Trajectory Clustering and an Application to Airspace Monitoring

Maxime Gariel* Ashok N. Srivastava † Eric Feron ‡

This paper presents a framework aimed at monitoring the behavior of aircraft in a given airspace. Trajectories that constitutes typical operations are determined and learned using data driven methods. Standard procedures are used by air traffic controllers (ATC) to guide aircraft, ensure the safety of the airspace, and to maximize the runway occupancy. Even though standard procedures are used by ATC, the control of the aircraft remains with the pilots, leading to a large variability in the flight patterns observed. Two methods to identify typical operations and their variability from recorded radar tracks are presented. This knowledge base is then used to monitor the conformance of current operations against operations previously identified as typical. A tool called AirTrajectoryMiner is presented, aiming at monitoring the instantaneous health of the airspace, in real time. The airspace is "healthy" when all aircraft are flying according to the typical operations. A measure of complexity is introduced, measuring the conformance of current flight to typical flight patterns. When an aircraft does not conform, the complexity increases as more attention from ATC is required to ensure a safe separation between aircraft.

*Ph.D Candidate, Georgia Institute of Technology, 270 Ferst Drive, Atlanta, GA, 30332.

†Principal Investigator, Integrated Vehicle Health Management; Group Lead, Intelligent Data Understanding, NASA Ames Research Center, Moffett Field, CA

‡Dutton/Ducoffe professor of Aerospace Engineering, Georgia Institute of Technology, 270 Ferst Drive, Atlanta, GA, 30332.

I. Introduction

To address the challenges of increase in air traffic volume, new technologies and procedures are being developed in the context of NextGen[1] in the US and SESAR[2] in Europe. Automation is a key element, necessary to achieve the goals set by those programs. New procedures involving more accurate navigation are predicted to increase the capacity of the airspace. Analyzing trajectory records is a key element to assess the performances and the accuracy of new concepts of operations. Automated tools are needed to process the large amount of daily flights and corresponding records. This work presents two methods to cluster trajectories and identify flights that followed identical air routes. The first method is based on the identification of way-points in the trajectories, and the second method is based on a principal components analysis of re-sampled trajectories. Operations in the terminal area are managed by Air Traffic Controllers (ATC) and are not part of the flight plans. It was therefore decided not to use any flight plan knowledge or aircraft intent other than the destination airport. Then using the knowledge gathered from the clustering methods, we propose a real time airspace monitoring tool that evaluates the conformance of current flight to pre-identified typical trajectories. A measure of airspace complexity based on this conformance is also proposed with the tool. The overall method developed in this paper is neither location nor data specific and can easily be adapted to other data sets since unsupervised methods are used, and the data is not labeled. This paper considers radar tracks in a terminal radar approach control (TRACON). However, the underlying principles may also be used for other applications, such as a fleet of GPS-equipped trucks. Since this paper deals with different problems such as trajectory clustering, airspace monitoring and airspace complexity, the literature review is spread along the paper at the beginning of the corresponding section. The remainder of this paper is organized as follows: The first section presents the data set used for the study. The second section presents the trajectory clustering methods, and finally, before the concluding remarks, the third section introduces AirTrajectoryMiner, the airspace monitoring tool that detects in real time the aircraft that do not comply to typical operations. In the paper, trajectories constituting typical operations will also be refereed as nominal trajectories.

II. Available Data

The available data [a] consists of records of flight tracks over the San Francisco bay area, for the first 3 months of 2006. The records cover the Northern California TRACON (NCT), that is, a cylinder of radius 80km and height 6,000m centered at Oakland International Airport (OAK). The NCT contains 3 main airports − Oakland, San Francisco (SFO) and San Jose International airports − as well as many smaller airports. The NCT is the fourth busiest terminal area in the US[3] with an average of 133,000 flight instrument operations per month in 2006. The data, made of the position and speed of aircraft, is organized by flight and also contains meta-data for each flight that include: type of operation (departure/arrival), origin and destination airports, aircraft type (business, jet, helicopter, other, etc), date and time of beginning of record, duration of the record, etc.

Using the available meta data, visual flight rules (VFR) traffic is discarded, since it is more unpredictable and does not follow the same rules as instrument flight rules (IFR) traffic. The meta data is used to sort trajectories by airport and operation type, i.e. take off or landing. After a visual analysis of the flight patterns for the different airports, it was decided to focus the study on the landings at SFO. It is the busiest airport in the NCT and the arrival tracks present the most interesting patterns by their numbers and variety. The most frequent configuration is the "West" configuration, where aircraft land on runways 28L/R and take off from runways 19L/R. A diagram of SFO is presented in Figure 1, and Figure 2 depicts the NCT traffic patterns typically used in the west configuration.

In this paper, the axes are set by the radar, located at $(0, 0, 0)$. The x and y axes define the horizontal plane and z the vertical direction, positive going upward. To each recorded flight, corresponds an aircraft i and a trajectory $T_i, i = 1 \dots r$, where r is the total number of trajectories of interest in the dataset. Each trajectory T_i is a $n_i \times 4$ matrix, and the line T_i^l of T_i is the l^{th} radar echo, given by $T_i^l = (x_i^l, y_i^l, z_i^l, t_i^l)$, where (x_i^l, y_i^l, z_i^l) is the 3 dimensional coordinates of aircraft i at time t_i^l. The trajectories have different numbers of points n_i, varying from 10 to about 550 points, depending on the duration of the trajectory. Trajectories with a few data-points usually correspond to short flights from San Jose International Airport or Oakland International Airport to SFO. The interval between points is between 4 and 5 seconds and is given by the rotational speed of the radar (most likely 4.8 sec). The time stamp t_i^l is rounded to the nearest second.

[a] The complete dataset is available for download https://dashlink.arc.nasa.gov/data/flight-tracks-northern-california-tracon/

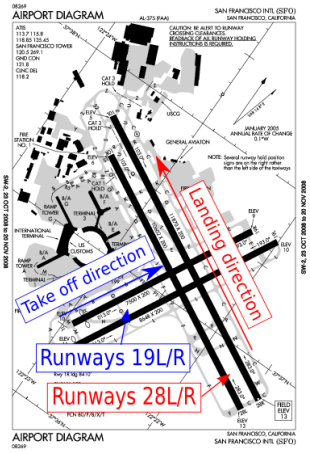


Figure 1. San Francisco airport diagram with take off and landing direction in the west configuration

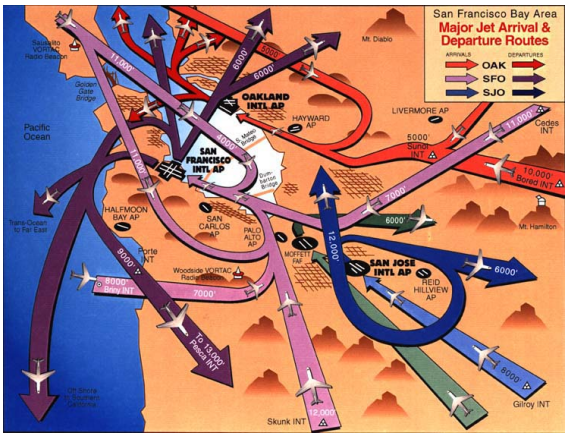


Figure 2. NCT standard traffic patterns, west configuration, image courtesy of Federal Aviation Administration

III. Trajectory Clustering

In this section, two trajectory clustering methods are presented. After a review of existing trajectory clustering methods, a technique based on trajectories' "way-points" is presented. Then, a technique based on a principal component analysis of re-sampled and augmented trajectories is introduced.

A. Literature Review

The use of positioning devices such as GPS and the collection of data has increased over the past 15 years leading to an increasing number of tracking applications. An objective of tracking is to discover common patterns on the one hand, and detect outliers on the other hand.

Piciareli et al.[4] presented an on-line trajectory clustering method for real time video surveillance. Moving objects, such as pedestrians, are identified in video frames and their trajectories are compared against existing cluster representatives, that is, an average of all the trajectories in the cluster. The match between a trajectory and a cluster is determined using the mean of the normalized distances of every trajectory point to the nearest point of the cluster representative. If a match is found, the cluster representative is updated. If not, a new cluster is created. In this approach, the cluster representatives evolve with time. This clustering method was used by Dahlbom and Niklasson for coastal surveillance but failed to provide satisfactory results when dealing with real data sets[5] such as ship trajectories.

