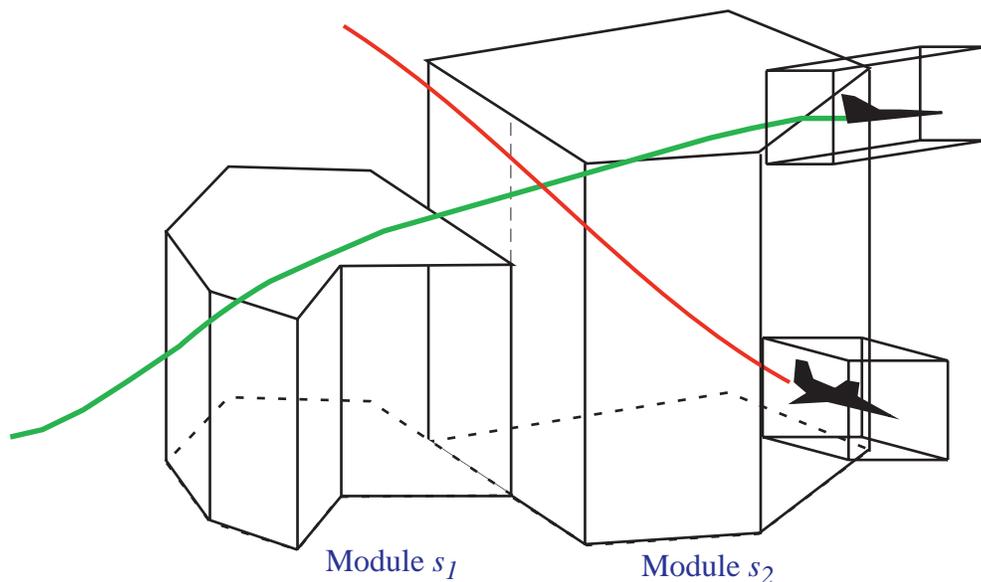


Development of Airspace Sector and Encounter Models to Support the Analysis of Aircraft Separation and Collision Risk

Final Report



NEXTOR Research Report RR-98-16

A. A. Trani and H. D. Sherali (Principal Investigators)
J. C. Smith, S. Sale and C. Quan (Graduate Research Assistants)

Virginia Polytechnic Institute and State University
Blacksburg, VA 24061

November, 1998

Intentionally Left Blank

Executive Summary

This report presents a first-order analysis of blind conflicts expected to affect the NAS system in the near future under two Free Flight operational concepts: RVSM and Cruise Climb. The study focused on the development and use of two computer models (AOM and AEM) to respectively predict traffic flows across well defined volumes of airspace, and the number of potential blind conflicts if all flight plans are executed without controller or pilot intervention. The models developed have been coded in Matlab, a general engineering language, facilitating their execution on any computer platform (PCs, PowerPC Macs, and UNIX workstations) without modifications.

While this study provides a first-order approximation of the level of conflict exposure in a particular center or sector it does not provide a measure of collision risk in the true sense. Further investigation of the end-game ATC controller and pilot dynamics (including aircraft navigational accuracy) is needed to truly quantify collision risk.

Some insightful computational test are conducted to understand traffic pattern variations and blind conflicts in four enroute control centers in CONUS. The time and spatial characteristics of these conflicts were studied using the tools developed to provide a view into the type of conflict encounters expected in future NAS operations. The hope is that these tools would be further refined to assess collision risk incorporating human and vehicle reliability models.

Several conclusions can be derived from this case study:

- 1) There would be likely moderate to substantial variations in traffic flow patterns across various ARTCC sectors in NAS. The introduction of flexible flight planning rules expected in Free Flight would affect differently various ARTCC centers according to their geographical location. In this study ZMA and ZJX centers had less variation in 15-minute traffic flows than those observed across ZID and ZTL.
- 2) The number of potential conflicts in the enroute airspace system would decrease with the introduc-

Executive Summary

tion of Free Flight operations if reduced vertical separation criteria is allowed. It is not possible to quantify the risk associated with reduced separation blind conflicts using the models developed. However, further investigation is needed since ATC controllers and pilots operating under RVSM rules might have less time to react to blunders under these circumstances (assuming current levels of automation).

3) The number of blind conflicts expected under Cruise Climb and RVSM modes (as defined in this report in Chapter 5) are of the same order of magnitude. It is not clear how ATC controllers would react to potential conflicts between two or more aircraft operating in a cruise climb and what would be their influence on collision risk. Further investigation is necessary.

4) In general, there are substantial to moderate differences in the time and space distribution of blind conflicts under RVSM and Cruise Climb scenarios. The effect of these distributions in ATC controller monitoring workload and eventual reliability to intervene under blunder conditions should be further investigated.

5) In general, vertical transition conflict times under RVSM and Cruise Climb scenarios are expected to be shorter in duration due to the smaller vertical separation criteria. Enroute conflict times (i.e., coplanar conflicts) varied significantly. Under some circumstances, enroute conflict times increased for at least one of the Free Flight scenarios investigated.

6) The distribution of relative headings of conflicts varied in the transition to some Free Flight scenarios (i.e., cruise climb). This parameter could have important implications on how controllers perceive conflicts and eventually, on the intervention modes used to separate traffic. Further investigation of this important parameter is also needed.

Preface

This report documents research undertaken by the National Center of Excellence for Aviation Operations Research, under Federal Aviation Administration Research Grant Number 96-C-001. This document has not been reviewed by the Federal Aviation Administration (FAA). Any opinions expressed herein do not necessarily reflect those of the FAA or the U.S. Department of Transportation.

The authors are appreciative of the support provided by the FAA Office of Operations Research and Investment Analysis (ASD), and in particular, the guidance and leadership provided by Dr. Steven Cohen. We would also like to acknowledge support of the following individuals at FAA: Ken Geisenger, Diana Liang, Bruce Ware, Dan Crittenbaum and Steve Bradford. Finally, we would like to thank Carolyn Sorensen (Crown Communications) and Bill Coligan and Stefan Mondoloni of CSSI Inc. for sharing NARIM tools and information with us.

Table of Contents

Chapter 1 Introduction	1
Air Traffic Operations.....	2
Free Flight	3
Research Scope, Objective and Approach	3
Chapter 2 Review of Existing Models and Tools	5
Airspace Encounter Models.....	5
Airspace Analysis Models	7
Chapter 3 Airspace Occupancy Model	9
Model Assumptions	9
Flight Plan Generation	10
Airspace Sector Description	11
Occupancy Determination	11
Definition of Terms	13
Pre-processing Sector Data	16
Sector and Module Mathematical Definitions	17
Types of Vertices	17
Adjacency with Respect to Nodes	18
Adjacency Information with Respect to Sector Modules	19
Identifying Pseudo-Vertices	20
Adjacency Information with Respect to Vertical Faces	21
Pre-processing Airport Data	22
Pre-processing Flight Plans	22
Dummy Sectors	22
Vacuums	22
Sector Occupancy Determination Algorithm	23
Extension of Algorithm to Handle Airspace Vacuums	24
Extension of the Algorithm for the Three Dimensional Case	25
Procedure to Determine Exit Points	26
Determination of the Next Sector Module after Exiting	27
Determination of the Next Sector Module after Passing through a Vacuum	28
Chapter 4 Aircraft Encounter Model	29
Coordinate System Definitions and Transformations	31
Spherical Model	33

Table of Contents

Truncated Spherical Model	34
Box-Model for Aircraft Encounter Analysis	35
Example 4.1	40
Example 4.2.	43
An Aggregate Metric for Conflict Severity	45
Example 4.3.	48
Example 4.4	48
Summary of Sector Conflict Analysis	48
Chapter 5 Model Application	55
NARIM Scenarios	55
National Airspace (NAS) Concept of Operations	56
Wind-Optimized Routing with Hemispherical Rules and Assigned Altitudes	56
Wind-Optimized Routing with a Reduced Vertical Separation and Assigned Altitudes	57
Wind-Optimized Routing with Hemispherical Rules (Cardinal)	57
Wind-Optimized Profiles with Reduced Vertical Separation Criteria (RVSM)	57
Wind-Optimized Profiles without Hemispherical Rules (Climb-Cruise)	58
Model Validation	59
Model Results	61
Traffic Flow Patterns	62
Conflict Analysis Results	66
Chapter 6 Conclusions	81
Recommendations	83
Possible Model Extensions	84
Bibliography	85
Appendix A Model Data Structures and M-File Definitions	89
AOM and AEM Data Structures	89
M-Files	96
Appendix B ARTCC Sector Traffic	113

List of Figures

Figure 3.1 Airspace Occupancy Model (AOM) and Airspace Encounter Model (AEM).	11
Figure 3.2 Typical Sector Geometry (showing a sector made up of 2 FPAs).	13
Figure 3.3 Occupancy Determination Flowchart.	14
Figure 3.4 Geometric Components of a Sector Module.	15
Figure 3.5 Basic Face and Vertex Definitions Inside a Sector Module.	16
Figure 3.6 Types of Vertices.	17
Figure 3.7 Adjacency with Respect to a Vertex.	18
Figure 3.8 Adjacency with Respect to a Vertex.	19
Figure 3.9 Pseudo-Vertices on a Face.	20
Figure 4.1 AEM Model Block Diagram.	29
Figure 4.2 Definition of Latitudes and Longitudes.	31
Figure 4.3 Polar To Cartesian Coordinate Transformation.	31
Figure 4.4 Standard Envelope and Proximity Shell for Aircraft A.	36
Figure 4.5 Trajectories of Aircraft A and B.	41
Figure 4.6 Circular Aircraft Trajectory Term Definitions.	44
Figure 4.7 Example to Illustrate the Effect of Using Percent Rates.	48
Figure 4.8 Example of Parallel Flights with Opposite Headings.	49
Figure 4.9 Illustration of 2-D Rectangle Created for Neighboring Sector Analysis.	52
Figure 5.1 Partial ZMA-ZJX Traffic Data used for Model Validation (August 18, 1997).	59
Figure 5.2 AOM Sector Traffic Results (Sector 85 ZID).	63
Figure 5.3 AOM Sector Traffic Results (Sector 77 ZMA).	64
Figure 5.4 Relative Heading Angle Conflict Instances (ZID and ZTL).	74
Figure 5.5 Vertical Transition Conflict Times (ZID and ZTL).	75
Figure 5.6 Enroute Conflict Times (ZID and ZTL).	76
Figure 5.7 Minimum Distance Distribution for Vertical Transition Conflicts (ZID and ZTL).	77
Figure 5.8 Minimum Distance Distribution for Enroute Conflicts (ZID and ZTL).	78
Figure 5.9 Proposed R Statistic for Vertical Transition Conflicts (ZTL and ZID).	79

List of Figures

List of Tables

Table 4.1 Sector Piercing Patterns.	27
Table 5.1 Scenarios Used in the Model Study.	58
Table 5.2 Validation Results for ZMA-ZJX ARTCC Traffic (August 18, 1997 Data).....	60
Table 5.3 Statistical Analysis of Sector Traffic Flows (Wilcoxon Rank Sum Test $\alpha=0.05$).....	65
Table 5.4 ZMA and ZJX ARTCC Conflict Statistics (Baseline).....	66
Table 5.5 ZMA and ZJX ARTCC Conflict Statistics (RVSM).	67
Table 5.6 ZMA and ZJX ARTCC Conflict Statistics (Cruise Climb).....	67
Table 5.7 ZTL and ZID ARTCC Conflict Statistics (Baseline).	68
Table 5.8 ZTL and ZID ARTCC Conflict Statistics (RVSM).	68
Table 5.9 ZTL and ZID ARTCC Conflict Statistics (Cruise Climb).....	69
Table 5.10 Statistical Analysis of 15-Minute ARTCC Center Conflicts.	70
Table 5.11 Relative Heading Statistics for Various NAS Operational Scenarios.....	72
Table 5.12 Vertical Transition Conflict Time Statistics for Various NAS Operational Scenarios.....	72
Table 5.13 Enroute Conflict Time Statistics for Various NAS Operational Scenarios.....	73
Table 5.14 Closest Point of Approach Statistics for Various NAS Operational Scenarios (Vertical Transition Conflicts)	73
Table 5.15 Closest Point of Approach Statistics for Various NAS Operational Scenarios (Enroute Con- flicts)	73
Table B.1 ZMA Center Sector Traffic Patterns.	110
Table B.2 ZJX Center Sector Traffic Patterns.	112
Table B.3 ZID Center Sector Traffic Patterns.	114
Table B.4 ZTL Center Sector Traffic Patterns.	115

List of Tables

According to recent Federal Aviation Administration (FAA) statistics (FAA, 1997) all Air Traffic Control (ATC) units handle over twenty five million flights per year. The number of aircraft operations in NAS is growing at a modest pace of 1.5-3% per year. The safety record of these operations speaks very highly of the safety net built around the current system and of the highly trained personnel managing aircraft traffic on a daily basis. Nevertheless collisions occur with low frequency and are of concern to FAA and the National Aeronautics and Space Administration (NASA). According to the Office of Technology assessment, the number of midair collisions in the U.S. has the potential of growing quadratically with the number of operations (OTA, 1988). New ATC automation tools and air traffic control procedures could significantly change this trend in the future. The introduction of *Free Flight* would certainly add an element of uncertainty to the equation to estimate collision risk. At this point it is clear that better tools and techniques are needed to quantify the risk of midair collisions in the future NAS once Free Flight concepts of operations are introduced.

This report summarizes the results of a preliminary task given to the *National Center of Excellence for Aviation Operations Research* (NEXTOR) to develop airspace scenarios to estimate blind conflicts and their geometries in the enroute airspace. The models developed as part of this report constitute a flexible toolset that could be extended to include ATC/pilot interventions in the future, and thus assess col-

lision risk in a more realistic fashion. The study is currently restricted to the enroute airspace system. However, it is not restricted to the cruising portion of the flight, and recognizes that portions of the enroute airspace system are used for vertical transitions. These transitions involve periods of high workload activity for both pilots and controllers and are of concern from the collision risk modeling perspective.

This report deals with the development of two computer models to study airspace occupancy and potential collision risk encounters. Given the sector geometry and flight schedules, the models developed determine the occupancy of sectors and the number of potential collisions if the flight trajectories are maintained according to schedule without intervention. Of special interest to this project is the potential geometry of blind flying collisions. This is judged important because in Free Flight, the role of ATC controllers could become more passive (i.e., monitoring vs. active role), and aircraft would have lateral and vertical freedom to fly near-optimal trajectories. Both of these events could certainly add to the complexity of the Air Traffic Management (ATM) system.

1.1 Air Traffic Operations

The entire airspace over the United States is divided into twenty-one centers, each regulated by an Air Route Traffic Control Center (ARTCC). Each of these centers is sub-divided into sectors. Sectors are classified into three groups: low, high and super-high sectors depending upon the floor and ceiling boundaries. Low sectors lie below the flight level 240 (FL 240). High sectors extend between FL 240 and FL 350. The super-high sectors lie above FL 350.

Air traffic operations are monitored by air traffic controllers, having assigned duties pertaining to a particular sector. Air traffic controllers keep an eye on radar displays and communicate with the pilots in order to resolve any potential conflicts. Controllers coordinate their activities with their counterparts in adjacent sectors so that the monitoring of flight operations is smooth and continuous. The workload imposed on the air traffic controllers will depend on the number of flights crossing the sector at any instant of time, the number of potentially conflicting flights, the level of ATC equipment automation, and the conflict geometry of each conflicting flight pair. Human factor parameters such as conflict de-

tection time, conflict resolution strategy, and secondary task loading also play a role in workload assessment. This in turn might affect collision risk.

1.2 Free Flight

Free Flight offers a new paradigm in how air traffic operations will be conducted in the future. *Free Flight* operations will be mainly governed by communications, navigation, and surveillance information transmitted through satellites, using advanced on-board navigation equipment and transponders. The existing ATC system establishes aircraft positions (i.e. surveillance function) through ground-based radar equipment. In the current system, navigation is also dependent upon ground navigation aids, and communications are based on a hybrid of Very High Frequency (VHF) line-of-sight and satellite-based techniques. In *Free Flight* pilots will receive real-time information regarding nearby flights, and on-board traffic advisories will provide cues required for air traffic control separation. This way, a decentralized air traffic control service could be provided. In a critical situation, the air traffic controller may interfere to resolve the conflict. The main motivation behind Free Flight is that the airlines will have more flexibility in filing their flight plans using point-to-point routes without reliance on ground navigation aids. This will result in more efficient and cost effective flight trajectories. The FAA will perhaps have some degree of oversight to approve these flight plans making sure that they will not impose excessive workload on any of the enroute air traffic control centers.

1.3 Research Scope, Objective and Approach

In the future ATM system, it is imperative to have a set of models to assess aircraft flows across regions of congested airspace to reduce the costs of airspace users and service providers. These models may serve as an advisory tool to: 1) approve flight plans in the Free Flight concept of operations; 2) reschedule flights around Special Use Airspace (SUA) areas such as in the event of spaceport launches, and 3) estimate blind conflicts in the enroute and terminal airspace among others.

The specific goals of this research are:

- 1) to determine sector occupancy changes if various Free Flight concept of operations are adopted in the National Airspace System (NAS), and
- 2) to identify the types and geometry of future conflicts arising from new concepts of operations.

Two computer models have been developed for this purpose using Matlab 5.2, a general engineering software package developed by the Mathworks (1997). The models developed are: 1) the Airspace Occupancy Model (AOM) and 2) the Airspace Encounter Model (AEM). Both models can be executed on any Windows 95/NT compatible PC, Macintosh, or Unix Workstation without modifications.

Review of Existing Models and Tools

This chapter reviews various airspace analysis models and tools that provide some capability to quantify traffic density, conflict potential and collision risk. We briefly mention models that have attempted to describe workload as function of sector traffic density in this study for completeness. These are judged relevant since workload might be one of many variables involved in the assessment of collision risk in the future. A more extensive literature review on collision risk tools and models is included in the Concept Paper prepared by the Joint FAA/Eurocontrol Separation and Collision Risk Modeling group (FAA/Eurocontrol, 1998).

2.1 Airspace Encounter Models

Several past studies have attempted to quantify collision risk metrics for various general traffic scenarios. Ratcliffe and Ford (1981) developed several analytic and computer models to quantify the conflicts arising between various aircraft interacting in regions of uncontrolled airspace. Hourly conflict rates were found to be proportional to the aircraft warning time, the number of aircraft in the study area (quadratic function) and inversely proportional to the radius of the airspace area (also quadratic). This study also detailed some of the scenarios that could, in principle, be considered of higher threat than

others. For example, it was found that collision threats were more likely to occur with 45 degrees off head-on than in dead ahead scenarios for all range of speeds investigated.

In a related study, Ford (1982) investigated the intrinsic safety features of height rules in uncontrolled airspace operations using random distribution of flights in the vertical domain. The findings of this paper indicate that the collision risk under current height rules is greater than in vertical random flight mode (a version of what we call *Free Flight* today) unless significant height keeping errors are introduced. Various height rules were investigated including quadrantal, linear, spiral and semicircular.

Uncontrolled airspace flight analysis and collision risk assessment provides a first order approximation of conflict probabilities under various airspace operational rules. Almost all conflict risk assessment models require intrinsically some knowledge of the expected number of conflicts in the airspace over time. Many of the well know models reported in the literature to derive collision risk metrics over the North Atlantic have used procedural uncontrolled airspace assumptions (Reich, 1966; Brooker, 1982). These models have been refined over the years in support of reduced separation standards (Machol, 1995). The challenge in modeling collision risk for NAS operations is the nature of conflict paths, sector density and intervention rules used by ATC controllers to separate traffic.

Past approaches to generate flight paths have range hypothetically generating random flights to actual flight schedules. Goodwin and Ford (1984) describe two methods to generate random aircraft traffic in a volume of airspace. First a two dimensional model exploiting the properties of a circle was developed. Two variations of this model are then discussed for rectangular airspace regions. Finally, a spherical traffic flux model is proposed to extend the random flight generation to the three dimensional airspace case. While this work was primarily devoted to random flight generation issues it also presents some interesting conflict statistics resulting from the scenarios modeled. The approach taken in this research project uses flight plans derived from actual NAS operational data (i.e., flight plan data collected for a typical day in 1996). FAA data for various NAS operational concepts is used as input to the models developed in this research project. The FAA has also derived data for future NAS scenarios using expected traffic operations in 2005 and 2015.

Quantifying collision risk requires some knowledge of the navigation performance of the aircraft and

the reliability of the avionic systems onboard. Several past studies have looked at the navigation performance capabilities of aircraft operating in jet routes (Polhemus and Livingston, 1981 and Hsu, 1982; Harrison and Moek, 1986; Ten Have et al., 1988). These studies serve to corroborate expected height and lateral navigation keeping abilities of aircraft. Other studies have concentrated in the development of suitable mathematical functions and models to estimate probabilities of lateral and vertical overlaps (Nagaoka, 1984; Hsu, 1982). Other studies have looked at the protection offered by anti-collision aircraft devices such as TCAS (Ford, 1982). While these studies do not attempt to estimate collision risk directly, they provide a frame of reference to further develop blind conflict models such as those proposed in this research project.

2.2 Airspace Analysis Models

Previous studies looking at the complex dynamics of air traffic control tasks have used the temporal and spatial variations of flights to determine metrics that precede workload measures. Some of these studies used extended time line analysis (Lauderman and Palmer, 1995) and dynamic density (K. Smith et al., 1998). Other studies have attempted measuring controller workload in Free Flight tasks (Hilburn et al., 1997; Wyndemere, 1996). Finally, several complex models have been developed to assess human response times in ATC/pilot tasks. One such a model is MIDAS developed at NASA Ames Research Center (Corker et al., 1997).

All these efforts attempt to understand the dynamics of pilot/ATC control interactions under a specific set of circumstances. Few studies, however, provide insight on how Free Flight concept of operations will affect the conflict exposure level of aircraft to collisions. The conceptual models developed in this research could, in principle, provide a foundation to add ATC controller/pilot end-game dynamics. This aspect will be detailed in Chapter 6 of this report.

This chapter describes the Airspace Occupancy Model (AOM). This model is used to estimate module and sector occupancies and constitutes the input to the Aircraft Encounter Model (AEM) described in Chapter 4. The main routines of this model are shown in Figure 3.1. In general the model takes flight plans or flight tracks, converts them into mathematical terms, scrutinizes the flight trajectory over a defined region of airspace to determine sector crossings and occupancies over time. The model provides graphical outputs of sector occupancies and generates data structures used to analyze pairwise aircraft conflicts.

3.1 Model Assumptions

The assumptions made in the development of AOM are as follows:

1. All flights are assumed to fly along straight lines between way-points, (dummy way-points could be specified to further discretize curvilinear flight trajectories).
2. Two nodes which are less than 0.35 nautical miles apart are assumed to define the same point in the airspace. This assumption is made to correct for inaccuracies in data that sometimes assign different slightly perturbed locations to the same node, and hence create vacuums within the airspace.

3. A flight that moves along a common boundary of some sector modules, is assumed to pass through only one of them. The choice is made based on selecting the currently occupied sector, if applicable, or arbitrarily otherwise.

AOM requires a series of aircraft flight plans and the sector geometry as inputs. The model processes the information to determine the occupancy of each sector by different flights over time. The essence of the model lies in storing the adjacency information of sectors, and identifying the sectors crossed by a flight plan. AEM uses the outputs of AOM and conducts a microscopic evaluation of all possible aircraft blind conflicts in every airspace sector. The outputs of AEM are conflict geometry statistics. The inter-relationships between these models are illustrated in Figure 3.1. AOM analyzes individual flight paths from an origin to a destination airport and estimates time traversals over each sector encountered. This output is then used by AEM to estimate the number of times aircraft pairs could be in conflict if blind flying occurs.

3.2 Flight Plan Generation

The flight plans for a particular day were used for the purpose of analyzing these scenarios. Flight plans obtained from the FAA ETMS database along with the corresponding air traffic situation on November 12, 1996, were used for this purpose. Whenever a flight is assumed not to rely on the ground-based navigation aids, a wind-optimized trajectory is adopted. Wind optimized routing is the three dimensional trajectory that will have the least possible impedance from the prevailing winds.

The flight plan inputs to AOM can take three forms: 1) flight plans filed by pilots on a given day (ETMS data), 2) flight tracks extracted from SAR data, or 3) flight plans predicted by NARIM flight plan generators such as OPGEN . There are common elements to all these data sources and, in general, a flight plan should contain the following information.

- 1 . Way-points in latitude (degree), longitude (degree) and altitude (hundreds of feet).
- 2 . Time tags corresponding to the crossing of each of the above way-points (during any time interval).
- 3 . The originating airport (a three letter airport designator). (Optional)

4. The destination airport (a three letter airport designator). (Optional)

The flight plans for any particular day in the past can be obtained from the FAA Enroute Traffic Management System (ETMS) database or from the Sector Design and Analysis Tool (SDAT) database. In order to use the model to analyze predicted air traffic, an independent flight generator that develops flight plans having the above mentioned five attributes, could be coupled with the Airspace Occupancy Determination Model.

