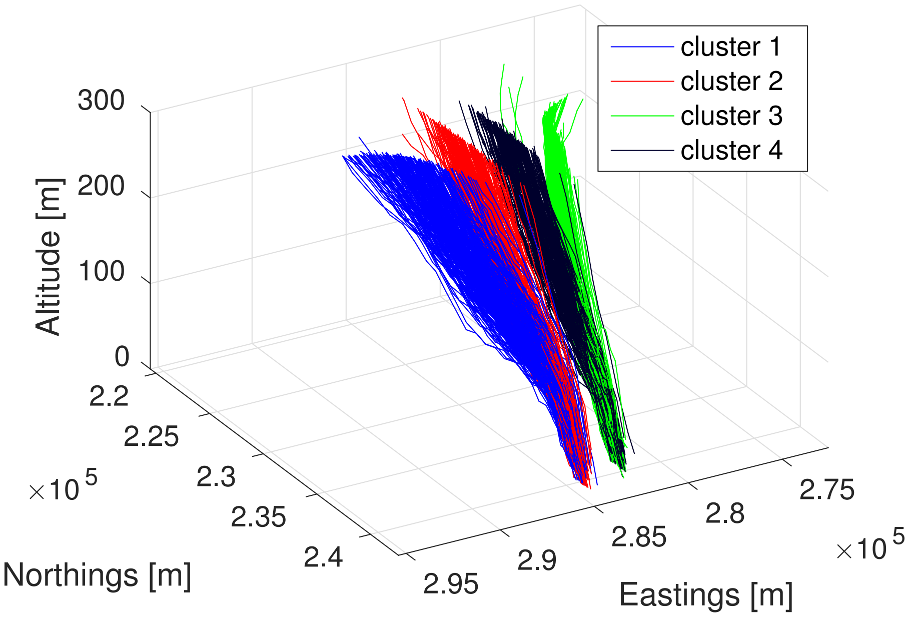


Figure 3: Three-dimensional view of all the clustered original trajectory data.

3 Results

This section applies the step-by-step guide as described in section 2 on a case-study. The case-study here is based on aircraft trajectories near an airport as measured by ground radar. The first step is to cluster the trajectories, after which the individual clusters of trajectories (sorted per common flight-path) are modelled, and in the final step a sub-set of representative trajectories is generated. For the evaluation a 100×100 grid is placed near the airport at a ground level. For each of these grid-points the percentage of the total number of trajectories in the (post-clustered) data that come within 300 metres (as measured from the centre) is calculated. Under the assumption that each trajectory is equally likely to occur, it can be said that a person standing in the area with the highest percentage is more likely to 'spot' an aircraft within a 300 metres radius. Where in the areas with 0% there is no chance at all. This relates back to the introduction where it was pointed out that there is evidence that terrorists have access to RPGs, and that these weapons are effective up to a range of 300 metres against a moving target, in this case an aircraft.

The trajectory data are clustered according to the technique outlined in section 2.1, where the result is visible in figure 3. For the analysis in this section, only the *departure* trajectories are included to keep the amount of trajectories manageable and the eventual evaluation of a reasonable scale. The number of trajectories found in each cluster is shown in table 3. The ϵ-neighbourhood parameter ϵ of the DBSCAN algorithm is set to 0.20 and the minimum number of trajectories in each cluster is 25, causing just over 19% (193 trajectories) to be considered outliers. These outliers are not shown. Furthermore, as the ultimate goal is to create a footprint based on the distance of 300 metres, only the parts of the trajectories below an altitude of 500 metres are used in the analysis.

Based on the method described in section 2.2, a model is generated for each cluster. The basis functions $\boldsymbol{\phi}(\tau)$ in this paper consist of 17 radial basis functions, uniformly distributed over the interval $\tau = [0, 1]$. And when including the bias function, the total number of basis function J is 18. The convergence criteria is set such that the difference between two sequential iterations, and evaluation of the log-likelihood, is at most 10^{-3}. The result is shown in figure 4, where the volume represents the range corresponding with a standard deviation of two ($\sigma = 2$).