Lee et al.[6] presented a partition-and-group framework for trajectory clustering. Trajectories are partitioned in sub-trajectories. Sub-trajectories are represented by line segments and grouped using a distance function. The distance function incorporates three components that measure the perpendicular distance, the parallel distance and the angular distance between the line segments. The clustering algorithm is density based, i.e clusters are created where the density of points is the highest. The formulation is powerful but the results are presented on very noisy data where it is difficult to visually cluster the trajectories.There exists no well-defined measure to assess the results of the clustering method. Based on the same distance measure, Lee et al.[7] present a trajectory outliers detection procedure. The results are presented on the same noisy datasets and therefore difficult to evaluate visually.

Vlachos et al. used similarity functions based on the longest common subsequence (LCS) to discover similar multidimensional trajectories.[8] Their LCS based clustering method appears to be more efficient than Euclidean distance based measures and dynamic time warping

distance functions, especially in the presence of noise.

Eckstein proposed an automated flight track taxonomy[9] . The trajectories are first re-sampled, then clustered using k-means on a reduced order model. The model reduction is the truncation of a proper orthogonal decomposition (POD), also called principal components analysis. The trajectories are clustered using only the first two modes of the decomposition, as they capture 95% of the fluctuations of the dataset used.

B. Overview of k-means and DBSCAN Clustering Algorithms

OVERVIEW OF k-MEANS[10] This paragraph presents a brief overview of the k-means algorithm. For more details, the reader is refereed to.[11] Given a set $S = (\mathbf{tp}_1, \dots \mathbf{tp}_{|S|})$ of $|S|$ observations (turning points in our case), where each observation is a d-dimensional real vector, then k-means clustering aims at partitioning the $|S|$ observations into k sets, or clusters, $(k < |S|), C = \{C_1, C_2, \dots, C_k\}$ so as to minimize the within-cluster sum of squares:

$$\arg\min_{\mathbf{C}} \sum_{i=1}^{k} \sum_{\mathbf{tp}_j \in C_i} \left\| \mathbf{tp}_j - \mathbf{m}_i \right\|^2 \tag{1}$$

where $\mathbf{m}_i$ is the mean of C_i. The mean $\mathbf{m}_i$ of a cluster is called centroid and is the center of mass of all the elements in the cluster. The number k of clusters is the only input required from the user.

Starting with an initial set of k centers $\mathbf{m}_1^{(1)}, \dots, \mathbf{m}_k^{(1)}$, which may be specified randomly or by some heuristic, the algorithm proceeds by alternating between two steps, also known as Lloyd Algorithm:[12]

Assignment step: Assign each observation to the cluster with the closest mean, that is partition the observations according to the Voronoi diagram generated by the centroids of the clusters. Figure 3 presents the results of k-means clustering and the corresponding Voronoi diagram.

$$\begin{aligned} C_i^{(t)} = \{ & \mathbf{tp}_j : \left\| \mathbf{tp}_j - \mathbf{m}_i^{(t)} \right\| \leq \left\| \mathbf{tp}_j - \mathbf{m}_{i^*}^{(t)} \right\|, \\ & \text{for all } i^* = 1, \dots, k \} \end{aligned} \tag{2}$$

Update step: Calculate the new means to be the centroid of the observations in the

cluster.

$$\mathbf{m}_i^{(t+1)} = \frac{1}{|C_i^{(t)}|} \sum_{\mathbf{tp}_j \in C_i^{(t)}} \mathbf{tp}_j \tag{3}$$

The algorithm is deemed to have converged when the assignments no longer change. Since it is a heuristic algorithm, there is no guarantee that it will converge to the global optimum, and the result may depend on the initial clusters. Since the algorithm is usually very fast, it is common to run it multiple times with different starting conditions and keep the run that resulted in the minimum value for equation 1.

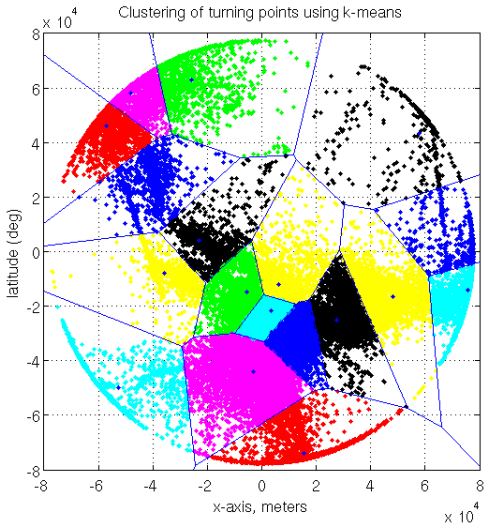


Figure 3. Clusters of turnings points and corresponding Voronoi diagram

Overview of DBSCAN This paragraph presents a brief overview of the DBSCAN algorithm. For more details, the reader is refereed to.[13] DBSCAN [14] stands for Density-Based Spatial Clustering of Applications with Noise. DBSCAN clusters points that are close together (in an ϵ neighborhood), and surrounded by sufficiently many points. DBSCAN requires two parameters: a real,ϵ, and the minimum number of points, $MinPts$, required to form a cluster. The ϵ-neighborhood of a point p consists of all the points q s.t $dist(p,q) \leq \epsilon$. If the ϵ-neighborhood of a point p contains more than $MinPts$, a new cluster is started, with p as a core object. DBSCAN then iteratively collects directly density-reachable objects from these core objects. An object q is said to be directly density-reachable from an object p if q is in the ϵ-neighborhood of p and p is a core object.

If a core object q of a cluster C_i is added a cluster C_j, C_i and C_j are merged. When no

point can be added to any cluster, the process terminates.

C. Way-point based Trajectory Clustering

This section presents a novel algorithm for aircraft trajectory clustering. This algorithm takes advantage of aircraft trajectory properties: aircraft usually fly straight, with a limited number of turns. This method arises from the current instrument flight rules procedures. When approaching an airport, aircraft usually follow published procedures made of a sequence of way-points. A way-point is characterized by its GPS coordinates and, sometimes, an altitude indication. The planar localization of a way-point is very accurate but its vertical component often looks like "at or above —ft". Vertical clearances are delivered by ATC and trajectories' vertical profiles are then at the discretion of the pilots. Therefore, this method focuses on the 2D coordinates of the way-points in the (x, y) plane. This method is an efficient way to determine the compliance of flown trajectories with published procedures. Nevertheless, published procedures cannot be used because of the limited number of way-points or reporting points located in the TRACON. In Section IV, we further show this by comparing the results of the trajectory clustering with the published way-points.

The objective is to identify and group the turning points into "way-points". A turning point is a point in the trajectory where the aircraft changes heading. Then trajectories are represented by a sequence of way-points. Finally, trajectories are clustered using the Longest Common Subsequence (LCS). The algorithm proceeds using the following steps and it is summarized in Figure 4:

1. Identify the location of the turning points of each trajectory.

2. Cluster the set of all the turning points of all the trajectories. This clustering task is done using k-means[11,15] or DBSCAN[14] (Density-Based Spatial Clustering of Applications with Noise). Section B gives an overview of those algorithms. This clustering provides a finite number of way-points where it has been determined that aircraft usually turn.

3. Represent each trajectory by its sequence of way-points.

4. Cluster the sequences of way-points using the SequenceMiner algorithm.[16,17] SequenceMiner provides us with a representative trajectory for each cluster.

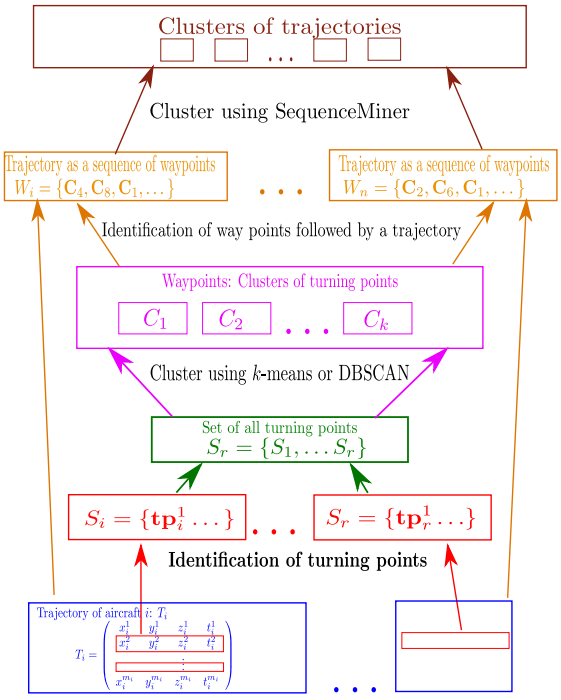


Figure 4. Way-point clustering method

1. *Turning Points Identification*

The first step is to extract the location of the turning points of each trajectory. To simplify the notations, the aircraft index i is omitted in the following equations. The heading Ψ^l of an aircraft at time t^l can be estimated by $\psi^l = \arctan \frac{y^{l+1}-y^{l-1}}{x^{l+1}-x^{l-1}}$, at each point of the trajectory, l, $l = 2 \ldots n-1$, where n is the total number of points. Since the trajectory is a bit noisy, a low pass filter is applied:

$$\tilde{\psi}^1 = \psi^1 \tag{4}$$

$$\tilde{\psi}^l = \alpha\psi^l + (1-\alpha)\tilde{\psi}^{l-1}, \quad l = 2 \ldots n-1, \tag{5}$$

where α is a constant for the filter. On this data, setting $\alpha = 0.4$ provided good noise filtering results and not too much delay. A turning point **tp** is identified when the heading difference between two consecutive values of the heading exceed a threshold: $|\tilde{\Psi}^l - \tilde{\Psi}^{l-1}| > \Psi_c$. The threshold was chosen relatively small in order to capture small heading changes but not small enough not to capture meaningless heading changes variations: $\Psi_c = 0.025\text{rad} = 1.43°$. This value was set experimentally. The results are not very sensitive to a small change in Ψ_c. The

number of turning points is trimmed to avoid long sequences when aircraft are executing large turns: if two consecutive data-points are determined to be turning points, then only the first one is kept; if three, only the middle one, etc.

The trajectory of aircraft i is now represented as a sequence of turning points S_i :

$$S_i = \{\mathbf{tp}_i^1 \dots \mathbf{tp}_i^s\},$$

where $\mathbf{tp}_i^s$ is the 3D coordinate of the s^{th} turning point of trajectory i. The first point of the trajectory is labeled as a turning point. Figure 5 presents a sample of 11 trajectories and the points identified as turning points.

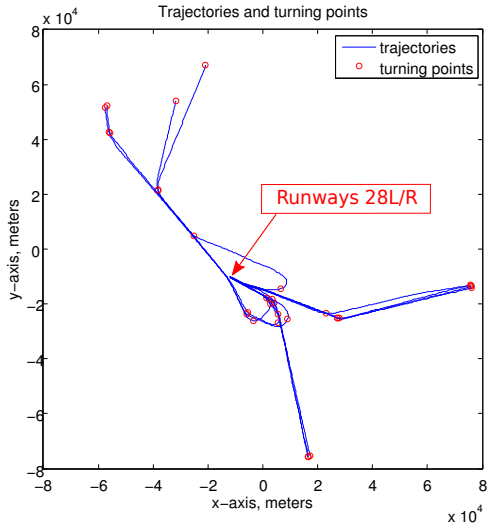


Figure 5. Trajectories and identified turning points

Denote by S the set of all the turning points for all the trajectories: $S = \{S_1 \dots S_n\}$.

The second step is to cluster the set S of turning points. The following section introduces the two clustering algorithms used in this paper: k-means[10] and DBSCAN.[14]

2. Turning Points Clustering; Creation of Way-points

To determine the way-points, the turning points are clustered: a way-point is defined as the planar (x, y) coordinates of a cluster of turning points. The idea is to create a way-point where it has been determined that many aircraft turned. Depending on the number and on the density of available turning points, two different algorithms are used. When the spatial

distribution of turning points is sparse, k-means is used, and when the distribution of turning points is dense, DBSCAN is used.

Case when the Data is Sparse When the number of turning points is small, a density based clustering algorithm would provide poor results, identifying most of the points as outliers. Therefore, a distance-based algorithm is used so all the turning points available are used. A way-point is created for each cluster produced by k-means. Using cylindrical coordinates, the coordinates of the center of a way-point are given by $(r_m\, \theta_m)$. The center is the center of mass of all the points in the cluster. The coordinates of the corners of the way-points are given by $\{(r_m + 2std_r, \theta_m + 2std_\theta), (r_m - 2std_r, \theta_m + 2std_\theta), (r_m - 2std_r, \theta_m - 2std_\theta), (r_m + 2std_r, \theta_m - 2st$ where std_r and std_θ are the standard deviation of the radial coordinates and angular coordinates of the points in the cluster, respectively. Figure 6 presents the outcome of clustering the way-points for one day of trajectories. Each cluster is represented using a different color/shape combination. The way-points are represented by pairs of nested polygons on the figure. The inside polygon corresponds to $(r_m \pm std_r, \theta_m \pm std_\theta)$ and the outside one to $(r_m \pm 2std_r, \theta_m \pm 2std_\theta)$. The number is the label of the cluster.

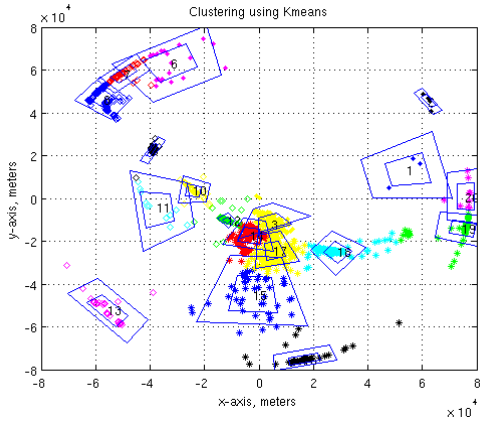


Figure 6. Result of the clustering of the turning points for one day using k-means

Case when the Data is Dense When the the number of turning points is large, a large share of the airspace is covered with turning points. A distance based algorithm such as k-means provides meaningless clusters for our application. Figure 3 shows the clusters provided by k-means and the corresponding Voronoi diagram for the turnings point of almost 3 months of data (30,000 trajectories).

To overcome this issue, the turning points were clustered using DBSCAN. DBSCAN

is particularly efficient at cluster data in the presence of noise. Way-points are created using the convex hull of the clusters resulting from DBSCAN. Figure 7 shows the result of the clustering of the turning points using DBSCAN. The blue polygons represent the way-points. All the points identified as outliers, i.e not associated with any way-point, are not depicted. The parameters used were $\epsilon = 350$m and $minPts = 10$. The main issue with DBSCAN is its execution time since its complexity is in $O(n \log n)$. Here, the number of turning points to cluster is $n = 118,179$ for 30,000 trajectories.

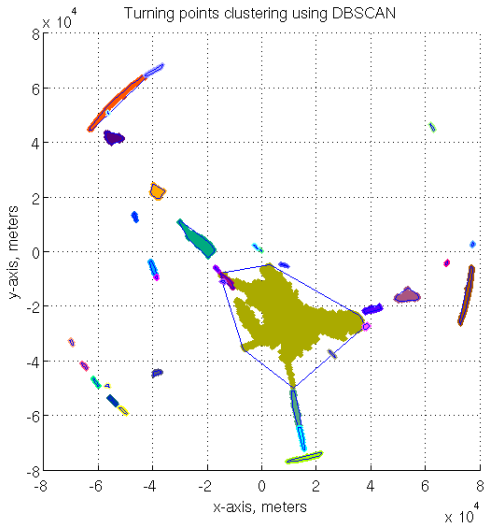


Figure 7. Result of the clustering of the turning points for the entire dataset using DBSCAN. Outliers are not displayed

3. Converting a Trajectory into a Sequence of Way-points

The way-points have been discovered using the turning points of the trajectories. Nevertheless, some trajectories might go over way-points without actually turning. To identify the sequence of way-points followed by a trajectory, the following procedure is used for each trajectory: start with an empty sequence of way-points, and given the set of all way-points, run the trajectory along its original direction. If one of the points is located over a way-point, the way-point is added to the sequence. Each trajectory is now represented as an ordered sequence of way-points, where the number of way-points is finite. The next step is to cluster the trajectories determining the longest common subsequence (LCS) of way-points.

4. Longest Common Subsequence Determination

The sequences of way-points are clustered using the longest common subsequence. The LCS problem is to find the longest subsequence common to all sequences in a set of sequences. SequenceMiner[16,17] is an algorithm that identifies the LCS and generates clusters of sequences. With this method it is possible to cluster sequences that do not contain the same number of elements. When sequences have only a small number of points, say fewer than 3, this clustering method does not work well. Therefore, only the sequences containing more than 4 way-points are kept. The total number of way-points being small, it is preferable to focus on the way-points at the beginning of the trajectory: since most aircraft do a final turn to get aligned with the runway, this turning point does not bring much information about the trajectory. Therefore, if the last turning point is in the large brown cluster (Figure 7), it is removed from the sequence.

