IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received June 20, 2019, accepted July 4, 2019, date of publication July 11, 2019, date of current version August 14, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2928355

A Methodology to Perform Air Traffic Complexity Analysis Based on Spatio-Temporal Regions Constructed Around Aircraft Conflicts

THIMJO KOCA, MIQUEL ANGEL PIERA, AND MARKO RADANOVIC
Logistic and Aeronautics Group, Department of Telecommunications and System Engineering, Autonomous University of Barcelona, 08202 Sabadell, Spain
Corresponding author: Thimjo Koca (thimjo.koca@uab.cat)

This work was supported by the Logistic and Aeronautics Group, Department of Telecommunications and System Engineering, Autonomous University of Barcelona.

ABSTRACT One of the main missions of air traffic management is to guarantee en route safety. This safety is quantified through some minimum separation distance between pairs of flying aircraft. Current systems are human-based, i.e., have human air traffic controllers assuring minimum separation is maintained and in cases a loss of separation is predicted, they take actions to prevent the occurrence of such events. The constant and rapid increment of the air traffic demand is pushing current air traffic control systems to their limits. Development of automatic decision support systems, which can be used to automate, or support aircraft conflict detection and resolution, is considered a possible solution. However, the combinatorial nature of the problem poses several challenges for such a task. Metrics or various analytical procedures to produce information about the complexity of given scenarios can be used to guide the solution search process. In this paper, we present a complexity analysis based on the spatio-temporal interdependencies identified by the use of spatio-temporal regions constructed around the aircraft conflicts.

INDEX TERMS Aircraft conflict, complexity metrics, continuous space-time regions.

I. INTRODUCTION

Air traffic management's (ATM) mission is to make air transportation possible. This is attained by the means of efficient, environmentally friendly, and socially valuable systems, which have safety as their principal goal [1], [2]. In en-route traffic, safety is quantified through a minimum separation distance that need to be maintained between aircraft. A violation of the minimum separation is called a conflict. Current ATM provides minimum pairwise separation through a system with human air traffic controllers (ATCo) at the core of its decision making.

In a context where air traffic demand is increasing in a rapid and continuous manner, a higher level of automation to support conflict detection and resolution (CD&R), is considered a way to make use of the latent capacity, while maintaining safety levels [3].

Complexity in ATM is defined in various ways. It can be seen as a ''measure of the difficulty that a particular traffic situation will present to an ATCo'' [4]. This approach fits within the current, ground-based ATM system, where ATCo are actively resolving conflicts. Delahaye and Puechmorel [5] propose an alternative that is more adequate within the new generation of ATM systems, where automated CD&R will be deployed and the role of the ATCo will be more of a monitoring nature. There, complexity is defined as a measure of the air traffic disorder. Such a definition can fit in the tactical and also the strategic level of ATM. Providing such an information at tactical level, and more specifically on the time horizon after a potential loss of separation has been identified, can allow an automated system, or a human ATCo to adapt its/her solving strategies according to the geometry and complexity of a given scenario and make better use of the capacity. Moreover, in the case of automatic CD&R solvers, given the combinatorial nature of the problem, such an adaptation is crucial in providing a fast, efficient, and safe resolution procedure.

After a potential loss of separation occurs, the aircraft that might be affected by an issued resolution to the conflict need to be identified and further be considered during the conflict resolution procedure. In several proposed resolution methodologies presented in literature, including the Mixed Integer Nonlinear Programming (MINP) family of approaches [6]–[13], the reach sets family of approaches [14]–[16] and various other approaches, such as the light propagation model [17], one of the main issues raised is the

The associate editor coordinating the review of this manuscript and approving it for publication was Haluk Eren.

long time needed to find a resolution as the system scales, i.e. more aircraft need to be considered in the conflict resolution.

In [18] a methodology to identify surrounding conflict and analyze how the number of possible resolutions evolve over time is presented. The proposed approach, however, suffers from two main issues. Firstly, initially filtering through the cluster window, the way it is performed, might not capture all the relevant aircraft. Secondly, the possible resolutions are sought in a discretized space-time, which results in a significant computational effort to treat all possible maneuvers.

In this work, we propose a methodology which overcomes the two drawbacks of [18]. Firstly, to avoid missing any relevant member, we define the proposed structures, cluster and ecosystem, in a hierarchical way. Secondly, to reduce the required computational effort, we introduce a way to construct spatio-temporal regions that contain possible maneuvers an aircraft can perform around a part of its trajectory, given a time interval. We use these regions further to define and express spatio-temporal interdependencies between aircraft at tactical level. We use these interdependencies to identify the aircraft that should be considered by a CD&R solver, or an ATCo during a conflict resolution. The obtained information, which is significantly richer than the one [18] provides, can be used to reduce the number of aircraft that are considered at tactical level by them in a conflict resolution process, to get a clearer insight on the level of complexity they will face in a specific configuration, and eventually make a more efficient use of capacity.

The rest of this paper is organized as follows: section II contains a review of complexity metrics used in literature and some desired characteristics of a complexity metric proposed in [19]. We present in section III the idea of continuous space-time regions. In section IV, we present clusters, as the methodology deployed in various works to identify the relevant aircraft to a conflict resolution process and further, we use the idea of the continuous space-time regions to define some finer sets of relevant members, ecosystems, based on the corresponding spatio-temporal interdependencies. We then give in section V some results of identifying clusters and ecosystems in real and forecast traffic, and perform an analysis of the interdependencies in two detected ecosystems. In section VI, we provide the concluding remarks of this work and some possible future ideas to investigate. Appendix contains some concrete implementations of spatio-temporal regions based on heading alteration.

II. A REVIEW OF CURRENT METRICS & SOME DESIRED CHARACTERISTICS OF A COMPLEXITY METRIC

There are several ATM complexity metrics that are frequently used in literature. Most of them are intended to be used at tatical level, within a sector, and independent of identified conflicts. Our main interest lies in cases where a detected conflict is already detected. However, we choose to present first the following metrics, to provide a broader perspective on approaches to qunatify complexity. Most of the constructed metrics try to relate complexity to the ATCo workload. Aircraft Density (AD) [20], [21], probably the most widely used metric, is such an example. It expresses the number of aircraft flying in a given area, or sector. It is very simple to calculate, but it does not consider other factors that can affect complexity beyond the number of the aircraft and neither the evolution in time of these factors. The former among these issues is solved by the introduction of Dynamic Density (DD) [22], [23], which is a metric that combines various traffic characteristics to produce a scalar complexity value. DD metrics, which can perform this combination in a linear manner [22], or a nonlinear one, using quite often state of the art machine learning techniques [24], still fail to consider evolution over time. The temporal element is introduced in Interval Complexity (IC) [25], a dynamic density-like metric equipped with a time dimension. IC is calculated on a sector level and it is defined as the average over a 5–10 minutes time window of the linear combination of the number of aircraft flying within the sector; the number of aircraft that fly on nonlevel segments; and the number of aircraft that fly close to the border of the sector. So even though a time dimension is added, the factors that are considered are not far from the simplistic AD metric. Fractal Dimension (FD) [26] is another metric, which intents to measure the geometrical complexity. It tries to capture the degrees of freedom of an aircraft given its route and constraints. FD considers time, but comes with the drawback of being dependend on the airspace structure (it requires an airspace structure made up of piece-wise linear segments). A quite different approach is the Input-Output Approach (IO Approach), presented in [27]. The IO Approach quantifies complexity through the effort needed by controller in case a new aircraft is introduced in the airspace. More specifically, a new aircraft is introduced in the sector and in such a way that some loss of separation with the existent aircraft occurs. A Mixed Integer Linear Program is further utilized to solve the conflict. The amount of total deviation introduced in the system in the worst case scenario is taken as the complexity metric. This method comes at a considerable computational cost.

Intrinsic Complexity (InComp) [5], [28] is a methodology that captures the level of disorder of the air traffic and quantifies it in vector fields. It is the only one, among the mentioned metrics, that ignores the relation of complexity to air traffic controller's workload. As the IO Approach however, it comes at some high computational cost.

Considering further works on CD&R, we notice that in all of the approaches mentioned during the introduction sections some way of identify the relevant aircraft to the resolution process, or some complexity information that could guide the solution process are absent. An approach is presented in [29], but being a AD-like approach it is too simplistic and therefore results in a big number of aircraft that are considered during the resolution process.

In [18], after a conflict is detected, the surrounding relevant traffic is identified and some complexity analysis is performed. The relevant traffic identification is divided in two steps. On the first step, a big spatio-temporal window that includes the trajectories (the parts of the trajectories we are interested in) of the conflict aircraft is constructed and all the other aircraft that pass through this are identified. The set of all the identified aircraft plus the two conflict aircraft make what the authors call the ''cluster''. On the second step, using a simulation, discrete maneuvers in space and time of the conflict aircraft are used to deviated them from their original trajectory. The proposed concept of ''ecosystem'' is based on these deviations. In case some deviation leads to a new loss of separation between one of the aircraft involved in the initial conflict with some aircraft AC_3, then AC_3 is declared member of the ecosystem and should be considered during the resolution process. This approach presents two issues. Firstly, taking the cluster cannot guarantee, especially in dense scenarios, that all relevant aircraft will be identified. Secondly, the use of discretization leads to not considering all possible maneuvers, or procedures with a high computational cost.

In this work, we present a methodology that deals with both these issues. Firstly, to not miss any relevant member, we propose the use of clusters and ecosystems in hierarchical structures. Secondly, to decrease computational costs we substitute the discrete maneuvers of the second step by continuous spatio-temporal regions. The reduction in computational cost is significant and therefore more realistic cases, with non-linear trajectories can be treated.

