

# DART: A Machine-Learning Approach to Trajectory Prediction and Demand-Capacity Balancing

Esther Calvo Fernández, José Manuel Cordero  
CRIDA, ATM R&D Reference Center  
Madrid, Spain

George Vouros, Nikos Pelekis, Theocharis Kravaris,  
Harris Georgiou  
University of Piraeus Research Center  
Piraeus, Greece

Georg Fuchs, Natalya Andrienko, Gennady Andrienko  
Fraunhofer Institute IAIS  
Sankt Augustin, Germany

Enrique Casado, David Scarlatti, Pablo Costas  
Boeing Research & Technology Europe  
Madrid, Spain

Samet Ayhan  
University of Maryland  
College Park, United States

**Abstract**— The current Air Traffic Management (ATM) system worldwide is managing a high (and growing) amount of demand that sometimes leads to demand-capacity balancing (DCB) issues. These further impose limitations to the ATM system that are resolved via airspace management or flow management solutions, including regulations that generate delays (and costs) for the entire system. These demand-capacity imbalances are difficult to predict in the pre-tactical phase (prior to operation), as the existing ATM information is not accurate enough during this phase. With the aim of overcoming these drawbacks, the ATM system is moving towards a new, trajectory-based operations (TBO) paradigm, where the trajectory becomes the cornerstone upon which the ATM capabilities rely on. This transformation, however, requires reliable information available in pre-tactical phase or, at least, high-fidelity aircraft trajectory prediction capabilities to reach sufficient levels of confidence in the available planning information.

In this scenario, the DART (Data-driven Aircraft Trajectory Prediction Research) project from SESAR 2020 Exploratory Research aims at reaching this goal, by means of machine learning and agent-based modeling methods in two different use cases: trajectory prediction and demand-capacity balancing. This paper presents the machine learning approach followed, as well as the promising results already achieved by the project.

**Keywords**- DCB; data-driven; trajectory prediction; machine-learning; collaborative reinforcement.

## I. INTRODUCTION

### A. DART Project description

Within SESAR 2020 Exploratory Research, DART project has the main objective of exploring the applicability of data mining, machine learning and agent-based models and algorithms to derive a data-driven trajectory prediction capability. In addition to the expectation that data-driven techniques will enhance trajectory predictability and thus, will reduce uncertainty factors during the pre-tactical phase, agent-based modeling methods are expected to provide increased levels of accuracy while considering ATM network effects in the prediction process, which have been rarely introduced by current state-of-the art solutions. For this, the project relies on

extensive, high-quality operational datasets which support the data-driven approach.

Machine-learning algorithms with promising results, will be used for predictions in a collaborative trajectory scenario, accounting for delays due to ATM network effects. Towards an agent based modeling approach for collaborative trajectory prediction, DART leverages reinforcement learning techniques to refine predictions based on (a) potential trajectory predictions and (b) contextual information, in a coordinated way, for groups of trajectories.

In combination, the ultimate goal of DART is to demonstrate how machine learning methods can help in refining single trajectory predictions (learned from surveillance data linked to weather data and other contextual information), considering also cases where demand of airspace use exceeds capacity, resulting to hotspots. This is referred as the Demand and Capacity Balance (DCB) problem, which is the testing use case identified but not the only potential application environment of such techniques. In this work we focus on the way trajectories are affected due to the influence of the surrounding traffic (i.e., considering interactions among individual predicted trajectories), taking into account an important aspect of ATM system complexity by determining delays for affected trajectories at the pre-tactical stage in order to resolve DCB problems, so improving trajectory prediction.

So, this paper addresses (i) the DART research approach both in terms of data-driven trajectory prediction (individual) and agent-based collaborative learning applied to DCB environment in pre-tactical phase, (ii) the positive results obtained so far; and (iii) next steps of project research.

## II. BACKGROUND

### A. Trajectory Prediction

In the context of this work, the first required step is the determination or common understanding of what a trajectory is. Basically, a trajectory is a chronologically ordered sequence of aircraft states described by a list of state variables. The most relevant ones are airspeeds (True Airspeed, TAS, Calibrated

Airspeed, CAS, or Mach Number,  $M$ ), 3D position (latitude  $\phi$ , longitude  $\lambda$  and geodetic altitude  $h$  or pressure altitude  $H_p$ ), the bearing ( $\chi$ ) or heading ( $\psi$ ) and the instantaneous aircraft mass ( $m$ ). A predicted trajectory can be defined as the future evolution of the aircraft state as a function of the current flight conditions, a forecast of the localized weather conditions, contextual information regarding the airspace and a description of how the aircraft is to be operated from this initial state and so on.

Even though there might be available extremely accurate aircraft performance models, such as BADA (Base of Aircraft Data) models released by EUROCONTROL, or weather forecasts, such as those generated by the Global Forecast System (GFS) provided by the National Oceanic and Atmospheric Administration (NOAA), there are intrinsic errors that produce unavoidable deviations between predicted and actual trajectories. Those deviations are the result of representing a stochastic process (prediction of an aircraft trajectory affected by stochastic sources) by a deterministic approach (formulation of a kinematic or kinetic aircraft motion problem).

The concept of data-driven trajectory prediction used in DART project, does not consider any representation of any realistic aircraft behavior, only exploits trajectory information recorded from the ground-based surveillance infrastructure or by onboard systems (e.g., Flight Recorded Data, FDR, or Quick Access Recorder Data, QAR) and other contextual data that may impact the final trajectory, which constitutes an innovative approach. This decoupled solution from the mathematical formulation of the aircraft motion should capture variations of the trajectory that cannot be derived directly from the filed Flight Plans (FPs), both during the pre-tactical and tactical phases. These discrepancies usually come from Air Traffic Control (ATC) interventions to ensure optimum traffic management and safe operations (e.g., delays added due the effect of adverse weather). If these interventions respond to a pattern, big data analytics and machine learning algorithms might potentially identify them once the proper system features are considered.

