AIAA Guidance, Navigation, and Control (GNC) Conference
August 19-22, 2013, Boston, MA

AIAA 2013-4781

Separation-compliant speed control in terminal airspace

Ali Rezaei* Alexander V. Sadovsky† Jason Speyer* Douglas R. Isaacson†

To accommodate the growing air traffic demand, flights will need to be planned and navigated with a much higher level of precision. The Next Generation Air Transportation System (NextGen) stands to benefit significantly in safety and efficiency from such movement of aircraft along precisely defined paths. Air Traffic Operations (ATO) relying on such precision–the Precision Air Traffic Operations or PATO–are the foundation of high throughput capacity envisioned for the future airports. In PATO, the preferred method is to manage the air traffic by assigning a speed profile to each aircraft in a given fleet in a given airspace (in practice known as *speed control*). In this paper, we develop an algorithm, set in the context of a Hybrid Control System (HCS)[1] model, that determines whether a speed control solution exists for a given fleet of aircraft in a given airspace and if so, computes this solution as a collective speed profile that assures separation if executed without deviation. In this paper, uncertainties such as weather are not considered but the algorithm can be modified to include uncertainties. The algorithm first computes all feasible sequences by looking at all pair combinations of aircraft. Then the most likely sequence is determined and the speed control solution is constructed by a backward algorithm starting with the aircraft last out and proceeds to the first out. This computation for all sequences can be done in parallel which helps to reduce the computation time.

Nomenclature

N	The number of aircraft in the fleet
$x_i = x_i(t)$	The arc length coordinate of aircraft i (at time t) along its path
$\mathbf{X}$	The state vector $(x_1, x_2, \ldots, x_N)$ of arc length coordinates
$V_i = \frac{dx_i}{dt}$	The speed of aircraft i
$\mathbf{V}$	The control vector $(V_1, V_2, \ldots, V_N)$ of speeds
$x_i^0 = x_i(0)$	The initial arc length coordinate of aircraft i
$x_i^f = x_i(t_f)$	The end arc length coordinate of aircraft i
$\mathbf{X}^0$	The initial state vector $\left(x_1^0, x_2^0, \ldots, x_N^0\right)$ of arc length coordinates
s_{ij}	The slope of the tangent to a trajectory in 2-dimensional coordinate state space of aircraft i and j
$\mathcal{L}_{a_{ij}}$	The half-plane (6)
$\mathcal{U}_{a_{ij}}$	The half-plane (7)
$\mathcal{A}$	The N-dimensional attainable cone
$\mathcal{C}$	The N-dimensional separation-loss set
$\mathcal{C}'$	The N-dimensional separation-compliant set
$\mathbf{Q}_p$	The matrix of normal vectors of the constraint hyperplanes of the p^{th} segment of the trajectory
$\mathbf{U}_p$	The set of N-dimensional vector along the p^{th} segment of the trajectory

I. Introduction

In today's Air Traffic Operations (ATO), the FAA[2] requires that each aircraft pair is separated by a distance no smaller than the prescribed minimum (the minimum separation requirement) and each aircraft reaches

*UCLA, Department of Aerospace and Mechanical Engineering
†NASA Ames Research Center

its destination at a time as close as possible to the Scheduled Time of Arrival (STA). The responsibility for enforcing such separation (*separation assurance*) today lies on the human Air Traffic Control (ATC). To fulfill this responsibility, controllers issue clearances that not only instruct an aircraft to assume a certain speed (in practice known as *speed control*), but also modify an aircraft's flight path from the original plan (*path control*). However, during the periods of peak air traffic, the task of navigating a fleet of N aircraft to destinations and providing separation assurance is a problem whose running time can be as high as order $N!$,[3] as it requires a choice of an orthant [a] in an N-dimensional Euclidean space. Therefore, the forecast increase in traffic demand will result in a rapid growth of workload for the ATC,[4] hindering NextGen. To afford the growing air traffic demand, flights needs to be planned and navigated with much higher level of precision. In the NextGen aircraft move along precisely defined paths. Also, these paths are parameterized by time so as to assure the required separation distances between pairs of aircraft, in the airspace. Therefore, speed control is the preferred method, as path control generally entails readdressing separation assurance.

There has been recent research[3, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18] into automating the process of computing for each aircraft in a given fleet. A detailed flight path, parameterized by time, in a way that assures separation for the entire fleet at all times, was aimed at designing automated decision support tools for the ATC. The goal was to alleviate the aforementioned workload increase. This research has relied on the modeling framework of Hybrid Control Systems (HCS).[1] In this framework, the arc length coordinate of each aircraft will be mapped to one of state space axes (for more explanation see section II). Hence, a solution for a fleet of N aircraft, each with a precise path prescribed, is ultimately a pair $\mathbf{X}(t)$ and $\mathbf{V}(t)$, where

$$\mathbf{X}(t) = \left(x_1(t), x_2(t), \ldots, x_N(t)\right), \tag{1}$$

and

$$\mathbf{V}(t) = \left(V_1(t), V_2(t), \ldots, V_N(t)\right), \tag{2}$$

of time-dependent, N-dimensional vectors, where $x_i(t)$ is the arc length coordinate of aircraft i along its prescribed path at time t, and

$$V_i(t) = \frac{d}{d\tau} x_i(\tau)\bigg|_{\tau=t}, \quad i = 1 \ldots N.$$

Remark I.1. *In what follows, the coordinates x_i are assumed to correspond to the* standard basis $(1, 0, \ldots)$, $(0, 1, 0, \ldots), \ldots$ *of the Euclidean space of the appropriate dimension (N), and all projections that appear below are orthogonal projections onto a subspace spanned by a subset of this basis. Furthermore, all matrices that represent linear transformation in the state space refer to this basis.*

The problem of finding such a solution for N aircraft in a given airspace can be defined as follow:

Problem I.1. *Consider an airspace with*

- *a fleet of N aircraft, each assigned to a specific path to destination,*
- *each aircraft is subject to the feasible speed range, $V_i \in [V_{i_{min}}, V_{i_{max}}]$, where V_i is the speed of aircraft i,*
- *the minimal required separation distances are considered to be identical between all aircraft pairs.*

One seeks a speed profile $V_i(t)$ for each aircraft, i, such that the corresponding motion $\mathbf{X}(t)$ of the fleet along the assigned paths satisfies the initial condition

$$\mathbf{X}(0) = \mathbf{X}^0 = \begin{bmatrix} x_1^0 \\ \cdot \\ \cdot \\ \cdot \\ x_N^0 \end{bmatrix} \tag{3}$$

and exits the airspace without violating the minimum separation requirement (speed control solution).

[a] In geometry, an orthant is the analogue in N-dimensional Euclidean space of a quadrant in the plane or an octant in three dimensions. In general an orthant in N-dimensions can be considered the intersection of N mutually orthogonal half-planes. By permutations of half-plane signs, there are 2^N orthants in N-dimensional space.

In particular, Ref.[3] gives an algorithm to search for speed control solutions, but there is no guarantee of finding a solution even if one exists. Whether or not it is found, the algorithm will exit after a number of steps that is polynomial in the number of aircraft.

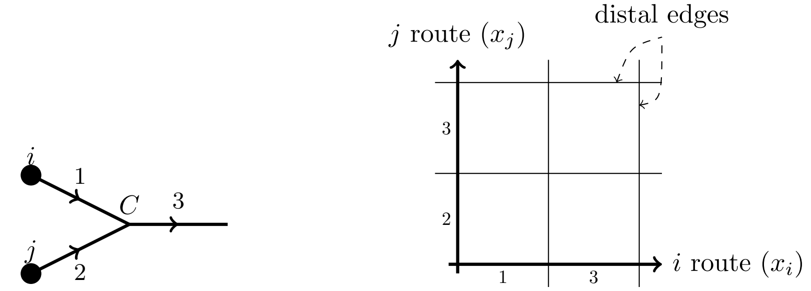
The contribution of this paper is an algorithm, defining a N-dimensional coordinate state space by using the above HCS framework (explained below in more detail) to find a speed profile which satisfies the feasible speed range for each aircraft i such that each aircraft flies on its preassigned path from the initial point to the exit point without violating the minimal required separation (Problem I.1). Using the state space (briefly introduced in section II; see Ref.[3] for a detailed exposition), problem I.1 will be formulated as one in control theory with states (1) and controls (2). The constraint of feasible speed range defines the set of admissible controls and, consequently, defines in the state space the reachable set that turns out to be a pointed polyhedral cone (called the *attainable cone).* The algorithm allows substantial parallelization, which promises to reduce the physical run time (section III).

II. HCS model in ATM problems

In ATM problems - like Problem I.1- there are N aircraft in a given airspace. In order to model this problem, the airspace can be modeled as a directed graph. A simple form of a directed graph is depicted in Figure 1a. Each aircraft has been assigned to a path in this directed graph and each path has been parameterized by an arc length coordinate, $x_i(t)$. This model captures the simultaneous positions of all N aircraft by specifying one point in the N-dimensional coordinate space, $\mathbf{X(t)}$. A 2-dimensional coordinate space is depicted in Figure 1b. The following subsections introduce some subsets of this coordinate space that will be used in the subsequence algorithm.

A. State space coordinate

Consider two aircraft, denoted i and j, each flying its own route. Their routes, in this example chosen to be polygonal paths for simpilicity, merge at point C. This situation is depicted in Figure 1a. The path assigned to aircraft i consists of segments 1 and 3; the path assigned to aircraft j, of segments 2 and 3. As indicated in the Nomenclature, the variable x_i is an arc length coordinate on the path of aircraft i; the variable x_j, an arc length coordinate on the path of aircraft j. A simultaneous position of the two aircraft will be depicted as a point (x_i, x_j) in the $x_i x_j$-coordinate space, depicted in Figure 1b. The lines in the $x_i x_j$-coordinate space corresponding to the states in which at least one aircraft has reached the end of edge 3 are called *distal edges* and are illustrated in Figure 1b. If a trajectory reaches one of the distal edges, it means the corresponding aircraft has left the airspace. The dimension of the state space will be reduced by one when one aircraft leaves the airspace.