A. DESIRED CHARACTERISTICS OF A COMPLEXITY METRIC

In [19] five characteristics, which are desired within the new generation air traffic management systems, are discussed. The first one states that a complexity metric should account for the traffic dynamics, property which AD and DD do not possess. Independence of the airspace structure, which could make operation in a sector-free context possible, is the second one and IC and FD are the methodologies that fail based on this criterion. The third one pays attention to the fact that different time horizons should have different complexity treatments. None of the introduced metrics can be judged under this criterion, since their scope is at sector level only. The forth property is the independence of the control effort, property which, only the IntComp metric possesses. Since the system will have a higher level of automation, it is not necessary to relate complexity to the air traffic controller's effort. The last property concerns the output format and claims that different output formats are required in different space and time horizons.

III. SAFE SPACE-TIME REGIONS

A. TRAJECTORY DYNAMICS MODEL

We employ a widely used technique to model the aircraft dynamics [6], [11], [30]–[33]. The trajectory of the flight is modeled as a series of 4D (space-time) waypoints. The aircraft is treated as a point mass in a 3D Euclidean space, evolving over time. We obtain its x and y coordinates by applying the stereographic projection [34] on the its latitude and longitude. The z coordinate represents the aircraft's altitude. During the flight, the involved aircraft are assumed to have piece-wise constant velocity between two consecutive waypoints. Moreover, planar maneuverability constraints are modeled by the impose of a maximum angle by which an aircraft can deviate.

Given the above, the flight state variables of the aircraft is specified as (x, y, z, v_x, v_y, v_z), where (x, y, z) are its coordinates and (v_x, v_y, v_z) its velocity components.

B. ASSIGNING CONTINUOUS SPACE-TIME REGIONS TO EN ROUTE AIRCRAFT

The core idea of continuous space-time regions lies in the observation that instead of trying to assign a single trajectory to each aircraft that must maneuver to solve a detected conflict, a space-time region can be given to each one of them.

Mathematically, classical approaches assign to each aircraft a function describing their motion:

$$\begin{cases} x = x(t) \\ y = y(t) \\ z = z(t) \end{cases} \tag{1}$$

Assigning a region instead, as suggested in [35]–[37] could be expressed as:

$$[x(t), y(t), z(t)] \in V(t) \tag{2}$$

where $V(t)$ is a dynamic volume, evolving over time.

Fig. 1 illustrates a safe space-time region assigned to an aircraft in a world with a single spatial dimension (z coordinate) and time. The black continuous curves represent the border of AC_1 safe region (i.e. a guaranteed conflict-free area), the green dashed lines represent feasible legs that AC_1 can fly, the red dashed lines represent legs which might cause a loss of separation, i.e. a conflict and the black dots are feasible, conflict-free waypoints for AC_1.

IV. IDENTIFICATION OF AIRCRAFT RELEVANT TO A CONFLICT RESOLUTION USING SPATIO-TEMPORAL REGIONS

We propose, in this section, a two-step procedure, similar to [18], to detect aircraft that are relevant to a conflict resolution procedure. Different than [18], here the concepts of cluster and ecosystem contain hierarchical structures.

Both the cluster and the ecosystem are sets of aircraft based on pairwise spatio-temporal interdependencies. The concept of spatio-temporal interdependencies was the object of investigation in several European projects, such as PHARE [35], 4DCo-GC [36], PARTAKE [38], and AGENT [37]. Even though the time horizons and purpose of use vary between them, in all of these projects two aircraft were declared interdependent if their assigned space-time regions (which can consist of a single trajectory, an assemble of trajectories, or some continuous space-time region) were closer than they should be (different distance metrics were used in each project).

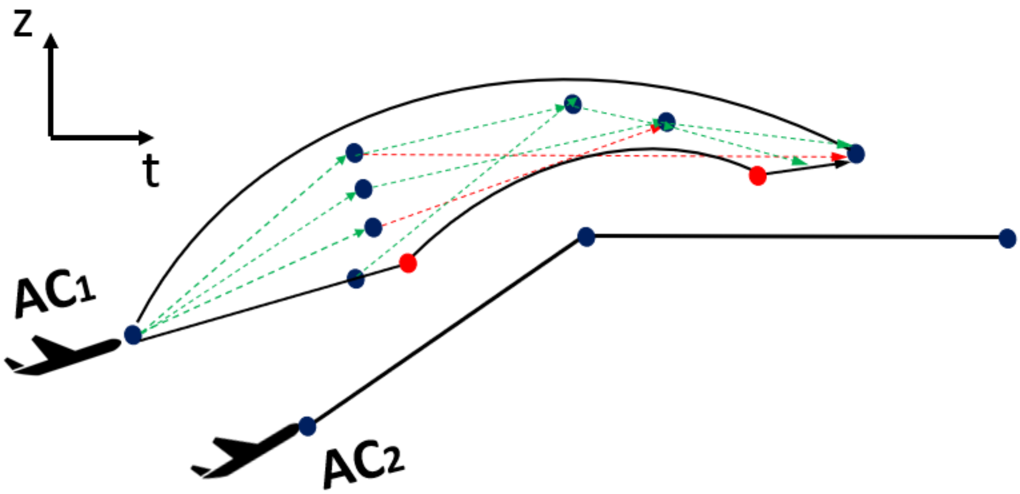

FIGURE 1. Assigned safe region for AC_1 and examples of various legs it can construct (green segments), or not (red segments).

More concisely, let AC_i and AC_j be two aircraft which can be en route, or have a planned flight. Let R_i and R_j be the space-time regions assigned to AC_i and AC_j respectively. Then:

$$AC_i \perp\!\!\!\perp AC_j \iff R_i \cap R_j \neq \emptyset \tag{3}$$

where $\perp\!\!\!\perp$ stands for "interdependent".

Based on the concept of the interdependency, we will construct a hierarchical structure over aircraft. Let there be a potential conflict between at least two en route aircraft. We will denote the set of aircraft involved in this conflict by C and define a hierarchy over the traffic, based on C, and denoted by H_C. The members of the first order of H_C are the members of C. Members of the i^{th} order, where $i \neq 1$, are the aircraft that are not members of a lower order, but have an interdependency with a member of the $(i-1)^{th}$ order. Formally, given that H_C is the defined hierarchy, and $H_C(i)$ is the set of members of the i^{th} order of this hierarchy, we define:

$$\begin{cases} H_C(1) := C \\ H_C(i) := \{\mathrm{AC} \in F | \mathrm{AC} \in H_C^-(i) \wedge (\exists \mathrm{AC}' \in \\ \qquad H_C(i-1) : \mathrm{AC} \perp\!\!\!\perp \mathrm{AC}')\} \end{cases} \tag{4}$$

where F is the set of all aircraft we can consider, and $H_C^-(i) := F \setminus \cup_{j=1}^{i-1} H_C(j))$.

In other words, H_C contains at its first order the preselected set of aircraft C. At its second order it contains aircraft which are not members of the first order, but have at least an interdependency with a member of the first order. In the third order we find aircraft that are not members of the first, or second order, but have at least one interdependency with a member of the second order. The logic goes on recursively.

A. CLUSTER IDENTIFICATION

Concrete implementations of the hierarchical traffic idea, defined above, depend on concrete implementation of the spatio-temporal regions we use to define the interdependencies. Let F denote the set of aircraft we will consider and AC_i and AC_j be two aircraft in it. Let further B_i and B_j be two spatio-temporal boxes constructed respectively around the trajectories of AC_i and AC_j, big enough to contain all the possible locations of the aircraft after feasible maneuvers are possibly performed. Then:

$$AC_i \perp\!\!\!\perp_{cl} AC_j \iff B_i \cap B_j \neq \emptyset \tag{5}$$

where $\perp\!\!\!\perp_{cl}$ stands for "dependent on clustering level".

In Fig. 2 an example is given, by representing the horizontal components (x and y) of the spatio-temporal boxes. In this example, the blue box is the box of AC_1, the red one of AC_2 and the green one of AC_3. There is a cluster-level interdependency between AC_1 and AC_2, another one between AC_2 and AC_3, but no cluster-level interdependency exists between AC_1 and AC_3.

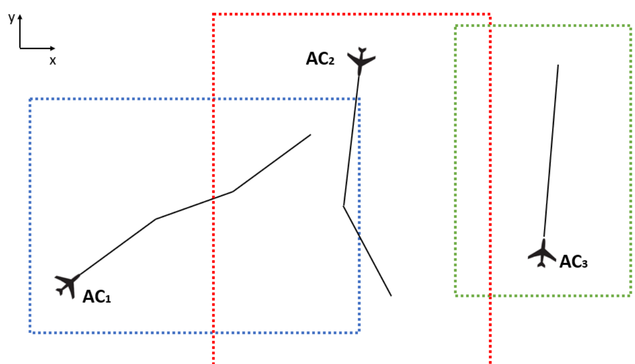

FIGURE 2. Example to illustrate cluster pairwise interdependencies.

Based on the given interdependency definition (5), and the hierarchy definition (4), we will construct the hierarchical structure over aircraft, called cluster. Let there be a potential conflict between at least two en route aircraft. We will denote the set of aircraft involved in this conflict by C. Members of C then, are members of the first order of the cluster. Members of the i^{th} order, where $i \neq 1$, are the aircraft that are not members of a lower order, but have an interdependency with a member of the $(i-1)^{th}$ order. Formally, given that Cl is the set of cluster members and $Cl(i)$ is the set of cluster members of the i^{th} order, we define:

$$\begin{cases} Cl(1) := \; C \\ Cl(i) := \{\mathrm{AC} \in F | \mathrm{AC} \in Cl^{-}(i) \wedge (\exists \mathrm{AC}' \in \\ \qquad\qquad Cl(i-1) : \mathrm{AC} \perp\!\!\!\perp_{cl} \mathrm{AC}')\} \end{cases} \tag{6}$$

where $Cl^{-}(i) := F \setminus \cup_{j=1}^{i-1} Cl(j))$.

