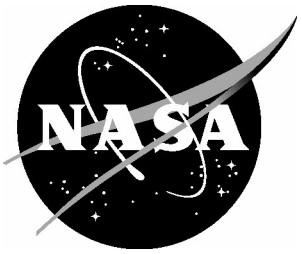
NASA/TM-2004-212999



Formal Modeling and Analysis of a Preliminary Small Aircraft Transportation System (SATS) Concept

Victor A. Carreno
Langley Research Center, Hampton, Virginia

Hanne Gottliebsen
National Institute of Aerospace, Hampton, Virginia

Ricky Butler
Langley Research Center, Hampton, Virginia

Sara Kalvala
University of Warwick, United Kingdom

March 2004

The NASA STI Program Office . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) Program Office plays a key part in helping NASA maintain this important role.

The NASA STI Program Office is operated by Langley Research Center, the lead center for NASA's scientific and technical information. The NASA STI Program Office provides access to the NASA STI Database, the largest collection of aeronautical and space science STI in the world. The Program Office is also NASA's institutional mechanism for disseminating the results of its research and development activities. These results are published by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers, but having less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services that complement the STI Program Office's diverse offerings include creating custom thesauri, building customized databases, organizing and publishing research results ... even providing videos.

For more information about the NASA STI Program Office, see the following:

- Access the NASA STI Program Home Page at ***http://www.sti.nasa.gov***

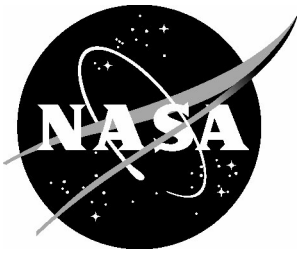
- E-mail your question via the Internet to help@sti.nasa.gov

- Fax your question to the NASA STI Help Desk at (301) 621-0134

- Phone the NASA STI Help Desk at (301) 621-0390

- Write to:
  NASA STI Help Desk
  NASA Center for AeroSpace Information
  7121 Standard Drive
  Hanover, MD 21076-1320

NASA/TM-2004-212999



Formal Modeling and Analysis of a Preliminary Small Aircraft Transportation System (SATS) Concept

Victor A. Carreno
Langley Research Center, Hampton, Virginia

Hanne Gottliebsen
National Institute of Aerospace, Hampton, Virginia

Ricky Butler
Langley Research Center, Hampton, Virginia

Sara Kalvala
University of Warwick, United Kingdom

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23681-2199

March 2004

Acknowledgment

The authors would like to thank Sheila Conway, Ken Jones, Maria Consiglio, Dan Williams, Cathy Adams, Gary Milsaps and other members of the SATS team for discussions regarding landing procedures, ATC practice, performance requirements and many other characteristics of the National Air Space. Their help was invaluable in formulating the model and parameters of the preliminary concept of operation. We want to especially thank Ken Jones for his leadership of the High Volume Operations Team and his recognition of the need for formal methods on this project.

Contents

1 Introduction

The Small Aircraft Transportation System (SATS) program aims to provide an efficient transportation alternative to commercial air and ground transportation through general aviation. The overall goals are to increase mobility, reduce door-to-door travel times, and provide air transportation to under-served markets at an affordable cost. To accomplish these goals, the SATS program is developing concepts of operations and enabling technologies. One of the concepts of operation being developed as part of the SATS program is entitled "Higher Volume Operations (HVO) at non-tower, non-radar airports during Instrument Meteorological Conditions (IMC)".

Current operations at non-tower, non-radar airports during IMC rely on procedural separation based on a method of one-in/one-out. This method results in a significant reduction in airport capacity. The SATS HVO concept will enable multiple operations to non-tower, non-radar airports during IMC. It is imperative that this concept be developed in a rigorous manner to insure that safety is not compromised. This requires that the concept undergo an extensive evaluation by both simulation and analytical methods.

In this paper, we demonstrate how the key safety properties can be established by a mathematical verification method based on formal logic and theorem proving. The system is represented in a formal mathematical language and the required properties are formulated as conjectures. A mathematical proof is constructed to show that these conjectures are indeed mathematical theorems and consequently that the modeled system has the required properties.

A preliminary concept of operation was developed prior to the completion of the first draft of the official concept of operations. This was done to give us a head start on the development of a rigorous mathematical analysis method that can be used to verify the final concept of operations in 2004. The models and proofs presented in this paper concern only this preliminary concept and not the latest SATS HVO concept documented in the summer of 2003. This preliminary concept has enabled us to develop a viable verification method and create a significant amount of reusable libraries, theories, and automated strategies that will be useful for the verification of the final concept of operation and other systems similar to this one.

The preliminary concept is described in the next section and in more details in [2]. Three basic elements of the system are modeled using the PVS formal mathematical language: (1) the airspace surrounding the airport, called the Self Controlled Area (SCA), (2) a ground based automated system called the Airport Management Module (AMM), (3) the aircraft trajectories. The safety requirement is formulated as a geometric separation property. From the models and the safety requirement, proofs are developed that support the safety claim.

2 System Description

The objective of this concept of operation is to provide an automated service which will guarantee separation assurance for aircraft operating in the airport airspace. The system must be implementable with minimal infrastructure (i.e. low cost) and should be verifiable to a high confidence level. The system consists of four primary functional parts: (1) the

Self Controlled Area (SCA), (2) the Airport Management Module (AMM), (3) on-board navigation tools, and (4) data communication. Only the first three parts of the system are modeled in this paper. The data communication is assumed to be available and error free. In future verification efforts, the communication part of the system, including errors, may be considered.

The Self Controlled Area (SCA) is a cylinder surrounding the airport facility. The approach procedure is based on a GPS "T" approach as described in [1]. Figure 1 is a top view of the T configuration. Aircraft approaching from the straight-in region are expected

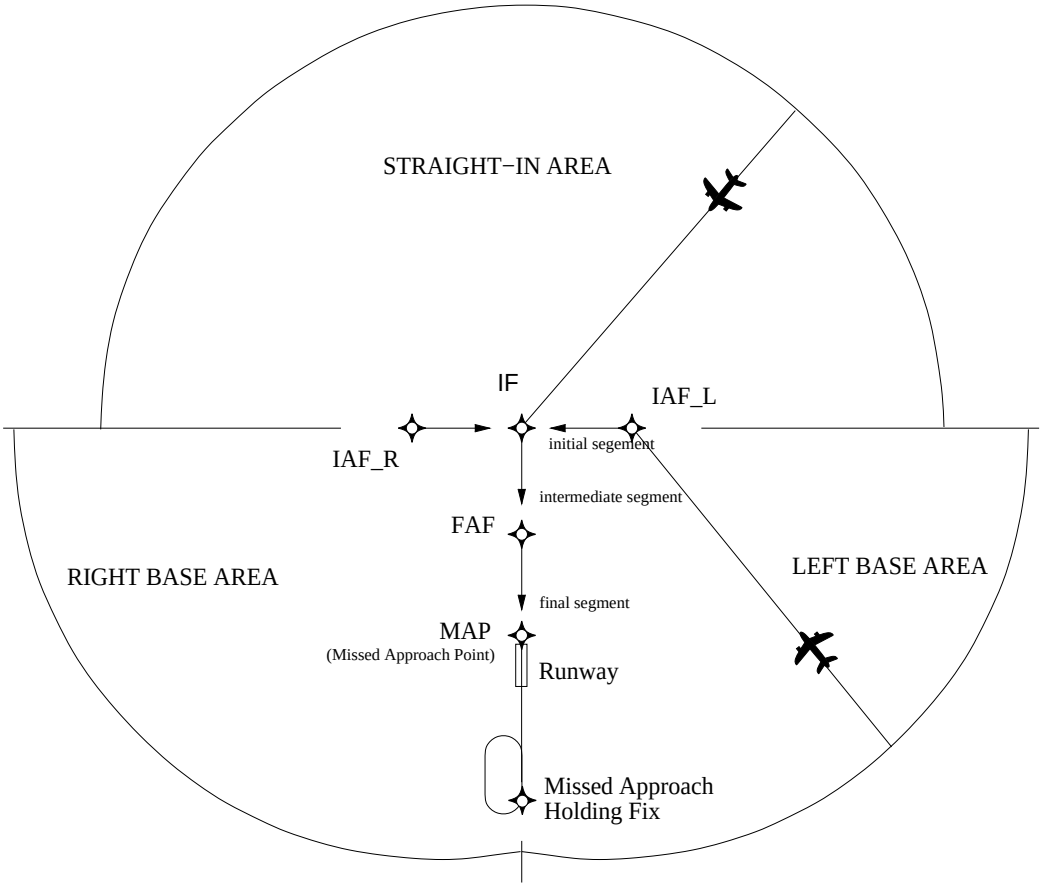


Figure 1: Basic T Design for GPS Standard Instrument Approach

to proceed directly to the Intermediate Fix (IF). Aircraft approaching from the base right or base left are expected to proceed directly to Initial Arrival Fix (IAF) right or left, respectively[1]. Aircraft entering the SCA accept responsibility for separation. That is, air traffic control services are not provided inside the SCA.

The Airport Management Module (AMM) is a centralized automated system which communicates via data link with aircraft around the airport. The AMM will typically reside on the airport grounds. The AMM serves as an arbiter and sequencer. It receives requests from

[1]The right and left regions are labeled with respect to the pilots view on final approach.

aircraft to enter the SCA and grants or denies access. Grant or denial of access is based on a time-separation criteria. When an aircraft requests entry into the SCA, the AMM checks that the requesting aircraft will be time separated, at designated points, with all other aircraft already given access[2]. To implement the time-separation scheme in a way that does not overly constrain the airspace, but achieves a simplified access criteria, the SCA was divided into 6 regions. Figure 2 shows the access regions. Aircraft in the same or adjacent regions

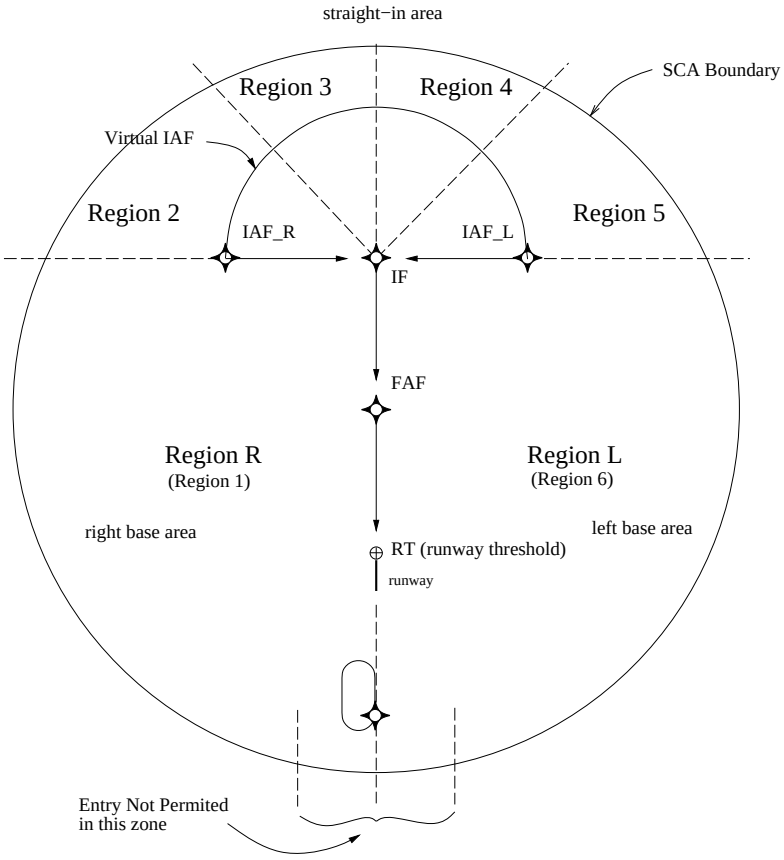


Figure 2: SCA Regions for AMM Entry Criteria

must be time separated at the following designated points:

- The SCA boundary
- The IAF (Initial Arrival Fix) or Virtual IAF
- The IF (Intermediate Fix)

[2]To insure time separation at all of these points it is necessary for the AMM to have knowledge of the nominal speed profiles of different types of aircraft and their trajectories. This means that the AMM must have a database of aircraft types and their associated descent speeds as a function of distance from the runway. The details of how the AMM performs these calculations are not included in the model. The nominal trajectories are determined by the concept itself and are included in our model.

- The FAF (Final Arrival Fix)
- The RT (Runway Threshold)

Aircraft in non adjacent regions must be time separated at the following designated points:

- The IF (Intermediate Fix)
- The FAF (Final Arrival Fix)
- The RT (Runway Threshold)

When the AMM grants access to an aircraft, it broadcasts to the aircraft an Estimated Time of Arrival (ETA). The ETAs correspond to the following designated points: SCA boundary, IAF, IF, FAF, and RT. The ETAs are based on the expected trajectory, the type of the aircraft, and the nominal speed profile of the aircraft. It is assumed that suitably-equipped SATS aircraft will have on-board navigation tools that generate heading and speed advisories (vectoring) to enable the pilot to fly the expected trajectory and meet the ETAs.

The objective of this concept is to enable a guarantee that all aircraft inside the SCA will remain separated as long as the pilots fly in accordance with the instructions given by this system. To establish that this guarantee is valid, we must show, for all possible times and all allowed aircraft trajectories that geometric separation is maintained. This verification is accomplished using a formal mathematical method, which is described in the following sections. The guarantee is elaborated as a top level safety property as follows:

Theorem 1 (Safety_Top)

$$
\begin{aligned}
&\mathsf{AMM\_properties?}(\mathsf{ac}_1, \mathsf{ac}_2) \ \wedge \\
&\mathsf{tm\_in\_SCA?}(t, \mathsf{ac}_1) \ \wedge \\
&\mathsf{tm\_in\_SCA?}(t, \mathsf{ac}_2) \ \wedge \\
&\mathsf{not\_in\_no\_enter\_zone?}(\mathsf{ac}_1, \mathsf{ac}_2) \\
&\supset \\
&\mathsf{safely\_separated?}(\overrightarrow{\mathsf{ac\_loc}}(\vec{\mathsf{ac}}_1)(t), \overrightarrow{\mathsf{ac\_loc}}(\vec{\mathsf{ac}}_2)(t))
\end{aligned}
$$

This theorem will be explained in detail in the next sections, but informally this theorem states that if the AMM protocol properties are satisfied, both aircraft are within the SCA airspace, and both of their entry points are not within the no entry zone at the base of the T, then the trajectories of both aircraft $\overrightarrow{\mathsf{ac\_loc}}(\vec{\mathsf{ac}}_1)(t)$ and $\overrightarrow{\mathsf{ac\_loc}}(\vec{\mathsf{ac}}_2)(t)$ are safely separated[3].