Figure 8 presents the results of the clustering process using k-means and LCS on a low number of trajectories. The dataset used is the tracks of all the aircraft landing at San Francisco (SFO) airport on February 10, 2006. Only the trajectories of that day were used to determine the way-points. Each cluster is represented by a color. The algorithm identifies the main flows but a few trajectories seem not to belong to the expected cluster. The quality of the results is subjective and can only be visually assessed. Figure 9 presents the results for an initial set of 30,000 trajectories, using DBSCAN and LCS. Here, the denomination "Nominal" qualifies the trajectories containing more than 4 way-points. The colors correspond to the clusters. The colors differ on Figures 8 and 9 because the indexing of clusters is random and depends on the order of the data in the dataset.

Overall, this method presents good clustering results. One of the main drawbacks of this method is that it only keeps the trajectories going over the way-points. For instance, consider two parallel trajectories: one going over the way-points and the other one slightly off. The latter will be considered as an outlier even though it is very similar to the first trajectory, resulting in excluding many trajectories. In addition, trajectories containing large rerouting periods will belong to the clusters as long as they pass over way-points.

D. Trajectory-based Clustering via Principal Components Analysis

This method proceeds with the following steps, which are summarized in a diagram in figure 10:

1. Re-sample the trajectories, to obtain time series of equal length for each aircraft.

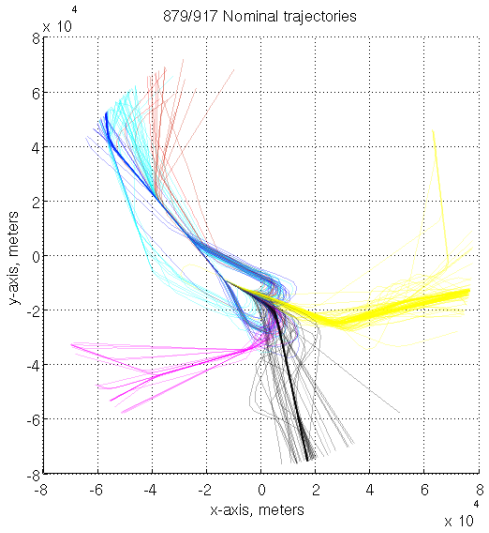


Figure 8. Results of trajectory clustering for the landings of one day at SFO

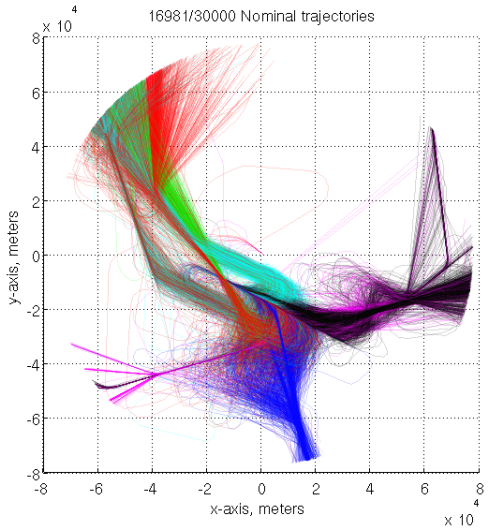


Figure 9. Results of trajectory clustering for 30,000 trajectories

2. Augment the dimensionality of the data. Normalize and concatenate all the data into a single vector for each flight.

3. Run a principal components analysis (PCA) and keep the first 5 principal components (PCs).

4. Cluster using a density-based clustering algorithm.

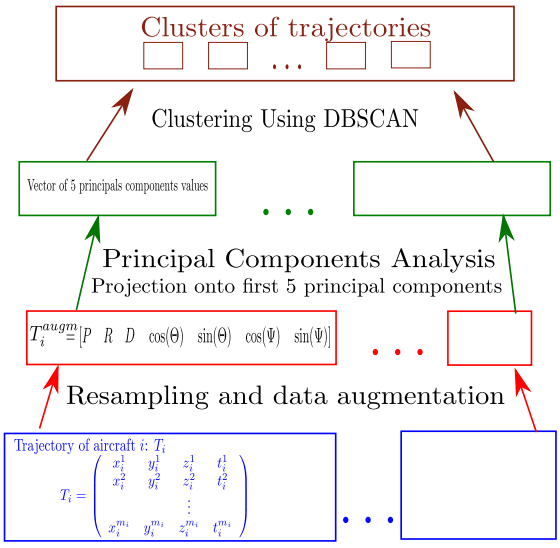
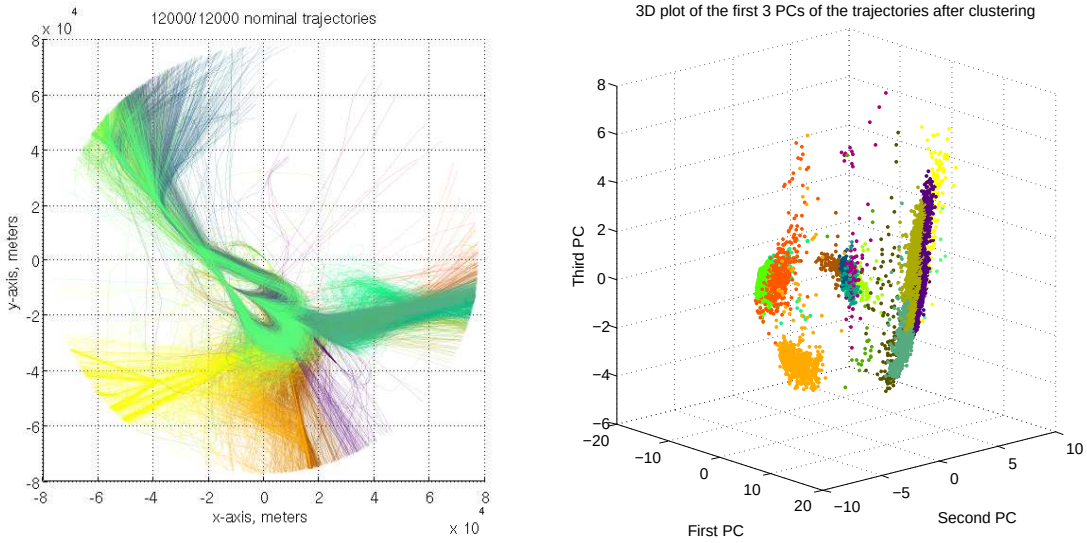


Figure 10. Trajectory clustering method based on Principal Components Analysis

This paper proposes improvements to the approach used by Eckstein[9] to realize a trajectory taxonomy. In,[9] trajectories are first re-sampled, then the principal components are extracted and finally, the clustering is realized using k-means on the projections onto the first two principal components. Figure 11(a) presents the resulting clusters on the principal components and on the trajectories, using the methods introduced in.[9] The clustering technique proposed in[9] does not provide results precise enough for our data set and there is no identification of outliers. Figure 11(b) presents a 3D view of the projection onto the first three PCs (section 3) so it can be compared with our method. Eckstein used only the first two PCs for clustering. The first improvement is to augment the dimensionality of the data. Then, the PCs are computed and the projections of the augmented trajectories onto the first five PCs are clustered using a density based clustering algorithm. This algorithm presents the advantage of identifying outliers. Another advantage is that the number of clusters is not set a priori.



(a) Clusters of trajectories

(b) Clusters of the first principal components

Figure 11. Clustering results using the method presented by Eckstein

1. Trajectory Resampling

The dataset is well organized and fairly clean. Trajectories with fewer than 50 points are removed from the dataset: to be able to use a clustering algorithm such as DBSCAN, each trajectory must be represented as a vector. All vectors must have the same number of elements n, so their distance can be computed. Since all trajectories do not have the same number of points, re-sampling is necessary. Trajectories are resampled so that the total number of points for each trajectory is 50. For the sole purpose of clustering, fewer than 50 points would have been enough. Nevertheless, to improve the accuracy of the airspace monitoring function presented in section IV, 50 points were used. The re-sampled trajectory T_i' is given by $T_i^{samp} = \left\{T_i^l, l = \{\text{round}(\frac{k\ n_i}{50}), k = 1 \dots 50\}\right\}$. During this operation, the notion of speed that was given by the distance between the radar echoes is lost. For example, consider the trajectories of two aircraft with the exact same flight path, but one going twice as fast as the other. After re-sampling, the trajectories will have the exact same points and it will be impossible to determine that there was a speed difference.

2. Dimensionality Augmentation

To improve the results of the clustering, the dimensionality of the data was increased. Some of the added dimensions present symmetry with respect to a point or a line and some do not.