In the example of Fig. 2, if we assume that AC_1 is a conflict aircraft,[1] but AC_2 and AC_3 are not, then AC_1 will be a member of the first order, AC_2 a member of the second order and AC_3 a member of the third one.

B. ECOSYSTEM IDENTIFICATION

The cluster structure, from the way its interdependencies are constructed, is too conservative. The use of spatio-temporal boxes, even though it is computationally efficient, results in an overestimation of the complexity of a given scenario. In order to provide a better estimation of the scenario complexity and the interdependencies between the aircraft, we introduce here the "ecosystem" structure. The difference between a cluster and its corresponding ecosystem lies on the nature of their interdependencies. While to construct a cluster, spatio-temporal boxes were used, as a mean of approximating the results of performing some maneuvers, to construct an ecosystem we will check the actual possible maneuvers.

Let F be the set of all aircraft we will consider. Let $tr(AC_k)$ denote the original trajectory of an aircraft AC_k in F, and M_k the set of possible maneuvers for AC_k. Furthermore, let $tr(AC_k, m)$ be the modified trajectory of AC_k after performing a maneuver $m \in M_k$. Then, given two aircraft AC_i and AC_j from F and their corresponding set of possible maneuvers M_i and M_j,

$$AC_i \perp\!\!\!\perp_{ec} AC_j \iff \exists m_k \in M_i, \\ m_l \in M_j : \mathrm{conf}(AC_i, AC_i, m_k, m_l) \tag{7}$$

where $\perp\!\!\!\perp_{ec}$ denotes an "interdependency in ecosystem level" and $\mathrm{conf}(AC_i, AC_j, m_k, m_l)$ denotes that aircraft AC_i and AC_j will be in conflict if they perform maneuvers m_k and m_l respectively.

Spatio-temporal regions become relevant in the calculation of interdependecies at the ecosystem level. Instead of considering all possible pairs of trajectories, it is enough to check for an inter-regional conflict, between regions that include all possible maneuvers for each aircraft. This will suffice to consider all physically feasible simple heading maneuvers that a CD&R system can issue. Note that, while both the spatio-temporal regions proposed in the clustering procedure (i.e. the spatio-temporal boxes) and the ones proposed here contain all feasible heading maneuvers, the spatio-temporal boxes claim a lot of extra space which the aircraft cannot actually utilize. These claims are eliminated in the spatio-temporal regions used in the ecosystem, the construction of which is given in appendix.

Given the ecosystem interdependency definition, the ecosystem can be also defined. Let Cl be a given cluster. The aircraft which are first order members of the cluster are also first order members of the ecosystem. Members of the i^{th} order, where $i \neq 1$, are the aircraft that are members of $Cl(i)$ and that exists an ecosystem member of $(i-1)^{th}$ order with which they have an interdependency at the ecosystem level. Formally, if Ec is the set of ecosystem members and $Ec(i)$ the set of ecosystem members of i^{th} order, then:

$$\begin{cases} Ec(1) := Cl(1) \\ Ec(i) := \{\mathrm{AC} \in Cl(i) | (\exists \mathrm{AC}' \in Ec(i-1) : \mathrm{AC} \perp\!\!\!\perp_{ec} \mathrm{AC}')\} \end{cases} \tag{8}$$

An ecosystem can be clearly defined directly in a given traffic, without the need of a corresponding, predefined cluster. Because on the implemtation used for this work, to identify pairwise interdependencies we use a brute force approach and check all the pairs, the use of a cluster comes with high computational advantages. The introduction, however, of more sophisticated techniques, like the hextree subdivisions [39] can vanish this advantage and even increase further the computational efficiency. In such a scenario, the ecosystem structure can be constructed directly from the traffic, without the cluster structure being a mediator.

Note that, because of the more complex structure of the ecosystem windows, which are changing shape in time, a schematic representation, similar to the one provided in Fig. 2 for the cluster case, could be misleading. We therefore, decide not to provide one.

1) THE ECOSYSTEM STRUCTURE AS A COMPLEXITY INFORMATION PROVIDER

From the way we construct it, an ecosystem demonstrates the ability to identify latent capacity, if there is some, and guide the search for a resolution. If, furthermore, regions of other maneuvers are constructed, with velocity module adjustment for example, altitude change or other complex maneuvers, the most promising among them can be identified.

Moreover, even though the ecosystem is not a metric (it is not a function by which we can obtain a numerical value), from the five proposed properties, the ecosystem structure and the related interdependencies account for the air traffic dynamics and provide independence of the airspace structure and the control effort. The other two properties, are irrelevant since the construction is proposed for tactical level only. The ecosystem structure therefore, can provide some valuable complexity information regarding conflict resolution.

[1] We assume AC_1 is in conflict with aircraft AC_0. AC_0 is not depicted in Fig. 2.

V. SIMULATION RESULTS

A. DATA AND THE PARAMETERS USED

We evaluate the method using traffic data from Eurocontrol's Demand Data Repository II (DDR II). We present two scenarios, one with real historical traffic from 12.02.2019 and another one with synthetic dense traffic generated using Eurocontrol's STATFOR.[2] The predicted traffic is a high density estimate on the date 10.09.2021. In both scenarios, we consider en route conflicts around London TMA, more specifically aircraft flying between 51.01 and 52.05 degrees of latitude, −0.85 and −0.14 degrees of longitude and above flight level 245 (i.e. above 24500 feet).

Conflicts are detected using the methodology proposed in [40] with an added filter to discard the false positive results, and conflicts with a conflict interval larger than one minute. As soon as a conflict is detected, the originally planned trajectory of each involved aircraft is filtered from five minutes before entering the conflict interval until two minutes after exiting the conflict interval. The time length of the cluster is therefore seven minutes plus the length of the conflict. Fig. 3 illustrates the planar cluster window. For each geographic direction (North, South, East, West)

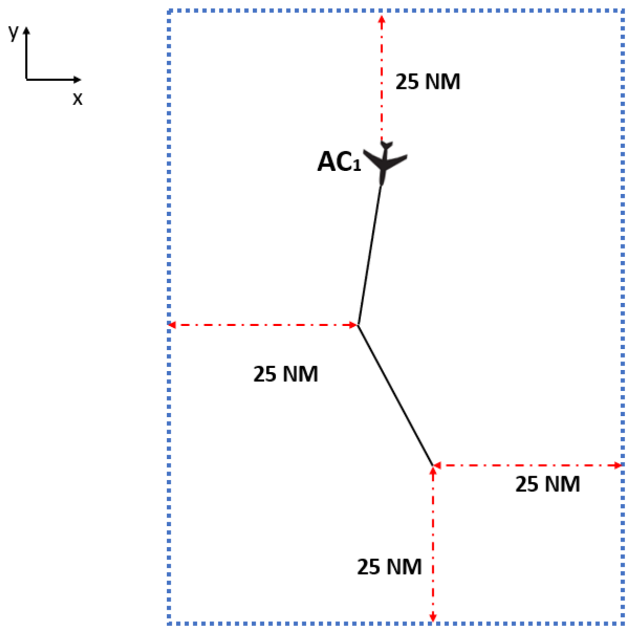

FIGURE 3. The planar part of the cluster box.

we take the furthest waypoint and add to it 25 nautical miles (NM). The vertical cluster window, which is not illustrated in Fig. 3 for clarity reasons, is formed in a similar manner with a margin of 3000 feet.[3] We extend both clusters and ecosystems up to a third level. On ecosystem level, the time window is kept the same as in the cluster case and a maximum deviation angle of 60 degrees is used to construct the aircraft's space-time regions. Three regions are build for each aircraft, region "left" bounded by aircraft's original trajectory and its leftmost trajectory that the aircraft can achieve, given its configuration, region "right" bounded by aircraft's original trajectory and its rightmost trajectory it can achieve given its configuration, and region straight including only the original trajectory of the aircraft.

To check if a pair of aircraft has an interdependency at cluster, or ecosystem level, a brute-force approach was used and each pair of aircraft was investigated. This procedure exhibits a computational complexity of $\Theta(n^2)$, where n is the number of aircraft we consider. This performance can significantly be improved by the use of more sophisticated techniques, as the hextree subdivision method [39], which displays a computational complexity of $\Theta(n)$ on usual setup and $\Theta(n^2)$, in the unlikely case of having near identical trajectories.

B. COMPARISON OF CLUSTERS TO THE ECOSYSTEMS

3306 conflicts, out of which 3275 short enough,[4] were detected in the historical traffic scenario and 12848 conflicts, out of which 12746 short enough, were detected in the futuristic traffic scenario. An analysis was performed for all the detected cases, however 2000 conflicts were sampled from each scenario and used to construct the histograms provided in this section.

Fig. 4 contains the histogram of the identified clusters in the scenario with the historical traffic, while Fig. 5 contains the histogram of the corresponding ecosystems. In the clustering step, 7.25% (i.e. 145 out of 2000) of the clusters have more than forty members each and 37.35% of them (i.e. 747 out of 2000) have only two members. The corresponding ecosystems have a maximum of six members, found in three cases, and 90.3% of the total amount (i.e. 1806 out of 2000) have only two members. The reduction is drastic.

The futuristic scenario exhibits a similar reduction tendency, where in the clustering case we have 7.25% (i.e. 145 out of 2000) of the clusters with more than forty members and 34.9% (i.e. 698 out of 2000) of them with two members only. The corresponding ecosystems have a maximum of seven members in ten cases and 88.55% (i.e. 1771 out of 2000) of the total with only two members.

What the clustering structure does essentially is count the amount of aircraft flying in a given space-time volume relevant to a detected conflict. In this sense, the number of cluster members is a measure similar to the dynamic aircraft density. An interesting observation, in both scenarios, is that even though the clustering structure can have significantly large amount of members, counting the possible dependencies that can come by performing feasible maneuvers of the aircraft through the ecosystem structure, reduces drastically the amount of aircraft that need to be considered in a conflict resolution procedure.