The predicate safely_separated? expresses the fundamental property that two points in space are sufficiently separated:

$$
\begin{aligned}
&\mathsf{safely\_separated?}(\vec{p}_1, \vec{p}_2) \equiv \\
&\qquad \mathsf{dist}(\vec{p}_1, \vec{p}_2) \geq \mathsf{sep\_min} \ \vee \\
&\qquad (\mathsf{on\_close\_corner?}(\vec{p}_1, \vec{p}_2) \ \wedge \ \mathsf{dist}(\vec{p}_1, \overrightarrow{\mathsf{ifix}}) + \mathsf{dist}(\vec{p}_2, \overrightarrow{\mathsf{ifix}}) \geq \mathsf{sep\_min})
\end{aligned}
$$

[3]We will be using the notation ac to represent the initial state record and $\vec{\mathsf{ac}}$ to represent the location of the aircraft in the initial state.

where $\vec{p}_1$ and $\vec{p}_2$ are points in a two-dimensional vector space. The dist function is defined as follows:

$$\mathsf{dist}(\vec{p}, \vec{q}) \equiv \sqrt{(\vec{p}_x - \vec{q}_x)^2 + (\vec{p}_y - \vec{q}_y)^2}$$

When two aircraft are on the T and one of them is on a base leg and the other is on final approach, linear path separation can be used rather than geometric separation. A more detailed description of on_close_corner? will be given in section 3.3. The constant sep_min is nominally 3 nautical miles.

3 Model of Self Controlled Area

The Self Controlled Area (SCA) is defined as a circle of radius SCA_radius nominally set at 12 nautical miles, as can be seen in figure 2.

3.1 Model of the T approach

The approach to the runway follows a standard T approach. The T is defined by five fixes: Runway Threshold ($\vec{\mathsf{rt}}$), Final Approach Fix ($\vec{\mathsf{faf}}$), Intermediate Fix ($\overrightarrow{\mathsf{ifix}}$), Left Initial Arrival Fix ($\overrightarrow{\mathsf{iaf\_L}}$), and Right Initial Arrival Fix ($\overrightarrow{\mathsf{iaf\_R}}$), each of which are points in 2D space, but exact coordinates are not specified, except for the $\vec{\mathsf{faf}}$ which is the origin of the coordinate system $(0, 0)$. For all fixes but the $\vec{\mathsf{faf}}$, minimal constraints are given axiomatically, defining the relative locations of the fixes:

$$\begin{aligned}
&\overrightarrow{\mathsf{iaf_L}}_y = \overrightarrow{\mathsf{ifix}}_y \ \wedge \\
&\overrightarrow{\mathsf{iaf_R}}_x = -\ \overrightarrow{\mathsf{iaf_L}}_x \ \wedge \\
&\overrightarrow{\mathsf{iaf_R}}_y = \overrightarrow{\mathsf{ifix}}_y \ \wedge \\
&\overrightarrow{\mathsf{iaf_L}}_x + \overrightarrow{\mathsf{ifix}}_y < \mathsf{SCA_radius}
\end{aligned}$$

From this we see, that the T is assumed symmetric, that is the two initial arrival fixes ($\overrightarrow{\mathsf{iaf\_L}}$ and $\overrightarrow{\mathsf{iaf\_R}}$) are at equal distances from the intermediate fix ($\overrightarrow{\mathsf{ifix}}$). The last property ensures that virtual initial fixes as mentioned below are within the SCA.

The following additional properties are derived from the type constraints of the fixes:

$$\begin{aligned}
&\vec{\mathsf{rt}}_x = 0 \wedge \ \vec{\mathsf{rt}}_y < 0 \wedge \ \vec{\mathsf{rt}}_y > -\mathsf{SCA_radius} \ \wedge \\
&\overrightarrow{\mathsf{ifix}}_x = 0 \wedge \ \overrightarrow{\mathsf{ifix}}_y > 0 \wedge \ \overrightarrow{\mathsf{ifix}}_y < \mathsf{SCA_radius} \ \wedge \\
&\overrightarrow{\mathsf{iaf_L}}_x > 0 \wedge \ \overrightarrow{\mathsf{iaf_L}}_x < \mathsf{SCA_radius} \wedge \ \overrightarrow{\mathsf{iaf_L}}_y = \overrightarrow{\mathsf{ifix}}_y
\end{aligned}$$

This orients the T in the coordinate system, so that the $\vec{\mathsf{rt}}$ is directly below the $\vec{\mathsf{faf}}$, and the initial arrival fixes ($\overrightarrow{\mathsf{iaf\_L}}$ and $\overrightarrow{\mathsf{iaf\_R}}$) are exactly level with the $\overrightarrow{\mathsf{ifix}}$.

The following distances are defined for convenience:

$$\mathsf{iaf2if} \equiv \mathsf{dist}(\overrightarrow{\mathsf{ifix}}, \overrightarrow{\mathsf{iaf\_L}})$$
$$\mathsf{if2faf} \equiv \mathsf{dist}(\vec{\mathsf{faf}}, \overrightarrow{\mathsf{ifix}})$$
$$\mathsf{faf2rt} \equiv \mathsf{dist}(\vec{\mathsf{rt}}, \vec{\mathsf{faf}})$$
$$\mathsf{d\_iaf} \equiv \mathsf{iaf2if} + \mathsf{if2faf} + \mathsf{faf2rt}$$

The distance d_iaf is the distance measured along the T from an initial arrival fix to the runway threshold.

3.2 Division of the SCA

The airspace is decomposed into disjoint regions, as seen in figure 2:

$$\begin{aligned}
\mathsf{regionR?}(\vec{p}) &\equiv (\vec{p}_x < 0 \ \wedge \ \vec{p}_y < \overrightarrow{\mathsf{ifix}}_y) \ \vee \ (\vec{p}_x = 0 \ \wedge \ \vec{p}_y < \vec{\mathsf{rt}}_y) \ \vee \ (\vec{p}_x < \overrightarrow{\mathsf{iaf_R}}_x \ \wedge \vec{p}_y = \overrightarrow{\mathsf{ifix}}_y) \\
\mathsf{regionL?}(\vec{p}) &\equiv (\vec{p}_x > 0 \ \wedge \ \vec{p}_y < \overrightarrow{\mathsf{ifix}}_y) \ \vee \ (\vec{p}_x > \overrightarrow{\mathsf{iaf_L}}_x \ \wedge \vec{p}_y = \overrightarrow{\mathsf{ifix}}_y) \\
\mathsf{regionM?}(\vec{p}) &\equiv \vec{p}_y > \overrightarrow{\mathsf{ifix}}_y \\
\mathsf{baselegR?}(\vec{p}) &\equiv \overrightarrow{\mathsf{iaf_R}}_x \leq \vec{p}_x \ \wedge \ \vec{p}_x < \overrightarrow{\mathsf{ifix}}_x \ \wedge \vec{p}_y = \overrightarrow{\mathsf{ifix}}_y \\
\mathsf{baselegL?}(\vec{p}) &\equiv \overrightarrow{\mathsf{ifix}}_x < \vec{p}_x \ \wedge \ \vec{p}_x \leq \overrightarrow{\mathsf{iaf_L}}_x \ \wedge \vec{p}_y = \overrightarrow{\mathsf{ifix}}_y \\
\mathsf{final_1?}(\vec{p}) &\equiv \vec{p}_x = \vec{\mathsf{faf}}_x \ \wedge \ \vec{\mathsf{faf}}_y < \vec{p}_y \ \wedge \ \vec{p}_y \leq \overrightarrow{\mathsf{ifix}}_y \\
\mathsf{final_2?}(\vec{p}) &\equiv \vec{p}_x = \vec{\mathsf{faf}}_x \ \wedge \ \vec{\mathsf{rt}}_y < \vec{p}_y \ \wedge \ \vec{p}_y \leq \vec{\mathsf{faf}}_y \\
\mathsf{runway?}(\vec{p}) &\equiv \vec{p}_x = \vec{\mathsf{faf}}_x \ \wedge \ \vec{\mathsf{rt}}_y = \vec{p}_y
\end{aligned}$$

The first three regions divide up the space outside the T based on which initial arrival fix would be used for aircraft in that position. Thus an aircraft in region R would fly to the $\overrightarrow{\mathsf{iaf\_R}}$. Region M is the area with y-coordinates higher than that of $\overrightarrow{\mathsf{ifix}}$, that is the area above the T in figure 2, region M is also called the *straight-in area*. Each of these three regions extend into the airspace outside the SCA. The last five regions together make up the T. The baselegs (*initial segments* in figure 1) are the paths between the initial arrival fixes ($\overrightarrow{\mathsf{iaf\_R}}$ and $\overrightarrow{\mathsf{iaf\_L}}$) and the $\overrightarrow{\mathsf{ifix}}$. Then between the $\overrightarrow{\mathsf{ifix}}$ and the $\vec{\mathsf{faf}}$ is the first part of the final approach (final_1), and from $\vec{\mathsf{faf}}$ to $\vec{\mathsf{rt}}$ is the second part (final_2). Finally, the runway is given as a single point.

To further facilitate our algorithm, we divide region M into 4 sub–regions: Regions two through five:

$$\begin{aligned}
\mathsf{region2?}(\vec{p}) &\equiv \vec{p}_y > \overrightarrow{\mathsf{ifix}}_y \ \wedge \vec{p}_x < 0 \ \wedge \ \vec{p}_y - \overrightarrow{\mathsf{ifix}}_y \leq -\vec{p}_x \\
\mathsf{region3?}(\vec{p}) &\equiv \vec{p}_y > \overrightarrow{\mathsf{ifix}}_y \ \wedge \vec{p}_x \leq 0 \ \wedge \ \vec{p}_y - \overrightarrow{\mathsf{ifix}}_y > -\vec{p}_x \\
\mathsf{region4?}(\vec{p}) &\equiv \vec{p}_y > \overrightarrow{\mathsf{ifix}}_y \ \wedge \vec{p}_x > 0 \ \wedge \ \vec{p}_y - \overrightarrow{\mathsf{ifix}}_y > \vec{p}_x \\
\mathsf{region5?}(\vec{p}) &\equiv \vec{p}_y > \overrightarrow{\mathsf{ifix}}_y \ \wedge \vec{p}_x \geq 0 \ \wedge \ \vec{p}_y - \overrightarrow{\mathsf{ifix}}_y \leq \vec{p}_x
\end{aligned}$$

Each of these regions cover a 45° slice of region M. The dividing lines are assigned as follows: The dividing line between regions 2 and 3 belongs to region 2, the dividing line between regions 3 and 4 belongs to region 3 and the dividing line between regions 4 and 5 belongs to region 5.

3.3 Predicates On Points

It is convenient to be able to express whether a point is within a certain group of regions. This is easily accomplished by defining some additional predicates as conjunctions of the basic regions.

First we have predicates on_baseleg? and on_final?, each of which combines two of the regions defined above. on_T? then combines these two new predicates.

$$\begin{aligned} \text{on\_baseleg?}(\vec{p}) &\equiv \text{baselegR?}(\vec{p}) \vee \text{baselegL?}(\vec{p}) \\ \text{on\_final?}(\vec{p}) &\equiv \text{final\_1?}(\vec{p}) \vee \text{final\_2?}(\vec{p}) \vee \text{runway?}(\vec{p}) \\ \text{on\_T?}(\vec{p}) &\equiv \text{on\_baseleg?}(\vec{p}) \vee \text{on\_final?}(\vec{p}) \end{aligned}$$

Given two points in the SCA, we can determine if they are on different (opposite) baselegs:

$$\begin{aligned} \text{opposite\_baselegs?}(\vec{p}_1, \vec{p}_2) \equiv\ & (\text{baselegR?}(\vec{p}_1) \wedge \text{baselegL?}(\vec{p}_2)) \vee \\ & (\text{baselegL?}(\vec{p}_1) \wedge \text{baselegR?}(\vec{p}_2)) \end{aligned}$$

The minimal separation safety criteria is relaxed a little when both aircraft are on the T in that the separation may be along the T. In some cases this is the same as the geometrical distance, however in the instance where one aircraft is on a baseleg and the other is on final, the distance along the flight path is shorter than the geometrical distance. Thus it is useful to be able to distinguish this situation, for which we define a predicate on_close_corner?:

$$\begin{aligned} \text{on\_close\_corner?}(\vec{p}_1, \vec{p}_2) \equiv\ & (\text{on\_baseleg?}(\vec{p}_1) \wedge \text{on\_final?}(\vec{p}_2)) \vee \\ & (\text{on\_baseleg?}(\vec{p}_2) \wedge \text{on\_final?}(\vec{p}_1)) \end{aligned}$$

The timing comparisons for aircraft are dependent on whether the aircraft are in the same region, in adjacent regions or in non-adjacent regions.

$$\begin{aligned} &\text{same\_region?}(\vec{p}_1, \vec{p}_2) \equiv \\ &\quad (\text{regionR?}(\vec{p}_1) \wedge \text{regionR?}(\vec{p}_2)) \vee (\text{regionL?}(\vec{p}_1) \wedge \text{regionL?}(\vec{p}_2)) \vee \\ &\quad (\text{region2?}(\vec{p}_1) \wedge \text{region2?}(\vec{p}_2)) \vee (\text{region3?}(\vec{p}_1) \wedge \text{region3?}(\vec{p}_2)) \vee \\ &\quad (\text{region4?}(\vec{p}_1) \wedge \text{region4?}(\vec{p}_2)) \vee (\text{region5?}(\vec{p}_1) \wedge \text{region5?}(\vec{p}_2)) \\ &\text{adjacent\_region?}(\vec{p}_1, \vec{p}_2) \equiv \\ &\quad (\text{regionR?}(\vec{p}_1) \wedge \text{region2?}(\vec{p}_2)) \vee (\text{regionR?}(\vec{p}_2) \wedge \text{region2?}(\vec{p}_1)) \vee \\ &\quad (\text{regionL?}(\vec{p}_1) \wedge \text{region5?}(\vec{p}_2)) \vee (\text{regionL?}(\vec{p}_2) \wedge \text{region5?}(\vec{p}_1)) \vee \\ &\quad (\text{region2?}(\vec{p}_1) \wedge \text{region3?}(\vec{p}_2)) \vee (\text{region2?}(\vec{p}_2) \wedge \text{region3?}(\vec{p}_1)) \vee \\ &\quad (\text{region3?}(\vec{p}_1) \wedge \text{region4?}(\vec{p}_2)) \vee (\text{region3?}(\vec{p}_2) \wedge \text{region4?}(\vec{p}_1)) \vee \\ &\quad (\text{region4?}(\vec{p}_1) \wedge \text{region5?}(\vec{p}_2)) \vee (\text{region4?}(\vec{p}_2) \wedge \text{region5?}(\vec{p}_1)) \end{aligned}$$

It is worth noting that although regions L and R are adjacent in a geometrical sense, they are not defined as adjacent in this formalization, since aircraft in regions R and L fly toward two different IAFs ($\overrightarrow{\mathsf{iaf\_R}}$ and $\overrightarrow{\mathsf{iaf\_L}}$).