Thus, the preparation of available trajectory data is crucial to train the algorithms in accordance to the expected performance. Several solutions aim at predicting some aircraft state variables (Target Times) for a representative scenario. The DART goal is to assess generic prediction methods to be applied in different possible scenarios envisioned in the future Trajectory Based Operations (TBO) environment.

### B. Demand Capacity Balancing

The DCB process considers two important types of objects in the ATM system: aircraft trajectories and airspace sectors, and is divided in three phases: Strategic, Pre-tactical and Tactical Phase. The overall objective is to optimize traffic flows according to ATC capacity while enabling airlines to operate safe and efficient flights.

Planning operations start as early as possible - sometimes more than one year in advance. Given that the objective is to protect ATC of overload, this service is always looking for

optimum traffic flow through a correct use of the capacity, guaranteed safety, but also potentially considering other dimensions such as better use of capacity, equity, information sharing among stakeholders and fluency.

In DART research, it is considered the demand-capacity balancing process during the pre-tactical phase. Pre-tactical flow management is applied days prior to the day of operations, and consists of planning and coordination activities. This phase aims to compute the demand for the operations day, compare it with the predicted airspace capacities on that day, and make any necessary adjustments to the flight plans. Since DART goal is trajectory predictions and is focused on a TBO environment, this research considers individual predicted trajectories instead of flight plans, in order to determine the delay that should be imposed on them due to traffic.

At this pre-tactical phase, trajectories are sent to the Network Manager who takes into account sector capacities to detect problematic areas. The main objective of this stage is to optimize efficiency and balance demand and capacity through an effective organization of resources, as much as possible given the accuracy of existing information, which will be greatly improved in a TBO environment. This is done by determining delays at the pre-tactical stage in order to resolve DCB problems. Actually, the current work methodology today is based on a collaborative decision making process between the stakeholders resulting to an Air Traffic Flow Control Management Daily Plan (ADP).

## III. METHODOLOGY

### A. Individual (single) Trajectory Prediction

This section details the big data analytics (BDA) and machine learning (ML) algorithms applied to aircraft single trajectory prediction. The potential three candidates chosen to be assessed throughout the execution of DART have been considered as most suitable and promising techniques to tackle with the problem of data-driven aircraft trajectory prediction. The selection of these three main ML-based approaches is based on the current state-of-the-art, as well as the specifications of the problem. These options are briefly described below:

- **Hidden Markov Models (HMM):** one of the most popular and well-known approaches for studying the state transitions of a system, with applications ranging from time series analysis to speech recognition and medical diagnostics [1][6].

The HMM approach models the evolution of a system by a set of states and transitions between them, each one accompanied by a probability that is typically extracted by analyzing historic data. In the context of TP, the flight route and all the associated information are encoded into discrete values that constitute the HMM states. Then, the trajectory itself is treated as an evolution of transitions between these states, using the raw trajectory data of many flights for training, plus spatio-temporal constraints. Some very recent case studies with this approach show that its results on real data are very promising [7].

- **Trajectory prediction via appropriate kernel-based distance metrics for clustering.** Many approaches to data-driven trajectory prediction based on surveillance data makes use of the flight path itself as the feature vector and test its similarity with other tracks.

In practice, the input vector can include several other properties associated with any trajectory segment but not necessarily derived from the spatio-temporal data of the trajectory. For example, each trajectory segment could be enriched with weather variables, the type of the aircraft, as well as any other semantic information that is relevant.

Similar approaches have been widely used in time series classification, as well as the encoding of local spatial features in image analysis (e.g. see [8]). In trajectory prediction, k-NN classifiers have been used extensively in similar works with trajectory data [7][9][10].

- **Advanced ML models for non-linear regression.** The current state-of-the-art in regression models for raw-data TP includes various methods from the statistical point of view, as well as some ML-based methods. More specifically, several types of localized linear regression, such as Locally Weighted Linear Regression (LWLR) [11] and Locally Weighted Polynomial Regression (LWPR) [12], have been applied to similar problems. As the scale becomes more and more local, the margin of stochastic effects becomes smaller and the regression becomes more accurate. At the same time, there are numerous robust ML algorithms [11][13]-[16] that are much more efficient than standard linear regression or variants. These include kernel-based approaches like Support Vector Machines (SVM) for regression, Decision Tree methods like Classification and Regression Trees (CART), as well as typical soft-margin classification methods like Neural Networks [10][17] that can also be used for regression of the trajectory at different levels and scales.

In this general context, DART addresses the TP task by combining elements of these three basic approaches, in order to produce innovative solutions that are: (a) purely data-driven, (b) efficient and accurate, (c) scalable to very large amounts of input data when applied in the real world (ATM).

The three main approaches, i.e., HMM, clustering and regression, are being developed in parallel and the main focus of work is currently allocated to designing a hybrid clustering/HMM two-phase algorithm for the single TP task. More specifically, clustering is applied as a first processing phase for aircraft trajectories, using a rich set of “annotated” trajectories that include flight plans, localized weather and aircraft properties, which enable modeling in a space higher than the typical 4-D spatio-temporal trajectories domain. Clustering is applied using properly designed distance functions that implement similarity metrics for the complete N-dimensional *enriched* domain, thus providing a more effective matching between “similar” trajectories, not only with regard to their spatio-temporal path but also to local weather, aircraft properties, calendar properties (e.g. weekday), etc. This first phase essentially creates compact groups of aircraft trajectories, typically separating airport pairs (departure

/destination), but also differences in takeoff and landing patterns and severe weather deviations even for the same flight route. Then, each group is represented by one median route or *medoid*, which scales down the complexity of the TP task by at least two orders of magnitude for the next phase (e.g. treating 5-8 medoids instead of 600-800 single trajectories, per month per airport pair).