[2]STATFOR is a tool that predicts future traffic demand based on the current one and airport constraints.

[3]Both spatial values were chosen based on experimentation.

[4]With a duration shorter than two minutes.

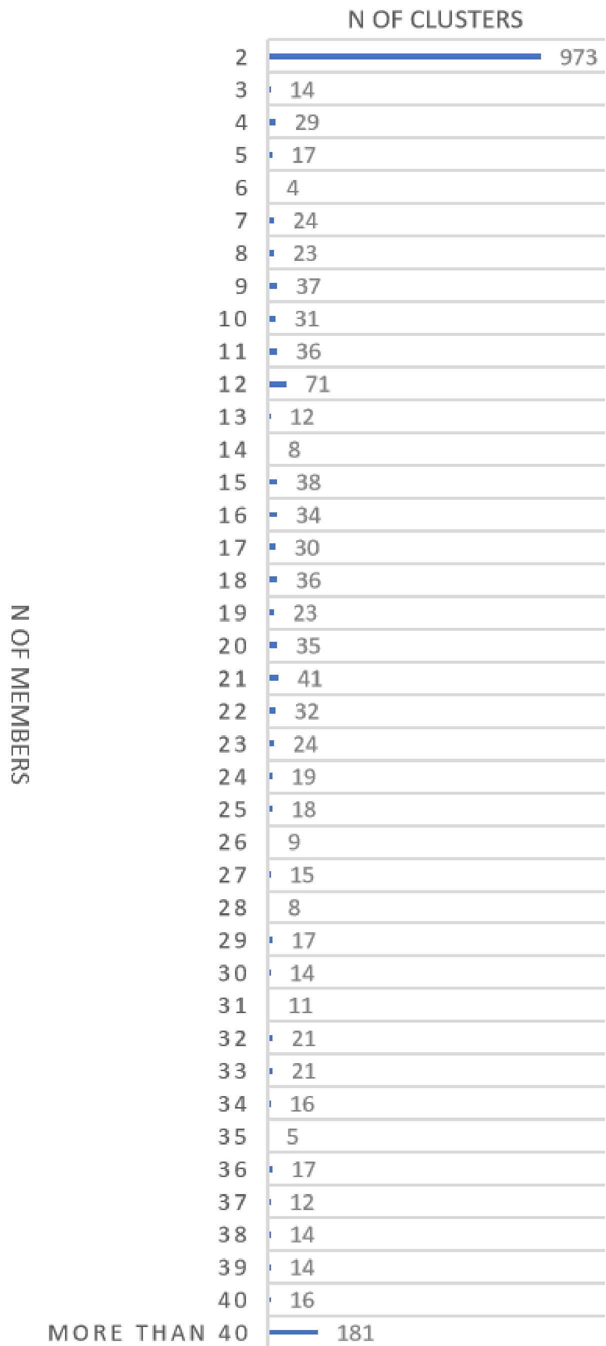

FIGURE 4. Histogram of clusters identified in the historical traffic.

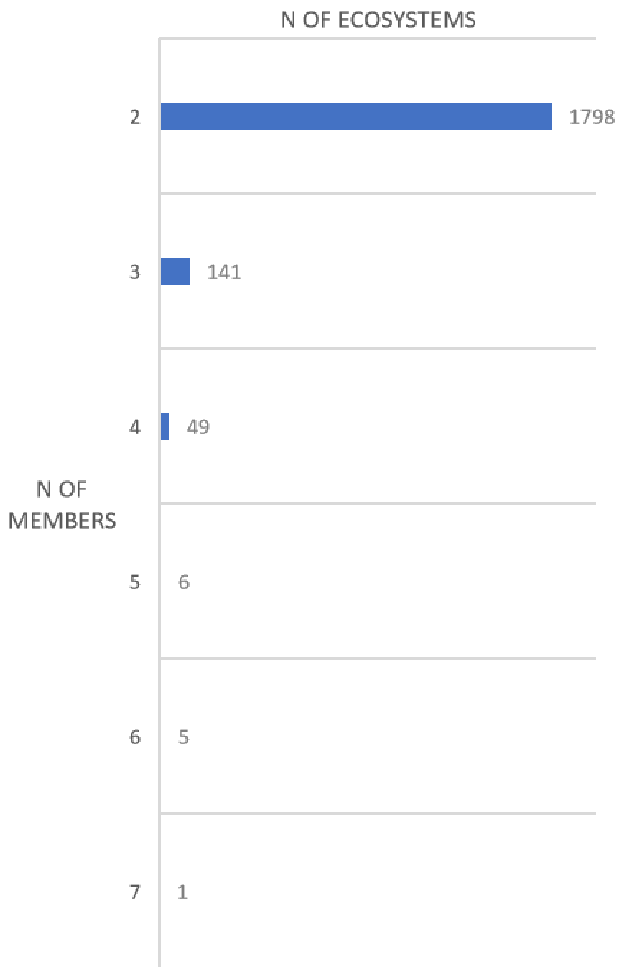

FIGURE 5. Histogram of ecosystems identified in the historical traffic.

C. ANALYSIS OF AN ECOSYSTEM FOR EACH TRAFFIC

Going further than the number of members of each structure, the dependencies between various regions used in the ecosystem can give us a deeper insight. Two ecosystems, each with 7 members are analyzed and compared for this purpose. A similar analysis can be also performed at the cluster level. However, the over-conservative approached by which the clusters are constructed, would result in minimal insights.

The first ecosystem is taken from the historical traffic. The corresponding cluster contains 39 members, 2 of which of the first order, 23 of the second, and 14 of the third one. The members of the ecosystem are classified in 2 members of the first order, 3 of the second, and 2 of the third one. Fig. 8 shows the graph of interdependencies for ecosystem 1. Each node of the graph represents an aircraft of the ecosystem. Two nodes are connected via an edge if there exist an interdependency between them at ecosystem level, i.e. if there exist a pair of feasible maneuvers, one for each aircraft, that can put them into conflict. For example in a certain scenario a left maneuver of AC_1 could result in a loss of separation with AC_3. In this case, AC_1 and AC_3 are declared interdependent and the vertices that represent them in the graph of the interdependencies are connected through an edge. We see that even though there are 7 members, they are loosely connected to each other and the chances of finding an efficient solution quickly are high. An interesting observation comes if we constrain ourselves to seek solutions only by maneuvering the ecosystem members of the first order and letting the rest of the members maintain their original trajectories. This case is depicted in Fig. 9. Interestingly, AC_2 can perform a maneuver without inducing a new conflict.

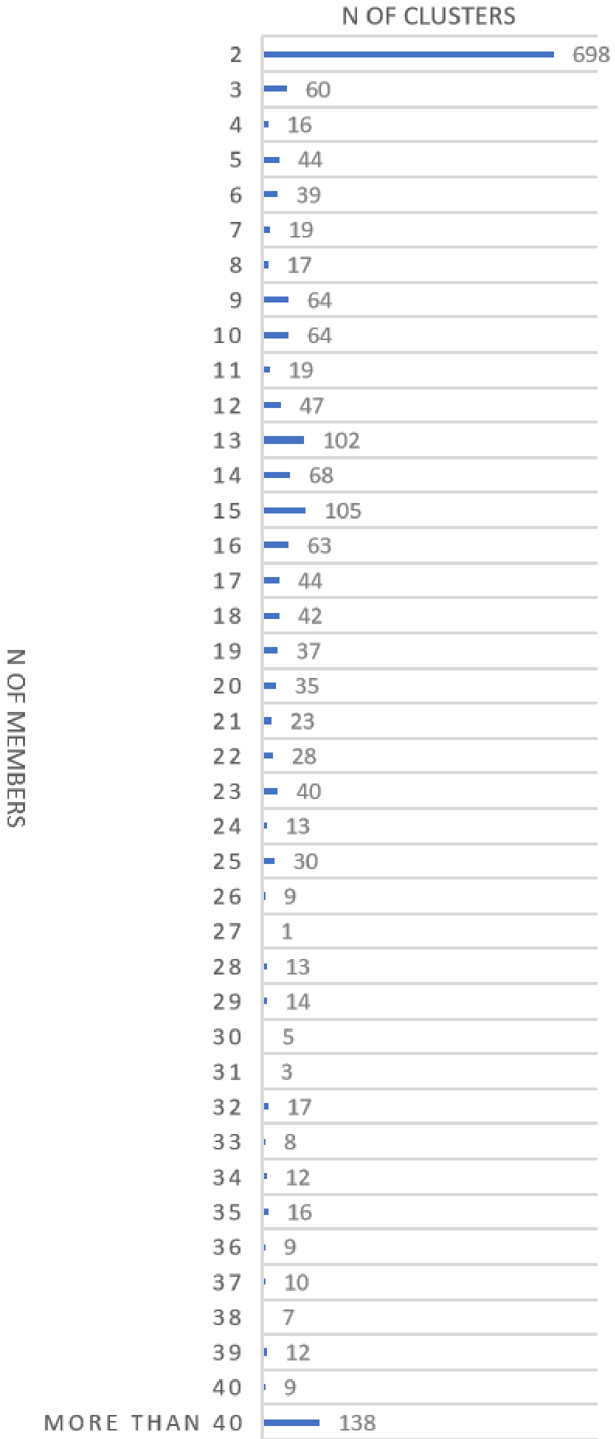

FIGURE 6. Histogram of clusters identified in the futuristic traffic.

