University of California
Los Angeles

# Speed and Path Control for Conflict-Free Flight in High Air Traffic Demand in Terminal Airspace

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Mechanical Engineering

by

**Ali Rezaei**

2013

## Abstract of the Dissertation

# Speed and Path Control for Conflict-Free Flight in High Air Traffic Demand in Terminal Airspace

by

**Ali Rezaei**

Doctor of Philosophy in Mechanical Engineering

University of California, Los Angeles, 2013

Professor Jason L. Speyer, Chair

To accommodate the growing air traffic demand, flights will need to be planned and navigated with a much higher level of precision than today's aircraft flight path. 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 research, an algorithm has been developed, set in the context of a Hybrid Control System (HCS) 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. Uncertainties such as weather are not considered but the algorithm can be modified to include uncertainties. The algorithm first computes all feasible sequences (i.e., all sequences that allow the given fleet of aircraft to reach destinations without violating the FAA's separation requirement) by looking at all pairs of aircraft. Then, the most likely sequence is determined and the speed control solution is constructed by a backward trajectory generation, starting with the aircraft last out and proceeds to the first out. This computation can be done for different sequences in parallel which helps to reduce the computation time. If such a solution does not exist, then the algorithm calculates a minimal path modification (known as *path control*) that will allow separation-compliance speed control. We will also prove that the algorithm will modify the path without creating a new separation violation. The new path will be generated by adding new waypoints in the airspace. As a byproduct, instead of minimal path modification, one can use the aircraft arrival time schedule to generate the sequence in which the aircraft reach their destinations.

The dissertation of Ali Rezaei is approved.

Alan J. Laub

Robert M'Closkey

Steven Gibson

Jason L. Speyer, Committee Chair

University of California, Los Angeles

2013

*To my wife . . .*

*who supported me each step of the way*

*To my parents . . .*

*for their unconditional support*

*To my children . . .*

*who always bring joy and happiness to my life*

# Table of Contents

# List of Figures

# List of Tables

# Acknowledgments

First of all, I would like to express my sincere appreciation to my advisor and committee chair, Professor Jason Speyer for invaluable support, patience and guidance throughout my Ph.D. at University of California-Los Angeles.

I am also extremely thankful to Dr. Alexander Sadovsky and Douglas Isaacson at NASA Ames research center, for their technical comments and also special thanks to Dr. Sadovsky for his editorial comments.

Last, but far from least, I want to express my deepest gratitude to my parents and my wife who generously provided me with their love, patience and encouragement. I am forever indebted to them.

This research was supported by UARC under ARP (Aligned Research Program).

# Vita

| | |
|---|---|
| 1975 | Born, Tehran, Iran. |
| 1997 | B.S. (Mechanical Engineering), University of Tehran, Tehran, Iran. |
| 2000 | M.S. (Mechanical Engineering in Applied Mechanic), Sharif University of Technology, Tehran, Iran. |
| 1997–1998 | Design Engineer, Khavar Industrial Group, Research and Develop Center, Tehran, Iran. |
| 1998-2000 | Research Assistance, Rail way Research Center, Tehran, Iran. |
| 1998–2000 | Design Engineer, Tarh Negasht Co., Tehran, Iran. |
| 2000–2006 | Design Engineer, TAM Co., Tehran, Iran. |
| Summer 2002 | Design Engineer, EDAG Co., Cologne, Germany. |
| 2006-2007 | Project Engineer, Pioneer Construction Inc., Valencia, California. |
| 2007-2008 | Research Assistance, Multidisciplinary Flight Dynamics and Control Laboratory, California State University, Los Angeles, California.<br>Sponsored by NASA and NSF |
| 2008 | M.S. (Mechanical Engineering in Systems and control), California State University, Los Angeles, California. |
| 2011 | Lecturer, California State University, Los Angeles, California. |
| 2011-2013 | Research assistance, Mechanical Engendering Department, UCLA. |
| 2012-Present | Lecturer, California State University, Long Beach, California. |

# Publications

- Rezaei, A., Sadovsky, A. V., Speyer, J., and Isaacson, D. R., *Separation-compliant speed control in terminal airspace*, AIAA Guidance, Navigation, and Control (GNC) Conference,Boston,MA, 2013.

- Rezaei, A., Sadovsky, A. V., Speyer, J., and Isaacson, D. R., *Separation-compliant in the terminal airspace, Part I: speed control*, In progress.

- Rezaei, A., Sadovsky, A. V., Speyer, J., and Isaacson, D. R., *Separation-compliant in the terminal airspace, Part II: speed and path control*, In progress.

# CHAPTER 1

# 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 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 $N!$ [24], as it requires a choice of an orthant[1] in an $N$-dimensional Euclidean space. Therefore, the forecast increase in traffic demand will result in a rapid growth of workload for the ATC[18], hindering NextGen. To afford the growing air traffic demand, flights need to be planned and navigated with much higher level of precision than they are under current operations. 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 is likely to entail having to readdress separation assurance.

In this paragraph, the recent research in ATM will be reviewed. They have been categorized based on the speed control and the path control in ATM. In speed control, the recent research effort [16] in ATM gives a strategy for managing the aircraft flow in order to main-

[1] 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-space. a closed orthant in $\mathbf{R}^N$ is a subset defined by constraining each Cartesian coordinate to be nonnegative or nonpositive.

tain the predefine sector capacity. The research used different methods to model the airspace. A directed graph model of airspace has been used in [14, 1] to estimate a time of arrival at waypoints for each aircraft. Hybrid system [2] has been used in [22, 11, 3] to address other aerospace problems. A neural network has been used in [9] to estimate the landing speed in order to maintain the minimum spacing of the leading and trailing aircraft. The authors of Ref. [13] used the similar model as this research to predict conflicts in a given airspace. Refs. [24, 25, 23, 12] contain propositions to automate the process of computing a speed profile for each aircraft in a given fleet. In path control Refs. [26, 5, 17, 15, 7] used path control in ATM to avoid severe weather. Another major research trend involving path control is the "free flight" operational concept. Under this concept, researchers try to determine a path for each aircraft in a way that maintains separation assurance [10, 19, 4, 28, 6].

A detailed flight path, parameterized by time in a way that assures separation for the entire fleet at all times, is aimed at designing automated decision support tools for the ATC. The goal is to alleviate the aforementioned workload increase. This research relies on the modeling framework of Hybrid Control Systems (HCS) [27]. In this framework, the arc length coordinate of each aircraft is mapped to one of state space axes (for more explanation see Chapter 2). Hence, a solution for a fleet of $N$ aircraft, each with a precise path prescribed, is ultimately a pair $\mathbf{X}(t)$, $\mathbf{V}(t)$ of time-dependent functions that take $N$-dimensional vector values, where

$$\mathbf{X}(t) = (x_1(t), x_2(t), \ldots, x_N(t)) \tag{1.1}$$

and

$$\mathbf{V}(t) = (V_1(t), V_2(t), \ldots, V_N(t)) \tag{1.2}$$

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 1.1.** *In what follows, the coordinates $x_i$ correspond to the* standard orthogonal basis $(1, 0, \ldots)$*,* $(0, 1, 0, \ldots), \ldots$ *of the Euclidean space of the appropriate dimension ($N$), and all*

[2] Hybrid model is used to model a system with discrete and continuous nature.

*projections that appear in this research are orthogonal projections onto a subspace spanned by a subset of this basis. Furthermore, all matrices that represent linear transformations 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 1.1.** *Given is an airspace modeled as a directed graph $G = (V, E)$ embedded in a Euclidean space of dimension 2 or 3, and a fleet of $N$ aircraft, each assigned to move along a specific path in the airspace, with the following data prescribed:*

- *the permissible speed range of each aircraft, $V_i \in [V_i^{min}, V_i^{max}]$, where $V_i$ is the speed of aircraft $i$,*
- *the minimal required separation distance assumed here to be the same for all pairs of aircraft,*
- *the airspace just contain merge points between two paths (no diverge point or crossing point exist in the airspace).*

*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*

$$\mathbf{X}(0) = \mathbf{X}^0 = \begin{bmatrix} x_1^0 \\ . \\ . \\ . \\ x_N^0 \end{bmatrix} \tag{1.3}$$

*and meet the minimum separation requirement during the flight until exits the airspace. The vector $\mathbf{V}(t)$ is called a speed control solution..*

Ref. [24] gives an algorithm to search for speed control solutions, but 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 research is an algorithm, defining a $N$-dimensional coordinate state space by using the above HCS model (explained in Chapter 2 in more detail) to find a speed profile which satisfies the permissible speed range for each aircraft $i$ such that each aircraft flies on its preassigned path from the initial point to the destination without violating the minimal required separation (Problem 1.1), if such a solution exists (Chapter 3). If speed profile $\mathbf{V}(t)$ does not exist, then the algorithm will find a minimal path modification and determine a permissible speed range for each aircraft to fly on assigned path, satisfy separation requirement and exit the airspace (Chapter 4). Using the state space (See Chapter 2 or Ref. [24] for a detailed exposition), problem 1.1 will be formulated as one in control theory with states (1.1) and controls (1.2). The constraint of permissible speed range defines the set of admissible controls and, consequently, defines in the state space the reachable set that, in absence of wind, turns out to be a pointed polyhedral cone (called the *attainable cone).* The algorithm will be described in detail in Chapter 3. The algorithm allows substantial parallelization, which promises to reduce the physical run time. In Chapter 4, an extension of the algorithm to include path control will be presented.

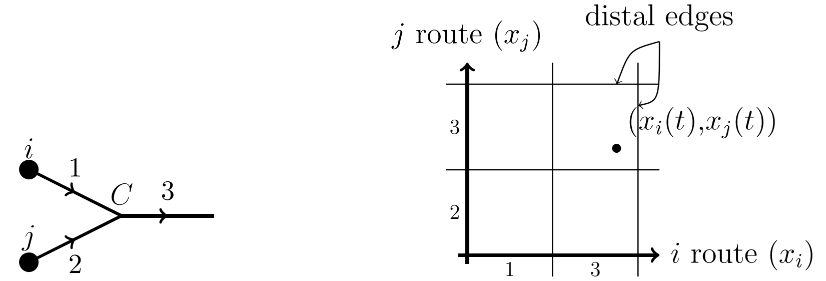
# CHAPTER 2

# HCS model in ATM problems

In Problem 1.1, there are $N$ aircraft in a given airspace. The airspace is modeled as a directed graph. A simple form of a directed graph is depicted in Figure 2.1a. 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 Euclidean coordinate space is depicted in Figure 2.9 (this plane will henceforth be called the *pairwise position state space* of the two aircraft). The following subsections introduce some subsets of this coordinate space that will be used in chapters 3 and 4.

## 2.1 State space in a Euclidean coordinate

Consider two aircraft, denoted $i$ and $j$, each flying on its own route. In this example, their routes, chosen to be polygonal paths for simpilicity, merge at point $C$. This situation is depicted in Figure 2.1a. The path assigned to aircraft $i$ consists of segments 1 and 3; the path assigned to aircraft $j$ consists of segments 2 and 3. The variable $x_i(t)$ is an arc length coordinate on the path of aircraft $i$; the variable $x_j(t)$, an arc length coordinate on the path of aircraft $j$. A simultaneous position of the two aircraft at time $t$ will be depicted as a point $(x_i(t), x_j(t))$ in the $x_i x_j$-coordinate space, depicted in Figure 2.1b. The lines in the $x_i x_j$-coordinate space that consist of the states in which at least one aircraft has reached the end of edge 3 are called *distal edges* and are denoted in Figure 2.1b. If a parametrized curve $(x_i(t), x_j(t))$ for $t_0 \leqslant t \leqslant t^{\text{exit}}$ representing the movement of two aircraft $i$ and $j$, reaches one of the distal edges, it means that the corresponding aircraft has reached its destination



(a) Merge point of aircraft $i$ and $j$ routes.

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

Figure 2.1: The position state space for two aircraft.

(which may or may not be an airport). The arc length of aircraft $i$ varies between $x_i^{\text{min}} = 0$ and $x_i^{\text{max}}$ where the $x_i^{\text{min}}$ corresponds to the beginning of the current route segment of aircraft $i$ and $x_i^{\text{max}}$ corresponds to the point where aircraft $i$ reaches its destination.

**Definition 2.1.** The pairwise position state space *is the set of all points* $(x_i(t), x_j(t))$ *where* $x_i(t) \in [0, x_i^{\text{max}}]$ *and* $x_j(t) \in [0, x_j^{\text{max}}]$.

**Definition 2.2.** The position state space *is a set of all points*

$$\left\{(x_1, x_2, \cdots, x_N) : x_i^{min} \leqslant x_i \leqslant x_i^{max}, 1 \leqslant i \leqslant N\right\}.$$

**Definition 2.3.** *A point in the position state space is called a* state*.*

**Definition 2.4.** *When a parameterized curve,* $\mathbf{X}(t)$ *representing the movement of* $N$ *aircraft, reaches hyperplane* $x = x_i^{\text{max}}$ *it means aircraft* $i$ *has reached its destination. Hyperplane* $x = x_i^{\text{max}}$ *is called* distal hyperplane*.*

**Definition 2.5.** *The intersection of all distal hyperplanes is the point where all aircraft have* *reached destination and is called the* distal state *and denoted by* $\mathbf{X}^{\max}$

$$\mathbf{X}(t^{exit}) = \mathbf{X}^{\max} = \begin{bmatrix} x_1^{\max} \\ \cdot \\ \cdot \\ \cdot \\ x_N^{\max} \end{bmatrix}. \tag{2.1}$$

**Definition 2.6.** *A trajectory is a continuous, piecewise differentiable curve*

$$\mathbf{X}(t) = \{x_1(t), \cdots, x_N(t)\}$$

*that starts from* $\mathbf{X}^0$ *and ends at* $\mathbf{X}^{\max}$*.*

## 2.2 The attainable cone

Let $\mathbf{X}(t)$ be a trajectory in a $N$-dimensional state space, at the point of differentiability of $\mathbf{X}(t)$, the permissible speed range constraint stated in problem 1.1 translates, for each 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

$$\underline{s}_{ij} = \frac{V_j^{min}}{V_i^{max}}, \tag{2.2}$$

$$\overline{s}_{ij} = \frac{V_j^{max}}{V_i^{min}}. \tag{2.3}$$

**Definition 2.7.** *In all pairwise position state space of aircraft $i$ and $j$ ($i, j = 1, \ldots, N$ and $i < j$), the slope of the trajectory $\mathbf{X}(t)$ for all $t$ at which $\mathbf{X}(t)$ is differentiable must be*

$$\underline{s}_{ij} \leqslant \frac{dx_j/dt}{dx_i/dt} \leqslant \overline{s}_{ij}. \tag{2.4}$$

*This condition hereafter called* slope condition.

**Definition 2.8.** *In the pairwise position state space of aircraft $i$ and $j$,*the pairwise attainable cone with initial state $(x_i^0, x_j^0)$ *is a set of all states* $\mathbf{X}$ *such that the slope of line segment* $\mathbf{X}^0\mathbf{X}$ *is in the range of* $[\underline{s}_{ij}, \overline{s}_{ij}]$ *and is denoted by*

$$\mathcal{A}_{ij}(\mathbf{X}^0).$$

The two bounds (2.2) and (2.3) can be described as half-planes (depicted by the shaded regions in Figure 2.2), defined by

$$\mathcal{L}_{ij}^{a} \,:\, x_j - x_j^0 \geq \underline{s}_{ij} \left(x_i - x_i^0\right) \tag{2.5}$$

and

$$\mathcal{U}_{ij}^{a} \,:\, x_j - x_j^0 \leq \overline{s}_{ij} \left(x_i - x_i^0\right). \tag{2.6}$$

The pairwise attainable cone for aircraft pair $i, j$ with initial state $(x_i^0, x_j^0)$ is the intersection of these two half-planes ,

$$\mathcal{A}_{ij}(\mathbf{X}^0) = \mathcal{U}_{ij}^{a} \cap \mathcal{L}_{ij}^{a} = \{X \,| A_{ij} X \leqslant B_{ij}\},$$

where

$$A_{ij} = \begin{bmatrix} \mathbf{0} & -\overline{s}_{ij} & \mathbf{0} & 1 & \mathbf{0} \\ \mathbf{0} & \underline{s}_{ij} & \mathbf{0} & -1 & \mathbf{0} \end{bmatrix}, \qquad B_{ij} = \begin{bmatrix} x_j^0 - \overline{s}_{ij} x_i^0 \\ -x_j^0 + \underline{s}_{ij} x_i^0 \end{bmatrix},$$

$$X = \begin{bmatrix} x_1 & x_2 & \cdots & x_i & \cdots & x_j & \cdots & x_N \end{bmatrix}^T,$$

and $\mathbf{0}$ is a row of zeros of an appropriate dimension. The pairwise attainable cone of pair $i$,$j$ is illustrated in Figure 2.3. The boundary lines of two half-planes (2.5) and (2.6) are given by:

$$\mathcal{H}_{ij}^{ua} \,:\, x_j = \overline{s}_{ij} \left(x_i - x_i^0\right) + x_j^0, \tag{2.7}$$

$$\mathcal{H}_{ij}^{la} \,:\, x_j = \underline{s}_{ij} \left(x_i - x_i^0\right) + x_j^0. \tag{2.8}$$

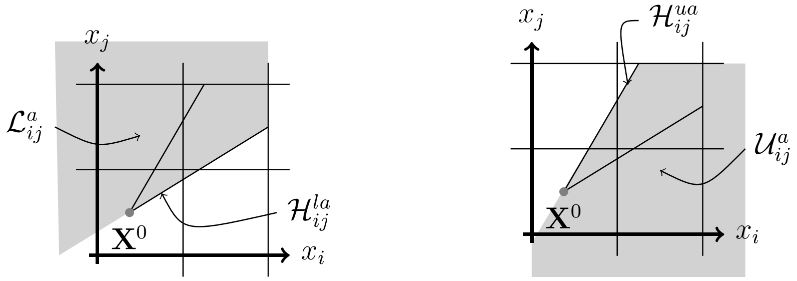


Figure 2.2: The half-planes associated with the minimal and maximal slopes.

Imposing constraints (2.5) and (2.6), we define the *attainable cone for $N$ aircraft with initial state* $\mathbf{X}^0$ ($N$ arbitrary).

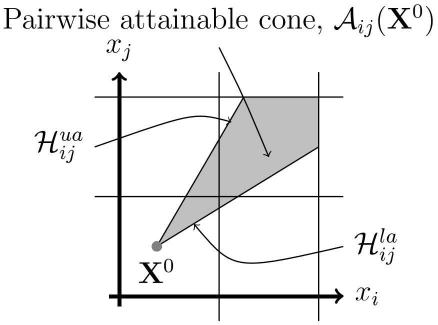


Figure 2.3: The pairwise attainable cone set of two aircraft $i$ and $j$.

**Definition 2.9.** *The attainable cone for $N$ aircraft with initial state $\mathbf{X}^0$ is the set of intersection of all pairwise attainable cones*

$$\mathcal{A}(\mathbf{X}^0) = \bigcap_{\substack{i,j=1 \\ i<j}}^{N} \mathcal{A}_{ij}(\mathbf{X}^0). \tag{2.9}$$

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

$$\mathcal{A}(\mathbf{X}^0) = \{\mathbf{X}\,|A\mathbf{X} \leqslant B\}\,. \tag{2.10}$$

**Definition 2.10.** *The state $\mathbf{X}$ is said to be* attainable from $\mathbf{X}^0$ *if*

$$\mathbf{X} \in \mathcal{A}(\mathbf{X}^0)$$

There are $\frac{N(N-1)}{2}$ unordered pairs of distinct indices $(i,j)$ and in each pair there are two $N$-dimensional half-spaces (2.5) and (2.6). Therefore, $A$ is an $N(N-1) \times N$ matrix and $B$ is an $N(N-1)$ vector. For example, the attainable cone in the case of two aircraft, indexed 1 and 2, is given by

$$A = \begin{bmatrix} -\overline{s}_{12} & 1 \\ \underline{s}_{12} & -1 \end{bmatrix}, \qquad B = \begin{bmatrix} x_2^0 - \overline{s}_{12}x_1^0 \\ -x_2^0 + \underline{s}_{12}x_1^0 \end{bmatrix}.$$

## 2.3 Separation-loss and separation-compliant state Sets

The constraints on control $\mathbf{V}(t)$ in state space are discussed in the previous section. In this section, the separation constraint between aircraft which is a constraint on the state $\mathbf{X}(t)$ will be explained.

**Definition 2.11.** *The set of all states* $\mathbf{X}$ *that violate the separation requirement is called the* separation-loss set*.*

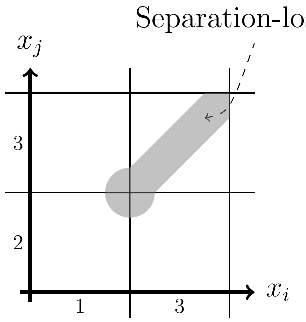


Figure 2.4: The separation-loss set for two aircraft $i$ and $j$.

The separation-loss set for the configuration shown in Figure 2.1a is depicted in Figure 2.4 and is computed by considering the following four cases.

Case 1. *Aircraft $i$ is on segment 1 and aircraft $j$ is on segment 2 and they are flying toward the waypoint.*

When both aircraft are near the end of their current route segments, they lose separation. This situation is illustrated in Figure 2.5.

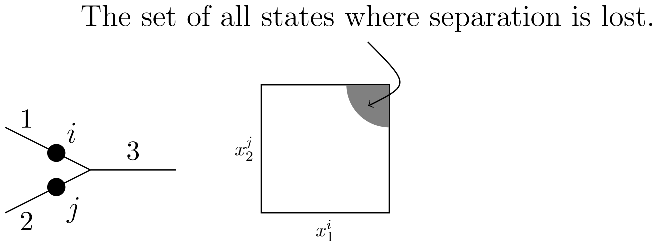


Figure 2.5: Separation-loss set in case 1

Case 2. *Aircraft $i$ is on segment 1 and aircraft $j$ is on segment 3.*

In this case the separation loss occurs if aircraft $i$ is near the end of its current route segment while aircraft $j$ is at the beginning of its current route segment. This situation is illustrated in Figure 2.6.

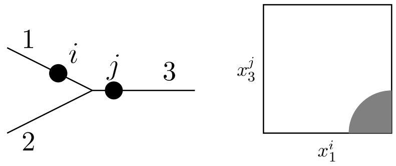


Figure 2.6: Separation-loss set in case 2

Case 3. *Aircraft $i$ is on segment 3 and aircraft $j$ is on segment 2.*

In this case the separation loss occurs if aircraft $i$ is near the beginning of its current route segment and aircraft $j$ is at the end of its current route segment. This situation is illustrated in Figure 2.7.

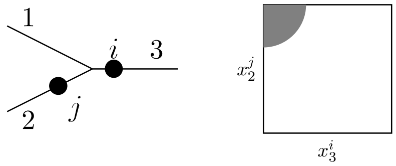


Figure 2.7: Separation-loss set in case 3

Case 4. *Both aircraft are flying on the segment 3.*

In this case the separation loss occurs if the distance between the two aircraft is less than the minimum required separation $r$. This situation is illustrated in Figure 2.8.

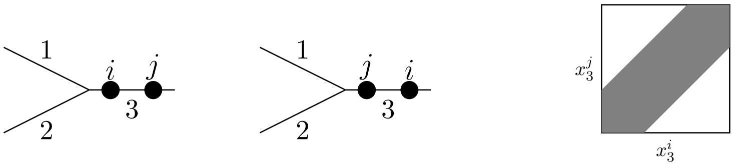


Figure 2.8: Separation-loss set in case 4

The pairwise position state space of aircraft $i$ and $j$ is constructed as depicted in Figure 2.9. The route of aircraft $i$ is segment 1 followed by segment 3 and for aircraft $j$ is segment 2 followed by segment 3. Next, the associated position state space of cases 1 through 4 are glued into the pairwise position state space as shown in Figure 2.10. The separation-loss set will be approximated with a polyhedral set. This approximation is depicted in Figure 2.10: the hyperplane $\mathcal{H}_{ij}^{uc}$ is the set of all states where the distance between aircraft $j$ and $i$ is equal to the minimum required separation overestimated by the approximation and aircraft $j$ is closer to its distal edge. Similarly, the hyperplane $\mathcal{H}_{ij}^{lc}$ is the set of all states where the distance between two aircraft is equal to the minimum required separation overestimated

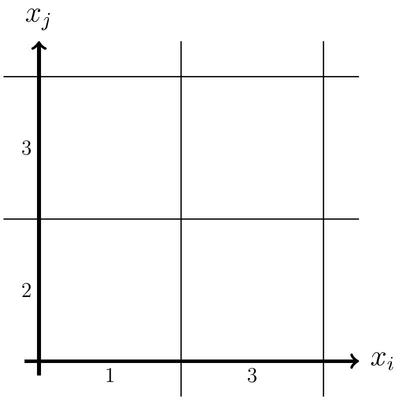


Figure 2.9: The pairwise position state space for a merge point

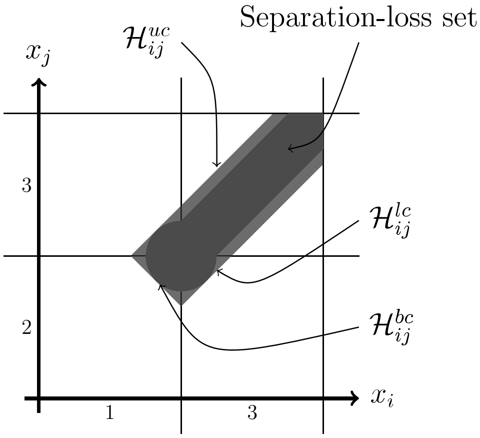


Figure 2.10: The separation-loss set between two airplanes in a merge point

by the approximation but aircraft $i$ is closer to its distal edge. In both cases, in order to keep the overestimated minimum required separation distance, both aircraft must fly with the same speed. Therefore, the slope of the orthogonal projection of these two hyperplanes onto $x_i x_j$-coordinate is equal to 1,

$$s_{ij} = \frac{V_j}{V_i} = 1.$$

Hence, the slope of the projection of hyperplane $\mathcal{H}_{ij}^{bc}$ which is perpendicular to projections of $\mathcal{H}_{ij}^{uc}$ and $\mathcal{H}_{ij}^{lc}$ and tangent to separation-loss set is equal to -1. The region bounded by hyperplanes $\mathcal{H}_{ij}^{uc}$, $\mathcal{H}_{ij}^{lc}$ and $\mathcal{H}_{ij}^{bc}$ will be called the *pairwise separation-loss set for aircraft pair* $i, j$, where $\mathcal{H}_{ij}^{bc}$, $\mathcal{H}_{ij}^{uc}$ and $\mathcal{H}_{ij}^{lc}$ are hyperplanes defined as follows:

$$\mathcal{H}_{ij}^{bc} = \left\{ \mathbf{X} \left| x_j = -x_i + d_{bc_{ij}} \right. \right\}, \tag{2.11}$$

$$\mathcal{H}_{ij}^{uc} = \left\{ \mathbf{X} \left| x_j = x_i + d_{uc_{ij}} \right. \right\}, \tag{2.12}$$

$$\mathcal{H}_{ij}^{lc} = \left\{ \mathbf{X} \left| x_j = x_i + d_{lc_{ij}} \right. \right\}, \tag{2.13}$$

where $d_{bc_{ij}}$, $d_{uc_{ij}}$ and $d_{lc_{ij}}$ are appropriate constants in line equations which are boundaries of polyhedral region. See Appendix for more detail.

**Definition 2.12.** *A state is said to be* infeasible *if it lies in the interior of the separation loss set, and is said to be* feasible *otherwise.*

**Definition 2.13.** *A trajectory*

1. *is said to be* feasible *if all states* $\mathbf{X}$ *on the trajectory are feasible.*
2. *is said to be* attainable from state $\mathbf{X}^0$ *if in all pairwise position state space of aircraft* $i$ *and* $j$ *(*$i, j = 1, \ldots, N$ *and* $i < j$*), the slope of the trajectory at all* $\mathbf{X}(t)$ *satisfies the slope condition.*

**Definition 2.14.** *There is a set of feasible states that can not serve as initial state for any trajectory which is feasible and attainable. The set of these points is called the* roof set.

The roof set for 2-dimensional subspace is illustrated in Figure 2.11. 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. The pairwise roof set is in part bounded by hyperplanes

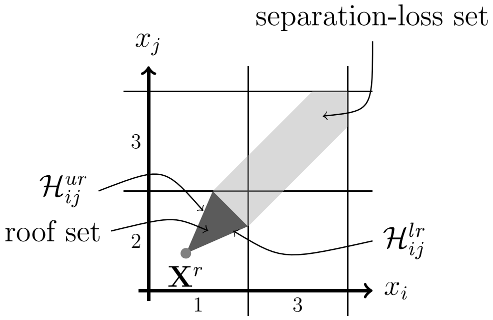


Figure 2.11: The separation-loss set and the roof set between aircraft $i$ and $j$ in a merge point

$\mathcal{H}_{ij}^{bc}$ (defined in equation (2.11)), $\mathcal{H}_{ij}^{ur}$ and $\mathcal{H}_{ij}^{lr}$, defined as:

$$\mathcal{H}_{ij}^{ur} = \left\{ \mathbf{X} \left| x_j = \overline{s}_{ij} x_i + d_{ur_{ij}} \right. \right\}, \tag{2.14}$$

$$\mathcal{H}_{ij}^{lr} = \left\{ \mathbf{X} \left| x_j = \underline{s}_{ij} x_i + d_{lr_{ij}} \right. \right\}, \tag{2.15}$$

where $d_{ur_{ij}}$ and $d_{lr_{ij}}$ are appropriate constants of line equations (see Appendix for more detail). The intersection of hyperplanes $\mathcal{H}_{ij}^{ur}$ and $\mathcal{H}_{ij}^{lr}$ will be called the *roof apex* and denoted $\mathbf{X}^r$.