In section 2.3 two approaches to generate representative trajectories were described. One automatically generates the trajectories while only taking into account the lateral dispersion. The cross-section

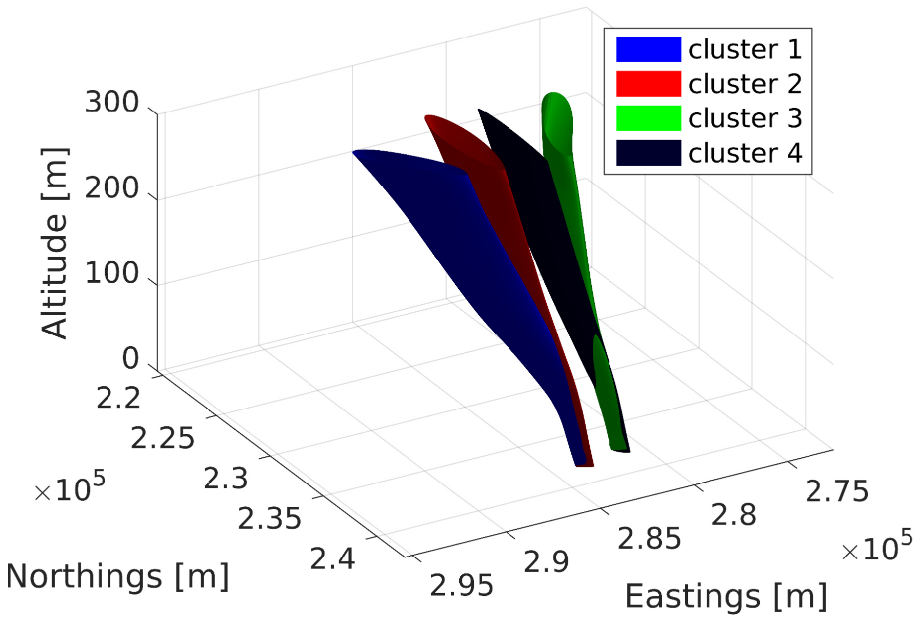


Figure 4: Three-dimensional view of all models - where the volume corresponds with a standard deviation of two.

of this 'flat' version is seen in figure 2(a), and the resulting top-view is visible in figure 6. This modelling approach is currently being used in calculating noise footprints around airports, as described in the official publication [3]. The other approach takes into account both the lateral and vertical dispersion simultaneously. The cross-section of this 'round' version is seen in figure 2(b), and the resulting top-view is visible in figure 7.

The difference by taking into account the vertical dispersion becomes apparent when examining the side-views seen in figures 8 to 10, where the original trajectory data and the two version of representative trajectories are displayed. It clearly shows that modelling the vertical dispersion, actually *represents* the vertical dispersion as seen in the original data.

For the evaluation a 100 × 100 grid is placed near the airport at a ground level. For each of these grid-points the percentage of the 815 trajectories in the data that come within 300 metres is calculated. This simulates a simplistic approach to calculate from which locations the aircraft are vulnerable to an attack using RPGs, without taking into account any mapping nor specific weapon characteristics. For comparison, the analysis is performed on the original trajectory and the two versions of the representative trajectories.

The resulting footprint calculated for the original trajectory data is visible in figure 11, where the range of percentages has been set to visualise the sweep of low percentages towards the end. This sweep is missing from figure 12, where only the lateral dispersion is taken into account. While the visible roughness is a direct effect of only using 5 trajectories to represent each cluster, the missing sweep is a direct result of missing trajectories at a lower altitude. As such, this sweep is visible in figure 13. Here there are representative trajectories at a lower altitude, even at a distance further away from the airport.

The quantitative results comparing the methods to generate representative trajectories have been gathered in table 4. While the complete evaluated grid is 100 × 100, only the grid-points that have a non-zero value are analysed. For this reason the *lateral dispersion only* has 2458 grid-points and the *lateral and vertical dispersion* has 2586, as the latter extends further beyond the influence of the original trajectory data. It demonstrates that including the vertical dispersion is beneficial for both spectrum of the extremes. Furthermore, the balance between underestimation and overestimation is included. Finally, the percentage of grid-points being under- and overestimated by more than 5% is reported. This shows that less than 0.2% of the *active area* (i.e. non-zero grid-points) is underestimated by more than 5%. From a safety perspective, this is exactly the value we want as low as possible. Interesting is also that 14.08% is overestimated more than 5% for the lateral and vertical dispersion, while 24.74% is overestimated when only modelling the lateral dispersion. This can be interpreted as the vertical dispersion expanding the area of influence, yet reducing the intensity of the peaks.