- Cartesian position of the aircraft in the re-sampled trajectory: $P = [x_i^1 \dots x_i^{50} y_i^1 \dots y_i^{50} z_i^1 \dots z_i^{50}]$. P is a row vector with 150 components. This vector is unique to each trajectory.

- Distance from the center of the TRACON $R = \{r_i^l = \sqrt{(x_i^l)^2 + (y_i^l)^2 + (z_i^l)^2}, l = 1 \dots 50\}$. Provides information about the rate of convergence of the aircraft toward the center of the TRACON, which is located close to the airport. This distance presents a symmetry with respect to the center of the TRACON, i.e two trajectories that are symmetric with respect to the center of the TRACON will be represented with the same vector R.

- Distance from the top left corner: $D = \{d_i^l = \sqrt{(x_i^l - x_{ref})^2 + (y_i^l - y_{ref})^2 + (z_i^l)^2}, l = 1 \dots 50\}$, where (x_{ref}, y_{ref}) are the coordinates of the top left corner of a square containing the TRACON. The top left corner has coordinates $(x_{ref}, y_{ref}) = (-80, 80)km$. This distance presents a symmetry with respect of the diagonal joining the top left corner $(-80, 80)$ and the bottom right corner $(80, -80)$, i.e two trajectories that are symmetric with respect to this diagonal will be represented with the same vector D.

- Angular position in cylindrical coordinates: $\Theta = \{\theta_i^l = \arctan(\frac{y_i^l}{x_i^l}), l = 1 \dots 50\}$. With only one dimension, the angular position provides information about the overall location of the trajectory in the TRACON, i.e in which quadrant of the circle the trajectory lies. This information does not present any symmetry.

- Heading of the aircraft $\Psi = \{\psi_i^l, l = 1 \dots 50\}$. The computation of the heading was done using the filter of equation 4 and then re-sampled to 50 points. A constant value or a slow rate of change indicate a straight trajectory, while a high variability indicates a curved trajectory. This vector is unique to each trajectory.

The sine and cosine values of the angular position and heading are used instead of their actual value to avoid the discontinuity at 2π. Each augmented trajectory is now represented by a vector of dimension 450 given by: $T_i^{augm} = [P \quad R \quad D \quad \cos(\Theta) \quad \sin(\Theta) \quad \cos(\Psi) \quad \sin(\Psi)]$. The initial vector had dimension 150. The values of each parameter are normalized between 0 and 1 in order to balance their importance during the clustering process. It was decided

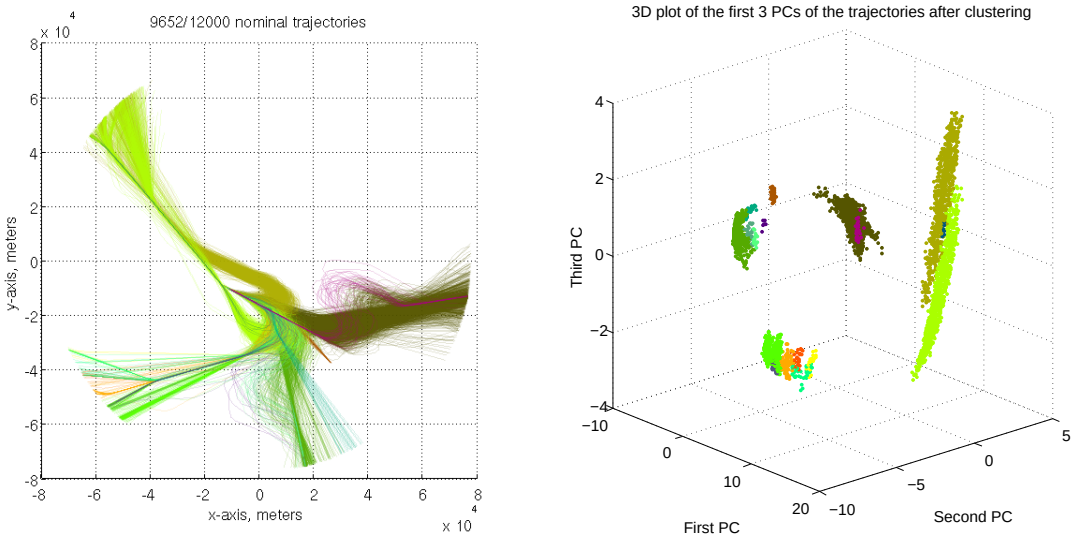
to add meaningful data such as heading or rate of convergence toward the center of the TRACON. For instance, two aircraft on parallel trajectories will fly the same heading, even if the trajectories are slightly apart from each other. Distance to the center was chosen to identify trajectories that have particular patterns such as vectoring and holding pattern: the distance to the center will present some irregularities as the aircraft flies back and forth. Such irregularities will be highlighted by dimensions such as the heading that will change 180° while the position will only change slightly.

3. Principal Components Analysis

A principal components analysis[18] is run on matrix that contains all the re-sampled trajectories. Each trajectory is then projected onto the first p principal components and is now represented by a vector of p values. The choice in the value of p is a trade-off between computational speed when p is small and accuracy when p gets larger. There is no need to get a value of p too large since the first principal components contain most of the information. Different values of p were tried and $p = 5$ gave a satisfactory level of accuracy for this type of data. The added dimensions increase the range of the projection of the trajectories onto the principal components. This makes the clustering task easier as the clusters are "further apart" in the principal components space.

4. Clustering

The projections of the trajectories onto the first 5 PCs are clustered using DBSCAN. A density based clustering algorithm like DBSCAN is preferred to a distance based algorithm because of the shape of the clusters can be arbitrary. The other advantage of DBSCAN is the identification of outliers. Figure 12(b) presents the resulting clusters. The axes correspond to the values of the first 3 principal components. Clusters are clearly differentiated, even if they are not easy to distinguish on the plot due to the perspective effect. The resulting clusters of trajectories are visually very clean (Figure 12(a)). Figure 13 presents the centroids, that is the center of mass of the trajectories of each cluster. Those centroids can be seen as "typical operations". Some clusters are minor variations from each other, such as the flights coming from the bottom left corner. This comes from the settings used for DBSCAN. On Figure 12(b), one can clearly identify clusters of points. The algorithm was run with a high sensitivity (ϵ small and $minPts$ large). The parameter ϵ reflects the similarity between trajectories (the smaller the more similar), and $minPts$ is the number of "similar"



(a) Clusters of trajectories

(b) Clusters of the first 5 principal components

Figure 12. Clustering results using re-sampling, data augmentation, PCA decomposition, and DBSCAN on the first 5 principal components.

trajectories needed to create a new cluster (Section 2). A small ϵ generates "narrow" cluster while a larger ϵ will generate clusters with more variability in trajectories. In the application presented in section IV, the algorithm is run with a lower sensitivity and provides fewer clusters, with larger variability. The resulting centroids of this run can be seen on Figure 20.

5. Analysis of Outliers

Figure 14 shows the outliers detected by the clustering algorithm. Outliers represent 19.5% of all trajectories. A visual inspection shows that the main reasons for being detected as an outlier is the presence of holding patterns, large vectoring maneuvers or direct routes.

Figure 15 presents the number and frequency of outlier trajectories as a function of the type of aircraft. Commercial jets represent the largest share in numbers, but the frequency is much smaller. Among the trajectories of regional and business aircraft, 4% and 5% are identified as outliers, respectively. A possible explanation is the size, the speed and the maneuverability of the aircraft. To ensure a safe separation at the runway threshold, air traffic controllers "vector" aircraft, that is give a sequence of headings to follow. The vectors given to business and regional aircraft might be different and sharper than the vectors given

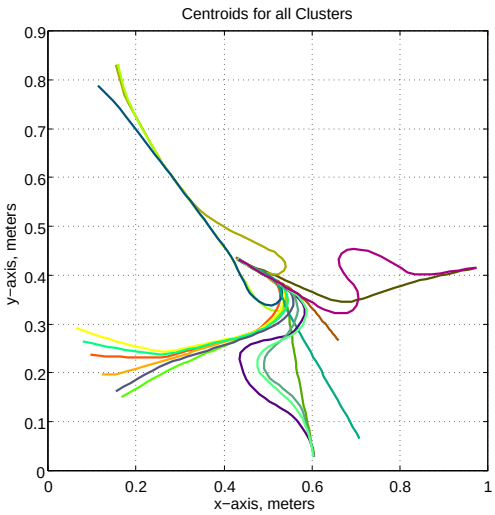


Figure 13. Clusters centroids (average trajectory)

to larger size jets.