**Definition 2.15.** *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 *for aircraft pair (i,j) and is denoted by* $\mathcal{C}_{ij}$*.*

In mathematical notation, the pairwise infeasible state set is defined as the intersection (shown shaded gray, part light and part dark, in Figure 2.11) of the four open half-planes (individually depicted and shaded gray in Figure 2.12)

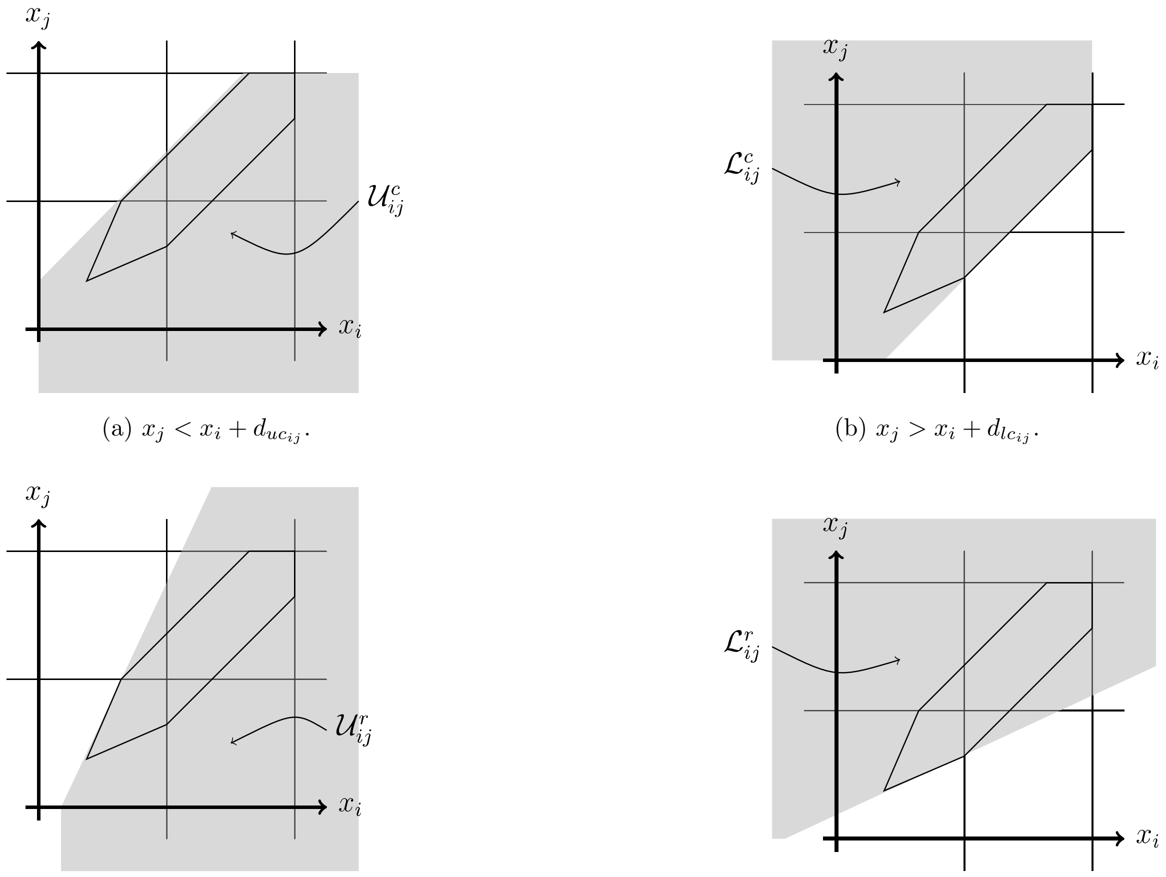
$$\mathcal{U}_{ij}^{c} = \left\{ X \left| x_j < x_i + d_{uc_{ij}} \right. \right\}, \tag{2.16}$$

$$\mathcal{L}_{ij}^{c} = \left\{ X \left| x_j > x_i + d_{lc_{ij}} \right. \right\}, \tag{2.17}$$

$$\mathcal{U}_{ij}^{r} = \left\{ X \left| x_j < \overline{s}_{ij} x_i + d_{ur_{ij}} \right. \right\}, \tag{2.18}$$

$$\mathcal{L}_{ij}^{r} = \left\{ X \left| x_j > \underline{s}_{ij} x_i + d_{lr_{ij}} \right. \right\}. \tag{2.19}$$

To determine the infeasible state set for an $N$-dimensional state space, one needs first to



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

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

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

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

Figure 2.12: Three half-planes of the pairwise separation-loss set.

compute the pairwise infeasible state sets

$$\mathcal{C}_{ij} = \mathcal{U}^c_{ij} \cap \mathcal{L}^c_{ij} \cap \mathcal{U}^r_{ij} \cap \mathcal{L}^r_{ij} = \{\mathbf{X}\,|C_{ij}\mathbf{X} > D_{ij}\}, \quad 1 \leqslant i \leqslant j \leqslant N,$$

where $C_{ij}$ is a 4-by-$N$ matrix and $D_{ij}$ is a 4-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:

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

Since, each $\mathcal{C}_{ij}$ is open, so is $\mathcal{C}$.

**Definition 2.16.** *The* separation-compliant state set *is defined as the complement of $\mathcal{C}$ in the position state space and, in what follows, is denoted $\mathcal{C}'$.*

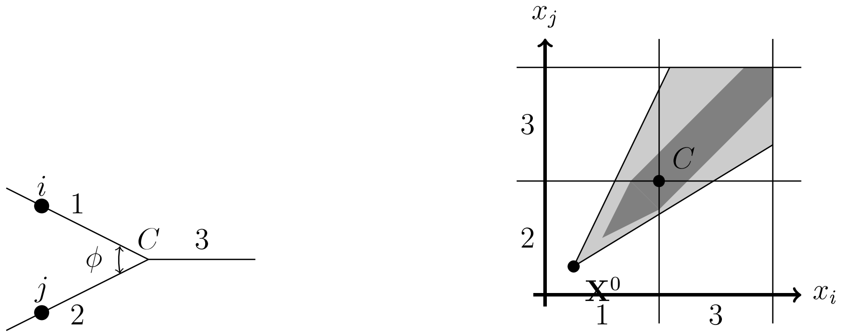
The separation-compliant state set is a closed set. In set-theoretic notation, we have

$$\mathcal{C}' = \bigcap_{\substack{i,j=1\\ i<j}}^{N} \mathcal{C}'_{ij}, \tag{2.20}$$

where,

$$\mathcal{C}'_{ij} = \{X \,|C_{ij}X \not> D_{ij}\}\,.$$

## 2.4 Trajectory



(a) Merge point.

(b) pairwise position state space between aircraft $i$ and $j$

Figure 2.13: (a) Two aircraft, $i$ and $j$, in an airspace. Paths (1,3) and (2,3) merge at point $C$. (b) The corresponding pairwise position state space, with arc length coordinates $x_i, x_j$. The separation-loss set is shaded in the darker gray. The light-gray cone is the set of states attainable, for the permissible speed ranges of $i$ and $j$, from the initial state $\mathbf{X}^0$.

For simplicity, the solution states of problem 1.1 is restricted to consist only of polygonal trajectories[1]. This restriction simplifies the computation procedure of verifying a trajectory for feasibility and attainability. Each rectilinear piece of the trajectory will be called a

[1] Continuous and piecewise linear.

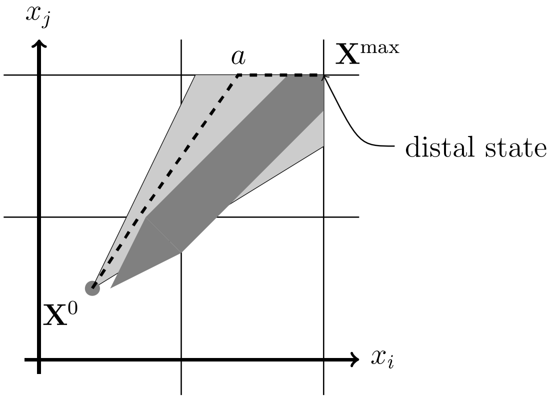


Figure 2.14: A trajectory which is feasible and attainable from initial state $\mathbf{X}^0$

*segment*[2] hereafter. As noted before, if the trajectory reaches one of the distal hyperplanes, then it means the corresponding aircraft has reached destination.

In Figure 2.14, a trajectory for a 2-dimensional position state space is shown as a dashed polygonal curve. When the trajectory reaches point $a$, the aircraft $j$ has reached destination. After that aircraft $i$ is the only aircraft still traveling toward destination until reaching the distal state $\mathbf{X}^{\max}$.

**Definition 2.17.** *A* speed control solution *is a feasible-attainable trajectory starts from initial state (1.3) and ends at final state (2.1).*

A speed control solution for a 2-dimensional position state space is depicted in Figure 2.14. The algorithm introduced in Chapter 3 looks for such a solution and generates a polygonal trajectory in the suitable $N$-dimensional position state space (See definition2.2).

## 2.5 Guiding hyperplanes

Each segment of the polygonal trajectory lies in a 1-dimensional submanifold of the $N$-dimensional state space. The algorithm in chapter 3 will compute this submanifold by intersecting of $N-1$ hyperplane suitably chosen from among boundaries of the attainable

[2] Note that the route segment is referring to the route in physical airspace while segment is refer to the $N$-dimensional position state space

cone set ($\mathcal{H}_{ij}^{ua}$ and $\mathcal{H}_{ij}^{la}$) and the pairwise infeasible state set (($\mathcal{H}_{ij}^{uc}$, $\mathcal{H}_{ij}^{lc}$, $\mathcal{H}_{ij}^{ur}$ and $\mathcal{H}_{ij}^{lr}$). These $N-1$ hyperplane are called *guiding hyperplane* hereafter. The trajectory will be constructed starting from distal state to the initial state .

### 2.5.1 Permanent guiding hyperplanes; the hyperplanes that bound the attainable cone

When the trajectory intersects with one of the hyperplanes $\mathcal{H}_{ij}^{ua}$ or $\mathcal{H}_{ij}^{la}$, the next segment (in time reverse) must stay on the intersected hyperplane in order to satisfy the slope condition (See equation 2.4). This situation is depicted for hyperplane $\mathcal{H}_{ij}^{ua}$ in Figure 2.15. In this figure the trajectory intersects the hyperplane $\mathcal{H}_{ij}^{ua}$ at point $b$. The slope of the next segment must also 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 $\mathcal{H}_{ij}^{ua}$. After point $b$, the trajectory must stay on this hyperplane and cannot leave this hyperplane. Therefore, hyperplanes $\mathcal{H}_{ij}^{ua}$ and $\mathcal{H}_{ij}^{la}$ will be called *permanent guiding hyperplanes.*

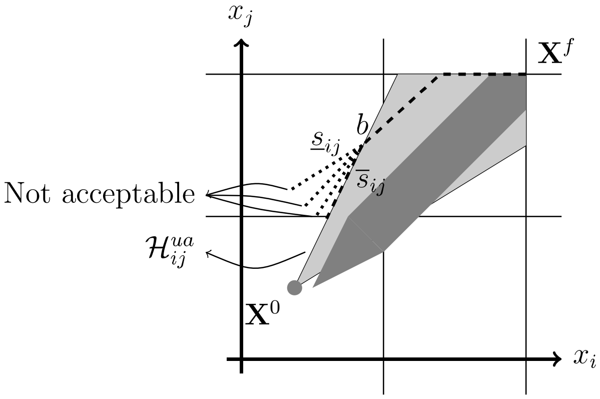


Figure 2.15: The trajectory must stay on the hyperplane $\mathcal{H}_{ij}^{ua}$ of the attainable cone after the intersection point $b$.

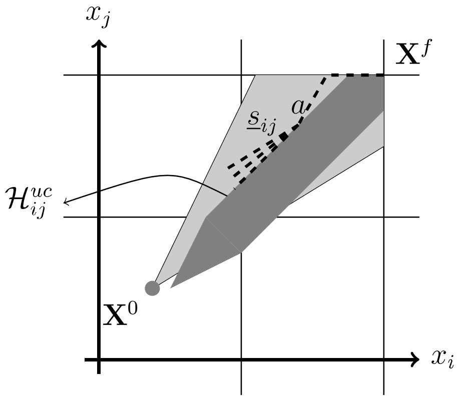


Figure 2.16: After the intersection point $a$, the trajectory can stay on the hyperplane $\mathcal{H}_{ij}^{uc}$ or leave it.

### 2.5.2 Temporary guiding hyperplanes; boundaries that bound the separation-loss set

When the trajectory intersects one of the hyperplanes $\mathcal{H}_{ij}^{uc}$ or $\mathcal{H}_{ij}^{lc}$, the slope of the next segment (in time reverse) 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 $\mathcal{H}_{ij}^{uc}$ is depicted in Figure 2.16. In addition, these hyperplanes generally end before trajectory reaches initial state $\mathbf{X}^0$. The hyperplanes $\mathcal{H}_{ij}^{uc}$ and $\mathcal{H}_{ij}^{lc}$ will be called *temporary guiding hyperplanes*.

### 2.5.3 Boundaries of the roof set

In the merge point, configuration depicted in Figure 2.1a, the two hyperplanes, $\mathcal{H}_{ij}^{ur}$ and $\mathcal{H}_{ij}^{lr}$, do not constrain the trajectory in the time-reversed trajectory generation method. As it has been shown in equations (2.14) and (2.15), the slopes of the boundaries of the roof set are $\overline{s}_{ij}$ and $\underline{s}_{ij}$. In time-reverse trajectory generation method, if the trajectory reaches to the intersection of $\mathcal{H}_{ij}^{uc}$ and $\mathcal{H}_{ij}^{ur}$ then the next segment of the trajectory must continues on $\mathcal{H}_{ij}^{ur}$ or away from it to comply slope condition. This is depicted in Figure 2.17. Similarly, when the trajectory reaches to the intersection of $\mathcal{H}_{ij}^{lc}$ and $\mathcal{H}_{ij}^{lr}$, then the trajectory can either stay on $\mathcal{H}_{ij}^{lr}$ or move away from it. Therefore, the hyperplanes $\mathcal{H}_{ij}^{ur}$ and $\mathcal{H}_{ij}^{lr}$ do not impose any

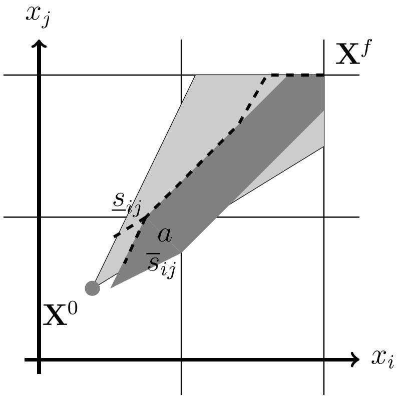
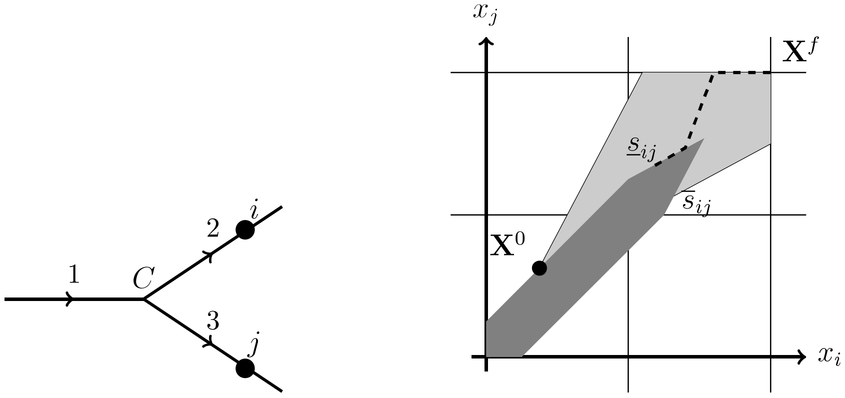


Figure 2.17: The hyperplane $\mathcal{H}_{ij}^{ur}$ does not impose any constraint on the trajectory.

restriction on the trajectory slope and are not considered as constraints.

In summary, among the six hyperplanes defined, four of them can be used as guiding hyperplanes to generate the polygonal trajectory. If the trajectory intersects one of the boundaries of the attainable cone set, $\mathcal{H}_{ij}^{ua}$ or $\mathcal{H}_{ij}^{la}$, then the trajectory must stays on them in order to satisfy the slope condition until it reaches the initial point ($\mathbf{X}^0$). Therefore, $\mathcal{H}_{ij}^{ua}$ and $\mathcal{H}_{ij}^{la}$ are called permanent guiding hyperplanes. In other hand, the boundaries of separation-loss set, $\mathcal{H}_{ij}^{uc}$ and $\mathcal{H}_{ij}^{lc}$ just limit the slope of the next segment of the trajectory. Hence, the trajectory can leave or stay on these boundaries. In addition, in the merge point case (Figure 2.1a) these two boundaries generally ends before trajectory reaches to the initial point, $\mathbf{X}^0$. Therefore, these two constraint, $\mathcal{H}_{ij}^{uc}$ and $\mathcal{H}_{ij}^{lc}$ are called temporary guiding hyperplanes. In the merge point case, the boundaries of the roof set do not impose any restriction on the slope of the trajectory and they are not considered as constraint. Note that, in diverge case these are hard constraint like the boundaries of attainable cone and the trajectory must continue on these boundaries until they be come inactive (i.e. intersection of $\mathcal{H}_{ij}^{uc}$ and $\mathcal{H}_{ij}^{ur}$.). A diverge configuration and the corresponding state space for two aircraft is depicted in Figure 2.18. A sample trajectory is depicted in Figure 2.18b. In the time-reverse, when the trajectory intersects one of the boundaries of the roof set it must stay on the boundary until it becomes inactive. In this research, airspace that only contains the merge point is considered (See Problem 1.1).



(a) Diverge point of aircraft $i$ and $j$ routes.

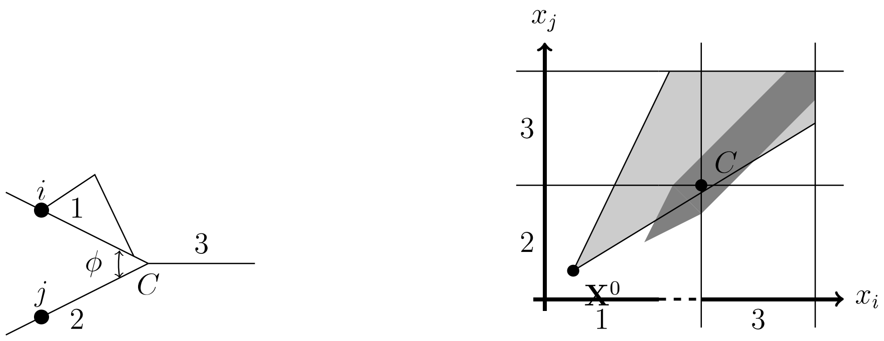
(b) The state space coordinate for two aircraft $i$ and $j$.

Figure 2.18: The state space for two aircraft in a diverge configuration.

## 2.6 Effect of path modification on state space of 2 aircraft

In previous sections, the arc length positions of the two aircraft along their respective paths have been defined. In this section, the geometric machinery needed when a speed control solution does not exist is introduced. When there is no speed control solution, then path control is required. To obtain a path control solution, one has to deform the existing paths of the aircraft, and it is preferable, from the operational point of view, to keep such deformations minimal in a sense defined later. If the path of one or more aircraft is deformed slightly, the resulting new arc length coordinate space will be similar to the original one. The geometric relationship between two state spaces that differ by such a small deformation is now examined.

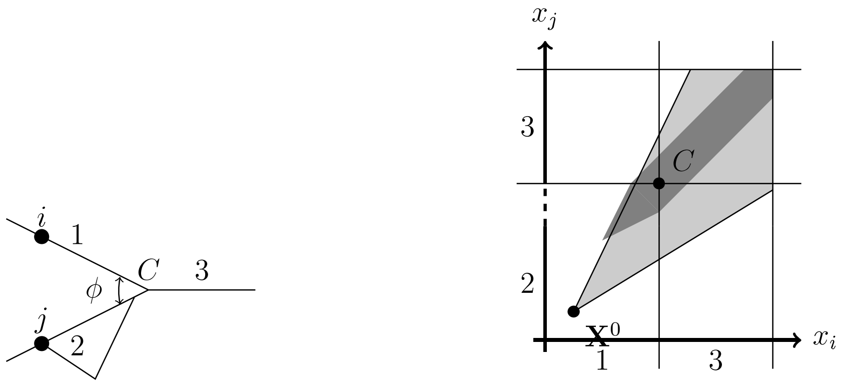
Consider modifying one or both of the edges 1 and 2 in Figure 2.13 by inserting new length *in front of the aircraft* without decreasing the merge angle and without bringing any parts of the two paths closer together than the allowed minimal separation. These modifications effect such parallel translations of the separation-loss set that move it away from $\mathbf{X}^0$. As a result, more of the rays that generate the attainable cone become clear of (in set-theoretic language, *disjoint from*) the separation-loss set; this is a precise geometric statement that can be proved. One such modification of the setting shown in Figure 2.13 is illustrated in Figures 2.19 and 2.20.



(a) Modified path of aircraft $i$

(b) Modified pairwise position state space between aircraft $i$ and $j$

Figure 2.19: Effect of path modification of aircraft $i$ on pairwise position state space between aircraft $i$ and $j$.



(a) Modified path of aircraft $j$

(b) Modified pairwise position state space between aircraft $i$ and $j$

Figure 2.20: Effect of path modification of aircraft $j$ on pairwise position state space between aircraft $i$ and $j$.

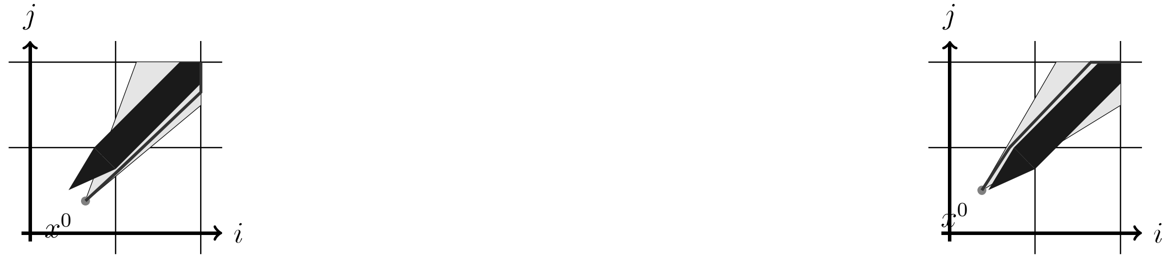
# CHAPTER 3

# Time-Reversed Trajectory Synthesis: a speed control algorithm

In this chapter, an algorithm to find a speed control solution (See Definition 2.17) will be constructed. Before, introducing the algorithm, the conditions for existence of a solution needs to be studied. In the 2-aircraft case of the state space defined in Section 2.4, there are two possible sequences for aircraft to exit the airspace: *I)* first aircraft $i$ exit then $j$, See Figure 3.1a, *II)* first aircraft $j$ exit then $i$, See Figure 3.1b.

**Definition 3.1.** *In a 2-dimensional state space, a sequence is called* pairwise separation-promising sequence *if there exist a speed control solution (feasible-attainable trajectory) that satisfies the sequence of existing the aircraft.*

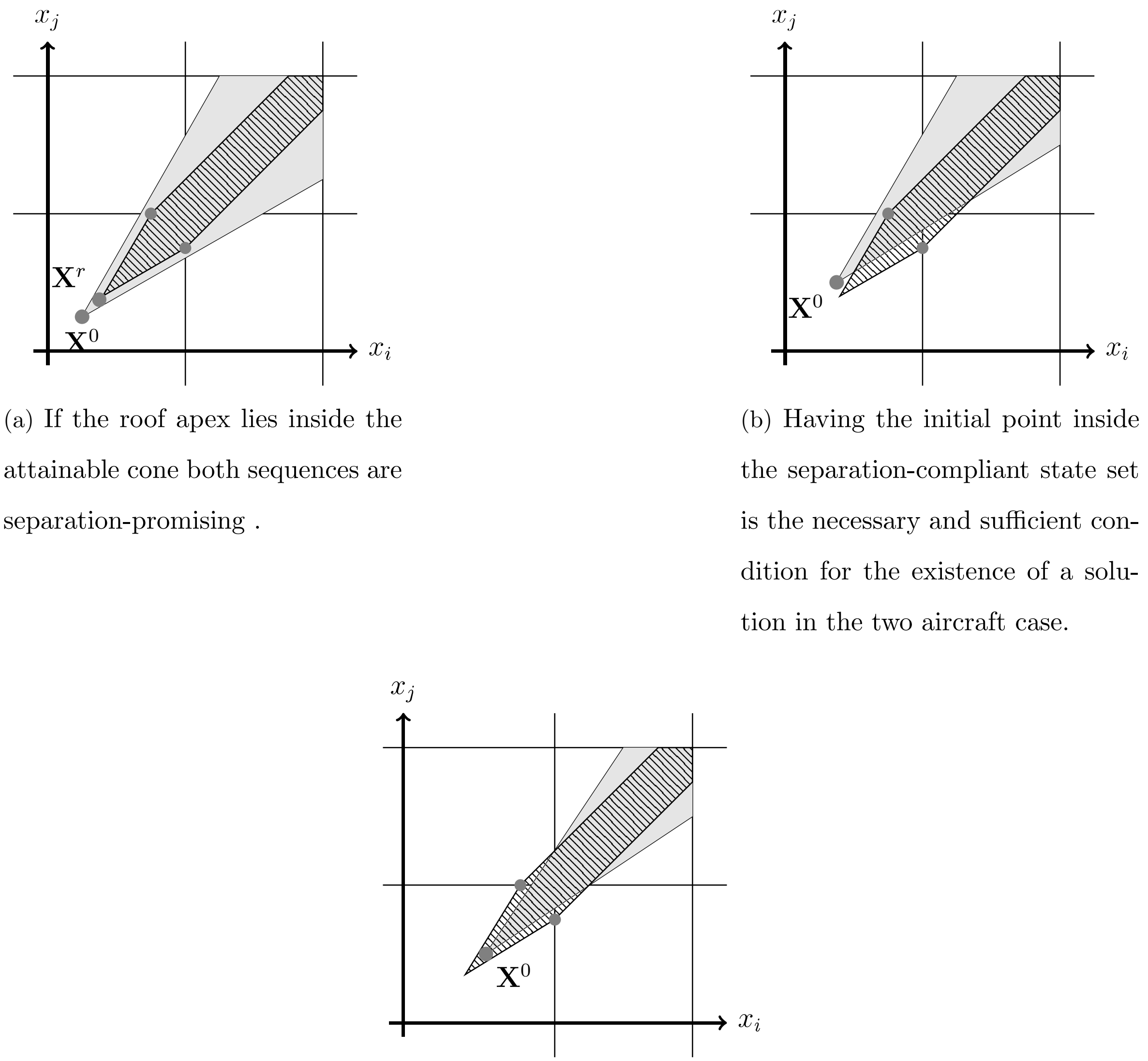
In Figures 3.1a and 3.1b, the feasible-attainable trajectories that satisfy the pairwise separation-promising sequences between aircraft $i$ and $j$ are shown.



(a) Aircraft $i$ leaves the airspace before aircraft $j$.

(b) Aircraft $j$ leaves the airspace before aircraft $i$.

Figure 3.1: separation-promising sequences in 2-dimensional airspace



(a) If the roof apex lies inside the attainable cone both sequences are separation-promising .

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

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

Figure 3.2: Pairwise position state space

If the roof apex, $\mathbf{X}^r$(See Section 2.3, Figure 2.11), lies in the pairwise attainable cone, $\mathbf{X}^r \in \mathcal{A}_{ij}$, then both sequences are separation-promising (see Figure 3.2a), otherwise, if $\mathbf{X}^0 \notin \mathcal{C}_{ij}$, then depending on the position of the initial point, $\mathbf{X}^0$, with respect to the separation-loss set, one of the two sequences is separation-promising. For instance, in Figure 3.2b there is one separation-promising sequence in which aircraft $j$ must exit the airspace before aircraft $i$. If $\mathbf{X}^0 \in \mathcal{C}_{ij}$ then none of the sequences is separation-promising. This is depicted in Figure 3.2c. In general, for the $N$-dimensional state space ($N \geqslant 2$), there are at most $N!$ sequences. Therefore, by looking at all pairwise position state spaces one can find all separation-promising sequences. [1] For instance, if just one of the pairwise position state spaces allows one separation-promising sequence, i.e. in subspace $(x_i, x_j)$ aircraft $j$ must exit the airspace before aircraft $i$, then all sequences that have aircraft $i$ exit before $j$ are not separation-promising and the number of separation-promising sequences among all $N$ aircraft will reduce by $\sum\limits_{i=0}^{N-1} i = \frac{N(N-1)}{2}$.

In the 2-aircraft case, the existence of a separation-promising sequence is a necessary and sufficient condition for existence of a speed control solution.