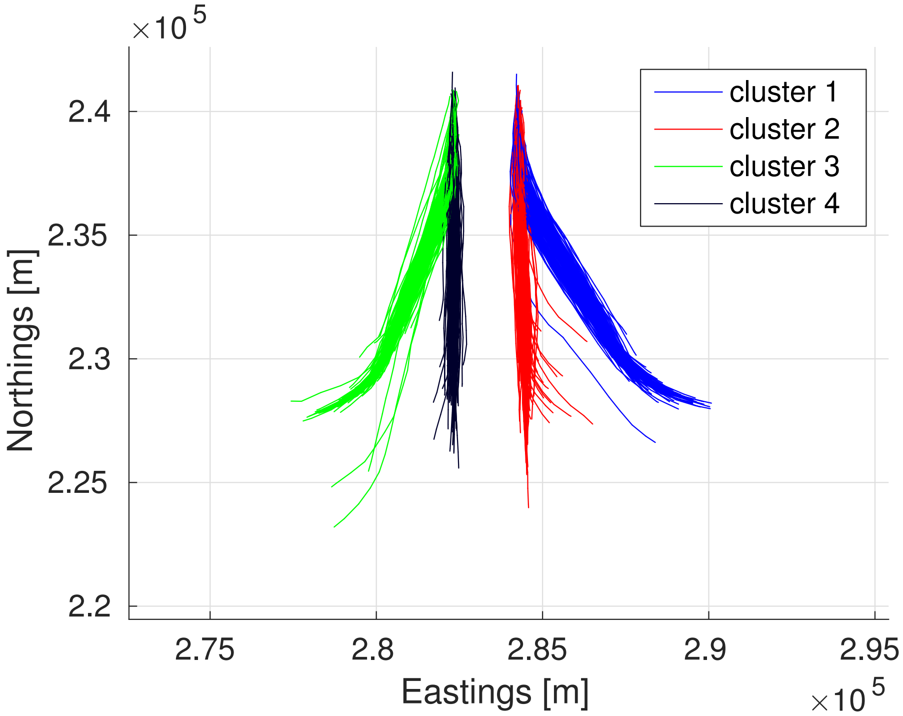


Figure 5: Top-view of all the clustered trajectories.