(a) Simple merge point of aircraft i and j routes.

(b) The state space coordinate for two aircraft i and j.

Figure 1: The state space for two aircraft.

B. Attainable cone

The feasible speed range constraint stated in problem I.1 translates, for the aircraft pair i, j, into the mathematical statement that the minimal and maximal slopes of $\frac{dx_j}{dx_i} = \frac{dx_j/dt}{dx_i/dt}$, denoted $\underline{s}_{ij}$ and $\overline{s}_{ij}$, respectively, are given by

$$\underline{s}_{ij} = \frac{V_{j_{min}}}{V_{i_{max}}}, \tag{4}$$

$$\overline{s}_{ij} = \frac{V_{j_{max}}}{V_{i_{min}}}. \tag{5}$$

The intersection of the half-planes (depicted in Figure 2) constructed by

$$\mathcal{L}_{a_{ij}} \;:\; x_j - x_j^0 \geq \underline{s}_{ij}\left(x_i - x_i^0\right), \tag{6}$$

and

$$\mathcal{U}_{a_{ij}} \;:\; x_j - x_j^0 \leq \overline{s}_{ij}\left(x_i - x_i^0\right), \tag{7}$$

is called the *pairwise attainable cone for aircraft pair i, j at initial state* (x_i^0, x_j^0) and is illustrated in Figure 3. The boundary of these two half-planes are denoted as

$$\mathcal{H}_{ua_{ij}} \;:\; x_j = \overline{s}_{ij}\left(x_i - x_i^0\right) + x_j^0,$$
$$\mathcal{H}_{la_{ij}} \;:\; x_j = \underline{s}_{ij}\left(x_i - x_i^0\right) + x_j^0,$$

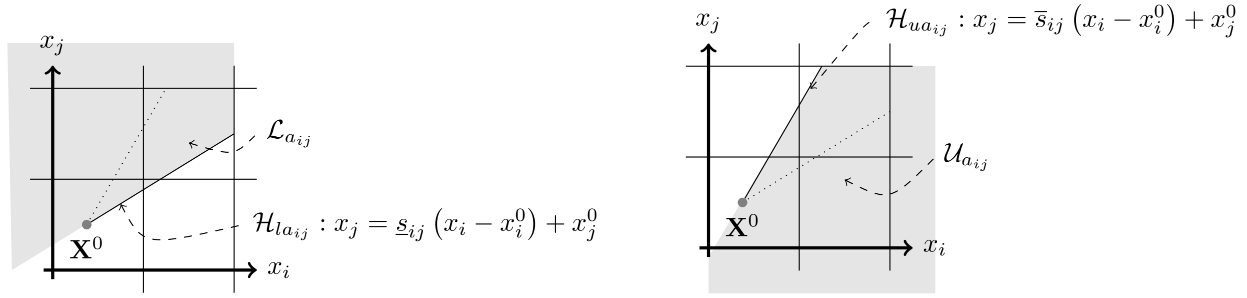

x_j
$\mathcal{L}_{a_{ij}}$
$\mathcal{H}_{la_{ij}} : x_j = \underline{s}_{ij}\left(x_i - x_i^0\right) + x_j^0$
$\mathbf{X}^0$
x_i
x_j
$\mathcal{H}_{ua_{ij}} : x_j = \overline{s}_{ij}\left(x_i - x_i^0\right) + x_j^0$
$\mathcal{U}_{a_{ij}}$
$\mathbf{X}^0$
x_i

Figure 2: The half-planes associated with the minimal and maximal slopes.

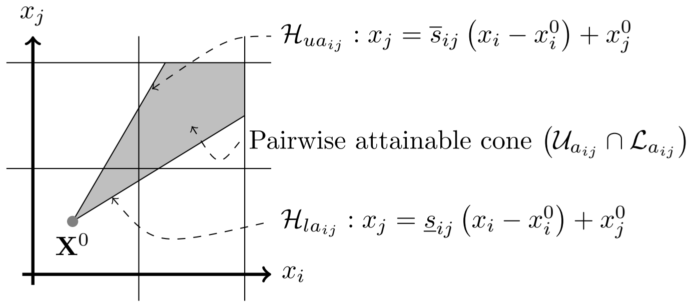

x_j
$\mathcal{H}_{ua_{ij}} : x_j = \overline{s}_{ij}\left(x_i - x_i^0\right) + x_j^0$
Pairwise attainable cone $\left(\mathcal{U}_{a_{ij}} \cap \mathcal{L}_{a_{ij}}\right)$
$\mathcal{H}_{la_{ij}} : x_j = \underline{s}_{ij}\left(x_i - x_i^0\right) + x_j^0$
$\mathbf{X}^0$
x_i

Figure 3: The pairwise attainable cone set of two aircraft i and j.

Using analogous reasoning (i.e., imposing the suitable set of inequality constraints), the *attainable cone set for N aircraft at initial state* $\mathbf{X}^0$ (N arbitrary) is defined as the set of intersection of all pairwise attainable cones

$$\mathcal{A}(\mathbf{X}^0) = \bigcap_{\substack{i,j=1 \\ i<j}}^{N} \left(\mathcal{U}_{a_{ij}} \cap \mathcal{L}_{a_{ij}}\right). \tag{8}$$

This set, being an intersection of closed half-planes, is a closed polyhedron, and can be shown to be a pointed cone with vertex $\mathbf{X}^0$. This cone can also be written as

$$\mathcal{A}(\mathbf{X}^0) = \{\mathbf{X}\,|A\mathbf{X} \leqslant B\}\,. \tag{9}$$

There are $\frac{N(N-1)}{2}$ pairwise frameworks and in each one there are two N-dimensional half-planes to define the pairwise attainable cone set. Therefore, the matrix A is a $N(N-1)$ by N and the vector B has dimension $N(N-1)$.

C. Separation-loss and Separation-compliant Set

The set of all states $\mathbf{X}$ that violate the separation requirement is called the *separation-loss* set. The separation-loss set for situation shown in Figure 1a–i.e., for an HCS model consisting only of the aircraft i and j–is approximated by a open polyhedral region illustrated in Figure 4. The hyperplane $\mathcal{H}_{uc_{ij}}$ is the set of all states where the distance between aircraft j and i is equal to the minimum required separation and aircraft j is closer to it's distal edge. Similarly, the hyperplane $\mathcal{H}_{lc_{ij}}$ is the set of all states where the distance between two aircraft is equal to the minimum required separation but aircraft i is closer to it's distal edge. In both cases, in order to keep the minimum required separation distance both aircraft must fly with the same speed. Therefore, the slope of the projection of these two hyperplanes in ij-coordinate is equal to 1,

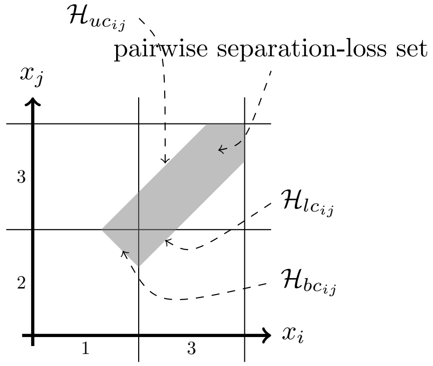


Figure 4: The separation-loss set for the aircraft pair i, j arising from a merge point.

$$s = \frac{V_j}{V_i} = 1.$$

Hence the slope of the projection of hyperplane $\mathcal{H}_{bc_{ij}}$ which is perpendicular to projections of $\mathcal{H}_{uc_{ij}}$ and $\mathcal{H}_{lc_{ij}}$ is equal to -1. This region will be called the *pairwise separation-loss set for aircraft pair* i, j, where $\mathcal{H}_{bc_{ij}}$, $\mathcal{H}_{uc_{ij}}$ and $\mathcal{H}_{lc_{ij}}$ are hyperplanes defined as follow:

$$\begin{aligned}
\mathcal{H}_{bc_{ij}} &= \left\{\mathbf{X}\,\middle|x_j = -x_i + d_{bc_{ij}}\right\},\\
\mathcal{H}_{uc_{ij}} &= \left\{\mathbf{X}\,\middle|x_j = x_i + d_{uc_{ij}}\right\},\\
\mathcal{H}_{lc_{ij}} &= \left\{\mathbf{X}\,\middle|x_j = x_i + d_{lc_{ij}}\right\},
\end{aligned}$$

where $d_{bc_{ij}}$, $d_{uc_{ij}}$ and $d_{lc_{ij}}$ are appropriate constants in line equations which are boundaries of polyhedral region, the calculation of these constants is out of scope of this paper, for more detail see ref.[3]

The *feasible trajectory* is a trajectory that does not have intersection with the interior of the separation-loss set and the slope of the trajectory at all points must be in the range of $[\underline{s}_{ij}, \overline{s}_{ij}]$ which is called *slope condition* hereafter. However, there is a region that cannot satisfy these two conditions simultaneously. This region is called the *roof set*. The roof set for 2-dimensional subspace is illustrated in Figure 5. Any line from any point in the roof set needs to have slope smaller than $\underline{s}_{ij}$ or greater than $\overline{s}_{ij}$ to avoid intersection with the separation-loss set. Where hyperplanes $\mathcal{H}_{ur_{ij}}$ and $\mathcal{H}_{lr_{ij}}$ are defined as:

$$\begin{aligned}
\mathcal{H}_{ur_{ij}} &= \left\{\mathbf{X}\,\middle|x_j = \overline{s}_{ij}x_i + d_{ur_{ij}}\right\},\\
\mathcal{H}_{lr_{ij}} &= \left\{\mathbf{X}\,\middle|x_j = \underline{s}_{ij}x_i + d_{lr_{ij}}\right\},
\end{aligned}$$