3.4 Determining the Correct Arrival Fix

Based on the position of an aircraft outside the SCA, the appropriate initial arrival fix is uniquely determined. If an aircraft is in region R (region L), the initial arrival fix is $\overrightarrow{\mathsf{iaf\_R}}$ ($\overrightarrow{\mathsf{iaf\_L}}$), however if the aircraft is in region M, it heads directly to the $\overrightarrow{\mathsf{ifix}}$. Nevertheless it is useful to define virtual initial fixes for aircraft entering through region M:

$$\overrightarrow{\mathsf{vir\_iaf}}\ (\vec{p_i}) \equiv \vec{p_i} + (1 - \frac{\overrightarrow{\mathsf{iaf\_L}}_x}{\mathsf{dist}(\vec{p_i}, \overrightarrow{\mathsf{ifix}})})(\overrightarrow{\mathsf{ifix}} - \vec{p_i})$$

where $\vec{p_i}$ is an initial point in region M.

Thus, given a position in the SCA (or indeed in the airspace outside) one can compute the fix that the aircraft proceeds toward as follows:

$$\begin{aligned}\overrightarrow{\mathsf{which\_iaf}}(\vec{p_i} : \mathit{init\_point}) \equiv\ & \mathsf{IF}\ \mathsf{regionR?}(\vec{p_i})\ \mathsf{THEN}\ \overrightarrow{\mathsf{iaf\_R}} \\ & \mathsf{ELSIF}\ \mathsf{regionL?}(\vec{p_i})\ \mathsf{THEN}\ \overrightarrow{\mathsf{iaf\_L}} \\ & \mathsf{ELSE}\ \overrightarrow{\mathsf{vir\_iaf}}\ (\vec{p_i}) \\ & \mathsf{ENDIF}\end{aligned}$$

Although we use the virtual initial arrival fixes for aircraft entering through region M, we also often just assume that they go straight to the $\overrightarrow{\mathsf{ifix}}$. However, since the virtual initial arrival fixes are on the straight line between the entry point for the aircraft and the $\overrightarrow{\mathsf{ifix}}$, this does not change our assumptions on the flight path.

4 Model of Aircraft Trajectory

Fundamental to the specification of the SATS system is the delineation of the trajectories of aircraft in the SATS airspace. These trajectories are modeled using a function $\overrightarrow{\mathsf{ac\_loc}}$ of time:

$$
\begin{array}{l}
\overrightarrow{\mathsf{ac\_loc}}(\vec{\mathsf{ac}})(t) \equiv \\
\quad \mathsf{LET}\ t_{\mathsf{iaf}} = \mathsf{time\_at\_iaf(ac)}, \\
\qquad \overrightarrow{\mathsf{wh\_fix}} = \overrightarrow{\mathsf{which\_iaf}}(\vec{\mathsf{ac}}), \\
\qquad t_{\mathsf{ent}} = \mathsf{entry\_time(ac)}\ \ \mathsf{IN} \\
\quad \mathsf{IF}\ t < t_{\mathsf{iaf}}\ \ \mathsf{THEN}\ \ \vec{\mathsf{ac}} + (t - t_{\mathsf{ent}}) * \overrightarrow{\mathsf{vel\_from\_spd}}(\vec{\mathsf{ac}}, \overrightarrow{\mathsf{wh\_fix}}, \mathsf{ac.gs}) \\
\quad \mathsf{ELSIF}\ t \geq \mathsf{time\_to\_rt(ac)}\ \ \mathsf{THEN}\ \ (\mathsf{rt}_x,\ \mathsf{rt}_y) \\
\quad \mathsf{ELSE} \\
\qquad \mathsf{IF}\ \ t \geq \mathsf{time\_to\_if(ac)}\ \ \mathsf{THEN} \\
\qquad\quad (\overrightarrow{\mathsf{ifix}}_x, \overrightarrow{\mathsf{ifix}}_y - \mathsf{dist\_gone(ac)}(t) - \mathsf{iaf2if}) \\
\qquad \mathsf{ELSIF}\ \ \mathsf{regionR?}(\vec{\mathsf{ac}})\ \vee\ \mathsf{regionL?}(\vec{\mathsf{ac}})\ \ \mathsf{THEN} \\
\qquad\quad \mathsf{IF}\ \ \overrightarrow{\mathsf{wh\_fix}} = \overrightarrow{\mathsf{iaf\_R}}\ \ \mathsf{THEN}\ \overrightarrow{\mathsf{loc\_on\_legR}}(\mathsf{dist\_gone(ac)}(t)) \\
\qquad\qquad\qquad\qquad \mathsf{ELSE}\ \overrightarrow{\mathsf{loc\_on\_legL}}(\mathsf{dist\_gone(ac)}(t)) \\
\qquad\quad \mathsf{ENDIF} \\
\qquad \mathsf{ELSE} \\
\qquad\quad \overrightarrow{\mathsf{wh\_fix}} + \dfrac{\mathsf{dist\_gone(ac)}(t)}{\mathsf{iaf2if}}(\overrightarrow{\mathsf{ifix}} - \overrightarrow{\mathsf{wh\_fix}}) \\
\qquad \mathsf{ENDIF} \\
\quad \mathsf{ENDIF}
\end{array}
$$

where $\vec{\mathsf{ac}}$ is the initial location of the aircraft when it enters the SATS airspace. This function decomposes the calculation of the position of the aircraft based upon time. The key times are

time_at_iaf(ac):	time aircraft arrives at the initial approach fix
	$= \mathsf{entry\_time(ac)} + \mathsf{dist}(\vec{\mathsf{ac}}, \overrightarrow{\mathsf{which\_iaf}}(\vec{\mathsf{ac}}))/\mathsf{ac.gs}$
time_to_if(ac):	time aircraft arrives at the initial fix
time_to_rt(ac):	time aircraft arrives at the runway threshold

The time that an aircraft enters the SATS airspace is represented by an uninterpreted function, entry_time(ac).

4.1 Before time_at_iaf(ac)

Prior to time_at_iaf(ac), the aircraft travels at a constant velocity. It was convenient to define the aircraft trajectory using a line in 2D space. The traditional way to define a line in 2D space is by specifying two distinct points, $\vec{p_0}$ and $\vec{p_1}$, on it. But a line can also be defined by a point and a direction vector. Furthermore, we can also add dynamics to our line using an

initial point $\vec{p_0}$ and a velocity vector $\vec{v}$ as follows:

$$\vec{p_0} + t\vec{v}$$

which designates the location of a moving particle at time t. Thus if $\vec{\mathsf{ac}}$ is the position of the aircraft when it enters the SATS airspace, it position up to the IAF can be calculated as follows:

$$\vec{\mathsf{ac}} + (t - t_{\mathsf{ent}}) * \mathsf{vel\_from\_spd}(\vec{\mathsf{ac}}, \overrightarrow{\mathsf{wh\_fix}}, \mathsf{ac.gs})$$

where t_{ent} is the entry time and $\mathsf{vel\_from\_spd}(\vec{\mathsf{ac}}, \overrightarrow{\mathsf{wh\_fix}}, \mathsf{ac.gs})$ is the constant velocity of the vehicle. Note that this velocity vector is computed from the initial point, the final point and the speed as follows:

$$\begin{aligned} \mathsf{vel\_from\_spd}(\vec{p}_1, \vec{p}_2, s) \quad \equiv \quad & \mathsf{IF}\ \vec{p}_1 = \vec{p}_2\ \mathsf{THEN\ zero} \\ & \mathsf{ELSE}\ \frac{s}{\mathsf{dist}(\vec{p}_1, \vec{p}_2)}(\vec{p}_2 - \vec{p}_1) \\ & \mathsf{ENDIF} \end{aligned}$$

4.2 Between time_at_iaf(ac) and time_to_if(ac)

Once the aircraft reaches an IAF or a virtual IAF it begins to decrease its speed in accordance with a speed profile that is a function of remaining distance to the runway threshold. Therefore a function $\mathsf{dist\_gone(ac)}(t)$ is needed to compute the relative distance. Details are provided in section 4.4. The position on the T at time t depends upon which of the two IAFs or virtual IAFs the aircraft passed through. The $\overrightarrow{\mathsf{ac\_loc}}$ function first tests to see whether the aircraft is currently at an IAF (i.e. entered from region R or region L) or a virtual IAF. If it is at an IAF then the position is calculated as follows:

$$\begin{aligned} \mathsf{IF}\ \overrightarrow{\mathsf{wh\_fix}} = \overrightarrow{\mathsf{iaf\_R}}\ \ \mathsf{THEN}\ & \overrightarrow{\mathsf{loc\_on\_legR}}(\mathsf{dist\_gone(ac)}(t)) \\ \mathsf{ELSE}\ & \overrightarrow{\mathsf{loc\_on\_legL}}(\mathsf{dist\_gone(ac)}(t)) \\ \mathsf{ENDIF} & \end{aligned}$$

where the subfunctions are defined as follows:

$$\begin{aligned} \overrightarrow{\mathsf{loc\_on\_legR}}(l) \equiv\ & \mathsf{IF}\ l < \mathsf{iaf2if}\ \mathsf{THEN}\ (\overrightarrow{\mathsf{iaf\_R}}_x + l,\ \ \overrightarrow{\mathsf{iaf\_R}}_y) \\ & \mathsf{ELSE}\ (\overrightarrow{\mathsf{ifix}}_x,\ \ \overrightarrow{\mathsf{ifix}}_y - (l - \mathsf{iaf2if})) \\ & \mathsf{ENDIF} \end{aligned}$$

$$\begin{aligned} \overrightarrow{\mathsf{loc\_on\_legL}}(l) \equiv\ & \mathsf{IF}\ l < \mathsf{iaf2if}\ \mathsf{THEN}\ (\overrightarrow{\mathsf{iaf\_L}}_x - l,\ \ \overrightarrow{\mathsf{iaf\_R}}_y) \\ & \mathsf{ELSE}\ (\overrightarrow{\mathsf{ifix}}_x,\ \ \overrightarrow{\mathsf{ifix}}_y - (l - \mathsf{iaf2if})) \\ & \mathsf{ENDIF} \end{aligned}$$

If the aircraft entered the SCA through region M and has already passed through the virtual IAF, the position is calculated as follows:

$$\overrightarrow{\mathsf{wh\_fix}} + \frac{\mathsf{dist\_gone(ac)}(t)}{\mathsf{iaf2if}}(\overrightarrow{\mathsf{ifix}} - \overrightarrow{\mathsf{wh\_fix}})$$

4.3 After time_to_if(ac)

If the time is after time_to_rt(ac) then the function returns the location of the runway threshold $(\mathsf{rt}_x,\ \mathsf{rt}_y)$. Otherwise, the aircraft is on the final approach. The calculated location is:

$$(\overrightarrow{\mathsf{ifix}_x},\ \ \overrightarrow{\mathsf{ifix}_y} - \mathsf{dist\_gone(ac)}(t) - \mathsf{iaf2if})$$

4.4 Calculation of dist_gone(ac)(t)

The speed of the aircraft after it reaches an IAF or a virtual IAF is defined by a speed profile determined by its aircraft type. This speed profile is a function of remaining distance to the runway threshold. For example, the speed profile for the Cessna 172 is:

$$\begin{aligned}\mathsf{speed\_profile\_c172}(d_r) \equiv\ & \mathsf{IF}\ d_r \le 1\ \ \mathsf{THEN}\ \ 90 + 25(d_r - 1)\\ & \mathsf{ELSIF}\ \ d_r \le 5\ \ \mathsf{THEN}\ \ 90\\ & \mathsf{ELSIF}\ \ d_r \le 7\ \ \mathsf{THEN}\ \ 120 + \frac{(120-90)}{2}(d_r - 7)\\ & \mathsf{ELSE}\ \ 120\\ & \mathsf{ENDIF}\end{aligned}$$

Since this function is continuous, we can define a time-to-point function as the integral, with respect to distance, of one over the speed profile, plus an absolute time constant A_t:

$$\mathsf{tm2pt(ac)}(l) = \int_0^l \frac{1}{\mathsf{speed\_profile(ac,\, d\_iaf} - l)}dl\ + A_t$$

where A_t is the time at which the aircraft crosses the IAF, l is defined as the distance traveled after crossing the IAF, d_iaf is the path distance from the IAF to the runway threshold, and the argument ac in the speed profile function supplies the type of aircraft. Therefore, $\mathsf{tm2pt(ac)}(0) = \mathsf{time\_at\_iaf(acnv)}$. Note that $\mathsf{d\_iaf} - l$ is the remaining distance to the runway threshold.

Since the speed profile is continuous and positive, the function tm2pt is continuous and increasing, so we can define an inverse function as follows:

$$t = \mathsf{tm2pt(ac)}(l)\ \Leftrightarrow\ \mathsf{dist\_gone(ac)}(t) = l$$

It is important to keep in mind that dist_gone returns the relative distance traveled from the IAF but takes as an argument absolute time.