**Lemma 3.1.** *In the 2-aircraft case, if the permissible speed range* $[V_i^{min}, V_i^{max}]$ *of the aircraft* $i$ *has an overlap with the permissible speed range* $[V_j^{min}, V_j^{max}]$ *of the aircraft* $j$*, i.e. if*

$$[V_i^{min}, V_i^{max}] \cap [V_j^{min}, V_j^{max}] \neq \emptyset, \tag{3.1}$$

*then, the necessary and sufficient condition for the existence of a speed control solution is that the initial point* $\mathbf{X}^0$ *lies in the interior of the separation-compliant state set (see Definition 2.16), i.e.*

$$\mathbf{X}^0 \in \mathcal{C}'. \tag{3.2}$$

*Proof.* Equation (3.1), implies,

$$\underline{s}_{ij} \leqslant 1 \leqslant \overline{s}_{ij}. \tag{3.3}$$

If $\mathbf{X}^0$ does not lie in the interior of the separation-compliant state set (and, therefore, lies inside the pairwise infeasible state set), then no feasible-attainable trajectory can be initiated from $\mathbf{X}^0$. This situation is depicted in Figure 3.2c. From equations (2.14) and (2.15), in 2-aircraft case, the lines $\mathcal{H}_{ur_{ij}}$ and $\mathcal{H}_{lr_{ij}}$ are parallel to $\mathcal{H}_{ua_{ij}}$ and $\mathcal{H}_{la_{ij}}$, respectively and the slopes of the lines $\mathcal{H}_{uc_{ij}}$ and $\mathcal{H}_{lc_{ij}}$ are equal to one (see equations (2.12) and (2.12)).

[1] The method of calculating the separation-promising sequences for all $N$ aircraft by using all pairwise separation-promising sequences is described in detail in Chapter 4

Therefore, if $\mathbf{X}^0$ lies inside the separation-compliant state set, from equation (3.3) one obtains that at least one of the lines $\mathcal{H}_{uc_{ij}}$ or $\mathcal{H}_{lc_{ij}}$ entirely lies inside the pairwise 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 3.2a and 3.2b. □ □

**Corollary 3.1.** *In the 2-aircraft case, a speed control solution exists if and only if one of points*

$$\mathbf{X}_{ij}^{urc} = \mathcal{H}_{ij}^{uc} \cap \mathcal{H}_{ij}^{ur},$$

*and*

$$\mathbf{X}_{ij}^{lrc} = \mathcal{H}_{ij}^{lc} \cap \mathcal{H}_{ij}^{lr},$$

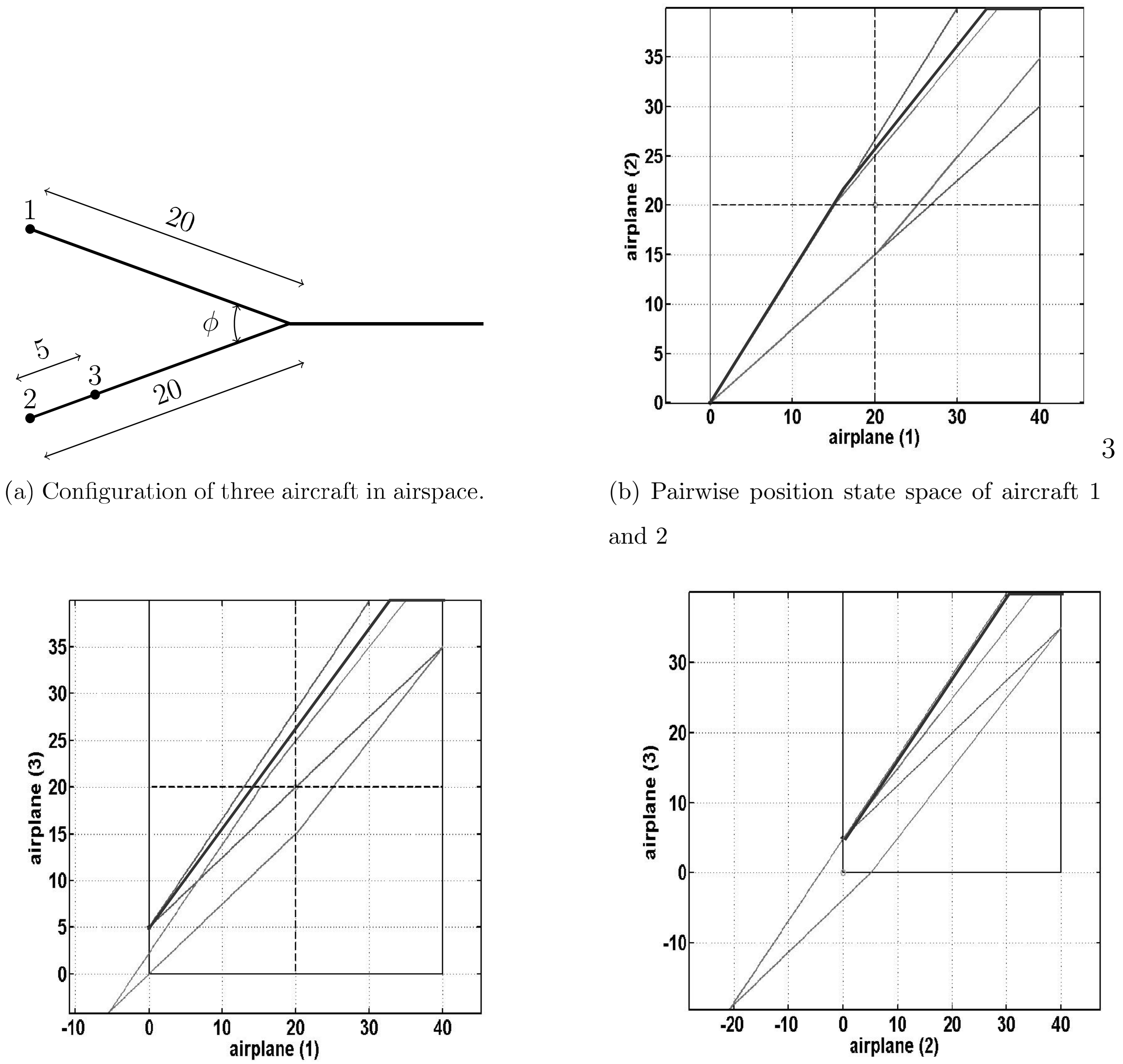
*is attainable.*

In the $N$-dimensional case, condition (3.2) is necessary but not sufficient for the existence of a speed control solution. This can be shown by an example. Consider three aircraft, numbered 1,2 and 3, positioned as shown in Figure 3.3a. The speed range for each aircraft is as follow

$$300 \leqslant V_1 \leqslant 400(mi/hr)$$
$$300 \leqslant V_2 \leqslant 400(mi/hr)$$
$$300 \leqslant V_3 \leqslant 350(mi/hr)$$

The minimum required separation distance is 5 nmi.

There are 3 pairwise position state spaces for this example shown in Figure 3.3: one can find a speed control solution in each pairwise position state space. In the 3-dimensional state space of this problem, however, the feasible state set dose not contain any of the distal edges or the distal point. Therefore, there is no speed control solution (feasible-attainable trajectory) in 3-dimensional state space. The 3-dimensional state space is depicted in Figure 3.4. Hence, condition (3.2) is necessary but not sufficient condition for existence of a solution in $N$-dimensional case ($N > 2$). In the $N$-dimensional case, if a feasible-attainable trajectory exists for a sequence, then this trajectory is a speed control solution.



(a) Configuration of three aircraft in airspace.

(b) Pairwise position state space of aircraft 1 and 2

(c) Pairwise position state space of aircraft 1 and 3

(d) Pairwise position state space of aircraft 2 and 3

Figure 3.3: The configuration of three aircraft in a given airspace and pairwise position state spaces.

**Remark 3.1.** *In $N$-dimensional case, existence of a separation-promising sequence of arrival at destination among $N$ aircraft grantees the existence of speed control solution in all pairwise position state spaces (necessary condition) but it does not imply the existence of a speed*

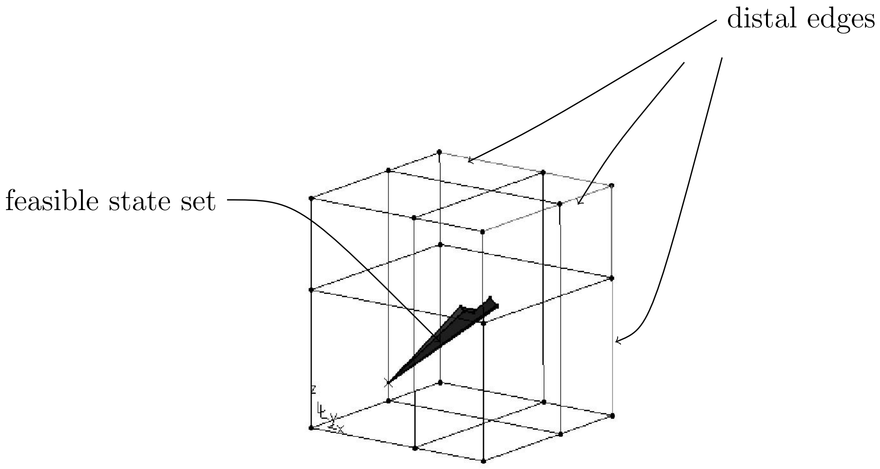


Figure 3.4: 3-dimensional position state space of three aircraft example.

*control solution in $N$-dimensional position state space.*

The algorithm first generates all the separation-promising sequences, then looks for the existence of a polygonal feasible-attainable trajectory in each separation-promising sequence. After finding all the separation-promising sequences, searching for a feasible-attainable trajectory over all the separation-promising sequences can be done in parallel. This parallelism will reduce the computation time. If there are $N-1$ linearly independent equality constraints for each segment, then there is just one 1-dimensional intersection to satisfy all constraints and the polygonal trajectory is a 1-dimensional subspace of the $N$-dimensional feasible state set. The case with multidimensional intersection results multi-solution for speed profile and will be discussed in Section 3.3.

Assume all $N$ aircraft in the Problem 1.1 have reached the destination. Then, the state vector is represented by equation (2.1). Also assume one of the separation-promising sequences is:

$$\{i, j, \ldots, \ell, \ldots, k, m\}. \tag{3.4}$$

Then aircraft $m$ is the first one to reach its destination and aircraft $i$ is the last one. Note that, the likelihood of the existence of a speed control solution in a separation-promising sequence (3.4) is higher than the other separation-promising sequences if

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

where $L_i = x_i^{\max} - x_i^0$ is the total arc length of the path assigned to the aircraft $i$. Hence, aircraft $i$ is the furthest and aircraft $m$ is the closet one to the destination point. Similarly, the likelihood of the existence of a speed control solution in separation-promising sequence (3.4) is lower than the other separation-promising sequences if

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

Therefore, the algorithm starts from the separation-promising sequence with higher likelihood of the existence of a speed control solution.

## 3.1 The algorithm: how it works

### 3.1.1 The setting of the algorithm and notation

Henceforth, we assume that the aircraft have been numbered $1, 2, \ldots, N$ in the order of the separation-promising sequence found above. Thus, aircraft 1 is the first to reach its destination (whether an airport or not) and exit the model; aircraft 2, the second; aircraft $N$, the last. As before, the initial time of the model will be denoted $t_0$. The time at which the $n$-th aircraft ($n = 1, \ldots, N$) exits the model will be denoted $t_n^{\text{exit}}$. It follows from the accepted numeration of the aircraft that

$$t_{n-1}^{\text{exit}} \leq t_n^{\text{exit}}, \quad n = 2, \ldots, N.$$

The sought speed control solution will be constructed in the form of a collective trajectory

$$\mathbf{X}(t) = [x_1(t), x_2(t), \ldots, x_N(t)]^T,$$

where $x_n(t)\,(n = 1, \ldots, N)$, while defined on the entire interval $[t_0, t_N^{\text{exit}}]$, stops playing a role in the model after time $t_n^{\text{exit}}$ by remaining constant and being no longer subject to any constraints (e.g., separation).

The algorithm will construct first the portion of the trajectory defined for the latter time interval $\left[t_{N-1}^{\text{exit}}, t_N^{\text{exit}}\right]$ (with only aircraft $N$ remaining in the model), then the portion defined for the earlier interval $\left[t_{N-2}^{\text{exit}}, t_{N-1}^{\text{exit}}\right]$ (with only aircraft $N$ and $(N-1)$ in the model), and so on, from the later intervals to the earlier ones. For this reason, the algorithm is called *Time-Reversed Feasible-attainable Trajectory Synthesis.*

The trajectory portion on $\left[t_{N-1}^{\text{exit}}, t_N^{\text{exit}}\right]$ is a rectilinear segment. For $n < N$, the trajectory portion on $\left[t_{n-1}^{\text{exit}}, t_n^{\text{exit}}\right]$ is a polygonal path that can consist of one or more rectilinear segments.

The following notation will be used for brevity:

$$\mathbf{X}_n := \mathbf{X}\left(t_n^{\text{exit}}\right), \quad n = 1, \ldots, N.$$

The algorithm first finds the polygonal paths $\mathbf{X}_{n-1}\mathbf{X}_n$ without a time parameterization. The found trajectory is then parameterized by time.

**Remark 3.2.** *The coordinates are chosen so that the lower bound of each arc length coordinate $x_n$ is 0, $x_n^{\min} = 0$ and the upper bound is distal hyperplane $x_n = x_n^{\max}$ (see Definition 2.4). The hyperplanes $x_n = x_n^{\min}$ and $x_n = x_n^{\max}$ $(n = 1, \ldots, N.)$ bound a parallelotope. The intersection of this parallelotope, the attainable cone with initial state $\mathbf{X}^0$ and the separation-compliant state set $\mathcal{C}'$ is called* feasible state set*.*

### 3.1.2 The steps of the algorithm

1. Find all separation-promising sequences

   (a) Generate all pairwise position state spaces and determine pairwise separation-promising sequences.

   (b) Find all separation-promising sequences among $N$ aircraft, sorted in descending order of aircraft path length. If no separation-promising sequence exist then there is no speed control solution.

2. Define $\mathbf{X}_N$ to be the distal point, $\mathbf{X}^{\max}$, in the feasible state set specified in remark 3.2.

3. The trajectory portion $\mathbf{X}_{n-1}\mathbf{X}_n$ for $1 \leqslant n \leqslant N$ will be sought in the form of a polygonal curve with the vertices (their total number will be denoted $M_n$) that are indexed as follows:

$$\mathbf{X}_{n-1} = \mathbf{Y}_n^{M_n}, \quad \dots, \quad \mathbf{Y}_n^1, \quad \mathbf{Y}_n^0 = \mathbf{X}_n.$$

The trajectory portion $\mathbf{X}_{n-1}\mathbf{X}_n$ is illustrated in Figure 3.5. Each segment $\mathbf{Y}_n^m\mathbf{Y}_n^{m+1}$ $(0 \leqslant m \leqslant M_n - 1)$ will be constructed as a segment on the line obtained by intersecting $(N-1)$ suitable hyperplanes. Therefore, the segment $Y_n^m Y_n^{m+1}$ will be found in the parameterized form

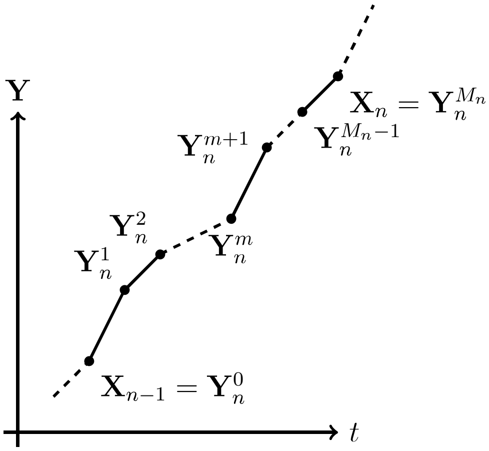


Figure 3.5: Portion $\mathbf{X}_{n-1}\mathbf{X}_n$ of the trajectory plotted verses time.

$$Y_n^{m+1} = Y_n^m - \mu_n^m \mathbf{d}_n^m,$$

where $\mu_n^m$ is a scalar parameter and $\mathbf{d}_n^m$ is a director (whose Euclidean product with the vector $\mathbf{X}^0\mathbf{X}^{\max}$ is positive). The line containing the director $\mathbf{d}_n^m$ will be found by intersecting a system (denoted by $\mathbb{H}_n^m$) of suitably chosen $(N-1)$ hyperplanes, hereafter called *the guiding hyperplanes for segment* $Y_n^m Y_n^{m+1}$. Since, in segment $Y_n^m Y_n^{m+1}$, $(n-1)$ of the aircraft have reached their destinations, but the others have not, $(n-1)$ of guiding hyperplanes are given by

$$x_k = x_k^{\max}, \quad k = 1, \dots, n-1. \tag{3.5}$$

The remaining $(N-n)$ guiding hyperplanes will be chosen from among $\mathcal{H}_{ua_{ij}}$, $\mathcal{H}_{la_{ij}}$, $\mathcal{H}_{uc_{ij}}$ and $\mathcal{H}_{lc_{ij}}$ where $i, j = n, \dots, N$ and $i < j$.

Suppose $Y_n^m$ is the most recently computed vertex, and suppose $\mathbb{H}_n^m$ has also been computed. (For $n = N$, the system $\mathbb{H}_N^1$ consists of the distal hyperplanes $x_k = x_k^{\max}, \quad k = 1, \ldots, N-1$.)

- The intersection of the guiding hyperplanes in $\mathbb{H}_n^m$ gives a line, oriented by a director $\mathbf{d}_n^m$, which we chose to have a positive scalar product with the vector $\mathbf{X}^0\mathbf{X}^{\max}$.
- To compute $\mu_n^m$ and $\mathbb{H}_n^{m+1}$, we consider these two different cases: (a) $m = M_n - 1$, and (b) $m < M_n - 1$.

  (a) $m = M_n - 1$. (This is the case $\mathbf{Y}_n^{m+1} = \mathbf{X}_{n-1}$.)
  At state $\mathbf{Y}_n^{m+1} = \mathbf{X}_{n-1}$, there are $p = (N - n + 1)$ aircraft that have not reached their destinations. Hence, there are $\frac{p(p-1)}{2}$ pairwise position state spaces between these aircraft and there are $2p(p-1)$ hyperplanes that can be considered as guiding hyperplanes (see Equations (2.7), (2.8), (2.12), (2.13) and Section 2.5). The ray

  $$\mathbf{Y}_n^m - \lambda \mathbf{d}_n^m, \quad \lambda > 0,$$

  intersects one or more of the hyperplanes in the $\frac{p(p-1)}{2}$ pairwise position state spaces. Every such intersection is a single point (intersection of $N-1$ guiding hyperplanes and the new hyperplane). Consider only such points that, regarded as states, are attainable from $\mathbf{X}^0$ and feasible. Each of these states corresponds to a specific value of $\lambda$. Let $\mu_n^m$ be the smallest of these values, and let $\mathcal{G}_n^m$ denotes the corresponding hyperplane.
  The system $\mathbb{H}_n^{m+1}$ is now obtained by replacing the distal hyperplane $x_{n-1} = x_{n-1}^{\max}$ in $\mathbb{H}_n^m$ with $\mathcal{G}_n^m$.

  (b) $m < M_n - 1$.
  In this case, at state $\mathbf{Y}_n^{m+1}$, there are $p = (N - n)$ aircraft that have not reached their destinations. The ray

  $$\mathbf{Y}_n^m - \lambda \mathbf{d}_n^m, \quad \lambda > 0,$$

intersects one or more of the hyperplanes in the $\frac{p(p-1)}{2}$ pairwise position state spaces. Every such intersection is a single point. Consider only such points that, regarded as states, are attainable from $\mathbf{X}^0$ and feasible. Each of these states corresponds to a specific value of $\lambda$. Let $\mu_n^m$ be the smallest of these values, and let $\mathcal{G}_n^m$ denotes the corresponding hyperplane, and let

$$\mathbf{Y}_n^{m+1} = \mathbf{Y}_n^m - \mu_n^m \mathbf{d}_n^m.$$

The system $\mathbb{H}_n^{m+1}$ is now obtained as follows. The hyperplane $\mathcal{G}_n^m$ replaces exactly one of hyperplanes $\mathcal{H}_{ij}^{cl}, \mathcal{H}_{ij}^{cu}$ in $\mathbb{H}_n^m$ in the following order

i. If there exist one in the pairwise position state space $(i, j)$ corresponding to $\mathcal{G}_n^m$.

ii. If there exist one that imposes a constraint which is inactive at $\mathbf{Y}_n^{m+1}$.

iii. If none of cases above is true, choose one with the largest value of $(i + j)$.

4. If $\mu_n^m$ not found, report, "No speed control solution found for separation-promising sequence $\{1, 2, \ldots, N\}$". If $\mathbf{X}^0$ has been reached, parametrized the computed trajectory by time and report the speed profile of each aircraft.

The steps of the algorithm is depicted in a flow chart in Figure 3.6.

**Remark 3.3.** *Since, the guiding hyperplanes in the system $\mathbb{H}_n^m$ for $n = 1, \ldots, N$ and $m = 0, \ldots, M_n$ are chosen from different 2-dimensional subspaces then they are independent and there is a 1-dimensional intersection for each segment. Except for cases when some temporary guiding hyperplane become inactive. This case will be discussed later.*

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

**Lemma 3.2.** *In $N$ aircraft case, the speed control solution exists if and only if there exists at least a separation-promising sequence for all $N$-aircraft and a feasible-attainable trajectory for the separation-promising sequence.*

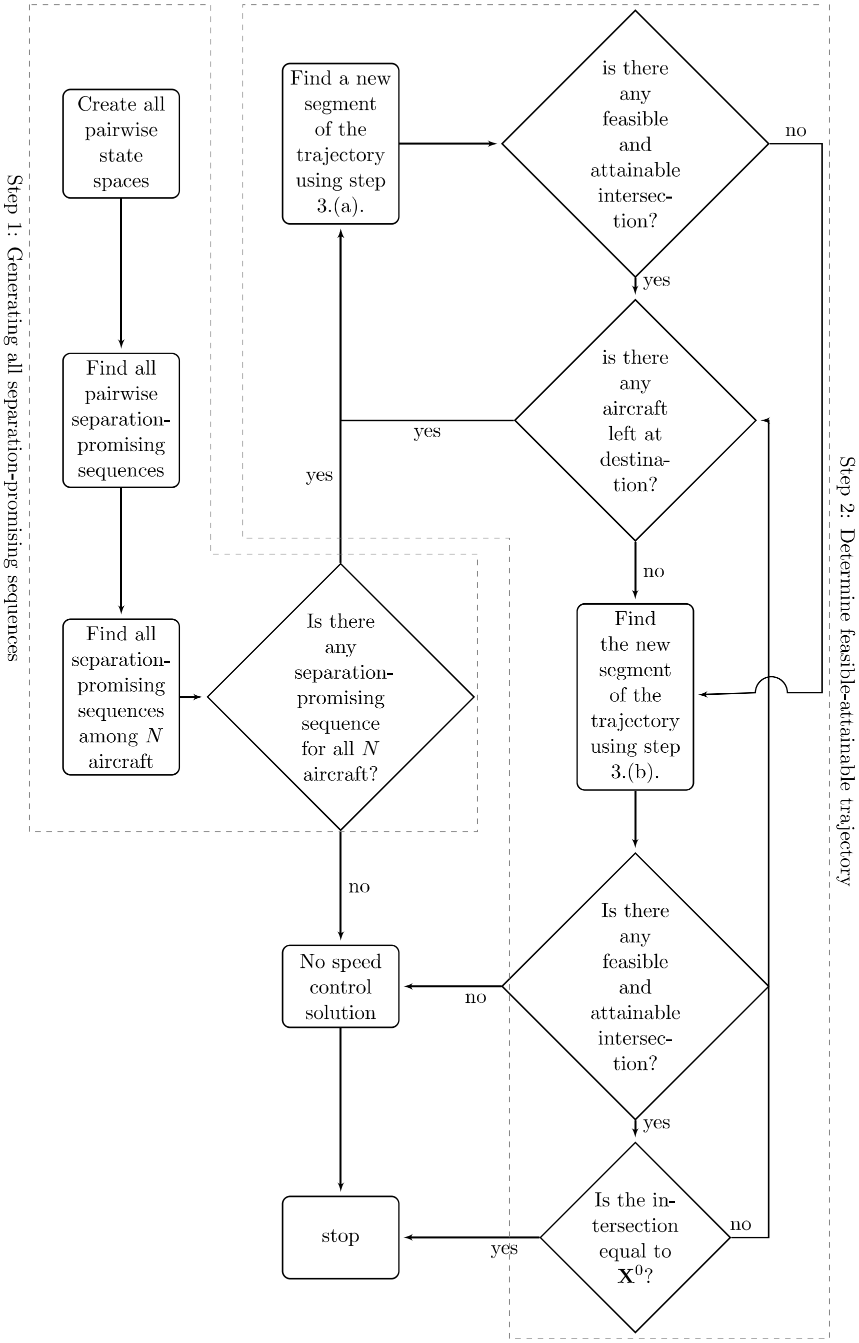

Step 1: Generating all separation-promising sequences
Create all pairwise state spaces
Find all pairwise separation-promising sequences
Find all separation-promising sequences among $N$ aircraft
Is there any separation-promising sequence for all $N$ aircraft?
no
yes
No speed control solution
stop
Find a new segment of the trajectory using step 3.(a).
is there any feasible and attainable intersection?
no
yes
is there any aircraft left at destination?
yes
no
Find the new segment of the trajectory using step 3.(b).
Is there any feasible and attainable intersection?
no
yes
Is the intersection equal to $\mathbf{X}^0$?
yes
no
Step 2: Determine feasible-attainable trajectory

Figure 3.6: Flow chart of the algorithm.

The algorithm is exhaustive, it checks the existence of a speed control solution in all separation-promising sequences and in each sequence looks for the feasible-attainable trajectory that provides minimum possible spacing between all pairs of aircraft (minimum positive $\lambda$ for $\mu_n^m$). A criterion, i.e. minimum delay with respect to the flight schedule, can be defined to choose the solution among all separation-promising sequences with feasible-attainable trajectory. Therefore, if such a solution does not exist in any separation-promising sequence, then there is no speed control solution.

**Definition 3.2.** *Each separation-promising sequence corresponds to one safe wedge* [2] *which in all 2-dimensional subspaces there exist feasible-attainable trajectory that satisfy the sequence (see Definition 3.1). The safe wedge corresponds to a separation-promising sequence is called* separation-promising safe wedge*.*

**Remark 3.4.** *If there exists an attainable state on the separation-compliant part of the bounding hyperplane of the roof set then from that state one can construct a feasible-attainable trajectory that enters a safe wedge.*

The algorithm first generates all separation-promising sequences among $N$ aircraft. Then, it looks for a feasible-attainable trajectory in separation-promising sequences. Therefore, algorithm finds all separation-promising safe wedges and then it looks for a feasible-attainable trajectory in these safe wedges. Given Remark 3.4, there exist a feasible-attainable trajectory, if there exist at least one attainable state on the intersection between the separation-compliant portion of bounding hyperplanes of the roof set and the bounding hyperplanes of the separation-loss set associated with a separation-promising safe wedge,

$$\left(\mathcal{X}^{urc} \cup \mathcal{X}^{lrc}\right) \cap \mathcal{A}(\mathbf{X}^0) \neq \emptyset,$$

where

$$\mathcal{X}^{urc} = \partial \left( \bigcup_{\substack{i,j=1 \\ i<j}}^{N} \mathcal{U}_{ij}^{c} \right) \cap \partial \left( \bigcup_{\substack{i,j=1 \\ i<j}}^{N} \mathcal{U}_{ij}^{r} \right), \tag{3.6}$$

[2] *In $N$-dimensional state space ($N > 1$ , the intersections between attainable cone and separation-complaint state set are convex polyhedrons called* safe wedges *[24]*

$$\mathcal{X}^{lrc} = \partial \left( \bigcup_{\substack{i,j=1 \\ i<j}}^{N} \mathcal{L}^c_{ij} \right) \cap \partial \left( \bigcup_{\substack{i,j=1 \\ i<j}}^{N} \mathcal{L}^r_{ij} \right), \tag{3.7}$$

and $\mathcal{U}^c_{ij}$, $\mathcal{L}^c_{ij}$, $\mathcal{U}^r_{ij}$ and $\mathcal{L}^r_{ij}$ are half-planes defined in equations (2.16)-(2.19) and $\mathcal{U}^c_{ij}$ or $\mathcal{U}^r_{ij}$ is associated with a separation-promising safe wedge. The faces of the separation-loss set are the set of the innermost feasible states including states in $\mathcal{X}^{urc}$ and $\mathcal{X}^{lrc}$ (see equations (3.6) and (3.7)). The algorithm generates a feasible-attainable trajectory which lies on this faces associated with a separation-promising safe wedge (Definition 3.2) by choosing the smallest positive $\lambda$ for the value of $\mu^m_n$. Therefore, if there exits at least an attainable state in $\mathcal{X}^{urc}$ or $\mathcal{X}^{lrc}$ associated with a separation-promising safe wedge then the algorithm can generate a feasible-attainable trajectory. Hence, the algorithm guarantees to find the feasible-attainable trajectory in a separation-promising sequence if such a trajectory exists.

## 3.2 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 3.7. 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 3.8-3.12 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 $C^{++}$.