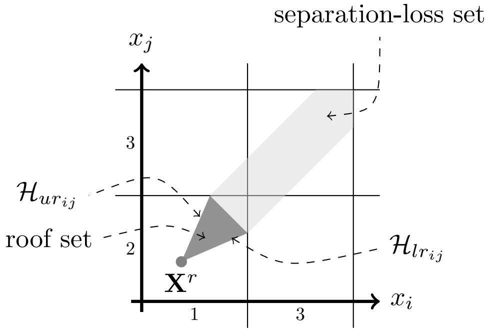


Figure 5: The separation-loss set and the roof set between aircraft i and j in a simple merge point

where $d_{ur_{ij}}$ and $d_{lr_{ij}}$ are appropriate constants of line equations which are boundaries of the pairwise roof set. The intersection of hyperplanes $\mathcal{H}_{ur_{ij}}$ and $\mathcal{H}_{lr_{ij}}$ will be called *roof apex* and denoted $\mathbf{X}^r$. Therefore, the union of the pairwise separation-loss set and the roof set is exactly the set of states through which no feasible trajectory passes. This set will be called the *pairwise infeasible state set*. The pairwise infeasible state set is defined as the intersection (shown shaded gray in Figure 5) of the four open half-planes (individually depicted in Figure 6).

$$\begin{aligned}
\mathcal{U}_{c_{ij}} &= \left\{ X \left| x_j < x_i + d_{uc_{ij}} \right. \right\}, \\
\mathcal{L}_{c_{ij}} &= \left\{ X \left| x_j > x_i + d_{lc_{ij}} \right. \right\}, \\
\mathcal{U}_{r_{ij}} &= \left\{ X \left| x_j < \overline{s}_{ij} x_i + d_{ur_{ij}} \right. \right\}, \\
\mathcal{L}_{r_{ij}} &= \left\{ X \left| x_j > \underline{s}_{ij} x_i + d_{lr_{ij}} \right. \right\}.
\end{aligned}$$

To determine the infeasible state set for an N-dimensional state space, first the pairwise infeasible state set needs to be computed, denoted $\mathcal{C}_{ij}$, as

$$\mathcal{C}_{ij} = \mathcal{U}_{c_{ij}} \cap \mathcal{L}_{c_{ij}} \cap \mathcal{U}_{r_{ij}} \cap \mathcal{L}_{r_{ij}} = \left\{ \mathbf{X} \left| C_{ij} \mathbf{X} > D_{ij} \right. \right\},$$

where C_{ij} is 4 by N matrix and D_{ij} is N-dimensional vector. The pairwise infeasible state set $\mathcal{C}_{ij}$ is an intersection of four open half-planes, hence is an open set. The infeasible state set for N aircraft will now be defined as the union of all pairwise infeasible state sets:

$$\mathcal{C} = \bigcup_{\substack{i,j=1 \\ i<j}}^{N} \mathcal{C}_{ij}.$$

Similarly, the infeasible state set for N aircraft is an open set. The *separation-compliant state set* is defined as the complement of $\mathcal{C}$ and, in what follows, is denoted $\mathcal{C}'$. Note that the separation-compliant set is a closed set. In set-theoretic notation, we have

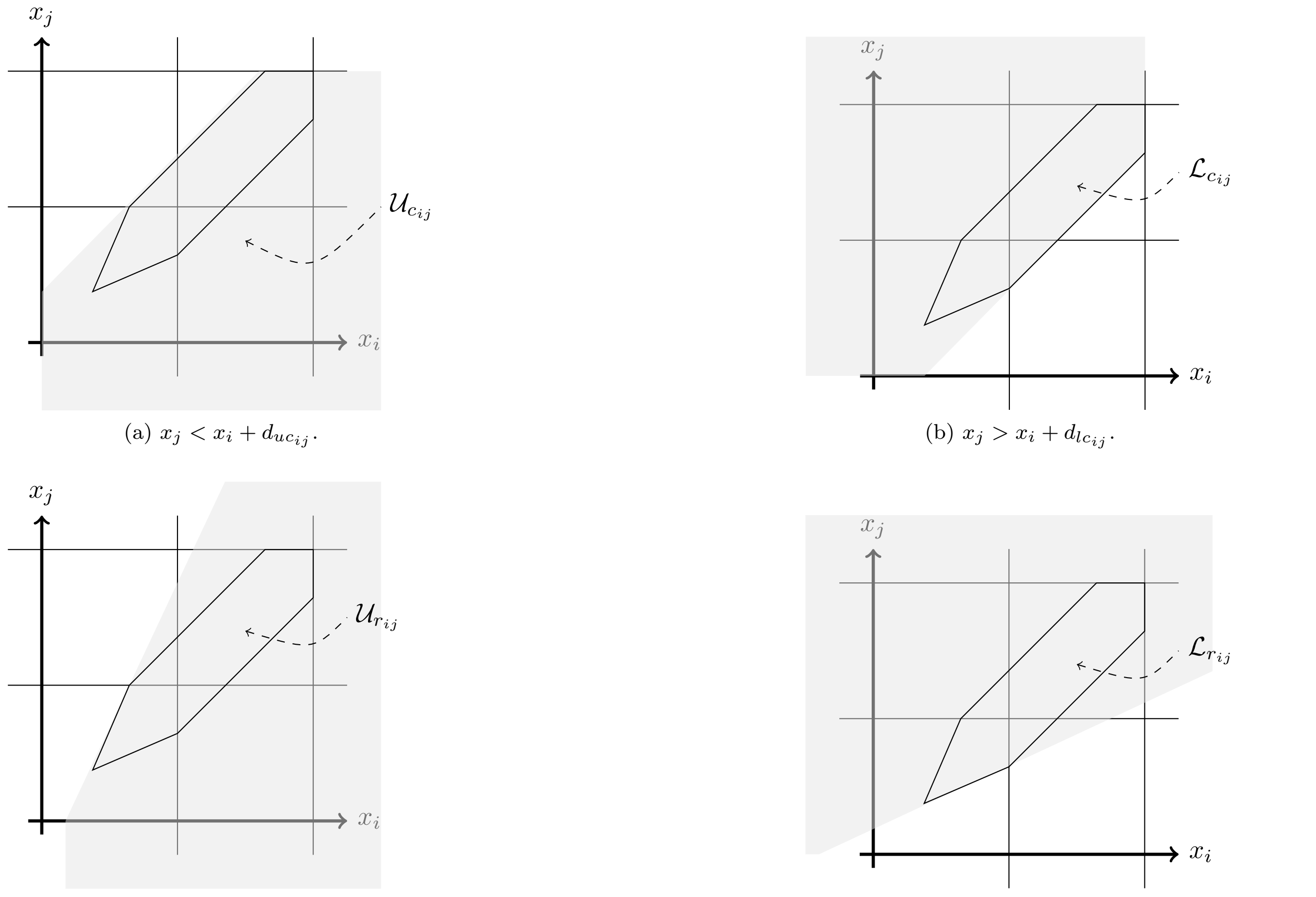
$$\mathcal{C}' = \bigcap_{\substack{i,j=1 \\ i<j}}^{N} \mathcal{C}'_{ij}, \tag{10}$$

where,

$$\mathcal{C}'_{ij} = \left\{ X \left| C_{ij} X \ngtr D_{ij} \right. \right\}.$$

D. Trajectory

A trajectory is a subset of the intersection between the attainable cone set and the separation-compliant state set. In general, the trajectory is a curve. For simplicity in this paper, the algorithm generates a polygonal



(a) $x_j < x_i + d_{uc_{ij}}$.

(b) $x_j > x_i + d_{lc_{ij}}$.

(c) $x_j < \bar{s}_{ij} x_i + d_{ur_{ij}}$.

(d) $x_j > \underline{s}_{ij} x_i + d_{lr_{ij}}$.

Figure 6: Three half-planes of the pairwise separation-loss set.

trajectory [b]. Each point on the trajectory is representing the arc length coordinate of all aircraft at time t. As noted before, if the trajectory reaches one of distal edges then it means the corresponding aircraft has left the airspace and the dimension of state space will reduce by one. Therefore, the intersection of all distal edges is the point where all aircraft have left the airspace which is called *distal point* and denoted by $\mathbf{X}^f$.

$$\mathbf{X}(t_f) = \mathbf{X}^f = \begin{bmatrix} x_1^f \\ \cdot \\ \cdot \\ \cdot \\ x_N^f \end{bmatrix} \tag{11}$$

In Figure 7 a trajectory for a simple 2-dimensional state space is depicted. When the trajectory reaches point a, it means the aircraft j has left the airspace. After this point aircraft i is the only aircraft in the airspace and the dimension of airspace reduces to one and the trajectory moves along the distal edge corresponding to aircraft j until reaches to distal point $\mathbf{X}^f$.

Hence, a trajectory must start from initial point (3) and ends at distal point (11). A trajectory that satisfies the initial condition (3) and terminal condition (11) is said to be *feasible* if lies in the separation-compliant state set and the tangent $\mathbf{S}$ of the trajectory at each point satisfies for all pairs i, j the slope condition. A feasible trajectory for a simple 2-dimensional state space is depicted in Figure 7. If such a trajectory exists, then there is a speed control solution. The algorithm introduced in Section III looks for such a solution and generates a polygonal trajectory in the suitable N-dimensional state space coordinate.