Lemma 1 (derivative_relation)

$$\begin{aligned}\mathsf{derivative\_relation}: \ \mathsf{LEMMA}\ (\forall l : \frac{d\ \mathsf{tm2pt}(\mathsf{ac}_1)(l)}{dl} \geq \frac{d\ \mathsf{tm2pt}(\mathsf{ac}_2)(l)}{dl}) \ \vee \\ (\forall l : \frac{d\ \mathsf{tm2pt}(\mathsf{ac}_2)(l)}{dl} \geq \frac{d\ \mathsf{tm2pt}(\mathsf{ac}_1)(l)}{dl})\end{aligned} \tag{1}$$

Proof. This is established for all combinations of aircraft speed profiles by a case split on each of the speed profile functions. This is a key property that we rely on the establish separation on the T.
□

5 AMM Requirements Model

In this section, the requirements for the Aircraft Management Module (AMM) are described. These requirements basically define an abstract time separation protocol, which do not specify any of the details of an implementation. They are intrinsic to the concept itself or are a product of the formal proof process (i.e. they were added in order to complete a proof). Properties that were needed in order to establish the separation lemmas were collected under a predicate named AMM_properties? defined as follows:

$$\begin{aligned}\mathsf{AMM\_properties?}(\mathsf{ac}_1, \mathsf{ac}_2) \equiv\ & \mathsf{AMM\_PP2?}(\mathsf{ac}_1, \mathsf{ac}_2) \ \wedge \\ & \mathsf{time\_sep\_prop} \ \wedge \\ & \mathsf{entry\_time}(\mathsf{ac}_2) > \mathsf{entry\_time}(\mathsf{ac}_1) \ \wedge \\ & \mathsf{iaf\_L\_gt\_sep\_min} \ \wedge \\ & \mathsf{init\_sep\_prop}(\mathsf{ac}_1, \mathsf{ac}_2)\end{aligned}$$

We will discuss each of these conjuncts in the order that they appear. First the predicate AMM_PP2? is the abstract representation of the AMM timing protocol and defined as follows:

$$\begin{aligned}\mathsf{AMM\_PP2?}(\mathsf{ac}_1, \mathsf{ac}_2) \equiv\ & \mathsf{time\_separation\_at\_rt?}(\mathsf{ac}_1, \mathsf{ac}_2) \ \wedge \\ & \mathsf{time\_separation\_at\_faf?}(\mathsf{ac}_1, \mathsf{ac}_2) \ \wedge \\ & \mathsf{time\_separation\_at\_if?}(\mathsf{ac}_1, \mathsf{ac}_2) \ \wedge \\ & (((\mathsf{same\_region?}(\vec{\mathsf{ac}}_1, \vec{\mathsf{ac}}_2) \\ & \ \vee \mathsf{adjacent\_region?}(\vec{\mathsf{ac}}_1, \vec{\mathsf{ac}}_2))) \\ & \mathsf{IMPLIES} \\ & \quad \mathsf{time\_separation\_at\_iaf?}(\mathsf{ac}_1, \mathsf{ac}_2) \ \wedge \\ & \quad \mathsf{time\_separation\_at\_entry?}(\mathsf{ac}_1, \mathsf{ac}_2))\end{aligned}$$

The first three constructs specify that there is time separation at the runway threshold (rt), the final approach fix (faf) , and the initial fix (if). The next implication states that if the two aircraft are in the same region or in adjacent regions, then there are two additional

timing constraints, namely, that there is time separation at the IAFs and at the point of entry. The time_separation predicates are defined as follows:

$$\mathsf{time\_separation\_at\_entry?}(\mathsf{ac}_1, \mathsf{ac}_2) \equiv \mathsf{entry\_time}(\mathsf{ac}_2) - \mathsf{entry\_time}(\mathsf{ac}_1) \geq \mathsf{time\_sep}$$

$$\mathsf{time\_separation\_at\_iaf?}(\mathsf{ac}_1, \mathsf{ac}_2) \equiv \mathsf{tm2pt}(\mathsf{ac}_2)(0) - \mathsf{tm2pt}(\mathsf{ac}_1)(0) \geq \mathsf{time\_sep}$$

$$\mathsf{time\_separation\_at\_if?}(\mathsf{ac}_1, \mathsf{ac}_2) \equiv \mathsf{tm2pt}(\mathsf{ac}_2)(\mathsf{iaf2if}) - \mathsf{tm2pt}(\mathsf{ac}_1)(\mathsf{iaf2if}) \geq \mathsf{time\_sep}$$

$$\mathsf{time\_separation\_at\_faf?}(\mathsf{ac}_1, \mathsf{ac}_2) \equiv \mathsf{tm2pt}(\mathsf{ac}_2)(\mathsf{iaf2if} + \mathsf{if2faf}) - \mathsf{tm2pt}(\mathsf{ac}_1)(\mathsf{iaf2if} + \mathsf{if2faf}) \geq \mathsf{time\_sep}$$

$$\mathsf{time\_separation\_at\_rt?}(\mathsf{ac}_1, \mathsf{ac}_2) \equiv \mathsf{tm2pt}(\mathsf{ac}_2)(\mathsf{d\_iaf}) - \mathsf{tm2pt}(\mathsf{ac}_1)(\mathsf{d\_iaf}) \geq \mathsf{time\_sep}$$

The AMM implementation will have to perform many calculations involving airport geometry, aircraft trajectories and speed profiles, and other external factors such as wind, communications delay, and pilot error in order to calculate time delays that will meet these requirements. It is important that the refinement of these AMM requirements into executable code be carried out in a rigourous manner and formally verified as well as the overall system concept. It is also important that the AMM code be implemented on a fault-tolerant computing platform, because the reliability requirements will be very high. This was not attempted in this effort, but will necessarily be a part of our future efforts on the evolving SATS concept of operation.

The second conjunct in the predicate AMM_properties? is time_sep_prop which is defined as follows:

$$\mathsf{time\_sep\_prop} \equiv \mathsf{time\_sep} * \mathsf{min\_speed} \geq \mathsf{sep\_min}$$

This constraint essentially defines time_sep in terms of the the minimum geometric separation (sep_min) and the speed of the slowest possible aircraft (min_speed). The third conjunct in the predicate AMM_properties? is just a naming convention establishing that the first aircraft to enter the SATS airspace is labeled as 1 and the second is labeled 2:

$$\mathsf{entry\_time}(\mathsf{ac}_2) > \mathsf{entry\_time}(\mathsf{ac}_1)$$

The fourth conjunct in the predicate AMM_properties? is iaf_L_gt_sep_min defined as follows:

$$\mathsf{iaf\_L\_gt\_sep\_min} \equiv \mathsf{iaf\_L}_x \geq \mathsf{sep\_min}$$

This is a restriction on size of sep_min compared to the width of the T (or vice versa). Finally, the fifth conjunct in the predicate AMM_properties? is init_sep_prop which is defined as:

$$\begin{aligned}\mathsf{init\_sep\_prop}(\mathsf{ac}_1, \mathsf{ac}_2) \equiv\ & \\ & \mathsf{same\_region?}(\vec{\mathsf{ac}}_1, \vec{\mathsf{ac}}_2) \ \supset \\ & Z * \mathsf{min\_speed} \geq \mathsf{dist}(\vec{\mathsf{ac}}_1, \overrightarrow{IAF}) - \mathsf{dist}(\vec{\mathsf{ac}}_2, \overrightarrow{IAF})) + \mathsf{sep\_min}\end{aligned}$$

where $Z \equiv \mathsf{entry\_time}(\mathsf{ac}_2) - \mathsf{entry\_time}(\mathsf{ac}_1)$ and $\overrightarrow{IAF}$ is the common initial approach fix. This last conjunct was added to facilitate a proof. It adds an additional restriction to the timing protocol. Simulation runs with this additional restriction shows that the performance penalty in terms of airport operational capacity is very small. This conjunct requires that the AMM determine the distances to the IAF when the two aircraft are in the same region.

6 Safety Property

In the introduction, we stated that the top level theorem (1) is:

$$\begin{array}{l}
\mathsf{AMM_properties?}(\mathsf{ac}_1, \mathsf{ac}_2) \ \wedge \\
\mathsf{tm_in_SCA?}(t, \mathsf{ac}_1) \ \wedge \\
\mathsf{tm_in_SCA?}(t, \mathsf{ac}_2) \ \wedge \\
\mathsf{not_in_no_enter_zone?}(\vec{\mathsf{ac}}_1, \vec{\mathsf{ac}}_2) \\
\ \supset \\
\mathsf{safely_separated?}(\overrightarrow{\mathsf{ac_loc}}(\vec{\mathsf{ac}}_1)(t), \overrightarrow{\mathsf{ac_loc}}(\vec{\mathsf{ac}}_2)(t))
\end{array}$$

$\mathsf{AMM\_properties?}(\mathsf{ac}_1, \mathsf{ac}_2)$ was explained in the previous section, $\mathsf{tm\_in\_SCA?}(t, \mathsf{ac})$ is a predicate use to ensure that the aircraft is inside the SCA, and $\mathsf{not\_in\_no\_enter\_zone?}$ is a predicate which excludes entry near the base of the T. In figure 2 this zone is designated by the phrase "Entry Not Permitted in this zone". In this section we will describe each of the predicates $\mathsf{tm\_in\_SCA?}$, $\mathsf{not\_in\_no\_enter\_zone?}$ and $\mathsf{safely\_separated?}$.

6.1 Timing Predicates

The predicate $\mathsf{tm\_in\_SCA?}$ is defined by:

$$\mathsf{tm\_in\_SCA?}(t, \mathsf{ac}) \ \equiv \ \mathsf{tm\_bef\_T?}(t, \mathsf{ac}) \ \vee \ \mathsf{tm\_on\_T?}(t, \mathsf{ac})$$

and determines if an aircraft is within the SCA at time t.

The predicates $\mathsf{tm\_bef\_T?}$ and $\mathsf{tm\_on\_T?}$ are defined as follows:

$$\begin{array}{rcl}
\mathsf{tm_bef_T?}(t, \mathsf{ac}) & \equiv & \mathsf{IF\ regionM?}(\vec{\mathsf{ac}})\ \mathsf{THEN}\ t \geq \mathsf{entry_time(ac)} \ \wedge \ t < \mathsf{tm2pt(ac)(iaf2if)} \\
& & \mathsf{ELSE}\ t \geq \mathsf{entry_time(ac)} \ \wedge \ t < \mathsf{tm2pt(ac)}(0) \\
& & \mathsf{ENDIF} \\
\mathsf{tm_on_T?}(t, \mathsf{ac}) & \equiv & \mathsf{IF\ regionM?}(\vec{\mathsf{ac}})\ \mathsf{THEN}\ t \geq \mathsf{tm2pt(ac)(iaf2if)} \ \wedge \ t < \mathsf{tm2pt(ac)(d_iaf)} \\
& & \mathsf{ELSE}\ t \geq \mathsf{tm2pt(ac)}(0) \ \wedge \ t < \mathsf{tm2pt(ac)(d_iaf)} \\
& & \mathsf{ENDIF}
\end{array}$$

The first predicate, $\mathsf{tm\_bef\_T?}(t, \mathsf{ac})$, determines if at time t the aircraft ac is in the SCA and it has not yet acquired the T. Since aircraft entering through region M go straight to the $\overrightarrow{\mathsf{ifix}}$, they do not acquire the T until the time given by $\mathsf{tm2pt(ac)(iaf2if)}$. Likewise, $\mathsf{tm\_on\_T?}(t, \mathsf{ac})$ determines if an aircraft has acquired the T.

Some further timing predicates are useful, distinguishing between the different stages of flight:

$$\begin{aligned} \mathsf{tm\_on\_baseleg?}(t, \mathsf{ac}) &\equiv \neg \mathsf{regionM?}(\vec{\mathsf{ac}}) \ \wedge \ t \geq \mathsf{tm2pt}(\mathsf{ac})(0) \ \wedge \ t < \mathsf{tm2pt}(\mathsf{ac})(\mathsf{iaf2if}) \\ \mathsf{tm\_on\_final?}(t, \mathsf{ac}) &\equiv (t \geq \mathsf{tm2pt}(\mathsf{ac})(\mathsf{iaf2if}) \ \wedge \ t < \mathsf{tm2pt}(\mathsf{ac})(\mathsf{d\_iaf})) \end{aligned}$$

The predicate $\mathsf{tm\_on\_baseleg?}(t, \mathsf{ac})$ determines whether an aircraft that entered through regions R or L is on a baseleg at time t. Since an aircraft that enters through region M does not travel along the baselegs of the T, they are excluded here. The predicate tm_on_final? determines if an aircraft is on final at time t, this is independent of the entry region.

Figure 3 shows how the various predicates are true for different stages of flight, depending on whether the aircraft is entering through regions R or L or through region M.

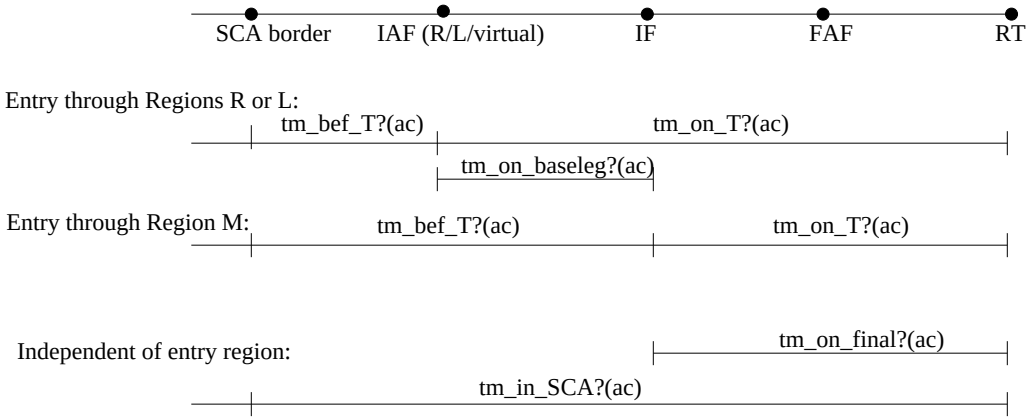


Figure 3: Timing predicates compared to stage of flight

6.2 Excluding Special Cases

Due to the construction of the protocol, there is a narrow wedge of airspace surrounding the border between regions L and R (around $x = 0$) in which separation is somewhat harder to establish. We call this the *no enter zone*:

$$\begin{aligned} \mathsf{not\_in\_no\_enter\_zone?}(\vec{\mathsf{ac}}_1, \vec{\mathsf{ac}}_2) \equiv \ & \mathsf{RL\_case?}(\vec{\mathsf{ac}}_1, \vec{\mathsf{ac}}_2) \ \supset \\ & \mathsf{one\_outside\_of\_sepmin?}(\vec{\mathsf{ac}}_1, \vec{\mathsf{ac}}_2) \ \vee \\ & \mathsf{one\_outside\_of\_sepmin?}(\vec{\mathsf{ac}}_2, \vec{\mathsf{ac}}_1) \ \vee \\ & \mathsf{both\_outside\_of\_sepmin\_div2?}(\vec{\mathsf{ac}}_1, \vec{\mathsf{ac}}_2) \end{aligned}$$

It should be noted here that this wedge is really rather small. This also splits regions R and L into Ra/Rx and La/Lx with Rx and Lx denoting the right and left *no enter zone*.