## 3.3 Multiple solution case

In segment $\mathbf{Y}^m_n\mathbf{Y}^{m+1}_n$, the director $\mathbf{d}^m_n$ can be calculated from the following equation

$$\mathbf{V}^m_n \mathbf{d}^m_n = 0, \tag{3.8}$$

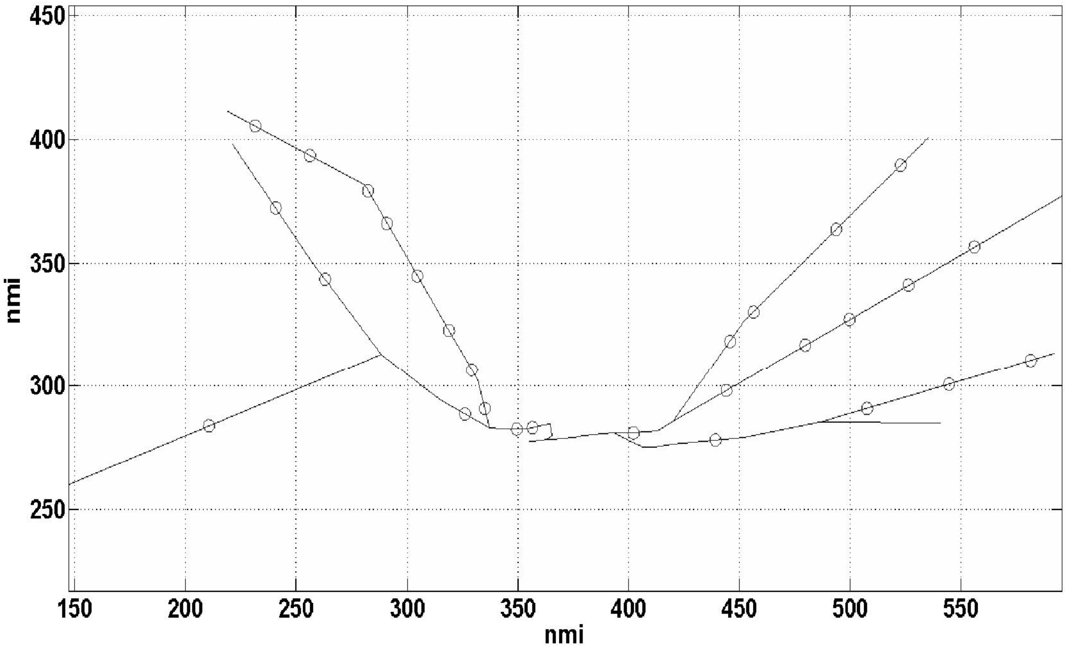


Figure 3.7: Initial positions of aircraft in LAX airspace.

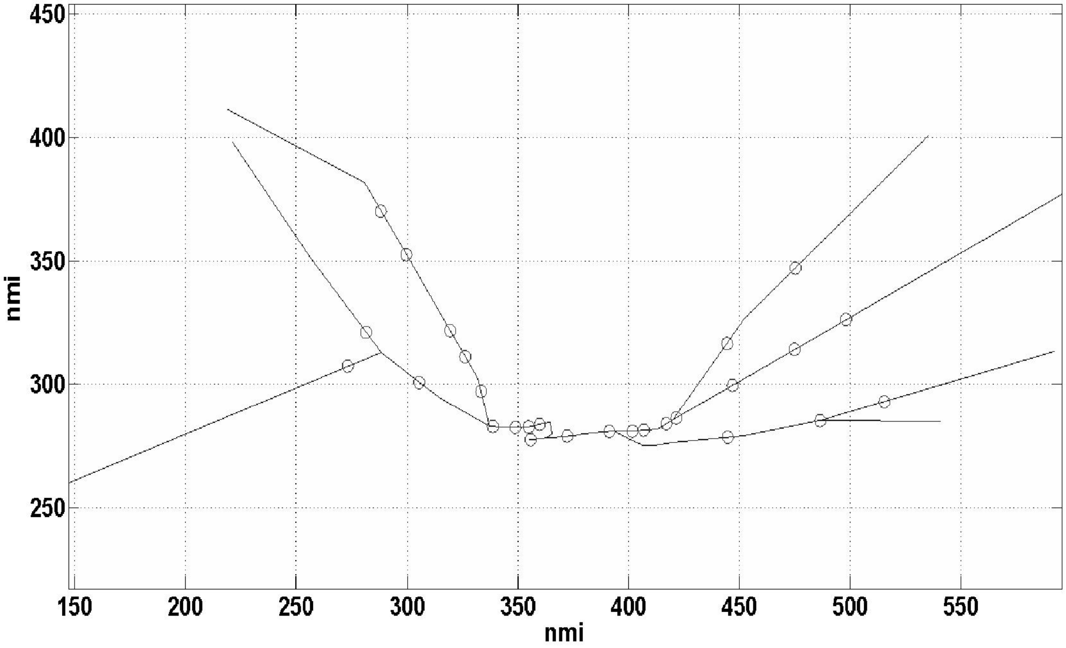


Figure 3.8: Positions of aircraft in LAX airspace at t=30.5 s.

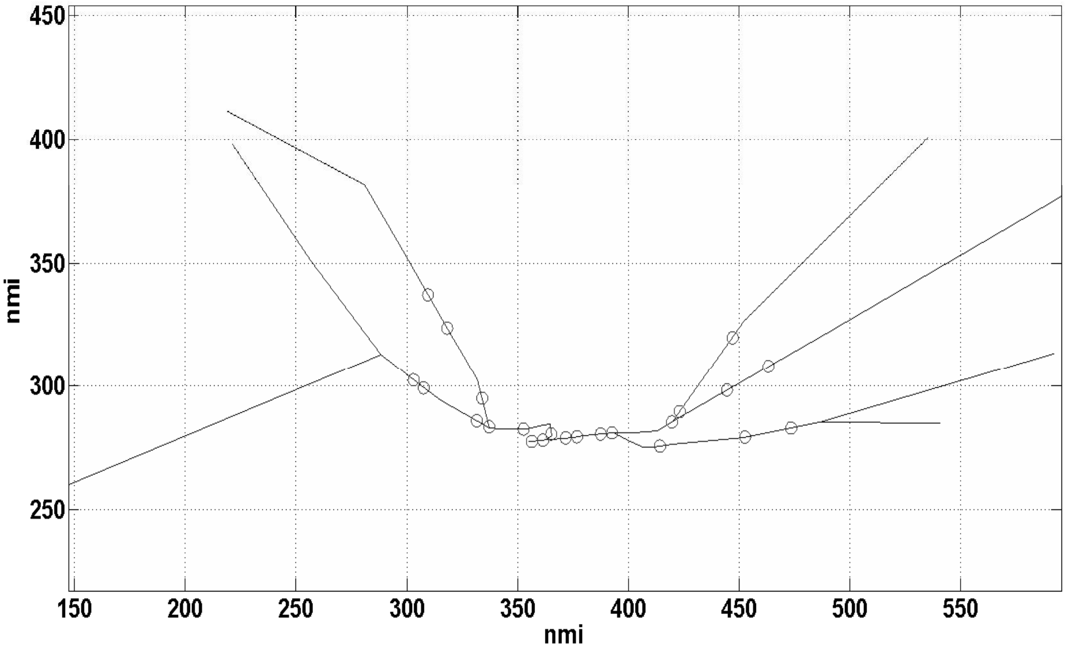


Figure 3.9: Positions of aircraft in LAX airspace at t=61 s.

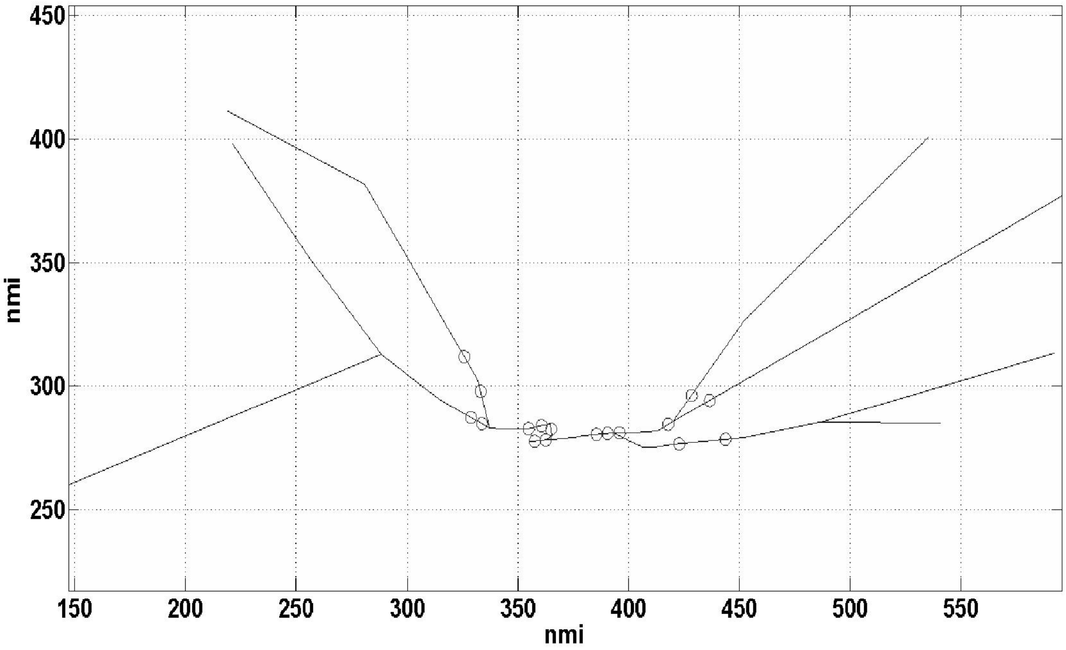


Figure 3.10: Positions of aircraft in LAX airspace at t=91.5 s.

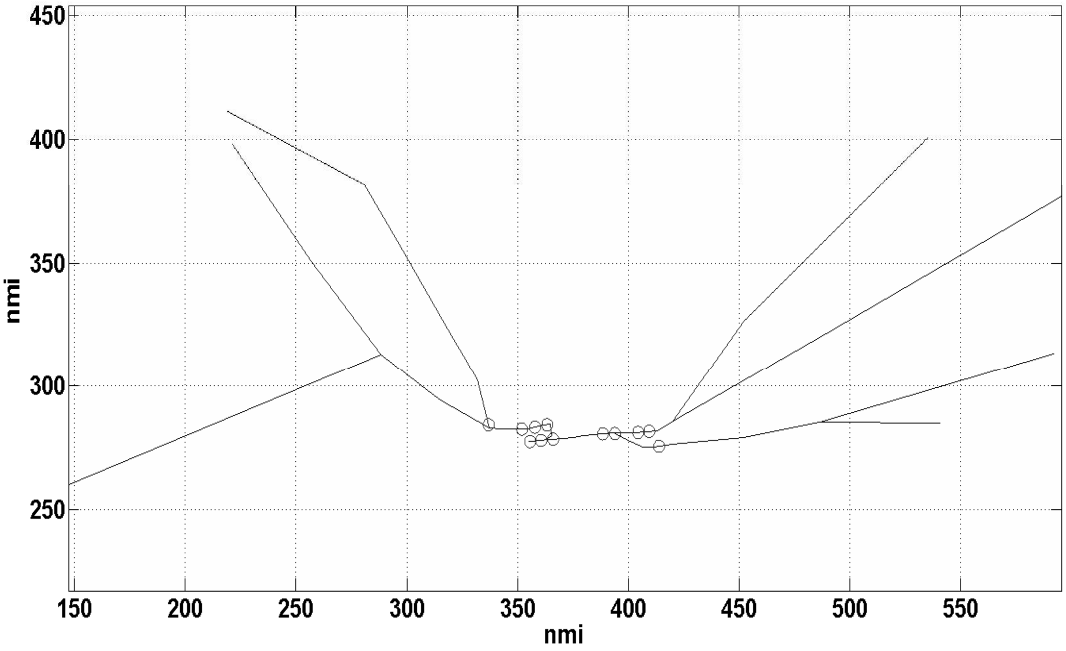


Figure 3.11: Positions of aircraft in LAX airspace at t=122 s.

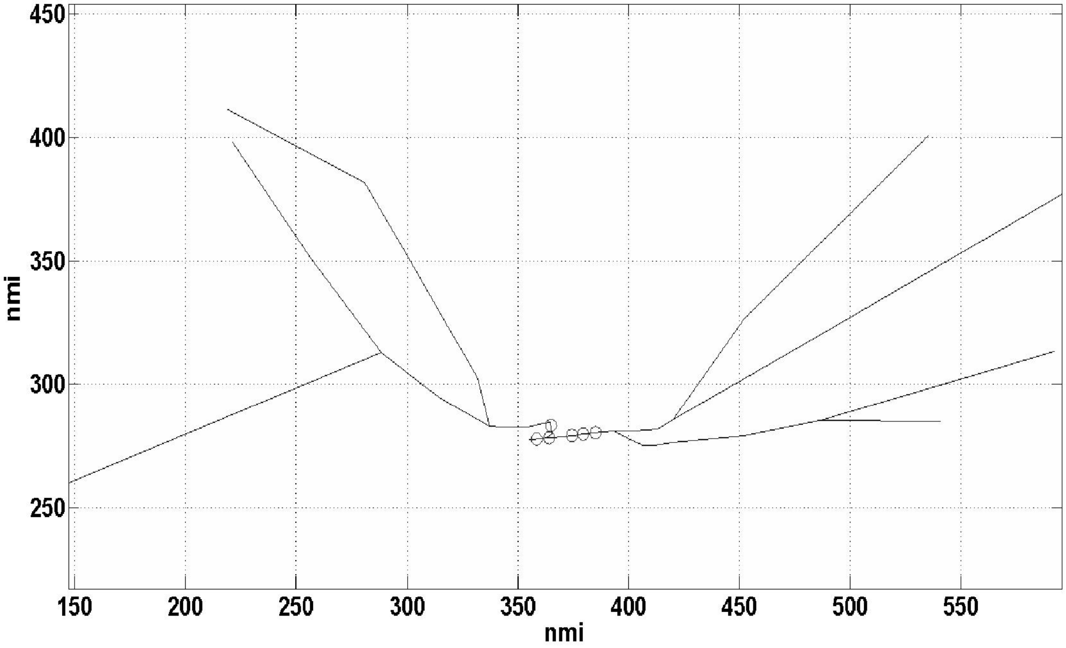


Figure 3.12: Positions of aircraft in LAX airspace at t=152.5 s.

where $\mathbf{V}_n^m$ is an $(N-1)\times N$ matrix where rows of $\mathbf{V}_n^m$ are normal vectors of guiding hyperplanes in $\mathbb{H}_n^m$. If there are $(N-1)$ independent guiding hyperplane then the null space of matrix $\mathbf{V}_n^m$ has dimension one. therefore, director $\mathbf{d}_n^m$ is unique. If there exist any inactive guiding hyperplanes in $\mathbb{H}_n^m$ and one removes them from the set of guiding hyperplanes, then the dimension of the null space of matrix $\mathbf{V}_n^m$ is more one. Hence, director $\mathbf{d}_n^m$ is not unique.

Assume the dimension of null space of $\mathbf{V}_n^m$ is $M > 1$. Then, the null space of $\mathbf{V}_n^m$ is spanned by $M$ independent unit vectors,

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

where

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

The vector $\mathbf{d}_n^m$ in equation (3.8) can be expressed as,

$$\begin{aligned}\mathbf{d}_n^m &= \begin{bmatrix}d_1 & d_2 & \cdots & d_N\end{bmatrix}^T = \alpha_1\mathbf{e}_1 + \alpha_2\mathbf{e}_2 + \cdots + \alpha_M\mathbf{e}_M,\\ d_i &= \alpha_1 e_{i1} + \alpha_2 e_{i2} + \cdots + \alpha_M e_{iM}, \quad i = 1,\ldots,N, \end{aligned} \tag{3.9}$$

where $\alpha_k$ $(k = 1,\cdots,M)$ are scalars and not all zero. The vector $\mathbf{d}_n^m$ define the direction and slope of the trajectory, where the components of $\mathbf{d}_n^m$ must satisfy the slope condition in each pairwise position state space,

$$\underline{s}_{ij} \leqslant \frac{d_j}{d_i} \leqslant \overline{s}_{ij}, \quad i,j = 1,\cdots,N \quad \text{and} \quad i<j.$$

to obtain a general form for $\alpha$'s, substitute for $d_i$ and $d_j$ from equation (3.9), then

$$\alpha_1\left(e_{j1} - \overline{s}_{ij}e_{i1}\right) + \alpha_2\left(e_{j2} - \overline{s}_{ij}e_{i2}\right) + \cdots + \alpha_M\left(e_{jM} - \overline{s}_{ij}e_{iM}\right) \leqslant 0, \tag{3.10}$$

$$\alpha_1\left(\underline{s}_{ij}e_{i1} - e_{j1}\right) + \alpha_2\left(\underline{s}_{ij}e_{i2} - e_{j2}\right) + \cdots + \alpha_M\left(\underline{s}_{ij}e_{iM} - e_{jM}\right) \leqslant 0. \tag{3.11}$$

For each pairwise position state space the above two inequalities must be satisfied. Assume there are $p$ aircraft that have not reached their destination. Therefore, there are $p(p-1)$ inequalities which can be represented in matrix form.

$$\mathbf{F}\alpha \leqslant 0, \tag{3.12}$$

where,

$$\mathbf{F} = \begin{bmatrix} e_{21} - \overline{s}_{12}e_{11} & e_{22} - \overline{s}_{12}e_{12} & \cdots & e_{2M} - \overline{s}_{12}e_{1M} \\ \underline{s}_{12}e_{11} - e_{21} & \underline{s}_{12}e_{12} - e_{22} & \cdots & \underline{s}_{12}e_{1M} - e_{2M} \\ \vdots & \vdots & & \vdots \\ e_{N1} - \overline{s}_{(N-1)N}e_{(N-1)1} & e_{N2} - \overline{s}_{(N-1)N}e_{(N-1)2} & \cdots & e_{NM} - \overline{s}_{(N-1)N}e_{(N-1)M} \\ \underline{s}_{(N-1)N}e_{(N-1)1} - e_{N1} & \underline{s}_{(N-1)N}e_{(N-1)2} - e_{N2} & \cdots & \underline{s}_{(N-1)N}e_{(N-1)M} - e_{NM} \end{bmatrix},$$

$$\alpha = \begin{bmatrix} \alpha_1 & \alpha_2 & \cdots & \alpha_M \end{bmatrix}^T$$

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.

# CHAPTER 4

# Extension of the algorithm to path control

The algorithm developed in Chapter 3 guarantees a speed control solution for a given fleet of aircraft in a given airspace whenever such a solution exists at all. The modeling framework in which this algorithm is set also indicates the aircraft that need path modification. Furthermore, this model enables one to define a new path for the aircraft and then to use the speed control solution to define the speed profile for each aircraft. This ability of the algorithm will be discussed and used in this chapter.

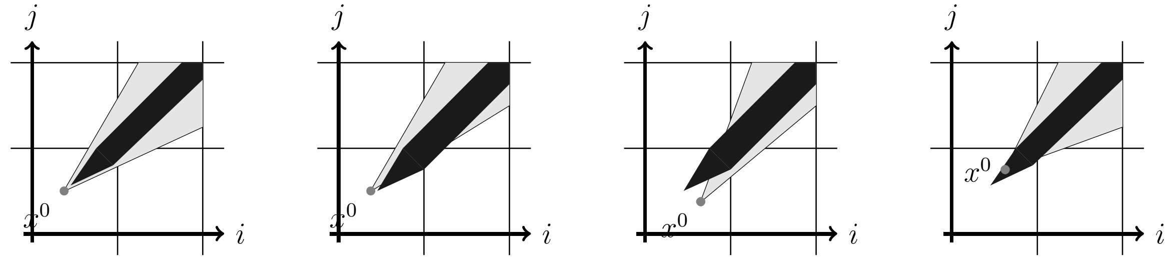
In the present chapter, the speed control algorithm developed in Chapter 3 is extended to address cases in which the separation-compliance can not be achieved by speed control alone and, consequently, in which path control is required. From the Lemma 3.2, the necessary and sufficient conditions of existence of a speed control solution imply that there is a separation-promising sequence among all $N$ aircraft and a feasible-attainable trajectory in the separation-promising sequence. Thus, if one or both of the necessary and sufficient conditions is violated, then new paths must be sought for some (or, possibly, all) of the aircraft. The new path of an aircraft can be shorter or longer than the original path. If the assigned path between two waypoints is the shortest possible path (straight line), then a shorter path cannot be generated for path control, but it is always possible, in all situations, to find a longer path. Therefore, the path control algorithm presented here is designed to modify a path only toward elongation, never toward shortening.

The effect of path modification in the modeling framework is analyzed in Section 2.6. As mentioned above, according to the lemma 3.2, there are two situations where the speed control does not exist. If the necessary condition for speed control is not met, the path control for a particular aircraft will be developed. This case is discussed in Section 4.1. In

Section 4.2, path control when the sufficient condition for speed control is not satisfied will be constructed for a particular set of aircraft. In Section 4.3, the approximate shape of the new path will be discussed.

## 4.1 Failure of the necessary condition: no separation-promising sequence

The algorithm in Chapter 3 (also presented in [20, 21]) begins by constructing all separation-promising sequence for all $N$ aircraft by looking at all pairwise separation-promising sequences of all pairwise position state spaces. In each pairwise position state space, there are two possible sequences; *i*) aircraft $j$ exits the airspace before aircraft $i$ which is denoted by $\{i, j\}$, and *ii*) aircraft $i$ exits the airspace before aircraft $j$, $\{j, i\}$. All possible scenarios in a 2-dimensional subspace are depicted in Figure 4.1 where the attainable cone intersects the infeasible set.



(a) Both possible sequences are separation-promising

(b) Just sequence $\{i, j\}$ is separation-promising

(c) Just sequence $\{j, i\}$ is separation-promising

(d) No sequence is separation-promising

Figure 4.1: All possible scenarios in the pairwise position state space of aircraft $i$ and $j$.

In order to discuss the method of finding the set of aircraft with minimum path modifications, it is necessary to understand the algorithm given in Chapter 3 of generating the separation-promising sequence. The algorithm generates the sequence by first finding the last aircraft that reaches its destination and continues until finding the first aircraft that reaches its destination. Consider constructing a $N$-by-$N$ matrix $\mathbf{S}$, where $N$ is the number of aircraft in the airspace. In each 2-dimensional subspace pick the aircraft that reaches its destination last. Assign the index of this aircraft to the $(i, j)$ and $(j, i)$ components of the matrix $\mathbf{S}$. If both sequences are separation-promising sequence, Figure 4.1a, then assign zero and if no sequence is separation-promising sequence, Figure 4.1d, assign $N + 1$ to the $(i, j)$ and $(j, i)$ components. Therefore, matrix $\mathbf{S}$ is a symmetric matrix with zeros on diagonal components where $i^{th}$ column (row) of this matrix represent the last aircraft that are reaching their destination in all pairwise position state spaces associating with aircraft $i$.

The assumption in the first step is that all aircraft have reached their destinations. Then, if the $i^{th}$ column (row) just contains 0 or $i$ then one obtains aircraft $i$ satisfies all pairwise separation-promising sequences and can be the last aircraft in the sequence of $N$ aircraft to reach its destination (or first aircraft in the time-reverse to enter the airspace). Now, the aircraft $i$ is considered to be inside the airspace and all other aircraft are at their destinations. Hence, all components of $i^{th}$ row and column must turn to zero. Also, the $i^{th}$ column (row) can not be picked again in proceeding steps. Now, consider an $N$-dimensional vector $\mathbf{O}$, where the component $o_i$ is one if aircraft $i$ has reached its destination and is zero if aircraft $i$ is inside the airspace and consider set $I$ which contains all the indices of aircraft that have reached their destinations. In order to switch the rows and columns associated with aircraft inside the airspace to zero, the matrix $\mathbf{S}$ can be post multiplied by $diag(\mathbf{O})$. Where $diag(.)$ is called the *diag* function [8]. Now, aircraft $j$ is the next aircraft if the $j^{th}$ column (row) of matrix $diag(\mathbf{O}) * \mathbf{S}$ contains 0 or $j$. This can be simplified by using the following equation

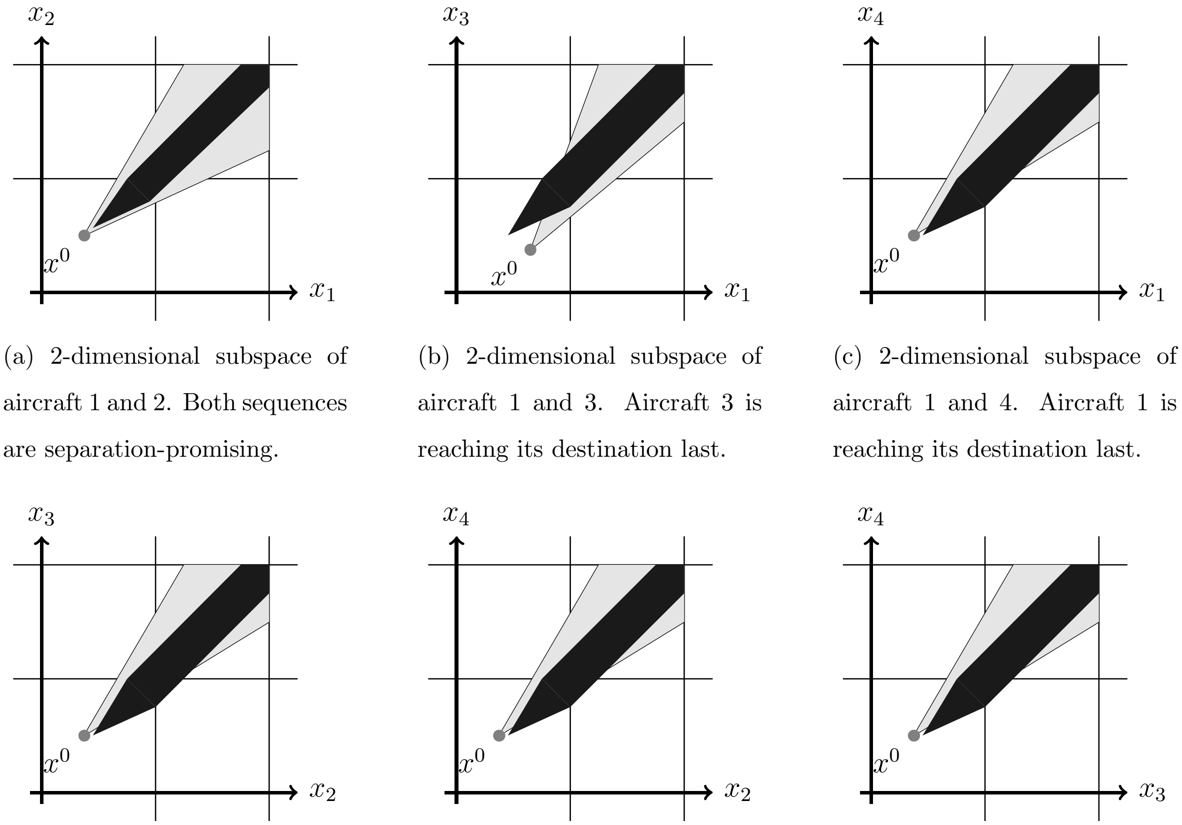
$$\mathbf{s'}_j = diag(\mathbf{O}) * \mathbf{s}_j - j * sign\left(diag(\mathbf{O}) * \mathbf{s}_j\right), \tag{4.1}$$

where

$$sign(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x = 0 \\ -1 & \text{if } x < 0 \end{cases}$$

and $\mathbf{s}_j$ is the $j^{th}$ column of matrix $\mathbf{S}$ and $j \in I$. If the vector $\mathbf{s'}_j = \mathbf{0}$, then aircraft $j$ is the next aircraft in the sequence. After finding each aircraft in the sequence, the vector $\mathbf{O}$ and the set $I$ must be updated by setting $o_j = 0$ and removing $j$ from the set $I$. If there are multiple aircraft with $\mathbf{s}'_j = 0$, then there is more than one separation-promising sequence.

In order to make the above procedure clear, an example of four aircraft is considered. There are six pairwise position state spaces for 4 aircraft. The pairwise position state spaces of this example are depicted in Figure 4.2. Therefore, the matrix $\mathbf{S}$ for this example is as



(a) 2-dimensional subspace of aircraft 1 and 2. Both sequences are separation-promising.

(b) 2-dimensional subspace of aircraft 1 and 3. Aircraft 3 is reaching its destination last.

(c) 2-dimensional subspace of aircraft 1 and 4. Aircraft 1 is reaching its destination last.

(d) 2-dimensional subspace of aircraft 2 and 3. Aircraft 2 is reaching its destination last.

(e) 2-dimensional subspace of aircraft 2 and 4. Aircraft 2 is reaching its destination last.

(f) 2-dimensional subspace of aircraft 3 and 4. Aircraft 3 is reaching its destination last.