[b] Piecewise linear trajectory

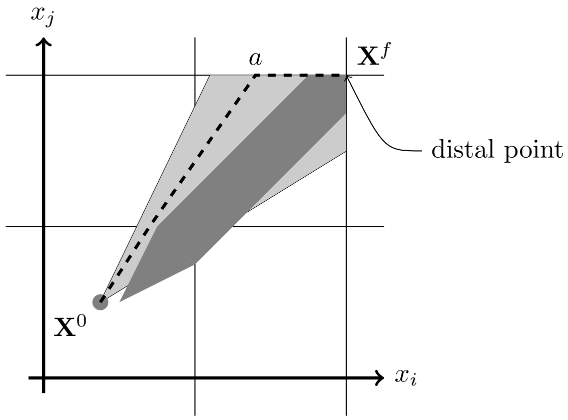


Figure 7: Feasible Trajectory

E. Constraints

The polygonal trajectory is a 1-dimensional subspace of the N-dimensional state space. If one can define $N-1$ linearly independent equality constraints for each segment, then there is just one solution to satisfy all constraints, otherwise there is more than one solution for that segment of the trajectory. The equality constraints are the hyperplanes defined in preceding subsections. The constraints in each segment specify a 1-dimensional intersection of hyperplanes which will define the current segment of the trajectory. Six hyperplanes have been defined as boundaries of the attainable cone set ($\mathcal{H}_{ua_{ij}}$ and $\mathcal{H}_{la_{ij}}$), the infeasible state set (($\mathcal{H}_{uc_{ij}}$, $\mathcal{H}_{lc_{ij}}$, $\mathcal{H}_{ur_{ij}}$ and $\mathcal{H}_{lr_{ij}}$). The effect of each hyperplane will be discussed here. The algorithm will construct the polygonal trajectory "backward", i.e. from the terminal point to the initial point. For this reason, the algorithm will be hereafter called *the Backward Sweep Method*. The slope of each segment of the trajectory must satisfy the slope condition.

When the trajectory intersects with one of the hyperplanes $\mathcal{H}_{ua_{ij}}$ or $\mathcal{H}_{la_{ij}}$ the next segment must stay on the intersected hyperplane in order to satisfy the slope condition. This situation is depicted for hyperplane $\mathcal{H}_{ua_{ij}}$ in Figure 8. In this figure the trajectory intersects the hyperplane $\mathcal{H}_{ua_{ij}}$ at point a. The slope of the next segment must satisfy the slope condition, $\underline{s}_{ij} \leq s_{ij} \leq \overline{s}_{ij}$, but the only segment that satisfies the slope condition and lies inside the attainable cone is the one with the slope of $\overline{s}_{ij}$ which is on the hyperplane $\mathcal{H}_{ua_{ij}}$. After point a the trajectory must stay on this hyperplane and cannot leave this hyperplane. Therefore, hyperplanes $\mathcal{H}_{ua_{ij}}$ and $\mathcal{H}_{la_{ij}}$ will be called *permanent constraints.*

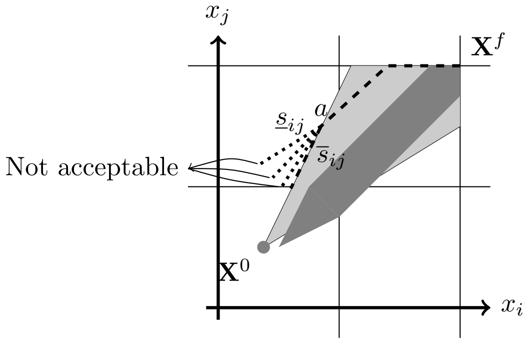


Figure 8: The trajectory must stay on the hyperplane of the attainable cone after the intersection point a.

When the trajectory intersects one of hyperplanes $\mathcal{H}_{uc_{ij}}$ or $\mathcal{H}_{lc_{ij}}$, the slope of the next segment can be in the range of $s_{ij} \in [\underline{s}_{ij}, 1]$ or $s_{ij} \in [1, \overline{s}_{ij}]$, respectively. Therefore, the next segment of the trajectory can either stay on the intersected hyperplane or leave it. The intersection with hyperplane $\mathcal{H}_{uc_{ij}}$ is depicted in Figure 9. The hyperplanes $\mathcal{H}_{uc_{ij}}$ and $\mathcal{H}_{lc_{ij}}$ will be called *temporary constraints.*

The remaining two hyperplanes, $\mathcal{H}_{ur_{ij}}$ and $\mathcal{H}_{lr_{ij}}$, do not constraint the trajectory in the Backward Sweep Method. An example of this is depicted in Figure 10. Therefore, the hyperplanes $\mathcal{H}_{ur_{ij}}$ and $\mathcal{H}_{lr_{ij}}$ are not

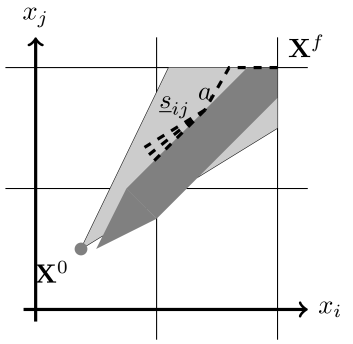


Figure 9: After the intersection point a, the trajectory can stay on the hyperplane $\mathcal{H}_{uc_{ij}}$ or leave it with the slope $\underline{s}_{ij} \leq s_{ij} \leq 1$.

considered as constraints.

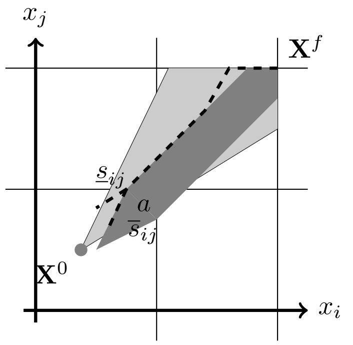


Figure 10: The hyperplane $\mathcal{H}_{ur_{ij}}$ does not impose any constraint on the trajectory.

In summary, among the six hyperplanes defined, four of them $\mathcal{H}_{ua_{ij}}$ and $\mathcal{H}_{la_{ij}}$ (boundaries of the attainable cone set) and $\mathcal{H}_{uc_{ij}}$ and $\mathcal{H}_{lc_{ij}}$ (boundaries of the separation-loss set) are used as constraints to generate the polygonal trajectory.

III. The Algorithm

Lemma III.1. *In 2-aircraft case, if the feasible speed range of the aircraft i has overlap with the feasible speed range of the aircraft j,*

$$\begin{cases} V_{i_{min}} < V_{j_{max}} \\ V_{j_{min}} < V_{i_{max}} \end{cases}, \tag{12}$$

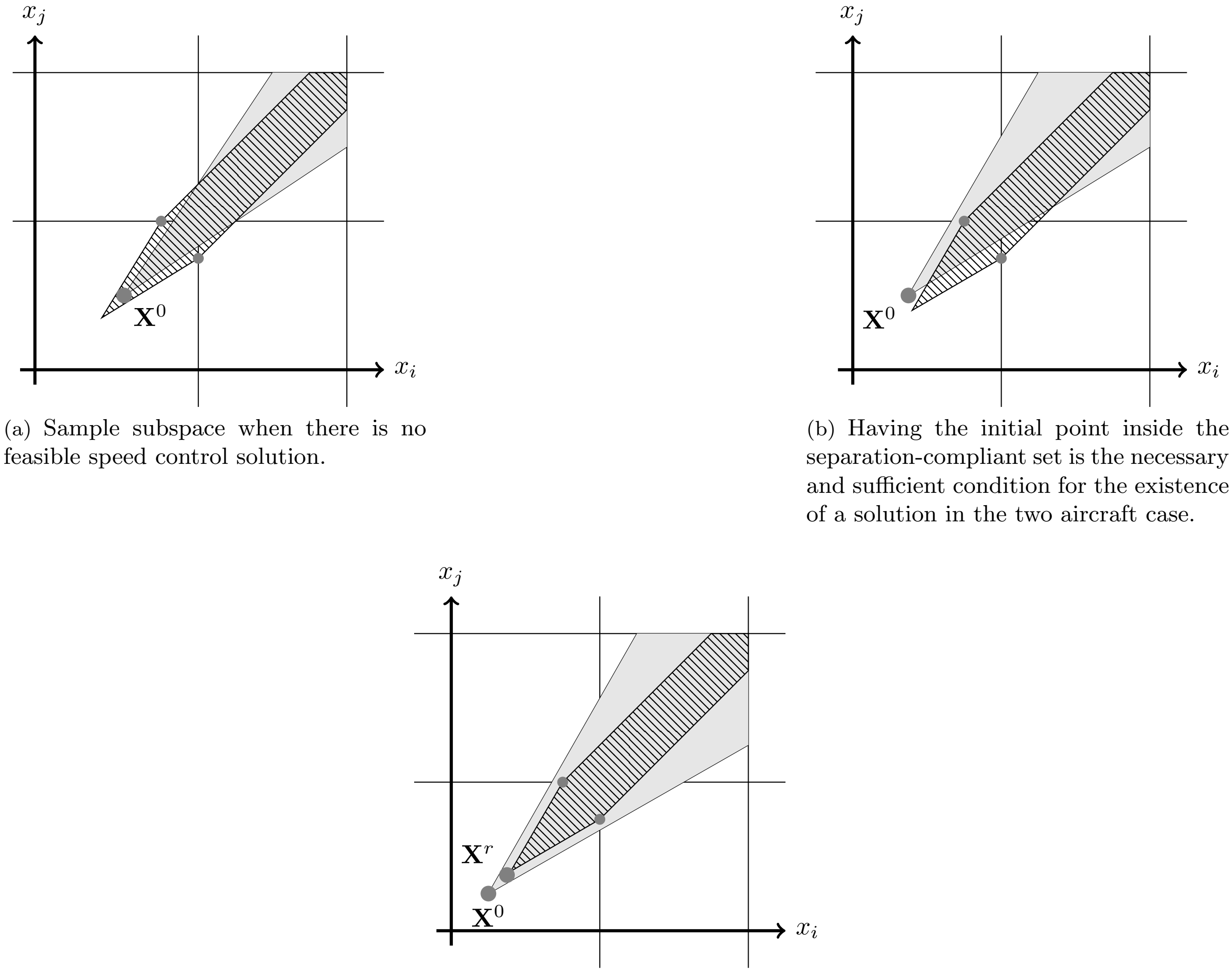
then, the necessary and sufficient condition for the existence of a speed control solution is that the initial point $\mathbf{X}^0$ lies inside the separation-compliant state set, i.e.