3.3 Airspace Sector Description

Sectors are well-defined airspace regions specified by the FAA for regulating air traffic. Each sector is comprised of Fixed Posting Areas (FPA) and each of these FPAs is made up of modules. A module is a convex or non-convex airspace polytope in shape defined by its vertices and its floor and ceiling altitudes. Modules are stacked one over another to form an FPA, and several such adjacent FPAs form a sector as shown in Figure 3.2. The main source of enroute and Terminal Radar Approach Control (TRACON) sector information used in this study came from the FAA ACES database.

3.4 Occupancy Determination

A flight that crosses a sector will be detected by the model based on the adjacency information that is generated and stored during the pre-processing step. Since each sector is complex in shape, the analysis is done at the module level and the result is translated to the sector level by considering the particular modules that make up the sector.

The model first identifies the initial module encountered by the flight. This may be the module that encompasses the originating airport. Sometimes, the originating airport may not lie within the defined modules. In such a case, the model identifies the module through which the flight enters the defined airspace. Once the initial module through which the flight passes is detected, the point and time of exit is identified. This point is found by checking if the flight crosses any of the faces, the floor, or the ceiling defining the module, without merely glancing at it and remaining within the same module.

3.4 Occupancy Determination

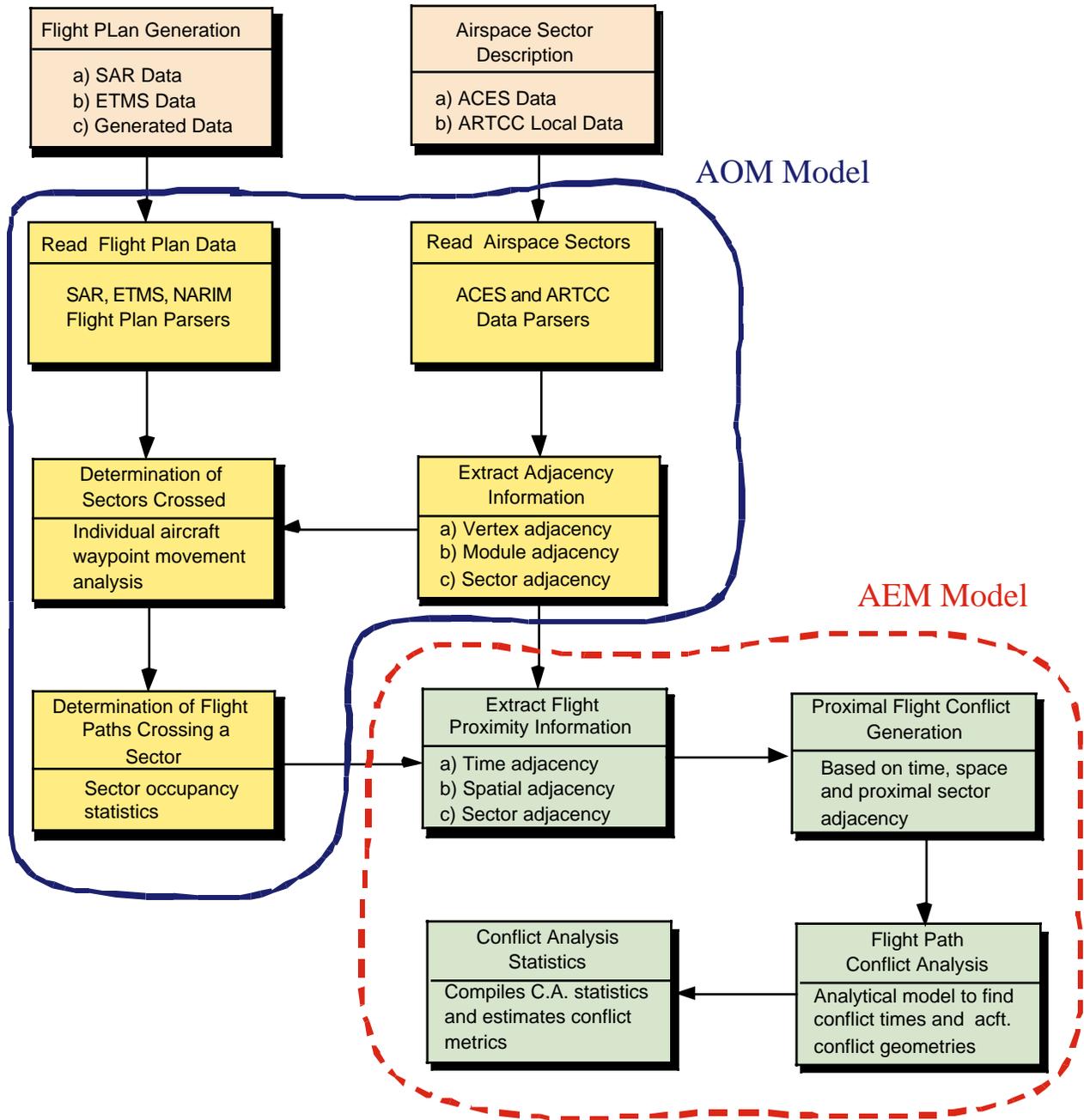


Figure 3.1 Airspace Occupancy Model (AOM) and Airspace Encounter Model (AEM).

The program also identifies the way the flight exits the module, i.e., if the flight exits across a face, or the floor, or the ceiling, or at a vertex, or across an edge. With this knowledge, and since module adjacency information is known, the next module into which the flight enters is determined. This process of identifying each traversed module and the corresponding occupancy time is continued until the flight reaches its destination. Next, the sectors through which the flight passes is identified by examining the modules that comprise each sector. This provides information on all flights that cross a particular sector along with related occupancy times. A flow chart illustrating the sector occupancy determination methodology is shown in Figure 3.3.

The procedures implemented in the AOM can be summarized into four steps: data input, pre-processing, processing, and post-processing. Data input reads flight plan (or track) and airspace sector data from an external source. Pre-processing refers to the creation of airspace mathematical boundaries including dummy sectors and vertex matching. Processing identifies sectors pierced by each flight and sector traversal times. Post-processing refers to the aggregation of flight traversals per sector and the computation of sector occupancies. These steps are illustrated in Figure 3.3.

3.5 Definition of Terms

In order to describe the mathematical procedures in AOM it is important to define some nomenclature used in the development of this model.

Sector Module. A sector module is a fundamental unit in the definition of an airspace. One or more sector modules form a sector. A sector module is a three dimensional convex or non-convex polytope in shape.

Vertical Faces. These are the rectangular, two dimensional, vertical bounding faces that define a sector module as shown in Figure 3.4.

Floor. Defines the lower horizontal face of a sector module.

Ceiling. Defines the top horizontal face of a sector module.

Vertex. A vertex is a corner point of a sector module.

Pseudo-Vertex. A pseudo-vertex for a sector module is a vertex for some other sector module that is present on a vertical face of the given sector, but is not an original defining vertex of its floor and ceiling.

Vertical Edge. This is the line of intersection of two adjacent vertical faces of a sector module.

Horizontal Edge. This is the line of intersection of the floor or ceiling with a vertical face.

Node. A node corresponds to a corner point formed by the two dimensional projection of a module onto its floor or ceiling. It is used to define the floor and ceiling geometry of a sector module, and might correspond to the projection of one or more vertical edges along with associated vertices belonging to adjacent modules.

Extreme Sector Module. These are the sector modules that lie along the boundaries of the defined airspace.

Extreme Vertical Faces. These are the vertical faces of the extreme sector modules that form the boundary of the defined airspace.

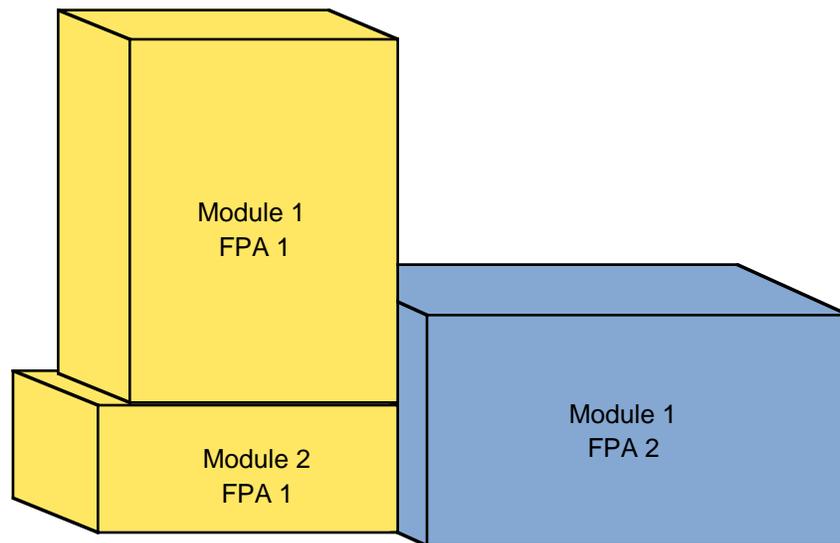


Figure 3.2 Typical Sector Geometry (showing a sector made up of 2 FPAs).

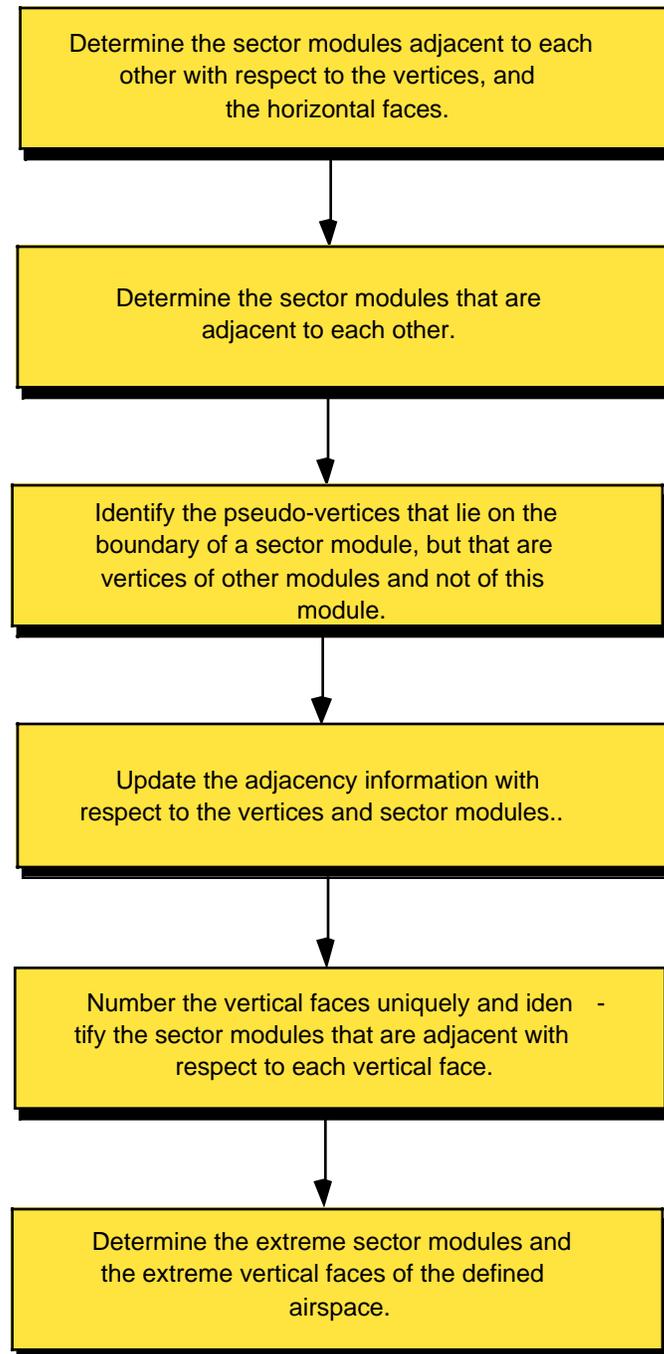


Figure 3.3 Occupancy Determination Flowchart.

3.6 Pre-processing Sector Data

The pre-processing of the sector data involves: 1) reading flight plans (or flight tracks if using SDAT derived data), 2) reading sector data (from the ACES database), and 3) converting the sector information into suitable mathematical representations to simplify the occupancy analysis. The analysis is initially done at the module level and later, the occupancy information is aggregated to the sector level.

All modules are represented in terms of their vertices, and the equations of the vertical faces (determined by the pre-processing routine) represented in the form $a \cdot X - c = 0$, where a is a normalized vector and c is the distance of the face from the origin in the direction of a . The adjacency information with respect to the faces and vertices is determined and stored during pre-processing.

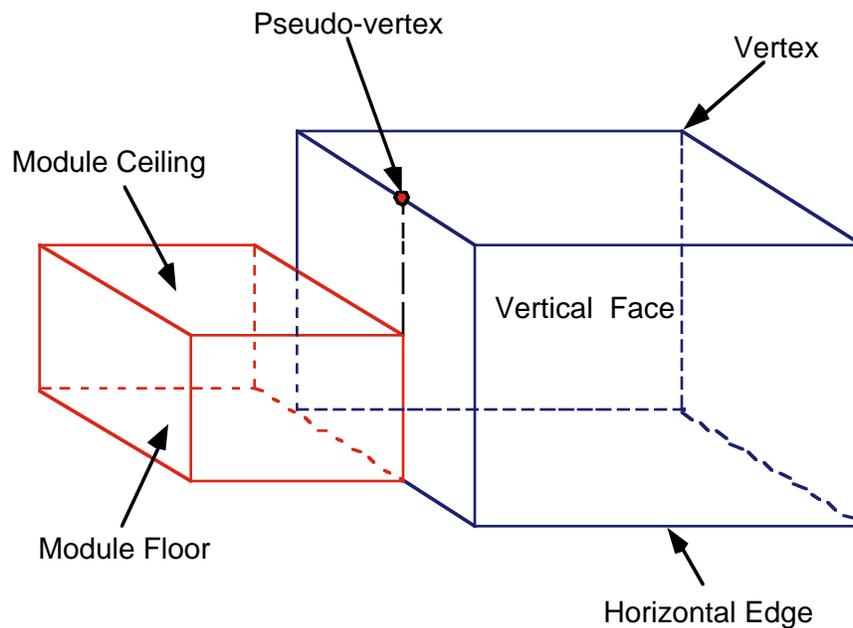


Figure 3.4 Geometric Components of a Sector Module.

3.7 Sector and Module Mathematical Definitions

Consider a two dimensional projection of a module. (A projection will always refer to a collapsing of the module in the vertical direction into a 2-D polygon.) An inward gradient F_{ps} for a face p of a projected sector module s is that gradient vector orthogonal to the face such that a trajectory which starts at an interior point of this face p and moves in a direction d , will reside in module s for some positive distance if and only if $F_{ps} \bullet d \geq 0$.

Examination of sector data derived from the FAA SDAT tool reveals coordinates of the vertices for all the modules in a clockwise sequence. Hence for any pair of vertices x_A and x_B defining the face p as shown in Figure 3.5, if the direction along the face is $d_p = x_A - x_B = [d_{p1}, -d_{p2}]$, then the inward gradient F_{ps} is given by $F_{ps} = [d_{p1}, -d_{p2}]$.

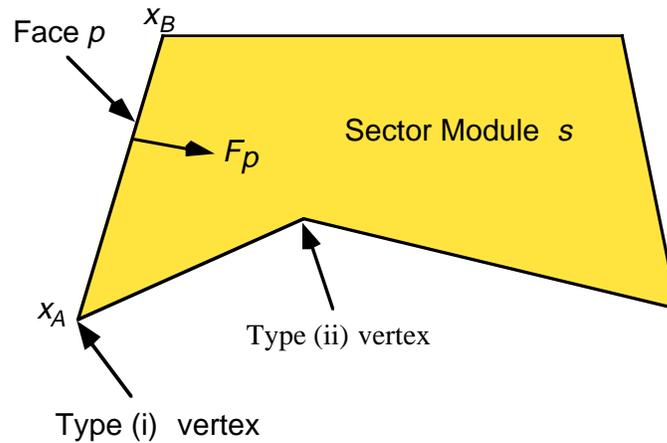


Figure 3.5 Basic Face and Vertex Definitions Inside a Sector Module.

3.7.1 Types of Vertices

Each vertex is classified as type (i) or type (ii), based on its associated faces p and q , as depicted in Figure 3.6. The following explanations help thereader to understand the mathematical differences between type (i) and type (ii) vertices.

Type (i): Here, the local neighborhood of the vertex is described by the conjunction of the faces p and q . Hence, if a trajectory starts at this vertex and moves in a direction d , then it would reside in module s for some positive step if and only if $F_{ps} \bullet d \geq 0$ and $F_{qs} \bullet d \geq 0$.

Type (ii): Here, the local neighborhood of the vertex is described by the disjunction of the faces p and q . Hence, if a trajectory starts at this vertex and moves in a direction d , then it would reside in module s for some positive step if and only if $F_{ps} \bullet d \geq 0$ or $F_{qs} \bullet d \geq 0$.

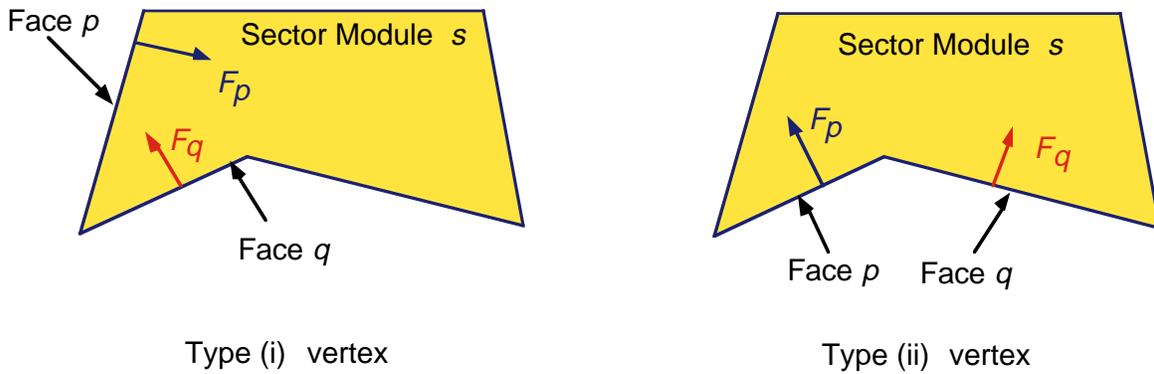


Figure 3.6 Types of Vertices.

3.7.2 Adjacency with Respect to Nodes

Consider a node V_m as shown in Figure 3.7, which might correspond to a real or a pseudo-vertex. All the sector modules which have V_m on the boundary of their two dimensional projections are considered to be adjacent with respect to V_m and are stored in the record Adjsecnode(m).sect. The pre-processing step will identify if there is any sector module s that contains the node V_m internally on a face of its two dimensional projection, and the program will then recognize s in terms of V_m and other defined nodes for s .

In Figure 3.7, the original nodes defining s_m are $[V_1, V_2, V_3, V_4, V_5, V_6, V_7]$. After preprocessing, the sector module s is redefined in terms of the nodes $[V_1, V_2, V_3, V_4, V_m, V_5, V_6, V_7]$. The sector modules $s, s_1, s_2,$ and s_3 will be considered to be adjacent with respect to V_m and this information will

be stored in the record `Adjsecnode(m).sect`.

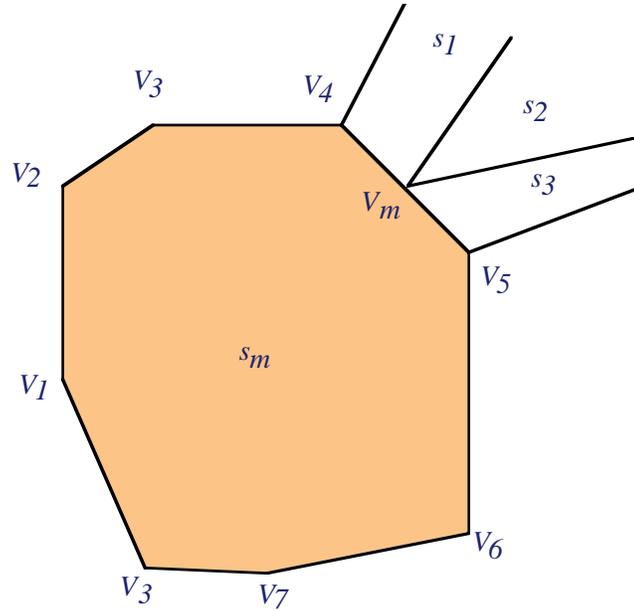


Figure 3.7 Adjacency with Respect to a Vertex.

3.7.3 Adjacency Information with Respect to Sector Modules

Sector modules adjacent to other sector modules are identified and stored during the pre-processing step. The adjacency information with respect to the nodes is used to identify this adjacency information. For a sector module s , let V_s be the set of nodes defining its floor and ceiling. Then, all the sector modules that share any V_s node in will be adjacent to s if they extend in part or whole over an altitude between the floor and ceiling of sector module s .

The main purpose of storing this adjacency information is to determine the nodes that lie around a projected sector module. Later, all nodes are checked to see if they lie on projected vertical faces of modules while not being defined as its original nodes.

3.7.4 Identifying Pseudo-Vertices

Consider the sector modules shown in Figure 3.8. Nodes V_2 and V_3 lie on the projected vertical face of module s , but are not defining nodes of the floor and ceiling of module s . Since the occupancy model makes use of the adjacency information in order to determine the next sector module into which a given flight enters after exiting a previous sector module, it becomes necessary to (a) identify nodes such as V_2 and V_3 as corresponding to pseudo-vertices of a sector module s and (b) to redefine its floor and ceiling faces in terms of all original, as well as such pseudo-vertex induced nodes.

In order to identify such nodes, a check is made for all the nodes lying around a sector module s to see if any lies on a projected vertical face of s . The nodes that lie around a sector module s are determined from the sectors that are adjacent to it.

Figure 3.9 illustrates an example of a pseudo-vertex in three dimensions. V_{m1} is a real vertex defining the floor and ceiling of the sector module s_1 . This induces a pseudo-vertex V_{m2} for the sector module s_2 . Both V_{m1} and V_{m2} correspond to the same node n_m and so sectors s_1 and s_2 will be considered adjacent with respect to node n_m .

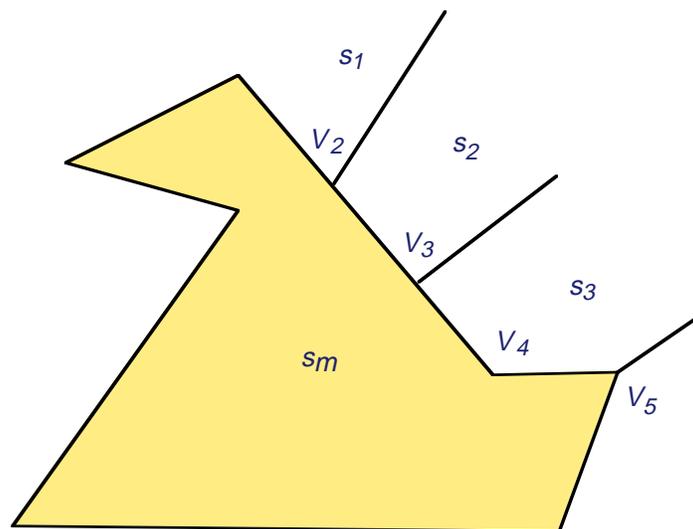


Figure 3.8 Adjacency with Respect to a Vertex.

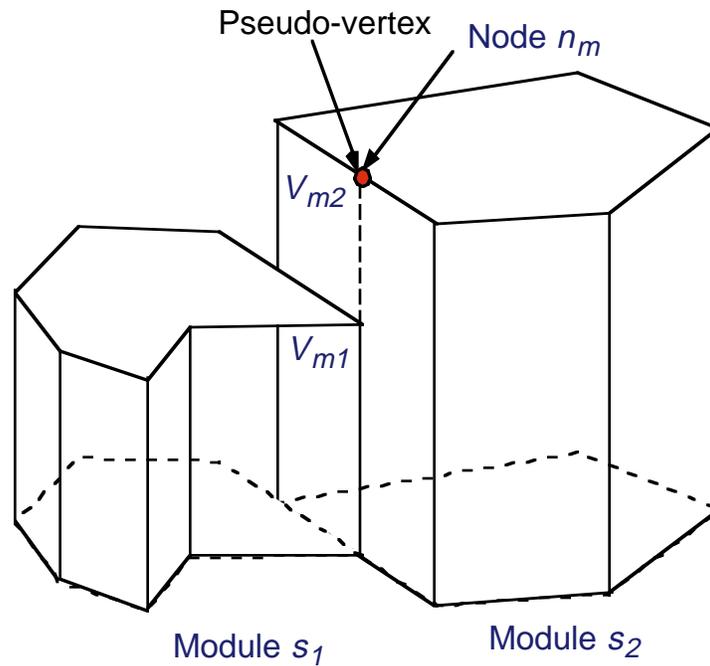


Figure 3.9 Pseudo-Vertices on a Face.

3.7.5 Adjacency Information with Respect to Vertical Faces

During the pre-processing step, the occupancy model stores sector modules that are adjacent to each other with respect to a given projected vertical face. This is done after identifying all the pseudo-vertices and revising the adjacency information of sector modules with respect to the nodes and modules. The projected vertical faces are distinguished from each other based on their defining end nodes. For any projected vertical face p having defining end nodes V_1 and V_2 , (including the pseudo-vertex induced nodes), all the sector modules that contain the nodes V_1 and V_2 are considered adjacent with respect to it. These sector modules can be determined from the adjacency information with respect to the nodes. The model also classifies the sector modules that are adjacent with respect to a particular vertical face into two categories based on whether the sector module lies on the side towards the origin (equator on Greenwich meridian) or on the side opposite to the origin. This additional information will be used to identify the extreme vertical faces. These extreme faces either define the external boundaries

of the defined airspace, or the vacuums that may be present in the airspace.