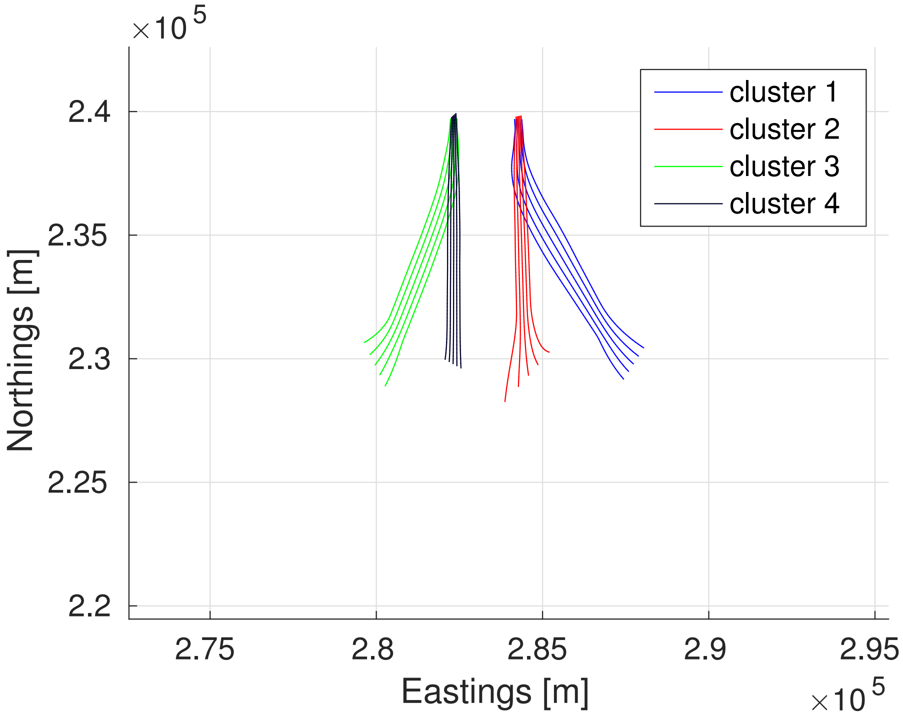


Figure 6: Top-view of the representative trajectories (the lateral direction only).

Table 4: Comparing the footprints of the representative trajectories with the original trajectory data.

	lateral dispersion only	**lateral and vertical dispersion**
minimum deviation	−23.80%	−8.30%
maximum deviation	24.53%	17.52%
grid-points underestimated	39.34% (967 out of 2458)	28.15% (728 out of 2586)
grid-points overestimated	60.66% (1491 out of 2458)	71.85% (1858 out of 2586)
grid-points underestimated (> 5%)	1.42% (35 out of 2458)	0.19% (5 out of 2586)
grid-points overestimated (> 5%)	24.74% (608 out of 2458)	14.08% (364 out of 2586)

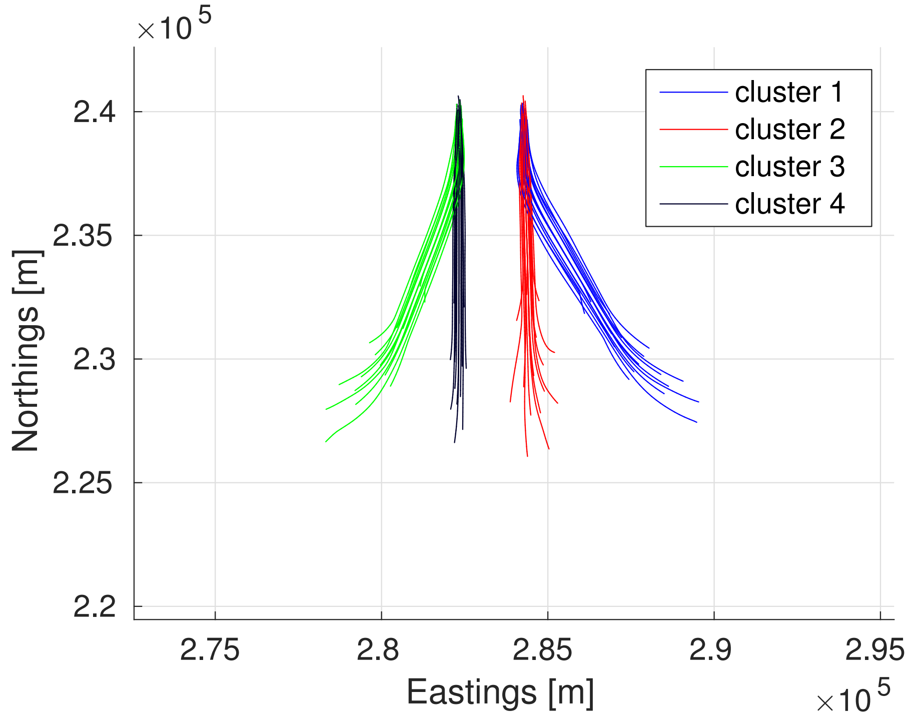


Figure 7: Top-view of the representative trajectories (lateral and vertical direction).