Next, a hidden Markov model (HMM) is defined and trained for each cluster, using non-uniform graph-based spatial grid and exploiting flight plans as constraints for a parametric model for the HMM emission probability. More specifically, the HMM states are not defined in a uniform grid of typically  $3+k$  dimensions, where  $k$  is the number of additional enrichment parameters (e.g. local weather) [7]. Instead, the waypoints of the filed flight plans of each specific flight are used as the reference points for the HMM states. Each of these points can be matched to the closest point of the medoid of the cluster that each flight is assigned to during the first phase (using the properly defined similarity metric). Thus, each of the individual flight plan is matched waypoint-to-waypoint to its assigned medoid and the true 3-D deviation (Haversine distance) between each pair is formulated probabilistically as the HMM emissions. In practice, instead of using the full-resolution medoid as the baseline, the waypoints of the flight plans are used for setting up the states and emissions for each HMM, one for each medoid. As a result, the complexity of the TP task is further scaled down by at least one more order of magnitude, since e.g. a 600-800 point 5-second sampling trajectory (IFS) is processed as a graph of 11-18 vertices and directional single-edge transitions.

This proposed method has been applied in real radar operational tracks and NOAA weather data for a one-month dataset of flights in Spanish airspace. Using parametric Gaussians as the base for the emissions model and confidence interval estimations for the associated errors, the proposed method exhibits exceptionally low HMM complexity and per-waypoint prediction accuracy of a few hundred meters compared to their filed flight plans submitted prior to the flight. Further enhancements are currently being developed, primarily focusing on enhancing the efficiency, scalability and optimal balance between spatio-temporal and enrichment parameters in the design of similarity metrics for the trajectory matching as k-nearest neighbors (k-NN) clustering with  $k=1$  used Dynamic Time Warping Euclidean distance. Additionally, the regression approach is being investigated independently for extending the current state-of-the-art methods on short-range single TP.

## B. Collaborative Trajectory Prediction: Demand Capacity Balancing

The objective is to demonstrate how agent-based modeling methods can help in trajectory forecasting when planned demand exceeds sectors capacity, taking into account interactions among trajectories, considered as self-interested agents that aim to minimize their delays and resolve demand-capacity imbalances. In this case, regulations of type C (i.e. delays) [18] are applied to the trajectories. This module deals with the trajectories provided by the previous data-driven TP.

Considering the problem specification, let there be trajectories in a set of trajectories  $T$  that must be executed over the airspace in a period of  $p$  time instants (e.g. hours). The airspace consists of a set of sectors  $S$ . Time is divided in intervals  $\Delta t$ , equal to the duration of the Occupancy Counting Period used for measuring demand [19].

Each trajectory is a sequence of timed positions in airspace, which can be exploited to compute the series of sectors that each flight crosses, together with the entry and exit time for each of these sectors. For the first (last) sector of the flight, i.e. where the departure (resp. arrival) airport resides, the entry (resp. exit) time is the departure (resp. arrival) time. Also, there may exist flights that cross the airspace but do not depart and/or arrive in any of the sectors of our airspace: In that case we only consider the entry and exit time of sectors within the airspace of our interest.

Thus, a trajectory  $T$  is a time series of elements of the form:

$$T = \{(s_1, \text{entryTime}_1, \text{exitTime}_1), (s_2, \text{entryTime}_2, \text{exitTime}_2), \dots, (s_m, \text{entryTime}_m, \text{exitTime}_m)\} \quad (1)$$

where  $s_i, i=1, \dots, m$  are sectors in  $S$ .

For instance, considering the trajectories T1, T2 and T4 in Figure 1, these are specified as follows:

$$T1 = \{(s_5, 10:00, 10:20), (s_2, 10:20, 10:45)\} \quad (2)$$

$$T2 = \{(s_2, 10:15, 10:30), (s_6, 10:30, 10:34), (s_7, 10:34, 11:00), (s_{12}, 11:00, 11:27)\} \quad (3)$$

$$T4 = \{(s_{12}, 12:00, 12:10), (s_{15}, 12:10, 12:25)\} \quad (4)$$

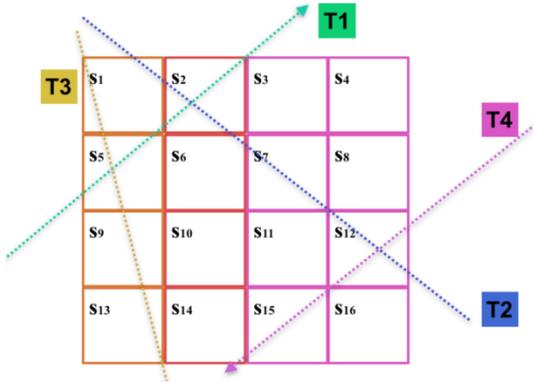


Figure 1: Example of trajectories crossing sectors

This information per trajectory suffices to measure the demand  $D_{s_i,p}$  for each of the sectors  $s_i$  in  $S$  in the airspace in any Occupancy Counting Period  $p$  of duration  $\Delta t$ . Specifically,  $D_{s_i,p} = \|T_{s_i,p}\|$ , i.e. the number of trajectories in  $T_{s_i,p}$ .

In other words, the demand equals to the number of trajectories co-occurring over of a period  $p$  in the same sector. For instance, considering the trajectories T1 and T2 and crossing the sector  $s_2$  in Figure 1, it holds that  $T_{s_2,p} = \{T1, T2\}$  with  $p=[10:00, 10:25]$ . The trajectories in  $T_{s_i,p}$  are defined to be **interacting trajectories** for the period  $p$  and the sector  $s_i$ .

