Guarantying Consistency of Spatio-temporal Regions that Solve Air Traffic Conflicts

Thimjo Koça*, Miquel Angel Piera

Logistic and Aeronautics Group, Department of Telecommunications and System Engineering, Autonomous University of Barcelona, Sabadell, Spain; *thimjo.koca@uab.cat

SNE 32(1), 2022, 15-21, DOI: 10.11128/sne.32.tn.10593
Received: 2020-11-10 (Selected EUROSIM 2019 Postconference Publication); Revised: 2021-09-21; Accepted: 2021-11-01
SNE - Simulation Notes Europe, ARGESIM Publisher Vienna, ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. Separation management together with more efficient conflict detection & resolution are two of the main challenges that Air Traffic Management faces in its quest to modernize itself. This quest for modernization comes as a result of the necessity to adapt to the increment in demand and complexity of the projected future air traffic. Several approaches are proposed to the problem and several sets of properties that they should satisfy. We identify among them robustness, the ability to provide realistic solutions, and consideration of uncertainties the most critical ones. These properties should of course come at a reasonable computational cost. Among the various approaches towards the problem, we believe the ones that try to solve conflicts using spatio-temporal regions are the most adequate base for such systems, because of their unique ability to consider post-decitional uncertainties. In one of the two such methodologies, the construction of such regions, can produce several inconsistencies. We present in this work a methodology by which such inconsistencies can be taken care of.

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]. On en-route traffic, safety is quantified through a minimum horizontal separation distance and a minimum vertical separation distance, that need to be maintained between aircraft. Current ATM provides minimum pairwise separation through a system with human air traffic controllers (ATC) at the core of its decision making.

In the quest to modernization of the airspace system to reduce congestion and delays and handle denser traffic flows, it is essential to develop, deploy, and maintain new decision support systems (DST) automation [3]. The DST fundamental function is a conflict resolution which is to provide aid in the process of resolving intruder's intent.

Such a DST, often called Conflict Detector & Resolver (CD&R), should demonstrate some properties that relate to the ATM's goals. Several properties have been proposed in literature [4]. Among them, the most basic ones are being robust (i.e. being able to always provide solutions), providing realistic solutions, and doing so in a computationally tractable, and resilient manner. Evidently solutions should be realistic, otherwise it will not be possible to fly them. Being computed in tractable manner is essential given the time criticality of the system. Being resilient "forces" the CD&R to consider uncertainties. Such uncertainties can be present before the time that a solution maneuver should be executed, or after that. Uncertainties present before the execution of a solution maneuver are mainly due to measurements errors, or wind uncertainties [5]. Some works count these kind of uncertainties [6, 7, 5]. Some others don't [8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20] with the explicitly, or implicitly expressed claim that such uncertainties in a short tactical time horizon, under the 4D trajectory concept [21], are insignificant, if not completely absent.

Uncertainties present after the time of the execution of a solution maneuver, which can also be human-caused (e.g. a pilot not executing the maneuver at the given fixed time, or with not the exact turning rate, etc.) however are of a higher criticality. The main reason for this is the shorter given time to deal with them. Two works provide solution that consider post-decisional uncertainities, specifically [22, 23].

In [23], the use of spatio-temporal regions throughout the execution of a flight to count for uncertainties is proposed. In cases where the assigned regions could not provide solutions to predicted conflicts, one, or more trajectories were deviated and new regions were built around them.

Alternatively, in AGENT project [22], a methodology is proposed such that spatio-temporal regions are constructed only in situations where a loss of separation is predicted and a conflict resolution process is initialized. In this work we present a methodology to make sure that the regions proposed by the resolution process proposed in AGENT are consistent and safe trajectories can actually be constructed within them.

The rest of this work is organized as follows. In section I the key idea of spatio-temporal regions and the how the consistency problem arises are explained. Section II contains the proposed algorithms. A concrete study case is provided in section III and the concluded remarks are given in section IV.

1 Assigning Continuous Space-Time Regions

1.1 Trajectory dynamics model

We employ a widely used manner to model the aircraft dynamics [10, 24, 22, 25, 15, 26]. 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 [27] 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.