The predicate RL_case? determines if $\vec{\mathsf{ac}}_1$ and $\vec{\mathsf{ac}}_2$ are entering through regions R and L with one of the aircraft entering through each region:

$$\begin{aligned} \mathsf{RL\_case?}(\vec{\mathsf{ac}}_1, \vec{\mathsf{ac}}_2) \equiv \ & (\mathsf{regionR?}(\vec{\mathsf{ac}}_1) \ \wedge \ \mathsf{regionL?}(\vec{\mathsf{ac}}_2)) \ \vee \\ & (\mathsf{regionR?}(\vec{\mathsf{ac}}_2) \ \wedge \ \mathsf{regionL?}(\vec{\mathsf{ac}}_1)) \end{aligned}$$

The predicate one_outside_of_sepmin? is not symmetrical in the two arguments, which is why it is applied twice in not_in_no_enter_zone?. The first argument (here $\vec{\text{ac}}_1$) must have the absolute value of the x value of its entry point greater than or equal to sep_min:

$$\begin{aligned}\text{one\_outside\_of\_sepmin?}(\vec{\text{ac}}_1, \vec{\text{ac}}_2) \quad \equiv \quad & (\text{regionR?}(\vec{\text{ac}}_1) \ \wedge \ \text{regionL?}(\vec{\text{ac}}_2) \wedge \ \vec{\text{ac}}_{1x} \leq -\text{sep\_min}) \\ \vee \quad & (\text{regionL?}(\vec{\text{ac}}_1) \ \wedge \ \text{regionR?}(\vec{\text{ac}}_2) \wedge \ \vec{\text{ac}}_{1x} \geq \text{sep\_min})\end{aligned}$$

Finally, if both aircraft have the absolute values of the x values of their entry points greater than or equal to $\frac{\text{sep\_min}}{2}$, the predicate both_outside_of_sepmin_div2? holds:

$$\begin{aligned}\text{both\_outside\_of\_sepmin\_div2?}(\vec{\text{ac}}_1, \vec{\text{ac}}_2) \quad \equiv \quad & (\text{regionR?}(\vec{\text{ac}}_1) \ \wedge \ \text{regionL?}(\vec{\text{ac}}_2) \ \wedge \\ & \vec{\text{ac}}_{1x} \leq -\text{sep\_min}/2 \ \wedge \\ & \vec{\text{ac}}_{2x} \geq \text{sep\_min}/2) \\ \vee \quad & (\text{regionL?}(\vec{\text{ac}}_1) \ \wedge \ \text{regionR?}(\vec{\text{ac}}_2) \ \wedge \\ & \vec{\text{ac}}_{2x} \leq -\text{sep\_min}/2 \ \wedge \\ & \vec{\text{ac}}_{1x} \geq \text{sep\_min}/2)\end{aligned}$$

Thus, if one of $\vec{\text{ac}}_1$ or $\vec{\text{ac}}_2$ is in region R, and the other in region L, then the top level theorem only considers those situations where either aircraft has an entry point with $|x| \geq$ sep_min, or both aircraft have entry points with $|x| \geq \frac{\text{sep\_min}}{2}$.

6.3 Safe Separation

Our overall aim is to show that for any two aircraft within the SCA, those two aircraft maintain the minimum required separation at all times. The predicate safely_separated? states exactly that:

$$\begin{aligned}\text{safely\_separated?}(\vec{p}_1, \vec{p}_2) \quad \equiv \quad & \text{dist}(\vec{p}_1, \vec{p}_2) \geq \text{sep\_min} \ \vee \\ & (\text{on\_close\_corner?}(\vec{p}_1, \vec{p}_2) \ \wedge \\ & \text{dist}(\vec{p}_1, \overrightarrow{\text{ifix}}) + \text{dist}(\vec{p}_2, \overrightarrow{\text{ifix}}) \geq \text{sep\_min})\end{aligned}$$

We see that safely_separated? is expressed in terms of the location of the aircraft given as points in 2D space. These are calculated using $\overrightarrow{\text{ac\_loc}}(\vec{\text{ac}})(\text{t})$, which is described more fully in section 4. In general, we require a simple geometrical separation, that is the distance between the two aircraft must be greater than or equal to sep_min nautical miles. However, if the two aircraft are already on the T, and the first one is on final approach and the other one is on a baseleg, it is enough to have sep_min distance as measured along the T, as is discussed in section 2.

So the main theorem says that if two aircraft are both in the SCA, the AMM_properties hold for the two aircraft, and they are not in the *no enter zone*, then separation as defined above is ensured.

7 Proof Concepts

7.1 Status of Verification

In section 6 we discussed the top-level safety property Theorem 1. In this section we will give an overview and status over the proof of this theorem.

The AMM protocol uses the various entry regions as well as timings to determine if access should be granted, and this is reflected in the proof, where we consider pairs of aircraft based on their entry regions and/or which stage of flight they are at. For example, the lemma

Lemma 2 (safety_RR_LL)

$$\begin{aligned} & \mathsf{AMM\_PP2?}(\mathsf{ac}_1, \mathsf{ac}_2) \ \wedge \ \mathsf{time\_sep\_prop} \ \wedge \\ & \mathsf{same\_region?}(\vec{\mathsf{ac}}_1, \vec{\mathsf{ac}}_2) \ \wedge \\ & (\mathsf{regionR?}(\vec{\mathsf{ac}}_1) \ \vee \ \mathsf{regionL?}(\vec{\mathsf{ac}}_1)) \ \wedge \\ & \mathsf{init\_sep\_prop}(\mathsf{ac}_1, \mathsf{ac}_2) \ \wedge \\ & \mathsf{tm\_bef\_T?}(t, \mathsf{ac}_1) \ \wedge \ \mathsf{tm\_bef\_T?}(t, \mathsf{ac}_2) \\ \supset \quad & \mathsf{safely\_separated?}(\overrightarrow{\mathsf{ac\_loc}}(\vec{\mathsf{ac}}_1)(t), \overrightarrow{\mathsf{ac\_loc}}(\vec{\mathsf{ac}}_2)(t)) \end{aligned}$$

express that if two aircraft both enter through region R, with the time separations required by the protocol and both fly straight from their entry point to the fix, then spacial separation is maintained as long as both aircraft are in region R. Once the first aircraft reaches the $\overrightarrow{\mathsf{iaf\_R}}$, it acquires the T, and another lemma is used to handle this case. Since the proof is symmetrical in the case where both aircraft enter through region L, the lemma safety_RR_LL is stated so that it covers both these cases.

The proof of Theorem 1 takes the form of a large case split. The following tables indicates the different cases together with the names of the lemmas covering that case. The tables also shows the proof status for each lemma, P indicates that the proof of the lemma is complete, U indicates that it is not.

First, the proof of Theorem 1 contains the the cases listed in Table 1. This shows that for the cases regarding regions R and L exclusively the proofs are completed, whereas for the cases covering region M and the mixed cases of regions R and L, and region M are not yet proven.

Furthermore, the proof of the lemma safety_both_on_T is based on the case splits in Table 2. We see that all the cases where both aircraft are already on the T have complete proofs.

Finally, the proof of lemma Safety_mixed_T is based on the case splits listed in Table 3. Here we see that the mixed cases where one aircraft is on the T and the other one is not yet on the T have not been proven.

In the next sections we will discuss in some more detail some of the cases, namely safety_RR_LL (section 7.2), safety_both_on_T (section 7.3) and safety_RaLa (section 7.4). In section 7.5 we present some observations about one case involving one aircraft on the T and the other off of the T. Some minor modifications to the models will be necessary to handle this case.

Name of lemma	Case(s) covered	Status
safety_RR_LL	$\vec{ac}_1$ and $\vec{ac}_2$ are both in region R or both in region L	P
safety_R1L_L1R	$\vec{ac}_1$ is in region R with x value of entry point less than or equal to sep_min and $\vec{ac}_2$ is in region L – Or symmetric with $\vec{ac}_1$ in region L and $\vec{ac}_2$ in region R	P
safety_RL1_LR1	Similar to safety_R1L_L1R, but with entry point of $\vec{ac}_2$ restricted instead of $\vec{ac}_1$	P
safety_RaLa	$\vec{ac}_1$ is in region R and $\vec{ac}_2$ is in region L, both with absolute value of x value of entry point greater than or equal to $\frac{\text{sep\_min}}{2}$	P
safety_M_same_or_adjacent	$\vec{ac}_1$ and $\vec{ac}_2$ are in the same or adjacent parts of region M	U
safety_M_non_adjacent	$\vec{ac}_1$ and $\vec{ac}_2$ are both in region M, but in non-adjacent parts	U
safety_M_RL	$\vec{ac}_1$ is in region M, $\vec{ac}_2$ in either region R or region L	U
safety_RL_M	$\vec{ac}_1$ is in region R or region L, $\vec{ac}_2$ in region M	U
safety_both_on_T	$\vec{ac}_1$ and $\vec{ac}_2$ are both on the T	P
Safety_mixed_T	One aircraft has reached T, the other has not	U

Table 1: Cases in the proof of Theorem 1

Name of lemma	Case(s) covered	Status
base_and_final	$\vec{ac}_1$ is on final and $\vec{ac}_2$ is on a baseleg	P
on_close_corner	$\vec{ac}_1$ and $\vec{ac}_2$ are on a corner, as described in section 3.3	P
both_on_T_final	$\vec{ac}_1$ and $\vec{ac}_2$ are both on final	P
both_on_T_same	$\vec{ac}_1$ and $\vec{ac}_2$ are on the same baseleg	P
both_on_T_not_same	$\vec{ac}_1$ and $\vec{ac}_2$ are on opposite baselegs	P

Table 2: Cases in the proof of lemma safety_both_on_T

Name of lemma	Case(s) covered	Status
safety_T_M	$\vec{ac}_1$ is on the T, $\vec{ac}_2$ is in region M	U
safety_M_T	$\vec{ac}_1$ is in region M, $\vec{ac}_2$ is on the T	U
safety_T_RL	$\vec{ac}_1$ is on the T, $\vec{ac}_2$ is in region R or region L	U
safety_RL_T	$\vec{ac}_1$ is in region R or region L, $\vec{ac}_2$ is on the T	U

Table 3: Cases in the proof of lemma Safety_mixed_T

7.2 Proof of safety_RR_LL

In this section, we will present the basic idea behind the proof of the case where both aircraft are in region R or region L. This is illustrated in figure 4. Since in this case the aircraft have

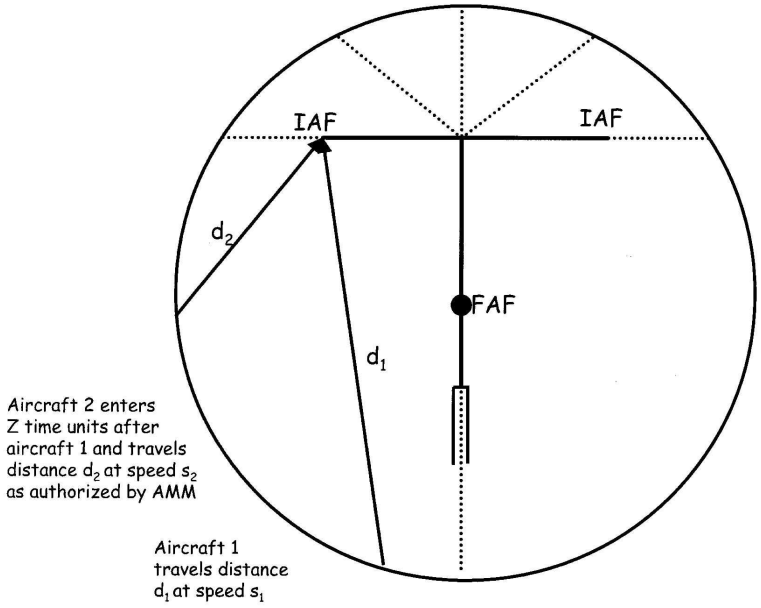


Figure 4: Approach Paths for Case RR

not reached the T, the speeds are constant and the $\overrightarrow{ac\_loc}$ function simplifies to the motion of a particle on a straight line. In fact, it is possible to reason about distances in a triangle rather than about points moving in 2D space as illustrated in figure 5. The parameters used in the figure are defined as follows:

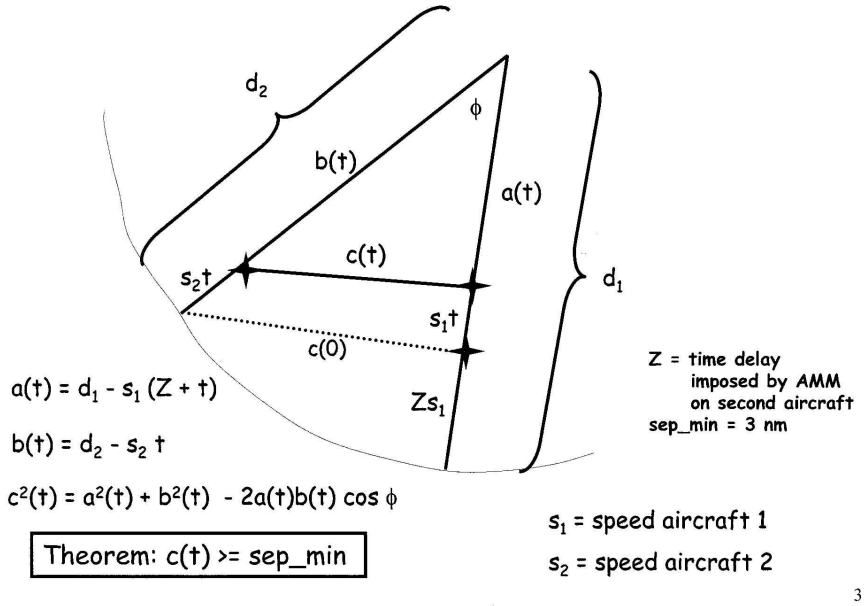


Figure 5: Abstraction To Distances in a Triangle

t	clock time, $t = 0$ when second aircraft enters SATS airspace
Z	time delay between when first aircraft enters SATS airspace and when second aircraft enters.
sep_min	separation minimum. Two aircraft should never get closer than this distance (e.g. 3 miles).
s_1	constant speed of the first aircraft
s_2	constant speed of the first aircraft
d_1	starting distance of first aircraft from the IAF (initial approach fix)
d_2	starting distance of second aircraft from the IAF (initial approach fix)

We switch to a local time t which is set to 0 when the second aircraft enters the airspace. Since aircraft 1 enters the SATS airspace Z seconds before the second aircraft, the following equations define their remaining distances to the IAF. $a(t)$ is the remaining distance for aircraft 1 and $b(t)$ is the remaining distance for aircraft 2:

$$\begin{aligned} a(t) &= d_1 - s_1(Z + t) \\ b(t) &= d_2 - s_2 t \end{aligned}$$

Using the Law Of Cosines, the distance between the two aircraft at time t can be computed as follows:

$$c^2(t) = a^2(t) + b^2(t) - 2a(t)b(t)cos\phi$$

where ϕ is the angle at the IAF.

To establish geometric separation, we must show:

THEOREM: IF $0 \leq t < \frac{d_1}{s_1} - Z$ THEN $c(t) > \mathsf{sep\_min}$

We remember that $x^2 \geq y^2 \iff x \geq y$ for non-negative x, y, so let's look at $c^2(t)$:

$$\begin{aligned} c^2(t) &= a^2(t) + b^2(t) - 2a(t)b(t)cos\phi \\ &\geq a^2(t) + b^2(t) - 2a(t)b(t) \\ &= (a(t) - b(t))^2 \end{aligned}$$

Thus, if we can establish

$$c(t) \geq |a(t) - b(t)| \geq \mathsf{sep\_min}$$

we are done.