Each sector  $i$  has a specific capacity  $C$  over a period. The aim is to resolve imbalances of sectors' demand and capacity: These are cases where demand  $D$  exceeds capacity  $C$ , for any period  $p$  of duration  $\Delta t$  (occupancy count period duration) in  $H$ , in any of the sectors  $s_i$  in  $S$ .

Subsequently we refer to these cases as demand-capacity imbalance cases, resulting to hotspots.

In case of imbalances for a period  $p$  and sector  $s_i$ , the interacting trajectories in  $T_{s_i,p}$  are defined as hotspot-constituting trajectories: one or more of these trajectories must be delayed in order to resolve the imbalance in  $s_i$ . Given the exploratory research nature of DART, at this stage of research no 4D measures are considered for hotspot resolution, just delays. Enhanced context of research foresees 4D measures.

This problem specification emphasizes on the following problem aspects: (a) agents, corresponding to a single trajectory, need to coordinate their strategies (i.e. chosen options to impose delays) to execute their trajectories jointly with others, taking into account traffic, operational constraints; etc... (b) agents need to explore and discover how different combinations of delays affect the joint performance of their trajectories in terms of the DCB process, given that the way different trajectories do interact is not known beforehand. Agents do not know the interacting trajectories that emerge due to own (and others) decisions, and of course they do not know whether these interactions result to new hotspots; and (c) agents' preferences on the options available may vary depending on the trajectory performed, and are kept private.

In principle, a collaborative multi-agent Markov decision problem (MDP) can be regarded as one agent in which each joint action is represented as a single action. However this may result to a huge state-action space and thus to high computational complexity. So, in order to exploit its various advantages, we use the model of collaborative multi-agent MDP framework [20][21] which assumes:

- The **society** of agents, where each agent  $A_i$  corresponds to a trajectory and is connected to a set of agents (denoted by  $N(A_i)$ ) corresponding to interacting trajectories, resulting to a graph  $(A,E)$ , where  $A$  is the set of agents and  $E$  the edges between them.
- A **time step**  $t=1,2, \dots, H$ , where  $H$  is the total number of time instants considered.
- A **local state** per agent  $A_i$  at time  $t$ , comprising state variables that correspond to (a) the delay imposed to the trajectory  $T_i$ , ranging to the sets of options assumed by  $A_i$ , and (b) the number of hotspots in which  $A_i$  is involved in (for any of the sectors and time periods). Such a state is denoted  $s_i^t$ . The joint state  $s_{\{i,j\}}^t$  of agents  $A_i$  and  $A_j$  at time  $t$  is the tuple of the state variables for both agents. A global state  $s^t$  at time  $t$  is the tuple of all agents' local states.
- The **local strategy** for agent  $A_i$  at time  $t$ , denoted by  $str_i^t$  is the action that performs at that specific point: An action for any agent at any time point, in case the agent is still on ground, may be, either impose a delay or not. Thus, at each time point the agent has to take a binary decision. When the agent flies, then it just follows the trajectory. The location (i.e. sector) of that agent at any time point can be calculated by consulting its trajectory. The joint strategy of a subset  $Ag$  of agents executing their trajectories at time  $t$ , is a tuple of local strategies, denoted by  $str_{Ag}^t$ . The joint strategy for all agents  $A$  at time  $t$  is denoted  $str^t$ .

- The **state transition function** gives the transition to the joint state  $s^{t+1}$  based on the joint strategy taken in joint state  $s^t$ .

It must be noticed that although this transition function may be deterministic in settings with perfect knowledge, the state transition per agent is stochastic, given that no agent has a global view.

- The **local reward** of agent  $A_i$ , denoted  $Rwd_i$ , is the reward that the agent gets by executing its own trajectory in a specific joint state of its peers in the society (i.e. the agents) according to the sectors' capacities, and the joint strategy of agent involved. The joint reward for a set of agents specifies the reward received by involved agents by executing their actions in their joint state, according to their joint strategy. It depends on the number of hotspots occurring while the agents execute their trajectories according to their joint strategy in their joint state, i.e. their decided delays, and also according to their preferences on the chosen delays while performing jointly.
- A **(local) policy** of an agent  $A_i$  is a function  $\pi_i: state_i \rightarrow strategy_{\{A_i\}}$  that returns local strategies for any given local state, for  $A_i$  to execute its trajectory. The objective for any agent in the society is to find an optimal policy  $\pi^*$  that maximizes the expected discounted future return for each state  $s$ , while executing its trajectory. This model assumes the Markov property, assuming also that rewards and transition probabilities are independent of time.

The next paragraphs describe three collaborative reinforcement learning methods that take advantage of the problem structure, considering that agents do not know the transition and reward model (model-free methods) and interact concurrently with all their peers.

- **Independent Reinforcement Learners (Ind-Colab-RL):** The independent learners Q-learning variant proposed in [22] decomposes the global Q-function into a linear combination of local agent-dependent Q-functions. Each local  $Q_i$  is based on the local state and local strategy for agent  $A_i$ . Dependencies between agents, and thus the coordination graph, are defined according to the agents' society specified above. It must be pointed out that these dependencies may be updated while solving the problem. Each agent observes its local state variables. A local  $Q$  is updated using the global temporal-difference error, the difference between the current global Q-value and the expected future discounted return for the experienced state transition. As opposite to [22], we use the reward received by the agent, taking into account only the joint state and joint strategy of its neighborhood.
- **Edge-Based Collaborative Reinforcement Learners (Ed-Colab-RL):** This is a variant of the edge-based update sparse cooperative edge-based Q-learning method proposed in [1]. Given two peer agents performing their tasks,  $A_i$  and  $A_j$ , the Q-function is denoted succinctly  $Q_{ij}(s_{ij}, str_{ij})$ , where  $s_{ij}$  with abuse of notation denotes the joint state related to the two agents, and  $str_{ij}$  denotes the joint strategy for the two agents. The sum of all these edge-specific Q-functions defines the global Q-function. In this case this is

approximated using the max-plus message-passing algorithm [2].