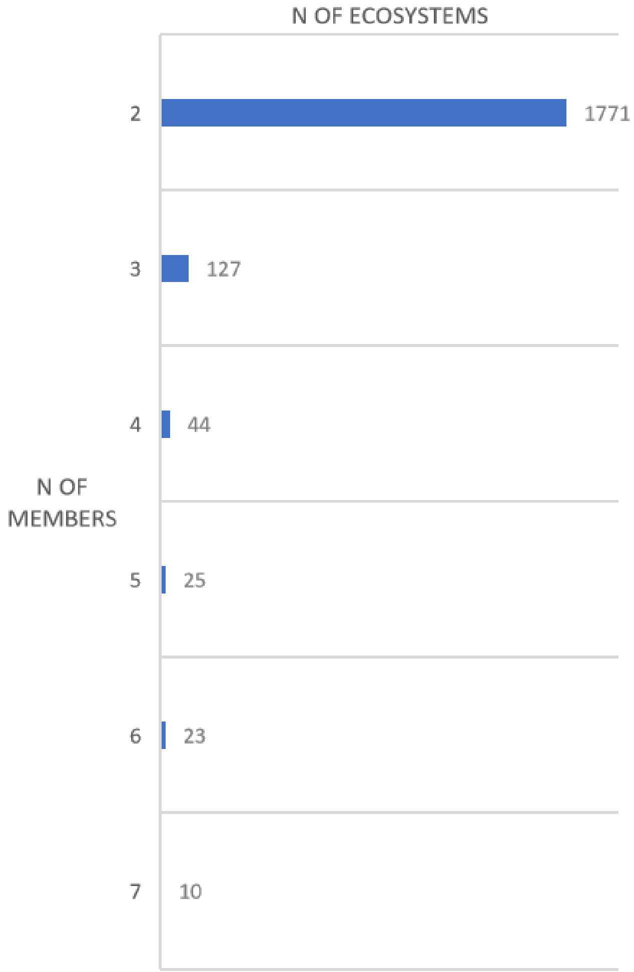

FIGURE 7. Histogram of ecosystems identified in the futuristic traffic.

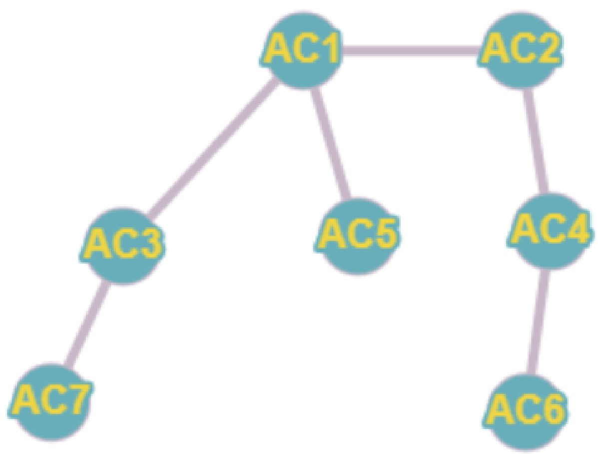

FIGURE 8. Graph of ecosystem 1 in the historical traffic.

The situation is different for ecosystem 2, the ecosystem extracted from the futuristic scenario, even though it also has 7 members. The corresponding cluster contains 35 members (i.e. less than the 39 members in the first case), 2 of which are of the first order, 18 of the second, and 15 of the third one. The members of the ecosystem are classified in 2 members of first order, 4 of the second order, and 1 of the third order. Moreover, as shown in Fig. 10 the graph of interdependencies is denser, almost fully connected. Constraining to single maneuver solutions, reduce the number of aircraft we need

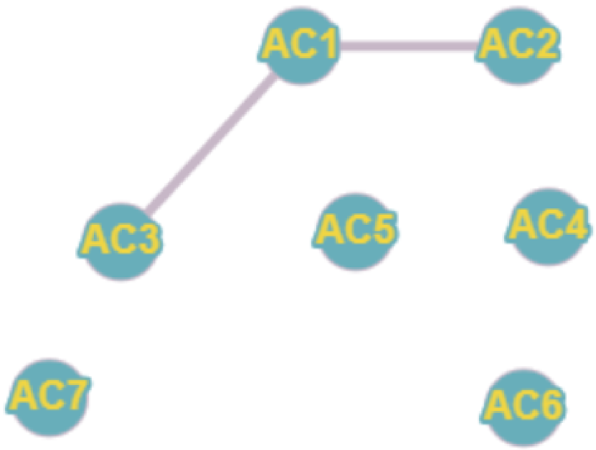

FIGURE 9. Graph of ecosystem 1 constrained to single-maneuver-solutions.

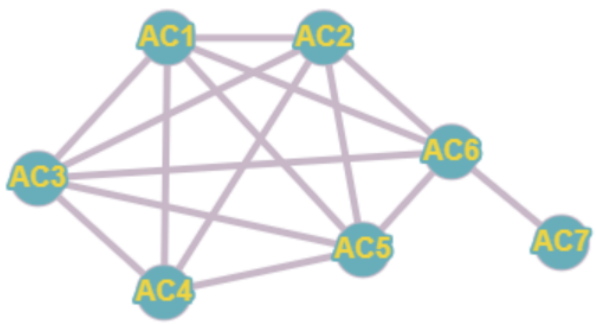

FIGURE 10. Graph of ecosystem 2.

to consider, but still leaves the rest of the members heavily connected, Fig. 11. Therefore, solving this ecosystem is not as straightforward as solving the previous one. The above comparative analysis hints the power that spatio-temporal interdependencies, based on only feasible maneuvers of the aircraft, have.

FIGURE 11. Graph of ecosystem 2 constrained to single-maneuver-solutions.

D. VALIDATION OF THE SPATIO-TEMPORAL REGIONS

Velocities of different directions are used for different parts of the regions, therefore, we implemented the spatio-temporal regions through several sub-regions. A sub-region is a moving polygon, i.e. a set of points, each of which has an initial location, a constant velocity (in module and direction), and the time interval during which it exists.

To validate the construction process of the proposed spatio-temporal regions for the treated scenarios, using an exhaustive generation of points, we simulate possible trajectories and check if they lie inside the defined regions. More precisely, we create a class validation-point, which contains 4D coordinates, a cumulative angle, and the accumulated delay. The 4D coordinates hold the current location of the point during the validation simulation, the cumulative angle holds the sum of all the angles that the validation-point has performed so far in the simulation. An example is given in Fig. 12. The blue line is the initial trajectory. Poin A lies on the initial trajectory, point B lies on the trajectory obtained after a turn of 30 degrees, point C lies on the trajectory obtained after taking another turn of 20 degrees, and point D lies at the trajectory obtained after performing another turn of −30 degrees. In this scenario the cumulative angle value is 0 degrees at point A, 30 degrees at point B, 50 degrees at point C, and 20 degrees at point D.

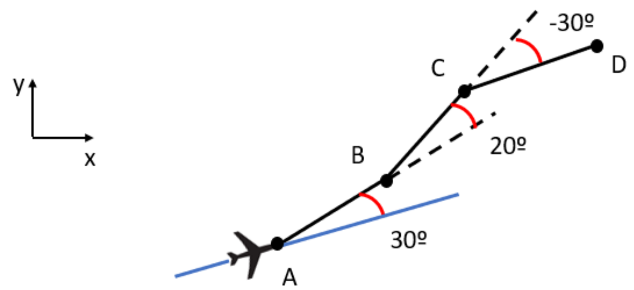

FIGURE 12. Example of an aircraft taking several turns.

To calculate the accumulated delay of the scenario in Fig. 13, we use the following procedure: Let the blue line be the original trajectory of the aircraft and the black one the trajectory after some deviation is performed. Let B_1 be the projection of B in the original trajectory, and B_2 the location of the aircraft if it would have maintained its original trajectory. Let $\vec{V}$ be the original velocity, α the deviation angle, and dt_1 the time interval it took the aircraft to fly from point A to point B. Then the accumulated delay, denoted by d, at point B is:

$$d = \frac{|B_1B_2|}{||\vec{V}||} = \frac{(||\vec{V}|| - ||\vec{V}\cos\alpha||)dt_1}{||\vec{V}||}$$

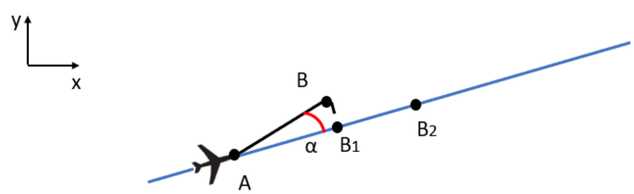

FIGURE 13. Example to illustrate accumulated delay.

The generation of trajectories to be validated works as follows: We start from the initial state of the expanding subregion, i.e. the first 4D point after which we are allowed to perform a maneuver. Discretizing in time by one second and in deviation angles by 5 degrees, we generate possible locations of the point, updating its cumulative angle and its accumulated delay. We perform this step recursively until an end point is reached, or until no more subsequent valid points exist, keeping the cumulative angle less than the maximum allowed (in our simulations it was 60 degrees) and the accumulated delay less than the maximum allowed delay (it can be calculated using the outer most deviation angle). For the regions to be valid, all the generated points need to lie within it.

We performed this validation procedure for all the regions we generated in both simulated traffic scenarios and in all cases the generated points lied within the corresponding spatio-temporal region.

VI. CONCLUSIONS & FUTURE WORK

New automatic decision support systems are essential in the quest to modernization of the airspace to reduce congestion and delays and handle denser traffic flows. A key element in such a system would be a tool which provides information regarding the complexity of given scenarios. We propose in this work a methodology by which we can identify the aircraft that should be considered in a conflict resolution process. This is done using the sets of all possible maneuvers that an aircraft can perform in a given time interval. Based on that methodology, we enrich and reinforce the ecosystem concept, found in [18], and use it to show firstly that even in cases of high aircraft density of current traffic and forecast one, the amount of aircraft that can be affected by a resolution is drastically lower than the amount of aircraft a density-like metric would suggest. Secondly, using the actual interdependencies information, deeper insight can be achieved, regarding the coupling of the involved conflict aircraft with the surrounding traffic aircraft, applying the interdependencies graphs of an ecosystem.