Figure 4.2: All pairwise position state spaces of 4 aircraft example

follows:

$$\mathbf{S} = \begin{bmatrix} 0 & 0 & 3 & 1 \\ 0 & 0 & 2 & 2 \\ 3 & 2 & 0 & 3 \\ 1 & 2 & 3 & 0 \end{bmatrix}.$$

The algorithm generates the sequence backward. Therefore, at the first step assume all the aircraft have reached their destinations. Hence, the vector $\mathbf{O}$ and the set $I$ are as follows;

$$\mathbf{O}^T = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix},$$
$$I = \{1, 2, 3, 4\}.$$

To find the first aircraft in the separation-promising sequence, use equation (4.1) to calculate $\mathbf{s}'_j$ for $j = 1, 2, 3, 4$. Then

$$\mathbf{s}'_1 = \begin{bmatrix} 0 \\ 0 \\ 2 \\ 0 \end{bmatrix}, \quad \mathbf{s}'_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{s}'_3 = \begin{bmatrix} 0 \\ -1 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{s}'_4 = \begin{bmatrix} -3 \\ -2 \\ -1 \\ 0 \end{bmatrix}.$$

Only $\mathbf{s}'_2 = \mathbf{0}$. Therefore, the first aircraft to reach its destination is the aircraft 2,

$$Seq. = \{2\}. \tag{4.2}$$

Before, using equation (4.1) to find the second aircraft, the vector $\mathbf{O}$ and the set $I$ must be updated as follow

$$\mathbf{O}^T = \begin{bmatrix} 1 & 0 & 1 & 1 \end{bmatrix},$$
$$I = \{1, 3, 4\}.$$

Then, in the next step $j = 1, 3, 4$. Then,

$$\mathbf{s}'_1 = \begin{bmatrix} 0 \\ 0 \\ 2 \\ 0 \end{bmatrix}, \quad \mathbf{s}'_3 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{s}'_4 = \begin{bmatrix} -3 \\ 0 \\ -1 \\ 0 \end{bmatrix}.$$

Since, the next aircraft is the aircraft 3, then the sequence (4.2) is updated to

$$Seq. = \{2, 3\},$$

and the update of the vector $\mathbf{O}$ and the set $I$ are

$$\mathbf{O}^T = \begin{bmatrix} 1 & 0 & 0 & 1 \end{bmatrix},$$

$$I = \{1, 4\}.$$

In the next step $j = 1, 4$. Then,

$$\mathbf{s'}_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{s'}_4 = \begin{bmatrix} -3 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

The third aircraft to reach its destination is the aircraft 1, then

$$Seq. = \{2, 3, 1\}.$$

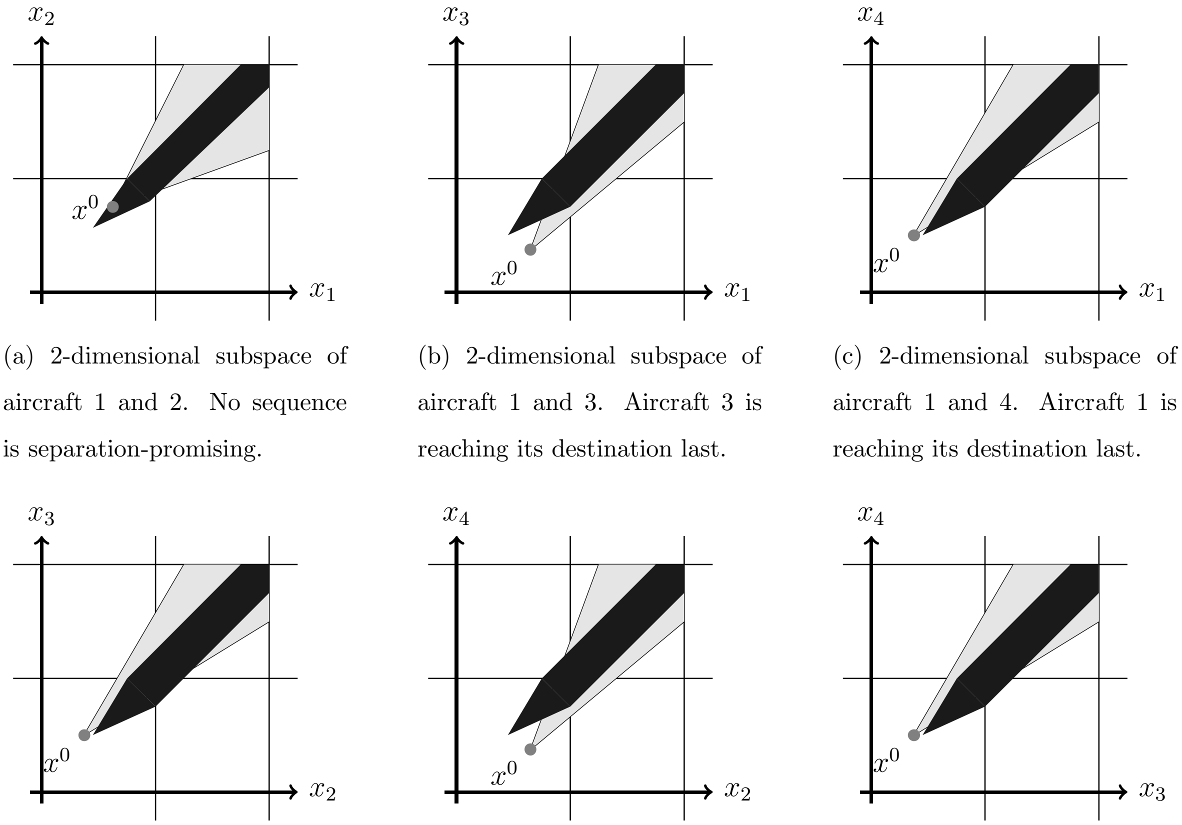
Therefore, the separation-promising sequence for this example is:

$$Seq. = \{2, 3, 1, 4\},$$

where aircraft 2 is the last aircraft that reaches its destination and aircraft 4 is the first one. If in one of the steps of finding the feasible sequence, all $\mathbf{s'}_j$ are nonzero, then there is no separation-promising sequence. In this case, the algorithm must use the path control.

The first step in the path control algorithm is to find the set of aircraft for path modification. The aircraft can be identified by looking at the nonzero components of vector $\mathbf{s}'_j$. The nonzero components are associated with the pairwise position state spaces where aircraft $j$ can not be the last aircraft to reach its destination. Then, in each of these pairwise position state spaces, the minimum path elongation needed for aircraft $j$ to be the last aircraft to reach its destination must be calculated. The maximum along these minimum path elongations is the minimum required path elongation for aircraft $j$ to be the last aircraft in all pairwise position state spaces associated with aircraft $j$. In order to find the aircraft with minimum required path modification, the path modification of all aircraft $j \in I$ must be calculated and then the one with minimum path modification is the candidate for path control.

To illustrate the above discussion, consider a set of four aircraft. The pairwise position state spaces of this example are depicted in Figure 4.3.



(a) 2-dimensional subspace of aircraft 1 and 2. No sequence is separation-promising.

(b) 2-dimensional subspace of aircraft 1 and 3. Aircraft 3 is reaching its destination last.

(c) 2-dimensional subspace of aircraft 1 and 4. Aircraft 1 is reaching its destination last.

(d) 2-dimensional subspace of aircraft 2 and 3. Aircraft 2 is reaching its destination last.

(e) 2-dimensional subspace of aircraft 2 and 4. Aircraft 4 is reaching its destination last.

(f) 2-dimensional subspace of aircraft 3 and 4. Aircraft 3 is reaching its destination last.

Figure 4.3: All pairwise position state spaces in example 2

The matrix $\mathbf{S}$ in this example is

$$\mathbf{S} = \begin{bmatrix} 0 & 5 & 3 & 1 \\ 5 & 0 & 2 & 4 \\ 3 & 2 & 0 & 3 \\ 1 & 4 & 3 & 0 \end{bmatrix}.$$

The vector $\mathbf{O}$ and the set $I$ are as follows;

$$\mathbf{O}^T = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix},$$
$$I = \{1, 2, 3, 4\}.$$

$\mathbf{s'}_j$s in the first step of generating the separation-promising sequence are,

$$\mathbf{s'}_1 = \begin{bmatrix} 0 \\ 4 \\ 2 \\ 0 \end{bmatrix}, \quad \mathbf{s'}_2 = \begin{bmatrix} 3 \\ 0 \\ 0 \\ 2 \end{bmatrix}, \quad \mathbf{s'}_3 = \begin{bmatrix} 0 \\ -1 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{s'}_4 = \begin{bmatrix} -3 \\ 0 \\ -1 \\ 0 \end{bmatrix}.$$

There is no $\mathbf{s'}_j = \mathbf{0}$. Therefore, no separation-promising sequence exists and the algorithm must use the path modification. The path modification associated with each aircraft $j \in I$ must be determined.

- Path modification associated with aircraft 1;

  The second and the third components of $\mathbf{s'}_1$ are nonzero. Therefore, the 2-dimensional subspace of aircraft 1 and aircraft 2 and subspace of aircraft 1 and 3 must be modified such that the aircraft 1 reaches its destination after aircraft 2 and 3. The maximum minimum modification that satisfies both subspaces is the minimum modification of the path of aircraft 1, denoted by $\Delta\ell_1$.

- Path modification of the aircraft 2;

  The first and the forth components of $\mathbf{s'}_2$ are nonzero. Like the previous scenario, the maximum minimum modification that lets aircraft 2 reach its destination after aircraft 1 and 4 is the minimum modification, denoted by $\Delta\ell_2$.

- Path modification of aircraft 3;

  Only the second component of $\mathbf{s}'_3$ is nonzero. Then, the minimum path modification associated with pairwise position state space of aircraft 3 and aircraft 2 is the the minimum modification of the path of aircraft 3 and is denoted by $\Delta\ell_3$.

- Path modification of aircraft 4;

  The minimum modification is equal to maximum minimum modification of the path of aircraft 4 that lets aircraft 4 reach its destination after aircraft 1 and 3, and is denoted by $\Delta\ell_4$.

The minimum modification along $\Delta\ell_1, ..., \Delta\ell_4$ will be picked. The aircraft associated to the minimum modification is the one that is chosen for path modification.

Instead of minimum path modification, one can choose any criteria, i.e. schedule of arrival could be used to pick the aircraft $j \in I$ for path modification. For instance, in the previous example, if according to the schedule, aircraft 4 is the last aircraft to reach its destination, then the algorithm will pick aircraft 4 as the first one in the sequence and find the minimum required path modification for it.

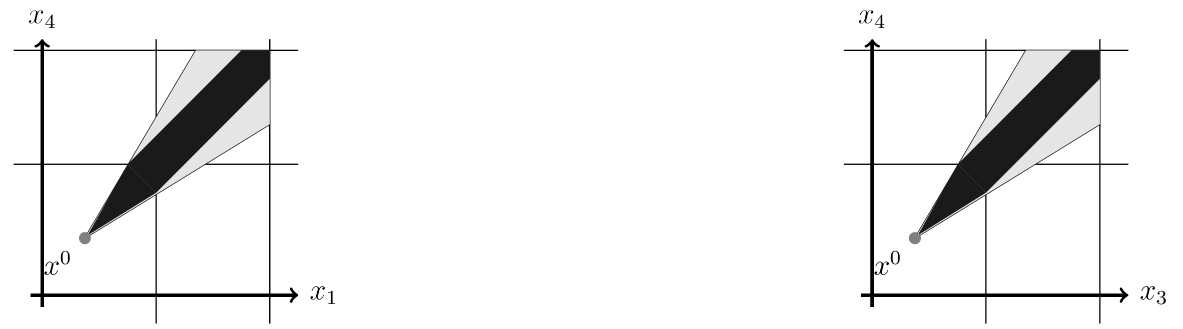
Assume the minimum path modification in the above example is $\Delta\ell_4$. Therefore, aircraft 4 is the first aircraft in the sequence. The modified 2-dimensional subspaces are depicted in Figure 4.4.

$$Seq. = \{4\}.$$

Before, using equation (4.1) to find the second aircraft, the vector $\mathbf{O}$ and the set $I$ must be updated as follow

$$\mathbf{O}^T = \begin{bmatrix} 1 & 1 & 1 & 0 \end{bmatrix},$$

$$I = \{1, 2, 3\}.$$



(a) Modified 2-dimensional subspace of aircraft 1 and 4. Aircraft 4 is leaving last.

(b) Modified 2-dimensional subspace of aircraft 3 and 4. Aircraft 4 is leaving last.

Figure 4.4: Modified 2-dimensional subspaces in second example, the boundaries of separation-loss set and attainable cone are coincidence.

Then, in the next step $j = 1, 2, 3$. Hence,

$$\mathbf{s'}_1 = \begin{bmatrix} 0 \\ 4 \\ 2 \\ 0 \end{bmatrix}, \quad \mathbf{s'}_2 = \begin{bmatrix} 3 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{s'}_3 = \begin{bmatrix} 0 \\ -1 \\ 0 \\ 0 \end{bmatrix}.$$

There is no $\mathbf{s'}_j = \mathbf{0}$. Then, a second aircraft is chosen for path modification. Then, path modifications $\Delta\ell_1$, $\Delta\ell_2$ and $\Delta\ell_3$ associated with aircraft 1,2 and 3, respectively must be calculated. Aircraft associated with the minimum path modification will be chosen for path modification, i.e. this is found to be $\Delta\ell_2$. Therefore, the 2-dimensional subspace of aircraft 1 and 2 must be modified such that aircraft 2 leaves the airspace after aircraft 1. The modified subspace is depicted in Figure 4.5.

The next aircraft is aircraft 2,

$$Seq. = \{4, 2\},$$

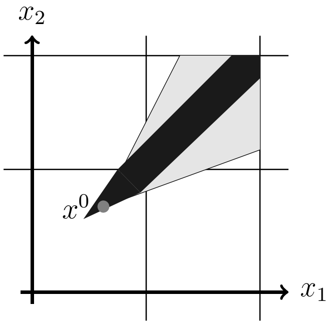


Figure 4.5: Modified 2-dimensional subspace of aircraft 1 and 2, the boundaries of separation-loss set and attainable set are coincidence.

and the update of the vector $\mathbf{O}$ and the set $I$ are

$$\mathbf{O}^T = \begin{bmatrix} 1 & 0 & 1 & 0 \end{bmatrix},$$

$$I = \{1, 3\}.$$

In the next step $j = 1, 3$, then

$$\mathbf{s'}_1 = \begin{bmatrix} 0 \\ 0 \\ 2 \\ 0 \end{bmatrix}, \quad \mathbf{s'}_3 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

$\mathbf{s'}_3 = 0$, then no path modification is needed and the next aircraft is aircraft 3,

$$Seq. = \{4, 2, 3\}.$$

Therefore, the separation-promising sequence for this example is:

$$Seq. = \{4, 2, 3, 1\},$$

where aircraft 4 is the last aircraft that reaches its destination and aircraft 1 is the first one. To make this sequence separation-promising sequence, the paths of aircraft 2 and 4 must be modified by $\Delta\ell_2$ and $\Delta\ell_4$, respectively.

## 4.2 Failure of the sufficiency condition:no trajectory exists in a separation-promising sequence

After generating a separation-promising sequence, renumber all aircraft based on the sequence. Then, the algorithm in Chapter 3 will generate a polygonal trajectory, if one exists. Each segment $\mathbf{Y}_n^m \mathbf{Y}_n^{m+1}$ of the trajectory can be described by the line equation,

$$\mathbf{Y}_n^m - \lambda \mathbf{d}_n^m, \tag{4.3}$$

where $\mathbf{Y}_n^m$ is the last point of the previous segment, the $\mathbf{d}_n^m$ is the director of current segment and $\lambda > 0$ is a scalar variable (See Section 3.1.2). The director $\mathbf{d}_n^m$ is the vector obtains by intersecting all $N-1$ guiding hyperplanes in $\mathbb{H}_n^m$. There are $p$ aircraft inside the airspace (see step 3.(a) and step 3.(b) in Section 3.1.2). The line equation (4.3) intersects all the hyperplanes in $\frac{p(p-1)}{2}$ subspaces and the current segment ends at the closet feasible-attainable intersection, denoted by $\mathbf{Y}^*$ hereafter. Therefore, the feasible-attainable trajectory does not exist if no director can be found or no feasible-attainable intersection between equation (4.3) and hyperplanes in $\frac{p(p-1)}{2}$ subspaces exists. In Section 4.2.1, path modification is considered for the case where no director exists. In Section 4.2.2, the case where no feasible-attainable intersection exists is considered.

### 4.2.1 No director exists

In the case, where no new aircraft has reached its destination (Step 3.(b) in Section 3.1.2). One of the existing inactive or temporary guiding hyperplanes must be replaced with the new guiding hyperplane. If no inactive or temporary guiding hyperplane exists, i.e. all existing guiding hyperplanes are permanent guiding hyperplanes, then by adding the new guiding hyperplane, the number of guiding hyperplane increases to $N$ and the null space of matrix $\mathbf{V}_n^m$ become empty. Therefore, no director exists. Depends on the new guiding hyperplane, there are two solutions.

- If the new guiding hyperplane is temporary guiding hyperplane;

  The pairwise position state space associated with the new guiding hyperplane must be modified such that the intersection point $\mathbf{Y}^*$ moves to the point where the new guiding hyperplane become inactive.

- If the new guiding hyperplane is permanent guiding hyperplane;
  - If all aircraft are inside the airspace;
    The pairwise position state space associated with the new guiding hyperplane must be modified such that the intersection point move to initial point $\mathbf{X}^0$.
  - If there exist aircraft that has reached its destination;
    No path control solution can be found.

### 4.2.2 No feasible-attainable intersection exists

The intersection $\mathbf{Y}^*$ is feasible and attainable if,

$$\mathbf{Y}^* \in \mathcal{A} \cap \mathcal{C}'$$

The new segment (equation (4.3)) starts from $\mathbf{Y}_n^m$ ($\lambda$=0) and moves along $-\mathbf{d}_n^m$ as scalar $\lambda$ increases and intersect with hyperplanes in all 2-dimensional subspaces. If $\mathbf{Y}_n^m$ lies in the interior of $\mathcal{A} \cap \mathcal{C}'$, then there exists at least one intersection with the boundary of the feasible set. So if $\mathbf{Y}_n^m$ lies in the interior of feasible set this condition can not be violated. If $\mathbf{Y}_n^m$ lies on the boundary of the feasible set, then it is possible to have all the intersections, $\mathbf{Y}^*$ outside the feasible set.

$$\mathbf{Y}^* \notin \mathcal{A} \cap \mathcal{C}'$$

By using the De morgan's law, this can happen in one of the following cases,

- $\mathbf{Y}^* \notin \mathcal{A}$ and $\mathbf{Y}^* \in \mathcal{C}'$,
- $\mathbf{Y}^* \in \mathcal{A}$ and $\mathbf{Y}^* \notin \mathcal{C}'$,
- $\mathbf{Y}^* \notin \mathcal{A}$ and $\mathbf{Y}^* \notin \mathcal{C}'$.

#### 4.2.2.1 $\mathbf{Y}^* \notin \mathcal{A}$ and $\mathbf{Y}^* \in \mathcal{C}'$

In this case, the state space must be modified such that the new intersection lies inside the attainable cone. In order to understand the method of modifying the attainable cone, a 2-dimensional subspace will be considered. A sample situation is depicted in Figure 4.6. The intersection $\mathbf{Y}^*$ is not inside the attainable cone. The path control must be used to modify the attainable cone. The algorithm picks the aircraft that extends the path beyond the original path by modifying the attainable cone. i.e. aircraft $n$ in Figure 4.6. In $N$-dimensional state space, in order to modify the attainable cone by just elongating the path of some aircraft, the initial point $\mathbf{X}^0$ must move to $\mathbf{X}^{0'}$ such that

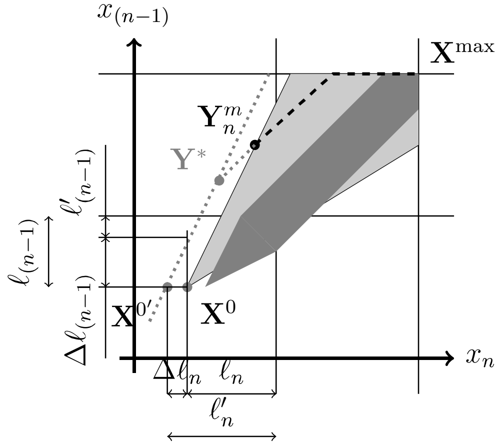


Figure 4.6: 2-dimensional subspace example of $\mathbf{Y}^* \notin \mathcal{A}$ and $\mathbf{Y}^* \in \mathcal{C}'$.

$$\Delta\mathbf{X}^0 = \mathbf{X}^{0'} - \mathbf{X}^0 \le 0.$$

The original $N$-dimensional attainable cone was defined with an inequality

$$\mathcal{A} = \{\mathbf{X} | \mathbf{A}\mathbf{X} \le \mathbf{B}\},$$

when $\mathbf{X} = \mathbf{X}^0$,

$$\mathbf{A}\mathbf{X}^0 = \mathbf{B}.$$

Similarly, the modified attainable cone can be defined as

$$\mathcal{A}' = \{\mathbf{X}|\mathbf{AX} \leq \mathbf{B}'\},$$
$$\mathbf{AX}^{0'} = \mathbf{B}'.$$

Note that, the matrix $\mathbf{A}$ just contains the normal vectors of boundaries of the attainable cone. Therefore, the matrix $\mathbf{A}$ does not change in the modified attainable cone.

In order to find the aircraft with minimum path modifications, the norm of vector $\Delta\mathbf{X}^0$ must be minimized. Therefore, the following optimization problem can be defined

$$\begin{aligned} \underset{\mathbf{X}^{0'}}{\text{minimize}} \quad & \|\mathbf{X}^{0'} - \mathbf{X}^0\| \\ \text{subject to} \quad & \mathbf{X}^{0'} \leq \mathbf{X}^0, \\ & \mathbf{AY}^* \leq \mathbf{B}', \end{aligned} \tag{4.4}$$

where $\mathbf{Y}^*$ is the intersection of the current segment of the trajectory with a hyperplane. The optimization problem (4.4) can also be written as

$$\begin{aligned} \underset{\boldsymbol{\Delta}X^{0'}}{\text{minimize}} \quad & (\boldsymbol{\Delta}X^0)^T(\boldsymbol{\Delta}X^0) \\ \text{subject to} \quad & \begin{bmatrix} \mathbf{I}_N \\ -\mathbf{A} \end{bmatrix} \boldsymbol{\Delta}X^0 \leq \begin{bmatrix} 0 \\ -\mathbf{A}\left(\mathbf{Y}^* - \mathbf{X}^0\right) \end{bmatrix}. \end{aligned} \tag{4.5}$$

The solution to the optimization problem, $\boldsymbol{\Delta}X^0$ is the minimum modification of the vertex of the attainability cone. The aircraft corresponding to the nonzero components of $\Delta\mathbf{X}^0$ are the ones that need to move out from their assigned path. The new path length of these aircraft can be calculated from

$$\ell'_n = \ell_n - \Delta x_n^0,$$

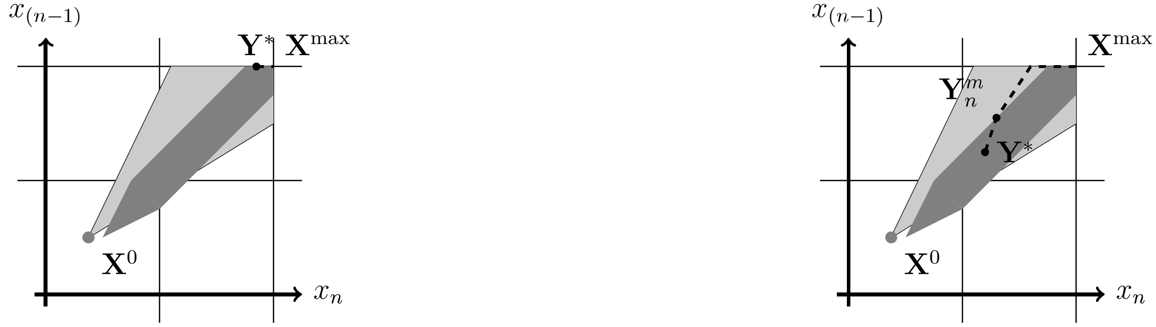
where $\ell'_n$ is the new arc length of the path of aircraft $n$, $\ell_n$ is the arc length of the original assigned path of aircraft $i$ and $\Delta x_n^0$ is the $n^{th}$ component of vector $\Delta\mathbf{X}^0$.

#### 4.2.2.2 $\mathbf{Y}^* \in \mathcal{A}$ and $\mathbf{Y}^* \notin \mathcal{C}'$

This situation can happen in two cases,

*i)* when a new aircraft reaches its destination (Step 3.(a) in Section 3.1.2). This case is depicted in Figure 4.7a. Aircraft $j$ is the new aircraft.

*ii)* when the previous intersection $\mathbf{Y}_n^m$ lies on the boundary of the separation-loss set and this boundary is not in the guiding hyperplane set $\mathbb{H}_n^{m+1}$, of the next segment. This case is depicted in Figure 4.7b.



(a) The intersection $\mathbf{Y}^*$ may lie inside the separation-loss set when aircraft $(n-1)$ has reached its destination.

(b) The intersection $\mathbf{Y}^*$ may lie inside the separation-loss set when the previous intersection $\mathbf{Y}_n^m$ lies on the boundary of the separation-loss set and the boundary is not in the guiding hyperplane set $\mathbb{H}_n^{m+1}$.

Figure 4.7: Two cases when the new intersection lies inside the separation-loss set.

In both cases, the state space must be modified such that the new intersection $\mathbf{Y}^*$ lies inside the separation-compliant state set. The $N$-dimensional separation-compliant state set is defined as

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

where,

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

In order to modify the state space, the algorithm must find the pairwise position state space where the intersection lies inside the separation-loss set. Then, identify the suitable boundary of the pairwise separation-loss set. In the first case, when an aircraft has reached its destination, the intersected boundary of the pairwise separation-loss set associated with the smallest positive $\lambda$ is the suitable boundary. In the second case, when the previous intersection $\mathbf{Y}_n^m$ is on the boundary of the separation-loss set, then this boundary is the suitable boundary. After picking the suitable boundary, the pairwise separation-loss set must be modified such that the last intersection $\mathbf{Y}^*$ lies on the suitable boundary. The modified pairwise separation-loss set in Figure 4.7a is depicted in Figure 4.8. For instance, in this state space the aircraft $n$ needs path modification and the arc length of the new path must be

$$\ell_n' = \ell_n + \Delta\ell_n$$

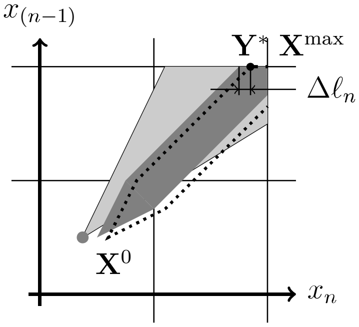
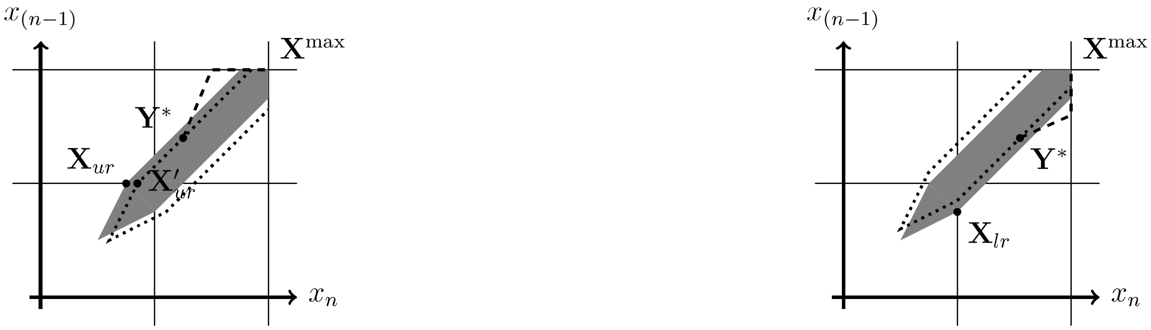


Figure 4.8: Modified pairwise separation-loss set when new aircraft is added to the airspace

In Figure 4.9, the modification of pairwise separation-loss set in both cases when the $\mathbf{Y}_n^m$ is on the hyperplane $\mathcal{H}_{(n-1),n}^{uc}$ or $\mathcal{H}_{(n-1),n}^{lc}$ are depicted. If $\mathbf{Y}_n^m$ is on $\mathcal{H}_{(n-1),n}^{uc}$ (Figure 4.9a), then the path of aircraft $n$ needs to be modified such that the intersection $\mathbf{Y}^*$ be on $\mathcal{H}_{(n-1),n}^{uc}$ . The slope of $\mathcal{H}_{(n-1),n}^{uc}$ is one. Therefore the equation of $\mathcal{H}_{(n-1),n}^{uc}$, after modification is,

$$x_{(n-1)} = x_n - y_n^* + y_{(n-1)}^*, \tag{4.6}$$

where $y_n^*$ and $y_{(n-1)}^*$ are the $n^{th}$ and $(n-1)^{th}$ components of intersection point $\mathbf{Y}^*$, respectively. Since the separation-loss set moves to the right, then the $x_{(n-1)}$ components of all points



(a) When $\mathbf{Y}_n^m$ is on the $\mathcal{H}_{(n-1),n}^{uc}$

(b) When $\mathbf{Y}_n^m$ is on the $\mathcal{H}_{(n-1),n}^{lc}$

Figure 4.9: Modification of separation-loss set when the previous intersection is on one of the boundaries of separation-loss set.

on the boundary stay constant and the $x_n$ components shift for $\Delta\ell_n$. i.e. intersection point between $\mathcal{H}_{(n-1),n}^{uc}$ and $\mathcal{H}_{(n-1),n}^{lc}$, denoted $\mathbf{X}_{ur}$ moves to $\mathbf{X}'_{ur}$, see Figure 4.9a. Then,

$$\Delta\ell_n = x'_{ur_n} - x_{ur_n}, \tag{4.7}$$

$$x'_{ur_{(n-1)}} = x_{ur_{(n-1)}}. \tag{4.8}$$

$\mathbf{X}'_{ur}$ is on the modified infeasible state set boundary, equation (4.6), hence,

$$x'_{ur_{(n-1)}} = x'_{ur_n} - x_n^* + x_{(n-1)}^*$$

using equations (4.7) and (4.8), $\Delta\ell_n$ can be written as

$$\Delta\ell_n = x_{ur_{(n-1)}} + x_n^* - x_{(n-1)}^* - x_{ur_n}.$$

The new arc length of the path of aircraft $n$ is

$$\ell'_n = \ell_n + \Delta\ell_n,$$

If $\mathbf{Y}_n^m$ is on $\mathcal{H}_{(n-1),n}^{lc}$ (Figure 4.9b), then the path of aircraft $(n-1)$ needs to be modified. The new arc length of the path of aircraft $(n-1)$ is

$$\ell'_{(n-1)} = \ell_{(n-1)} + \Delta\ell_{(n-1)}$$

and similarly,

$$\Delta\ell_{(n-1)} = x_{ur_n} - x_n^* + x_{(n-1)}^* - x_{ur_{(n-1)}}.$$

#### 4.2.2.3 $\mathbf{Y}^* \notin \mathcal{A}$ and $\mathbf{Y}^* \notin \mathcal{C}'$

Based on the argument at the beginning of Section 4.2.2, this case can happen if the beginning point of the current segment, $\mathbf{Y}_n^m$ is at the intersection of the boundaries of the attainable cone and the separation-loss set. This intersection happens in a non-separation-promising sequence. The configuration is shown in Figure 4.10. The algorithm first generate separation-promising sequences and then the feasible-attainable trajectory generates based on a separation-promising sequence. Therefore, this case never can happen.