3.8 Pre-processing Airport Data

Airport data constitutes one of several inputs defining an aircraft's three dimensional trajectory. The pre-processing routine identifies each airport with a sector by checking if the airport lies in any of the low lying sector modules. The built-in Matlab function *inpolygon* is used to check if a point lies within a polygon. If the airport lies outside the defined airspace, no sector module will be associated with it.

3.9 Pre-processing Flight Plans

This pre-processing routine identifies the first sector module that a flight trajectory encounters. It also records the entry point and the time of entry. If the originating airport lies within the defined airspace, the identification process will be trivial. If the flight originates outside the defined airspace, the point of entry and the first module entered will be determined by checking each of the flight segments for a possible crossing of an extreme face of the defined airspace. Dummy sectors are defined in order to speed up the computations in this step. More details on dummy sectors are explained below.

3.9.1 Dummy Sectors

During the initialization step, the first module that each flight encounters is determined. If the origin airport does not lie in the defined airspace, then the program will move along the flight trajectory, segment by segment, to identify that flight segment that crosses any extreme face of the defined airspace. Since this is computationally intensive, dummy sectors are defined around the modeled airspace so that the airports of concern lie within this extended airspace. This cuts down the search during the initialization step drastically.

3.9.2 Vacuums

The dummy sectors defined around the defined airspace under consideration are trapezoidal polytopes.

Within the rectangular region formed by these sectors, that contains the airspace under consideration, exists an undefined airspace. This space is termed as the vacuum airspace. The program will handle the case of a flight passing through this vacuum and identify its entrance into the defined airspace, if at all or its re-entrance into the dummy trapezoidal polytopes. In addition to this deliberately created vacuum between the dummy sectors and the actual sectors under consideration, there may be instances of vacuums being present between actual sectors because of inaccuracies in sector definitions.

3.10 Sector Occupancy Determination Algorithm

The algorithm for determining sector module occupancies is first described for a projected two dimensional case. The same algorithm has been extended to handle three dimensions.

Consider a flight path that is comprised of linear discretized flight segments represented in terms of the coordinates of way-points. Such a flight path will be represented as $[wp_1, wp_2, \dots, wp_i, \dots, wp_n]$. Let any linear segment of the trajectory be defined as l_i for $i = 1, 2, \dots, n-1$.

Suppose that for wp_i we know the sector module s in which the current point lies, and its actual location in this sector module (interior point, interior of a face or at a vertex). This is initially determined during the pre-processing routine, and is sequentially deduced by the algorithm as explained below.

Initialization

Set $x_o = wp_i$; current point $x = x_o$; $\lambda = 0$ and $d = wp_{i+1} - wp_i$. Let s be the sector module in which x lies.

Step 1: Determination of Exit Point

Examine the faces of the sector s and find a first face that the straight line trajectory $x + \lambda d$ intersects (internally or at a vertex of a face) at $\lambda = \lambda^*$.

Let $\lambda_{new} = \lambda + \lambda^*$ and $x_{new} = x_0 + \lambda_{new}d$.

Go to Step 2.

Note that the occupancy of module s can continue in case we have just internally glanced a vertex, and

this will be automatically determined in the next loop of this procedure.

Step 2: Checking the Processing of Linear Segments

If $\lambda_{new} < 1$, record the occupancy in the interval $[\lambda, \lambda_{new}]$. Set $x = x_{new}$ and $\lambda = \lambda_{new}$, and proceed to Step 3.

Else, if $\lambda_{new} = 1$, record the occupancy in the interval $[\lambda, \lambda_{new}]$. Stop if $i + 1 = n$. Else, proceed to the next linear segment of the trajectory by incrementing i by 1 and moving to Step 3.

Else, if $\lambda_{new} > 1$, record the occupancy in the interval $[\lambda, 1]$. Stop if $i + 1 = n$. Else, proceed to the next linear segment of the trajectory by incrementing i by 1 and returning to Step 1.

Step 3: Search for the Next Sector Module

Determine the next sector module into which the flight enters based on the adjacency information and replace s by this module. Return to Step 1.

In this procedure, all the sector modules that the flight passes through are sequentially determined. The above algorithm makes an assumption that the flight will enter another sector module after it exits one. However, in the definition of the airspace, there may be a case where two neighboring sector modules may not be close enough to share any common vertex. This will result in an undefined airspace "vacuum" enclosed between such sector modules that the flight enters. To accommodate this case, we adopt the following strategy.

3.10.1 Extension of Algorithm to Handle Airspace Vacuums

The algorithm is extended to incorporate the scenario where a flight may encounter a vacuum in the airspace. During the pre-processing, the program identifies all the vacuums that are present in the airspace and stores the information regarding the vertical faces that surround such vacuums as explained in Section 3.9.2, if the program is not able to identify the sector module that the flight enters based on the adjacency information, it will realize that the flight has entered into a vacuum. The flight's segments are then checked to see when and if they cross any of the extreme faces. Based on the extreme face encountered, the program will identify the sector module entered and then proceed as usual.

As explained in the previous section, such instances of vacuums being present between sector modules occur mainly because of inaccuracies in sector definitions. In order to correct for inaccuracies, nodes having slightly perturbed locations are assumed to define the same point. This circumvents the creation of such vacuum airspace.

3.10.2 Extension of the Algorithm for the Three Dimensional Case

The foregoing algorithm is extended to the three dimensional case since all flights have flight plans or flight track data represented by latitude, longitude and altitude.

Initialization:

Set $x_o = wp_i$; current point $x = x_o$; $\lambda = 0$ and $d = wp_{i+1} - wp_i$. Let s be the sector module in which x lies.

Step 1: Determination of Exit Point

Examine the faces of the sector module s and find the first face (vertical or horizontal) that the trajectory $x + \lambda d$ intersects (internally, or at an edge, or at a vertex) at $\lambda = \lambda^*$. This procedure is explained in Section 3.10.

Let $\lambda_{new} = \lambda + \lambda^*$ and $x_{new} = x_o + \lambda_{new}d$.

Go to Step 2.

Step 2: Checking the Processing of Linear Segments

If $\lambda_{new} < 1$, record the occupancy in the interval $[\lambda, \lambda_{new}]$. Set $x = x_{new}$ and $\lambda = \lambda_{new}$, and proceed to Step 3.

Else, if $\lambda_{new} = 1$, record the occupancy in the interval $[\lambda, \lambda_{new}]$. Stop if $i + 1 = n$. Else, proceed to the next linear segment of the trajectory by incrementing i by 1 and moving to Step 3.

Else, if $\lambda_{new} > 1$, record the occupancy in the interval $[\lambda, 1]$. Stop if $i + 1 = n$. Else, proceed to the next linear segment of the trajectory by incrementing i by 1 and returning to Step 1.

Step 3: Determination of the Next Sector Module

Determine the next sector module into which the flight enters as explained in Section 3.10 and replace s by this module. Return to Step 1. If no new sector is encountered, proceed to Step 4.

Step 4: Determination of the Next Sector Module after passing through a Vacuum

Determine the next sector module into which the flight enters by checking for the intersection of the flight segments starting, with the current segment, with all the extreme faces of the defined airspace. Update x , λ and i based on the entry point. Return to Step 1. If no sector module is encountered until the last segment, (i.e when $i = n$), the flight terminates in a vacuum. Record this and stop.

3.10.3 Procedure to Determine Exit Points

Given a sector module s , the point x that lies in it, the parameter λ , and the direction d of the flight path at that point, proceed to identify whether the flight will terminate in this sector module, or else, determine the point at which the sector module s is exited. The following steps are followed for this purpose.

Step 1:

Identify the vertical faces p for which, $F_{ps} \bullet d < 0$. Among these vertical faces, the ones that are crossed internally or at the boundary by the flight segment are selected, and of these, the one that is closest to x is the face that may be crossed. Record λ_{new} and $x_{new} = x_0 + \lambda_{new}d$. Proceed to Step 2.

Step 2:

Check if x_{new} lies within the floor and ceiling of the sector module s . If not, identify the point on the floor or the ceiling that is crossed and update x_{new} and λ_{new} that correspond to this new point. Record the following:

- 1) If the sector module is crossed across the relative interior of a vertical face, record the vertical face crossed.
- 2) If the sector module is crossed across the relative interior of a vertical edge, record the vertical edge that is crossed.

- 3) If the sector module is crossed across the relative interior of a horizontal face, record the horizontal face that is crossed.
- 4) If the sector module is crossed across the relative interior of a horizontal edge, record the horizontal face and the vertical face that contains the edge.
- 5) If the sector module is crossed across a vertex, record the horizontal face and the vertical edge that contains the vertex.

3.10.4 Determination of the Next Sector Module after Exiting

When a flight trajectory is on the boundary of a sector module, it will either be located on the interior of a vertical face, at the interior of a horizontal face, on a vertical edge, on a horizontal edge, or at a vertex. A flight which exits a sector module in one of the above ways, will enter another sector module in one of the same five ways. Table 3.1 shows the thirteen possible piercing patterns in which an exiting flight can enter a new sector.

Based on the adjacency information and the type of exit, the probable sector modules into which the flight may have entered are selected. From these, the sector module s that satisfies one of the requirements below will be the one entered.

Case (a) : x belongs to the interior of a vertical face p of s and, then $F_{ps} \bullet d \geq 0$.

Case (b): x belongs to the interior of a vertical edge, as determined by faces p and q , and if the node corresponding to this vertical edge is of type(i) for sector module s , then we have $F_{ps} \bullet d \geq 0$ and $F_{qs} \bullet d \geq 0$, and if it is of type(ii), then we have $F_{ps} \bullet d \geq 0$ or $F_{qs} \bullet d \geq 0$.

Case (c): x belongs to the interior of the ceiling and the z component of d is nonpositive. Alternatively if x belongs to the interior of the floor, and the z component of d is nonnegative.

Case (d): x belongs to the interior of a horizontal edge, and requirements (a) and (c) are satisfied, where (a) is applied to the corresponding vertical face containing the edge.

Case (e): x is a vertex, and the corresponding requirements (b) and (c) are satisfied.

If more than one sector module is entered, as when a flight moves along a vertical face or along a horizontal edge, only one of such modules will be considered, with a preference given to the currently occupied module.

3.10.5 Determination of the Next Sector Module after Passing through a Vacuum

The pre-processing routine identifies the extreme faces of the defined airspace (including the faces of the dummy sectors). If a flight enters a vacuum (i.e the volume of airspace is not defined by the sectors), it will either re-enter into the defined airspace through one of the extreme faces or terminate in the vacuum. The sector module entered after passing through the vacuum will be determined by identifying the next extreme face that is crossed by the flight trajectory. If no extreme face is encountered, the flight terminates in the vacuum. Here an assumption is made that a flight enters a sector only through a vertical face after passing through a vacuum. This is a valid assumption as it is observed that the vacuums, wherever present, are always bounded by vertical faces alone.

Table 3.1 Sector Piercing Patterns.

		Point of Entry				
		Vertical Face	Vertical Edge	Top or Bottom Face	Top or Bottom Edge	Vertex
Exit Point	Vertical Face	◆			◆	
	Vertical Edge		◆			◆
	Top or Bottom Face			◆	◆	◆
	Top or Bottom Edge	◆		◆	◆	
	Vertex		◆	◆		◆

The Aircraft Encounter Model (AEM) is a computer model developed to estimate blind conflicts in the airspace under various concept of operations. AEM uses the outputs of AOM to determine all possible conflicts among aircraft pairs occurring in a prescribed volume of airspace. The main goal of AEM is to assess the precise geometry of conflicts between pairs of aircraft. AEM is expected to be used in airspace analyses as a screening tool to understand aircraft conflict patterns under new concept of operations. The FAA/Eurocontrol Collision Risk Modeling Group identified conflict geometry and scenario evaluation as one of the basic tasks to develop a toolbox of collision risk models. AEM is a first step in this direction.

The main blocks comprising AEM are shown in Figure 4.1. Two external blocks in this figure are inputs from AOM. These blocks, shown outside the dotted line boundary of AEM estimate: 1) sector occupancies and flight path structure and 2) adjacency information to locate spatial relationships between neighboring sector modules. The first major task in AEM is the extraction of flight proximity information. This is done through the creation of three data structures containing time, spatial and sector adjacency information.

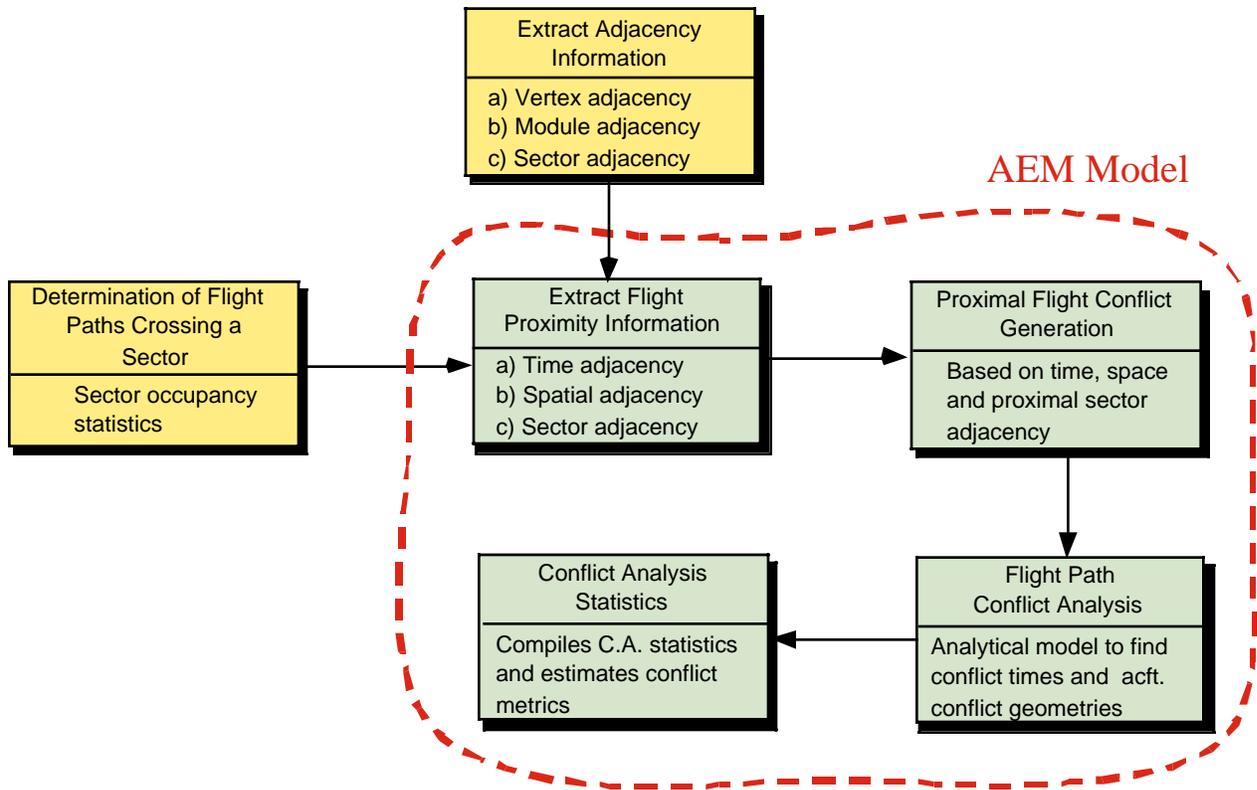


Figure 4.1 AEM Model Block Diagram.

The next block extracts proximal flights in time and space and initiates the flight conflict analysis. Once individual aircraft pairs are studied in detail using analytic trajectory equations, suitable conflict analysis statistics are collected and aggregated. This model has been coded in Matlab and can be executed in practically any operating system in use today without modifications.

An understanding of coordinate transformations is necessary to describe aircraft trajectories in flight. These trajectories are modeled using basic principles of spherical geometry. The following paragraphs provide some information about this issue.

4.1 Coordinate System Definitions and Transformations

Consider a point having a (Latitude, Longitude) = $(\alpha, \beta)^0$, where $-90^0 \leq \alpha \leq 90^0$ (being -90^0 at the south pole and 90^0 at the north pole) and where $0^0 \leq \beta \leq 360^0$, with the sweep of the vector in the horizontal plane occurring in an anticlockwise fashion when viewed from the north pole, as β goes from 0^0 to 360^0 . Figure 4.2 illustrates these angles for a point A on the surface of the earth.

Now, let us define a Cartesian system with the origin at O in Figure 4.2, with the x-axis oriented from O toward $(0, 0)^0$, the y-axis oriented from O toward $(0, 90)^0$ (orthogonal to the x-axis in the horizontal plane), and with the z-axis oriented from O toward $(90, 0)^0$ (vertically upward, where the longitudinal component for this can actually be arbitrary). Then, given $(\alpha, \beta)^0$, Figure 4.3 illustrates the Cartesian coordinates in (x, y, z) -space based on a transformation from the corresponding polar coordinate system, where R is the radius of the earth. This gives

$$(\alpha, \beta)^0 \rightarrow R(\cos \alpha, \cos \beta, \cos \alpha \sin \beta, \sin \alpha) \quad (4.1)$$

Remark 1. If an aircraft is located at $(\alpha, \beta)^0$ and at an altitude of h , its Cartesian coordinates are given by (4.1) with R replaced by $(R + h)$. Now consider two points $(\alpha_1, \beta_1)^0$ and $(\alpha_2, \beta_2)^0$. The straight line distance D between them is given by (4.2),

$$D^2 = R^2 [(\cos \alpha_1 \cos \beta_1 - \cos \alpha_2 \cos \beta_2)^2 + (\cos \alpha_1 \sin \beta_1 - \cos \alpha_2 \sin \beta_2)^2 + (\sin \alpha_1 - \sin \alpha_2)^2]$$

i.e.,

$$D^2 = 2R^2 [1 - \sin \alpha_1 \sin \alpha_2 - \cos \alpha_1 \cos \alpha_2 \cos(\beta_1 - \beta_2)] \quad (4.2)$$

Here, we have used the identity $\sin^2 \theta + \cos^2 \theta = 1$ for any angle θ , and also $\cos(\beta_1 - \beta_2) = \cos \beta_1 \cos \beta_2 + \sin \beta_1 \sin \beta_2$.

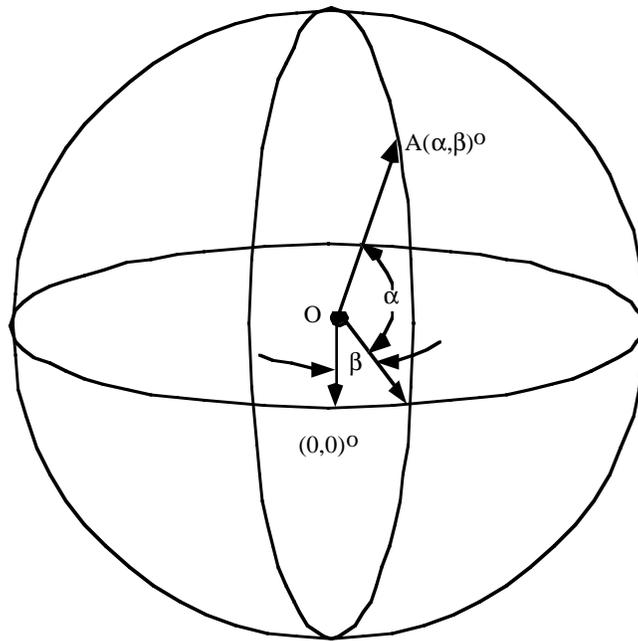


Figure 4.2 Definition of Latitudes and Longitudes.

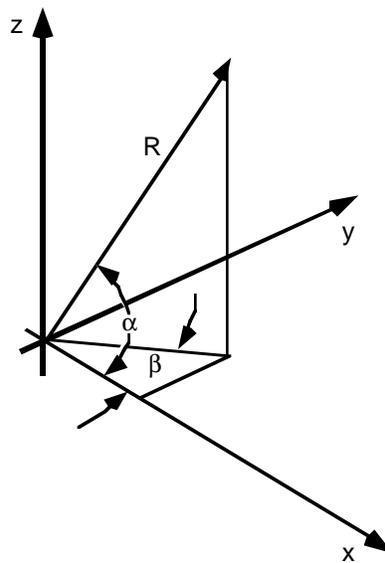


Figure 4.3 Polar To Cartesian Coordinate Transformation.

Hence, if we wish to determine the globe-circle distance between $(\alpha_1, \beta_1)^0$ and $(\alpha_2, \beta_2)^0$, this is given by $R\theta$ where θ (radians) is the angle subtended at the origin between the rays through $(\alpha_1, \beta_1)^0$ and $(\alpha_2, \beta_2)^0$. By the triangular cosine rule, we have

$$D^2 = 2R^2(1 - \cos \theta) \quad (4.3)$$

From (4.2) and (4.3), this gives

$$\theta = \cos^{-1}[\sin \alpha_1 \sin \alpha_2 + \cos \alpha_1 \cos \alpha_2 \cos(\beta_1 - \beta_2)].$$

4.2 Spherical Model

Consider any pair of aircraft A and B and suppose that their trajectories are known. Identify segments of durations (not necessarily of equal length) over which the trajectories of these aircraft are (approximately) linear, and assume that each aircraft is moving at a constant velocity over this duration. (The velocities might change from one duration segment to the next.) For any such time segment of duration T , let

$$x^A = \bar{x}^A + \lambda d^A \text{ for } 0 \leq \lambda \leq 1 \quad (4.4a)$$

and

$$x^B = \bar{x}^B + \lambda d^B \text{ for } 0 \leq \lambda \leq 1 \quad (4.4b)$$

denote the trajectories of aircraft A and B , respectively, where $x^A \in R^3$ denotes the coordinates of aircraft A , \bar{x}^A is its initial position and $\bar{x}^A + d^A$ is its final position over the given segment of duration T , and where the quantities for aircraft B are defined similarly.

Now, for the spherical model, let r_A denote the radius of aircraft A 's spherical *envelope* and R_A denote the radius of its spherical *proximity shell*. For example, R_A could be the total wing span of the aircraft A and R_A could be a somewhat larger quantity such that if the distance $d(A, B)$ between air-

craft A and B satisfies

$$d(A, B) \equiv \|x^A - x^B\| \leq \max\{r_A + R_A, r_b + R_A\} \quad (4.5)$$

then we say that a *conflict* has occurred, or that there exists a *conflict risk*. When

$$d(A, B) \equiv \|x^A - x^B\| \leq r_A + r_B \quad (4.6)$$

we will call this situation a *fatal conflict*. Using (4.4a) in (4.5) (or in (4.6)) it is easy to compute a duration over which (4.5) persists (if it holds at all), and this can be used to generate conflict constraints for each sector and each pair of aircraft as in the previous section.

4.3 Truncated Spherical Model

Suppose that we modify (4.5) to read

$$\text{If } d(A, B) \equiv \|x^A - x^B\| \leq \max\{r_A + R_B, r_B + R_A\}$$

and

$$\|X^A\| - \|X^B\| \leq S_3 \Rightarrow \text{there exists a conflict.} \quad (4.7)$$

Here, the absolute difference in altitude of the two aircraft is given by the left-hand side of the second in equality in (4.7), and S_3 is the standard imposed vertical separation parameter (say, 850 ft.). The advantage of (4.7) over (4.5) is that so long as a safe vertical separation is maintained, (4.7) does not trigger any conflict declarations. On the other hand, when the proximity shell radii are determined by in-trail or lateral separation standards that are relatively larger than the vertical separation criterion, (4.5) can raise too many false conflict alarms.

To implement (4.7), we simply need to examine the collision interval as determined above for (4.5), and then find that subinterval of this duration (if it exists) for which the additional vertical separation

violation criterion in (4.7) holds true.

Note that (4.7) does not distinguish between in-trail and lateral separation standards, which can be quite different in practice. This shortcoming can be overcome by using a box-model as described below, which further exploits linearity in its computations and permits a geometric classification of conflicts.

4.4 Box-Model for Aircraft Encounter Analysis

In this section, we consider a generalization of the box-model of Reich (1966) that examines rectangular envelopes and proximity shells as illustrated in Figure 4.4. Here, S_1 , S_2 , and S_3 , represent the proximity shell, respectively denoting the standard in-trail (along track), lateral (across track), and vertical separation parameters, and S_1' , S_2' and S_3' similarly represent a relatively tighter envelope for aircraft A. Note that the aircraft need not be centered in the box for the following type of analysis, although for simplicity in exposition, we assume this to be the case.

When an intruding aircraft B (treated as a point or "*particle*") enters the proximity shell, there exists a *conflict risk*. (The resulting conflict that must be resolved by FAA regulations is later designated as being of severity 1. An intermediate box of dimensions $S_i' = S_i/2$ for $i = 1, 2, 3$, could be defined, and a penetration into this box is designated as being of severity 2. A fatal conflict is labeled as being of severity 3) and occurs when an intruder penetrates a tight envelope having dimensions $S_1' = S_2' = 500$ ft., and $S_3' = 100$ ft. Each conflict of a given severity can be classified according to the actual (minimal) separation distance while the intruder is within the proximity shell, the duration of this intrusion, its entry and exit faces, and its relative heading with respect to aircraft A . The computation of such entities is discussed in the sequel. Note that the shell boxes are assumed to be also moving with the aircraft in the same direction of motion.