An interesting, in our opinion, path for future work is to investigate how more information can be included in the ecosystem graphs. The methodology that calculates interregional conflicts can provide information about time duration of each conflict and which parts of the regions are involved in the conflict. These pieces of information need to be utilized in the construction of the interdependencies graph. Another issue to be investigated is how these graphs can be used in actual conflict solvers and the impact they might have in computational efficiency terms. While in solvers based on Mixed-Integer Non-linear Programming (MINLP), or Constraint Programming (CP), the graphs can be used to control the constraints the model will have, the utility they can bring in other type of solvers needs to be investigated. A study on how the adaptation of the proposed concept and utilization of the provided information in different solvers, can result in a finite number of complexity classes is also of high interest.

Construction of spatio-temporal regions based on altitude change, or velocity module change only, given piecewise linear trajectories, are a direct adaptation of the provided methodology, therefore, regions based on simple maneuvers are defined. Also enlarging the time horizon can be addressed by changing the initial state of the constructed regions (provide an initial state in the form of a polygon instead of a single point). A topic to be investigated is the construction of such regions based on combination of simple maneuvers, i.e. based on compound maneuvers which are sometimes necessary in practice.

Complexity information needs also to be available at a higher level. Another topic to investigate, therefore, is how interdependencies between ecosystems can be constructed based on the aircraft interdependencies, i.e. how solving one ecosystem can affect the solution of another one. This analysis can certainly be performed at tactical level. It needs to be investigated if it can be performed at strategic level, too. In that case, a definition of an ecosystem which does not lie on a detected conflict might be necessary.

APPENDIX
CONTINUOUS SPACE-TIME REGIONS IMPLEMENTATION BASED ON HEADING CHANGES

Using the trajectory dynamics model and the general continuous space-time region ideas presented in section III, we develop here a concrete implementation of regions that include all feasible heading alteration maneuvers for an aircraft. Our assumptions include constant velocity and linear trajectories of the involved aircraft.[5]

We consider maneuvers with heading alterations only. Maneuvers based on altitude alteration, planar velocity module change, or combinations of some of the above, can be treated in a similar manner.

Let x_0, y_0, z_0, t_0 be the starting 4D waypoint from where we can consider possible heading alterations. Modeling physical limitations of the aircraft by the imposition of a maximal angle of deviation α_{max}, results in a situation where if the aircraft performs a heading change at t_0, at a given time t, it will be found on the arc bounded by the radius forming an angle of α_{max} with the original trajectory and the radius forming an angle of $-\alpha_{max}$ with the original trajectory. The arc will lie on the circle with center $[x_0, y_0]$ and has a radius of $||\vec{v}||(t - t_0)$. Fig. 14 depicts the situation.

For construction reasons, we want to divide this family of arcs into two subfamilies, each one of them being bounded, as depicted in Fig. 15 and 16, by the original trajectory and the limits of the original arcs.

While constructing regions in space-time, there are two constraints we need to initially consider. The first one is that a change of heading can be performed at various time instances, as shown in Fig. 17. The second one is the necessity for a way to return to the original 3D trajectory.

[5]These constraints are relaxed in the subsection where piece-wise linear trajectories and piece-wise constant velocities are assumed.

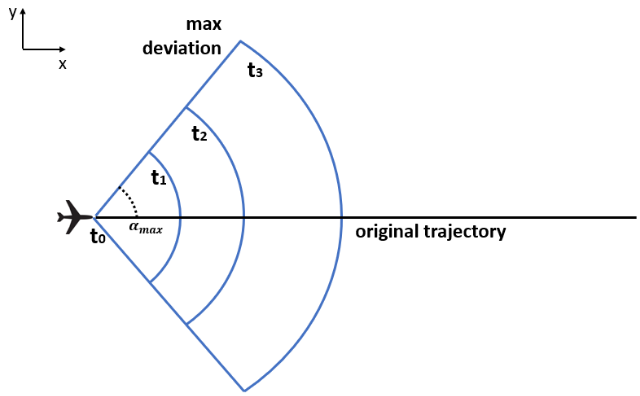

FIGURE 14. Resulting arcs by performing a heading change at t_0.

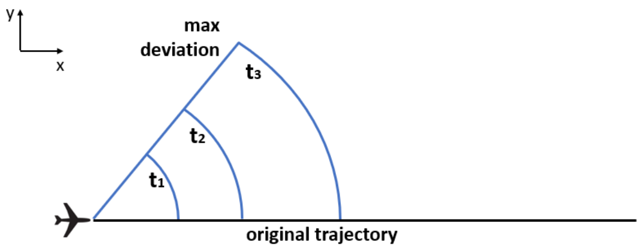

FIGURE 15. Family of upper arcs.

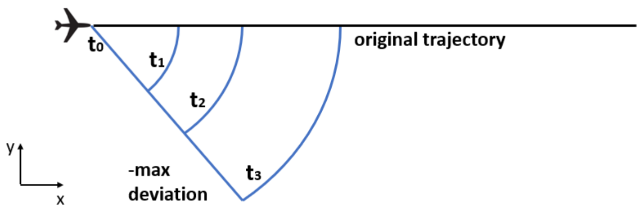

FIGURE 16. Family of lower arcs.

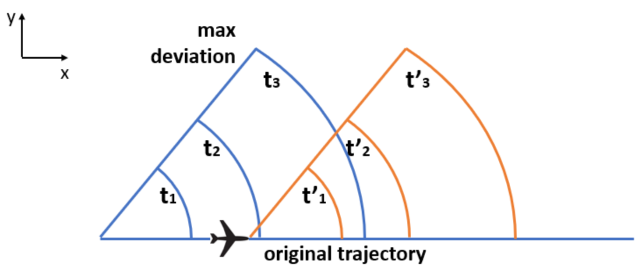

FIGURE 17. A heading change can be performed at various time instances.

Dealing with the first constraint is more straightforward. We can define some linear bounds[6] as the one depicted in Fig. 18. The outer bound can be interpreted as the worst trajectory (in terms of divergence from the original trajectory) that the aircraft can be assigned in the given space-time region.

[6]In a practical situation bounds are used to describe acceptable delays with respect to the RBT. To measure this maximal delay, some global perspective of the system considering the business model of the airline, the TTA's and CTA's, and the traffic conditions is required and this goes beyond the scope of this work.

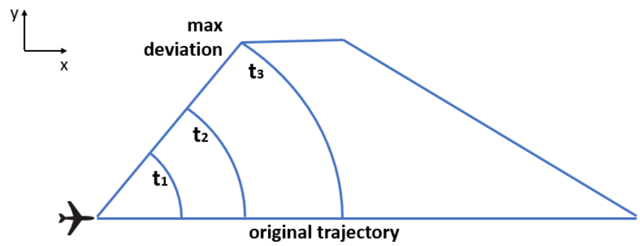

FIGURE 18. Constructed linear bounds.

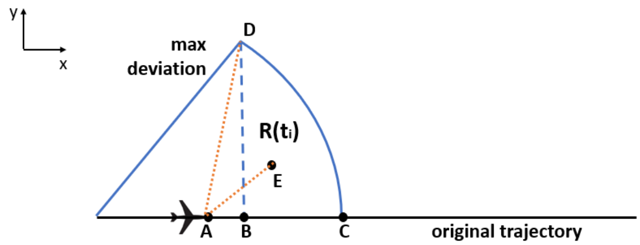

FIGURE 19. A heading change can be performed at various time instances.

To deal with the second constraint, at each time instance, regions as the one shown in Fig. 19 need to be considered. The depicted space region is bounded by the arc $\hat{CD}$ and the segments $[BC]$ and $[DB]$ and is assigned to the aircraft at time instance t_i. t_i is the time when the aircraft would have reached point C, if it would have followed its original trajectory. More formally, we would assert:

Theorem: Let the region $R(t_i)$, bounded by $[DB]$, $[BC]$, and $\hat{CD}$, depicted in Fig. 19, be the space region assigned to the aircraft at the time instance t_i. Let the aircraft be at time instance $t = t_A : t_A \leq t_i$ at the point A, which lies on its original trajectory. Moreover, let E be an arbitrary point that the aircraft want to reach at $t = t_i$ by maneuvering at $t = t_A$. Then $\angle EAC \leq \angle DAC \implies E \in R(t_i)$.

where $\angle EAC$ is the small angle formed by the segments EA and AC. The space-time region, $R(t)$, we want to assign to an aircraft is therefore an evolving arc surface, which can be divided in three sub-regions, Fig. 20. $R_e(t)$ has an arc which moves in the direction defined by the aircraft's initial velocity and is expanding as time evolves, $R_p(t)$ has an arc that only moves in the direction defined by the aircraft's velocity and $R_{sh}(t)$ has an arc that moves and also shrinks as time evolves. Time instances t_1 and t_2 are the instances of change between the regions and defined by the time when the aircraft would

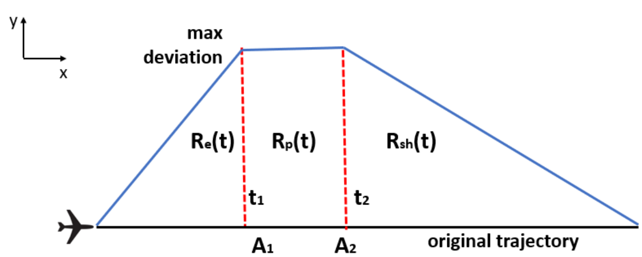

FIGURE 20. The three sub-regions of the defined region.

have reached points A_1 and A_2 respectively, if it would follow its original trajectory.