1.2 Continuous space-time regions

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 [23, 22] 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.

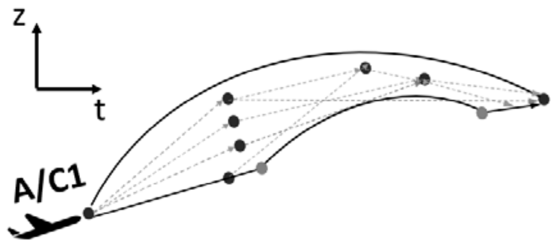

Figure 1: Assigned safe region for AC_1 and examples of various legs it can construct (green segments), or not (red segments).

Figure 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.

1.3 Representation through moving polygons

Getting in to more details, we assume that we are in a well-structured traffic [21], and not under free-flight conditions [28], and also that the original trajectory is the optimal one, actual regions are constructed around the original trajectories of the aircraft.

The possible maneuvers that can be issued to resolve a conflict can be classified in two big categories, simple maneuvers and compound maneuvers. Simple maneuvers come in three flavors, alteration of the horizontal velocity component without a change of its module, alteration of the flight level, or alteration of the module of the horizontal component of the velocity. Compound maneuvers are maneuvers that are made of several simple maneuvers. In our implementation, each region is constructed based on simple maneuvers only, i.e. if we implement a region based on alteration of horizontal velocity direction, within it we can construct only trajectories that are based on horizontal deviations from the original trajectory.

Since we operate in a well-structured traffic, our goal is that after an aircraft deviates to avoid a loss of separation, it should return to its original trajectory. Concentrating in the case of the horizontal deviation this means that we will have at least two changes in the velocity direction, one to go out of the original trajectory and another to go back to it, as illustrated in Figure 2. Because of this last fact, it is convenient to represent a region as a series of moving polygons.

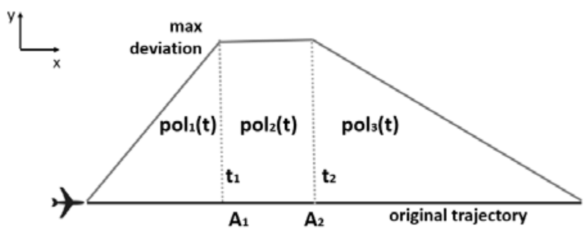

Figure 2: A spatio-temporal region for AC_1 is made up of at least three sub-regions, represented by moving polygons.

By moving polygon here we mean a convex polygon, made up of several spatial points, each one of them traveling in time through a constant (in direction and module) velocity. Note that each point can travel by a different velocity, as long as the convexity of the polygon is maintained throughout its movement.

Further on, since such regions need to be free of conflicts, some cuts might be performed on them. Because different parts of the region, i.e. different polygons have, in general, different conflicts, different cuts will be performed in them, as illustrated in Figure 3. Black lines represent the trajectories of each aircraft. In this scenario, AC_1 is asked to construct its spatio-temporal region, border by its original trajectory, the black segment and the other region limits, represented by the blue lines, to seek for a safe solution to the problematic situation. AC_1's region is made up of three moving polygons, pol_1, pol_2, and pol_3, separated by each other by the green dotted segments. As illustrated by the red dashed segments in the figure, pol_2 is in conflict with AC_2 and pol_3 is in conflict with AC_3. To make therefore this region safe some cuts will be performed resulting in the situation illustrated by Figure 4.

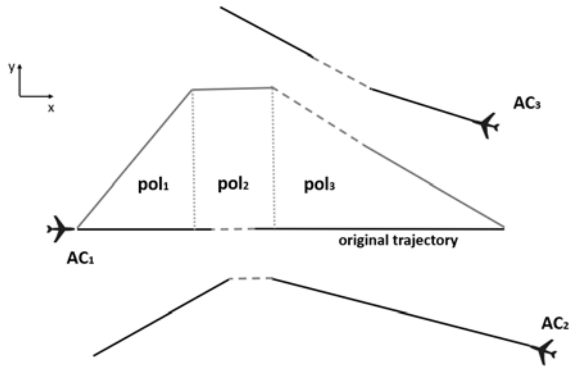

Figure 3: Situation illustrating case when different polygons have different conflicts.