Now, consider a pair of aircraft A and B over a duration of time T for which the trajectories of these aircraft are described by (4.4a-b). For this duration, consider A as the focal *aircraft* and B as the *intruder*. (The roles of being a focal aircraft and an intruder can be reversed symmetrically while considering this same duration for the aircraft pair.)

The first task here is to transform the coordinate system from x -space to a convenient y -space representation via the affine transformation

$$x = \bar{x}^A + Qy \quad (4.8)$$

where Q is a nonsingular matrix having orthonormal columns and where the y -axis corresponds to the in-trail direction of motion (d^A) of aircraft A , the y_3 -axis is orthogonal to the y_1 -axis and lies in the plane spanned by d^A and the position vector \bar{x}^A emanating from the center of the earth (the origin) and with the positive direction making an acute angle with \bar{x}^A , and the y_2 -axis is orthogonal to the (y_1, y_3) plane (this represents the wing span, and we arbitrarily take the *positive* y_2 -axis to point to the left of the aircraft).

Accordingly, we obtain

$$Q = \begin{bmatrix} Q_1 & Q_2 & Q_3 \\ \hline \|Q_1\| & \|Q_2\| & \|Q_3\| \end{bmatrix}$$

where,

$$Q_1 = d^A, Q_2 = Q_3 \times Q_1, \text{ and}$$

$$Q_3 = \bar{x}^A - d^A \left(\frac{d^A \cdot \bar{x}^A}{|d^A|^2} \right). \quad (4.9)$$

Note that $Q_1 = d^A$ defines the in-trail direction, and Q_3 lies in the plane spanned by the vector from the origin (center of the earth) to the location of the aircraft A being orthogonal to d^A and making an acute angle with \bar{x}^A . Hence, Q_3 is given by the difference between the vectors \bar{x}^A and the projection of \bar{x}^A onto the normalized direction d^A . Note that we assume $d^A \neq 0$, and that \bar{x}^A and d^A are non-collinear (or else the aircraft would be moving vertically with respect to the earth's surface).

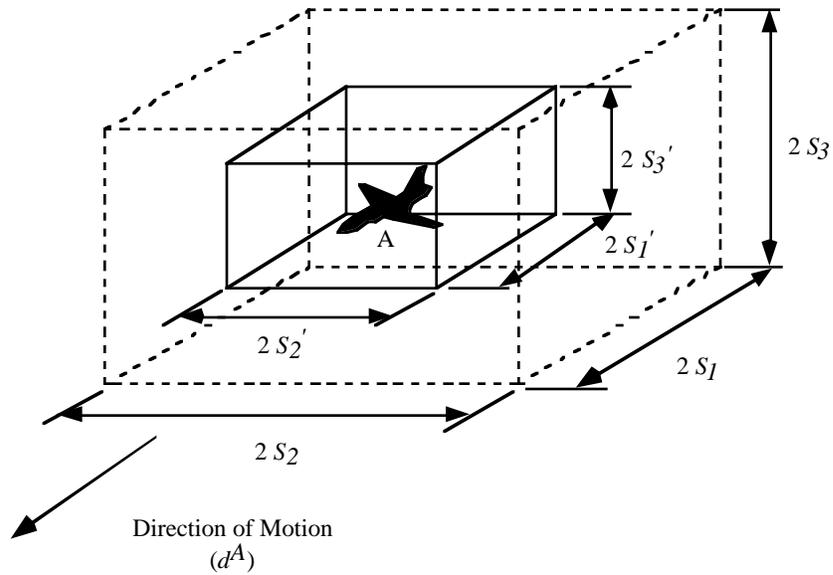


Figure 4.4 Envelope and Proximity Shell for Aircraft A.

Furthermore, we have written Q_2 as the cross-product of Q_3 and Q_1 following the right-hand cross product rule to ensure that the y_2 -axis points to the left of the aircraft. Hence,

$$Q_2 = \begin{bmatrix} Q_3(2)^* Q_1(3) - Q_3(3)^* Q_2(2) \\ Q_3(3)^* Q_1(1) - Q_3(1)^* Q_1(3) \\ Q_3(1)^* Q_1(2) - Q_3(2)^* Q_1(1) \end{bmatrix}.$$

Observe that since the columns of Q are orthonormal, we have $Q^{-1} = Q^t$. Consequently, in y -space, using (4.4) under the transformation (4.8), the trajectories of aircraft A and B are given by

$$y^A = \lambda Q^t d^A$$

and

$$y^B = Q^t \begin{pmatrix} -x^B & -x^A \end{pmatrix} + \lambda d^t d^B, \text{ for } 0 \leq \lambda \leq 1. \quad (4.10)$$

Consider now a box of dimension $2\delta_1 \times 2\delta_2 \times 2\delta_3$ centered at aircraft A and oriented along the y -axes, where, for example, $\delta \equiv S$ if we are considering the proximity shell based upon the standard separation criteria, and $\delta \equiv S'$ if we are considering some tighter envelope. Hence, as λ varies from 0 to 1, and the box in the y -space slides along the y_1 -axis, the (moving) aircraft (particle) B will lie in the box if and only if

$$-\delta \leq y^B - y^A \leq \delta, \text{ i.e.,}$$

$$-\delta \leq Q^t(\bar{x}^B - \bar{x}^A) + \lambda Q^t(d^B - d^A) \leq \delta. \quad (4.11)$$

The six inequalities in (4.11) (two for each dimension) define simple inequalities in the single variable λ , which when intersected with $0 \leq \lambda \leq 1$, will produce the restrictions

$$0 \leq \lambda_1 \leq \lambda \leq \lambda_2 \leq 1 \quad (4.12)$$

if this intersection is nonempty, or otherwise, will indicate that no conflict (of type determined by δ) occurs over this duration.

Given that a conflict occurs and that λ_1 and λ_2 in (4.12) are well defined, we will *classify* the conflict as being of

$$\text{Class } [\pm k_1, \pm k_2, \theta, \tau, d_{min}]_B \quad (4.13)$$

where B represents the intruding aircraft, and where the different entities in (4.13) are determined as follows. Later in Section 4.5 we will describe a metric which quantifies such a classification further by degree of severity. For $\lambda = \lambda_1$, we find the dimension k_1 (1, 2, or 3) for which the corresponding inequality in (4.11) is binding, using k_1 if this is the right-hand inequality for this dimension and $-k_1$ otherwise. If no inequality is binding (whence we must have $\lambda_1 = 0$), we use $k_1 = 0$. If there are ties in selecting the dimension k_1 , we break ties first in favor of a dimension that yields a nonzero λ -coefficient in the corresponding inequalities, and for continuing ties, we favor dimension 3 (vertical separation), then dimension 1 (in-trail separation), and lastly, dimension 2 (lateral separation). Hence,

the first entity in 4.13 designates the entry point of the intruder B within the box for aircraft A . If this entity is zero, then the conflict has been continuing since the previous segment because B lies in the interior of the box. Otherwise, $a + k_1(-k_1)$ indicates an entry via the positive (negative) k_1 -axis face, with ties broken according to the stated order based on the dimension for which a smaller separation is usually specified in practice in case the entry occurs on an edge (or a vertex) of the box.

Similarly, k_2 is defined with respect to $\lambda = \lambda_2$ and designates the face of exit (with ties broken as above), and where $k_2 = 0$ (whence λ_2 must be 1) if the intruder continues to lie in the interior of the box at the end of this duration segment.

The entity θ is a *relative heading angle* between the trajectories of aircraft A and B , and is given by

$$\theta = \cos^{-1} \left[\frac{d^A \bullet d^B}{\|d^A\| \|d^B\|} \right] \in [0, 180^0]. \quad (4.14)$$

The duration of intrusion over this time segment is given by

$$\tau = T(\lambda_2 - \lambda_1). \quad (4.15)$$

Note that for continuing consecutive segments of intrusion, the total duration of intrusion can be obtained by summing τ for the class vectors spanning from, $[\pm k_1 \neq 0, 0, \dots]_B, \dots$ to $[0, \pm k_2 \neq 0, \dots]_B$.

Finally, d_{min} denotes the minimum distance achieved between aircraft \bar{x}^A and \bar{x}^B over this duration segment. From (4.4), we have $\|x^B - x^A\| = \|\bar{x}^{BA} + \lambda d^{BA}\|$, where $\bar{x}^{BA} \equiv \bar{x}^B - \bar{x}^A$ and $d^{BA} \equiv d^B - d^A$. Hence, $d_{min} \equiv x^{-BA}$ if $d^{BA} = 0$, and otherwise, we have that $\|x^B - x^A\|^2$ is minimized when

$$\lambda = \bar{\lambda} \equiv \frac{-x^{-BA} \bullet d^{BA}}{\|d^{BA}\|^2}. \quad (4.16)$$

Since we must have $0 \leq \lambda \leq 1$ as well, and since $\|x^B - x^A\|$ is a convex function of λ we have,

$$d_{min} = \begin{cases} \bar{x}^{-BA} & \text{if } d^{BA} = 0 \\ \|\bar{x}^{-BA} + \lambda' d^{BA}\| & \text{otherwise} \end{cases} \quad (4.17a)$$

where, with $\bar{\lambda}$ given by (4.16),

$$\lambda' = \begin{cases} 0 & \text{if } \bar{\lambda} < 0 \\ 1 & \text{if } \bar{\lambda} > 1 \\ \bar{\lambda} & \text{otherwise.} \end{cases} \quad (4.17b)$$

Example 4.1

Consider the pair of aircraft A and B having trajectories as shown in Figure 4.5, where for some altitude α , we have

$\bar{x}^A = (0, 0, \alpha)$ with $d^A = (1, 1, 0)$, and

$$\bar{x}^B = (0, 1, \alpha) \text{ with } d^B = (3, -1, 0). \quad (4.18)$$

Hence, from (4.9), we obtain

$$Q_1 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, \quad Q_2 = \begin{bmatrix} -\alpha \\ \alpha \\ 0 \end{bmatrix}, \quad Q_3 = \begin{bmatrix} 0 \\ 0 \\ \alpha \end{bmatrix}$$

and so, we get

$$Q = \begin{bmatrix} 1/(\sqrt{2}) & (-1)/(\sqrt{2}) & 0 \\ 1/(\sqrt{2}) & 1/(\sqrt{2}) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad Q^{-1} = Q^t = \begin{bmatrix} 1/(\sqrt{2}) & 1/(\sqrt{2}) & 0 \\ (-1)/(\sqrt{2}) & 1/(\sqrt{2}) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (4.19)$$

Consequently, system (4.11) yields from (4.18) and (4.19) that

$$\begin{bmatrix} -\delta_1 \\ -\delta_2 \\ -\delta_3 \end{bmatrix} \leq \begin{bmatrix} 1/(\sqrt{2}) \\ 1/(\sqrt{2}) \\ 0 \end{bmatrix} + \lambda \begin{bmatrix} 0 \\ (-2)\sqrt{2} \\ 0 \end{bmatrix} \leq \begin{bmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \end{bmatrix}. \quad (4.20)$$

Hence, if $\delta_1 < 1/(\sqrt{2})$, no conflict occurs since (4.20) is then inconsistent. On the other hand, suppose that

$$\delta_1 = 1/(\sqrt{2}) \text{ and } \delta_2 = 1/(2\sqrt{2}). \quad (4.21)$$

Substituting (4.21) into (4.20) yields via the second dimension's inequalities that

$$-\frac{1}{2\sqrt{2}} \leq \frac{1}{\sqrt{2}} - 2\sqrt{2}\lambda \leq \frac{1}{2\sqrt{2}} \text{ or } \frac{1}{8} \leq \lambda \leq \frac{3}{8}. \quad (4.22)$$

Hence, in (4.12), we have $\lambda_1 = \frac{1}{8}$ and $\lambda_2 = \frac{3}{8}$. When $\lambda = \lambda_1 = 1/8$, the right-hand inequality of the second dimension is binding (note that the λ -coefficient for the first dimension is zero in breaking ties), while for $\lambda = \lambda_2 = 3/8$, the left-hand inequality of the second dimension is binding. Hence, the class of conflict in (4.13) is determined as follows, using Equations 4.14 through 4.18:

$$\text{Class} \left[2, -2, \cos^{-1}\left(\frac{1}{\sqrt{5}}\right), T\left(\frac{1}{4}\right), \frac{1}{\sqrt{2}} \right].$$

Here, the intruder B enters on the lateral face to the left of the aircraft A and exits on the opposite face of the box, the relative heading is along $\cos^{-1}(1/\sqrt{5})$, the duration of intrusion is for $T/4$, and the minimum distance achieved during this segment is $1/(\sqrt{2})$ which occurs when $\lambda = 1/4$.

Remark 2. As in Example 4.1, if only level enroute flights are being considered, then the above conflict analysis need only be conducted for flight pairs that are flying on altitudes that differ by less than or equal to a distance of S_3 .

Remark 3. For any given aircraft A, and for each linear segment it traverses, we have a corresponding matrix Q of Equation (4.9). The implementation used stores this matrix and its inverse to avoid repeated calculations when testing the conditions in (4.11) for various intruders B .

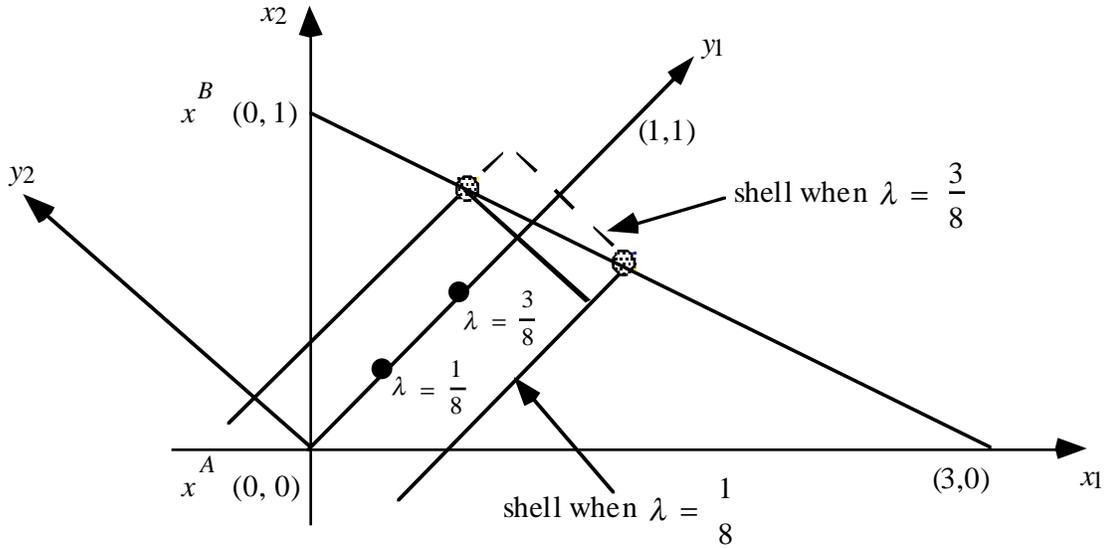


Figure 4.5 Trajectories of Aircraft A and B.

Remark 4. The foregoing analysis can just as well be conducted for various nonlinear envelopes and proximity shells. For example, we can consider cylinders having an ellipsoidal cross-section, with the major and minor axes oriented along the y_1 and y_2 axes, respectively, and (4.11) would reduce to examining suitable linear and quadratic equations in that can be readily derived because of the orientation of the axes in the transformed space. Also, as mentioned earlier, this type of an analysis can be readily conducted for the case where the aircraft is not necessarily centered in its envelope or proximity shell by suitably modifying the interval constraints in (4.11). For example, along the in-trail direction, a greater separation might be required ahead of the aircraft than behind it.

Remark 5. Similar to Remark 4, we could also consider nonlinear aircraft trajectories defined by some parametric (smooth) curves $x^A(\lambda)$ and $x^B(\lambda)$ for $0 \leq \lambda \leq 1$, where the linear trajectories in (4.4) are

a special case. Given any $\lambda \in [0, 1]$, the vector $d^A(\lambda) = \frac{d}{d\lambda}x^A(\lambda)$, where the derivative is taken by each component, gives the instantaneous direction of aircraft A. Define the instantaneous y-space transformed coordinate system as in (4.7), where Q is given by (4.8) with $d^A \equiv d^A(\lambda)$ and with \bar{x}^A replaced by $x^A(\lambda)$ (call this $Q(\lambda)$). Then, as in (4.11), the intruder B lies in the box at the instant determined by λ if and only if,

$$-\delta \leq Q(\lambda)^t [x^B(\lambda) - x^A(\lambda)] \leq \delta. \quad (4.23)$$

Note that (4.11) is a special case of (4.23). However, while the linear inequalities in (4.11) yielded a simple solution (4.11), (4.23) involves finding a solution to a nonlinear system of inequalities, albeit in a single variable λ .

Example 4.2.

Suppose that we have the following initial positions and directions in terms of λ :

$$\begin{aligned} \bar{x}^A &= (0, 0, \alpha) \text{ with } x^A(\lambda) = (\lambda, 0.5\lambda^2, \alpha), \text{ so that } d^A(\lambda) = \frac{d}{d\lambda}x^A(\lambda) = (1, \lambda, 0) \\ \bar{x}^B &= (0, 1, \alpha) \text{ with } x^B(\lambda) = (\lambda, 1 - 0.5\lambda^2, \alpha), \text{ so that } d^B(\lambda) = \frac{d}{d\lambda}x^B(\lambda) = (1, -\lambda, 0). \end{aligned}$$

Then we have from Remark 5 and (4.9) that,

$$Q_1 = \begin{bmatrix} 1 \\ \lambda \\ 0 \end{bmatrix}, \quad Q_3 = \begin{bmatrix} 0.5\lambda^3/(1+\lambda^2) \\ -0.5\lambda^2/(1+\lambda^2) \\ \alpha \end{bmatrix}, \quad Q_2 = \begin{bmatrix} -\alpha\lambda \\ \alpha \\ 0.5\lambda^2 \end{bmatrix}$$

and so, letting $C = \sqrt{(1/4)\lambda^4 + \alpha^2 + \alpha^2\lambda^2}$ and $D = \sqrt{1 + \lambda^2}$, we get

$$Q(\lambda) = \begin{bmatrix} 1/D & -\alpha\lambda/C & 0.5\lambda^3/CD \\ \lambda/D & \alpha/C & -0.5\lambda^2/CD \\ 0 & 0.5\lambda^2/C & \alpha D/C \end{bmatrix}, \quad Q(\lambda)^t = \begin{bmatrix} 1/D & \lambda/D & 0 \\ -\alpha\lambda/C & \alpha/C & 0.5\lambda^2/C \\ 0.5\lambda^3/CD & -0.5\lambda^2/CD & \alpha D/C \end{bmatrix}.$$

Now, letting $\alpha = 1$ be a unit measure with respect to the defined Cartesian system, and using (4.23), we have that the intruder B penetrates the box enveloping A of size 2δ at the instant determined by λ if and only if,

$$\begin{bmatrix} -\delta_1 \\ -\delta_2 \\ -\delta_3 \end{bmatrix} \leq (1 - \lambda^2) \begin{bmatrix} \lambda/D \\ 2/(\lambda^2 + 2) \\ -\lambda^2/D(\lambda^2 + 2) \end{bmatrix} \leq \begin{bmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \end{bmatrix}.$$

Suppose that $\delta_1 = 1/4$, $\delta_2 = 1/4$ and $\delta_3 = 1/8$. Then, it is verifiable that for $\lambda \in [0, 0.282] \cup [0.768, 1]$, the following condition holds for the x-coordinate:

$$-1/4 \leq \frac{\lambda(1 - \lambda^2)}{\sqrt{1 + \lambda^2}} \leq 1/4,$$

for $\lambda \in [0.816, 1]$, the following condition holds for the y-coordinate:

$$-1/4 \leq \frac{2(1 - \lambda^2)}{\lambda^2 + 2} \leq 1/4,$$

and for $\lambda \in [0, 1]$, the following condition holds for the z-coordinate:

$$-1/8 \leq \frac{-\lambda^2(1 - \lambda^2)}{\sqrt{1 + \lambda^2}(\lambda^2 + 2)} \leq 1/8.$$

Thus, the two aircraft will be in conflict for $\lambda \in [0.816, 1]$.

Remark 6. Another related issue is the piecewise linear discretization of the aircraft trajectories. Suppose that relatively large segments of this trajectory follow some circular path with radius R and that

we would like the maximum error in a linear chord approximation to be bounded by e_{max} , which might be some acceptable fraction of the separation standard in the plane containing the circular trajectory and its center (see Figure 4.6).

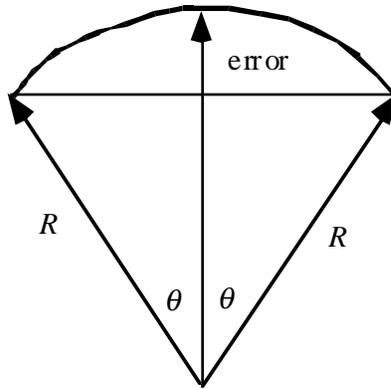


Figure 4.6 Circular Aircraft Trajectory Term Definitions.

Hence, the error given by $R(1 - \cos\theta)$ should be no more than e_{max} , which gives

$$\cos\theta \geq \left(1 - \frac{e_{max}}{R}\right) \text{ or } \theta \leq \cos^{-1}\left(1 - \frac{e_{max}}{R}\right). \quad (4.24)$$

The distance $S = 2R\theta$ traversed by the aircraft between breakpoints should therefore be bounded by

$$S \leq 2R \cos^{-1}\left(1 - \frac{e_{max}}{R}\right). \quad (4.25)$$

4.5 An Aggregate Metric for Conflict Severity

Some metrics used in previous studies include the Kip Smith Metric and the Laudeman Metric as described in Suchkov, et al. (1997). The Kip Smith Metric identifies separation as the single most important factor in estimating collision risk. It uses the number of aircraft, the distance between flights i and

j at time t (not separated by altitude) and an empirical factor to establish a measure of workload. The Laudeman metric incorporates nine traffic factors, using two-minute time increments and a twenty-minute projection of future aircraft positions. This metric attempts to compartmentalize workload as a series of time-space counts.

In contrast, the metric for the present study provides a measure of several aspects impacting the difficulty of conflict resolution by air traffic controllers. During the conflict analysis, three imaginary protective boxes are constructed around the primary aircraft. The first is the outer protective box (proximity shell) used to determine the presence of a conflict based on standard separation criteria. The second is an intermediate box with each dimension measuring half that of the first. The third is a tight box measuring 500 ft. in front of and behind the aircraft, 500 ft. to the left and right of the aircraft, and 100 ft. above and below the aircraft. The severity of a conflict is measured by placing each conflict into one of three possible severity classes based on the smallest box pierced. A conflict falls into severity class 1 if the intruding aircraft pierces the outer protective box (but not the other two boxes), severity class 2 if it pierces the intermediate box (but not the inner box), and severity class 3 if it pierces the inner box.

The metric used for this study is a vector describing the number of conflicts for each severity class within the region under consideration, the duration of the conflict within each severity class, and the percent rate of convergence for each severity class, where the last two measures are computed only for severity classes 1 and 2 because conflicts of severity class 3 are untenable and require no further quantification. This metric is given by

$$(N_1, L_1, R_1, N_2, L_2, R_2, N_3),$$

where N_k is the number of pairwise conflicting aircraft of severity at most k , and the measures L_k and R_k , respectively, describe average durations and percent rate of convergence of conflicts within class k as described below.

The average length of conflict durations for severity k , L_k , is the sum of the durations for which a conflict of severity (at most) k occurs normalized by N_k . That is,

$$L_k = \sum_{(i, j) \text{ of severity } k} l_k(i, j) / N_k \quad \text{for } k = 1, 2$$

where $l_k(i, j)$ is the duration of the conflict of severity k between flights i and j .

For flights i and j , define x^i and x^j to be the positions at which they are first at their minimum distance, and define d^i and d^j to be the negative normalized directions along which these flights approach at their respective positions x^i and x^j , so that for any sufficiently small duration t prior to the time at which the minimum distance is realized, the position of aircraft $a = i$ or j is given by $x^a + td^a$. The distance between the two conflicting aircraft of severity class k at such a time t , as a percentage (fraction) of the minimal distance achieved, is given by

$$r_{k(i, j)}(t) = \frac{\|(x^i + td^i) - (x^j + td^j)\|}{\|x^i - x^j\|} = \frac{\sqrt{\|x^i - x^j\|^2 + t^2\|d^i - d^j\|^2 + 2t(x^i - x^j) \bullet (d^i - d^j)}}{\|x^i - x^j\|}.$$

The derivative of $r_{k(i, j)}(t)$ evaluated at $t = 0$ therefore yields the instantaneous rate at which the percentage gap between the two aircraft is closing along their approach to the minimal separation point, and is given by:

$$\dot{r}_{k(i, j)}(0) = \frac{(x^i - x^j) \bullet (d^i - d^j)}{\|x^i - x^j\|^2}.$$

Note that if the percentage distance between the aircraft is a smooth function of time, then since the minimal distance is achieved at $t = 0$ by definition, we would therefore necessarily have $\dot{r}_{k(i, j)}(0) \equiv 0$. However, if this percentage gap function is nondifferentiable at the instant of minimal separation due to a breakpoint in at least one trajectory, then $\dot{r}_{k(i, j)}(0)$ is the (negative of) the left-hand derivative of this function with respect to time at this instant. Also, note that this measure is precisely the inverse of the limiting value of the Tau metric as the instant of minimal separation is approached.