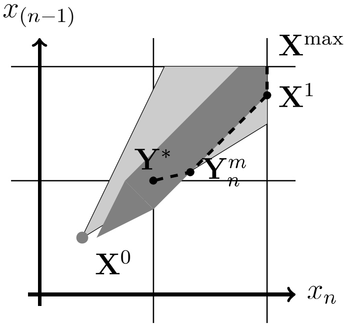
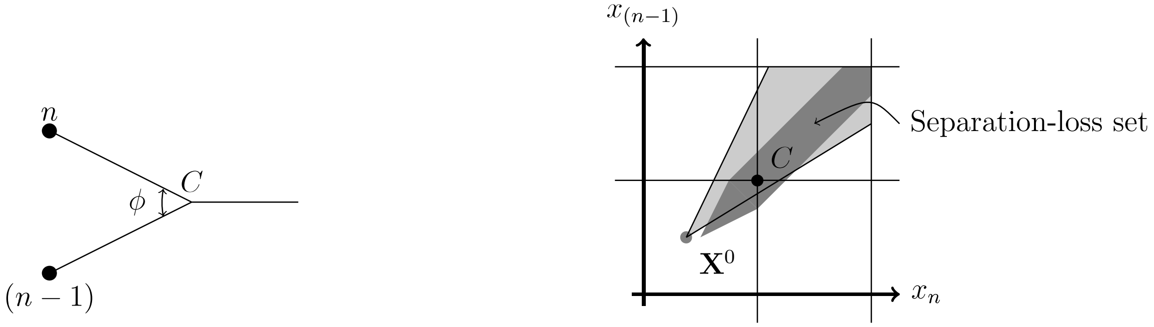


Figure 4.10: The intersection of boundaries of the attainable cone and separation-loss set happens in the non-separation-promising sequence

## 4.3 Path modification

The algorithm uses triangular path as the new path. The new path will add some new waypoints to the airspace which hereafter are called *auxiliary waypoints*. The aircraft leaves the nominal path and comes back to it before the merge point. The size of separation-loss set depends on the angle between two merging path segments. This angle, depicted in Figure 4.11a, is denoted by $\phi$. In order to keep the angle $\phi$ constant, the modified path must come back to the original path at or before the point where the separation-loss set starts. This is depicted in Figure 4.12. Therefore, three auxiliary waypoints will be added per path modification, i.e. in Figure 4.12, these auxiliary waypoints are denoted by $a$, $b$ and $c$.

In order to avoid a new violation of separation between the aircraft with the modified



(a) Merging of two path segments.

(b) pairwise position state space between aircraft $n$ and $(n-1)$

Figure 4.11: Merging of two path segments and the corresponding 2-dimensional subspace between aircraft $n$ and $(n-1)$.

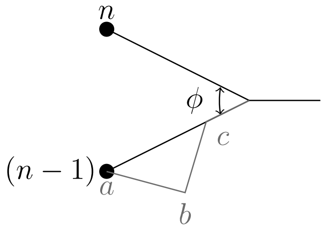
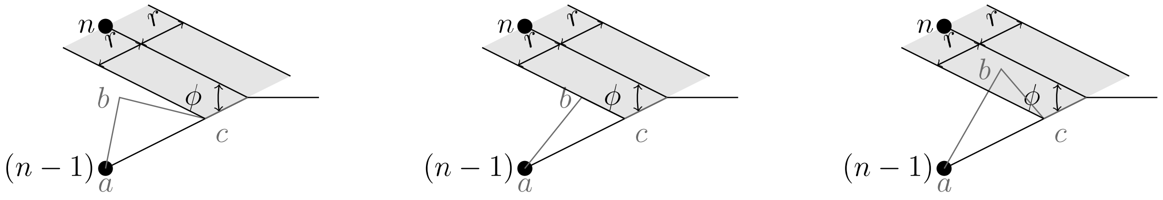


Figure 4.12: The modified path must back to the original path at or before the point where the separation-loss set starts.

path and other aircraft in the airspace and simplify the constraints, two boundaries will be considered around each route segment of the original paths. The distance between original route segment and each boundary is equal to the minimum separation requirement, $r$. In order to avoid a new violation, the auxiliary waypoint $b$ must lie outside of the boundaries of all route segments. This is depicted in Figure 4.13. In Figure 4.13a the distance between waypoint $b$ and the closest route segment is larger than the minimum separation required, also in Figure 4.13b the distance between waypoint $b$ and the closet route segment is equal to minimum separation required. Therefore, both path modifications are acceptable and satisfy the minimum separation requirement. In Figure 4.13c, the waypoint $b$ is in inside the safety boundary of closet route segment and the distance between waypoint $b$ and the route segment is less than the minimum separation requirement. In this case, there is a potential separation loss between two aircraft and algorithm does not accept this solution as a safe path modification.



(a) Safe path modification.

(b) Safe path modification.

(c) Potential separation violation.

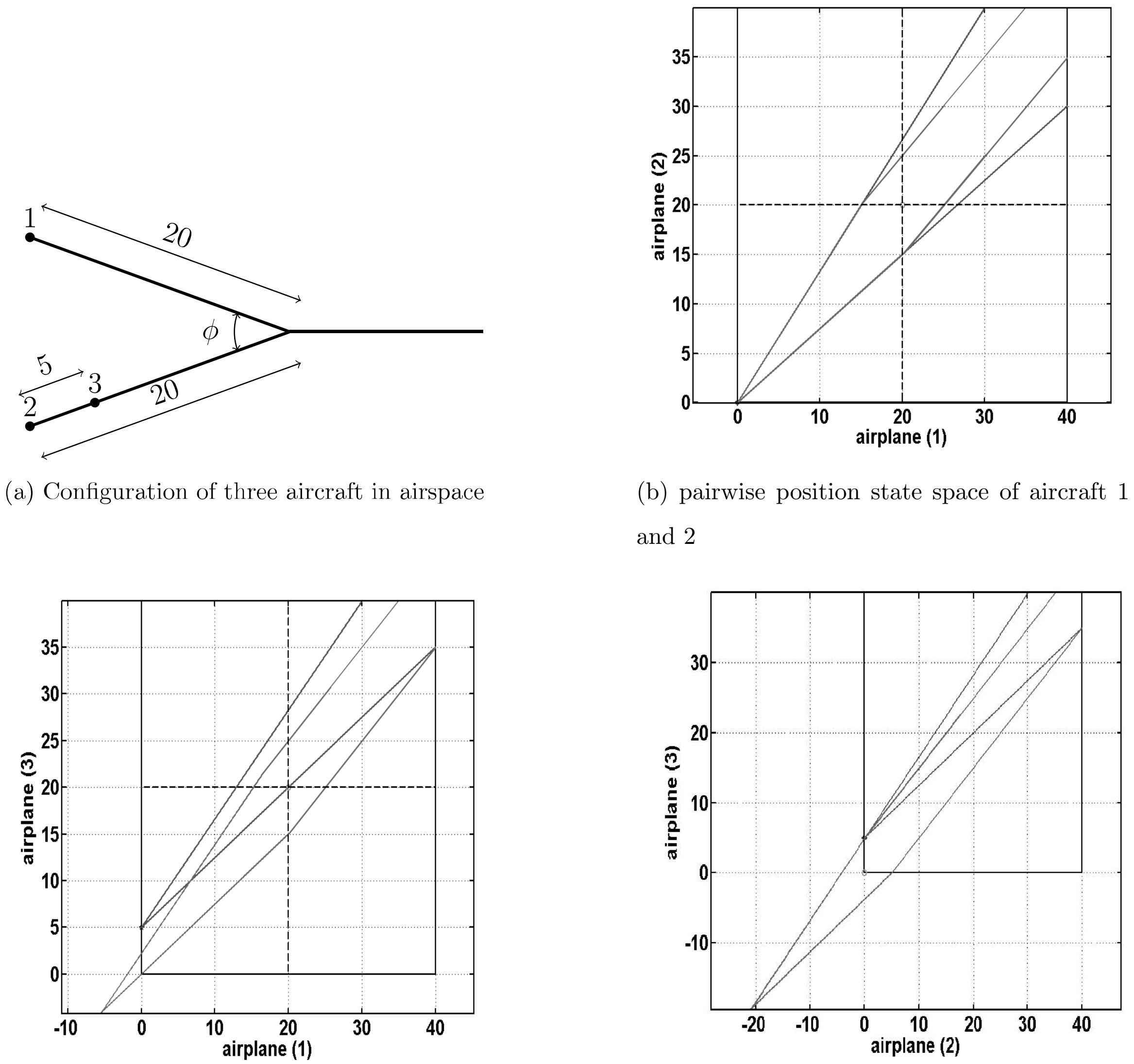
Figure 4.13: Different possible triangular path modification

## 4.4 Numerical example

Consider three aircraft in a given airspace. The configuration of these aircraft is shown in Figure 4.14. The minimum separation requirement between all pairs of aircraft is $r = 5$ nmi, $\phi = 40°$ and the speed range for aircraft are as follow,

$$\begin{aligned} V_1 &\in [300, 400] \quad \text{(nmi/hr)}, \\ V_2 &\in [300, 400] \quad \text{(nmi/hr)}, \\ V_3 &\in [300, 350] \quad \text{(nmi/hr)}. \end{aligned}$$

Also, this example is the counter example where the existence of a solution for each individual pairwise position state space, i.e. there is a separation-promising sequence, does not guarantee the existence of a speed control solution for all $N$ aircraft, i.e., there is no feasible-attainable trajectory. There are three pairwise position state spaces which are depicted in Figure 4.14. In this example, the following two sequences are separation-promising for all



(a) Configuration of three aircraft in airspace

(b) pairwise position state space of aircraft 1 and 2

(c) pairwise position state space of aircraft 1 and 3

(d) pairwise position state space of aircraft 2 and 3

Figure 4.14: The configuration of three aircraft in a given airspace and pairwise position state spaces.

three aircraft, but as will be shown no feasible-attainable trajectory exists in them.

$$Seq_1 = \{1, 2, 3\},$$

$$Seq_2 = \{2, 1, 3\}.$$

Therefore, there is no speed control solution and the path control along with the speed control solution must be considered. The algorithm picks the sequence with the higher likelihood of existence of a speed control solution. In this example, since the arc length of aircraft 1 is equal to the arc length of aircraft 2, both separation-promising sequences have the same likelihood. Here the first sequence has been chosen for path modification. In this sequence, by using the time-reversed trajectory generation method, the last aircraft to reach its destination is aircraft 1 (note, the aircraft are not renumbered based on the separation-promising sequence). The equation of the first segment is,

$$\mathbf{X}^{\max} - \lambda \mathbf{d}_1, \tag{4.9}$$

where

$$\mathbf{X}^{\max} = \mathbf{X}_1 = \begin{bmatrix} 40 & 40 & 40 \end{bmatrix}^T, \quad \mathbf{d}_1 = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T.$$

The feasible-attainable intersection associated with smallest positive $\lambda$ is at,

$$\mathbf{X}_2 = \begin{bmatrix} 34.92 & 40 & 40 \end{bmatrix}^T.$$

The first segment of the trajectory is depicted in Figure 4.15. The intersected hyperplane at

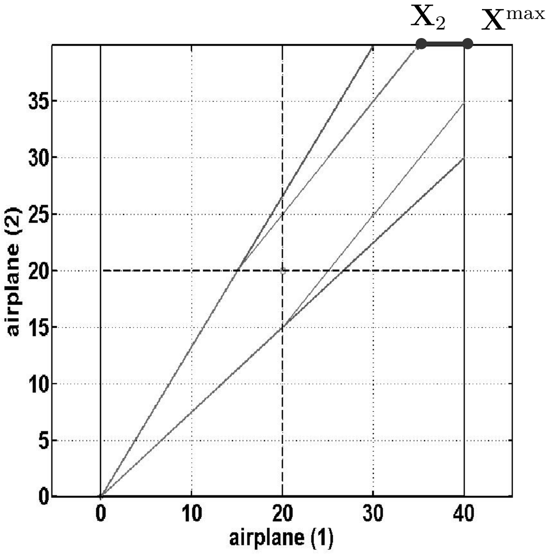


Figure 4.15: The first segment of the trajectory.

point $X_2$ is $\mathcal{H}_{12}^{uc}$, aircraft 2 reaches its destination at this state. The director of the second segment is

$$\mathbf{d}_2 = \begin{bmatrix} 1 & 1 & 0 \end{bmatrix}^T.$$

The equation of the second segment can be written as

$$\mathbf{X}_2 - \lambda \mathbf{d}_2. \tag{4.10}$$

In order to find the end point of the second segment, two cases must be considered, *i)* when a new aircraft reaches its destination (Step 3.(a)), *ii)* when no new aircraft reaches its destination (Step 3.(b)).

Case *i*: New aircraft reaches its destination:

In this case, aircraft 3 reaches its destination. Then, in order to find the end point of the second segment all three pairwise position state spaces must be considered. Hence, the intersection associated with the smallest positive $\lambda$ is,

$$\mathbf{Y}^* = \begin{bmatrix} 30 & 35.08 & 40 \end{bmatrix}^T.$$

The intersection $\mathbf{Y}^*$ in all pairwise position state spaces are depicted in Figure 4.16. At $\mathbf{Y}^*$ aircraft 1 has traveled 10 nmi and aircraft 2 has traveled 4.92 nmi. Therefore, aircraft 3 can not reach to its destination and must wait until aircraft 2 travel at least 5 nmi. This conflict can be seen in pairwise position state space between aircraft 2 and 3. In this pairwise position state space, the intersection point $\mathbf{Y}^*$ lies inside the separation-loss set. Hence, this case does not have solution.

Case *ii*: No new aircraft reaches its destination.

In this case, there are two aircraft in the airspace, then, there is only one pairwise subspace and the intersection is

$$\mathbf{Y}^* = \begin{bmatrix} 15.23 & 20.3 & 40 \end{bmatrix}^T,$$

which lies inside the intersection of the attainable cone and the separation compliance state set, See Figure 4.17. Then, by considering aircraft 3 this point will lie outside of the attainable cone. This is depicted in Figure 4.18.

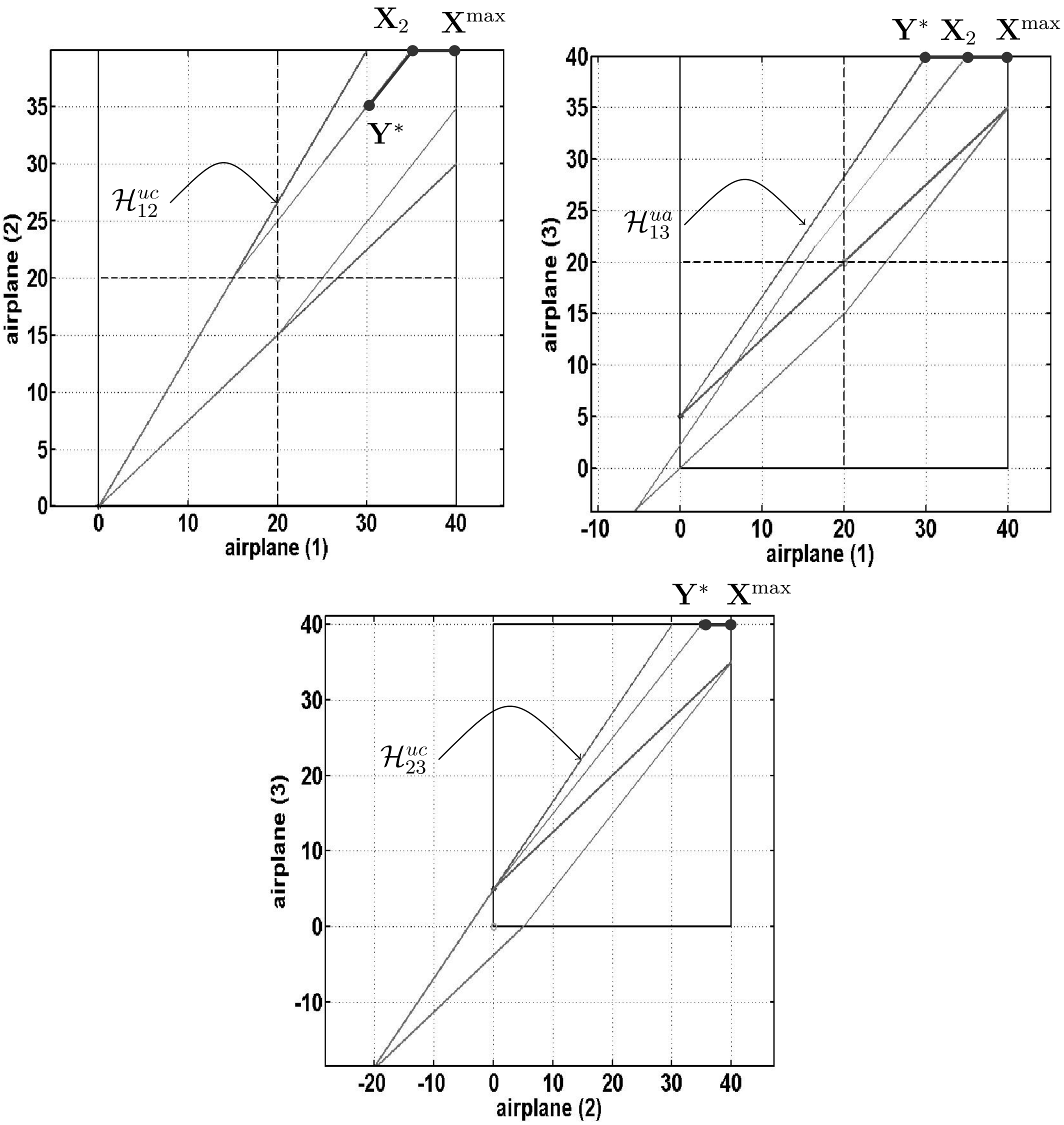


Figure 4.16: The intersection point $\mathbf{Y}^*$ in all pairwise position state spaces.

Therefore, no feasible-attainable intersection exists and the path control must be considered. The path control will be considered for the Case $i$, when the third aircraft reaches its destination. As it is illustrated in Figure 4.16, the intersection point lies inside the attainable cone but not the separation compliance state set. Hence, the method described in Section 4.2.2.2 is used. The path of aircraft 1 must be elongated by $\Delta\ell_1 = 0.0771$ nmi. After modifying the path of the aircraft 1, the pairwise position state spaces associated with aircraft 1 are changed. The modified pairwise position state spaces are depicted in Figure 4.19. After

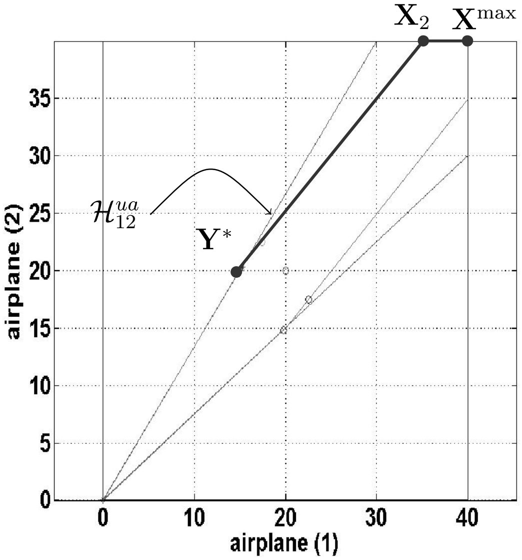


Figure 4.17: Position of intersection $\mathbf{Y}^*$ when no new aircraft reaches its destination.

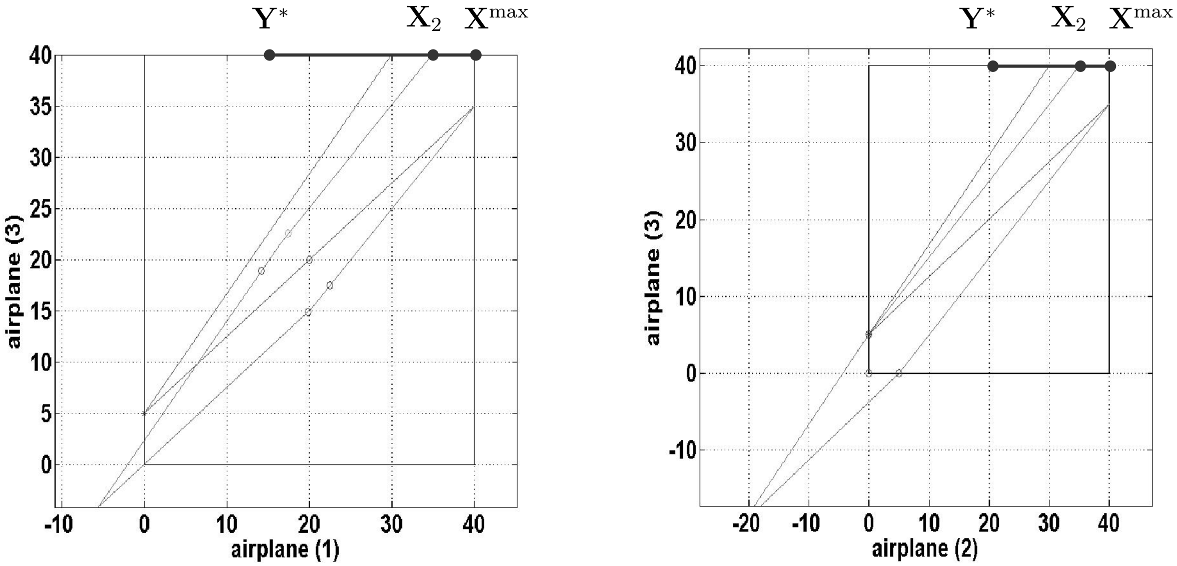


Figure 4.18: Position of intersection $\mathbf{Y}^*$, after considering aircraft 3.

modifying the pairwise position state spaces, the process of generating the feasible-attainable trajectory must start all over. Then, the first segment of the trajectory can be describe by equation (4.9) and ends at the intersection,

$$\mathbf{X}_2 = \begin{bmatrix} 35 & 40 & 40 \end{bmatrix}^T. \tag{4.11}$$

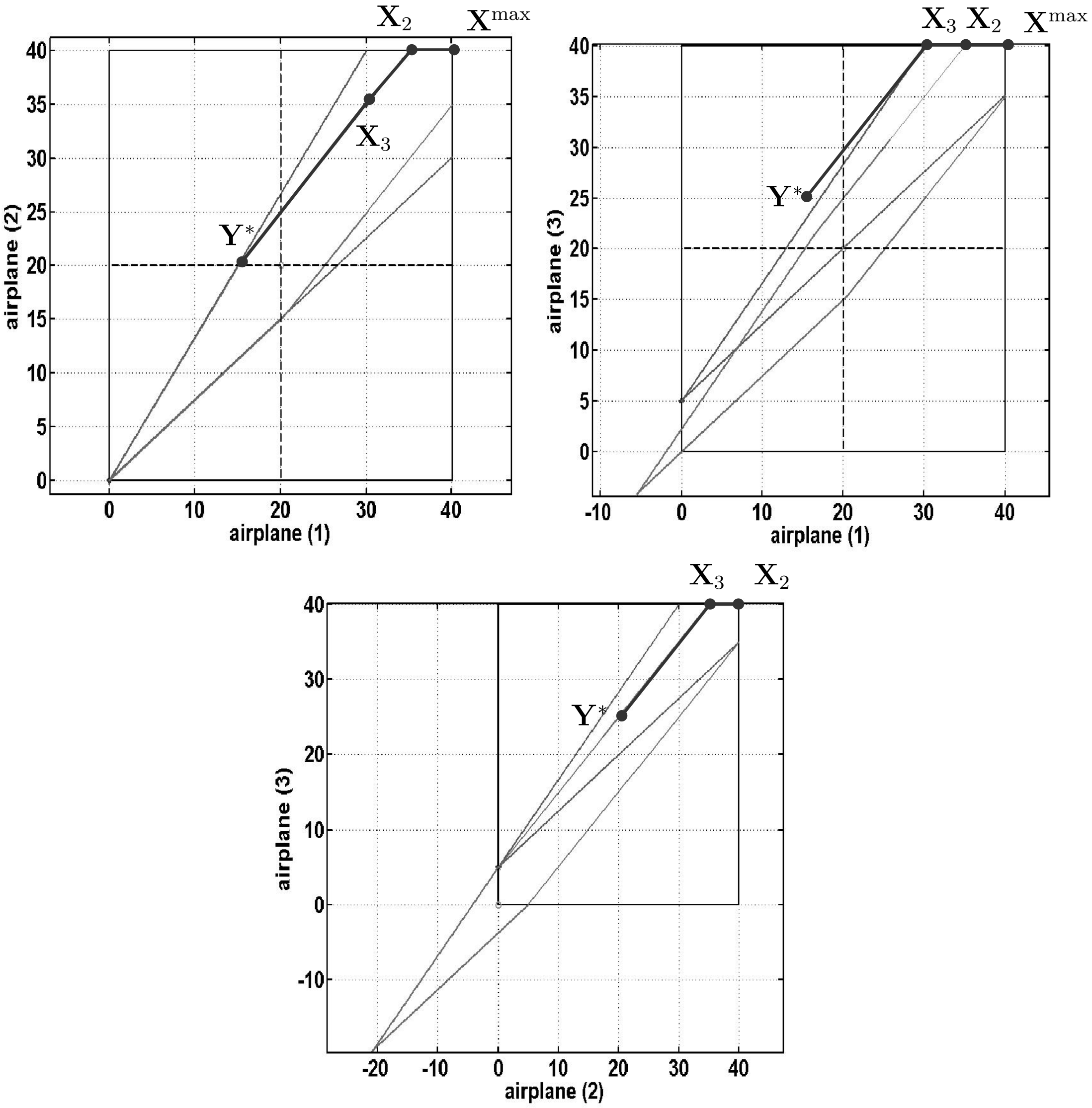


Figure 4.19: The intersection point $\mathbf{Y}^*$ in all pairwise position state spaces.

The second segment of the trajectory is described by equation (4.10), where $\mathbf{X}_2$ is given in equation (4.11). The second segment intersects the hyperplane $\mathcal{H}_{23}^{uc}$ and hyperplane $\mathcal{H}_{13}^{ua}$ at point

$$\mathbf{X}_3 = \begin{bmatrix} 30 & 35 & 40 \end{bmatrix}^T,$$

which is associated with the smallest positive $\lambda$. Therefore, two new guiding hyperplanes must be added to the guiding hyperplane set, $\mathbb{H}_2$. Aircraft 3 reached its destination, so there are 3 aircraft in the airspace. Then, 2 independent guiding hyperplanes are required while 3 guiding hyperplanes exist, the old one, $\mathcal{H}_{12}^{uc}$ and two new guiding hyperplanes $\mathcal{H}_{13}^{ua}$ and $\mathcal{H}_{23}^{uc}$. This case can be resolved in three different perspectives.

1) If both new guiding hyperplanes are added to the guiding hyperplane set, $\mathbb{H}_3$, then the normal vectors of these guiding hyperplanes form matrix $\mathbf{V}_3$.

$$\mathbf{V}_3 = \begin{bmatrix} -1.1667 & 0 & 1 \\ 0 & -1 & 1 \\ -1 & 1 & 0 \end{bmatrix}.$$

The null space of matrix $\mathbf{V}_3$ is empty. Therefore, no director can be found for the third segment. This is the case described in Section 4.2.1. Then, segment 2 must be extended until the current temporary guiding hyperplane ($\mathcal{H}_{12}^{uc}$) becomes inactive.

2 ) If just one of the new guiding hyperplanes is added to the guiding hyperplane set:

2a) if $\mathcal{H}_{23}^{uc}$ is added to the guiding hyperplane set, then matrix $\mathbf{V}_3$ and director $\mathbf{d}_3$ are

$$\mathbf{V}_3 = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 1 & 0 \end{bmatrix} \Rightarrow \mathbf{d}_3 = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}^T.$$

Then, the intersection point,

$$\mathbf{Y}^* = \begin{bmatrix} 15.3023 & 20.3023 & 25.3023 \end{bmatrix}^T, \tag{4.12}$$

will lie outside the attainable cone of aircraft 1 and 3. That is, the intersection lies inside the separation compliance state set, but not inside the attainable cone. This is described in Section 4.2.2.1. This is depicted in Figure 4.19.