$$\mathbf{X}^0 \in \mathcal{C}'. \tag{13}$$

Proof. If the initial point $\mathbf{X}^0$ does not lie inside the separation-compliant state set (lie inside the pairwise infeasible state set), then no feasible trajectory can be initiated from the initial point. This situation is depicted in Figure 11a. As mentioned in Section II.C, for the 2-aircraft case, the hyperplanes $\mathcal{H}_{ur_{ij}}$ and $\mathcal{H}_{lr_{ij}}$ are parallel to $\mathcal{H}_{ua_{ij}}$ and $\mathcal{H}_{la_{ij}}$, respectively. The slopes of the hyperplanes $\mathcal{H}_{uc_{ij}}$ and $\mathcal{H}_{lc_{ij}}$ are equal to one. Therefore, if the initial point $\mathbf{X}^0$ lies inside the separation-compliant state set from equation (12) one can conclude that at least one of the hyperplanes $\mathcal{H}_{uc_{ij}}$ or $\mathcal{H}_{lc_{ij}}$ is inside the attainable cone. Therefore, there exists at least one ray from the initial point $\mathbf{X}^0$ to the distal edge with no intersection with the pairwise infeasible state set. This situation is depicted in Figures 11b and 11c. □ □

Note that in the N-dimensional case, the condition (13) is just a necessary condition for the existence of a speed control solution.

In the 2-aircraft case of the state space defined in Section II, there are two possible sequences for aircraft to exit the airspace: *I)* first aircraft i exit then j, *II)* first aircraft j exit then i.



(a) Sample subspace when there is no feasible speed control solution.

(b) Having the initial point inside the separation-compliant set is the necessary and sufficient condition for the existence of a solution in the two aircraft case.

(c) If the roof apex lies inside the attainable cone both sequences are possible.

Figure 11: Pairwise subspace

If the roof apex lies in the attainable cone, $\mathbf{X}^r \in \mathcal{A}$ then both sequences are feasible (see Figure 11c), otherwise, depending on the position of the initial point with respect to the separation-loss set, one of the two possible sequences is feasible. For instance, in Figure 11b there is one feasible sequence in which aircraft j must exit the airspace before aircraft i. In general, for the N-dimensional state space ($N \geqslant 2$), there are $N!$ possible sequences. Therefore, by looking at all pairwise subspaces one can find all feasible sequences. For instance, if just one of pairwise subspaces allows one feasible sequence, i.e. in subspace (x_i, x_j) aircraft j must exit the airspace before aircraft i, then all sequences that let aircraft i exit before j are infeasible and the number of feasible sequences among all N aircraft will reduce to $\frac{(N+1)!}{2(N-1)}$.

The algorithm first will generate all the feasible sequences, then it will look for existence of a polygonal trajectory in each feasible sequence. Therefore, after finding all the feasible sequences, searching for a feasible trajectory in each feasible sequence can be done in parallel. This parallelism will reduce the computation time. The polygonal trajectory is a 1-dimensional subspace of the N-dimensional feasible set. First consider that there are $N-1$ linearly independent equality constraints for each segment, then there is just one solution to satisfy all constraints. The case with more than one solution will be discussed later.

If there are ℓ aircraft in the airspace, then the dimension of state space is ℓ. Therefore, to generate the trajectory, $\ell - 1$ constaints are needed. Each segment of the trajectory can be defined as,

$$\mathbf{X} = \mathbf{X^p} - \mu_{p+1}\mathbf{d}_{p+1}, \tag{14}$$

where $\mathbf{X^p} \in \mathbb{R}^N$ is the last point of the previous segment, $\mathbf{d}_{p+1} \in \mathbb{R}^N$ is the director of the segment $p+1$ and $\mu_{p+1} \in [0, \lambda_{p+1}]$ is a scalar variable in which $\mu_{p+1} = 0$ is associated with the starting point of the current segment and $\mu_{p+1} = \lambda_{p+1}$ is associated with the end point. In order to have the trajectory on the $\ell - 1$ constraints, the director of the line segment must be perpendicular to all normal vectors of $\ell - 1$ constraints.

$$\mathbf{Q}_{\ell-1}\mathbf{U}_{p+1} = 0, \tag{15}$$

where $\mathbf{Q}_{\ell-1}$ is a $\ell - 1$ by ℓ matrix which rows of $\mathbf{Q}_{\ell-1}$ are the normal vectors and $\mathbf{U}_{p+1} \in \mathbb{R}^\ell$ is the projection of $\mathbf{d}_{p+1}$ in the ℓ-dimensional subspace. Equation (15) shows that the vector $\mathbf{U}_{p+1}$ must be in the null space[19] of $\mathbf{Q}_{\ell-1}$. $\mathbf{Q}_{\ell-1}$ contains $\ell - 1$ independent normal vectors so the dimension of its null space is one. Hence, the vector $\mathbf{U}_{p+1}$ spans the null space of $\mathbf{Q}_{\ell-1}$. Equation (14) is a line equation. This line intersects all hyperplanes $\mathcal{H}_{ua_{ij}}$, $\mathcal{H}_{la_{ij}}$, $\mathcal{H}_{uc_{ij}}$ and $\mathcal{H}_{lc_{ij}}$ in ℓ-dimensional subspace, where $i, j = 1, \cdots, \ell$ and $i < j$. These intersections will be called *feasible intersections*: if i) they lie inside the intersection of the attainable cone set and the separation-compliance state set ($\mathcal{A} \cap \mathcal{C}'$), ii) there exists a line segment that starts from the intersection point, satisfies the slope condition and lies inside the $\mathcal{A} \cap \mathcal{C}'$. The current segment ends at $\mu_{p+1} = \lambda_{p+1}$ which is associated with the closest feasible intersection. The end point is denoted by $\mathbf{X}^{p+1}$.

$$\mathbf{X}^{p+1} \in \mathcal{A} \cap \mathcal{C}'.$$

Assume all N aircraft in the Problem I.1 have left the airspace. Then, the state vector is represented by equation (11). Also assume one of the feasible sequences is:

$$\{i, j, \ldots, k, m\} \tag{16}$$

then aircraft m is the first one to leave the airspace and aircraft i is the last one. Note that, the likelihood of the existence of a speed control solution in a feasible sequence (16) is higher than the other feasible sequences if

$$L_i > L_j > \ldots > L_k > L_m,$$

where $L_i = x_i^f - x_i^0$ is the total arc length of the path assigned to the aircraft i. Similarly, the likelihood of the existence of a speed control solution in feasible sequence (16) is lower than the other feasible sequences if

$$L_i < L_j < \ldots < L_k < L_m.$$

Therefore, the algorithm starts from the feasible sequence with higher likelihood of the existence of a speed control solution. The algorithm finds the trajectory by using the Backward Sweep Method. Hence, in this method the distal point (equation (11)) is the starting point. Therefore, the first aircraft to consider is the aircraft i. Until considering the next aircraft (aircraft j), there is one aircraft in the airspace, then the state space is spanned by the unit vector:

$$\mathbf{d}_1 = \mathbf{e}_i = (0, 0, \ldots, 1, \ldots, 0, 0)^T.$$

Therefore, by using the equation (14) one can define the first segment of the trajectory in the following form,

$$\mathbf{X} = \mathbf{X^f} - \mu_1\mathbf{d}_1, \tag{17}$$

The dimension of state space will increase to 2, when aircraft j enters the airspace. In Section II, 6 lines have been defined in order to determine the pairwise attainable cone and the pairwise infeasible state set. The intersection of the line defined by equation (17) and 4 of these lines can be considered as constraints. An example of 2-dimensional space is depicted in Figure 12. The intersection of the trajectory with the 4 lines are denoted by a through d in Figure 12. The intersections a and b are not acceptable since μ_1 is negative in both points. The intersections c and d are acceptable since they are inside the intersection of the attainable cone and the separation-compliant state set ($\mathcal{A} \cap \mathcal{C}'$) and the correspoinding μ_1's are positive. Therefore,

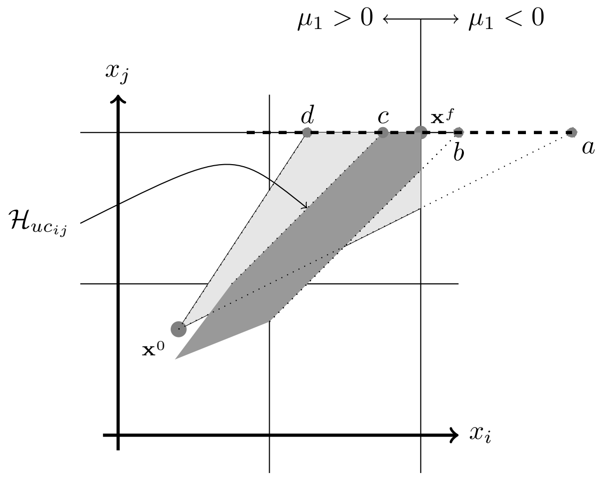


Figure 12: State space when just aircraft i and j are in the airspace.