In this figure we can see that AC_1 can travel within pol_2 and reach point A which is considered safe. As soon as it leaves this point however, it is outside the safe region and with no guarantee to be free of conflicts.

The algorithms presented in the following section of this work take these safe, but maybe inconsistent regions and transform them into safe and consistent ones.

2 Region Consistency Algorithm

Algorithm 1 is the main algorithm of this work and illustrated the general procedure that is being followed. Essentially what it expressed is that the intersection of all moving polygons should be calculated and used further to construct the modified, safe, consistent region.

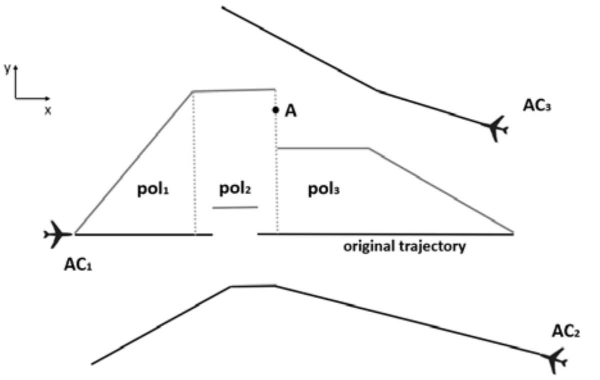

Figure 4: Situation illustrating case when different polygons get cut differently and are therefore inconsistent.

Algorithm 1 Region Consistency Algorithm

let pol_1 be the latest modified moving polygon
for all let pol_2 be a polygon in the range between pol_1 ancestor and the very first polygon of the region **do**
  pol = translate(pol_1, pol_2, backwards)
  $pol_1 := pol$
end for
initialize the array of moving polygons, *arrayPol*
Add pol_1 in *arrayPol*
for all pol_2 between pol_1 successor and the very last polygon of the region **do**
  let t_2 be the time interval during which pol_2 exists
  pol_1 = translate(pol_1, pol_2, forward)
  add pol_1 in *arrayPol*
end for
initialize new region using *arrayPol*

The main step in Algorithm 1 is the translate step, based on Algorithm 2. If the polygons would have been static, standard computational geometry clipping algorithms [29, 30, 31] could have been used to calculate their intersection. Instead the translation, for each pair of consecutive moving polygons, their state at the common time instance is calculated and then the intersection between these static polygons is performed using [29] and used further. More specifically, let pol_n be the moving polygon we are considering and t_n the time interval during which it exists. The static polygon s_n is calulcuated as the state of pol_n ant the beginning of t_n. At the same time, the static polygon s_{n-1} is calculated as the state of pol_{n-1}, i.e. ancestor of pol_n at the end of its time interval t_{n-1}. Note that that the end of t_{n-1} is equal to the beginning of t_n since the two polygons are consecutive. The intersection state s_i between s_{n-1} and s_n is calculated. As a next step the velocities, v_i, corresponding to each of the vertices of s_i are calculated and the new moving polygon pol_i is formed using s_i as its end state, v_i as its set of its velocities and t_{n-1} as its moving time interval.

Algorithm 2 Polygon Translation Algorithm

INPUT: pol_1 to be translated, pol_2 to constrain the translation
if forward translation **then**
  calculate s_1, the end state of pol_1
  calculate s_2, the starting state of pol_2
else
  calculate s_1, the starting state of pol_1
  calculate s_2, the end state of pol_2
end if
find their intersection, i_s
let t_2 be the time interval during which pol_2 exists
v = velocities(i_s, pol_2)
return polygon pol using i_s, v, and t_2