It is possible that two aircraft i and j may be at their minimum separation point before they enter the

area under consideration. The two aircraft may then be diverging when they enter the area under consideration, in which case $\dot{r}_{k(i,j)}(0) < 0$. In this event, we take the maximum of $\dot{r}_{k(i,j)}(0)$ and zero in computing R_k . A zero value is appropriate in this case because such diverging flights resolve their own conflict without controller intervention. Furthermore, this avoids canceling the effect of other such values in the aggregate metric. Having computed $\dot{r}_{k(i,j)}(0)$ in this fashion for all conflicting pairs i and j in severity class (at most) k , we compute

$$R_k = \sum_{(i,j) \text{ of severity } k} \max\{0, \dot{r}_{k(i,j)}(0)\} / N_k \quad .$$

Example 4.3.

Consider the situation in Figure 4.7. For Case I, let $(x^a - x^b) = [0, 2]$ and $(d^a - d^b) = [0, 3]$, and for Case II, let $(x^a - x^b) = [0, 1]$ and $(d^a - d^b) = [0, 3]$. Note that although the rates of convergence for Case I and Case II are the same, the percentage rate of convergence is not. That is, for Case I, $\dot{r}_{k(a,b)}(0) = 1.5$, and for Case B, $\dot{r}_{k(a,b)}(0) = 3$, where the latter reflects a relatively more critical situation in which the aircraft are converging at the same rate but with a lesser minimal separating distance.

Example 4.4

Consider Figure 4.8, where Flights a and b are on parallel tracks heading in opposite directions. Since $(d^a - d^b) = [-2, 0]$, $(x^a - x^b) \bullet (d^a - d^b) = 0$, we get $\dot{r}_{k(a,b)}(0) = 0$. Note that a zero value for this rate measure is appropriate since the instantaneous rate at which the percentage gap between the two aircraft is closing at the point of minimal separation is zero.

4.6 Summary of Sector Conflict Analysis

Following the procedures described in Chapter 3, the sector occupancy durations are first computed for each flight. A list of sectors entered by each flight is compiled, along with the times for which the flight enters and exits these sectors. Also, for each sector a list is compiled of all flights which traverse that sector, along with their entering and exiting times. This information is then used in the following pre-

processing steps.

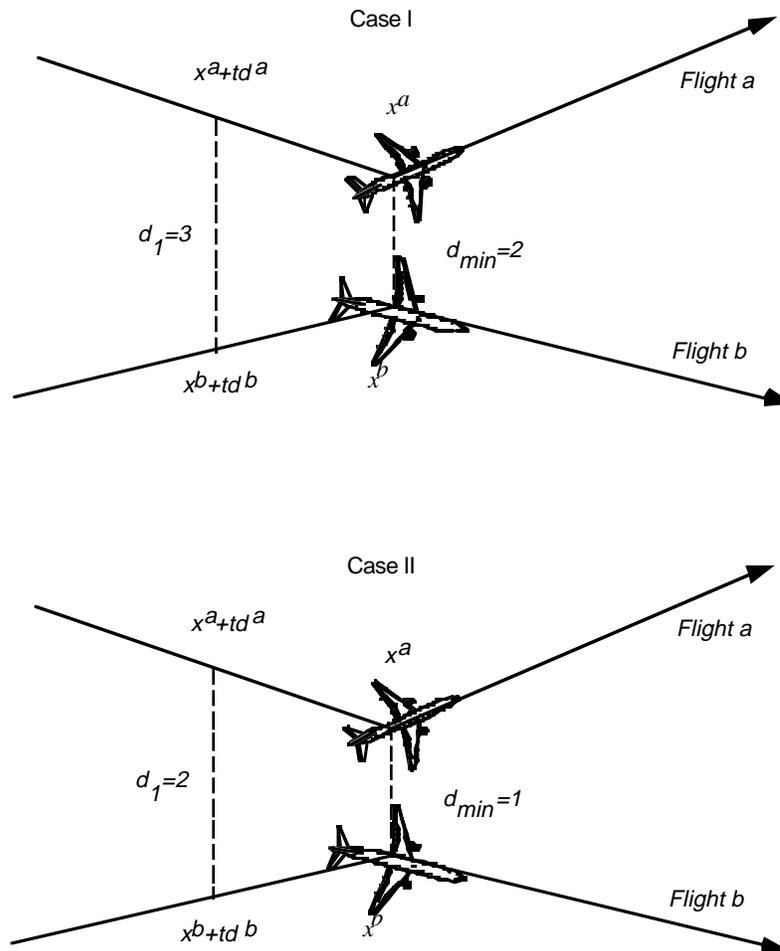


Figure 4.7 Example to Illustrate the Effect of Using Percent Rates.

Since testing each distinct pair of flights for conflicts is computationally expensive, logical tests are performed to eliminate pairs of flights which cannot conflict. A preprocessing is therefore conducted to determine all pairs of flights which occupy the same sector or adjacent sectors at the same time. These flights are recorded for performing a more detailed conflict analysis during the intervals in which they may possibly conflict.

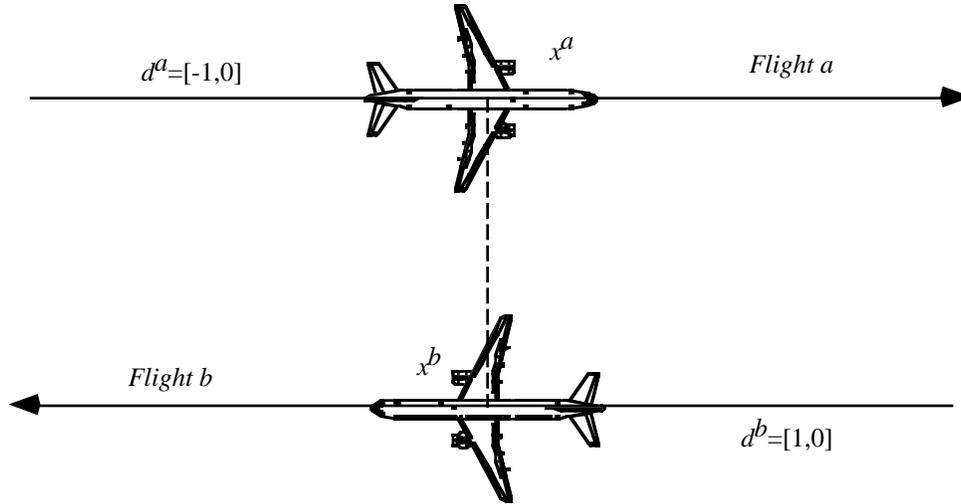


Figure 4.8 Example of Parallel Flights with Opposite Headings.

For each flight i in sector s , let $I^s(i) \equiv [d^s(i), a^s(i)]$ denote the interval between the entering and exiting time for i in s . Only flights that occupy s or the sectors neighboring s for a time interval overlapping $I^s(i)$ may conflict with i . For each sector s , a set of neighboring sectors is specified such that the only possible conflicts that can occur with a flight that occupies sector s are with respect to flights that simultaneously occupy some sector in this set of neighbors. These neighboring sectors are found by constructing a rectangular box which encompasses s plus a buffer area such that if a flight does not lie within this box, it may not conflict with a flight in s .

A rectangle is constructed around the two-dimensional cross section of s and then extended into three dimensions by examining the floor and ceiling of s .

First, the geometric center c of s is found (by taking the average of the defining vertices of s), and the largest distance from c to any vertex of s is determined. This longest distance becomes half of the length of the rectangle, with the other half extending in the opposite direction from the center. Each vertex is then examined on either side of the line that passes through c and is parallel to the side of the rectangle that defines its length. The rectangle is then widened as necessary on either side of this line

to include each vertex (see Figure 4.9). This rectangle, which encloses all the defining vertices of s , is then enlarged to include the buffer space, which should be the distance from the center of the protective box enveloping the largest aircraft to one of its corners. The protective box used is the one based on the standard separation criteria. Finally, the floor of this rectangle is set at the maximum of zero and the floor of the sector minus the buffer space, and the ceiling is set at the ceiling of the sector plus the buffer space.

Once this rectangular box has been constructed, any sector intersecting this box is included in the set of neighbors of s . Each defining vertex of a sector is tested for its inclusion within the two-dimensional rectangle. If a vertex is found to be within this rectangle, a separate check is performed to determine if it also lies within the floor and ceiling of the rectangular box (see Figure 4.9). For any vertex v which is found to meet these criteria, all sectors which include v on their boundaries are included in the set of neighbors of s .

Hence, for each sector r equal to or neighboring sector s , and for any other flight plan j , if j exits r before i enters s ($a^r(j) < d^s(i)$), or if i exits s before j enters r ($a^s(i) < d^r(j)$), flights i and j are not airborne in a close vicinity of each other at the same time, and need not be considered in the conflict analysis. Otherwise, the interval during which a conflict may exist, C , is computed, and a conflict analysis for flights i and j is performed over C . The record $[i, j, I^s(i)]$ is added to CA , which comprises the list of flights and durations for which a conflict analysis is to be performed.

The overall preprocessing procedure is stated below.

For each flight i

 For each sector s traversed by i

 Let $N(s) = \{s \cup \text{neighboring sector of } s\}$

 For each sector r in $N(s)$

 For each flight j occupying r

 If $a^r(j) \geq d^s(i)$ and $d^r(j) \leq a^s(i)$

 let $C = [\max\{d^s(i), d^r(j)\}, \min\{a^s(i), a^r(j)\}]$

 Add $[i, j, C]$ to CA

 end if

 end for

 end for

 end for

 end for

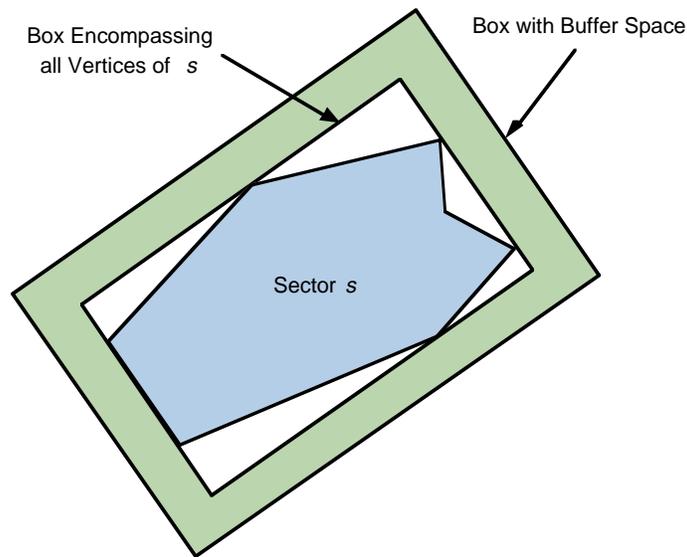


Figure 4.9 Illustration of 2-D Rectangle Created for Neighboring Sector Analysis.

Following this preprocessing, the list CA is passed to the conflict analysis routine developed in the previous sections. A conflict analysis is performed on each pair of flights for the given times in which the flights may possibly conflict. The conflict analysis routine considers the flights along linear trajectories between the union of their way-points. Since conflicts are not considered below FL100, and since the size of the protective box changes at FL290, extra way-points are created at these altitudes if necessary (along the corresponding linear segments) that pierce these altitudes.

Each entry of CA is considered independently, with each possible conflict being passed to the conflict analysis routine. For a given entry $[i, j, I^s(i)]$ of CA, the conflict analysis routine inserts the extra way-points at FL100 and FL290, and also at the beginning and ending times of $I^s(i)$. The conflict analysis considers each linear segment between way-points traversed during the interval $I^s(i)$. For a given pair of flight segments, if the altitude of either aircraft is below FL100 or if the two aircraft are sufficiently separated by altitude (see Remark 1), then no analysis is done for that pair of segments. Otherwise, the procedure determines the size of the protective box around the primary aircraft based on the altitude of the primary aircraft, and a detailed analysis begins. Using the axis transformation in (4.9)

and checking the conditions in (4.11) (along with the computation conserving techniques of Remark 2), the detailed conflict analysis procedure indicates whether or not a conflict exists, and reports the class of the conflict as in (4.13). Note that although CA only lists potentially conflicting aircraft i and j such that $i < j$, the conflict analysis must be performed twice, considering each aircraft as the primary aircraft.

The resulting output is sorted first by primary aircraft, next by secondary aircraft, and finally by the starting time of conflict to obtain a list describing the ongoing conflicts encountered by each aircraft. Note that for conflicting flights i and j , there may be many records describing the same conflict if the conflict continues over several linear segments. The overall conflict between i and j may be summarized by conglomerating all consecutive records of conflicts between i and j such that the ending time of one record corresponds to the beginning time of the next record. For this set of records, the maximum conflict severity, minimum separating distance, direction of flight while approaching the minimum separating distance, and the overall length of conflict duration are recorded and used to compute the overall aggregate metrics.

5.1 NARIM Scenarios

To test AOM and AEM, several FAA developed airspace scenarios were used. These scenarios represent a natural progression from current conditions (i.e., 5-7% National Route Program use) to three dimensional Free Flight (i.e., wind optimized cruise climb trajectories). At the time of preparation of this study the research team only had access to NAS traffic demand scenarios for 1996. These represent baseline conditions used by FAA according to the National Airspace Resource and Investment Model (NARIM). Recently, the FAA has developed future demand scenarios for horizon years 2010 and 2015, using the same basic assumptions.

The NAS traffic demand scenario database represents typical NAS conditions for five days of the year using 1996 ETMS traffic data as baseline. Each scenario or operational concept as defined in the NARIM program literature (CSSI, 1998) uses different wind patterns that capture seasonal variations in the jet stream. These scenarios were generated by CSSI using a combination of the Future Demand Generator and OPGENTM. OPGEN is an optimization model developed by CSSI that estimates flight trajectories between an origin and a destination airport using variable ATC rules, aircraft performance parameters and wind conditions. This model optimizes individual flight tracks above FL100. The ter-

minal airspace trajectory maintains a preferred arrival or departure pattern and is therefore based on the assumption that terminal airspace congestion precludes the use of optimal routes in class B airspace.

Another important assumption made relates to the optimization mode used. Flights longer than 1000 nautical miles (nm) were fully optimized subject to the constraints of the corresponding concept of operations (e.g., Wind-optimized routing with hemispherical rules) (CSSI, 1998). Shorter flights, less than 1000 nm were "straightened" subject to SUA constraints and placed on RVSM altitudes or where appropriate (i.e., above FL290) (CSSI, 1998).

Overall there are six operational scenarios proposed by the FAA to research transition to the concept of Free Flight. The following paragraphs summarize the ATC rules and wind conditions considered for each NAS traffic demand scenario.

5.1.1 National Airspace (NAS) Concept of Operations

This scenario represents 1996 traffic conditions for NAS. The trajectories are based on the flight plans filed by the airlines. This scenario includes mostly fixed route flight plans using the high altitude airway system in the US and consequently relies on ground based Navigational Aids (NAVAIDS) such as Very High Frequency, Omni-Directional Range instruments (VOR).

5.1.2 Wind-Optimized Routing with Hemispherical Rules and Assigned Altitudes (Cardinal_Asg)

This scenario reflects the removal of reliance on the ground-based NAVAIDS but retains the current directional flight levels. The altitudes for these routes are filed flight altitudes, and reside among the following levels as required under the current concept of operations.

Westbound Flights Levels	Eastbound Flights Levels
FL180 to FL290 at intervals of 2,000 ft beginning at FL180	From FL180 to FL290 at intervals of 2,000 ft beginning at FL190

Westbound Flights Levels	Eastbound Flights Levels
Above FL290 at intervals of 4,000 ft beginning at FL310	Above FL290 at intervals of 4,000 ft beginning at FL290

5.1.3 Wind-Optimized Routing with a Reduced Vertical Separation and Assigned Altitudes (RVSM_Asg)

This is an extension of the previous case that considers reduction in the restriction on vertical flight separation levels. The minimum vertical separation between the flight plans is reduced to 1000 feet across the complete Class A Airspace. Each altitude is assigned to lie at one of the following which is closest to the filed level.

Westbound Flights Levels	Eastbound Flights Levels
At intervals of 2,000 ft beginning at FL180	At intervals of 2,000 ft beginning at FL190

5.1.4 Wind-Optimized Routing with Hemispherical Rules (Cardinal)

This is similar to the second scenario (Cardinal_Asg) except that the flight levels are based on the aircraft performance. Cardinal flight altitudes apply, and the altitudes for this route belong to the following set of levels.

Westbound Flights Levels	Eastbound Flights Levels
FL180 to FL290 at intervals of 2,000 ft beginning at FL180	From FL180 to FL290 at intervals of 2,000 ft beginning at FL190
Above FL290 at intervals of 4,000 ft beginning at FL310	Above FL290 at intervals of 4,000 ft beginning at FL290

5.1.5 Wind-Optimized Profiles with Reduced Vertical Separation Criteria (RVSM)

This is similar to the previous scenario except that the flight levels in this scenario adopt the reduced

vertical separation rules, and belong to the following set of levels.

Westbound Flights Levels	Eastbound Flights Levels
At intervals of 2,000 ft beginning at FL180	At intervals of 2,000 ft beginning at FL190

5.1.6 Wind-Optimized Profiles without Hemispherical Rules (Climb-Cruise)

This scenario reflects a complete relaxation of the stated restrictions. The trajectories are not constrained by the ground-based navigation aids or the flight levels, or the current cardinal altitude rules, or the vertical separation standards. The profiles represent complete cruise-climb in the enroute airspace and with restrictions in the terminal area (CSSI, 1998).

Of these six scenarios provided by the FAA three were selected for further investigation in this study:

- a) Current National Airspace (NAS) Concept of Operations,
- b) Wind-Optimized Profiles with a Reduced Vertical Separation Method (RVSM), and
- c) Wind-Optimized Profiles without Hemispherical Rules (Cruise-climb).

The NARIM Concept of Operations database contains 18,000 flight plans per day in the baseline year (1996). All of these flights are flights cruising above FL 240. To restrict the number of flights analyzed in both AOM and AEM data sets of 8000 and 6000 flights were used as representative of the conditions expected at various ARTCC Centers analyzed. Since each RVSM and Cruise Climb scenario was derived from the baseline condition selecting the first 8000 flights does not introduce any significant bias of traffic between Origin-Destination (O-D) airport pairs. This is important in the study of sector occupancies to maintain a homogeneous flight data structure to derive valid traffic pattern conclusions. Four ARTCC Centers were selected for this study to restrict the number of runs to a reasonable level. Table 5.1 illustrates the case studies scrutinized in this study. In order to expedite the computations ZMA and ZJX were run simultaneously. ZID and ZTL were also processed as one batch.

5.2 Model Validation

Table 5.1 Scenarios Used in the Model Study.

ARTCC Center	Concept of Operations		
	Baseline (1996 Traffic)	RVSM (1996 Traffic)	Cruise Climb (1996 Traffic)
ZTL ^a	✓	✓	✓
ZID	✓	✓	✓
ZMA ^b	✓	✓	✓
ZJX	✓	✓	✓

a. 6,000 flights used of 18,000 daily flights

b. 8,000 flights used of 18,000 daily flights

5.2 Model Validation

SAR data derived from SDAT was used to validate AOM and AEM. In this validation study 4320 flights traversing Miami and Jacksonville Enroute Control Centers in August 18, 1997 (between 15:00 and 24:00 Zulu) were used as test case for AEM . A subset of this data is shown in Figure 5.1 illustrating all flights arriving and departing Miami International Airport in the period of analysis. Since the data included real aircraft trajectories flown under ATC intervention it was expected that few (if any) aircraft encounters would occur in the enroute airspace system.

The validation of AOM was relatively simple since this model keeps track of aircraft traversals across sectors. The SAR data contained track point information including sector names thus making possible the comparison of AOM outcomes with the SDAT output. In this study 500 flights were extracted and compared manually with no discrepancies observed between AOM outputs and the SAR data.

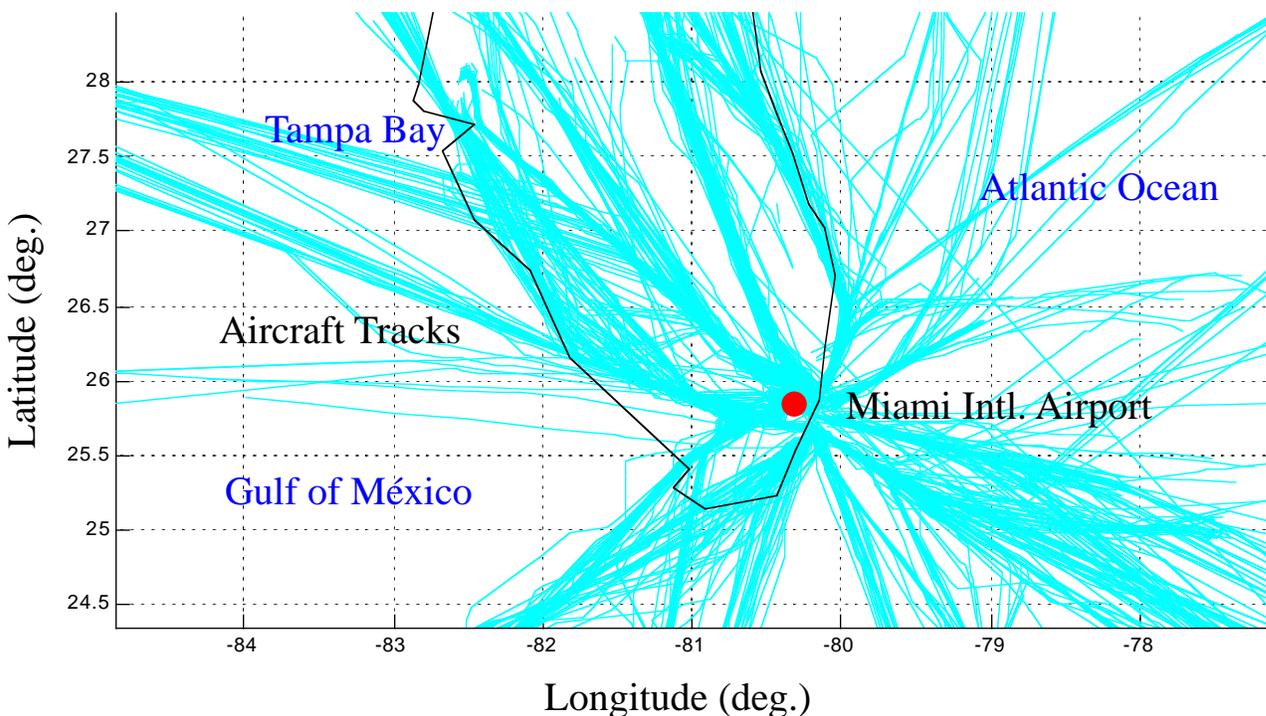


Figure 5.1 Partial ZMA-ZJX Traffic Data used for Model Validation (August 18, 1997).

In the validation of AEM, all 4,320 flights were used to test the number of conflicts in the airspace. In this study the sizes of the *collision envelopes* around the aircraft were chosen as 3D square boxes with dimensions 7nm, 3.5 nm and 500 ft, respectively. The heights of the boxes, or the minimum vertical detection thresholds for conflicts were set at 850 ft below FL180, and 1,700 ft above FL180. These thresholds were selected after multiple runs of AEM revealed that these choice offered a good sensitivity in detecting vertical conflicts given the physical limitations of the data. For example, a careful analysis of the flight track data suggested that aircraft could, in some instances, deviate up to 300 ft from the cruising altitude thus triggering many enroute conflicts if the vertical detection threshold was defined near the minimum vertical separation (i.e., 1,000 or 2,000 ft below and above FL290, respectively). Hence, slightly tighter (850 ft and 1,700 ft, respectively) vertical separation parameters were

5.3 Model Results

used in composing the heights of the box envelope. Results from this analysis are shown in Table 5.2

Table 5.2 Validation Results for ZMA-ZJX ARTCC Traffic (August 18, 1997 Data).

Blind Aircraft Encounter Type	No.of Total Conflicts	No.of Enroute Conflicts
Severity 1	462	6
Severity 2	70	2
Severity 3	2	0
Total	536	8

In the validation study the number of conflicts of severity 3 is zero as expected in the enroute airspace. The number of severity 2 conflict was checked manually and in fact two aircraft came within 5 nm of each other based on the SAR track data. Note that the total number of enroute conflicts is very small as expected. Most of the conflicts occur in vertical transitions as indicated by the difference between columns three and two in Table 5.2. A sampling rate limitation of the data is obvious from these results. Two conflicts of severity 2 seem quite remarkable to occur in a given set of two ARTCC in 8 hours. However, the reader should realize that using 'sparse' SAR track data leaves too many unknowns in the aircraft state variables between two adjacent track points. The linear model described in Chapter 4 coupled with large distances between track points in the enroute airspace are likely to produce 'ghost' conflicts if one considers the natural acceleration and speed noise of aircraft state (i.e., speed, altitude and position) variables. The current state in AEM assumes a linear 3D trajectory between waypoints at constant speed. This will certainly produce unwanted conflicts for some instances since aircraft could experience speed and altitude variations between adjacent track points due to external factors such as wind, barometric corrections, autopilot steady-state errors, etc.