both points c and d are feasible but the closest one, which is point c, is taken as the end point of the first segment of the trajectory and denoted by

$$\mathbf{X}^1 \in \mathcal{A} \cap \mathcal{C}'.$$

At this point, the equation of the trajectory changes to

$$\mathbf{X} = \mathbf{X^1} - \mu_2 \mathbf{d}_2, \tag{18}$$

where

$$\mathbf{d}_2 = \begin{bmatrix} \mathbf{0} & u_i & \mathbf{0} & u_j & \mathbf{0} \end{bmatrix}^T,$$

and the projection of $\mathbf{d}_2$ in 2-dimensional space is $\mathbf{U}_2 = \begin{bmatrix} u_i & u_j \end{bmatrix}^T$, which is the vector spans the (1-dimensional) null space of $\mathbf{Q}_1$:

$$\begin{aligned} \mathbf{Q}_1 \mathbf{U}_2 &= 0 \\ \mathbf{Q}_1 &= \begin{bmatrix} n_i & n_j \end{bmatrix} \end{aligned}$$

where $\begin{bmatrix} n_i & n_j \end{bmatrix}^T$ is the normal vector to the intersected line in $\mathbb{R}^2$ (i.e. line $\mathcal{H}_{lc_{ij}}$ in Figure 12). The equation (18) defines the second segment of the trajectory. From this point the process of generating the trajectory can be divided into two cases.

case 1) Suppose a new aircraft is adding to the airspace.
Assume the dimension of space is $\ell < N$. In this case the dimension of space will increase by one to $\ell{+}1$. Therefore, one extra constraint must be added to the previous constraints set. The previous segment of the trajectory intersects all hyperplanes in $(\ell+1)$-dimensional space and ends at the closest feasible intersection. The intersection point is the starting point for the next segment and the intersected hyperplane is the new constraint.

case 2) Suppose no aircraft is adding to the airspace.
Assume the dimension of space is $\ell \leqslant N$. Since the number of aircraft inside the airspace does not change then the dimension of space does not change. Hence, the number of constraints is $\ell - 1$, same as previous segment. Then one of the existing constraints needs to be replaced by the new constraint. The previous segment of the trajectory intersects all hyperplanes in ℓ-dimensional state space and ends at the closest feasible intersection. The intersection point is the starting point of the next segment and the intersected hyperplane is the new constraint.

The process will continue until the trajectory reaches the initial point $\mathbf{X}^0$ or no feasible intersection can be found.

During the process of finding the feasible trajectory some of temporary constraints become inactive, a sample configuration is shown in Figure 13. If one removes the inactive constraints from the set of constraints, then the dimension of null space of $Q_{\ell-1}$ will be more than one. Therefore, there are more than one solution for the feasible trajectory. Multiple solutions can also occur when the constraints are not independent. In case of multiple solutions, an optimization problem needs to look for a specific solution. The multiple solution case will be discussed later. In this algorithm, in order to avoid the optimization problem, the inactive constraints are kept so as to have a 1-dimensional null space.

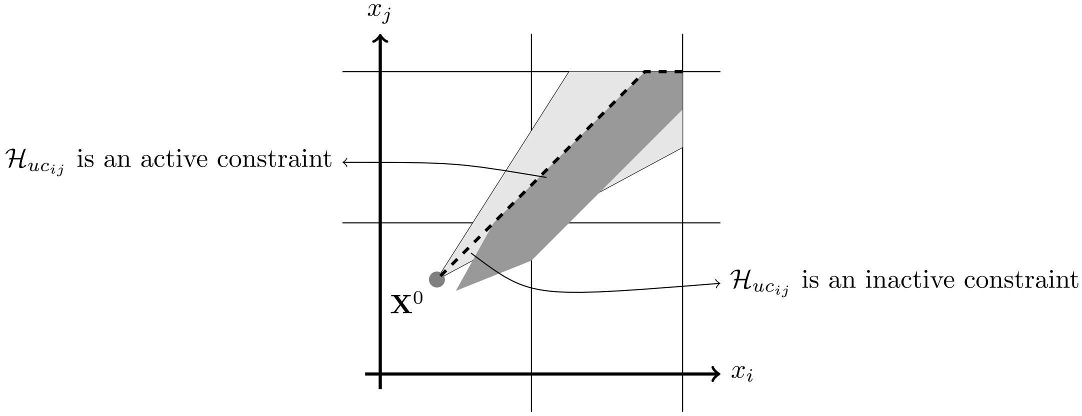

x_j
$\mathcal{H}_{uc_{ij}}$ is an active constraint
$\mathcal{H}_{uc_{ij}}$ is an inactive constraint
$\mathbf{X}^0$
x_i

Figure 13: Active and inactive constraint

When no new aircraft is added to the airspace -Case 2- one of the existing constraint will be replaced by the new constraint. The best constraints to be replaced are inactive constraints. If there is no inactive constraint then the new constraint must be replaced with an active temporary constraint. Therefore, The existing constraints are replaced in the following priority:

a) inactive constraint,

b) active temporary constraint.

In general, the algorithm has two parts; in the first part it finds all the feasible sequences, and in the second part it looks for the speed control solution in each feasible sequence. Finding the trajectory in each sequence is an independent process and can be done in parallel. Therefore, the algorithm has the following steps:

1) Find all feasible sequences

    1a) Generate all pairwise subspaces and determine pairwise feasible sequences.

    1b) Find all feasible sequences among N aircraft, sorted in descending order of aircraft path length. If no feasible sequence exist then there is no speed control solution.

2) Look for a speed control solution in a feasible sequence by using the Backward Sweep Method

    2a) Add one aircraft to the airspace and find the new segment of the trajectory. This segment can be found with the method described in case 1.

        2a-I) If no feasible intersection was found for the new segment, remove the added aircraft and go to step (2b),

        2a-II) If the feasible intersection exists but there is no aircraft left to add, go to step (2b),

        2a-III) If the feasible intersection exists and there exist more aircraft to add, go step (2a).

    2b) Find the new segment of the trajectory by using the method described in case 2. If the feasible intersection for the new segment,

        2b-I) Does not exist, then there is no speed control solution.

2b-II) Is equal to the initial point (X^0), then stop.

2b-III) Exists but if there is an airplane to add, go to step (2a), if not repeat step (2b).

In summery, the following lemma can be defined for the existence of a speed control solution for N aircraft,

Lemma III.2. *In N aircraft case, the speed control solution exists if and only if there exists a feasible sequence for all N-aircraft and a feasible trajectory for the given feasible sequence.*

IV. Numerical Example

As an example a portion of LAX routes has been used as a sample airspace. In this example 28 aircraft are flying in the airspace. The initial positions of those aircraft are shown in Figure 14. The radii of the circles are equal to half of the minimum required separation, and aircraft are at the center of circles. In this simulation, the minimum required separation is 5 nmi and the speed range is $[110, 250]$ nmi/hr for all aircraft. Figures 15-19 are depicting the position of aircraft after every 30.5 sec. The computation time of the algorithm for this example in a computer with 2.2 GHz, Xeon(R) processor is about 17 seconds in MATLAB. This computation time can be reduced by using lower level languages, such as C^{++}.

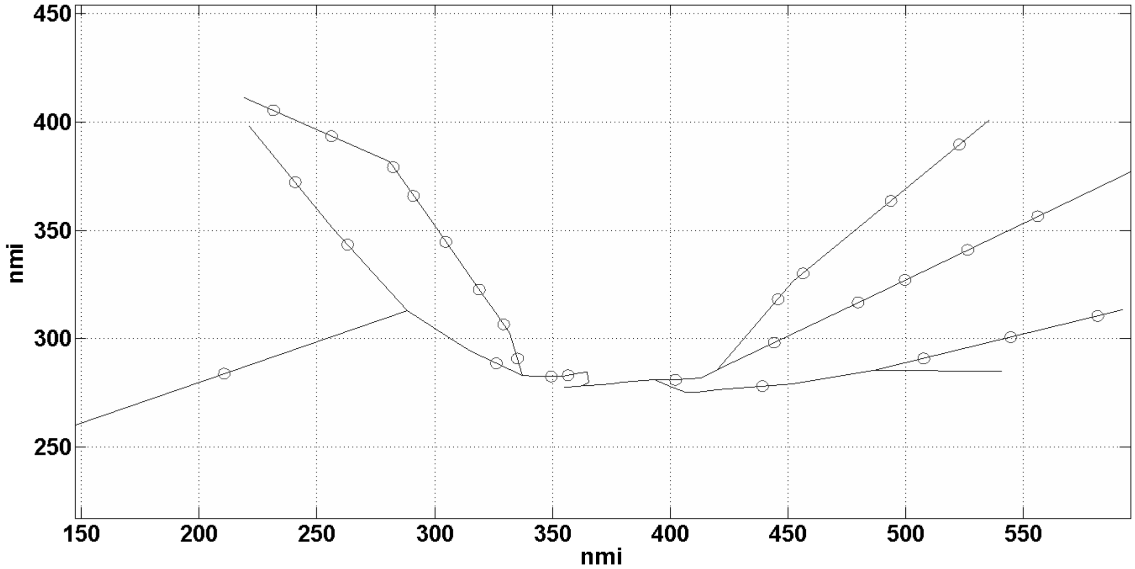


Figure 14: Initial positions of aircraft in LAX airspace.