We can simplify the formulas for $a(t)$ and $b(t)$ by using the following substitutions:

$$\begin{aligned} K &= d_1 - d_2 - s_1 Z \\ \rho &= s_2 - s_1 \end{aligned}$$

Then

$$\begin{aligned} |a(t) - b(t)| &= |d_1 - s_1(Z + t) - (d_2 - s_2 t)| \\ &= |d_1 - d_2 - s_1 Z + (s_2 - s_1)t)| \\ &= |K + \rho t| \end{aligned}$$

Thus, if we can establish

$$|K + \rho t| \geq \mathsf{sep\_min} \tag{2}$$

we are done. We can decompose this proof by a simple case analysis:

	$s_1 < s_2$	$s_1 \geq s_2$
$d_1 \leq d_2$	Case 1	Case 3
$d_1 > d_2$	Case 1	Case 4

We will illustrate the proof of case 3 and case 4.

7.2.1 Case 3

Proof. From the case 3 premises, we have:

$$d_1 \leq d_2 \ \wedge \ s_1 \geq s_2 \ \wedge \ K = d_1 - d_2 - s_1 Z \ \wedge \ \rho = s_2 - s_1 \ \wedge$$
$$\mathtt{AMM\_PP}(d_1, d_2, s_1, s_2, Z)$$

The premise AMM_PP is defined as follows

$$AMM\_PP(d_1, d_2, s_1, s_2, Z) \equiv \mathsf{Z} \geq \mathsf{Z\_min}(d_1, d_2, s_1, s_2) \ \wedge$$
$$d_2 - s_2(d_1/s_1 - Z) \geq \mathsf{sep\_min}$$

It follows trivially from the top-level premise AMM_properties for this situation. The following lemma holds:

$$\begin{array}{l} \quad \mathsf{same\_region?}(\vec{\mathsf{ac}}_1, \vec{\mathsf{ac}}_2) \ \wedge \ (\mathsf{regionR?}(\vec{\mathsf{ac}}_1) \ \vee \ \mathsf{regionL?}(\vec{\mathsf{ac}}_1)) \ \wedge \\ \quad \mathsf{AMM\_properties?}(\mathsf{ac}_1, \mathsf{ac}_2) \\ \mathsf{IMPLIES} \\ \quad \mathsf{AMM\_PP}(\mathsf{dist}(\vec{\mathsf{ac}}_1, \mathsf{which\_iaf}(\vec{\mathsf{ac}}_1)), \\ \qquad \mathsf{dist}(\vec{\mathsf{ac}}_2, \mathsf{which\_iaf}(\vec{\mathsf{ac}}_2)), \\ \qquad \vec{\mathsf{ac}}_1 \, .gs, \ \vec{\mathsf{ac}}_2 \, .gs, \ \mathsf{entry\_time}(\mathsf{ac}_2) - \mathsf{entry\_time}(\mathsf{ac}_1)) \end{array}$$

where $\vec{\mathsf{ac}}_1 \, .gs$, $\vec{\mathsf{ac}}_2 \, .gs$ are the ground speeds of aircrafts 1 and 2 respectively.

We will prove lemma (2) in two steps:

$$|K| \geq \mathsf{sep\_min} \tag{3}$$

$$|K + \rho t| \geq |K| \tag{4}$$

and Z_min simplifies to:

$$\begin{array}{rl} \mathsf{Z\_min}(d_1, d_2, s_1, s_2) \equiv & \mathsf{IF} \ d_1 > d_2 \ \wedge \ s_1 > s_2 \ \mathsf{THEN} \\ & \quad (\mathsf{sep\_min} + (d_1 - d_2))/s_1 \\ & \mathsf{ELSE} \\ & \quad \mathsf{sep\_min}/s_1 \\ & \mathsf{ENDIF} \\ \\ = & \mathsf{sep\_min}/s_1 \end{array}$$

First we need to establish step 1:

$$|K| \geq \mathsf{sep\_min}$$

Since $\mathsf{Z\_min}(d_1, d_2, s_1, s_2) = \mathsf{sep\_min}/s_1$ and $Z >= \mathsf{Z\_min}(d_1, d_2, s_1, s_2)$ we obtain:

$$s_1 Z \geq \mathsf{sep\_min}$$

From the assumptions we see that $K < 0$, so

$$\begin{array}{rcl} |K| = -K & = & d_2 - d_1 + s_1 Z \\ & \geq & s_1 Z \\ & \geq & \mathsf{sep\_min} \end{array}$$

Now all we have to do is establish step 2:

$$|K + \rho t| \geq |K|$$

But from the case 3 premises: we have $\rho \leq 0$ and hence $\rho t \leq 0$. Thus

$$\begin{aligned} |K + \rho t| &= -K - \rho t \\ &\geq -K \\ &= |K| \end{aligned}$$

and we are done.
□

7.2.2 Case 4

Proof. From the case 4 premises, we have:

$$d_1 > d_2 \ \wedge \ s_1 \geq s_2 \ \wedge \ K = d_1 - d_2 - s_1 Z \ \wedge \ \rho = s_2 - s_1 \ \wedge \\ \texttt{AMM\_PP}(d_1, d_2, s_1, s_2, Z)$$

First we will establish that

$$|K + \rho t| \geq |K| \tag{5}$$

To see this we note that the absolute value function achieves its minimum at zero, thus $|K + \rho t|$ is a minimum when $|t = -K/\rho|$. For values of t greater than this we have the relationship $t_1 \leq t_2 \supset |K + \rho t_1| \leq |K + \rho t_2|$. Thus, from $-K/\rho \leq 0 \leq t$ we obtain $|K| \leq |K + \rho t|$, the desired result. Now expanding $\texttt{AMM\_PP}(d_1, d_2, s_1, s_2, Z)$ we get

$$Z \geq (d_1 - d_2 + \mathsf{sep\_min})/s_1$$

after cross-multiplying we simplify and obtain $0 \geq -s_1 Z + d_1 - d_2 + \mathsf{sep\_min}$. Using the definition of K we get $-K \geq \mathsf{sep\_min}$ and hence $|K| \geq \mathsf{sep\_min}$. Combining this result with (5) finishes the proof.
□

7.3 Proof of both_on_T

In this section, we present the basic idea of the proof of the cases where both aircraft are currently on the T:

Lemma 3 (safety_both_on_T)

$$\mathsf{time\_sep\_prop} \ \wedge \ \mathsf{AMM\_PP2?}(\mathsf{ac}_1, \mathsf{ac}_2) \ \wedge \\ \mathsf{tm\_on\_T}(t, \mathsf{ac}_1) \ \wedge \ \mathsf{tm\_on\_T}(t, \mathsf{ac}_2) \\ \supset \\ \mathsf{safely\_separated?}(\overrightarrow{\mathsf{ac\_loc}}(\vec{\mathsf{ac}}_1)(t), \overrightarrow{\mathsf{ac\_loc}}(\vec{\mathsf{ac}}_2)(t))$$

Proof. The proof of this lemma involves 4 cases:

1. Both aircraft on same leg of the T
2. Aircraft are on opposite legs of the T
3. One aircraft is on final and the other is on a leg
4. Both aircraft are on final.

We will illustrate the proof of the first case:

$$\begin{aligned}\mathsf{both\_on\_T\_same} : \mathsf{LEMMA\ LET}\ \vec{p_1} &= \overrightarrow{\mathsf{ac\_loc}}(\vec{\mathsf{ac}}_1)(t),\\ \vec{p_2} &= \overrightarrow{\mathsf{ac\_loc}}(\vec{\mathsf{ac}}_2)(t)\ \mathsf{IN}\\ &\mathsf{same\_baseleg?}(p_1, p_2)\ \wedge\\ &\mathsf{AMM\_props?}(\mathsf{ac}_1, \mathsf{ac}_2)\ \wedge\\ &\mathsf{tm\_on\_baseleg?}(t, \mathsf{ac}_1)\ \wedge\ \mathsf{tm\_on\_baseleg?}(t, \mathsf{ac}_2)\\ &\mathsf{IMPLIES\ safely\_separated?}(\vec{p_1}, \vec{p_2})\end{aligned}$$

To establish the conclusion, it suffices to show that

$$\mathsf{dist}(p_1, p_2) \geq \mathsf{sep\_min}$$

Since both aircraft are on the same leg, we have

$$\mathsf{dist}(p_1, p_2) = \mathsf{dist\_gone}(\mathsf{ac}_1)(t) - \mathsf{dist\_gone}(\mathsf{ac}_2)(t) \tag{6}$$

(To see this for the case where both aircraft are on baselegR, note that $p_1 = \overrightarrow{\mathsf{ac\_loc}}(\vec{\mathsf{ac}}_1)(t)$ simplifies to $\mathsf{loc\_on\_legR}(\mathsf{dist\_gone}(\mathsf{ac}_1)(t))$ and $p_2 = \overrightarrow{\mathsf{ac\_loc}}(\vec{\mathsf{ac}}_2)(t)$ simplifies to $\mathsf{loc\_on\_legR}(\mathsf{dist\_gone}(\mathsf{ac}_2)(t))$. Using the definition of loc_on_legR, we have

$$\begin{aligned}p_{1x} &= \mathsf{dist\_gone}(\mathsf{ac}_1)(t) + \mathsf{iaf\_R}_x\\ p_{1y} &= \mathsf{iaf\_R}_y\\ p_{2x} &= \mathsf{dist\_gone}(\mathsf{ac}_2)(t) + \mathsf{iaf\_R}_x\\ p_{2y} &= \mathsf{iaf\_R}_y\end{aligned}$$

from which (6) is obtained.)

By definition of dist_gone we have

$$\begin{aligned}t = \mathsf{tm2pt}(\mathsf{ac}_1)(l_1)\ &\Leftrightarrow\ \mathsf{dist\_gone}(\mathsf{ac}_1)(t) = l_1\\ t = \mathsf{tm2pt}(\mathsf{ac}_2)(l_2)\ &\Leftrightarrow\ \mathsf{dist\_gone}(\mathsf{ac}_2)(t) = l_2\end{aligned}$$

Using lemma TD_safety_iaf (7), we have

$$l_1 - l_2 = \mathsf{dist\_gone}(\mathsf{ac}_1)(t) - \mathsf{dist\_gone}(\mathsf{ac}_2)(t) \geq \mathsf{sep\_min}$$

From this both_on_T_same follows.

□

Lemma 4 (TD_safe_sep_iaf)

$$\begin{aligned}&\mathsf{time\_is\_after\_iaf?}(\mathsf{ac}_1, \mathsf{ac}_2, t) \ \wedge \\ &\mathsf{time\_sep} * \mathsf{min\_speed} \geq \mathsf{sep\_min} \ \wedge \\ &\mathsf{time\_separation\_at\_iaf?}(\mathsf{ac}_1, \mathsf{ac}_2) \ \wedge \\ &\mathsf{time\_separation\_at\_rt?}(\mathsf{ac}_1, \mathsf{ac}_2) \ \wedge \\ &t = \mathsf{tm2pt}(\mathsf{ac}_1)(l_1) \ \wedge \ t = \mathsf{tm2pt}(\mathsf{ac}_2)(l_2) \\ \text{IMPLIES } &l_1 - l_2 \geq \mathsf{sep\_min} \end{aligned} \tag{7}$$

Proof. From TD_tm_sep_everywhere

$$\forall l : \mathsf{tm2pt}(\mathsf{ac}_2)(l) \geq \mathsf{tm2pt}(\mathsf{ac}_1)(l) + \mathsf{time\_sep}$$

In particular this is true for l_2:

$$\mathsf{tm2pt}(\mathsf{ac}_2)(l_2) \geq \mathsf{tm2pt}(\mathsf{ac}_1)(l_2) + \mathsf{time\_sep}$$

Since $\mathsf{tm2pt}(\mathsf{ac}_2)(l_2) = \mathsf{tm2pt}(\mathsf{ac}_1)(l_1)$ this becomes

$$\mathsf{tm2pt}(\mathsf{ac}_1)(l_1) - \mathsf{tm2pt}(\mathsf{ac}_1)(l_2) \geq \mathsf{time\_sep} \tag{8}$$

From definition of tm2pt (1), we get

$$\frac{d\ \mathsf{tm2pt}(\mathsf{ac})(l)}{dl} = \frac{1}{\mathsf{speed\_profile}(\mathsf{ac}, \mathsf{d\_iaf} - l)} \tag{9}$$

and so $d/dl\ \mathsf{tm2pt}(\mathsf{ac})(l) \geq 0$. Hence $\mathsf{tm2pt}(l)$ is an increasing function and so $l_1 \geq l_2$. From (9) we also obtain

$$\frac{1}{\mathsf{min\_speed}} \geq \frac{1}{\mathsf{speed\_profile}(\mathsf{ac}, \mathsf{d\_iaf} - l)} = d/dl\ \mathsf{tm2pt}(\mathsf{ac})(l)$$

and thus

$$\int_{l_2}^{l_1} \frac{1}{\mathsf{min\_speed}} dl \geq \int_{l_2}^{l_1} \mathsf{tm2pt}(\mathsf{ac})(l) dl$$

yielding

$$\frac{l_1}{\mathsf{min\_speed}} - \frac{l_2}{\mathsf{min\_speed}} \geq \mathsf{tm2pt}(\mathsf{ac})(l_1) - \mathsf{tm2pt}(\mathsf{ac})(l_2)$$

for all ac. Applying this to ac_1 and using (8), we get:

$$l_1/\mathsf{min\_speed} - l_2/\mathsf{min\_speed} \geq \mathsf{time\_sep}$$

Multiplying both sides by min_speed yields

$$l_1 - l_2 \geq \mathsf{time\_sep} * \mathsf{min\_speed}$$

But from premise 2, we have

$$\mathsf{time\_sep} * \mathsf{min\_speed} \geq \mathsf{sep\_min}$$

and we are done.
□

Lemma 5 (TD_tm_sep_everywhere)

$$\begin{aligned}&\mathsf{time\_separation\_at\_iaf?}(\mathsf{ac}_1, \mathsf{ac}_2) \ \wedge \ \mathsf{time\_separation\_at\_rt?}(\mathsf{ac}_1, \mathsf{ac}_2)\\ &\qquad \supset \ (\forall l : \mathsf{tm2pt}(\mathsf{ac}_2)(l) \geq \mathsf{tm2pt}(\mathsf{ac}_1)(l) + \mathsf{time\_sep})\end{aligned} \tag{10}$$

Proof. From lemma (1), we obtain

$$\begin{aligned}&\forall l : \frac{d\ \mathsf{tm2pt}(\mathsf{ac}_1)(l)}{dl} \geq \frac{d\ \mathsf{tm2pt}(\mathsf{ac}_2)(l)}{dl} \ \ \mathsf{OR}\\ &\forall l : \frac{d\ \mathsf{tm2pt}(\mathsf{ac}_2)(l)}{dl} \geq \frac{d\ \mathsf{tm2pt}(\mathsf{ac}_1)(l)}{dl}\end{aligned} \tag{11}$$