A. LOSS OF HORIZONTAL SEPARATION BETWEEN REGIONS

In order to provide safety, we need a way of defining and detecting pairs of unsafe regions, i.e. inter-regional conflicts. To define the inter-regional conflict, we propose to approximate the arcs by series of segments. In this way a region $R(t)$ will be approximated by an evolving polynomial $P(t)$.

There are well-known methods in Computational Geometry to find the distance between two fixed polygons [41], however in this section, to avoid time-discretization, we will make use of the fact that the vertices of the polygon evolve in a linear manner. A loss of separation between the evolving polygons can be seen as a loss of separation between pairs of moving segments.

The distance between two arbitrary segments $[P_{1,1}P_{1,2}]$ and $[P_{2,1}P_{2,2}]$ can be expressed as:

$$d([P_{1,1}P_{1,2}],[P_{2,1}P_{2,2}]) := \min_{i,j\in 1,2} d([P_{i,1}P_{i,2}], P_{3-i,j}) \quad (9)$$

which in our case would translate:

$$d([P_{1,1}(t)P_{1,2}(t)],[P_{2,1}(t)P_{2,2}(t)]) := \min_{i,j\in 1,2} d([P_{i,1}(t)P_{i,2}(t)], P_{3-i,j}(t)) \quad (10)$$

where $P_{i,j}(t) = v_{i,j} * t + p_{0,i,j}$ is a vertex of a polygon belonging to AC_i, $v_{i,j}$ its velocity and $p_{0,i,j}$ its initial position.

The closest point of a segment from a point can be one of the segment's end-points, or an inner point of the segment. Let our segment be $[\vec{p_1}(t), \vec{p_2}(t)]$ and the point $\vec{p_3}(t)$, where $\vec{p_i}(t) = \vec{p_{0i}} + t\vec{v_i}$, then the procedure to check if there is a loss of separation between a moving point and a moving segment is the following:

- Check if there is a loss of separation between the point $\vec{p_3}(t)$ and each of the two end-points.
- Check if there is a loss of separation between $\vec{p_3}(t)$ and an inner point of the segment $[\vec{p_1}(t), \vec{p_2}(t)]$.
- Calculate the conflict interval as the union of the three resulting conflict interval

1) LOSS OF SEPARATION BETWEEN TWO MOVING POINTS

The distance between two points is:

$$d(\vec{p_i}(t), \vec{p_3}(t)) := ||(\vec{p_{0i}} - \vec{p_{03}}) + t(\vec{v_i} - \vec{v_3})||$$

To have a loss of separation we require:

$$d(\vec{p_i}(t), \vec{p_3}(t)) < H$$

where H is the required minimum horizontal separation.

This is expression can be easily transformed in a polynomial equation of second order with respect to t and can be solved analytically.

2) LOSS OF SEPARATION BETWEEN A MOVING POINT AND AN INNER POINT OF A MOVING SEGMENT, WHEN THE CLOSEST POINT BETWEEN THE POINT AND THE SEGMENT IS AN INTERNAL SEGMENT POINT

Let $\vec{p_{ij}} := \vec{p_i} - \vec{p_j}$ and $\vec{n}(t) = \vec{p_n} + t\vec{v_n}$ be the normal to the line that includes $[\vec{p_1}(t), \vec{p_2}(t)]$, which means that $\vec{p_n}(t) := [x_{21}(t), y_{12}(t)]$ and $\vec{v_n} := [v_{x21}(t), v_{y12}(t)]$, then the procedure is the following:

- Find the time intervals during which the closest point between the line on which the segment lies and the point is an interior segment point.
  The condition to be satisfied is $sign(\vec{n}(t) \cdot \vec{p_{31}}(t)) \neq sign(\vec{n}(t) \cdot \vec{p_{32}}(t))$,[7] which transforms in a pair of second order polynomial inequalities with respect to time and can be solved analytically.
- Find the time intervals during which there's a loss of separation between the line and the point
  The distance between a point $p = (x_0, y_0)$ and a line $l : (a, b, c$ is:
  $$d = \frac{|ax_0 + by_0 + c|}{\sqrt{a^2 + b^2}}$$
  which translates in our case to:
  $$\frac{|\vec{n}(t) \cdot \vec{p_{31}}(t)|}{||\vec{n}(t)||} \vee \frac{|\vec{n}(t) \cdot \vec{p_{32}}(t)|}{||\vec{n}(t)||}$$
  and can be further transformed in forth order polynomial with respect to time, which can be solved analytically.
- Find their intersections t_1.
- Find the intersections t_2 of t_1 with the interested time interval.
- Find the intersection of t_2 with the time interval during which there is a vertical loss of separation.

B. GENERALIZATION TO PIECE-WISE LINEAR TRAJECTORIES

In this section we present a more general model. We will consider piece-wise linear trajectories with piece-wise constant velocity.

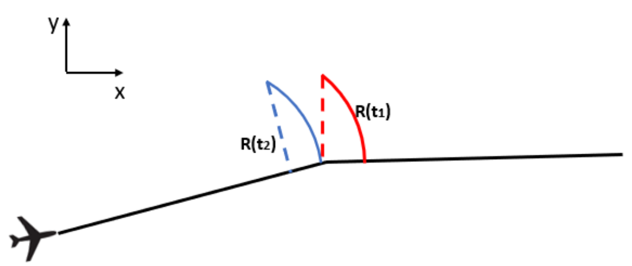

FIGURE 21. The space region before and after the change of the segment.

As can be seen in Fig. 21, to pass the region from one linear segment of the trajectory to the next one, a rotation must be performed if we want to maintain the inner bound on the original trajectory. Such a rotation would introduce

[7] Intuitively we are asking the point to be on opposite sides of the lines, perpendicular to our segment and passing each from one of its end-points.

non-linearities in the velocity of the regions for the transition period, which for now, considering piece-wise constant velocities, we want to avoid. There are various ways to get through such a problem, one of which is by constraining the expanding part of the space-time region to happen only during the time interval of one particular segment. As soon as the next segment starts, the parallel motion begins. And it continues until the segment during which we want the region to shrink, as seen in Fig. 22. Caution should be taken so that time interval during which the region has to shrink is long enough.

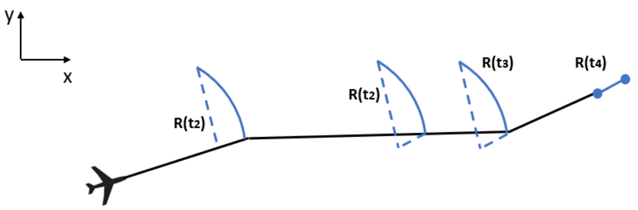

FIGURE 22. The evolution of a space region in a piece-wise linear trajectory.

C. USED DATA STRUCTURES TO REPRESENT MOVING POLYGONS & COMPLEXITY ANALYSIS OF INTER-POLYGON CONFLICT IDENTIFICATION

Each spatio-temporal region is represented as an array of moving polygons. Each one of them representing the spatio-temporal part that an aircraft claims during one of the segments of its trajectory. A moving polygon, as explained above is an array of moving points. A moving point has an initial position, a velocity and a time interval during which it exists.

To detect potential losses of separation between given regions a brute-force approach was used, i.e. each pair of regions is checked. Such a procedure exhibits a computational complexity of $O((nrp)^2)$, where n is the number of aircraft and r is the maximum number of regions each aircraft can have, and p is the maximum number of moving polygons each region can have. The core procedure used is the one that checks for conflicts between two moving polygons. This procedure has a computational complexity $O(s^2)$, where s is the maximum number of segments the polygon can have. The calculation of distances between the moving segments, as explained in the previous subsection, translates to the solution of some polynomial inequalities of second and forth order. The existence of analytic solutions for such cases [42] reduces their computational complexity to $O(c)$, where c is constant.

The used methodologies resulted in satisfactory running times. The use, however, of more sophisticated methods (such as the hextree subdivisions methodology [39]) instead of the brute-force approach, could reduce the computational complexity for usual setups to $O(nrp)$ and $O(s)$ respectively.[8]

[8] In the worst case of near identical trajectories, which is unlikely to happen in practice, the complexities will be $O((nrp)^2)$ and $O(s^2)$.

REFERENCES

[1] G. Di Gravio, M. Mancini, R. Patriarca, and F. Costantino, "Overall safety performance of the air traffic management system: Indicators and analysis," *J. Air Transp. Manage.*, vols. 44-45, pp. 65–69, May/Jun. 2015.

[2] A. Cook, S. Belkoura, and M. Zanin, "ATM performance measurement in Europe, the US and China," *Chin. J. Aeronaut.*, vol. 30, no. 2, pp. 479–490, Apr. 2017.

[3] J. Tang, M. A. Piera, and T. Guasch, "Coloured Petri net-based traffic collision avoidance system encounter model for the analysis of potential induced collisions," *Transp. Res. C, Emerg. Technol.*, vol. 67, pp. 357–377, Jun. 2016.

[4] C. Meckiff, R. Chone, and J. Nicolaon, "The tactical load smoother for multi-sectorplanning," in *Proc. 2nd USA/Eur. Air Traffic Manage. RD Seminar*, Dec. 1998, pp. 1–12.

[5] D. Delahaye and S. Puechmorel, "Air traffic complexity: Towards intrinsic metrics," in *Proc. 3rd USA/Eur. Air Traffic Manage. RD Seminar*, Jun. 2000, pp. 1–11.

[6] L. Pallottino, E. M. Feron, and A. Bicchi, "Conflict resolution problems for air traffic management systems solved with mixed integer programming," *IEEE Trans. Intell. Transp. Syst.*, vol. 3, no. 1, pp. 3–11, Mar. 2002.