- **Agent-Based Collaborative Reinforcement Learners (Ag-Colab-RL):** This is a variant of the agent-based update sparse cooperative edge-based Q-learning method proposed in [1]. As in Ed-Colab-RL method, given two peer agents performing their tasks,  $A_i$  and  $A_j$ , the Q-function is denoted succinctly  $Q_{ij}(s_{ij}, str_{ij})$ , where  $s_{ij}$  denotes the joint state related to the two agents, and  $str_{ij}$  denotes the joint strategy for the two agents.

Further details on these methods are reported in [24].

## IV. TRAINING AND TESTING

### A. Trajectory Prediction

This section summarizes how the aforementioned BDA and ML algorithms are applied to the data-driven trajectory prediction process based exclusively on raw surveillance data.

As described above, the first phase of the proposed approach is based on clustering. For our task, we adopt the SemT-OPTICS approach proposed in [23]. The dissimilarity between two enriched points is decomposed by two parts, one regarding their spatio-temporal dissimilarity and another regarding their dissimilarity on the semantic components.

**Definition 1 (distance between enriched points  $D_r$ ):** Given two enriched points  $r_i$  and  $r_j$ , their distance  $D_r(r_i, r_j)$  is defined by using the following monotone, ranking function with respect to Euclidean distance proximity of their points  $dist_e$ , and the relevancy of their enriched vectors  $dist_v$ :

$$D_{LS}(r_i, r_j) = \lambda \cdot dist_e(r_i, r_j) + (1 - \lambda) \cdot dist_v(r_i, r_j) \quad (5)$$

$$dist_e(r_i, r_j) = \sqrt{w_1 \cdot (x_i - x_j)^2 + w_1 \cdot (y_i - y_j)^2 + w_1 \cdot (z_i - z_j)^2 + \frac{w_2}{w_1} \cdot (t_i - t_j)^2} \quad (6)$$

$$dist_v(r_i, r_j) = 1 - \frac{v_i \cdot v_j}{\|v_i\|^2 + \|v_j\|^2 - v_i \cdot v_j} \quad (7)$$

where the distance proximity of the spatio-temporal components  $dist_e$  is the Euclidean distance in the 4-D vector  $(x, y, z, t)$ . Weights  $w_1$  and  $w_2$  can be defined by the user to weight the spatial versus the temporal dimension. Ratio  $w_2/w_1$  determines the spatial difference that “is equivalent” with one unit time difference (e.g. one second). This ratio can be estimated by the mean speed of all moving objects. As regarding  $maxEuclideanDistance(DB)$  function, it is the coverage in the 4-D spatio-temporal space that acts as a normalization factor. The “semantic” distance  $dist_v$  is measured by Jaccard distance, while  $\lambda \in [0, 1]$  is used to tune the relative importance between the two components.

Based on the Definition above, the distance  $D_R$  between two enriched trajectories is defined as follows:

**Definition 2 (distance between enriched trajectories,  $D_R$ ):** The distance  $D_R$  between two enriched trajectories  $R_i$  and  $R_j$  of arbitrary length (i.e., arbitrary number of enriched points), is given by:

$$D_R(R_i, R_j) = \min \left\{ \begin{array}{l} D_R(T(R_i), T(R_j)) + D_r(r_{i,1}, r_{j,1}), \\ D_R(T(R_i), T(R_j)) + D_r(r_{i,1}, gap), \\ D_R(T(R_i), T(R_j)) + D_r(gap, r_{j,1}) \end{array} \right\} \quad (8)$$

where  $T(R_i)$  denotes the tail of  $R_i$ , namely the enriched points of  $R_i$  after removing the 1-st enriched point of the  $i$ -th semantic trajectory  $(r_i, 1)$ , and gap is a virtual enriched point whose spatio-temporal value is the origin of the 4-D space of the entire dataset, while its “semantic” component corresponds to the zero vector.

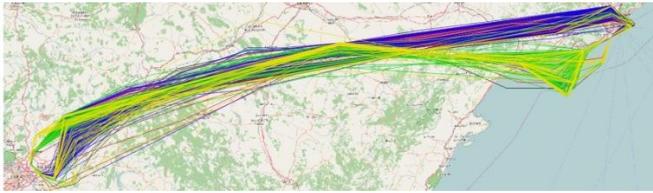


Figure 2: Example of four main clusters (colored) and one cluster of noise & outliers (black) produced in the clustering phase upon the RT (actual routes) using the EDR semantic-aware similarity metric.

Subsequently, in the second phase of the proposed approach, the medoid produced for each cluster is used as the base for designing a Hidden Markov model (HMM).

As described earlier, the states and the corresponding state transition matrix for each cluster are defined by the reference points included in the associated flight plans, while the emissions (not to be confused with fuel consumption related emissions) and the corresponding emissions matrix are defined by a probabilistic model of the pair-wise deviations between flight plans and the cluster’s medoid itself.

Typically, emissions are associated with some property or output from the system that is modeled by the HMM, in the sense that the system shifts between states internally and the emissions are the corresponding observations produced with every such transition, since the states themselves are not observed in a HMM. It is common to assume that the HMM emissions follow a Gaussian distribution in each state, if the number of observations allow such a statistical approximation (more than 30 unbiased samples). Thus, in this approach it is sufficient to have clusters of at least 30 member trajectories.