The last algorithm, Algorithm 3 shows how the velocities, v_i corresponding to the verties of s_i are calculated. In it, s_{n-1} is divided into triangles. For each vertex then of s_n the triangle within which it lies is identified. If we denote the vertex under consideration by $\vec{p} = (x,y)$ and the vertices of the triangle within which it lies by $\vec{p}_i = (x_i, y_i)$ for $i \in \{1,2,3\}$ then we have to solve the linear system:

$$\begin{cases} \sum_{i=1}^{3} \alpha_i \vec{p}_i = \vec{p} \\ \sum_{i=1}^{3} \alpha_i = 1 \end{cases}$$

Which is a linear system of three equations with three unknowns and a unique solution[1]. Then using the solution of this system, we can calculate the desired velocity $\vec{v}$ as follows:

$$\vec{v} = \sum_{i=1}^{3} \alpha_i \vec{v}_i$$

where $\vec{v}_i$ are the corresponding velocities for the triangles vertices $\vec{p}_i$.

[1]The guarantees for the uniqueness come from the fact that we are trying to express a point within a triangle as a convex combination of the triangle's vertices.

Algorithm 3 Velocity Initialization Algorithm

INPUT: $state_1$, pol_2
if forward translation **then**
  $state_2$ = state of pol_2 at its starting time
else
  $state_2$ = state of pol_2 at its end time
end if
divide $state_2$ into triangles
initialize *arrayV*
for all p, vertex of $state_1$ **do**
  let tr be the triangle of $state_2$ that contains p and p_i its vertices, $i \in 1,2,3$
  find the coefficient $\alpha_1, \alpha_2, \alpha_3$, s.t. $p = \sum_{i=1}^{3} \alpha_i p_i$
  $v := \sum_{i=1}^{3} \alpha_i v_i$, where v_i are the velocities corresponding to p_i
  add v to *arrayV*
end for
return *arrayV*

3 Study Case

In this section a real case is given. The studied region is a product of solving a conflict found in a traffic simulation based on historical flight data over Europe, taken from DDRII [2]. The predicted flying geometry before the conflict resolution is given in Figure 5. AC_1, with the black trajectory, will loose separation with AC_2, with the blue trajectory. The red segments denote the parts of the trajectories that will be in conflict. AC_3, with the green trajectory, is not in conflict with any other aircraft. However some possible solutions to the original conflict between AC_1 and AC_2 might cause a new conflict with AC_3.

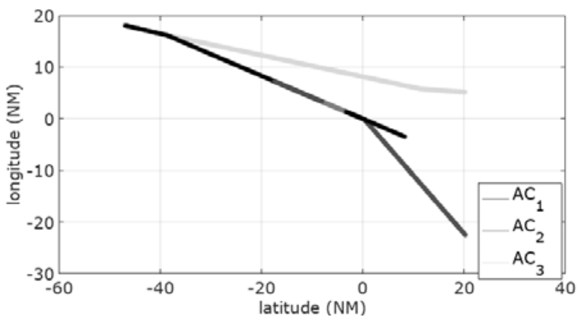

Figure 5: The predicted traffic geometry before the conflict resolution.

[2]DDRII is a data depository, provided by EUROCONTROL with extensive data regarding flights that pass over European sky

The resolution algorithm has chosen AC_2 as the aircraft that will need to maneuver to solve the conflict. Figure 6 contains the initial spatio-temporal region that AC_1 builds to seek a solution for the conflict. This region is not safe, having loss of separation with AC_2, as it contains the original trajectory of AC_1 as one of its borders, and also contains an induced conflict with AC_3. To avoid these conflicts the region needs to be cut, resulting in the shape depicted in Figure 7. There we can see that part of the expanding region leads to points that are not contained in the parallel region and therefore are not guaranteed to be safe. This is taken care by the methodology introduced in Section II and its results can be seen in Figure 8.

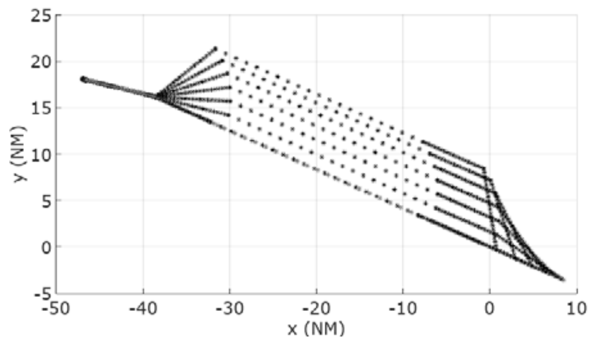

Figure 6: The region that AC_1 initializes to solve the conflict with AC_2.