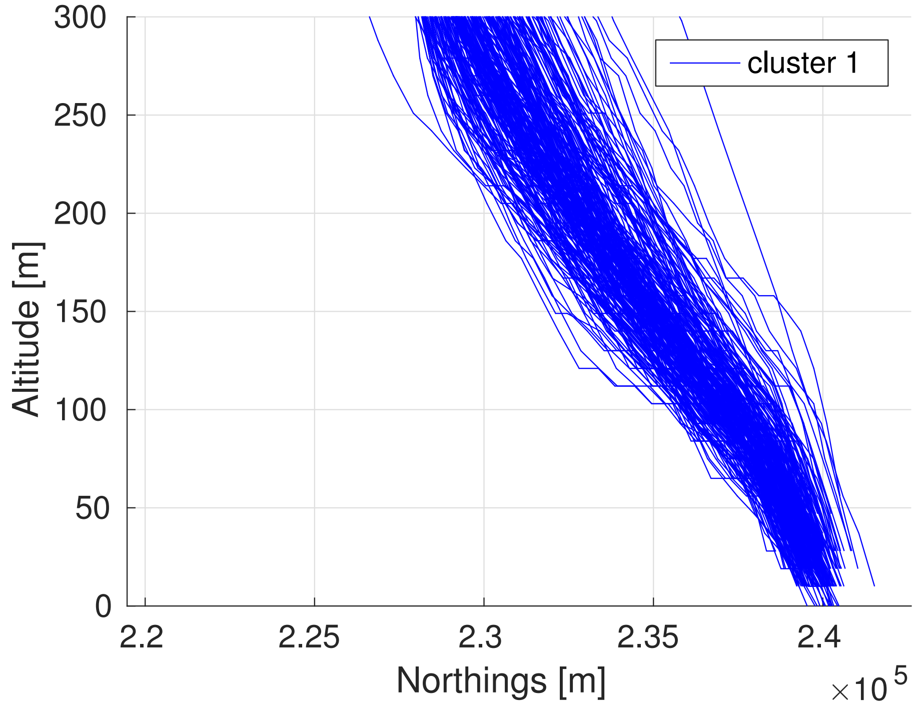


Figure 8: Side-view of the largest cluster of the original trajectory data.

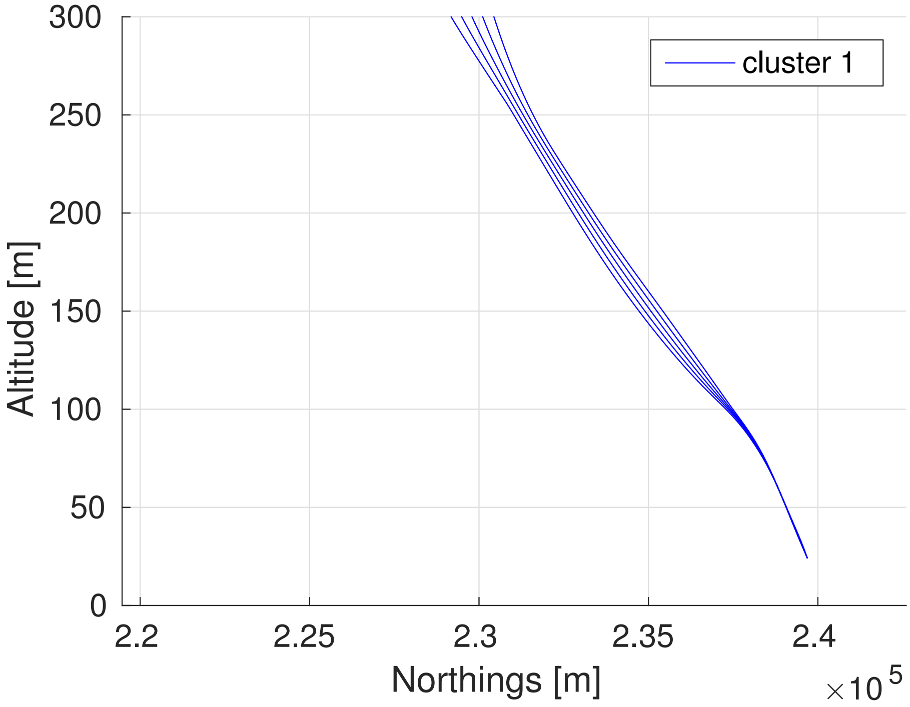


Figure 9: Side-view of the representative trajectories (lateral direction only).