V. Multiple solution case

If one removes the inactive constraints from the set of constraints, or the $\ell - 1$ constraints are not linearly independent, then the dimension of null space of $Q_{\ell-1}$ will be more than one. Therefore, there is more than one solution for the feasible trajectory. Assume the dimension of null space of $Q_{\ell-1}$ is $M > 1$. Then, the null space of $Q_{\ell-1}$ is spanned by M unit vectors,

$$\mathbf{E} = \begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_2 & \cdots & \mathbf{e}_M \end{bmatrix},$$

where

$$\mathbf{e}_i = \begin{bmatrix} e_{1i} & e_{2i} & \cdots & e_{\ell i} \end{bmatrix}^T,$$

where ℓ is the number of aircraft inside the airspace. The vector $\mathbf{U}$ in equation (14) can be expressed as,

$$\mathbf{U} = \begin{bmatrix} u_1 & u_2 & \cdots & u_M \end{bmatrix}^T = \alpha_1 \mathbf{e}_1 + \alpha_2 \mathbf{e}_2 + \cdots + \alpha_M \mathbf{e}_M,$$
$$u_i = \alpha_1 e_{i1} + \alpha_2 e_{i2} + \cdots + \alpha_M e_{iM}, \tag{19}$$

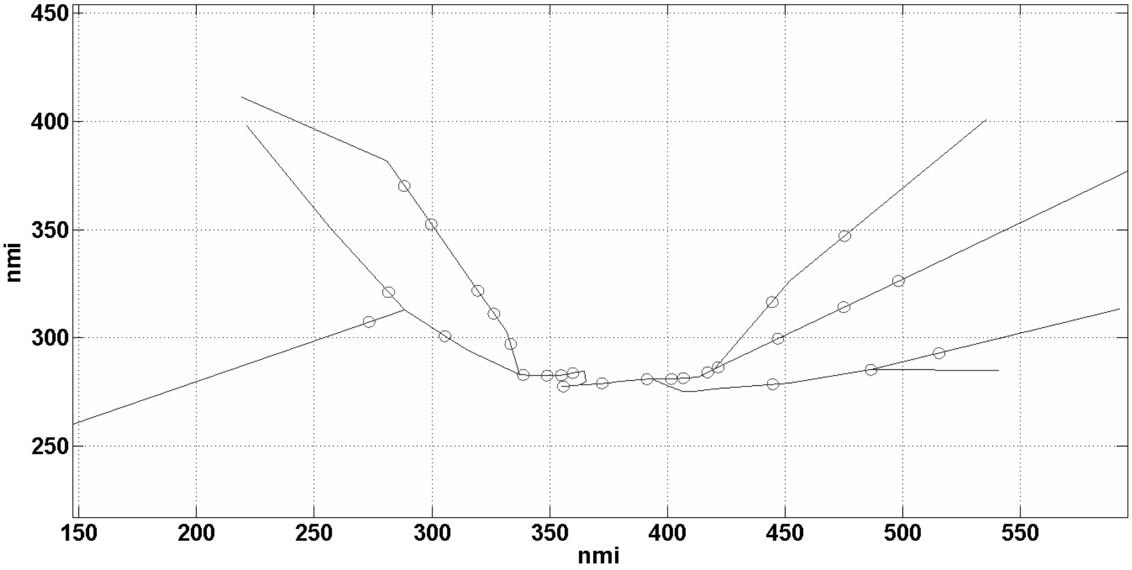


Figure 15: Positions of aircraft in LAX airspace at t=30.5 s.

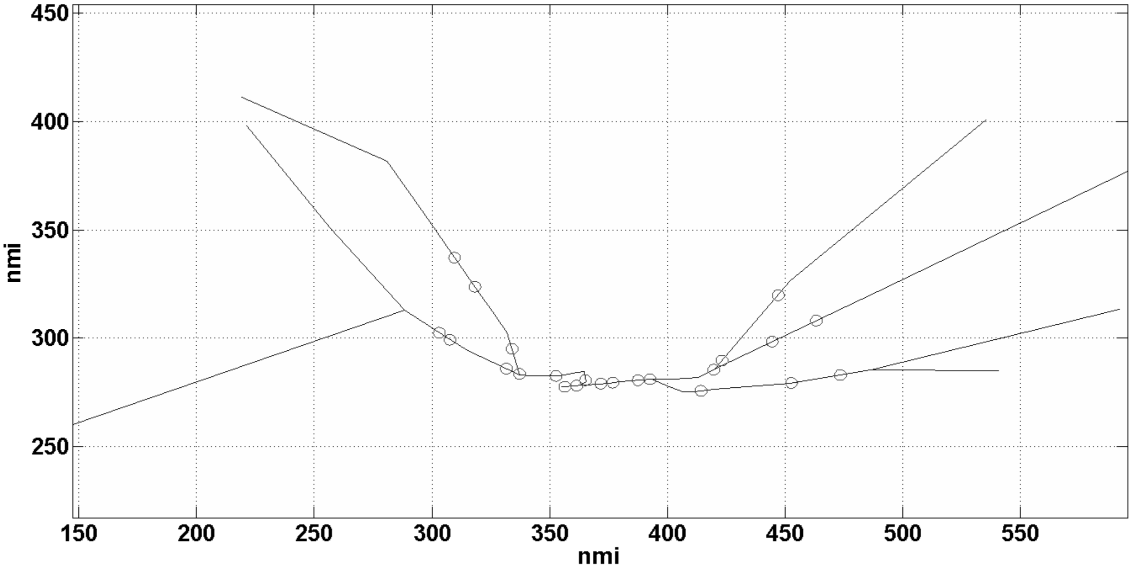


Figure 16: Positions of aircraft in LAX airspace at t=61 s.

where α_k are scalars and not all zero, $k = 1, \cdots, M$. The vector $\mathbf{U}$ define the direction and slope of the trajectory, where the components of $\mathbf{U}$ must satisfy the slope condition in each pairwise subspace,

$$\underline{s}_{ij} \leqslant \frac{u_j}{u_i} \leqslant \overline{s}_{ij}, \quad i, j = 1, \cdots, \ell \quad \text{and} \quad i < j.$$

to obtain a general form for α's, substitute for u_i and u_j from equation (19), then

$$\alpha_1 \left(e_{j1} - \overline{s}_{ij} e_{i1}\right) + \alpha_2 \left(e_{j2} - \overline{s}_{ij} e_{i2}\right) + \cdots + \alpha_M \left(e_{jM} - \overline{s}_{ij} e_{iM}\right) \leqslant 0, \tag{20}$$

$$\alpha_1 \left(\underline{s}_{ij} e_{i1} - e_{j1}\right) + \alpha_2 \left(\underline{s}_{ij} e_{i2} - e_{j2}\right) + \cdots + \alpha_M \left(\underline{s}_{ij} e_{iM} - e_{jM}\right) \leqslant 0. \tag{21}$$

For each pairwise subspace the above two inequalities must be satisfied. Therefore, for ℓ-dimensional space, there are $\ell(\ell - 1)$ inequalities which can be represented in matrix form.

$$\mathbf{F}\alpha \leqslant 0. \tag{22}$$

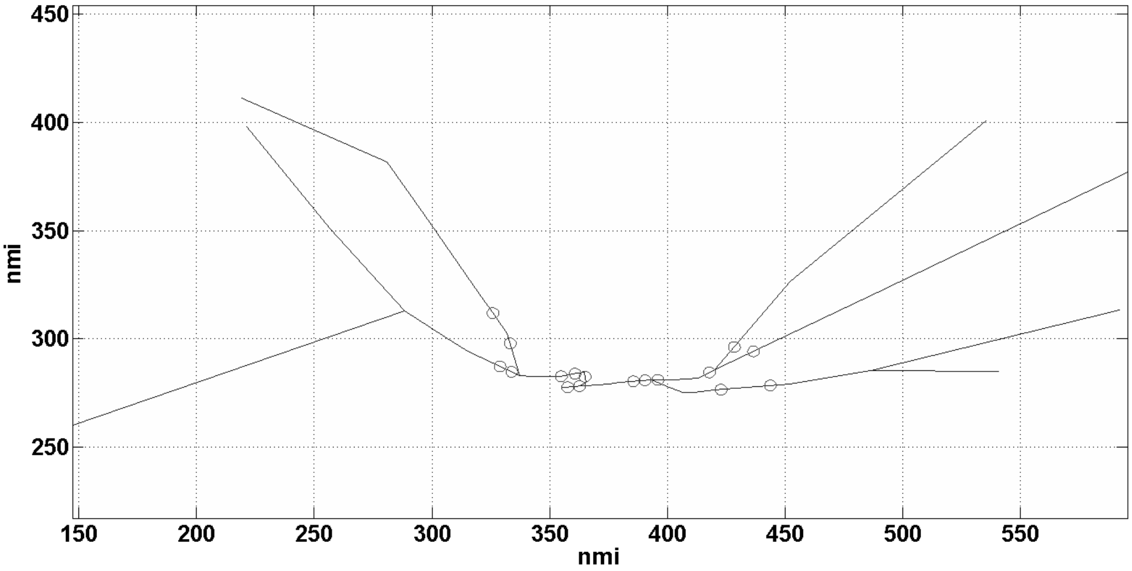


Figure 17: Positions of aircraft in LAX airspace at t=91.5 s.

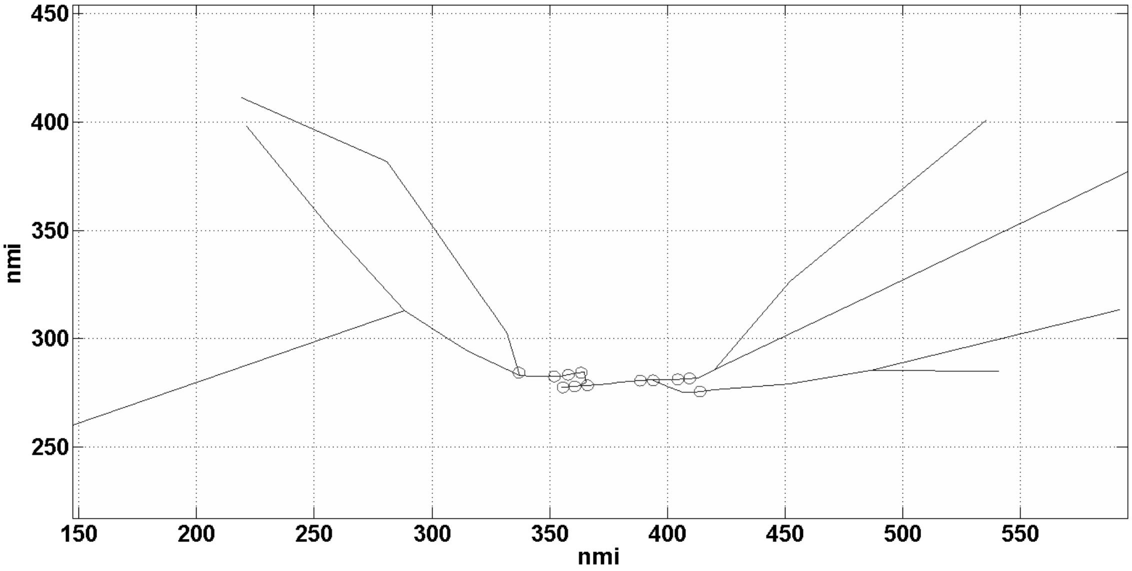


Figure 18: Positions of aircraft in LAX airspace at t=122 s.