5.3 Model Results

This section presents the outcomes of AOM and AEM under new NAS operational concepts (i.e.,

RVSM and Cruise Climb flight plan conditions). Traffic flow results are first presented to verify whether or not operational changes to flight plans using RVSM and Cruise Climb rules produces significant variations in sector traffic flows. Conflict results are also presented in a subsequent section to assess the number of expected blind conflicts in the airspace.

5.3.1 Traffic Flow Patterns

Traffic flow patterns in an airspace sector are important from a collision risk point of view because they are precursors to the number of blind collisions. Moreover, sector traffic volumes could later be used to model end-game dynamics between conflicting aircraft in a sector volume including ATC and pilot blunders. Until now, nobody has proven a known mathematical relationship exists between the number of aircraft in a given sector and the number of collisions or incidents in that sector. However, several collision risk models developed in the past suggest a quadratic relationship between the number of conflicts and the traffic in a random intersection pattern. Figures 5.2 and 5.3 illustrate two sample traffic flow results obtained with AOM for two sectors in ZID and ZMA, respectively. These figures show the number of aircraft in a given airspace region as a function of time.

Tables B.1 through B.4 in Appendix B show the traffic patterns for all sectors at four centers (ZID, ZTL, ZMA, and ZJX) under three NAS operational concepts (Baseline, RVSM and Cruise Climb). Each table contains a sector designator in the first column (as stated in the ACES database), the current baseline traffic (column 2), RVSM traffic (column 3), Cruise Climb traffic (column 4), and the Percent Traffic Changes (PTC) between baseline and new operational conditions in columns 5 and 6.

Tables B.1 through B.4 in Appendix B suggest that small to medium size changes in sector occupancies are observed with the transition to Free Flight. The changes are highly random since aircraft trajectories differ substantially (both laterally and vertically) when flights are conducted using wind optimized tracks. The values in the tables of Appendix B depict all flights crossing each sector at four enroute control centers from a random database of 6,000-8,000 flights in NAS (depending upon the center analyzed). The sequence of these flights was the same thus protecting the results against bias. Each NAS flight plan scenario provided by the FAA included twenty four hours of projected traffic.

A Wilcoxon signed rank test was performed to validate whether or not the sum of the sector occupancy rank differences is equal to zero (assuming that the distribution of ranks is symmetric about 0). The level of significance used was 0.05. Table 5.3 illustrates the results of this analysis. Note that there are numerous sectors whose traffic patterns are greatly affected by RVSM and cruise climb operations and thus differ from the baseline scenario. Table 5.3 demonstrates that both centers located in the Florida Peninsula show less variations in traffic flows across sectors than those in the mainland portion of the Continental US (CONUS). This result was expected since flight plans in Florida are well organized in a North-South direction whereas RVSM and CC flights across ZID and ZTL show significant more variations than their baseline counterparts. The hypothesis here is that aircraft flight tracks are impacted more in a *central* enroute control center where there is more latitude in optimizing flight plan trajectories laterally. For example, a westbound flight whose original track crossed ZID might take advantage of the prevailing jetstream traveling further south (i.e., not crossing ZID but ZTL).

The results of the analysis suggest that small to medium size changes in sector occupancies should be expected with the transition to *Free Flight*. The changes are highly random since aircraft trajectories differ substantially (both laterally and vertically) when flights are conducted using wind optimized tracks. The sequence of these flights was the same thus protecting the results against bias. Each NAS flight plan scenario provided by the FAA included twenty fours of projected traffic.

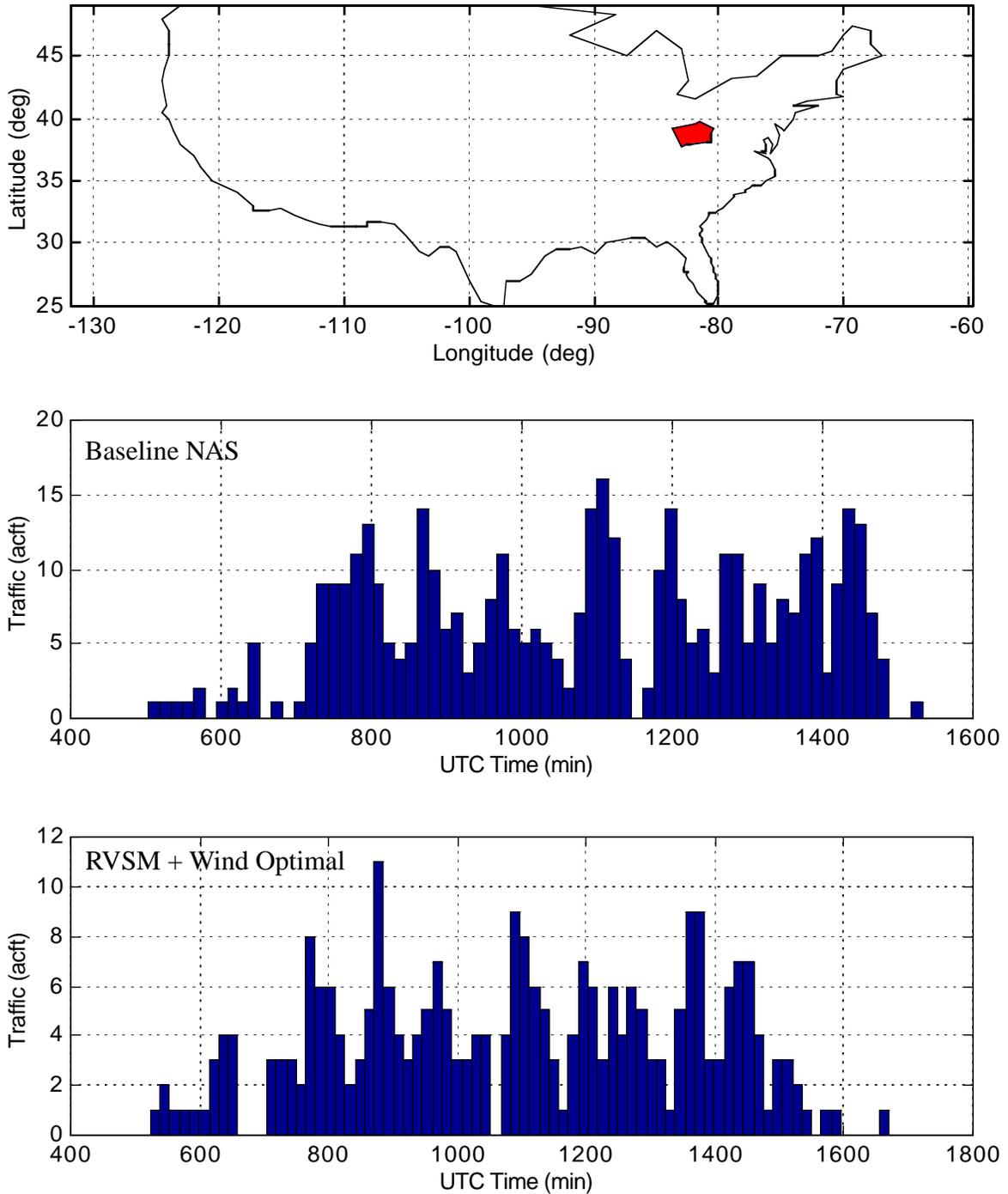


Figure 5.2 AOM Sector Traffic Results (Sector 85 ZID).

5.3 Model Results

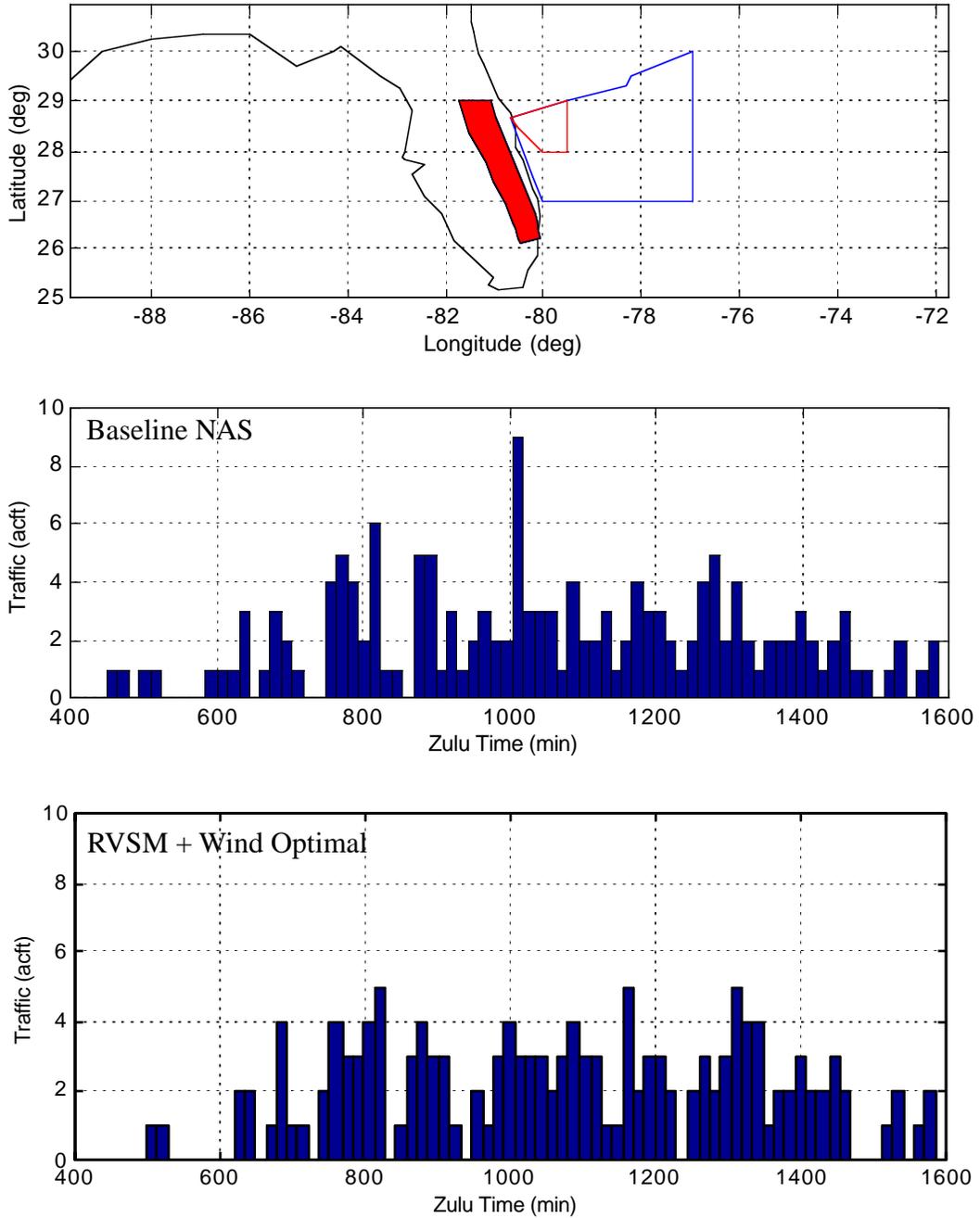


Figure 5.3 AOM Sector Traffic Results (Sector 77 ZMA).

Table 5.3 Statistical Analysis of Sector Traffic Flows (Wilcoxon Rank Sum Test $\alpha=0.05$).

	Number of Sectors in Center	Number of Sectors with Dissimilar Traffic Flows (RVSM / CC)	Average Difference Between Baseline and RVSM / CC Traffic Flows (%)
ZMA	37	8/9	11.2 / 13.8
ZJX	33	1/1	19.8 / 21.5
ZTL	58	5/5	35.2 / 34.9
ZID ^a	32	10/8	102.5 / 86.3

a. Using 6000 baseline flights

5.3.2 Conflict Analysis Results

Three concepts of operations were applied to four centers to assess the number of expected conflicts in the enroute airspace and during vertical transitions. Results of this analysis are shown in Tables 5.4 through 5.9. The analysis was carried out for a pair of centers in order to improve the validity of results over a larger geographical area. These tables show the severity of conflict (column 1), the number of conflicts while the aircraft are in vertical transitions (column 2), and the number of conflicts in the straight and level portion of the enroute flight (called enroute conflicts for simplicity). Vertical transition conflicts are defined as those where at least one of the aircraft is executing a vertical change at the time of the conflict. A Wilcoxon rank sum test was applied to the data to verify whether or not the mean conflicts under baseline conditions and new Free Flight operational concepts would be the same. Table 5.10 shows the summary of the nonparametric statistical test results. The number of conflicts per center was estimated at 15 minute intervals to be consistent with previous analyses. In general, the spatial and time variations of conflicts under current NAS operations and those expected under RVSM and Cruise Climb are significant as shown in Table 5.10. Only in two instances of twelve comparisons (RVSM vs CC at ZID and ZTL) the number of conflicts observed is judged to be the same. Another generalization

obtained after close examination of Tables 5.4 through 5.9 is that the number of blind conflicts under future NAS operational concepts is reduced dramatically (up to 52% in some cases) at the same demand levels.

Some would argue that due to an apparent reduction in the number of conflicts under RVSM and CC conditions (see Tables 5.4 through 5.9) ATC controllers would experience less workload and thus the system might be judged to be safer than under baseline conditions. This notion needs to be further investigated given that workload is not a simple linear function of the number of flights in a sector, and it certainly depends upon other complexities such as sector geometry, flight path geometry, human reliability, situational awareness, automation tools, etc., to name a few. The case in point to be made here is that from the individual collision risk assessment viewpoint, a blunder mode in RVSM or CC might be more likely to cause a midair collision than under baseline conditions due to reduced margins of system failure and recovery of the human controller. Judging the probability of such a failure taking place in a more automated environment is a challenging issue that warrants further investigation.

Table 5.4 ZMA and ZJX ARTCC Conflict Statistics (Baseline).

Conflict Type	Vertical Transition Conflicts	Enroute Conflicts
Severity 1	127	28
Severity 2	91	19
Severity 3	13	9
Total	231	56

5.3 Model Results

Table 5.5 ZMA and ZJX ARTCC Conflict Statistics (RVSM).

Conflict Type	Vertical Transition Conflicts	Enroute Conflicts
Severity 1	104	8
Severity 2	66	6
Severity 3	4	2
Total	174	16

Table 5.6 ZMA and ZJX ARTCC Conflict Statistics (Cruise Climb).

Conflict Type	Vertical Transition Conflicts	Enroute Conflicts
Severity 1	127	28
Severity 2	91	19
Severity 3	13	9
Total	231	56

5.3 Model Results

Table 5.7 ZTL and ZID ARTCC Conflict Statistics (Baseline).

Conflict Type	Vertical Transition Conflicts	Enroute Conflicts
Severity1	110	15
Severity2	64	13
Severity3	10	4
Total	184	32

Table 5.8 ZTL and ZID ARTCC Conflict Statistics (RVSM).

Conflict Type	Vertical Transition Conflicts	Enroute Conflicts
Severity 1	127	28
Severity 2	91	19
Severity 3	13	9
Total	231	56

5.3 Model Results

Table 5.9 ZTL and ZID ARTCC Conflict Statistics (Cruise Climb).

Conflict Type	Vertical Transition Conflicts	Enroute Conflicts
Severity 1	104	8
Severity 2	66	6
Severity 3	4	2
Total	174	16

5.3 Model Results

Table 5.10 Statistical Analysis of 15-Minute ARTCC Center Conflicts.

ARTCC Center	Scenario	P Values ($\alpha = 0.05$)
ZID/ZTL	Baseline vs. CC (enroute)	0.441
ZID/ZTL	Baseline vs. CC (transition)	0.021
ZID/ZTL	Baseline vs. RVSM (enroute)	0.016
ZID/ZTL	Baseline vs. RVSM (transition)	0.007
ZID/ZTL	RVSM vs. CC (enroute)	0.060
ZID/ZTL	RVSM vs. CC(transition)	0.562
ZMA/ZJX	Baseline vs. CC (enroute)	0.002
ZMA/ZJX	Baseline vs. CC (transition)	0.374
ZMA/ZJX	Baseline vs. RVSM (enroute)	0.000
ZMA/ZJX	Baseline vs. RVSM (transition)	0.382
ZMA/ZJX	RVSM vs. CC (enroute)	0.056
ZMA/ZJX	RVSM vs. CC(transition)	0.954

Other important conflict statistics are shown in Tables 5.11 through 5.15. Table 5.11 depicts the average relative heading of each blind conflict and its standard deviation (in parenthesis) for three NAS operational scenarios and two grouped ARTCC centers. These results are consistent with the AEM analysis shown in Table 5.11. It is interesting to observe that both baseline and RVSM scenarios have similar relative conflict geometries whereas the Cruise Climb scenario shows substantial differences across all four ARTCC analyzed. Table 5.11 includes all conflict instances and thus enroute and vertical transition conflicts are included. Figure 5.4 shows graphically relative heading results for the baseline and

CC conditions. Note the larger relative heading angles for the Cruise Climb conditions. The results shown in Figure 5.4 support the notion that further investigation of Free Flight controller strategies are needed to demonstrate advanced ATM capabilities while facing complex spatial and time distribution of the conflicts.

The results of Table 5.12 show the conflict time statistics for vertical transition conflicts. The data suggest some uniformity among RVSM and CC scenarios (shorter conflicts) and longer conflict times for the baseline conditions. This result is expected because the baseline scenario uses a 2,000 ft vertical separation criteria above FL 290 thus making conflicts last longer. Figure 5.5 shows graphically the vertical transition conflict times for the baseline and cruise climb scenario conditions. It is evident from this figure that there exist a large number of blind conflicts in the baseline scenario.

Table 5.13 shows the statistics for enroute conflict times. Enroute conflicts are coplanar conflicts where loss of separation would occur at the same flight level without intervention. In this table the RVSM scenario produced consistently higher conflict times across ZMA and ZJX. The cruise climb scenario, on the other hand, produced the lowest conflict times across all centers. Figure 5.6 illustrates graphically a histogram of enroute conflict times observed in ZID and ZTL for baseline and cruise climb conditions. The figure shows consistently lower conflict times for the cruise climb scenario.

Table 5.14 shows results for the predicted Closest Point of Approach (CPA) of aircraft conflicts where at least one aircraft performs a vertical transition maneuver. The means of CPA are expected to be slightly higher under Free Flight conditions due to the better distribution of flights in three dimensional airspace. Figure 5.7 illustrates one example of the CPA distribution for all vertical transition conflicts in ZID and ZTL. This figure shows moderate differences in the probability distribution of CPA for current, RVSM, and cruise-climb conditions.

Table 5.15 summarizes the results for the predicted Closest Point of Approach (CPA) for enroute conflicts. In this figure is evident that the RVSM mode of operation provides the best distribution of flights possible with the highest CPA values. Figure 5.8 presents a sample enroute CPA histogram comparing baseline with Cruise Climb conditions.

Finally, Figure 5.9 presents the expected variations of the R statistic presented in Chapter 4 for all ver-

5.3 Model Results

tical transition conflicts in ZID and ZTL. The R statistic represents an weighted average of the number of conflicts, the average durations of these conflicts and the percent rate of convergence of conflicts. Quantitatively, the R statistic values obtained for Free Flight scenarios are higher than under current NAS operational conditions.

The results presented in Figures 5.4 through 5.9 represent a small sample of the outcomes of AOM and AEM. Each scenario run generated a vast amount of conflict information that is not possible to reproduce here. This information could help understand the behavior of projected flight plans across NAS and their influence on collision risk precursors.

Table 5.11 Relative Heading Statistics for Various NAS Operational Scenarios.

ARTCC	Baseline Mean (standard dev.) (deg)	RVSM Mean (standard dev.) (deg)	Cruise Climb Mean (standard dev.) (deg)
ZMA/ZJX	36.48 (64.35)	37.91 (65.38)	45.49 (68.87)
ZID/ZTL	36.22 (59.00)	37.57 (53.81)	51.54 (63.88)

Table 5.12 Vertical Transition Conflict Time Statistics for Various NAS Operational Scenarios.

ARTCC	Baseline Mean (standard dev.) (min)	RVSM Mean (standard dev.) (min)	Cruise Climb Mean (standard dev.) (min)
ZMA/ZJX	4.56 (11.52)	2.85 (9.91)	2.86 (9.89)
ZID/ZTL	3.04 (2.40)	2.40 (5.47)	2.27 (5.18)

5.3 Model Results

Table 5.13 Enroute Conflict Time Statistics for Various NAS Operational Scenarios.

ARTCC	Baseline Mean (standard dev.) (min)	RVSM Mean (standard dev.) (min)	Cruise Climb Mean (standard dev.) (min.)
ZMA/ZJX	5.37 (9.04)	9.18 (11.84)	5.15 (9.42)
ZID/ZTL	6.31 (10.86)	6.21 (10.94)	4.48 (10.30)

Table 5.14 Closest Point of Approach Statistics for Various NAS Operational Scenarios (Vertical Transition Conflicts)

ARTCC	Baseline Mean (standard dev.) (nm)	RVSM Mean (standard dev.) (nm)	Cruise Climb Mean (standard dev.) (nm.)
ZMA/ZJX	3.26 (2.61)	3.37 (2.47)	3.74 (2.49)
ZID/ZTL	3.49 (2.63)	3.85 (2.66)	3.76 (2.75)

Table 5.15 Closest Point of Approach Statistics for Various NAS Operational Scenarios (Enroute Conflicts)

ARTCC	Baseline Mean (standard dev.) (nm)	RVSM Mean (standard dev.) (nm)	Cruise Climb Mean (standard dev.) (nm.)
ZMA/ZJX	3.54 (2.84)	4.68 (3.32)	3.62 (3.02)
ZID/ZTL	3.97 (2.88)	5.09 (2.57)	4.17 (2.30)

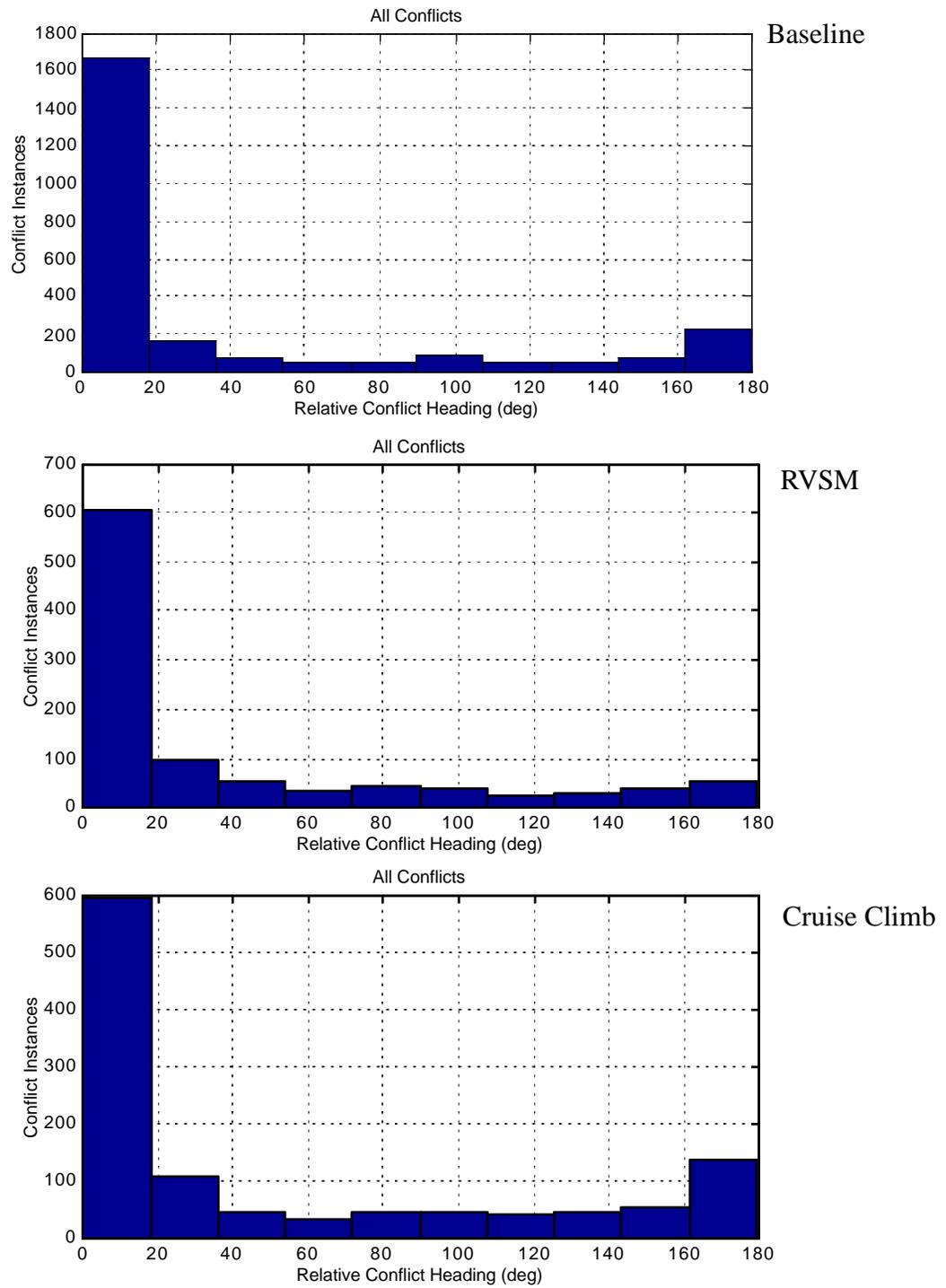


Figure 5.4 Relative Heading Angle Conflict Instances (ZID and ZTL).