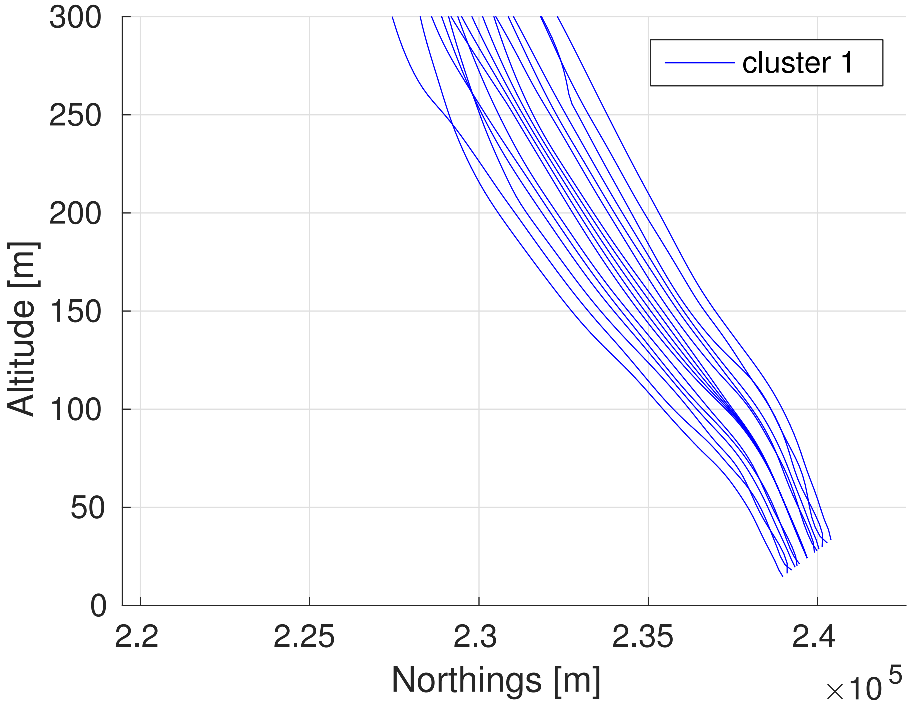


Figure 10: Side-view of the representative trajectories (lateral and vertical direction).

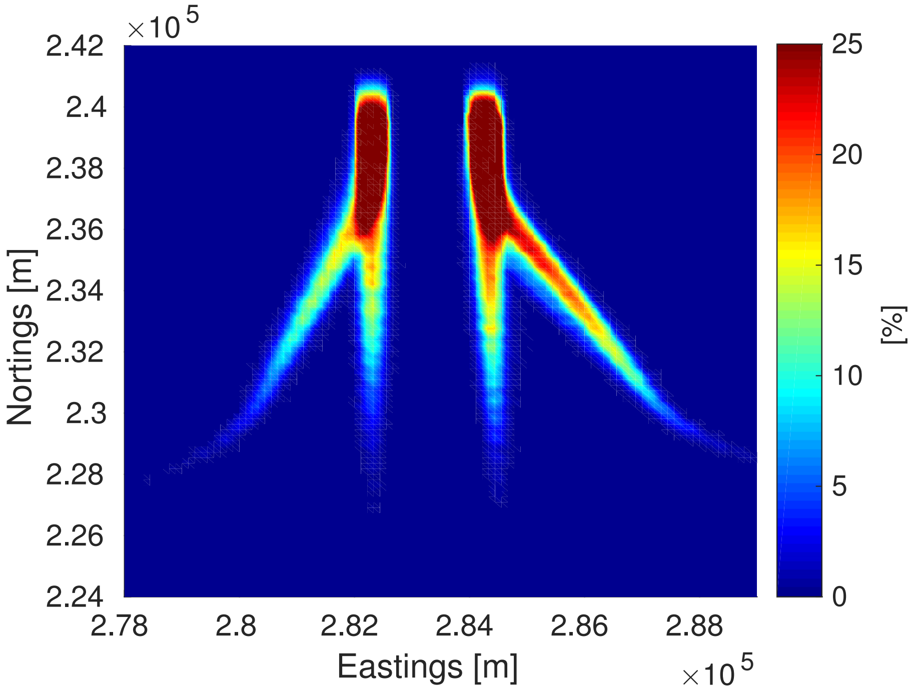


Figure 11: Footprint generated using the original trajectory data.