Figure 16 presents the frequency of outliers for each day of study. Each bar represents one day. The minimum percentage of outliers is less than 2% and goes up to 16%. The most likely explanation for the outliers is the weather. San Francisco airport usually operates with two close parallel runways. The runways are not independent, that is, they cannot be operated simultaneously when the weather does not permit visual approaches. When a runway is closed, the landing capacity is reduced from 60 to 30 aircraft per hour. Schedules and operations usually take the weather into account , but unexpected late fog dissipation or other type of convective weather might disrupt the operations and force controllers to vector aircraft and put them on holding patterns.

Figure 17 presents the frequency of outliers as a function of the time of the day. The local time is reported on the abscissa axis, starting at midnight. This diagram is an average over the entire period of interest. The frequency of outliers is higher during the period 12 a.m. - 4 a.m., then decreases in the early morning, to an increase again with a peak at 11 a.m.. Another peak is visible at 5 p.m.. The outliers identified at night are mostly due to direct routing that is allowed by the very low traffic density at night. During the morning, traffic density increases and requires more rerouting for efficient sequencing and merging. Another possible explanation is the late dissipation of the fog.

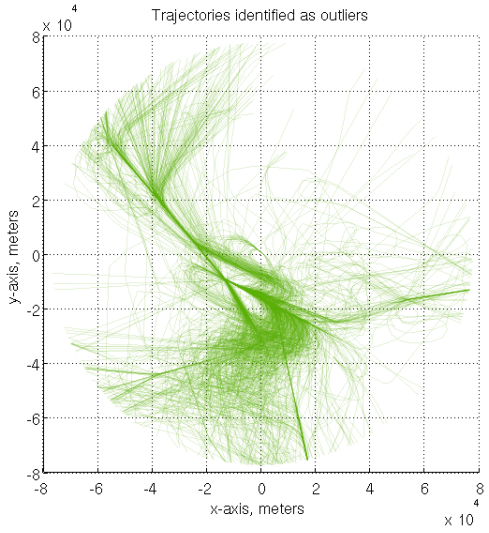


Figure 14. Trajectories identified as outliers

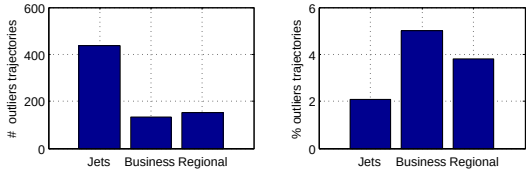


Figure 15. Distribution of outliers by aircraft category

IV. Airspace Monitoring

This section proposes an airspace monitoring technique that automatically detects when an aircraft is not conforming to typical operations. Typical operations are determined using the centroids of the clusters found using the previous clustering methods. Centroids correspond to flight path often flown and the value of the parameters used for clustering allows the trajectories to vary more around the centroids. The objective of the monitoring task is to detect when an aircraft deviates from typical path in real time.

A. Literature Review and Motivation

Krozel[19] proposed an intent based monitoring where the aircraft is tracked relative to a filed flight plan, using NavAids and way-points. The monitoring tasks requires knowledge of the airspace structure, of the trajectory way-points and of the intent of the aircraft. It

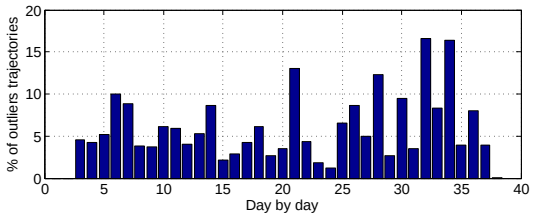


Figure 16. Histogram of outliers, day by day

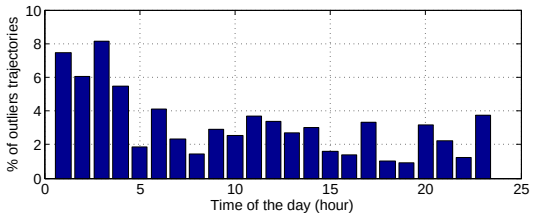


Figure 17. Histogram of outliers, hour by hour, local time

is a powerful tool when the flights behave according to their flight plan, but when dealing with arrivals, the sequence of way-points might change, some might be skipped or added to ensure an optimal separation of aircraft at the runway threshold and vectoring is often used. This monitoring method cannot be used. Reynolds et al.[20] introduced a framework for the development of an automated conformance monitoring system. The system described in[20] has two main inputs: the conformance basis, containing target states and trajectory information, and the observation of a surveillance system. Those inputs feed models for pilots' intents, aircraft intents, and aircraft control systems and dynamics. Those models provide an expected state vector that is compared with the observed state vector for conformance analysis. This structure is further used in[21] to monitor the conformance of a trajectory to a flight plan. For instance, it detects if an aircraft does not turn, turns too early or too late at a way-point. The monitoring is based on intent, and knowledge of the the exact expected trajectory is required. An off-line trajectory analysis and taxonomy for arrival trajectories was proposed by Eckstein.[9] The objective of Eckstein is to analyze the performance of area navigation (RNAV) operations for NextGen concepts of operations off-line. The method in [9] uses GPS coordinates of actual way-points to identify and classify segments of trajectories. This approach can be used only if aircraft follow RNAV operations, which is not the case with the data used.

Figure 18 displays an aerial view of the San Francisco Bay Area. The blue circle represents

the outer boundary of the TRACON, given by the area covered by the radar. The white lines are the centroids identified in section 4. The yellow dots are way-points or reporting points. The locations of those points come from Standard Terminal Arrival Routes (STAR) and track logs.[22] The centroids of the clusters pass over only a limited number of way-points. This shows that using published way-points and reporting points cannot be efficiently used to monitor traffic in the TRACON. The intent based methods cannot be used in the terminal area. Figure 18 also displays the arrival from the north for turboprops. This arrival procedure has not been identified by the clustering algorithm because of the relatively small number of aircraft using this route, and of the variability of the flight path following this procedure. A real time trajectory analysis tool built upon the knowledge gathered from the clustering

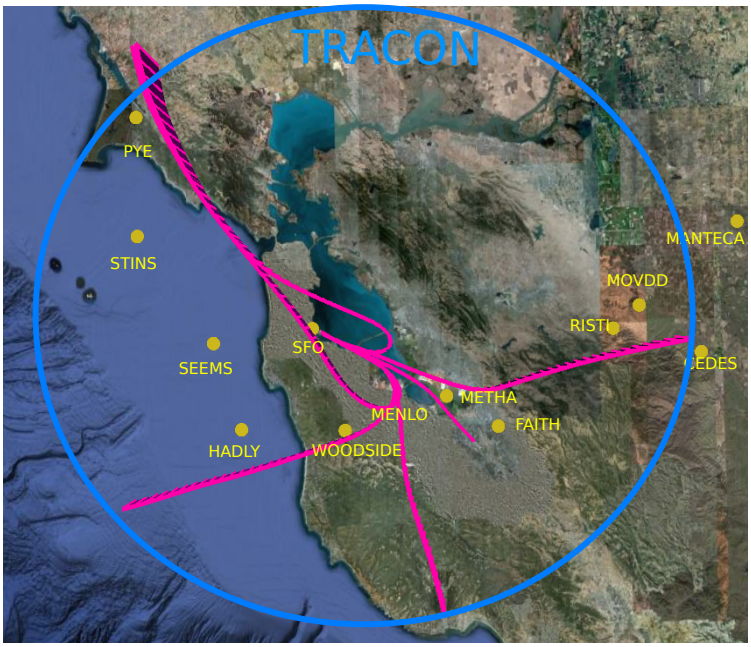


Figure 18. Centroids of the clusters and reporting points/way-points for SFO arrivals. Those centroids differ from the ones in Figure 13 because the algorithm was run with differen parameters with a smaller sensitivity.

analysis is now proposed. The tool is called AirTrajectoryMiner (ATM) since it enables the monitoring of operations in the TRACON. Current aircraft trajectories are compared against nominal trajectories, that is the trajectories in the clusters. If they differ too much, the current trajectory is tagged as abnormal, or outlier. The only intent used is the aircraft final destination airport. The tool automatically detects if the aircraft is flying one of the possible approaches, including most commonly used vectoring maneuvers.

B. Data Formatting

It is not possible to directly compare the current trajectories with nominal trajectories, since current trajectories are incomplete. During real-time system operations, only past data are known. Therefore, the nominal dataset is fragmented. Re-sampled trajectories that had 50 points are split in 10 fragments of 5 points. The average travel time in the TRACON for aircraft landing at SFO is about 14 minutes. Therefore, 5 data points correspond to about 14*60/10 = 84 seconds. A memory of 80 seconds is used for current tracks. The radar hits of the last 80 seconds of flight are re-sampled to 5 points that can now be compared against the database of nominal tracks. This comparison is done using the Inductive Monitoring System.