2b) if $\mathcal{H}_{13}^{ua}$ is added to the guiding hyperplane set, then matrix $\mathbf{V}_3$ and director $\mathbf{d}_3$ has the following forms

$$\mathbf{V}_3 = \begin{bmatrix} -1.1667 & 0 & 1 \\ -1 & 1 & 0 \end{bmatrix} \Rightarrow \mathbf{d}_3 = \begin{bmatrix} 1 & 1 & 1.1667 \end{bmatrix}^T.$$

Then, the intersection point,

$$\mathbf{Y}^* = \begin{bmatrix}15.3023 & 20.3023 & 22.8522\end{bmatrix}^T .$$

will lie inside the separation-loss of aircraft 2 and 3. In other words, the intersection lies inside the attainable cone and not inside the separation compliant state set. This is the case described in Section 4.2.2.2. This situation is depicted in Figure 4.20.

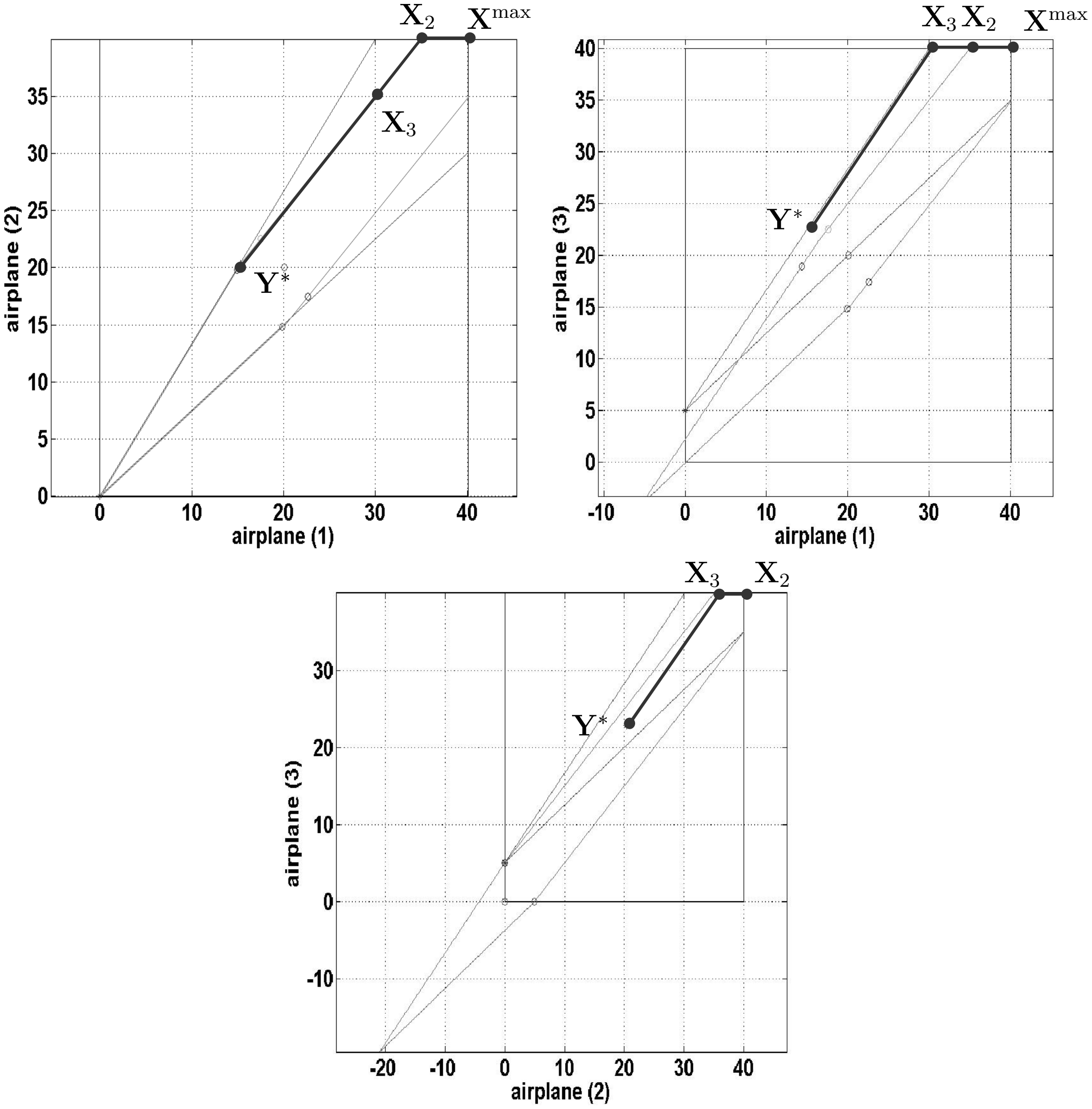


Figure 4.20: The intersection point $\mathbf{Y}^*$ in all pairwise position state spaces.

The goal of this algorithm is to find the trajectory which provides minimum possible spacing between all pairs of aircraft. This goal will be obtained in (2a). In (2a), the third segment lies on the boundary of separation-loss set which means the corresponding pair of aircraft have distance equal to the over estimated minimum separation requirement. The third segment ends at the intersection associated to the smallest positive $\lambda$ which is given in equation (4.12). By using the method described in Section 4.2.2.1, the path of aircraft 1 must be elongated by $\Delta \ell_1 = 2.0997$ nmi. Then, all pairwise position state spaces associated with aircraft 1 must be updated and the trajectory must be generated all over. The updated pairwise position state spaces are depicted in Figure 4.21. After modifying the pairwise position state spaces, a feasible-attainable trajectory can be generated. The speed for each aircraft is given in table 4.1. Therefore, to get the feasible-attainable trajectory, the path of aircraft 1 must be elongated by $\Delta \ell_1 = 2.1768$ nmi. By using method described in Section 4.3, the new path can be assigned to aircraft 1. The new path is depicted in Figure 4.22. The radius of circles are equal to half of the minimum separation requirement. The Figures 4.22 -4.27 show the positions of aircraft from $t = 0$ to $t = 5$ min.

| $\Delta$ t | 208.82 | 151.18 | 51.43 | 52.22 |
|---|---|---|---|---|
| $V_1$ | 300 | 350 | 350 | 350 |
| $V_2$ | 350 | 350 | 350 | |
| $V_3$ | 350 | 350 | | |

Table 4.1: Speed profile for each aircraft, times are in seconds and velocities in nmi/hr.

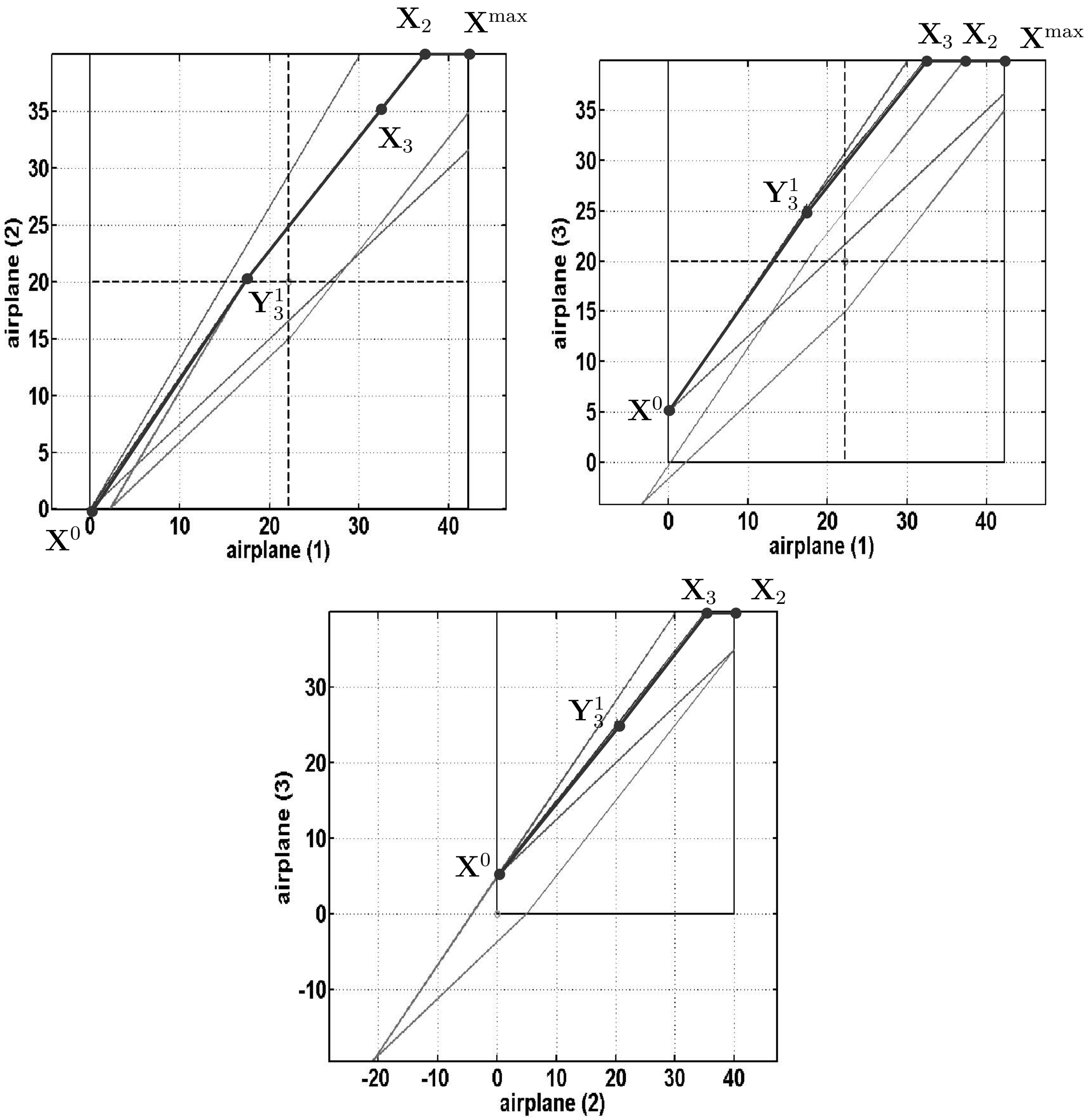


Figure 4.21: Feasible trajectory.

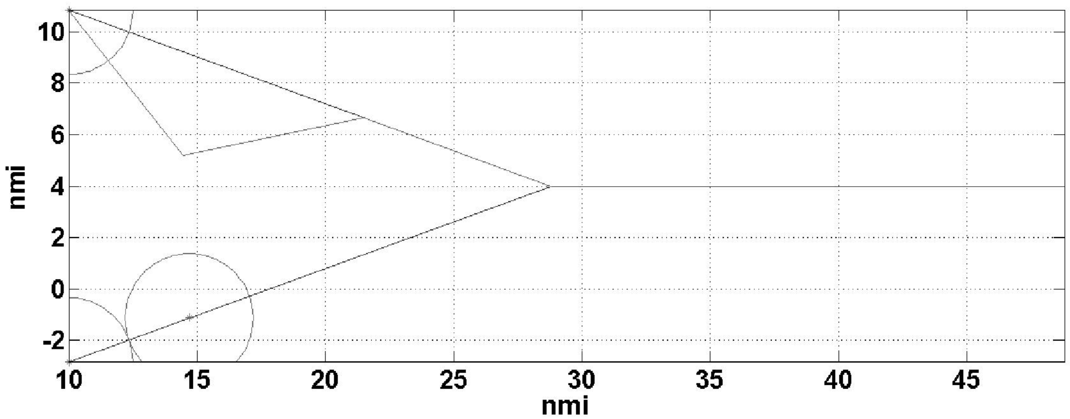


Figure 4.22: New path of aircraft 1 and positions of aircraft at $t = 0$.

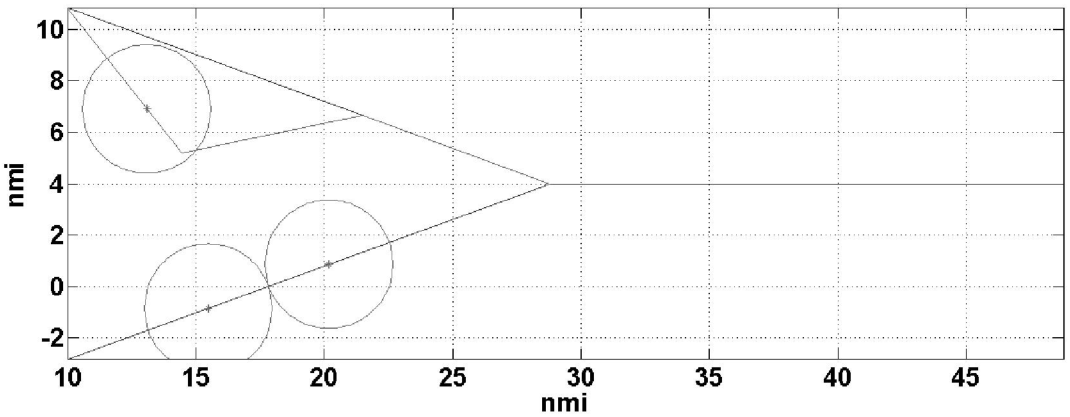


Figure 4.23: Positions of aircraft at $t = 1$ min.

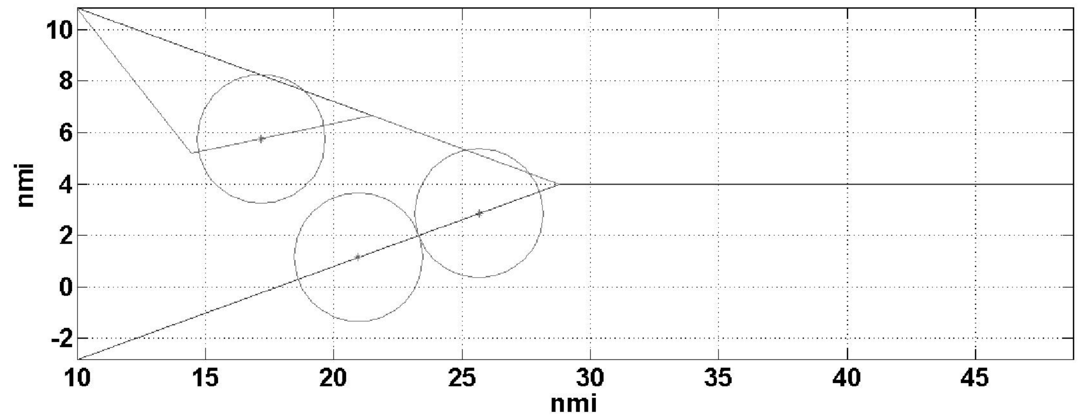


Figure 4.24: Positions of aircraft at $t = 2$ min.

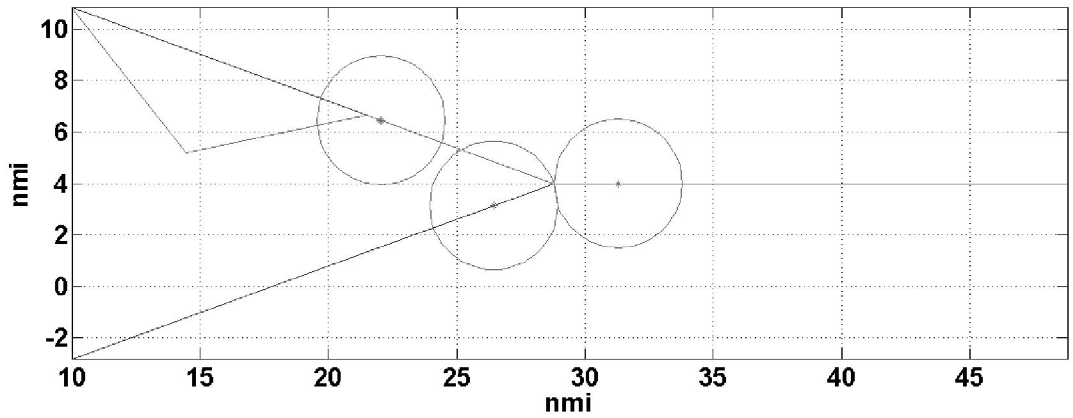


Figure 4.25: Positions of aircraft at $t = 3$ min.

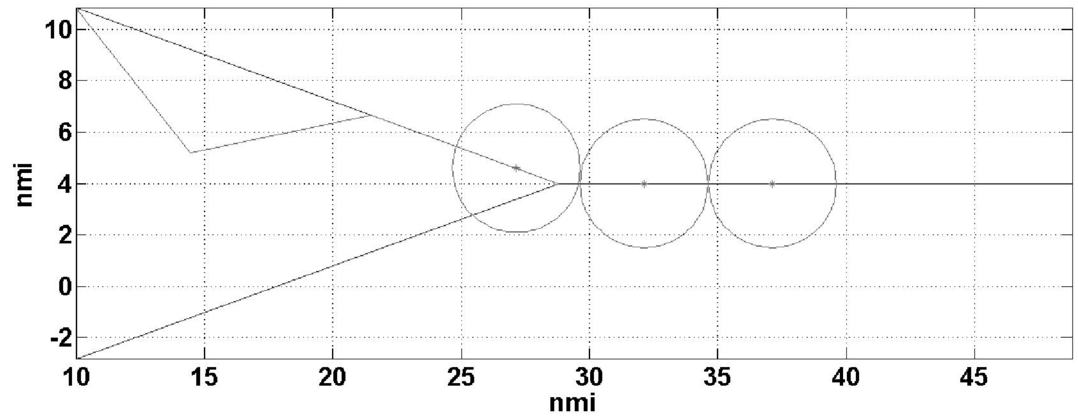


Figure 4.26: Positions of aircraft at $t = 4$ min.

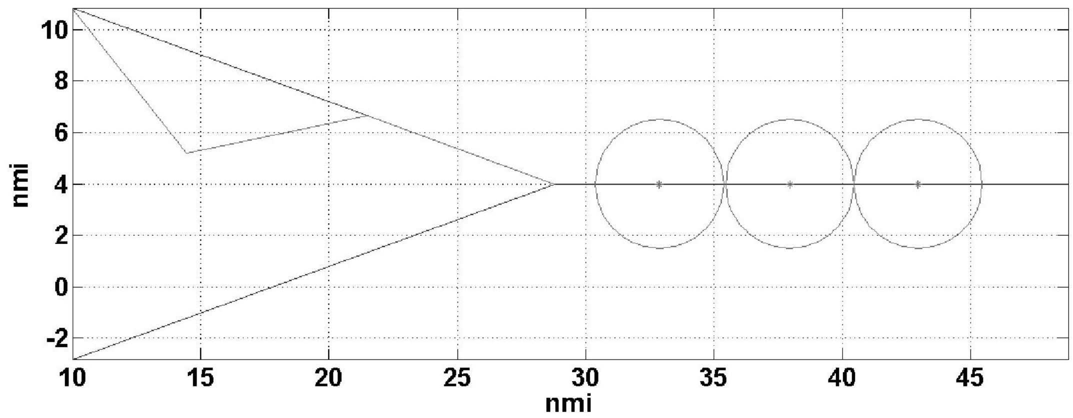


Figure 4.27: Positions of aircraft at $t = 5$ min.

# CHAPTER 5

## Conclusion and future research

The algorithm developed in Chapter 3 is searching for a feasible-attainable trajectory in all separation-promising 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 separation-promising sequence. The parallelism helps make it computationally practical. By increasing the number of aircraft, the number of separation-promising 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 separation-promising sequence and then look for speed control solution in the sequences with the higher probability.

The algorithm developed in Chapter 3 is extended in Chapter 4 to use the path control, when the speed control solution does not exist. If no separation-promising sequence exist, the algorithm will generate a new path, which will have the highest likelihood of the existence of a speed control solution. It has been shown that the schedule of aircraft arrival can be imposed to generate the separation-promising sequence. If more than one separation-promising sequence exist but there is no speed control solution, similar to speed control algorithm in Chapter 3, the path modification for each one can be done in parallel. The algorithm finds the set of aircraft that need path modification and is able to calculate the minimum elongation for each aircraft's path. The new path has a triangular shape and it will be defined by adding some auxiliary waypoints in the given airspace.

This algorithm 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 on-board FIM equipment until reaching a planned termination point. The interval management clearance issued by ATC is the minimum separation required, given the permissible 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.

There are some areas for further research and extension of this algorithm;

- This algorithm generates a discontinuous speed profile for each aircraft which implies infinite acceleration at those discontinuities. In order to get a continuous speed and acceleration profile for each aircraft, the aircraft dynamics must be considered. Also, the current algorithm picks some auxiliary waypoints which guarantee the separation requirement, but does not suggest the exact shape of the new route. The shape of the new route must be generated by considering the aircraft dynamics.

- If there are time constraints at waypoints, then the velocity must be varied. These additional constraints need to be incorporated into the speed and path control algorithm.

- Current scheme assumes constant permissible speed intervals for each aircraft among its assigned route and identical minimal separation requirement between aircraft pairs. However, minimal separation and permissible speed intervals may depend upon;

  - Aircraft weight and size in the terminal area.
  - It may be a restriction that one aircraft leads another, i.e. based on size and the vortex that may be generated.
  - Permissible speed interval narrows at landing. The effect of this narrowing may or may not be a cone velocity, but the attainable set most likely will be convex.

  These considerations can be included in the speed and path control algorithm.

- Robustness of the speed and path control scheme can be added by considering velocity and possibly position uncertainties due to environmental effects, such as weather.

- Currently, the speed control algorithm is guaranteed to find the speed control solution in a finite time, if such a solution exists. Also, after modifying the algorithm to consider the aircraft dynamics and environmental uncertainties, the algorithm must guarantee the computation time to be applicable to real-time air traffic management. An analytic solution for conflict resolution is sufficient to predict computation time. All addition constraints, such as uncertainties, time constraints at waypoints and path modification to retard or advance the aircraft along the nominal path, can be formulated such that the algorithms retain the property of convergence in polynomial time.

- The dimension of the geometric model introduced in Chapter 2 is equal to the number of aircraft in the given airspace. The dimension of the state space increases by one when a new aircraft is added to the airspace. Currently, the algorithm assumes a fixed number of aircraft. In the real-time situation aircraft are continuously adding and leaving the airspace. Therefore, the algorithm must be able to calculate the speed or path control in real-time situation. 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.

# APPENDIX A

# Calculation of pairwise separation-loss set

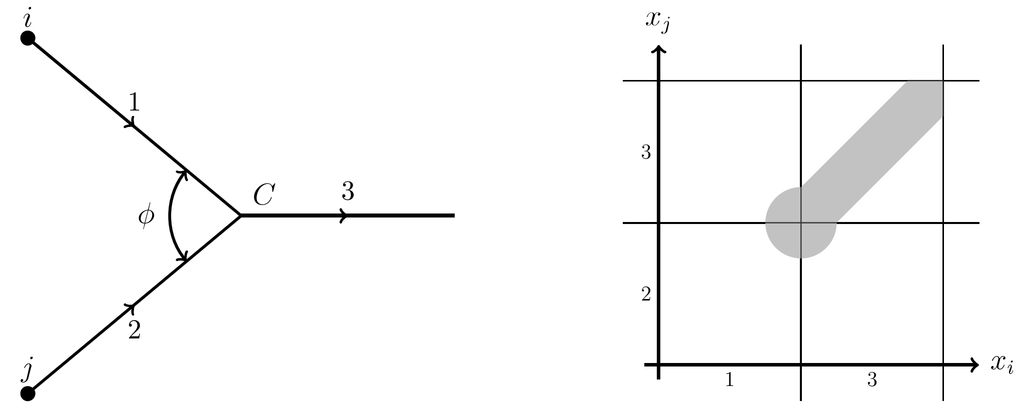
In the algorithm the $N$-dimensional separation-loss set is constructed by taking the union of all pairwise separation-loss sets. The pairwise infeasible state set is the union of separation-loss set and the roof set. In the following sections, the separation-loss set will be constructed and then the approximated separation-loss set and the pairwise infeasible state set will be formulated. The most common case to violate the separation is the merge points. Hence, calculation of the pairwise separation-loss set for a simple merge point is the most practical case to be considered.

## A.1 Pairwise separation-loss set

To calculate the pairwise separation-loss set, consider two aircraft flying on their own paths which they are merging at point $C$. The configuration and the corresponding separation-loss set are depicted in Figure A.1. The points on the boundaries of the pairwise separation-loss are representing the configurations where the distance between two aircraft is equal to the minimum required separation. All different cases of this configuration must be considered. Five different possible configuration between aircraft $i$ and $j$ where the distance between two aircraft is equal to the minimum required separation are as follows,

1) aircraft $i$ and $j$ are flying on segments 1 and 2, respectively, but aircraft $i$ is closer to the merge point,

2) aircraft $i$ and $j$ are flying on segments 1 and 2, respectively, but aircraft $j$ is closer to the merge point,



(a) The configuration of merging of aircraft $i$ and aircraft $j$ paths.

(b) The pairwise separation-loss set for two aircraft $i$ and $j$.

Figure A.1: The simple merge problem and the corresponding 2-dimensional state space.

3) aircraft $i$ and $j$ are flying on segments 1 and 3, respectively,

4) aircraft $i$ and $j$ are flying on segments 3 and 2, respectively,

5) both are flying on segment 3.

All of these cases are considered in the following sections.

### A.1.1 Aircraft $i$ and $j$ are flying on segments 1 and 2, respectively, but aircraft $i$ is closer to the merge point

In order to drive the equation for this case, define a local coordinate such that the $x$ axis of the coordinate is aligned with the segment 1 and the $y$ axis is perpendicular to segment 1. The local coordinate is depicted in Figure A.2. The separation-loss region for aircraft $i$ is shown by a dotted circle, where the radius of the circle is equal to the minimum separation requirement $r$. The equation of this circle in the local coordinate can be written as,

$$(x - x_{1_i})^2 + y^2 = r^2, \tag{A.1}$$

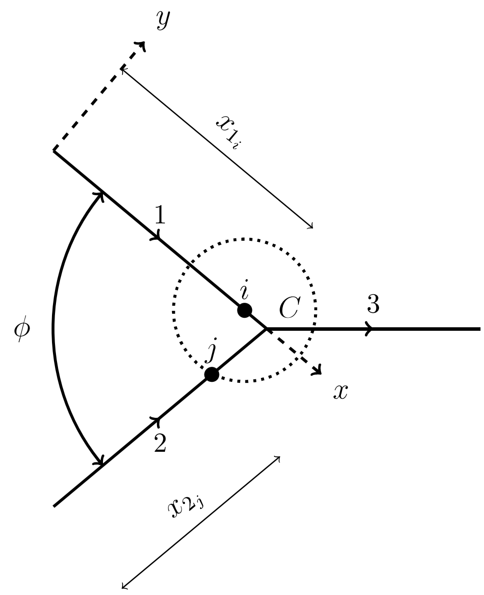


Figure A.2: The configuration of the first case. The radius of the circle around aircraft $i$ is equal to the minimum sepqration requirement.

where $x_{1_i}$ is the arc position of aircraft $i$ on segment 1. Aircraft $j$ is flying on segment 2, which the equation of this segment in the local coordinate is,

$$y = tan(\phi)\,(x - \ell_1)\,, \tag{A.2}$$

where $\ell_1$ is the total arc length of segment 1. Since, the distance between aircraft $i$ and $j$ is equal to $r$, then aircraft $j$ is on the intersection of the safety circle and segment 2. Note that, in this scenario the separation can be violated if

$$0 < \phi < 90°. \tag{A.3}$$

From equations (A.1) and (A.2), the $x$-component of aircraft $j$ position with respect to local coordinate is

$$x = x_{1_i}\cos^2\phi + \ell_1\sin^2\phi \pm \cos\phi\sqrt{r^2 - (\ell_1 - x_{1_i})^2\sin^2\phi}. \tag{A.4}$$

Since, aircraft $j$ is further from the merge point than aircraft $i$, then the minus sign in equation (A.4) is acceptable. This equation valid after the point where first intersection occurs. The first intersection occurs at,

$$x_{1_i} = \ell_1 - \frac{r}{\sin\phi}. \tag{A.5}$$

In order to find the boundary of separation-loss set in pairwise subspace, the equation (A.4) must be written as a function of arc length positions of aircraft $i$ and $j$, $x_i$ and $x_j$, respectively. In this case, the arc length position of aircraft $i$ is equal to the position of aircraft $i$ in segment 1,

$$x_i = x_{1_i}. \tag{A.6}$$

In equation (A.4), $x$ is the x-component of aircraft $j$ position with respect to the local coordinate. $x$ can be written as the arc length position of aircraft $j$

$$x = \ell_1 - \left(\ell_2 - x_{2_j}\right)\cos\phi, \tag{A.7}$$

$$x_j = x_{2_j}. \tag{A.8}$$

The suitable equation to use in 2-dimensional state space can be obtain by substituting equations (A.6), (A.7) and (A.8) into equation (A.4).

$$x_j = \ell_2 - \left[(\ell_1 - x_i)\cos\phi + \sqrt{r^2 - (\ell_1 - x_i)^2\sin^2\phi}\right]. \tag{A.9}$$

According to equations (A.3) and (A.5), equation (A.9) is valid when,

$$0 < \phi < 90^\circ,$$

$$\ell_1 - \frac{r}{\sin\phi} \leq x_i \leq \ell_1.$$

The boundary of separation-loss set corresponding to the first case is depicted in Figure A.3.