Case 1: $(\forall l : d/dl\ \mathsf{tm2pt}(\mathsf{ac}_1)(l) \geq d/dl\ \mathsf{tm2pt}(\mathsf{ac}_2)(l))$:
Thus

$$\int_l^{\mathsf{d\_iaf}} \mathsf{tm2pt}(\mathsf{ac}_1)(l)\ dl \geq \int_l^{\mathsf{d\_iaf}} \mathsf{tm2pt}(\mathsf{ac}_2)(l)\ dl$$

and hence

$$\mathsf{tm2pt}(\mathsf{ac}_1)(\mathsf{d\_iaf}) - \mathsf{tm2pt}(\mathsf{ac}_1)(l) \geq \mathsf{tm2pt}(\mathsf{ac}_2)(\mathsf{d\_iaf}) - \mathsf{tm2pt}(\mathsf{ac}_2)(l)$$

Rearranging:

$$\mathsf{tm2pt}(\mathsf{ac}_2)(l) - \mathsf{tm2pt}(\mathsf{ac}_1)(l) \geq \mathsf{tm2pt}(\mathsf{ac}_2)(\mathsf{d\_iaf}) - \mathsf{tm2pt}(\mathsf{ac}_1)(\mathsf{d\_iaf})$$

From definition of $\mathsf{time\_separation\_at\_rt?}(\mathsf{ac}_1, \mathsf{ac}_2)$

$$\mathsf{tm2pt}(\mathsf{ac}_2)(\mathsf{d\_iaf}) - \mathsf{tm2pt}(\mathsf{ac}_1)(\mathsf{d\_iaf}) \geq \mathsf{time\_sep}$$

and thus we are done.
Case 2: $(\forall l : d/dl\ \mathsf{tm2pt}(\mathsf{ac}_2)(l) \geq d/dl\ \mathsf{tm2pt}(\mathsf{ac}_1)(l))$:
Thus

$$\int_0^{l} \mathsf{tm2pt}(\mathsf{ac}_2)(l)\ dl \geq \int_0^{l} \mathsf{tm2pt}(\mathsf{ac}_1)(l)\ dl$$

and hence

$$\mathsf{tm2pt}(\mathsf{ac}_2)(l) - \mathsf{tm2pt}(\mathsf{ac}_2)(0) \geq \mathsf{tm2pt}(\mathsf{ac}_1)(l) - \mathsf{tm2pt}(\mathsf{ac}_1)(0)$$

Rearranging

$$\mathsf{tm2pt}(\mathsf{ac}_2)(l) - \mathsf{tm2pt}(\mathsf{ac}_1)(l) \geq \mathsf{tm2pt}(\mathsf{ac}_2)(0) - \mathsf{tm2pt}(\mathsf{ac}_1)(0)$$

From definition of $\mathsf{time\_separation\_at\_iaf?}(\mathsf{ac}_1, \mathsf{ac}_2)$

$$\mathsf{tm2pt}(\mathsf{ac}_2)(0) - \mathsf{tm2pt}(\mathsf{ac}_1)(0) \geq \mathsf{time\_sep}$$

and thus we are done.
□

7.4 Proof of safety_RaLa

In this section, we will present the basic idea behind the proof of the case where one of the aircraft is in region R and the other is in region L. This is illustrated in figure 6. Since in this case the aircraft have not reached the T, the speeds are constant and the $\vec{\mathsf{ac}}$ function simplifies to the motion of a particle on a straight line. In fact, it is possible to reason about distances in a quadrilateral rather than about points moving in 2D space. Furthermore, because the aircraft are restricted (by protocol) from entering the rectangular regions labelled RX and LX in figure 6 the proof is quite elementary. These regions have width sep_min/2 so the

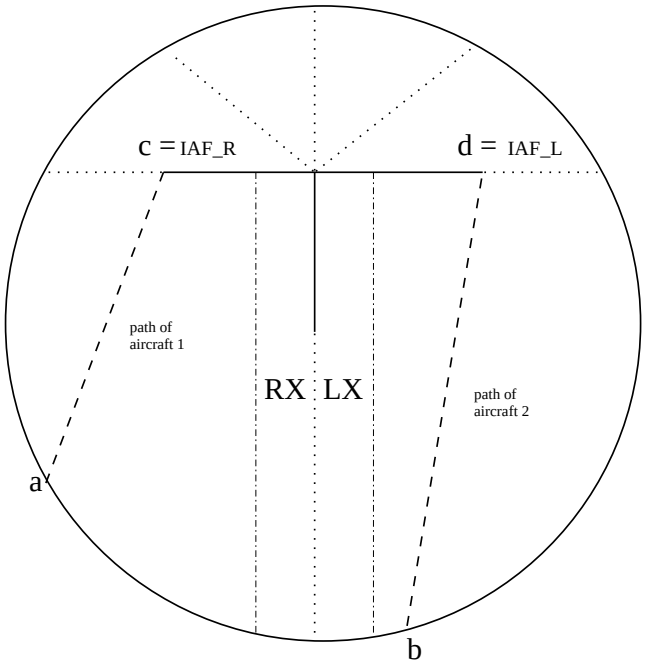


Figure 6: Restricted Regions

shortest distance between any possible trajectory is greater than sep_min. However, the proof has been formalized and checked in PVS. The key abstract lemma was:

$$
\begin{array}{l}
c_x \leq 0 \ \wedge \ d_x \geq 0 \ \wedge \\
a_x \leq 0 \ \wedge \ b_x \geq 0 \ \wedge \\
c_x \leq -\mathsf{sep\_min} \ \wedge \ d_x \geq \mathsf{sep\_min} \ \wedge \\
a_x \leq -\mathsf{sep\_min}/2 \ \wedge \ b_x \geq \mathsf{sep\_min}/2 \ \wedge \\
c_y \geq a_y \ \wedge \\
d_y \geq b_y \ \wedge \\
\mathsf{on\_segment?}(\vec{a}, \vec{c}, \vec{p_1}) \ \wedge \\
\mathsf{on\_segment?}(\vec{b}, \vec{d}, \vec{p_2}) \\
\supset \quad \mathsf{dist}(\vec{p_1}, \vec{p_2}) \geq \mathsf{sep\_min}
\end{array}
$$

The premises with the predicate on_segment? constrain locations $\vec{p_1}$ and $\vec{p_2}$ to be somewhere on the line segments from $\vec{a}$ to $\vec{b}$ and $\vec{b}$ to $\vec{c}$ respectively.

Interestingly, we originally expected to be able to prove this theorem without the removal of the restricted zones. The following informal argument was conceived as illustrated in figure 7. Since the aircraft are originally separated at their entrance times and the paths either

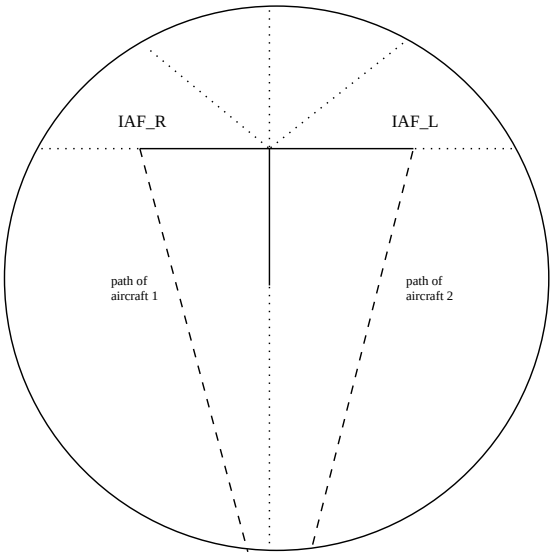


Figure 7: Some Erroneous Thinking

diverge or converge but never come closer together than the length of one of the legs of the T, they must be adequately separated no matter what the relative speeds are. Unfortunately this *obviously correct* "theorem" is false! What the informal reasoning overlooked is the fact that the original separation guarantee applies when the first aircraft enters and the second aircraft is *still outside* of the SATS airspace. Once the first aircraft enters, the regional controller no longer has responsibility for separation. What we did not realize was that there were divergent trajectories where the dynamic point of closest approach occurs after the first aircraft enters the SATS airspace. This is illustrated in figure 8. The points labeled A are the locations of the aircraft when aircraft 1 first enters. Here they are adequately but minimally separated. However when the second aircraft is faster it travels further in its trajectory than the first aircraft over a time interval. Therefore shortly after aircraft 1 enters, when they are at the positions labelled B they are closer together than when they were at points A. Therefore separation is lost. Many vain attempts were made to prove the separation property in the PVS theorem prover (e.g. using vector formulas for the point of closest approach) before it was realized that the "lemma" was not true.

7.5 Proof of safety_M_T

While in the process of writing up this paper, we decided to attempt the proof of another case. Since we had not verified any cases where one aircraft was on the T and the other aircraft

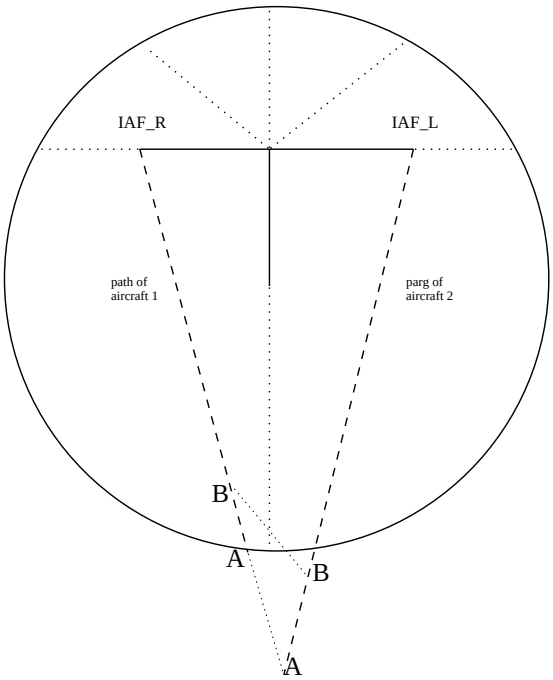


Figure 8: Some Erroneous Thinking

was still approaching the T, we decided to attempt a proof of safety_M_T, where one aircraft is in region 4 and another one is on final approach. While attempting to prove this case, it was discovered that this lemma was not true. In this case there is no time synchronization enforced by the AMM on the aircraft at their virtual or real IAFs. Unfortunately time synchronization at the IF (initial fix) and RT (runway threshold) is not sufficient to guarantee that they are geometrically separated at all times. There are two possible resolutions:

1. Require time synchronization at the IAFs.
2. Generalize the notion of safely_separated to allow path separation in these cases as in the close_corner situation.

If the second solution is selected, the definition of safely_separated will have to changed from

$$\begin{aligned}&\mathsf{safely\_separated?}(\vec{p}_1, \vec{p}_2) \equiv \\ &\quad \mathsf{dist}(\vec{p}_1, \vec{p}_2) \geq \mathsf{sep\_min} \ \vee \\ &\quad (\mathsf{on\_close\_corner?}(\vec{p}_1, \vec{p}_2) \ \wedge \ \mathsf{dist}(\vec{p}_1, \overrightarrow{\mathsf{ifix}}) + \mathsf{dist}(\vec{p}_2, \overrightarrow{\mathsf{ifix}}) \geq \mathsf{sep\_min})\end{aligned}$$

to something like

$$\begin{aligned}&\mathsf{safely\_separated?}(\vec{p}_1, \vec{p}_2) \equiv \\ &\quad \mathsf{dist}(\vec{p}_1, \vec{p}_2) \geq \mathsf{sep\_min} \ \vee \\ &\quad (\mathsf{one\_final\_one\_within\_iaf?}(\vec{p}_1, \vec{p}_2) \ \wedge \ \mathsf{dist}(\vec{p}_1, \overrightarrow{\mathsf{ifix}}) + \mathsf{dist}(\vec{p}_2, \overrightarrow{\mathsf{ifix}}) \geq \mathsf{sep\_min})\end{aligned}$$

The first resolution is a change to the operational concept itself, whereas the second is a change to the safety property itself. The first resolution would impose some additional timing constraints that would have to be enforced by the AMM and consequently could reduce the performance. We suspect that the impact would be fairly small, but this will have to be assessed via simulation. Alternatively, the relaxation of the safety property does not appear to be unreasonable in that it is directly analogous to what is already done when both aircraft are on the T. In fact in this situation the geometric separation is greater than when both aircraft are on the T.

8 Conclusion

In this paper we have presented a formal model of a concept for sequencing aircraft into a SATS airport without a tower, radar, or airport controller. The concept relies upon a timing protocol implemented in software named the Aircraft Management Module (AMM). The concept has been formally modeled using both continuous and discrete mathematics and consists of three main pieces: (1) Model of the SATS airspace using 2-dimensional vectors, (2) model of aircraft trajectories as functions of time that are determined by aircraft speed, which is dependent upon the remaining distance to the runway threshold, and (3) a requirements model of the AMM, which specifies the high-level timing properties enforced by the AMM software.

A mathematical theorem has been formulated which states that the AMM timing protocol will maintain adequate separation between all aircraft assuming that all aircraft follow the instructions given by the AMM and the "rules of the road" associated with the concept. The proof of this theorem was decomposed into ten cases of which five have been proved. The preliminary concept modeled in this paper was developed 6 months prior to the completion of the SATS draft 1 operational concept so that formal verification techniques and libraries suitable for this problem domain could be developed. Hopefully, much of the this will be reused when the final SATS operational concept is modeled and analyzed formally. In order to move on to the final SATS operational concept, the verification was not fully completed. We believe that the completion of 50% of the subcases is enough to demonstrate the feasibility of this verification approach even though everything we set out to accomplish in 6 months was not finished. However, because of the incompleteness of the proofs we do not know whether the preliminary concept itself is safe. We have run many simulations of the concept but we do not believe that simulation alone provides a rational basis for assuring safety. We do believe that with sufficient effort (probably 2 to 4 man months) the remaining proofs could be completed with at most fairly minor modifications to the AMM protocols or stated safety properties.

References

[1] *Federal Aviation Regulations/Aeronautical Information Manual.* Jeppesen Sanderson Inc, 1999.

[2] Carreno, Victor: Concept for Multiple Operations at Non-Tower Non-Radar Airports During Instrument Meteorological Conditions. In *Proceedings of the 22nd Digital Aviation System Conference*, Indianapolis, Indiana, Oct. 2003.