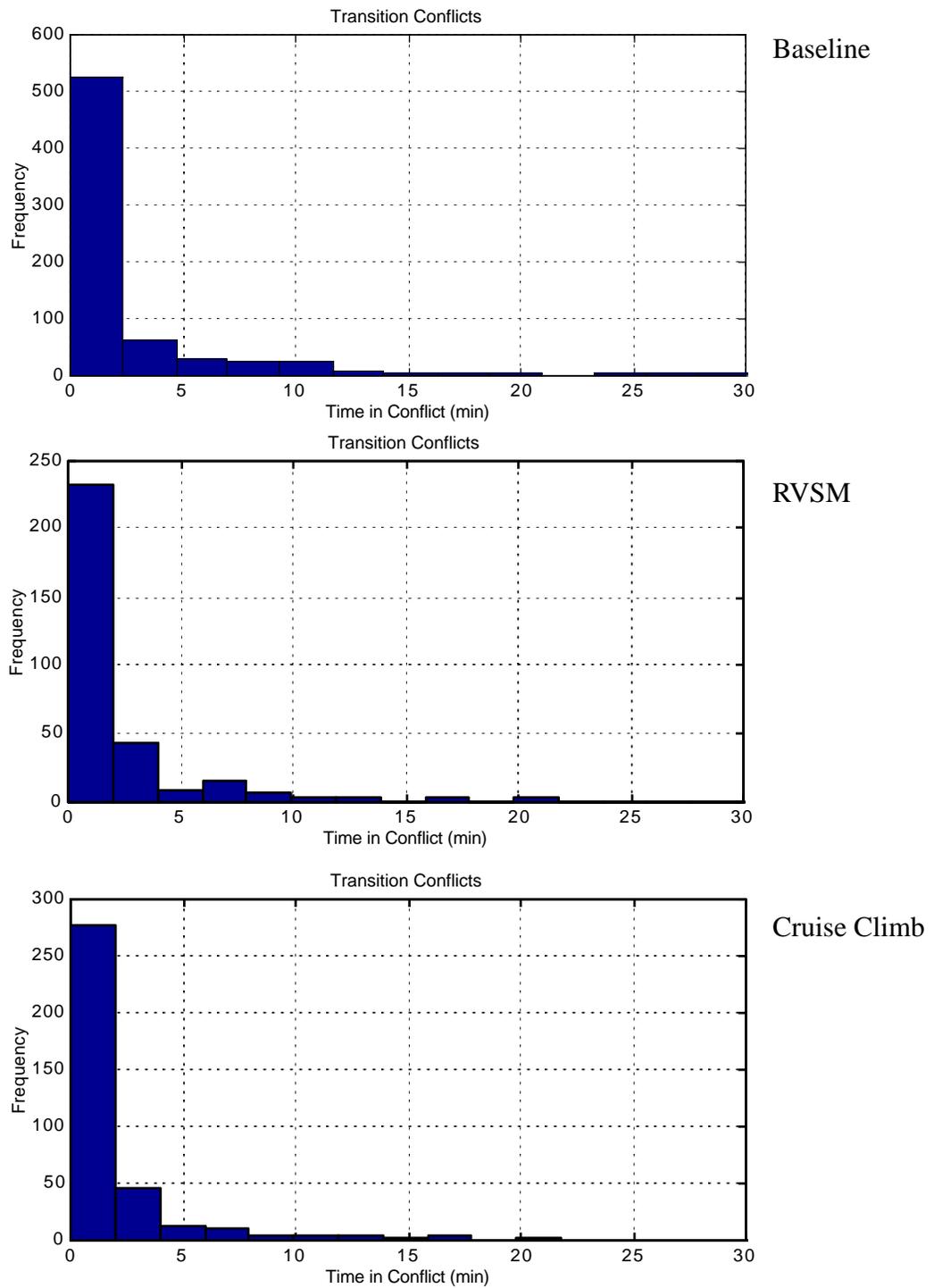


Figure 5.5 Vertical Transition Conflict Times (ZID and ZTL).

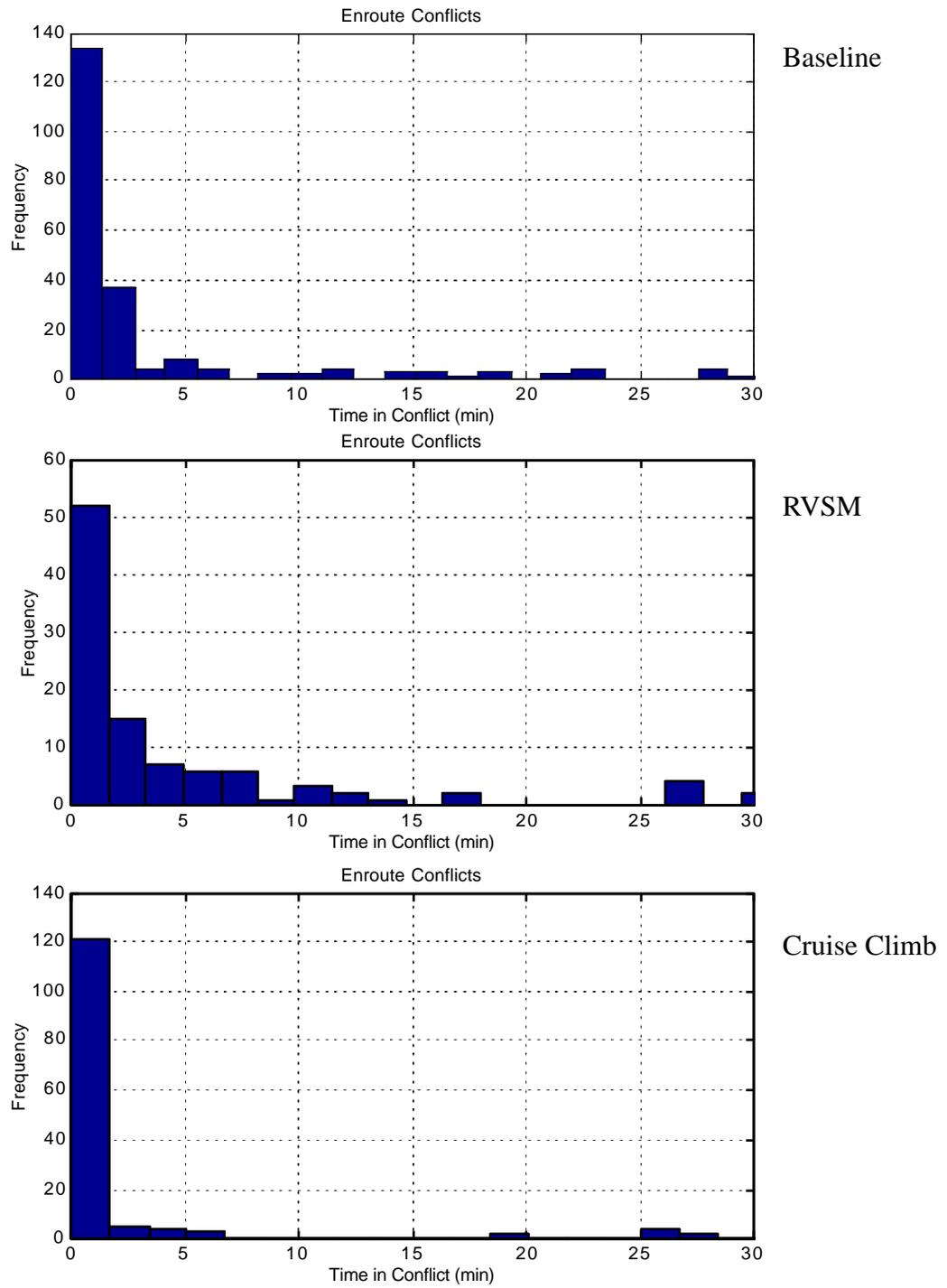


Figure 5.6 Enroute Conflict Times (ZID and ZTL).

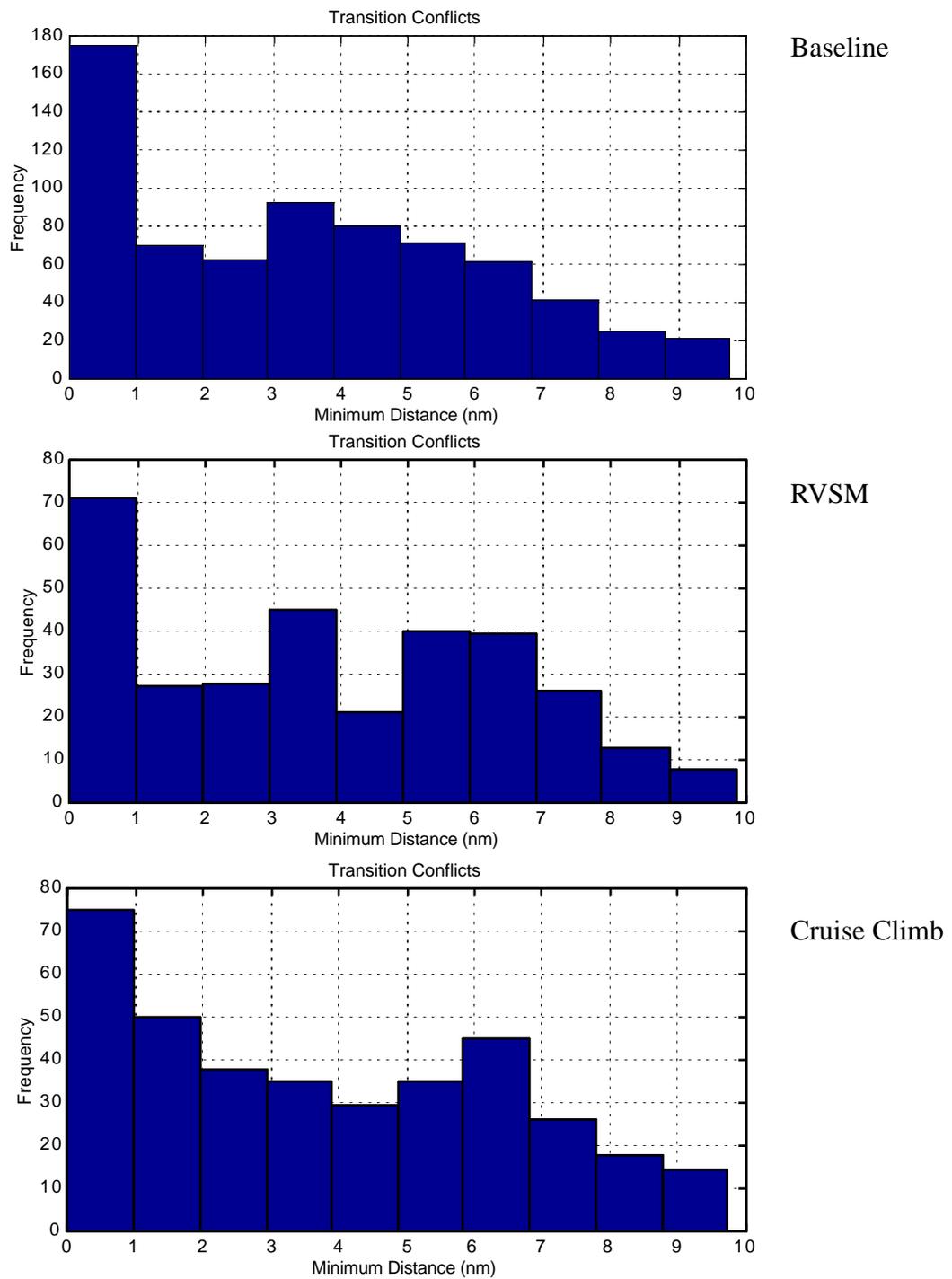


Figure 5.7 Minimum Distance Distribution for Vertical Transition Conflicts (ZID and ZTL).

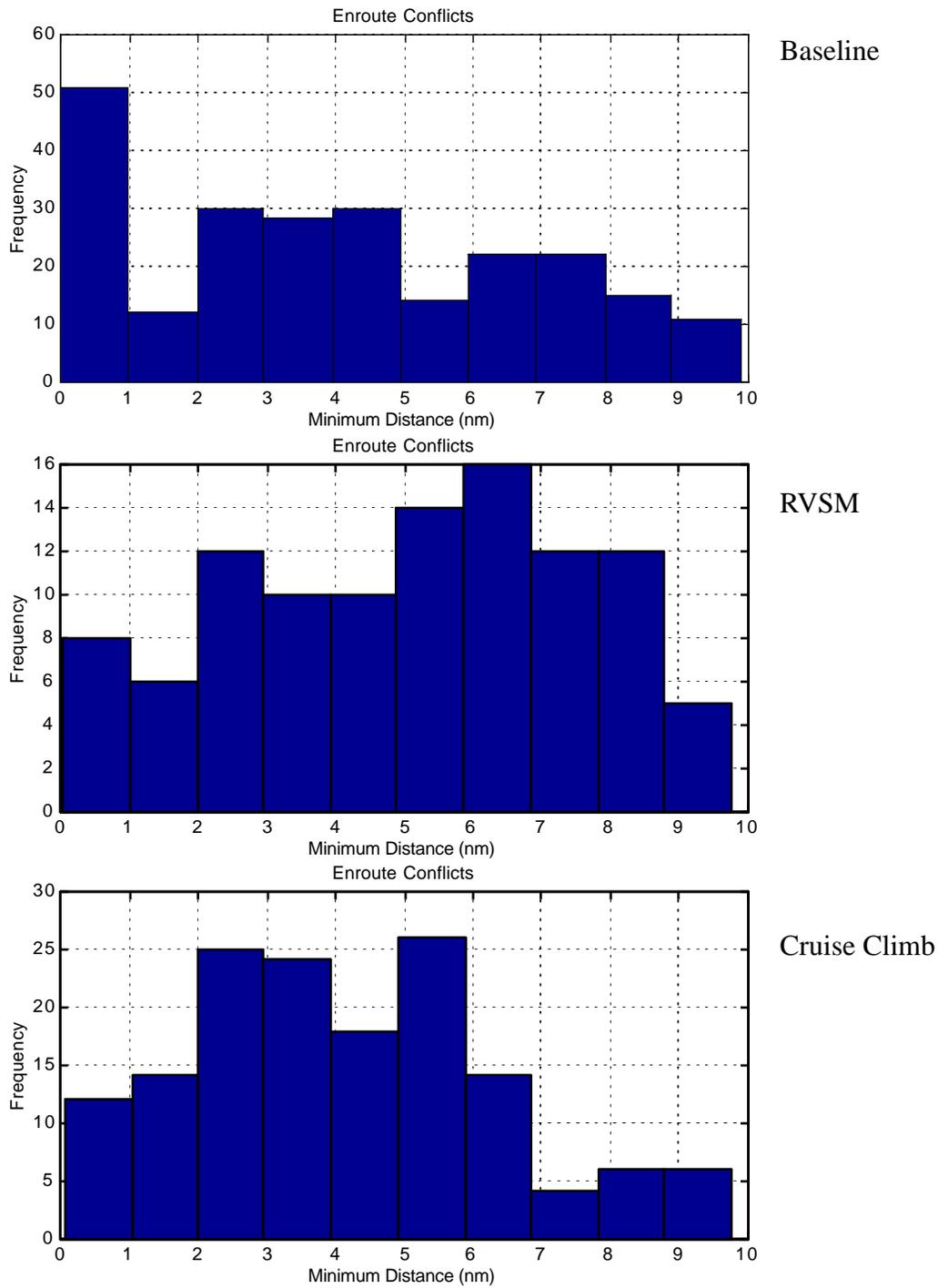


Figure 5.8 Minimum Distance Distribution for Enroute Conflicts (ZID and ZTL).

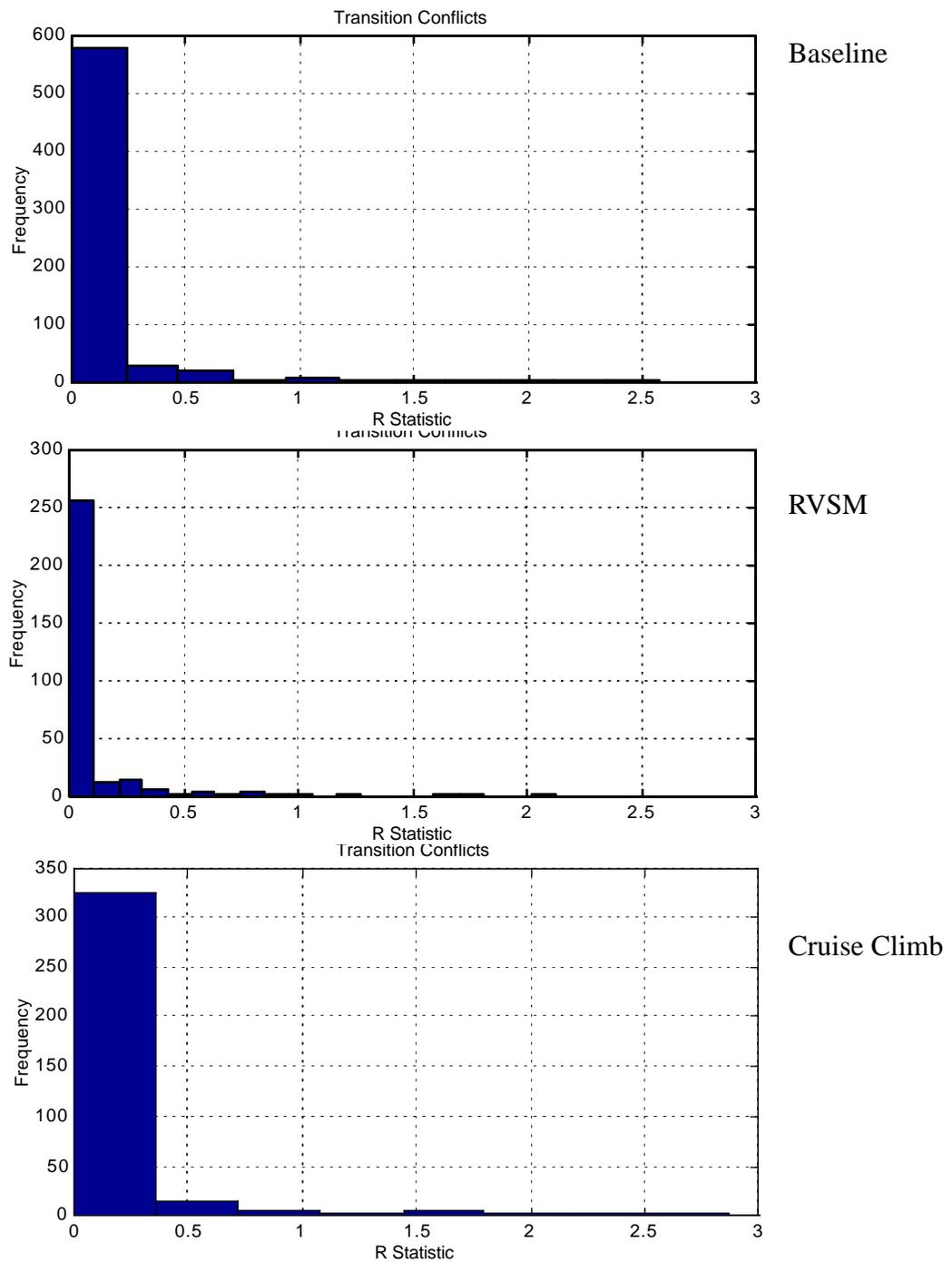


Figure 5.9 Proposed R Statistic for Vertical Transition Conflicts (ZTL and ZID).

This report presents a first-order analysis of blind conflicts expected to affect the NAS system in the near future under two Free Flight operational concepts: RVSM and Cruise Climb. The study focused on the development and use of two computer models (AOM and AEM) to respectively predict traffic flows across well defined volumes of airspace, and the number of potential blind conflicts if all flight plans are executed without controller or pilot intervention. The models developed have been coded in Matlab, a general engineering language, facilitating their execution on any computer platform (PCs, PowerPC Macs, and UNIX workstations) without modifications.

While this study provides a first-order approximation of the level of conflict exposure in a particular center or sector it does not provide a measure of collision risk in the true sense. Further investigation of the end-game ATC controller and pilot dynamics (including aircraft navigational accuracy) is needed to truly quantify collision risk.

Some insightful computational test are conducted to understand traffic pattern variations and blind conflicts in four enroute control centers in CONUS. The time and spatial characteristics of these conflicts were studied using the tools developed to provide a view into the type of conflict encounters expected in future NAS operations. The hope is that these tools would be further refined to assess collision risk incorporating human and vehicle reliability models.

Several conclusions can be derived from this case study:

- 1) There would be likely moderate to substantial variations in traffic flow patterns across various ARTCC sectors in NAS. The introduction of flexible flight planning rules expected in Free Flight would affect differently various ARTCC centers according to their geographical location. In this study ZMA and ZJX centers had less variation in 15-minute traffic flows than those observed across ZID and ZTL.
- 2) The number of potential conflicts in the enroute airspace system would decrease with the introduction of Free Flight operations if reduced vertical separation criteria is allowed. It is not possible to quantify the risk associated with reduced separation blind conflicts using the models developed. However, further investigation is needed since ATC controllers and pilots operating under RVSM rules might have less time to react to blunders under these circumstances (assuming current levels of automation).
- 3) The number of blind conflicts expected under Cruise Climb and RVSM modes (as defined in this report in Chapter 5) are of the same order of magnitude. It is not clear how ATC controllers would react to potential conflicts between two or more aircraft operating in a cruise climb and what would be their influence on collision risk. Further investigation is necessary.
- 4) In general, there are substantial to moderate differences in the time and space distribution of blind conflicts under RVSM and Cruise Climb scenarios. The effect of these distributions in ATC controller monitoring workload and eventual reliability to intervene under blunder conditions should be further investigated.
- 5) In general, vertical transition conflict times under RVSM and Cruise Climb scenarios are expected to be shorter in duration due to the smaller vertical separation criteria. Enroute conflict times (i.e., coplanar conflicts) varied significantly. Under some circumstances, enroute conflict times increased for at least one of the Free Flight scenarios investigated.
- 6) The distribution of relative headings of conflicts varied in the transition to some Free Flight scenarios (i.e., cruise climb). This parameter could have important implications on how controllers perceive conflicts and eventually, on the intervention modes used to separate traffic. Further investigation of this

important parameter is also needed.

6.1 Recommendations

Based on the results obtained in this study the research team would like to make the following recommendations:

- 1) Future NAS scenarios should be modeled using projected traffic growth instead of the 1996 traffic levels used in this study. This extension would require relatively small resources because NAS scenarios for the years 2005 and 2015 have been developed recently by the FAA/CSSI.
- 2) Detailed investigation of ATC controller, pilot and aircraft dynamics should be the next step to quantify collision risk under ATC controller and pilot intervention. A few ideas on how to accomplish this are explained in Section 6.2 of this chapter.
- 3) There is a critical need to collect information on ATC controller responses and separation heuristics under new NAS operational concepts. This is viewed as an important step to quantify the reliability of human controllers in an intervention model. Perhaps carefully planned human-in-the-loop studies should be undertaken using the most likely scenarios found in this study.
- 4) Pilot and airline operational practices should also be investigated if collision risk is to be modeled using an enhanced AEM model.
- 5) A complete study of NAS behavior should be undertaken in order to understand geographical and procedural differences across various ARTCC. In our study only four of twenty centers were analyzed. The models developed can process all NAS data at the expense of longer computational times.
- 6) AEM could be further enhanced with various detection envelopes to assess differences in conflict rates under various vertical, in-trail and lateral separation criteria.
- 7) Explore the possible integration of existing analytic collision risk models with AEM. A first attempt to quantify collision risk could use the Analytic Blunder Model (ABM).

6.2 Possible Model Extensions

AOM and AEM provide a valid start-up framework to model end-game controller and pilot interactions in a complex ATC system. These interactions are necessary to assess causal relationships that lead to loss of separation and ultimately to mid-air collisions. A possible approach to quantify risk assessment is then to enhance the deterministic nature of the AEM model to introduce end-game dynamics using causal theory, dynamic fault tree analysis or other suitable techniques followed by mathematical models of the complex human-in-the-loop interactions preceding a conflict.

The steps to be used in this approach could be as follows:

- Development of an integrated framework to predict aircraft collision risk in enroute airspace
- Assessment of ATC control strategies to resolve conflicts in Free Flight
- Assessment of enroute airspace conflict dynamics using failure mode models
- Mapping of tasks (2) and (3) into a suitable mathematical model (fast time model)
- Integration of task (4) into existing the blind conflict assessment model or other suitable fast time simulation models (i.e., RAMS etc.)