C. Anomaly Detection: Inductive Monitoring System

To detect anomalous trajectories, the Inductive Monitoring System (IMS)[23] is used. IMS is a good alternative to model based health monitoring systems. It provides a high fidelity detection tool, and there is no need to manually build a model. IMS runs in two steps: learning phase and anomaly detection. IMS learns the nominal behaviors using a training dataset provided by the user. IMS builds clusters using k-means clustering and density-based clustering. During the anomaly detection phase, the input data is compared with knowledge base built from the training data. The anomaly score can be interpreted as the distance to the nearest cluster. The input data belongs to a cluster if all the parameters values are within the range specified by the cluster limits.

The training dataset includes all the trajectories identified as nominal and fragmented into 10 segments of 5 points. The total number of segments was 276,040.

D. AirTrajectoryMiner: Monitoring tool

AirTrajectoryMiner (ATM) is a real time TRACON monitoring tool. Figure 19 shows how ATM could be incorporated into the air traffic management environment. The inputs to the tool are the set of all the trajectories identified as nominal, work resulting from section 4, and the radar tracks of the flights of interest. ATM delivers two types of outputs. On the one hand, it delivers an indication of conformance of current flight to nominal procedures, and on the other hand, it delivers a measure of the complexity in the TRACON that can be incorporated in Traffic Management Advisor (TMA) software.[24]

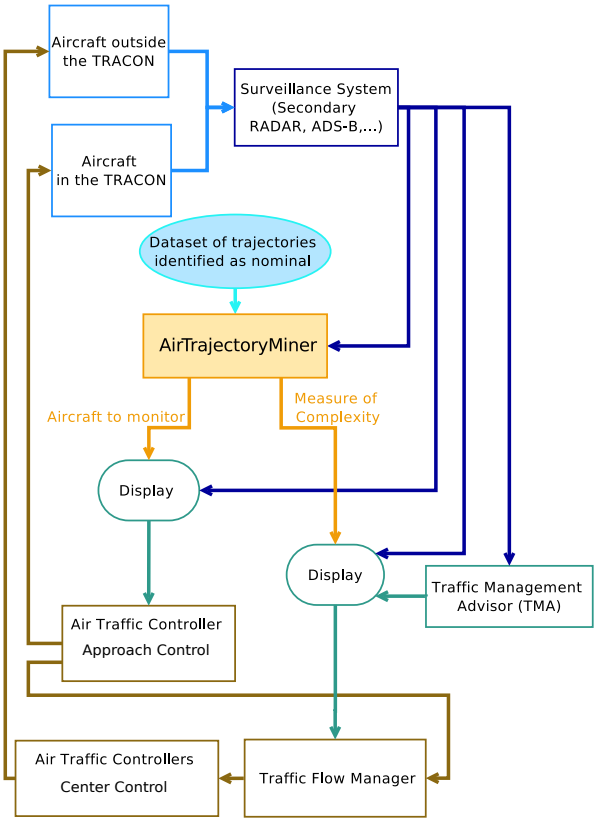


Figure 19. Schematic view of the air traffic control system in and around the TRACON - Integration of AirTrajectoryMiner

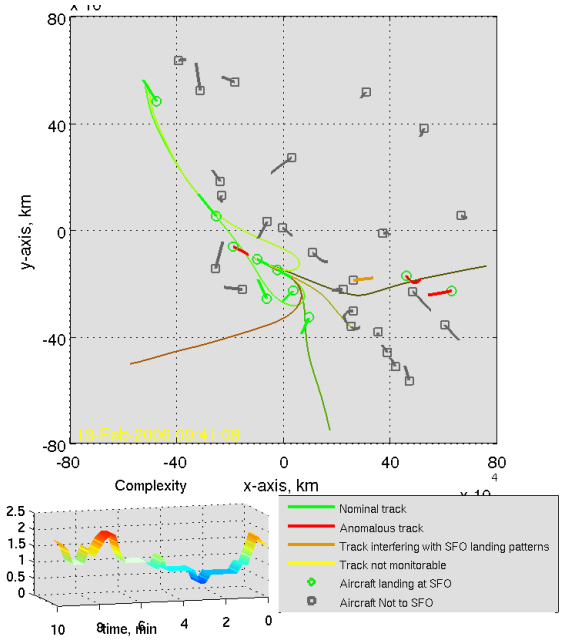


Figure 20. AirTrajectoryMiner display. Top frame: conformance to typical operations. Bottom frame: time history of complexity in the TRACON.

Figure 20 displays the monitoring environment. The top frame is a 2D view of the airspace. The airspace correspongs to a cylinder of radius 80km, going from the ground up to 6000 m and centered at OAK. The following provides some information about the display nad associated aircraft count.

- **Green circle**: aircraft intended to land at SFO (associated count: n_{SFO}).
- **Grey square**: aircraft not intended to land at SFO (associated count: $n_{\overline{SFO}}$).
- **Green segment**: trajectory of an aircraft intended to land at SFO and following the procedures (associated count: $n_{OK,SFO}$).
- **Red segment**: trajectory of an aircraft intended to land at SFO and whose trajectory is identified as an outlier: it does not to follow the procedures (associated count: $n_{\overline{OK},SFO}$).
- **Grey segment**: trajectory of an aircraft not intended to land at SFO and whose trajectory does not interfere with traffic landing at SFO $n_{OK,\overline{SFO}}$).
- **Orange segment**: trajectory of an aircraft not intended to land at SFO and whose trajectory may interfere with traffic intended to land at SFO. The trajectory was

identified as nominal for landing at SFO, but it is not intended to SFO (associated count: $n_{\overline{OK},\overline{SFO}}$).

- **Colored lines**: Centroids of the clusters of trajectories identified as nominal. The centroids presented differ from the ones on Figure 13, because the clustering algorithm was re-run with different parameters allowing more variability and therefore creating fewer clusters.

Aircraft intent information comes from the data. The length of the line following the aircraft corresponds to the part of the trajectory being analyzed, that is the last 80 seconds of the trajectory. The length of the line is therefore proportional to the velocity of the aircraft. This display is intended for an air traffic controller managing the arrivals at SFO. A similar display would be used for managing other arrivals or departures. The only change would be the training data for IMS and the centroids displayed. Aircraft with a gray segment can be ignored, since they are not landing at SFO and are not interfering with landing traffic at SFO. Aircraft with a green segment are following the typical operations to land at SFO. Aircraft in orange require special attention since they are not intended to land at SFO but present characteristics that identify them as "in the pattern to land at SFO". They conform with some of the SFO landing trajectories. Aircraft in red also need special attention since they are supposed to land at SFO but currently not on typical tracks. The controller needs to make sure they are not generating conflicts or interfering with other traffic.

E. AirTrajectoryMiner: Measure of complexity

Based on the compliance of current flights to procedures, we define a measure of complexity for the TRACON, which could provide an automatic feedback of the health of the TRACON to the traffic flow manager who regulates the flow of aircraft arriving in the TRACON. Complexity in air traffic management is a widely studied topic.[25] A measure of airspace complexity is called dynamic density[26] and was intended to understand the effect of changing airspace configuration and traffic controller workload. It is a function of the traffic density and the number of aircraft changing heading, speed or altitude, and, the separation between aircraft. This measure is a weighted sum of several parameters and the weights have been determined using human in the loop experiments. The model was fitted to the observations of controllers. Delahaye and Puechmorel[27] propose a measure of complexity based on the Lyapunov exponents of a time varying vector field that interpolates aircraft position and velocities. This intrinsic complexity measure reflects the stability of the traffic configuration.

Lee[28] proposes complexity measure based on the response of the airspace to a disturbance. Disturbance can be an intruder aircraft in the airspace and the corresponding measure of complexity is the deviation required by the other aircraft to solve all the conflicts. Gariel and Feron[29] proposed complexity maps based on the degradation of communication, navigation and surveillance capacities. The degradation results in a required increase in separation distances creating new potential conflicts. The complexity measures the difficulty to steer the traffic from the nominal mode of operation with initial separation distances to degraded mode of operation with increased separation distances.