Using the formulation above, this two-phase hybrid clustering/HMM approach was tested in a benchmark dataset of actual flight trajectories (around 1400 flights). One airport pair was considered from the Spain airspace (Barcelona/Madrid) and each direction was modeled separately, as it involves different takeoff/landing approaches. Each direction and pair of airports will be associated with a separate clustering/HMM model, in order to capture the fine details of each case. For other different city-pairs, the process can be straightforwardly applied, although the identified clusters, the related medoids and the associated HMM will be different.

Figure 3 illustrates the per-waypoint means and confidence intervals for Latitude in cluster 1 as described above. The height of each bounding box is directly linked to the uncertainty associated with producing the maximum-likelihood deviation from the HMM emissions in each reference waypoint, i.e., the difference between the flight plan and the aircraft actual route. As expected, most of the waypoints just after takeoff and before landing have the tightest confidence intervals, while sharp turns are the most difficult to predict. Figure 4 illustrates the distributions of the confidence intervals

(ranges) of Lat/Lon/Alt and inclusion radius R, providing an overview of the statistical uncertainty per dimension and in 3-D for cluster 1. The height of each box, i.e., the size two central quartiles, is directly linked to the statistical uncertainty in predicting each dimension of the pair-wise deviations between flight plans and the cluster medoid.

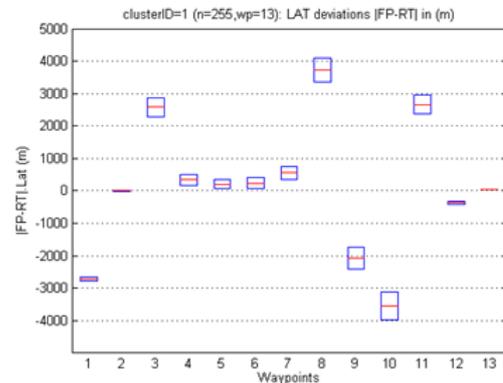


Figure 3: Mean and confidence interval of the Latitude deviations (in meters) within cluster 1 over the minimum common length of flight plans included.

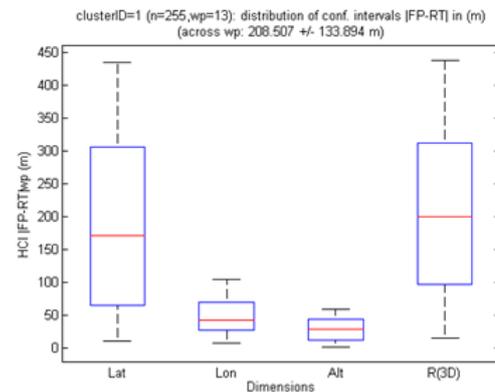


Figure 4: Distributions of confidence intervals (ranges) of Lat/Lon/Alt and radius of inclusion sphere (in meters) within cluster 1 over the minimum common length of flight plans included.

In this sense, flights in cluster 1 (255/703 members) were predicted with accuracy of roughly 183...234 meters upon each reference waypoint of filed flight plans. In contrast, flights in the much smaller cluster 4 (75/703 members) were predicted with accuracy of roughly 595...736 meters. In practice, these implies that for each reference waypoint of the flights in the cluster, there is  $1-\alpha$  probability (here 90%) that the pair-wise deviation in Lat/Lon/Alt between the flight plan and the cluster’s medoid will reside within the corresponding confidence interval of the mean (emission output) and the true 3-D distance of this deviation will be at most R (in meters). In other words, these numbers define how compact is the cluster.

These results demonstrate the robustness and the statistical significance of the proposed hybrid clustering/HMM approach. As described earlier, this method exploits the constraints imposed by the flight plans, i.e., the intended flight path, as well as other “enrichment” parameters such as localized weather and aircraft properties. It should be noted that the proposed method is inherently generic. It does not rely on spatio-temporal grid sizes or resolution, number of

semantic parameters or discretization of them. It does rely on pre-flight constraints, more importantly the flight plan that is associated with each actual route.

### B. Demand Capacity Balancing

There has been performed a series of experiments in order to test and compare the efficiency of the three collaborative Q-learning methods. The efficiency is measured by means of the resulting number of hotspots, the mean delay achieved and the distribution of interacting flights in Occupancy Counting Periods, in conjunction to the number of learning periods needed for methods to compute policies. Simulation scenarios of trajectories crossing airspace have been used based on actual traffic situations (nominal). The airspace comprises a grid of sectors (and capacities). Parameters used in producing the experimental cases are the following: size of the grid of sectors, sector capacity ( $C$ ), number of flights ( $N$ , in this case equal to 100), occupancy count period, total time, and maximum delay.

To evaluate the three approaches in cases of varying difficulty we modify the capacity of sectors, and the number  $m$  of sectors that each flight crosses. Results included here are the most challenging cases in the grid considered, where  $m \in [3, 4]$ . For every capacity value  $C \in [4, 10]$ , 10 experiments were run. This approach will be extended in a further stage to usual sectors being defined around traffic crossing areas.