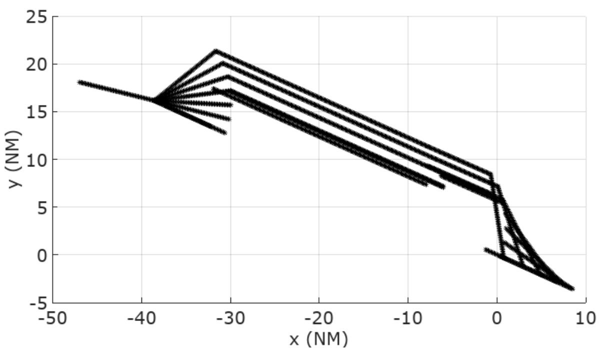

Figure 7: AC_1's cot region containing inconsistencies.

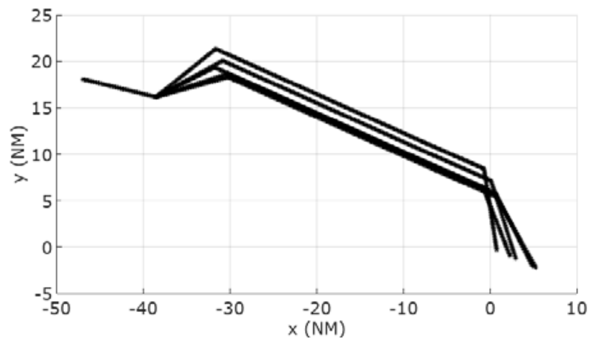

Figure 8: AC_1's consistent region.

4 Concluding Remarks

Separation management together with more efficient conflict detection & resolution are two challenges of importance that modern ATM faces in its attempt to adapt to the increment in demand and complexity. Among the various approaches towards the problem, we believe the ones that try to solve conflicts using spatio-temporal regions are an adequate base for such systems, because of their ability to consider post-decitional uncertainties. While producing such regions, according to the methodology proposed in [22], several inconsistencies can arise. We presented in this work a set of algorithms by which such inconsistencies can be eliminated.

Acknowledgement

This research is supported by the national Spanish project: "EU-TM" (ref. TRA2017-88724-R). Opinions expressed in this article reflect the authors' views only.

References

[1] Di Gravio G, Mancini M, Patriarca R, Costantino F. Overall safety performance of the air traffic management system: Indicators and analysis. *Journal of Air Transport Management*. 2015;.

[2] Cook A, Belkoura S, Zanin M. ATM performance measurement in Europe, the US and China. *Chinese Journal of Aeronautics*. 2017;.

[3] Tang J, Piera MA, Guasch T. Coloured Petri net-based traffic collision avoidance system encounter model for the analysis of potential induced collisions. *Transportation Research Part C: Emerging Technologies*. 2016;.

[4] Pritchett AR, Genton A. Negotiated Decentralized Aircraft Conflict Resolution. *IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS*;1.

[5] Yang Y, Zhang J, Cai KQ, Prandini M. Multi-aircraft Conflict Detection and Resolution Based on Probabilistic Reach Sets. *IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY*. 2017;25(1).

[6] Visintini AL, Glover W, Lygeros J, Maciejowski J, Glover W, Maciejowski J. Monte Carlo Optimization for Conflict Resolution in Air Traffic Control. *IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS*. 2006;7(4).

[7] Yang Y, Zhang J, Cai KQ, Prandini M. A stochastic reachability analysis approach to aircraft conflict detection and resolution. 2014;.

[8] Radanovic M, Eroles MAP. Spatially-Temporal Interdependencies for the Aerial Ecosystem Identification. In: *Procedia Computer Science*. 2016; .