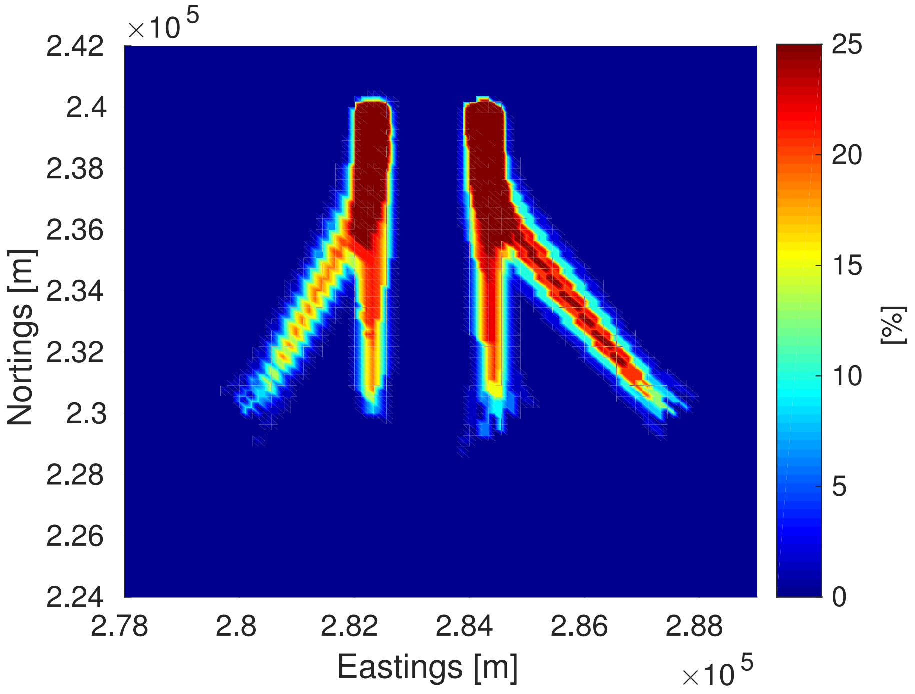


Figure 12: Footprint generated using representative trajectories (lateral dispersion only).

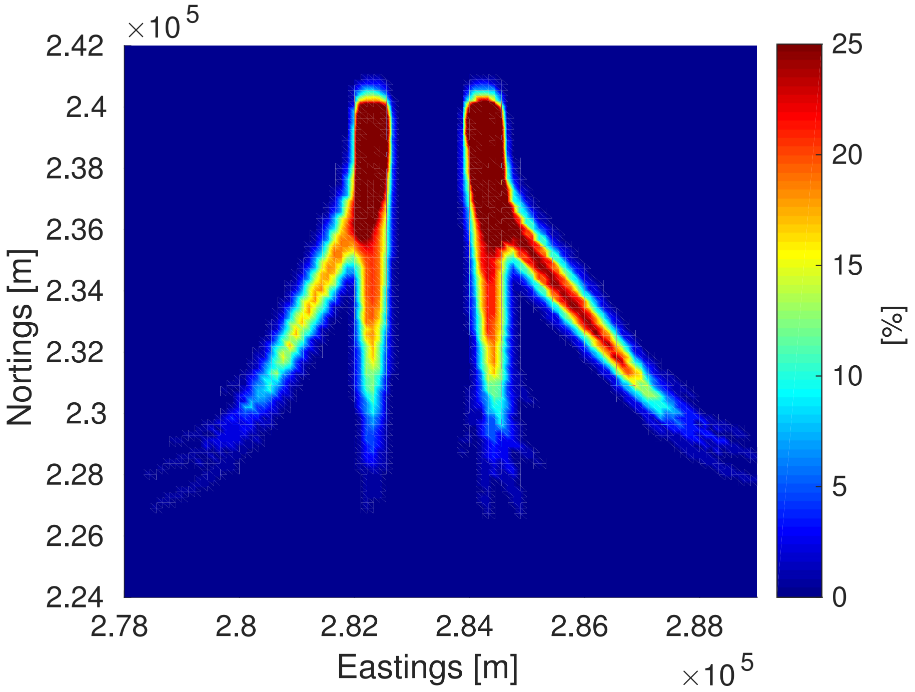


Figure 13: Footprint generated using representative trajectories (lateral and vertical dispersion).