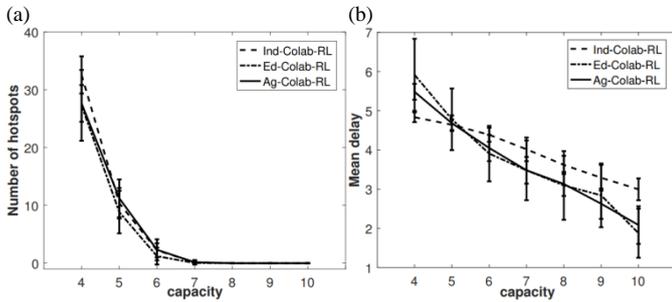


Figure 5: Comparative results: (a) the number of hotspots and (b) the mean delay estimated by each method in terms of various values of sectors' capacity

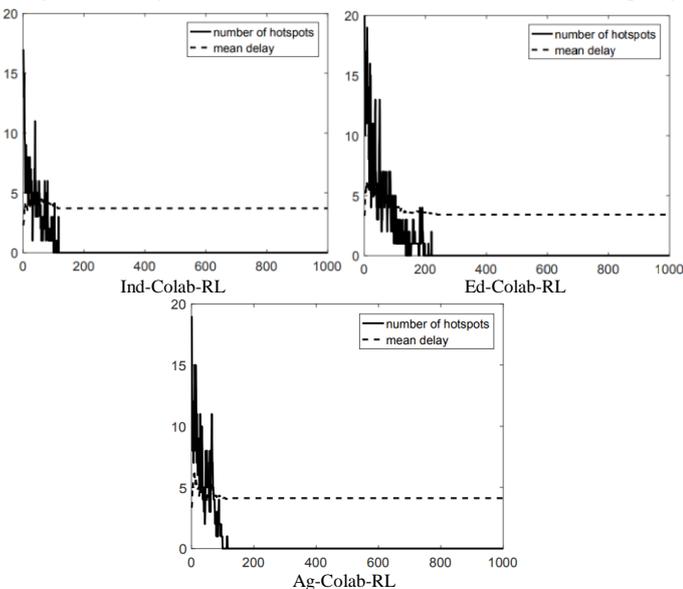


Figure 6: Learning curves received by three methods in a setting considering sectors' capacity equal to 7

Figure 5 shows the mean value and the standard deviation of the final (after learning) number of hotspots, as well as the mean delay for all flights. According to the results, all methods showed a similar behavior in terms of the number of hotspots (Fig. 5.a). A significant improvement in the 'mean delay of all flights' criterion is shown in Fig. 5.b concerning the edge-based and the agent-based collaborative RL approaches.

Figure 6 illustrates an example of the received learning curves by each method, i.e. the number of hotspots and mean delay as estimated in the first 1000 episodes during learning. All methods were able to converge rapidly, achieving strategies with zero hotspots to any sector, and with flights' delay much less than the maximum acceptable delay.

Finally, Figure 7 shows an example of the distribution of interacting flights in terms of Occupancy Counting Periods. This was obtained by measuring the interacting flights to a specific sector in different periods: (a) at the beginning and (b) at the end of learning. As can be seen, the proposed collaborative RL schemes manage to offer strategies with significantly reduced interactions among flight trajectories.

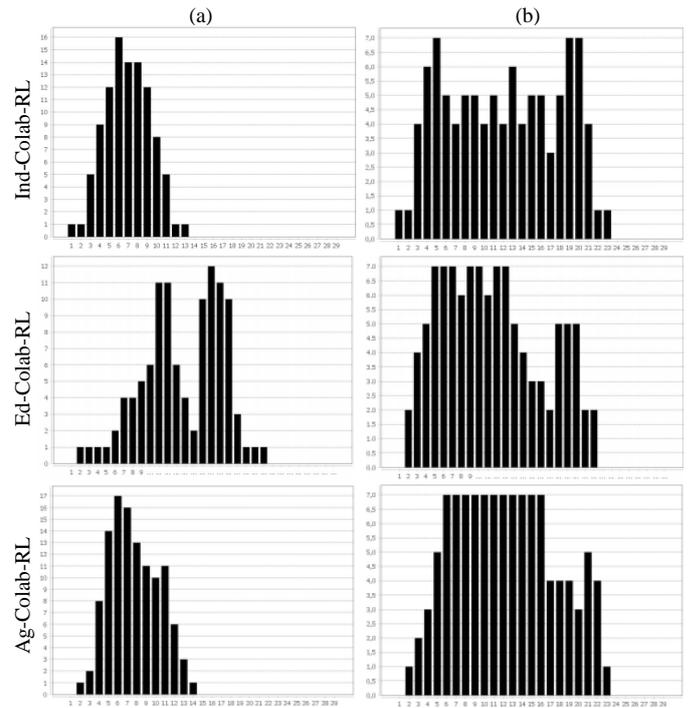


Figure 7: Example of the distribution of interacting flights

The final experiment was created using operational data from Spanish airspace, corresponding to one day in January 2016. The main difference here, regarding the parameters, is that the delays applied are no longer a multiple of the occupancy period, but plain minutes. They are the same parameters as above considerably higher values (for instance, number of flights equals to 3195). In this case results are presented for just one method (Independent Learners), but they are representative of those provided by the different methods.

This change brings the experiment closer to a real world situation, but poses an advanced difficulty for two reasons. Firstly, the maximum delay is much bigger than in the previous

experiment, which means that every agent has many more states to explore. Secondly, a flight can be delayed for less than one occupancy period, as opposed to the previous experiments.

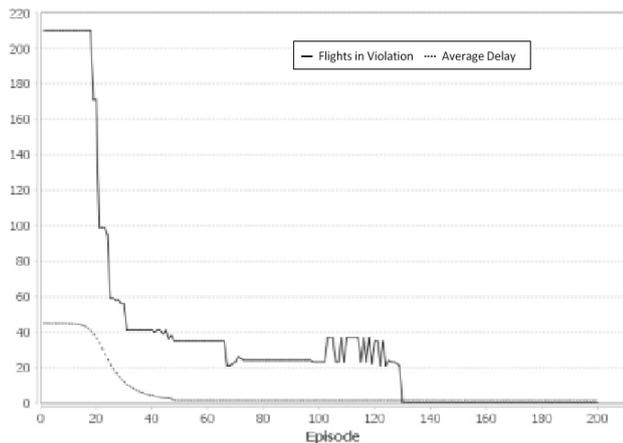


Figure 8: Learning curve received by the Independent Learners

Figure 8 shows the learning curve received by the Independent Learners (Ind-Colab-RL) method, which converges to a solution with average delay close to 0. The exploration-exploitation policy used was the  $\epsilon$ Greedy strategy. The exploration stops at episode 130, where the exploitation begins. Figure 9 shows the initial and final distribution of flights in the sector with two out of seven total hotspots.