A Vectors Library

The NASA PVS library contains three distinct vectors libraries

1. 2-dimensional vectors
2. 3-dimensional vectors
3. N-dimensional vectors

One might wonder why there should be 2D and 3D versions, when an N-dimensional version is available. The answer is that there are some notational conveniences for doing this. For example, in the 2D version we represent a vector as

```
Vector: TYPE = [#  x, y: real  #]
```

whereas in the N-dimensional library a vector is

```
Index     : TYPE = below(n)
Vector    : TYPE = [Index -> real]
```

where n is a formal parameter (posnat) to the theory. Thus, in the two dimensional case, the x-component of a vector v is v`x whereas in the N-dimensional library it is v(0). Also certain operations are greatly simplified in the 2D case. The dot product is

```
*(u,v): real  =  u`x * v`x + u`y * v`y;          % dot product
```

in the 2-dimensional case, whereas in the N-dimensional case it is

```
*(u,v): real = sigma(0,n-1,LAMBDA i:u(i)*v(i));     % Dot Product
```

where sigma is a summation operator imported from the reals library.

In this appendix we will present the 2-dimensional version because that is what is used in the SATS work. However, the differences in the libraries are kept to a minimum. All operators, definitions, and lemmas are given identical names to simplify the use of these libraries.

A.1 2D Vectors

Two names are available for a vector type are provided in the theory vectors2D.

```
Vector    : TYPE = [#  x, y: real  #]
Vect2     : TYPE = Vector
```

The vector operators are defined as follows:

```
a     : VAR real
u,v,w : VAR Vector

-(v)  : Vector = (-v`x, -v`y);

+(u,v): Vector = (u`x + v`x, u`y + v`y);

-(u,v): Vector = (u`x - v`x, u`y - v`y);

*(u,v): real  =  u`x * v`x + u`y * v`y;          % dot product

*(a,v): Vector = (a * v`x, a * v`y);
```

A conversion is provided so that one can create 2D vectors as follows

```
(xv,yv)
```

rather than having to write

```
(# x := xv, y := yv #)
```

There are several functions and predicates provided such as

```
sqv(v): nnreal = v*v
norm(v): nnreal = sqrt(sqv(v))

zero_vector?(v) : MACRO bool = (norm(v) = 0  AND
                                v`x = 0 AND v`y = 0)

nz_vector?(v)   : MACRO bool = (norm(v) /= 0 AND
                                (v`x /= 0 OR v`y /= 0))

normalized?(v)  : MACRO bool = (norm(v) = 1)

zero    : Zero_vector =  (0,0) ;

^(nzv)         : Normalized = (1/norm(nzv))*nzv

parallel?(nzu,nzv): bool = ^(nzu)*^(nzv) = 1 OR
                           ^(nzu)*^(nzv) = -1

orthogonal?(u,v): bool =  u * v = 0 ;
```

There are several dozen lemmas available for manipulating vectors such as

```
add_assoc           : LEMMA u+(v+w) = (u+v)+w
add_move_right      : LEMMA u + w = v IFF u = v - w
```

```
add_cancel_left     : LEMMA u + v = u + w IMPLIES v = w
neg_distr_sub       : LEMMA -(v - u) = u - v
dot_eq_args_ge      : LEMMA u*u >= 0
dot_distr_add_right : LEMMA (v+w)*u = v*u + w*u
dot_scal_left       : LEMMA (a*u)*v = a*(u*v)
dot_scal_canon      : LEMMA (a*u)*(b*v) = (a*b)*(u*v)
sqv_scal            : LEMMA sqv(a*v) = sq(a)*sqv(v)
sqrt_sqv_norm       : LEMMA sqrt(sqv(v)) = norm(v)
norm_eq_0           : LEMMA norm(v) = 0 IFF v = zero
cauchy_schwartz     : LEMMA sq(u*v) <= sqv(u)*sqv(v)
```

A.2 Positions in 2D space

The theory positions2D enhances the vector space with constructs for specifying distances. One frequently wants to use a vector to designate a location in 2D space. To make this more explicit, the following type definition was added

```
Pos2D: TYPE = Vect2
```

though it is really just a synonym. Next it is useful to have a metric or distance function:

```
sq_dist(p1,p2: Pos2D): nnreal = sq(p1`x - p2`x) + sq(p1`y - p2`y)

dist(p1,p2: Pos2D)   : nnreal = sqrt(sq_dist(p1,p2))
```

Many lemmas are available, including

```
dist_refl    : LEMMA dist(p,p) = 0
dist_sym     : LEMMA dist(p1,p2) = dist(p2,p1)
dist_eq_0    : LEMMA dist(p1,p2) = 0 IFF p1 = p2
dist_norm    : LEMMA dist(u,v) = norm(u-v)
sq_dist_le   : LEMMA sq_dist(v1,v2) <= sq_dist(p1,p2) IMPLIES
                          dist(v1,v2) <= dist(p1,p2)
dist_ge_x    : LEMMA dist(p1,p2) >= abs(p1`x - p2`x)
dist_ge_y    : LEMMA dist(p1,p2) >= abs(p1`y - p2`y)
dist_triangle: LEMMA sq(dist(p2,p0)) = sq(dist(p1,p0)) + sq(dist(p1,p2))
                                          - 2*(p1-p0)*(p1-p2)
```

The following predicates are available:

```
on_circle?(p,r): bool = dist(p,zero) = r

on_line?(p1,p2,p): bool =
             EXISTS (x : real) : p = p1 + x * (p2 - p1)

on_segment?(p1,p2,p): bool =
             EXISTS (x : { y: nnreal | y <= 1}) : p = p1 + x * (p2 - p1)
```

A.3 2D Lines

The theory lines2D provides convenient formalizations for lines in 2-dimensional space. The traditional way to defines a line L is by specifying two distinct points, $\vec{p_0}$ and $\vec{p_1}$, on it. A line L can also be defined by a point and a direction. Let $\vec{p_0}$ be a point on the line L and let $\vec{dv}$ be a nonzero vector specifying the direction of the line. This is equivalent to the two point definition, since we could just put $\vec{dv} = (\vec{p_1} - \vec{p_0})$. We can also add dynamics to our line. If we assume a particle is moving in a line with a constant velocity, then we can define this linear motion using the location of the point at time zero, a velocity vector and a time parameter t:

$$\vec{p_0} + t * \vec{\mathsf{vel}}$$

which provides the location of the particle at time t.

In the library, lines are defined as a tuple:

```
                                    %     Basic         |   Dynamic
                                    %-------------------|-----------------
Line : TYPE = [# p: Vect2,          % point on the line| position at time 0
                 v: Nz_vect2 #]     % direction vector | velocity vector

Line2D: TYPE = Line
```

This enables one to represent a line using a point and a direction vector

```
p(L) + v(L)                 or              L`p + L`v
```

or using a point and a velocity vector

```
p(L) + t v(L)               or              L`p + t * L`v
```

The following alternate field names are provided

```
p0 (L: Line): MACRO Vect2 = p(L)  % alternate field names
vel(L: Line): MACRO Vect2 = v(L)
```

For example

```
 L`p0 + t * L`vel
```

This can be appreviated using the following macro:

```
loc(L: Line)(tt: real): MACRO Vect2 = p(L) + tt*v(L)
```

Two functions are provided to calculate the velocity vector for different situations:

vel_from_tm:	generates velocity vector from two points and transport time
vel_from_spd:	generates velocity vector from two points and speed

These are defined as follows

```
vel_from_tm(p1,p2,t): { v | p2 = p1 + t*v } = 1/t*(p2 - p1)

vel_from_spd(p1,p2,s):  Vect2 = IF p1 = p2 then zero
                                ELSE s/dist(p1,p2)*(p2-p1)
                                ENDIF
```

Other useful lemmas include

```
vel_from_tm_rew     : LEMMA vel_from_tm(p1,p2,t) = 1/t*(p2 - p1)
vel_from_tm_eq_args : LEMMA vel_from_tm(p,p,t) = zero
vel_from_spd_lem    : LEMMA p1 /= p2 IMPLIES
           vel_from_spd(p1,p2,ps) = vel_from_tm(p1,p2,dist(p1,p2)/ps)
vel_from_spd_norm   : LEMMA p1 /= p2 IMPLIES
           vel_from_spd(p1,p2,s) = s*normalize(p2-p1)
```

Some predicates on lines are also provided:

```
L,L1,L2: VAR Line

on_line?(p,L): bool = EXISTS (x : real) : p = p(L) + x * v(L)

on_segment?(p,L): bool =
             EXISTS (x : { y: nnreal | y <= 1}) : p = p(L) + x * v(L)

orthogonal?(L1,L2): bool = ^(v(L1))*^(v(L2)) = 0

parallel?(L1,L2)  : bool = ^(v(L1))*^(v(L2)) = 1 OR ^(v(L1))*^(v(L2)) = -1
```

A.4 Intersecting Lines

The theory intersections2D provides some efficient methods for determining whether two lines intersect or not and the point of intersection if they do so. The theory is built around a function named cross:

$$\mathsf{cross}(p, q) = p_x * q_y - q_x * p_y$$

The following simple property hold for cross:

$$\mathsf{cross}(p, q) = -\mathsf{cross}(q, p)$$

There are three cases for two lines $L0$ and $L1$:

intersecting:	$\mathsf{cross}(L0_v, L1_v) \neq 0$
parallel:	$\mathsf{cross}(L0_v, L1_v) = 0$ AND $\mathsf{cross}(\Delta, L0_v) \neq 0$
same line:	$\mathsf{cross}(L0_v, L1_v) = 0$ AND $\mathsf{cross}(\Delta, L0_v) = 0$

where $\Delta = L1_p - L0_p$. Correspondingly, the library provides the following predicates:

```
intersect?(L0,L1): bool = cross(L0`v,L1`v) /= 0

same_line?(L0,L1): bool = LET DELTA = L1`p - L0`p IN
                      cross(L0`v,L1`v) = 0 AND cross(DELTA,L0`v) = 0
```

Given two lines that intersect the function intersect_pt returns the intersection point:

```
intersect_pt(L0:Line2D,L1: Line2D | cross(L0`v,L1`v) /= 0): Pos2D =
                            LET DELTA = L1`p - L0`p,
                               ss = cross(DELTA,L1`v)/cross(L0`v,L1`v) IN
                               L0`p + ss*L0`v
```

Several key lemmas are provided:

```
intersection_lem     : LEMMA cross(L0`v,L1`v) /= 0 IMPLIES
                             LET DELTA = L1`p - L0`p,
                                ss = cross(DELTA,L1`v)/cross(L0`v,L1`v),
                                tt = cross(DELTA,L0`v)/cross(L0`v,L1`v)
                             IN
                          L0`p + ss*L0`v =  L1`p + tt*L1`v

pt_intersect         : LEMMA on_line?(p,L0) AND on_line?(p,L1) AND
                             NOT same_line?(L0,L1) IMPLIES
                                 intersect?(L0,L1)

intersect_pt_unique  : LEMMA intersect?(L0,L1) IMPLIES
                             pnot /= intersect_pt(L0,L1) AND
                             on_line?(pnot,L0)
                         IMPLIES
                             NOT on_line?(pnot,L1)

same_line_lem        : LEMMA p0 /= p1 AND
                            ( on_line?(p0,L0) AND on_line?(p0,L1) AND
                              on_line?(p1,L0) AND on_line?(p1,L1) )
                                 IMPLIES same_line?(L0,L1)

not_same_line        : LEMMA on_line?(p,L0) AND
                             NOT on_line?(p,L1)
                         IMPLIES
                             NOT same_line?(L0,L1)

 intersect_pt_lem     : LEMMA NOT same_line?(L0,L1) AND
                             on_line?(pnot,L0)  AND
                             on_line?(pnot,L1)
                         IMPLIES
                             intersect_pt(L0,L1) = pnot
```

A.5 Closest Approach

The theory closest_approach_2D provides some tools to calculate the point of closest approach (CPA) between two points that are dynamically moving in a straight line. This is an important computation for collision detection. For example, this can be used to calculate the time and distance of two aircaft (represented as line vectors) when they are at their closest point.

Suppose we have two time-parametric linear equations

$$\vec{p}(t) = \vec{p_0} + t\vec{u} \qquad \vec{q}(t) = \vec{q_0} + t\vec{v}$$

Minimum separation occurs at:

$$t_{\mathsf{cpa}} = -\frac{\vec{w_0}(\vec{u} - \vec{v})}{|\vec{u} - \vec{v}|^2}$$

where $\vec{w_0} = \vec{p_0} - \vec{q_0}$. The library provides a function time_closest:

```
time_closest(p0,q0,u,v): real =
   IF norm(u-v) = 0 THEN  % parallel, eq speed
      0
   ELSE
      -((p0-q0)*(u-v))/sq(norm(u-v))
   ENDIF
```

The following lemma gives an alternate way to calculate the function.

```
time_closest_lem: LEMMA norm(u-v) /= 0 AND
                        a = (u-v)*(u-v) AND
                        b = 2*(p0-q0)*(u-v)
                        IMPLIES
                          time_closest(p0,q0,u,v) = -b/(2*a)
```

The lemma time_cpa establishes that this time is indeed the point where the distance is at a minimum.

```
time_cpa: LEMMA t_cpa = time_closest(p0,q0,u,v)
             IMPLIES
                is_minimum?(t_cpa,(LAMBDA t: sq_dist(p0+t*u,q0+t*v)))
```

See

http://geometryalgorithms.com/Archive/algorithm_0106/algorithm_0106.htm

for a very nice discussion.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
01- 03 - 2004	Technical Memorandum	

4. TITLE AND SUBTITLE

Formal Modeling and Analysis of a Preliminary Small Aircraft Transportation System (SATS) Concept

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Carreno, Victor A.; Gottliebsen, Hanne; Butler, Ricky; and Kalvala, Sara

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

23-786-10-10

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

NASA Langley Research Center
Hampton, VA 23681-2199

8. PERFORMING ORGANIZATION REPORT NUMBER

L-18449

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

National Aeronautics and Space Administration
Washington, DC 20546-0001

10. SPONSOR/MONITOR'S ACRONYM(S)

NASA

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

NASA/TM-2004-212999

12. DISTRIBUTION/AVAILABILITY STATEMENT

Unclassified - Unlimited
Subject Category 61
Availability: NASA CASI (301) 621-0390 Distribution: Standard

13. SUPPLEMENTARY NOTES

Carreno and Butler, Langley Research Center; Gottliebsen, National Institute of Aerospace; Kalvala, University of Warwick
An electronic version can be found at http://techreports.larc.nasa.gov/ltrs/ or http://ntrs.nasa.gov

14. ABSTRACT

New concepts for automating air traffic management functions at small non-towered airports raise serious safety issues associated with the software implementations and their underlying key algorithms. The criticality of such software systems necessitates that strong guarantees of the safety be developed for them. In this paper we present a formal method for modeling and verifying such systems using the PVS theorem proving system. The method is demonstrated on a preliminary concept of operation for the Small Aircraft Transportation System (SATS) project at NASA Langley.

15. SUBJECT TERMS

Safety; Software; Formal methods; Verification; Air traffic management

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			STI Help Desk (email: help@sti.nasa.gov)
					19b. TELEPHONE NUMBER *(Include area code)*
U	U	U	UU	44	(301) 621-0390