[7] M. Christodoulou and C. Costoulakis, "Nonlinear mixed integer programming for aircraft collision avoidance in free flight," in *Proc. 12th IEEE Mediterranean Electrotechn. Conf.*, 2014, pp. 327–330.

[8] A. E. Vela, S. Solak, J.-P. B. Clarke, W. E. Singhose, E. R. Barnes, and E. L. Johnson, "Near real-time fuel-optimal en route conflict resolution," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 4, pp. 826–837, Dec. 2010.

[9] S. Cafieri and N. Durand, "Aircraft deconfliction with speed regulation: New models from mixed-integer optimization," *J. Global Optim.*, vol. 58, no. 4, pp. 613–629, Apr. 2014.

[10] A. Alonso-AyusoLaureano, F. Escudero, F. J. Martín-Campo, and N. Mladenović, "A VNS metaheuristic for solving the aircraft conflict detection and resolution problem by performing turn changes," *J. Global Optim.*, vol. 63, no. 3, pp. 583–596, Nov. 2015.

[11] A. Alonso-Ayuso, L. F. Escudero, and F. J. Martín-Campo, "Exact and approximate solving of the aircraft collision resolution problem via turn changes," *Transp. Sci.*, vol. 50, no. 1 pp. 263–274, Oct. 2016.

[12] A. Alonso-Ayuso, L. F. Escudero, and F. J. Martín–Campo, "An exact multi-objective mixed integer nonlinear optimization approach for aircraft conflict resolution," *TOP*, vol. 24, no. 2, pp. 381–408, Jul. 2016.

[13] S. Cafieri and R. Omheni, "Mixed-integer nonlinear programming for aircraft conflict avoidance by sequentially applying velocity and heading angle changes," *Eur. J. Oper. Res.*, vol. 260, no. 1, pp. 283–290, Jul. 2017.

[14] I. M. Mitchell, A. M. Bayen, and C. J. Tomlin, "A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games," *IEEE Trans. Autom. Control*, vol. 50, no. 7, pp. 947–957, Jul. 2005.

[15] Y. Yang, J. Zhang, K.-Q. Cai, and M. Prandini, "A stochastic reachability analysis approach to aircraft conflict detection and resolution," in *Proc. IEEE Conf. Control Appl. (CCA)* Oct. 2014, pp. 2089–2094.

[16] Y. Yang, J. Zhang, K.-Q. Cai, and M. Prandini, "Multi-aircraft conflict detection and resolution based on probabilistic reach sets," *IEEE Trans. Control Syst. Technol.*, vol. 25, no. 1, pp. 309–316, Jan. 2017.

[17] N. Dougui, D. Delahaye, S. Puechmorel, and M. Mongeau, "A light-propagation model for aircraft trajectory planning," *J. Global Optim.*, vol. 56, no. 3, pp. 873–895, Jul. 2013.

[18] M. Radanovic, M. A. P. Eroles, T. Koca, and J. J. R. Gonzalez, "Surrounding traffic complexity analysis for efficient and stable conflict resolution," *Transp. Res. C, Emerg. technol.*, vol. 95, pp. 105–124, Oct. 2018.

[19] M. Prandini, L. Piroddi, S. Puechmorel, and S. L. Brazdilova, "Toward air traffic complexity assessment in new generation air traffic management systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 3, pp. 809–818, Sep. 2011.

[20] B. Sridhar, K. S. Sheth, and S. Grabbe, "Airspace complexity and its application in air traffic management," in *Proc. 2nd USA/Eur. Air Trafic Manage. RD Seminar*, Dec. 1998, pp. 1–9.

[21] C. Bonaceto, "Cognitive Complexity in Air Traffic Control a Literature Review," *EEC note*, vol. 4, no. 4, pp. 1–80, Mar. 2013.

[22] A. J. Masalonis, M. B. Callaham, and C. R. Wanke, "Dynamic density and complexity metrics for realtime traffic flow management," in *Proc. 5th USA/Eur. Air Traffic Manage. RD Seminar*, Jun. 2003, pp. 1–10.

[23] P. Kopardekar and S. Magyarits, "Dynamic density: Measuring and predicting sector complexity [ATC]," in *Proc. 21st Digit. Avionics Syst. Conf.*, Oct. 2003, p. 2C4.
[24] X. Cao, X. Zhu, Z. Tian, J. Chen, D. Wu, and W. Du, "A knowledge-transfer-based learning framework for airspace operation complexity evaluation," *Transp. Res. C, Emerg. Technol.*, vol. 95, pp. 61–81, Oct. 2018.
[25] P. Flener, J. Pearson, M. Ågren, C. Garcia-Avello, M. Çeliktin, and S. Dissing, "Air-traffic complexity resolution in multi-sector planning," *J. Air Transp. Manage.*, vol. 13, no. 6, pp. 323–328, Nov. 2007.
[26] S. Mondoloni and D. Liang, "Airspace fractal dimension and applications.," in *Proc. 4th USA/Eur. Air Traffic Manage. RD Seminar*, Dec. 2001, pp. 4–7.
[27] K. Lee, E. Feron, and A. Pritchett, "Air traffic complexity: An input-output approach," in *Proc. Amer. Control Conf.*, Jul. 2007, pp. 474–479.
[28] S. Puechmorel and D. Delahaye, "New trends in air traffic complexity," in *Proc. Int. Workshop ATM/CNS (EIWAC)*, 2009, pp. 17–23.
[29] D. Gekht and M. Idan, "Tactical replanning algorithms for four-dimensional contracts in air traffic control," *J. Guid., Control, Dyn.*, vol. 17, no. 6, pp. 1050–1062, Apr. 2014.
[30] J. Hu, M. Prandini, and S. Sastry, "Optimal coordinated maneuvers for three-dimensional aircraft conflict resolution," *J. Guid., Control, Dyn.*, vol. 25, no. 5, pp. 888–900, Sep. 2002.
[31] "Report on AGENT functional and non-functional requirements," Autonomous Univ. Barcelona, Bellaterra, Spain, Tech. Rep., 2016.
[32] C. Munoz, A. Narkawicz, and J. Chamberlain, "A TCAS-II resolution advisory detection algorithm," in *Proc. AIAA Guid., Navigat., Control Conf. (GNC)*, 2013, p. 4622.
[33] E. Koyuncu, M. Uzun, and G. Inalhan, "Cross-entropy-based cost-efficient 4D trajectory generation for airborne conflict resolution," *Proc. Inst. Mech. Eng., G, J. Aerosp. Eng.*, vol. 230, no. 9, pp. 1605–1631. Feb. 2016.
[34] T. H. Kosel, "Computational techniques for stereographic projection," *J. Mater. Sci.*, vol. 19, no. 12, pp. 4106–4118, Dec. 1984.
[35] Eurocontrol. *PHARE*. Accessed: Jul. 29, 2019. [Online]. Available: https://www.eurocontrol.int/phare/public/subsite_homepage/homepage.html
[36] University of Patras. *4DCo-GC*. Accessed: Jul. 29, 2019. [Online]. Available: http://www.4dcogc-project.org/
[37] Autonomous University of Barcelona. *AGENT*. [Online]. Available: http://www.agent-aero.eu/
[38] Autonomous University of Barcelona. *PARTAKE*. [Online]. Available: http://www.partake-aero.eu/s
[39] P.-J. Du, T. Fang, H. Tang, and P.-F. Shi, "Encoding methods of spectral vector in hyperspectral romote sensing image," *J. Shanghai Univ.*, vol. 9, no. 1, pp. 52–57, Feb. 2005. Doi: 10.1007/s11741-005-0104-0.
[40] N. Schefers and J. J. R. González, P. Folch, and J. L. Munoz-Gamarra, "A constraint programming model with time uncertainty for cooperative flight departures," *Transp. Res. C, Emerg. Technol.*, vol. 96, pp. 170–191, Nov. 2018.
[41] M. De Berg, O. Cheong, M. Van Kreveld, and M. Overmars, "Computational geometry," in *Algorithms Applications*. Berlin, Germany: Springer, 2008.
[42] L. Euler and S. L. Hecht, *Elements Algebra*. Scotts Valley, CA, USA: CreateSpace, 2015.

THIMJO KOCA received the bachelor's degree in computer engineering from the Polytechnic University of Tirana, Tirana, Albania, 2014, and the master's degree in mathematical modeling, specialized in stochastic and optimization from a joint program between the University of L'Aquila, L'Aquila, Italy, Hamburg University, Hamburg, Germany, and Autonomous University of Barcelona, in 2016. He is currently pursuing the Ph.D. degree with the Department of Telecommunications and System Engineering, Autonomous University of Barcelona, Barcelona, Spain. His research line is on application of multi-agent systems in air conflict resolution.

MIQUEL ANGEL PIERA received the degree (Hons.) in computer engineering from the Autonomous University of Barcelona, Barcelona, Spain, in 1988, the M.Sc. degree in control engineering from the University of Manchester, Institute of Science and Technology, Manchester, England, 1991, and the Ph.D. degree from the Autonomous University of Barcelona, in 1993. He was a Deputy Director of the Autonomous University of Barcelona Engineering School and a Director of LogiSim (Research Group on Modeling and Simulation of Complex Systems). He is a delegate for Technical Innovation Cluster with the Autonomous University of Barcelona, where he is currently a full-time Professor with the System Engineering Department. He is also active in the innovation sector, contributing as the scientific advisor to the ASLOGIC Company.

MARKO RADANOVIC received the degree in air transportation engineering from the University of Belgrade, Belgrade, Serbia, the master's degree in satellite navigation from the Politecnico di Torino, Torino, Italy, and the Ph.D. degree in air traffic management from the Autonomous University of Barcelona, Barcelona, Spain, where he is currently a Postdoctoral Researcher with the Department of Telecommunications and Systems Engineering. His research interests include automation in air traffic management and control, modeling and simulation of air transportation systems, and air traffic safety.