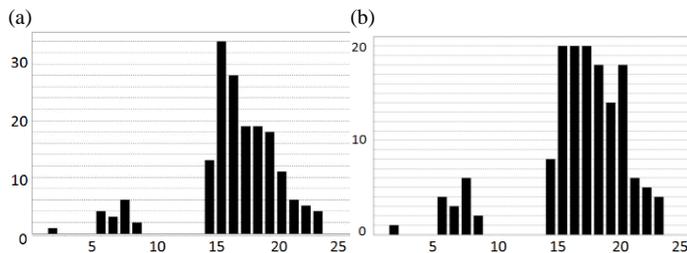


Figure 9: An example of the distribution of interacting flights in Occupancy Counting Periods (a) initially and (b). Finally the sector's capacity is 20

## V. CONCLUSION

The results achieved by DART project so far in terms of application of machine learning algorithms to both trajectory prediction and demand-capacity balancing problems are already very positive and promising, with still room for refinement in subsequent research stages of the project.

Different approaches have been presented, and tested with actual operational data. Future work will focus in improving the problem modeling to include further operational features that help to explore the benefits that such techniques can bring to the ATM domain. The results presented in this paper have already been shared within an Expert group involving including Network Managers, ANSPs and Airspace Users with positive feedback.

## REFERENCES

- [1] Jelle R. Kok and Nikos Vlassis. Collaborative multiagent reinforcement learning by payoff propagation. *J. Mach. Learn. Res.*, 7:1789–1828, December 2006
- [2] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988.
- [3] E. Keogh, “Exact Indexing of Dynamic Time Warping”, *Proc. of the 28th Very Large Databases Conf. (VLDB)*, Hong Kong, China, August 20–23, 2002
- [4] S.-W. Kim, S. Park, W. W. Chu, “An Index-Based Approach for Similarity Search Supporting Time Warping in Large Sequence Databases”, *17th Int'l Conf. on Data Engineering.*, Germany, 2001.
- [5] L.R. Rabiner, "Readings in speech recognition, chapter A: Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition" (Morgan Kaufmann: 1990).
- [6] L.F. Winder, J.K. Kuchar, "Hazard Avoidance Alerting with Markov Decision Processes", PhD thesis, Dept. of Aeronautics and Astronautics, MIT (Cambridge, MA), 2004.
- [7] S. Ayhan, H. Samet, “Time Series Clustering of Weather Observations in Predicting Climb Phase of Aircraft Trajectories”, *IWCCTS'16*, 2016, Burlingame, USA.
- [8] H. Georgiou, et al., “Multi-scaled Morphological Features for the Characterization of Mammographic Masses Using Statistical Classification Schemes”, *Artificial Intelligence in Medicine*. 2007.
- [9] J. Krozel, D. Andrisani, “Intent inference and strategic path prediction”, *AIAA GNC Conf. and Exhibit*, San Francisco, August 2005.
- [10] S. Ayhan, H. Samet, “Aircraft Trajectory Prediction Made Easy with Predictive Analytics”, *Proceedings of ACM SIGKDD*, 2016, pp. 21–30.
- [11] S. Theodoridis, K. Koutroumbas, “Pattern Recognition”, 4th Ed. (Academic Press: 2009).
- [12] Hamed, Mohammad Ghasemi, et al. "Statistical prediction of aircraft trajectory: regression methods vs point-mass model." *ATM 2013*, 10th USA/Europe ATM R&D Seminar. 2013.
- [13] B. Porat, “Digital Processing of Random Signals - Theory and Methods” (Dover Publications: 1994).
- [14] M. Tipping, C. Bishop, “Probabilistic principal component analysis”, *Journal of the Royal Statistical Society - Series B (Statistical Methodology)*, vol. 61, no. 3, pp. 611–622, 1999.
- [15] A. Hyvarinen, E. Oja, “Independent component analysis: Algorithms and applications,” *Neural Networks*, vol. 13, pp. 411–430, 2000.
- [16] I. Todic, P. Frossard, “Dictionary learning,” *IEEE Signal Processing Magazine*, vol. 28, no. 2, pp. 27–38, Mar. 2011.
- [17] Le Fablec, Yann, and Jean-Marc Alliot. "Using Neural Networks to Predict Aircraft Trajectories." *IC-AI*. 1999
- [18] EUROCONTROL, 2017. *ATFCM Users Manual*, Network Manager. Brussels, Belgium.
- [19] EUROCONTROL, 2007. *Hourly Entry Count versus Occupancy Count Relationship - Definitions and Indicators*. EEC Note No. 15/07. Brussels, Belgium.
- [20] M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., USA, , 1994.
- [21] C. E. Guestrin. *Planning Under Uncertainty in Complex Structured Environments*. PhD thesis, Stanford, CA, USA, 2003. AAI104233
- [22] C. Guestrin, M. Lagoudakis, and R. Parr. Coordinated reinforcement learning. In *Proceedings of the ICML-2002 The Nineteenth International Conference on Machine Learning*, pages 227–234, 2002
- [23] N. Pelekis et al. *Simulating our Life Steps by Example*. *ACM Transactions on Spatial Algorithms and Systems*. Vol 2, Issue 3. 2016
- [24] T. Kravaris et al.. *Learning Policies for Resolving Demand-Capacity Imbalances during Pre-tactical ATM*. *MATES*, Germany. 2017