In this case an optimization problem can be defined to look for a specific solution, i.e. minimizing the fuel consumption. Solving an optimization problem will increase the computation time.

VI. Discussion and Conclusion

The algorithm developed in Section III is searching for a feasible trajectory in all feasible sequences. Therefore, it is guaranteed to find a speed control solution, if such a solution exists. We have shown that the computation time is finite and can be done in parallel for each feasible sequence. The parallelism helps make it computationally practical. By increasing the number of aircraft, the number of feasible sequences will increase. But as it is discussed before, the likelihood of existence of a speed control solution in each sequence depends on the order of total arc length of assigned path to aircraft in the sequence. Therefore, for further reduction in computation time, a likelihood function can be defined for each feasible sequence and

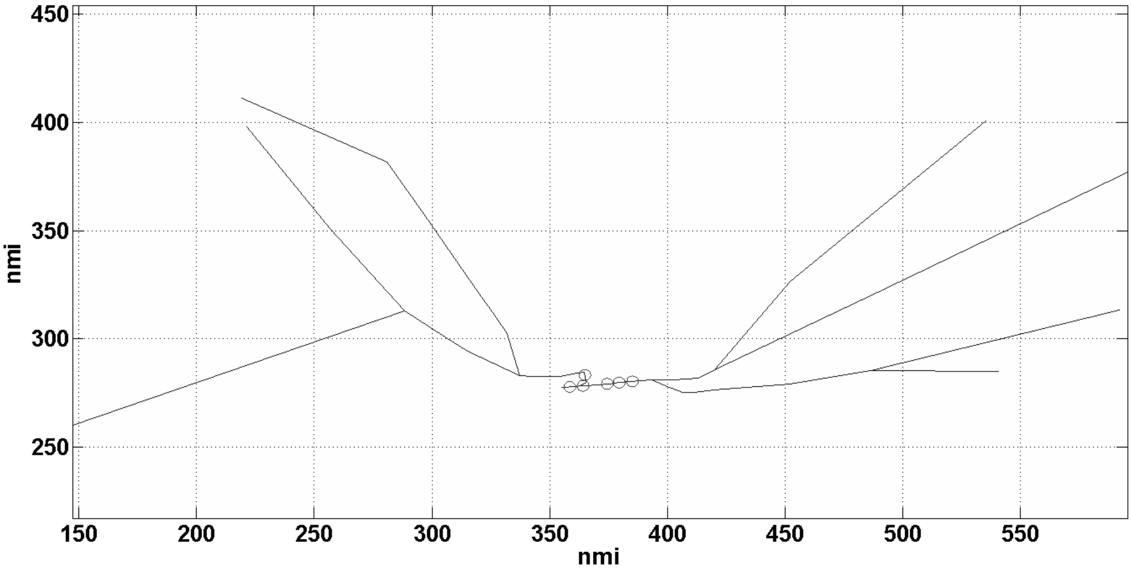


Figure 19: Positions of aircraft in LAX airspace at t=152.5 s.

then look for speed control solution in the sequences with the higher probability. In this paper, the effect of uncertainties such as wind did not considered. Since, the uncertainties will affect the size of the attainable cone set or the infeasible state set, then after considering the affects the algorithm can be used to find the solution if it exists. Also, the feasible speed range of aircraft will change when the aircraft approaches the terminal point, at which point the attainable cone set must be defined with more than two boundaries. This will increase the dimension of matrix A and B in equation (9) and the number of permanent constraints in each pairwise framework.

One extension for this algorithm is to calculate the speed control in real time. In real time, new aircraft are adding to the airspace while some others are living the airspace. Two approaches can be considered to find the speed control solution in the real time. If the computation time is small enough, after the arrival of new aircraft the algorithm can be run again to find the new speed control solution for all aircraft in the airspace. Otherwise, the algorithm needs to be modified to calculate the speed profile for the new aircraft while the speed profiles of the existing aircraft are assume to be known.

The algorithm also can be used in Flight deck Interval Management (FIM). In flight deck operation ATC issues an interval management clearance and flight crews manage spacing through speed adjustments generated by onboard FIM equipment until reaching a planned termination point. The interval management clearance issued by ATC is the minimum separation required, given the speed range of the aircraft on the flight deck, the infeasible state set and the attainable cone set can be defined. Therefore, this algorithm can be used to generate the aircraft speed profile.

If there is no speed control solution, then we have to use path control along with speed control. This algorithm can be modified to find the best aircraft for path modification in order to have the minimum changes in assigned path. The extension of the algorithm to path control is the topic for further research.

References

[1] Sussmann, H. J., “A maximum principle for hybrid optimal control problems,” *The 38th IEEE Conference on Decision and Control*, 1999, pp. 425–430.

[2] Administration, U. F. A., *Order JO 7110.65U, Air Traffic Control*, U.S. Dept. of Transportation, Washington, D.C., 2012.

[3] Sadovsky, A. V., Davis, D., and Isaacson, D. R., “Efficient Computation of Separation-Compliant Speed Advisories for Air Traffic Arriving in Terminal Airspace,” Technical Memorandum NASA/TM-2012-216033, NASA, Ames Research Center, Moffett Field, CA 94035-0001, USA, 2012.

[4] P. U. Lee, T. P., "Prediction of Traffic Complexity and Controller Workload in Mixed Equipage NextGen Environments," *SAGE Journal, Proceedings of the Human Factors and Ergonomics Society Annual Meeting, vol. 56 no. 1, 100-104*, 2012.

[5] Sadovsky, A. V., Davis, D., and Isaacson, D. R., "Optimal routing and control of multiple agents moving in a transportation network and subject to an arrival schedule and separation constraints," Technical Memorandum NASA/TM-2012-216032, NASA, Ames Research Center, Moffett Field, CA 94035-0001, USA, Oct. 2012.

[6] Sadovsky, A., Swenson, H., Haskell, W., and Rakas, J., "Optimal Time Advance in Terminal Area Arrivals: Throughput vs. Fuel Savings," *IEEE 30th Digital Avionics Systems Conference (DASC), Seattle, WA*, 2011.

[7] Isaacson, D. R. and Sadovsky, A. V., "Scheduling for Precision Air Traffic Operations: Problem Definition and Review of Prior Research," *(in progress)*.

[8] T. Farley, J. D. Foster, T. H. and Lee, K. K., "A time-based approach to metering arrival traffic to Philadelphia," *The 1st FIAA Aircraft Technology, Integration, and Operations Forum, Los Angeles, CA*, , No. AIAA-2001-5241, 2001.

[9] Diallo, O. N., "A Predictive Aircraft Landing Speed Model Using Neural Network," *The 31st IEEE/AIAA Conference on Digital Avionics Systems*, 2012.

[10] Moreau, D. C. and Roy, S., "A stochastic characterization of en route traffic flow management strategies," *AIAA Guidance, Navigation, and Control Conf., San Francisco, CA*, 2005.

[11] A. M. Bayen, C. J. Tomlin, Y. Y. and Zhang, J., "An Approximation Algorithm for Scheduling Aircraft with Holding Time," *The 43rd IEEE Conference on Decision and Control, Atlantis, Paradise Island, Bahamas*, 2004.

[12] J. Koeners, M. V., "Conflict resolution support for air traffic control based on solution spaces: Design and implementation," *The 27th IEEE/AIAA Conference on Digital Avionics Systems, St. Paul, MN*, 2008.

[13] Cone, A., "Effect of conflict resolution maneuver execution delay on losses of separation," *The 29th IEEE/AIAA Conference on Digital Avionics Systems, Salt Lake City, UT*, 2010.

[14] Koeners, "Exploring benefits of a separation assessment tool to achieve level 3 SA for speed advisories," *The 28th IEEE/AIAA Conference on Digital Avionics Systems, Orlando, FL*, 2009.

[15] S. Sastry, G. Meyer, C. T. J. L. D. G. and Pappas, G., "Hybrid control in air traffic management systems," *Proceedings of the 34th IEEE Conference on Decision and Control, New Orleans, LA*, 1995.

[16] C. Tomlin, G. Pappas, J. L. D. G. S. S., *Hybrid Systems IV, Pages 378-404*, Springer-Verlag, 1997.

[17] Kaushik Roy, A. M. B. and Tomlin, C. J., "Polynomial Time Algorithms for Scheduling of Arrival Aircraft," *AIAA Guidance, Navigation, and Control Conference and Exhibit, San Francisco, California*, 2005.

[18] E. Frazzoli, Z.-H. Mao, J.-H. O. and Feron, E., "Resolution of Conflicts Involving Many Aircraft via Semidefinite Programming," *JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS,Vol. 24, No. 1*, 2001.

[19] Chen, C., *Linear System Theory and Design*, Oxford University Press, 1999.