[9] Piera M, Radanovic M, Leal X. Multi-Agent systems for air traffic conflicts resolution by using a causal analysis of spatio-Temporal interdependencies. In: *Simulation Series*. 2016; .

[10] Pallottino L, Feron EM, Bicchi A. Conflict Resolution Problems for Air Traffic Management Systems Solved With Mixed Integer Programming. *IEEE Transactions on Intelligent Transportation Systems*. 2002;.

[11] Christodoulou M, Costoulakis C. Nonlinear Mixed Integer Programming for Aircraft Collision Avoidance in Free Flight. *Tech. rep.* 2004.

[12] Vela AE, Solak S, Clarke JPB, Singhose WE, Barnes ER, Johnson EL. Near real-time fuel-optimal en route conflict resolution. *IEEE Transactions on Intelligent Transportation Systems*. 2010;.

[13] Cafieri S, Durand N. Aircraft deconfliction with speed regulation: New models from mixed-integer optimization. *Journal of Global Optimization*. 2014;.

[14] Alonso-Ayuso A, Escudero LF, Martín-Campo FJ, Mladenović N. A VNS metaheuristic for solving the aircraft conflict detection and resolution problem by performing turn changes. *Journal of Global Optimization*. 2015;.

[15] Alonso-Ayuso A, Escudero L, Martín-Campo F. Exact and approximate solving of the aircraft collision resolution problem via turn changes. *Transportation Science*. 2016;.

[16] Alonso-Ayuso A, Escudero LF, Martín-Campo FJ. An exact multi-objective mixed integer nonlinear optimization approach for aircraft conflict resolution. *TOP*. 2016;.

[17] Cafieri S, Omheni R. Mixed-integer nonlinear programming for aircraft conflict avoidance by sequentially applying velocity and heading angle changes. *European Journal of Operational Research*. 2017;.

[18] Maas J, Sunil E, Ellerbroek J, Hoekstra J. The Effect of Swarming on a Voltage Potential-Based Conflict Resolution Algorithm. 2016;.

[19] Dougui N, Delahaye D, Puechmorel S, Mongeau M. A light-propagation model for aircraft trajectory planning. In: *Journal of Global Optimization*. 2013; .

[20] Mitchell IM, Bayen AM, Tomlin CJ. A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games. *IEEE Transactions on Automatic Control*. 2005;.

[21] Wichman K, Lindberg L, Kilchert L, Bleeker O. Four-Dimensional Trajectory Based Air Traffic Management. 2012; .

[22] team A. Report on AGENT functional and non-functional requirements. *Tech. rep.* 2016.

[23] ONERA, Alenia Aeronautica, CIRA, DLR, ENAC, Erdyn Consultans, Israel Aerospace Industries, Monitor Soft, NLR, Technion Israel Institute of Technology, Thales Communications France, TsAGI, University of Patras. 4DCo-GC.
URL `http://www.4dcogc-project.org/`

[24] Hu J, Prandini M, Sastry S. Optimal Coordinated Maneuvers for Three-Dimensional Aircraft Conflict Resolution. *Journal of Guidance, Control, and Dynamics*. 2002;.

[25] Munoz C, Narkawicz A, Chamberlain J. A TCAS-II Resolution Advisory Detection Algorithm. *AIAA Guidance, Navigation, and Control (GNC) Conference*. 2013;.

[26] Koyuncu E, Uzun M, Inalhan G. Cross-entropy-based cost-efficient 4D trajectory generation for airborne conflict resolution. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*. 2016;.

[27] Kosel TH. Computational techniques for stereographic projection. *Journal of Materials Science*. 1984;.

[28] Free Flight in a Crowded Airspace? In: *Air Transportation Systems Engineering*. 2012;.

[29] Sutherland IE, Hodgman GW. Reentrant polygon clipping. *Communications of the ACM*. 1974;.

[30] Vatti BR. A generic solution to polygon clipping. *Communications of the ACM*. 1992;.

[31] Weiler K, Atherton P. Hidden surface removal using polygon area sorting. *ACM SIGGRAPH Computer Graphics*. 2005;.