This paper introduces a new complexity metric, based on the compliance of aircraft to procedures identified as nominal. According to ,[30,31] controllers build a mental model of nominal operations. The complexity of a traffic configuration perceived by the controllers increases when an aircraft flight paths do not follow this mental model. When operations are running as expected, the controller is more efficient and can deal with more aircraft. Thus, increasing the number of aircraft not following nominal procedure will reduce the maximum number of aircraft a controller can deal with simultaneously, reducing the capacity of the airspace. The proposed complexity measure is based on Shannon's theory of communication.[32] Using the aircraft counts introduced earlier, the instantaneous probability of an aircraft inbound for SFO to be identified as nominal is

$$p(OK|SFO) = \frac{n_{OK,SFO}}{n_{SFO}}. \tag{6}$$

For the aircraft inbound for SFO and identified as outliers, it is assumed that each outlier aircraft is unique and independent from other aircraft. At each instant, each outlier is considered different from the other outliers, that is there are $n_{\overline{OK},SFO}$ types of outliers. Therefore, the probability of an aircraft to be a specific outlier is

$$p(\overline{OK_i}|SFO) = \frac{1}{n_{SFO}}, \quad i = 1 \dots n_{\overline{OK},SFO}. \tag{7}$$

The entropy I_{SFO} of the aircraft inbound to SFO is therefore

$$I_{SFO} = -p(OK|SFO)\log p(OK|SFO) - \sum_{i=1}^{n_{\overline{OK_i},SFO}} p(\overline{OK_i}|SFO)\log p(\overline{OK_i}|SFO) \tag{8}$$

$$= -\frac{n_{OK,SFO}}{n_{SFO}}\log\frac{n_{OK,SFO}}{n_{SFO}} - \frac{n_{\overline{OK},SFO}}{n_{SFO}}\log\frac{1}{n_{SFO}}. \tag{9}$$

The same reasoning is used for aircraft not inbound to SFO:

$$I_{\overline{SFO}} = -\frac{n_{OK,\overline{SFO}}}{n_{\overline{SFO}}} \log \frac{n_{OK,\overline{SFO}}}{n_{\overline{SFO}}} - \frac{n_{\overline{OK},\overline{SFO}}}{n_{\overline{SFO}}} \log \frac{1}{n_{\overline{SFO}}}. \tag{10}$$

The proposed measure of complexity C is the sum of the entropy of aircraft inbound to SFO and the entropy of aircraft not inbound to SFO.

$$C = I_{SFO} + I_{\overline{SFO}}. \tag{11}$$

This complexity measure is an indication of the disorder with regard to nominal operations. If no aircraft is identified as an outlier, the complexity is 0. The complexity increases with the number of outliers detected, but also with the number of aircraft. The bottom left plot of Figure 20 shows this measure of the complexity over the last 10 minutes. The plot is refreshed every 15 seconds. When the traffic flow manager sees that the complexity increases, ATM provides information about the operations in the TRACON. If the complexity gets high, the controller in charge of the TRACON is likely to have a high workload. Providing the traffic flow manager with this complexity measure can help him to manage the flow of arriving aircraft. A low complexity suggests that more aircraft can be allowed in the TRACON. Increasing complexity suggests that the TRACON controllers are subject to an heavy workload and that the aircraft arrival rate should be reduced.

This tool can also be used as an automatic independent monitoring tool. Intent based tools[21] cannot be used in terminal areas since controllers give vectors that do not appear in the flight plan. Moreover, there are many turns and altitude changes that are left to the pilot to execute.

V. Conclusion

This paper presented two trajectory clustering methods and an application to airspace monitoring. The monitoring tool compares the conformance of current flights to identified nominal procedures in real-time. The version of the tool presented in this paper monitors the landings at SFO, but it can easily be modified to monitor any traffic pattern, by modifying the input dataset.

VI. References

References

[1]Joint Planning and Development Office. Concept of Operations for the Next Generation Air Transportation System. Technical report, June 2007.

[2]SESAR Consortium. SESAR Master Plan, deliverable 5. Technical report, April 2008.

[3]US Department of Transportation; Federal Aviation Administration. Administrator's fact book. Technical report, December 2006.

[4]C. Piciarelli, GL Foresti, and L. Snidara. Trajectory clustering and its applications for video surveillance. In *IEEE Conference on Advanced Video and Signal Based Surveillance, 2005. AVSS 2005*, pages 40–45, 2005.

[5]A. Dahlbom and L. Niklasson. Trajectory clustering for coastal surveillance. In *Proceedings of the 10th International Conference on Information Fusion*, 2007.

[6]J.G. Lee, J. Han, and K.Y. Whang. Trajectory clustering: A partition-and-group framework. In *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, pages 593–604. ACM New York, NY, USA, 2007.

[7]J.G. Lee, J. Han, and X. Li. Trajectory outlier detection: A partition-and-detect framework. In *Proc. 24th Intl Conf. on Data Engineering*, pages 140–149.

[8]M. Vlachos, G. Kollios, and D. Gunopulos. Discovering similar multidimensional trajectories. In *Proceedings of the International Conference on Data Engineering*, pages 673–684. IEEE Computer Society Press; 1998, 2002.

[9]A. Eckstein. Automated flight track taxonomy for measuring benefits from performance based navigation. In *Integrated Communications, Navigation and Surveillance Conference*, 2009.

[10]J.B. MacQueen et al. Some methods for classification and analysis of multivariate observations, 1966.

[11]J. Friedman T. Hastie, R. Tibshirani. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, second edition edition, 2009.

[12]S.P. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.

[13]J. Han and M. Kamber. *Data mining: concepts and techniques*. Morgan Kaufmann, 2006.

[14]M. Ester, H.P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining, Portland, OR, AAAI Press*, pages 226–231, 1996.

[15]D.J.C. MacKay. *Information theory, inference, and learning algorithms*. Cambridge Univ Pr, 2003.

[16]S. Budalakoti, A.N. Srivastava, R. Akella, and E. Turkov. Anomaly detection in large sets of high-dimensional symbol sequences. *Submitted for evaluation*, 2005.

[17]S. Budalakoti, A.N. Srivastava, and M.E. Otey. Anomaly detection and diagnosis algorithms for discrete symbol sequences with applications to airline safety. *IEEE Transactions on Systems, Man and Cybernetics-Part C: Applications and Reviews*, 39:101–113, 2009.

[18]Jonathon Shlens. A tutorial on principal components analysis, April 2009.

[19]J. Krozel. Intelligent tracking of aircraft in the National Airspace System. In *AIAA Guidance, Navigation, and Control Conference and Exhibit, Monterey, CA*, 2002.

[20]T.G. Reynolds, J.M. Histon, H.J. Davison, and R.J. Hansman. Structure, Intent & Conformance Monitoring in ATC. *USA/EUROPE Air Traffic Management R&D Seminar*, 2002.

[21]T.G. Reynolds and R.J. Hansman. Conformance monitoring approaches in current and future air traffic control environments. In *21st Digital Avionics Systems Conference*, 2002.

[22]Flight Aware. Track log. http://flightaware.com.

[23]D.L. Iverson and M. Stop. Inductive system health monitoring. In *Proceedings of The 2004 International Conference on Artificial Intelligence (IC-AI04)*.

[24]K.K. Lee, C.M. Quinn, T. Hoang, and B.D. Sanford. Human Factors Report: TMA Operational Evaluations 1996 & 1998. *Moffett Field, CA, NASA Ames Research Center*, 2000.

[25]RH Mogford, JA Guttman, SL Morrow, and P. Kopardekar. The Complexity Construct in Air Traffic Control: A Review and Synthesis of the Literature, 1995.

[26]B. Sridhar, K.S. Sheth, and S Grabbe. Airspace complexity and its application in air traffic management. In *2nd USA/EUROPE Air Traffic Management R&D Seminar*, December, 1998.

[27]D. Delahaye and S. Puechmorel. Air traffic complexity map based on non linear dynamical systems. In *INO Workshop*, 2005.

[28]K. Lee. *Airspace Complexity: Airspace Response to Disturbance*. PhD thesis, Georgia Institute of Technology, 2007.

[29]M. Gariel, E. Feron, and J. Clarke. Air traffic management complexity maps induced by degradation of Communication, Navigation and Surveillance. In *AIAA Guidance, Navigation and Control Conference and Exhibit*, 18 - 21 August 2008, Honolulu, Hawaii.

[30]R.H. Mogford. Mental models and situation awareness in air traffic control. *The International Journal of Aviation Psychology*, 7(4):331–341, 1997.

[31]J.M. Histon, R.J. Hansman, D. Gottlieb, H. Kleinwaks, S. Yenson, D. Delahaye, and S. Puechmorel.

Structural considerations and cognitive complexity in air traffic control. *Digital Avionics Systems Conference, 2002. Proceedings. The 21st*, 1, 2002.

[32]CE Shannon. A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(1):3–55, 2001.