### A.1.2 Aircraft $i$ and $j$ are flying on segments 1 and 2, respectively, but aircraft $j$ is closer to the merge point

This is similar to the first case. The configuration and the local coordinate for this case is depicted in Figure A.4. Similarly, the separation can be violated if

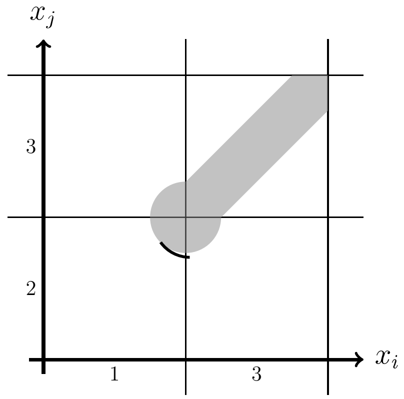


Figure A.3: The boundary of separation-loss set corresponding to the first case.

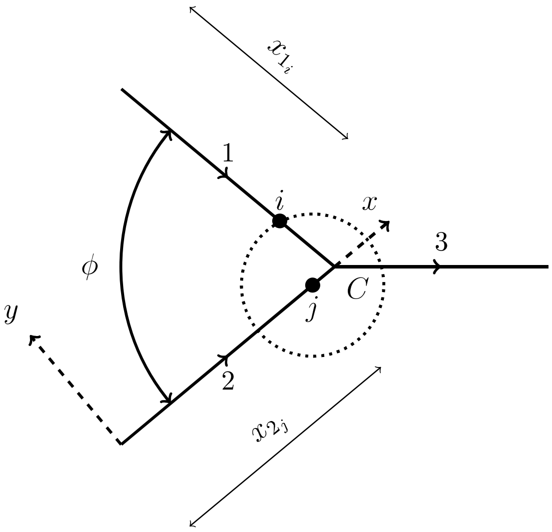


Figure A.4: The configuration of the second case. The radius of the circle around aircraft $i$ is equal to the minimum separation requirement.

$$0 < \phi < 90° \tag{A.10}$$

Then, the equation for separation region of aircraft $j$ with respect to the local coordinate is

$$\left(x - x_{2_j}\right)^2 + y^2 = r^2,$$

and the equation of segment 1 in the local coordinate is

$$y = -\tan(\phi)\left(x - \ell_2\right).$$

Then, the $x$-component of aircraft $i$ position with respect to local coordinate can be found as the $x$-component of the intersection point between the separation region of aircraft $j$ and segment 1 that is given as

$$x = x_{2_j} \cos^2 \phi + \ell_2 \sin^2 \phi \pm \cos \phi \sqrt{r^2 - \left(\ell_2 - x_{2_j}\right)^2 \sin^2 \phi}. \tag{A.11}$$

Similar to the previous case, in equation (A.11) the minus sign is acceptable. This equation valid after the point where the first intersection between the separation region and segment 1 occurs. Then

$$x_{2_j} = \ell_2 - \frac{r}{\sin \phi}. \tag{A.12}$$

The arc length position of each aircraft is given by,

$$x_i = x_{1_i}, \tag{A.13}$$

$$x_j = x_{2_j}. \tag{A.14}$$

Equation (A.11) gives the $x$-component of aircraft $i$ position with respect to the local coordinate, which also can be expressed as a function of the arc position of aircraft $i$ on segment 1.

$$x = \ell_2 - (\ell_1 - x_{1_i}) \cos \phi. \tag{A.15}$$

Substitute equations (A.13), (A.14) and (A.15) into equation (A.11) to solve for $x_i$ as a function of $x_j$.

$$x_i = \ell_1 - \left[ (\ell_2 - x_j) \cos \phi + \sqrt{r^2 - (\ell_2 - x_j)^2 \sin^2 \phi} \right]. \tag{A.16}$$

From equations (A.10) and (A.12), equation (A.16) is valid if

$$0 < \phi < 90^\circ,$$
$$\ell_2 - \frac{r}{\sin \phi} \leq x_j \leq \ell_2.$$

The boundary of the separation-loss set corresponding to the second case is depicted in Figure A.5.

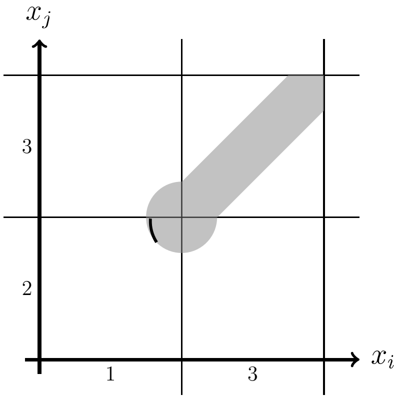


Figure A.5: The boundary of the separation-loss set corresponding to the second case.

### A.1.3 Aircraft $j$ is flying on segment 3 while aircraft $i$ is flying on segments 1

In this case, aircraft $j$ is flying on segment 3 while aircraft $i$ is flying on segment 1. The $x$-component of the local coordinate is set to be aligned with segment 1. Therefore, the separation region of aircraft $i$ and the equation of segment 3 can be express as:

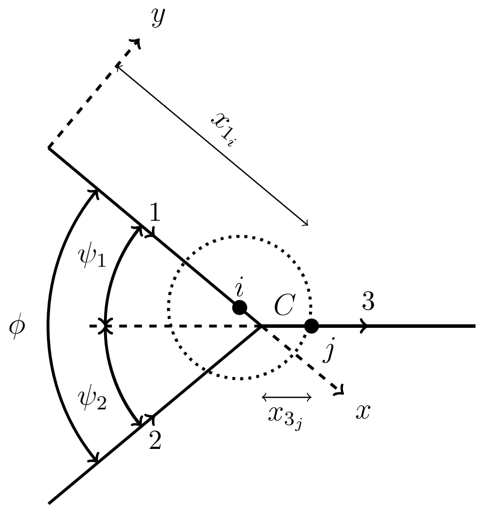


Figure A.6: The configuration of the third case. The radius of the circle around aircraft $j$ is equal to the minimum separation requirement.

$$(x - x_{1_i})^2 + y^2 = r^2$$

$$y = \tan(\psi_1)\,(x - \ell_1)$$

Therefore, the $x$-component of the intersection points are

$$x = x_{1_i} \cos^2 \psi_1 + \ell_1 \sin^2 \psi_1 \pm \cos \psi_1 \sqrt{r^2 - (\ell_1 - x_{1_i})^2 \sin^2 \psi_1}. \tag{A.17}$$

Since, aircraft $j$ is ahead of aircraft $i$, in equation (A.17), the answer associated with the positive sign is the $x$-component of aircraft $j$ position. The equation (A.17) is valid when

$$\ell_1 - r \leq x_{1_i} \leq \ell_1,$$
$$-90^\circ \leq \psi_1 \leq +90^\circ,$$

where angle $\psi_1$ is measured from segment 1. The $x$-component of aircraft $j$ position can be written as a function of arc position of aircraft $j$ on segment 3,

$$x = \ell_1 + x_{3_j} \cos \psi_1. \tag{A.18}$$

The arc length positions of aircraft $i$ and $j$ can be written as:

$$x_i = x_{1_i}, \tag{A.19}$$

$$x_j = \ell_2 + x_{3_j}., \tag{A.20}$$

The equation of the corresponding piece of the separation-loss boundary can be determined by substituting equations (A.18), (A.19) and (A.20) into equation (A.17).

$$x_j = \ell_2 - (\ell_1 - x_i) \cos \psi_1 + \sqrt{r^2 - (\ell_1 - x_i)^2 \sin^2 \psi_1}. \tag{A.21}$$

The corresponding separation-loss boundary for the third case is depicted in Figure A.7.

### A.1.4 Aircraft $i$ is flying on segment 3 while aircraft $j$ is flying on segments 2

This case is similar to the previous one. In this case, aircraft $i$ is flying on segment 3 while aircraft $j$ is flying on segment 2. The $x$-component of the local coordinate is set to be aligned with segment 2. Therefore, the separation region of aircraft $j$ and the equation of segment 3 can be express as:

$$(x - x_{2_j})^2 + y^2 = r^2$$
$$y = -\tan(\psi_2)\,(x - \ell_2)$$

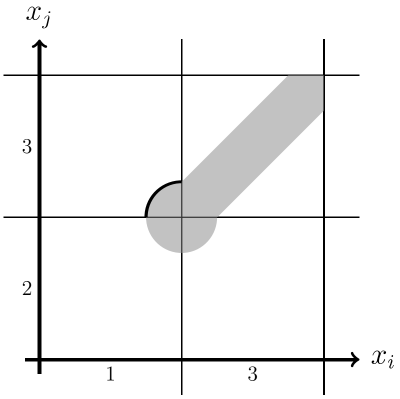


Figure A.7: The boundary of separation-loss set corresponding to the third case.

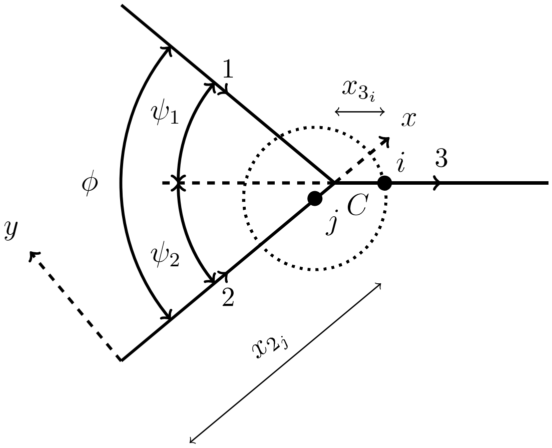


Figure A.8: The configuration of the fourth case. The radius of the circle around aircraft $j$ is equal to the minimum separation requirement.

Therefore, the $x$-component of the intersection points are

$$x = x_{2_j} \cos^2 \psi_2 + \ell_2 \sin^2 \psi_2 \pm \cos \psi_2 \sqrt{r^2 - \left(\ell_2 - x_{2_j}\right)^2 \sin^2 \psi_2}. \tag{A.22}$$

In equation (A.22), the answer associated with the positive sign is the $x$-component of aircraft $i$ position. The equation (A.22) is valid when

$$\ell_2 - r \leq x_{2_j} \leq \ell_2,$$

$$-90^\circ \leq \psi_2 \leq +90^\circ,$$

where angle $\psi_2$ is measured from segment 2. The $x$-component of aircraft $i$ position can be written as a function of arc position of aircraft $i$ on segment 3,

$$x = \ell_2 + x_{3_i} \cos\psi_2. \tag{A.23}$$

The arc length positions of aircraft $i$ and $j$ can be written as:

$$x_i = \ell_1 + x_{3_i}, \tag{A.24}$$

$$x_j = x_{2_j}. \tag{A.25}$$

The equation of the corresponding piece of the separation-loss boundary can be determined by substituting equations (A.23), (A.24) and (A.25) into equation (A.22).

$$x_i = \ell_1 - (\ell_2 - x_j)\cos\psi_2 + \sqrt{r^2 - (\ell_2 - x_j)^2 \sin^2\psi_2}. \tag{A.26}$$

The corresponding separation-loss boundary for the fourth case is depicted in Figure A.9.

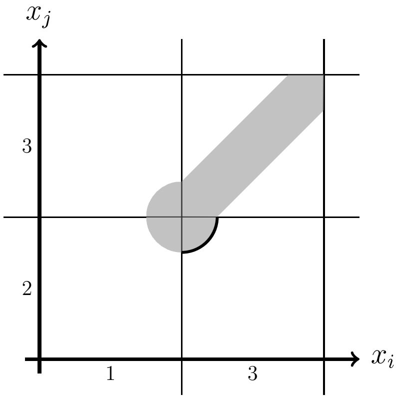
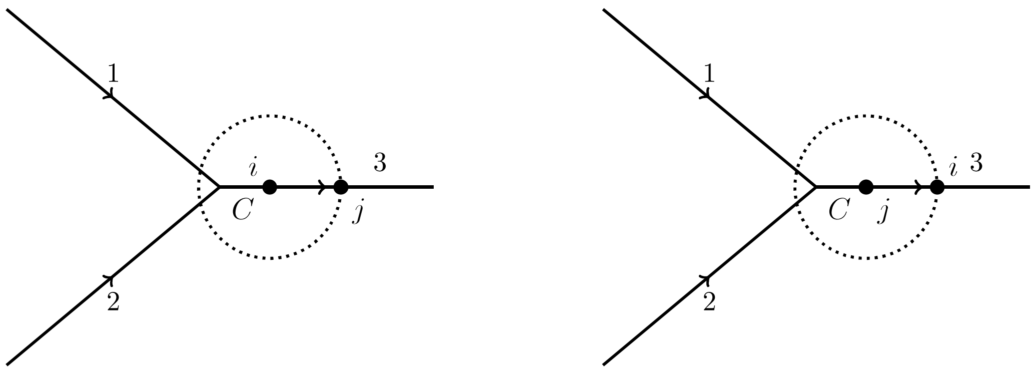


Figure A.9: The boundary of separation-loss set corresponding to the fourth case.

### A.1.5 Both are flying on segment 3

In this case, there are two configurations, 1) when aircraft $i$ is trailing aircraft $j$ 2) when aircraft $j$ is trailing aircraft $i$. Both configurations are depicted in Figure A.10.

In this case, in order to keep both aircraft at the minimum separation requirement distance, both must fly with the same speed. Therefore, the slope of the corresponding bound-



(a) When aircraft $i$ is trailing aircraft $j$.

(b) When aircraft $j$ is trailing aircraft $i$.

Figure A.10: Two configurations of aircraft $i$ and $j$ when both are flying on segment 3.

aries are equal to one. The first configuration (shown in Figure A.10a) can happen when,

$$x_{3_j} - x_{3_i} = r, \tag{A.27}$$

and the arc length position of aircraft $i$ and $j$ can be written as

$$x_i = \ell_1 + x_{3_i}, \tag{A.28}$$

$$x_j = \ell_2 + x_{3_j}. \tag{A.29}$$

Substitute equations (A.28) and (A.29) into (A.27)

$$x_j = x_i + (\ell_2 - \ell_1) + r.$$

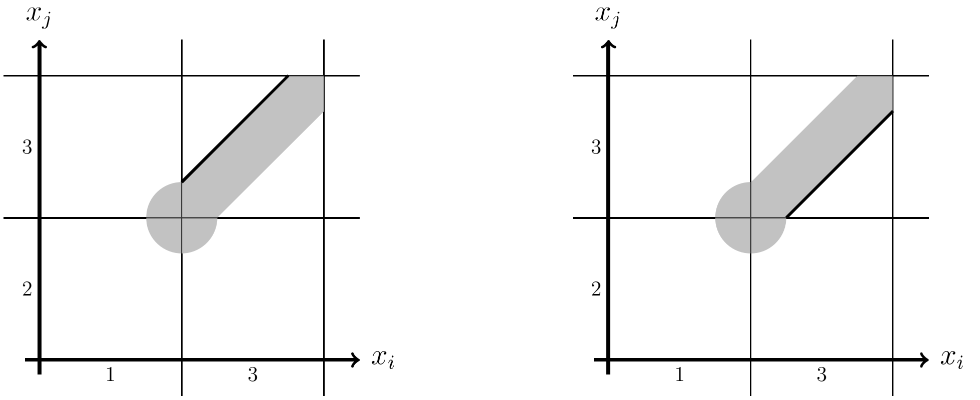
This boundary is depicted in Figure A.11a. Similarly, the second configuration (shown in Figure A.10b) happens when,

$$x_{3_i} - x_{3_j} = r, \tag{A.30}$$

Substitute equations (A.28) and (A.29) into (A.30)

$$x_j = x_i + (\ell_2 - \ell_1) - r.$$

This boundary is depicted in Figure A.11b.



(a) The boundary of separation-loss set when aircraft $i$ is trailing aircraft $j$.

(b) The boundary of separation-loss set when aircraft $j$ is trailing aircraft $i$.

Figure A.11: The boundary of separation-loss set corresponding to the fifth case.

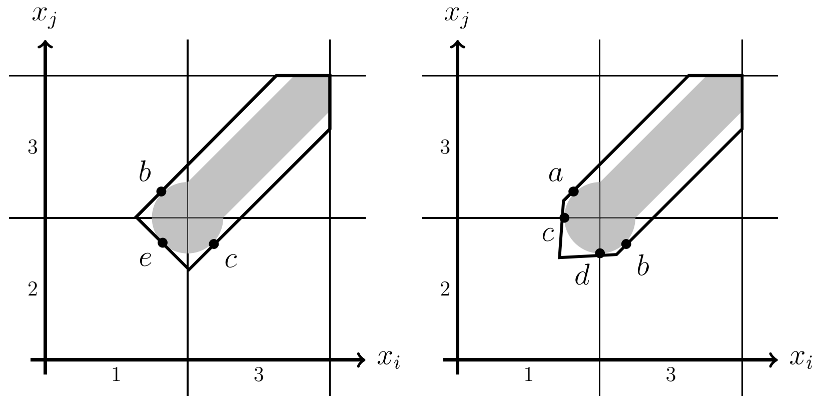
## A.2 Approximated pairwise separation-loss set

The separation-loss set is complicated to use. So a polyhedral approximation of this set is considered as the separation-loss set. The approximated separation-loss set is depicted in Figure A.12a. The algorithm uses the union of the separation-loss and the roof sets as the infeasible state set. The infeasible state set is depicted in Figure A.12b The boundaries of infeasible state set are tangent to the boundaries of the original separation-loss set at points $a$ and $b$ with the slope one and at points $c$ and $d$ with the slope $\overline{s}_{ij}$ and $\underline{s}_{ij}$, respectively. In the following, equation of each boundary of the infeasible state set will be calculated. At Point $a$, the slop is equal to one, so point $a$ can be found by setting the slope of the equation of the third case equal to one. From equation A.21.

$$\frac{dx_j}{dx_i} = \cos\psi_1 + \frac{(\ell_1 - x_i)\sin^2\psi_1}{\sqrt{r^2 - (\ell_1 - x_i)^2 \sin^2\psi_1}} = 1,$$

$$\Rightarrow \quad \begin{cases} x_i^a = \ell_1 - \frac{r}{\sin\psi_1}\sqrt{\frac{1-\cos\psi_1}{2}}, \\ x_j^a = \ell_2 + \frac{r}{\sin\psi_1}\sqrt{\frac{1-\cos\psi_1}{2}}, \end{cases}$$

where $x_i^a$ and $x_j^a$ are the corresponding arc length position of aircraft $i$ and $j$ at point $a$, respectively. The equation of the boundary of the infeasible state set that is tangent to the



(a) Approximated separation-loss set.

(b) Infeasible state set.

Figure A.12: Approximated separation-loss and infeasible state set.

original set at point $a$ is

$$x_j = x_i + (x_j^a - x_i^a).$$

Also, at point $b$, the slope of the boundary is one. Therefore, from equation (A.26),

$$\frac{dx_j}{dx_i} = \frac{\sqrt{r^2 - (\ell_2 - x_j)^2 \sin^2 \psi_2}}{(\ell_2 - x_j)\sin^2 \psi_2 + \cos\psi_2 \sqrt{r^2 - (\ell_2 - x_j)^2 \sin^2 \psi_2}} = 1,$$

$$\Rightarrow \quad \begin{cases} x_i^b = \ell_1 + \frac{r}{\sin\psi_2}\sqrt{\frac{1-\cos\psi_2}{2}}, \\ x_j^b = \ell_2 - \frac{r}{\sin\psi_2}\sqrt{\frac{1-\cos\psi_2}{2}}, \end{cases}$$

where $x_i^b$ and $x_j^b$ are the corresponding arc length position of aircraft $i$ and $j$ at point $b$, respectively. The equation of the boundary of the infeasible state set that is tangent to the original set at point $b$ is

$$x_j = x_i + (x_j^b - x_i^b).$$

The slope of the boundary of infeasible state set at point $c$ is equal to $\overline{s}_{ij}$. Point $c$ can be on the boundary corresponding to the third case or the boundary corresponding to the second case.

- If point $c$ is on the boundary corresponding to the third case, then from equation (A.21)

$$\frac{dx_j}{dx_i} = \cos\psi_1 + \frac{(\ell_1 - x_i)\sin^2\psi_1}{\sqrt{r^2 - (\ell_1 - x_i)^2 \sin^2\psi_1}} = \overline{s}_{ij},$$
$$\Rightarrow \quad \begin{cases} x_i^c = \ell_1 - \frac{r(\overline{s}_{ij} - \cos\psi_1)}{\sin\psi_1\sqrt{\sin^2\psi_1 + (\overline{s}_{ij} - \cos\psi_1)^2}}, \\ x_j^c = \ell_2 + \frac{r - r\overline{s}_{ij}\cos\psi_1}{\sin\psi_1\sqrt{\sin^2\psi_1 + (\overline{s}_{ij} - \cos\psi_1)^2}}, \end{cases}$$

where $x_i^c$ and $x_j^c$ are the corresponding arc length position of aircraft $i$ and $j$ at point $c$, respectively.

- If point $c$ is on the boundary corresponding to the second case, then from equation(A.16)

$$\frac{dx_j}{dx_i} = \frac{\sqrt{r^2 - (\ell_2 - x_j)^2 \sin^2\phi}}{\cos\phi\sqrt{r^2 - (\ell_2 - x_j)^2\sin^2\phi} - (\ell_2 - x_j)\sin^2\phi} = \overline{s}_{ij},$$
$$\Rightarrow \quad \begin{cases} x_i^c = \ell_1 - \frac{r|\overline{s}_{ij}\cos\phi - 1|\cos\phi + r\overline{s}_{ij}\sin^2\phi}{\sin\phi\sqrt{(\overline{s}_{ij}\cos\phi - 1)^2 + \overline{s}_{ij}^2\sin^2\phi}}, \\ x_j^c = \ell_2 - \frac{r|\overline{s}_{ij}\cos\phi - 1|}{\sin\phi\sqrt{(\overline{s}_{ij}\cos\phi - 1)^2 + \overline{s}_{ij}^2\sin^2\phi}}. \end{cases}$$

The equation of the boundary of the infeasible state set that is tangent to the original set at point $c$ is

$$x_j = \overline{s}_{ij}x_i + (x_j^c - \overline{s}_{ij}x_i^c). \tag{A.31}$$

The slope of the boundary of infeasible state set at point $d$ is equal to $\underline{s}_{ij}$. Point $d$ can be on the boundary of the fourth case or on the boundary of the first case.

- If point $d$ is on the boundary of the fourth scenario then from equation (A.26)

$$\frac{dx_j}{dx_i} = \frac{\sqrt{r^2 - (\ell_2 - x_j)^2\sin^2\psi_2}}{(\ell_2 - x_j)\sin^2\psi_2 + \cos\psi_2\sqrt{r^2 - (\ell_2 - x_j)^2\sin^2\psi_2}} = \underline{s}_{ij},$$
$$\Rightarrow \quad \begin{cases} x_i^d = \ell_1 + \frac{r\left(\underline{s}_{ij} - \cos\psi_2\right)}{\sin\psi_2\sqrt{\underline{s}_{ij}^2\sin^2\psi_2 + \left(1 - \underline{s}_{ij}\cos\psi_2\right)^2}}, \\ x_j^d = \ell_2 - \frac{r\left(1 - \underline{s}_{ij}\cos\psi_2\right)}{\sin\psi_2\sqrt{\underline{s}_{ij}^2\sin^2\psi_2 + \left(1 - \underline{s}_{ij}\cos\psi_2\right)^2}}, \end{cases}$$

where $x_i^d$ and $x_j^d$ are the corresponding arc length position of aircraft $i$ and $j$ at point $d$, respectively.

- If point $d$ is on the boundary of the first case then from equation (A.9)

$$\frac{dx_j}{x_i} = \cos\phi - \frac{(\ell_1 - x_i)\sin^2\phi}{\sqrt{r^2 - (\ell_1 - x_i)^2\sin^2\phi}} = \underline{s}_{ij},$$

$$\Rightarrow \quad \begin{cases} x_i^d = \ell_1 - \frac{r|\cos\phi - \underline{s}_{ij}|}{\sin\phi\sqrt{\sin^2\phi + (\cos\phi - \underline{s}_{ij})^2}}, \\ x_j^d = \ell_2 - \frac{r\sin^2\phi + r|\cos\phi - \underline{s}_{ij}|\cos\phi}{\sin\phi\sqrt{\sin^2\phi + (\cos\phi - \underline{s}_{ij})^2}}, \end{cases}$$

The equation of the boundary of the approximated infeasible state set that is tangent to the original set at point $d$ is

$$x_j = \underline{s}_{ij} x_i + (x_j^d - \underline{s}_{ij} x_i^d). \tag{A.32}$$

# References

[1] Y. Ye A. M. Bayen, C. J. Tomlin and J. Zhang. An approximation algorithm for scheduling aircraft with holding time. *The 43rd IEEE Conference on Decision and Control, Atlantis, Paradise Island, Bahamas*, 2004.

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

[3] Alexandre M Bayen and Claire J Tomlin. Real-time discrete control law synthesis for hybrid systems using milp: Application to congested airspace. In *American Control Conference, 2003. Proceedings of the 2003*, volume 6, pages 4620–4626. IEEE, 2003.

[4] Thomas A Becher, David R Barker, and Arthur P Smith. Near-term solution for efficient merging of aircraft on uncoordinated terminal rnav routes. In *Digital Avionics Systems Conference, 2005. DASC 2005. The 24th*, volume 1, pages 2–D. IEEE, 2005.

[5] Dimitris Bertsimas and Sarah Stock Patterson. The traffic flow management rerouting problem in air traffic control: A dynamic network flow approach. *Transportation Science*, 34(3):239–255, 2000.

[6] J. Lygeros D. Godbole S. Sastry C. Tomlin, G. Pappas. *Hybrid Systems IV, Pages 378-404.* Springer-Verlag, 1997.

[7] Yu-Heng Chang. Stochastic programming approaches to air traffic flow management under the uncertainty of weather. 2010.

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

[9] O. N. Diallo. A predictive aircraft landing speed model using neural network. *The 31st IEEE/AIAA Conference on Digital Avionics Systems*, 2012.

[10] J.-H. Oh E. Frazzoli, Z.-H. Mao and E. Feron. Resolution of conflicts involving many aircraft via semidefinite programming. *JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS,Vol. 24, No. 1*, 2001.

[11] Inseok Hwang, Hamsa Balakrishnan, and Claire Tomlin. State estimation for hybrid systems: applications to aircraft tracking. *IEE Proceedings-Control Theory and Applications*, 153(5):556–566, 2006.

[12] D. R. Isaacson and A. V. Sadovsky. Scheduling for precision air traffic operations: Problem definition and review of prior research. *(in progress).*

[13] D Karr and R Vivona. Conflict detection using variable four-dimensional uncertainty bounds to control missed alerts. In *Proc. of AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2006.

[14] Alexandre M. Bayen Kaushik Roy and Claire J. Tomlin. Polynomial time algorithms for scheduling of arrival aircraft. *AIAA Guidance, Navigation, and Control Conference and Exhibit, San Francisco, California*, 2005.

[15] Jerome Le Ny and Hamsa Balakrishnan. Feedback control of the national airspace system to mitigate weather disruptions. In *Decision and Control (CDC), 2010 49th IEEE Conference on*, pages 2055–2062. IEEE, 2010.

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

[17] Avijit Mukherjee and Mark Hansen. A dynamic rerouting model for air traffic flow management. *Transportation Research Part B: Methodological*, 43(1):159–171, 2009.

[18] T. Prevot P. U. Lee. 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.

[19] Lucia Pallottino, Eric M Feron, and Antonio Bicchi. Conflict resolution problems for air traffic management systems solved with mixed integer programming. *Intelligent Transportation Systems, IEEE Transactions on*, 3(1):3–11, 2002.

[20] A. Rezaei, A. V. Sadovsky, J. Speyer, and D. R. Isaacson. Separation-compliant speed control in terminal airspace. *AIAA Guidance, Navigation, and Control (GNC) Conference,Boston,MA*, 2013.

[21] Ali Rezaei, Alexander V. Sadovsky, Jason Speyer, and Douglas R. Isaacson. Separation-compliant in the terminal airspace, part i: speed control. *In progress.*

[22] C. Tomlin-J. Lygeros D. Godbole S. Sastry, G. Meyer and G. Pappas. Hybrid control in air traffic management systems. *Proceedings of the 34th IEEE Conference on Decision and Control, New Orleans, LA*, 1995.

[23] A. Sadovsky, H. Swenson, W. Haskell, and J. Rakas. Optimal time advance in terminal area arrivals: Throughput vs. fuel savings. In *IEEE 30th Digital Avionics Systems Conference (DASC), Seattle, WA*, 2011.

[24] A. V. Sadovsky, D. Davis, and D. R. Isaacson. 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.

[25] Alexander V Sadovsky, Damek Davis, and Douglas R Isaacson. Separation-compliant, optimal routing and control of scheduled arrivals in a terminal airspace. *Transportation Research Part C: Emerging Technologies*, 37:157–176, 2013.

[26] Joseph Sherry, Celesta Ball, and Stephen Zobell. Traffic flow management (tfm) weather rerouting. In *4th USA/Europe Air Traffic Management R&D Seminar*, 2001.

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

[28] Claire Tomlin, George J Pappas, and Shankar Sastry. Conflict resolution for air traffic management: A study in multiagent hybrid systems. *Automatic Control, IEEE Transactions on*, 43(4):509–521, 1998.