4 Conclusions

The step-by-step method presented in this paper has shown the potential to capture the dispersion in a large historical data-set of trajectories using representative trajectories. Less than 0.2% of the evaluations are underestimated by more than 5%, demonstrating its usefulness when performing a strategic analysis with the focus on safeguarding the airspace infrastructure. This while the number of trajectories has been diminished to 8.34% of the original situation (68 versus 815 trajectories). The strength of this technique is that the original data-set of trajectories can be increased, while the computational cost of the sequential calculations remain constant, and at the same time, retains its integrity.

Furthermore, while not explicitly demonstrated in this paper, it's possible to select key trajectories based on their location. E.g. when considering a *hit* or *miss* scenario concerning aircraft trajectories, a 'bottom belly' of trajectories can be selected. The reasoning here would be, if the trajectories at a high altitude can be hit, it is a sure hit for trajectories at a lower altitude. This strength is directly connected to the ability to construct representative trajectories at any angle. The resulting representative trajectory can then matched (based on similarity) with a trajectory found in the historical data.

Finally, while the generation of a noise footprint has been mentioned on several occasions, and the results indicate that the extreme peaks are reduced and the lower levels expanded, no analysis has been performed concerning noise footprints. Such an analysis would have to include a noise model, as the noise as perceived from the ground will have a logarithmic effect. Nevertheless, it would be interesting to see how the inclusion of vertical dispersion influences the results concerning a noise footprint.

Acknowledgements

The authors gratefully acknowledge the funding provided under research grant EP/L505067/1 and industry sponsor Cunning Running Software Ltd.

References

[1] Aviation Today. `http://www.aviationtoday.com/av/military/Beating-Ballistic-Threats_26287.html`, Oct. 2008. Accessed: 2015-11-12.

[2] E. R. Boeker, E. Dinges, B. He, G. Fleming, C. Roof, P. Gerbi, A. Rapoza, and J. Hemann. Integrated noise model (inm) version 7.0 technical manual. *Washington, DC, Federal Aviation Administration, Office of Environment and Energy*, 2008.

[3] ECAC CEAC. ECAC . CEAC Doc 29 3rd Edition Report on Standard Method of Computing Noise Contours around Civil Airports Volume 1 : Applications Guide, 2005.

[4] A. Eckstein. Automated flight track taxonomy for measuring benefits from performance based navigation. In *Proceedings of the 2009 Integrated Communications, Navigation and Surveillance Conference, ICNS 2009*, pages 1–12. IEEE, 2009.

[5] W. Eerland and S. Box. Modelling the dispersion of trajectories using gaussian processes. *(to be published)*, 2015. (to be published).

[6] M. Ester, H. P. Kriegel, J. Sander, and X. Xu. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *Second International Conference on Knowledge Discovery and Data Mining*, pages 226–231, 1996.

[7] M. Gariel, A. N. Srivastava, and E. Feron. Trajectory clustering and an application to airspace monitoring. *IEEE Transactions on Intelligent Transportation Systems*, 12(4):1511–1524, 2011.

[8] N. Hopkins, R. Norton-Taylor, and M. White. UK on missile terror alert. `http://www.theguardian.com/uk/2003/feb/12/terrorism.world1`, Feb. 2003. Accessed: 2015-11-26.

[9] R. A. Jr and C. H. Q. Forster. Analysis of Aircraft Trajectories Using Fourier Descriptors and Kernel Density Estimation. In *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on*, pages 1441–1446. IEEE, 2012.

[10] P. P. Klein. On the ellipsoid and plane intersection equation. *Applied Mathematics*, 3(11):1634, 2012.

[11] E. Salaun, M. Gariel, A. E. Vela, and E. Feron. Aircraft proximity maps based on data-driven flow modeling. *Journal of Guidance, Control, and Dynamics*, 35(2):563–577, 2012.