These items could be studied using an enhanced version of the AEM model to predict aircraft conflicts in the enroute airspace system. Extensions to the terminal airspace could then follow. It is important to realize that most of the analysis proposed here would require air traffic control simulators and new ATC controller performance data to quantify controller lags, blunder modes and controller heuristics employed in monitoring and separating traffic in an advanced ATM system. Developing human controller, pilot and aircraft navigational reliability functions is a large task that perhaps can be executed in low fidelity air traffic control simulators, and existing computer modeling tools such as MIDAS. These tools would generate databases of relevant ATC/pilot response times and eventually express these into suitable mathematical functions (i.e., table functions, neural/petri nets, etc.) that could be incorporated into an enhanced AEM model.

Bibliography

- 1) Brooker, P., Aircraft Separation Assurance: Systems Design, *Journal of Air Navigation*, Volume 36, pp.28-35, 1982.
- 2) Brooker, P., Aircraft Collision Risk in the North Atlantic Region, *Journal of Operational Research Society*, Volume 35, No. 8, pp.695-703, 1984.
- 3) Corker, K. Pisanich, G., and M. Bunzo "A Cognitive System model for Human/Automation Dynamics in Airspace Management". Proceedings of the First European/U.S. Symposium on Air Traffic Management, Saclay, France, June 16-19, 1997.
- 4) CSSI Corporation, Find Crossings – User's Manual, Washington, DC, March 1996.
- 5) CSSI Corporation, Verbal Correspondance about OPGEN, Washington, DC, October 1997.
- 6) CSSI Corporation, NARIM Scenario Assumptions Document, Washington, DC, July 1998.
- 7) Cox, M.E., ten Have, J.M. and D.A. Forrester, European Studies to Investigate the Feasibility of Using 1000 ft Vertical Separation Minima Above FL 290. Part I, *Journal of Air Navigation*, Volume 44, pp.171-183, 1987.

- 8) Federal Aviation Administration, 1997 Aviation System Capacity Plan, Office of System Capacity and Requirements, Washington, DC, September 1997.
- 9) Federal Aviation Administration/Eurocontrol, A Concept Paper for Separation Safety Modeling, Washington DC, May 1998.
- 10) Ford, R.L., On the Use of Height Rules in Off-Route Airspace, *Journal of Air Navigation*, Volume 36, pp.269-287, 1982.
- 11) Ford, R.L., The Conflict Resolution Process for TCAS II and Some Simulation Results, *Journal of Air Navigation*, Volume 40, pp.283-303, 1986.
- 12) Goodwin, J.L and R.L. Ford, Random Air Traffic Generation, *Journal of Air Navigation*, Volume 38, pp.219-233, 1984.
- 13) Harrison, D. and G. Moek, European Studies to Investigate the Feasibility of Using 1,000 ft Vertical Separation Minima Above FL 290. Part II: Precision Radar Data Analysis and Collision Risk Assessment, *Journal of Air Navigation*, Volume 45, pp.91-106, 1988.
- 14) Hilburn, B.G., Baker, M.W.P., Pakela, W.D, and R. Parasuramam. The Effect of Free Flight on Air Traffic Control Mental Workload, Monitoring and System Performance, Proceeding of the 10th International CEAS Conference on Free Flight, Amsterdam, 1997.
- 15) Hsu, D.A, A Theoretical Framework for Analysis of Lateral Position errors in VOR jet Route Systems, *Journal of Air Navigation*, Volume 36, pp.263-268, 1982.
- 16) Hsu, D.A, The Evaluation of Aircraft Collision Probabilities at Intersecting Air Routes, *Journal of Air Navigation*, Volume 24, pp.78-102, 1980.
- 17) Laudeman, I. V., and E.A. Palmer, "Quantitative Measurement of Observed Workload in the Analysis of Aircrew Performance", *International Journal of Aviation Psychology*, Volume 5, pp.187-197, 1995.
- 18) Nagaoka, S., Effects of Measurement Errors in Estimating the Probability of Vertical Overlap, *Journal of Air Navigation*, Volume 38, pp.235-243, 1984.

- 19) National Research Council, *Flight to the Future, Human Factors in Air Traffic Control*, National Academy Press, 1997.
- 20) Machol, R.E., Thirty Years of Modeling Midair Collisions, *Interfaces*, Volume 25, September-October, pp. 151-172, 1995.
- 21) Mathworks Inc., Matlab 5.2 Users Manual, Natick, MA, 1997.
- 22) Office of Technology Assessment, Safe Skies for Tomorrow: Aviation safety in a Competitive Environment. OTA-SET-381. Washington, DC: U.S. Government Printing Office, July 1988.
- 23) Polhemus, N.W. and D. Livingston, , Characterizing Cross-Track Error Distributions for Continental Jet Routes, *Journal of Air Navigation*, Volume 34, pp.134-141, 1981.
- 24) Ratcliffe, S. and R.L. Ford, Conflicts Between Random Flights in a Given Area, *Journal of Air Navigation*, Volume 35, pp.47-71, 1982.
- 25) Reich, P. G., "Analysis of Long-Range Air Traffic Systems: Separation Standards I," *The Journal of the Institute of Navigation*, Volume 19 88-98, 1966.
- 26) Smith, K., Scallen, S. F., Knecht, W., Hancock, P., "An Index of Dynamic Density", *The Journal of the Human Factors Society*, Volume 20, pp.69-78, 1998.
- 27) Suchkov, A., Embt, D., and W. Colligan, "Dynamic Density Study", CSSI Corporation, December 1997.
- 28) Ten Have, J.M., Harrison, D. and M.E. Cox, European Studies to Investigate the Feasibility of Using 1,000 ft Vertical Separation Minima Above FL 290. Part III: Further Results and Overall Conclusions, *Journal of Air Navigation*, Volume 46, pp.245-261, 1988.
- 29) Wyndemere, Inc., "An Evaluation of Air Traffic Control Complexity", Final Report for Contract Number NAS2-14284, October, 1996.

Model Data Structures and M-File Definitions

The data structures used in the program to determine the occupancies of the sectors are described in this section.

A.1 AOM and AEM Data Structures

Sector Module Information Structure (S)

S stores the information about each sector module. The records of S are explained below.

1. ver

Stores the vertices that define the floor and ceiling of the sector module. These vertices will be arranged in the order as they appear in the input file. The sector information extracted from Sector Design and Analysis (SDAT) contains vertices arranged in a clockwise order.

2. lat

Contains the latitude of the vertices in the order as they appear in S.ver.

3. long

Contains the longitude of the vertices in the order as they appear in S.ver. If a sector module has

n vertices, both S.lat and S.long will be rows of size (n+1) where the last value corresponds to the first value. This way of storing the coordinates is useful while plotting the sector using the Matlab built-in function plot. This arrangement is also helpful while using the Matlab function inpolygon to check if a point lies within a polygon.

4. name

This is a three letter designation of the sector to which this sector module belongs.

5. sectfpa

This is a four digit string where the first two correspond to the sector label and the last two correspond to the FPA label.

6. arts

If the sector module is a part of a terminal approach sector, this field will have a value that corresponds to the type of ARTS equipment available in the sector module.

7. approach

If the sector module is a part of a terminal approach sector, this field will have a value that corresponds to the approach control pertaining to this sector.

Node Information Structure (Node)

Node is a data structure storing the information about the vertices that define the sector modules.

Node has two records as explained below.

1. Name

This is a two dimensional character array storing the name of all the vertices.

2. N

This is a two dimensional array storing the latitudes and longitudes of all the vertices.

Height Information about the Sector Modules (h)

h stores the floor and ceiling altitudes of the sector modules. This is a two dimensional array where each row corresponds to the floor and ceiling altitude in hundreds of feet. $h(i, 1)$ will be the floor altitude of the i th sector module and $h(i, 2)$ will be the ceiling altitude of the i th sector module.

Structure with Mathematical Representation of Sector Modules (Se).

This data structure stores the mathematical representation of the sector modules. The records under Se are shown below.

1. line

This field defines the equation of each of the vertical faces. $Se(i).line(1,j)$ gives information about the j th face of the i th sector module. $Se.line$ has four records under it.

num

This is the number associated with the vertical face.

alpha

This is the inward gradient of the vertical face. Determination of the inward gradient is explained in Section 4.3.1.

c

This corresponds to the normal distance from the origin to the face in the direction of the inward gradient. Hence c will be negative if the origin (intersection of equator with Greenwich meridian) lies in the half-space toward the direction of the inward gradient and positive otherwise. $\alpha \sum(x, y) = c$ will hold true for any point (x,y) lying on the face.

flag

This is a digit that has a value 0 if the face is already numbered and a value 1 if the face is not numbered.

2. node

The field node is an array structure having two fields nodenum and type , corresponding respectively to the node number and the vertex type for each of the vertices of a sector module. Determination of the vertex type is explained in Section 4.3.2. The record Se(k).node(m) will have information regarding the mth vertex of the kth sector.

3. hmin

This corresponds to the floor altitude in hundreds of feet of the sector module.

4. hmax

This corresponds to the ceiling altitude in hundreds of feet of the sector module.

5. hminnum

This is a label corresponding to the floor altitude level.

6. hmaxnum

This is a label corresponding to the ceiling altitude level.

Structure with Adjacency Information of Sector Modules with Respect to Faces (Adjsec)

Adjsec is a data structure storing information about sectors that are adjacent with respect to a face. This has two records as described below.

1. pos

This is an array that contains all the sector modules that lie on that side of the face which does not contain the origin. It has sub-fields under it, namely, sect and loc, corresponding respectively to the sector module number and the location of the vertical face in the sector module.

2. neg

This is an array that contains all the sector modules that lie on that side of the face which contains the origin. Like pos, neg has two fields sect and loc.

Classifying the sector modules into those lying towards the origin and those lying away from the origin is helpful to identify the extreme faces of the defined airspace. A vertical face having sector modules lying toward only one side will be an extreme face.

Structure with Adjacency Information of Sector Modules with Respect to Nodes (Adjsecnode)

Adjsecnode is the data structure that stores adjacency information with respect to the nodes. It has two records as explained below.

1. sect

This is a row array of all the sector modules containing the vertex.

2. loc

This is a row array of the location of the vertex in the sector module corresponding to the record sect.

Adjsecnode(i).loc(1,j) gives the location of the ith vertex on the sector Adjsecnode(i).sect(1,j).

Sector Module Adjacency Information (Adj)

Adj is a data structure storing the information regarding all the sector modules that are adjacent to a sector module in question. This has a record sect which is a row array storing the sector module numbers.

Flight Plan Structure (Fp)

Fp is the data structure which stores the information about the Flight Plans. Fp has the following fields.

1. fname

Name designating the flight plan.

2. model

Designator indicating the type of aircraft model.

3. origin

Three letter designator of the origin airport.

4. dest

Three letter designator of the destination airport.

5. n

Number of way-points comprising the flight trajectory.

6. wp

Array storing the latitude, longitude and altitude of each of the n way points

7. twp

Array of size n by 1 storing the time corresponding to each way-point.

8. omodule

The sector module that is first encountered by the flight.

9. start_point

The point where the flight first encounters a defined sector module.

10. start_time

The time when the flight first encounters a defined sector module.

11. start_seg

The flight segment which enters the defined airspace.

12. start_lam

The location of start_point on the flight segment start_seg.

13. path

Structure storing all the sector modules encountered by the flight and the time of crossing. Path has the following three records under it.

- a) sectSector module encountered. This will be the number used by the program while storing the sector module information.
- b) entertTime of entry into the sector module.
- c) exittTime of exit from the sector module.

14. main_path

Structure storing all the sectors encountered by the flight and the time of crossing. Like path, main_path has three records storing the sector number, entry time, and the exit time corresponding to the crossing.

Sector Information (main_S)

This is a data structure corresponding to a sector having the following records.

1. name

Array of characters denoting the name of the sector. This is unique for a sector.

2. label

Row array of size 1 by 2 denoting the sector label.

3. subs

Row array which stores the sector module numbers that comprise the sector. The size of this array depends on the number of sector modules that make up the sector.

4. occup

Structure that stores the information about the flights that are crossed and the time of crossing. Occup has the following three records under it.

- a. fnum:Flight number that is crossed.
- b. entert:Time during which the fnum enters the sector.

c. `exitt`: Time during which the `fnum` exits the sector.

Sector Adjacency Information (`main_Adj`)

`main_Adj` is a data structure storing the information regarding all the sectors that adjacent to each sector. This has a record `sect` which is a row array storing the sector numbers.

Adjacency Information of Sectors with Respect to Nodes (`main_Adjsecnode`)

`main_Adjsecnode` is the data structure that stores adjacency information with respect to the nodes. It has one record as explained below.

1. `sect`

This is a row array of all the sectors containing the vertex.

A.2 M-Files

The m-files developed for the Airspace Sector Occupancy Model are described briefly in this section. The arrangement of these m-files and their hierarchy is depicted in Figures 25-28. Important m-files are discussed in Chapters 3 and 4. Important m-files used in the Airspace Sector Occupancy Model are described briefly below in alphabetical order.

1. `Addvertex`

Purpose

This function determines all the nodes that are present on the faces of sector modules but are not originally defined for it. This m-file also updates all the adjacency information.

Input

This contains information about the sector modules, the nodes, and the adjacency information with respect to nodes and sector modules.

Output

This contains revised information about the sector modules and the adjacency relationships.

2. Checkif_crossed

Purpose

This determines if a flight segment crosses a particular face of a sector module.

Input

This contains information about the flight segment the sector modules.

Output

This contains a binary flag indicating if a crossing has taken place. If yes, the coordinate of the exit point and location of the exit point on the flight segment is determined.

3. Checkif_internal

Purpose

This checks if a point lies on a line connecting two other given points.

Input

This contains the coordinates of the three points.

Output

This contains a binary flag with a value 1 if the given third point lies internally on the line connecting the other two points, and is 0 otherwise.

4. Checkif_same

Purpose

This checks if two points are within an acceptable tolerance to be considered as the same point.

Input

This contains the coordinates of the two points.

Output

This contains a binary flag with a value 1 if the two points are close enough to be considered as the same, and 0 otherwise

5. Exitloc

Purpose

This determines the information about the point where the current sector module is exited by the flight segment in question.

Input

This contains information about the flight segment, information about the current sector module, information regarding the point of entry and the previous sector module number.

Output

This contains information about the point where the current sector module is exited by the flight segment under consideration.

6. Find_ext_sect

Purpose

This identifies the vertical faces that are open to an undefined airspace on one of the sides.

Input

This contains information about the sector modules and the adjacency information of the sector modules with respect to vertical faces.

Output

This contains extreme faces identified based on their locations in the sector modules.

7. Get_mainpath

Purpose

This function identifies the sectors a flight will pass through knowing the sector modules it passes through.

Input

This contains information about the flight trajectory and the sector modules.

Output

This contains updated information about the flight trajectory with the information about the sectors that it passes through.

8. Getnextsect**Purpose**

This identifies the sector module the flight enters after exiting another module.

Input

This contains information about the sectors, the information about the exit pattern from the previous sector module and the adjacency information.

Output

This contains the sector module number that is entered. An indicator 0 is returned if the flight does not enter any of the sector modules.

9. Getnext_afterdummy**Purpose**

This determines the sector module entered by the flight after passing through a vacuum.

Input

This contains information about the flight trajectory under consideration, the extreme faces, and

the information about the sector modules.

Output

This contains the sector module number that is entered by the flight and the information about the point of entry. It includes the coordinates of the entry point and the flight segment number that enters the sector module.

10. Get_dummy

Purpose

This function extends the defined airspace by defining the dummy sectors surrounding the defined airspace.

Input

This contains information about the sector modules, the nodes and the order in which the nodes are used to define a sector module (clockwise or anti-clockwise).

Output

This contains modified information about the sector modules after the inclusion of the dummy sectors.

11. Get_main_Adj

Purpose

This determines the adjacency information of the sectors with respect to each other.

Input

This contains adjacency information about the sector modules with respect to each other and the information about the sectors and sector modules.

Output This contains adjacency information of the sectors with respect to each other.

12. Get_main_Adjsecnode

Purpose

This determines the adjacency information of the sectors with respect to the nodes.

Input

This contains adjacency information about the sector modules with respect to nodes and the information about the sectors and sector modules.

Output

This contains adjacency information of the sector with respect to nodes.

13. Main

Purpose

This is the main function that calls all other functions and determines the occupancy of the sectors.

Input

This contains the input file for the sector geometry and the flight plans.

Output

This contains complete information about the sectors and the flight plans including the occupancy information.

14. Main_occup

Purpose

This function identifies the flights that pass through a sector, knowing the sector modules it encounters.

Input

This contains information about the flight trajectories and the sectors.

Output

This contains updated information about the sectors along with the flights passing through each sector.

15. Next_sect_line**Purpose**

This determines the sector module a flight enters after crossing one sector module across a vertical face.

Input

This contains information about the sector modules, previous exit point and the adjacency information of the sector modules with respect to vertical faces.

Output

This contains the sector module number that the flight enters.

16. Next_sect_node**Purpose**

This determines the sector module a flight enters after crossing one sector module across a vertical edge.

Input

This contains information about the sector modules, the previous exit point, and the adjacency information of the sector modules with respect to nodes.

Output

This contains the sector module number that the flight enters.

17. Next_sect_tb

Purpose

This determines the sector module a flight enters after crossing one sector module across its ceiling or floor.

Input

This contains information about the sector modules, the previous exit point and the adjacency information of the sector modules with respect to floors and ceilings.

Output

This contains the sector module number that the flight enters.

18. Occup

Purpose

This determines the sector modules that a flight passes through.

Input

This contains information about the flight trajectory, sector modules, extreme faces, and the adjacency relationships.

Output

This contains updated information about the flight trajectory, identifying all the sector modules that it passes through.

19. Plot_hist_view

Purpose

This plots the histogram corresponding to the occupancies of the sector and depicts its location on the US map.

Input

This contains information about the occupancy of the sectors, the time interval of the histogram, and the sector number for which the plot is needed.

Output

This contains the histogram plot showing the occupancies of the sector and the plot showing the location of the sector on the US map.

20. Preproadj**Purpose**

This function identifies the sector modules adjacent to other sector modules

Input.

This contains information about the sector modules and the adjacency with respect to nodes.

Output

This contains the information about the adjacency of sector modules with respect to one another.

21. Preproadjsec**Purpose**

This function identifies the sector modules that are adjacent to one another with respect to vertical faces.

Input

This contains information about the sector modules and the adjacency information with respect to nodes.

Output

This contains information about the adjacency of sector modules with respect to vertical faces.

22. Preproadjsectb

Purpose

This function identifies the sector modules that are adjacent to one another with respect to horizontal faces.

Input

This contains information about the sector modules.

Output

This contains information about the adjacency of sector modules with respect to vertical faces.

23. Prepronode

Purpose

This function identifies the sector modules that are adjacent to one another with respect to nodes.

Input

This contains information about the sector modules.

Output

This contains information about the adjacency of sector modules with respect to nodes.

24. Prepro_airports

Purpose

This function scans the input file regarding the airports and identifies the sector modules the airports lie in.

Input

This contains information about the sector modules.

Output

This contains information about the airports.

25. Prepro

Purpose

This function obtains the mathematical representation of the sector modules when the vertices are defined in an anti-clockwise fashion.

Input

This contains information about the sector modules.

Output

This contains the mathematical representation of the sector modules.

26. Prepro_sdat

Purpose

This function obtains the mathematical representation of the sector modules when the vertices are defined in a clockwise fashion.

Input

This contains information about the sector modules.

Output

This contains the mathematical representation of the sector modules.

27. Prepro_sectors

Purpose

This function does the preprocessing of the sector information. It determines the mathematical representation of the sector modules, determines the adjacency information and identifies the ex-

trema faces of the defined airspace.

Input

This contains information about the sector modules.

Output

This contains the mathematical representation of the sector modules, the adjacency information and the information about the extreme faces.

28. Process_Fp**Purpose**

This function scans the flight plan input file, does the pre-processing of the flight plan data and determines the occupancy information.

Input

This contains information about the sector modules.

Output

This contains the occupancy information.

29. Prepro_Fp**Purpose**

This function does the pre-processing of the flight plan information.

Input

This contains information about the flight plans, airports and the sector modules.

Output

This contains pre-processed flight plan information.

30. Readetms

Purpose

This function scans the input file for flight plans. The input file should be in FAA ETMS Format.

Input

This contains the name of the input file.

Output

This contains the flight plan information.

31. Read_opt_reqd

Purpose

This function scans the input file for flight plans. The input file should be in FAA ETMS Optimized Trajectory Format.

Input

This contains the name of the input file.

Output

This contains information about the flight plans.

32. Read_opt_reqd_t

Purpose

This function scans the input file for flight plans corresponding to flights which are in the air-space during the time of interest. The input file should be in FAA ETMS Optimized Trajectory Format (Appendix B).

Input

This contains the name of the input file and time of interest.

Output

This contains the information about the flight plans corresponding to flights which are in the air-space during the time of interest.

33. Read_sdat_node**Purpose**

This function scans the input file which contains information about the nodes that define the sector modules. The input file should be in the FAA SDAT Generic Format (Appendix B).

Input

This contains the name of the input file.

Output

This contains information about the nodes.

34. Read_sdat_sect**Purpose**

This function scans the input file which contains information about the sector modules. The input file should be in the FAA SDAT Generic Format (Appendix B).

Input

This contains the name of the input file and the information about the nodes.

Output

This contains information about the sector modules.

35. Tocheck_vertex**Purpose**

This function determines all the nodes that are present on the faces of sector modules but are not originally defined for it. This m-file also updates the adjacency information on the sector modules with respect to nodes and with respect to each other.

Input

This contains information about the sector modules, the nodes and the adjacency information with respect to nodes and sector modules.

Output

This contains revised information about the sector modules and the adjacency relationships of the sector modules with respect to nodes and each other.

36. View_main_S**Purpose**

This function plots the sector of interest in three dimensions.

Input

This contains information about the sector and sector modules, and the number of the sector of interest.

Output

This contains the plot of the sector of interest in three dimensions.

37. View_sect_Fp_h**Purpose**

This function plots the flight trajectories and the sector modules present at a particular altitude of interest.

Input

This contains information about the sector modules, the flight trajectories, and the altitude of interest (hundreds of feet).

Output

This contains the plot of the flight trajectories and the sector modules present at a particular altitude of interest.

38. View_sect_ht**Purpose**

This function plots the sector modules present at a particular altitude of interest.

Input

This contains information about the sector modules and the altitude of interest (hundreds of feet).

Output

This contains the plot of the sector modules present at a particular altitude of interest.

APPENDIX B ARTCC Sector Traffic

This appendix shows various enroute control sector traffic demand patterns under current NAS conditions, RVSM and Cruise Climb Scenarios. These traffic flows have been derived using 8,000 flights for ZMA and ZJX and 6,000 flights for ZTL and ZID as explained in Chapter 5 of this report.

Table B.1 ZMA Center Sector Traffic Patterns.

Sector	Baseline Traffic	RVSM Traffic	CC Traffic	PTC (Base vs RVSM) (%)	PTC (Base vs CC) (%)
D01	117	108	108	7.69	7.69
D02	154	155	155	-0.65	-0.65
D03	2	0	0	100.00	100.00
D04	8	5	5	37.50	37.50
D05	83	68	68	18.07	18.07
D06	86	64	64	25.58	25.58
D07	29	27	27	6.90	6.90
D08	141	131	130	7.80	7.80

Table B.1 ZMA Center Sector Traffic Patterns.

Sector	Baseline Traffic	RVSM Traffic	CC Traffic	PTC (Base vs RVSM) (%)	PTC (Base vs CC) (%)
D09	0	0	0	0	0
D10	0	0	0	0	0
D20	121	110	110	9.09	9.09
D21	69	64	64	7.25	7.25
D22	45	40	40	11.11	11.11
D23	58	53	53	8.62	8.62
D24	75	78	73	2.67	2.67
D25	165	169	164	0.61	0.61
D26	14	10	10	28.57	28.57
D32	39	39	39	0.00	0.00
D33	56	54	54	3.57	3.57
D34	58	57	57	1.72	1.72
D39	92	91	91	1.09	1.09
D40	165	151	151	8.48	8.48
D41	84	75	75	10.71	10.71
D42	69	64	64	7.25	7.25
D45	3	2	2	33.33	33.33
D46	129	118	118	8.53	8.53
D47	148	155	154	-4.05	-4.05
AMIS	0	0	0	0	0
D59	51	46	46	9.80	9.80
D60	144	137	137	4.86	4.86
D61	28	24	24	14.29	14.29
D62	155	146	146	5.81	5.81
D63	126	122	122	3.17	3.17
D64	110	127	126	-14.55	-14.55

Table B.1 ZMA Center Sector Traffic Patterns.

Sector	Baseline Traffic	RVSM Traffic	CC Traffic	PTC (Base vs RVSM) (%)	PTC (Base vs CC) (%)
D65	117	111	110	5.98	5.98
D66	1	1	0	100.00	100.00
D67	85	86	86	-1.18	-1.18
D71	0	0	0	0	0

Table B.2 ZJX Center Sector Traffic Patterns.

Sector	Baseline Traffic	RVSM Traffic	CC Traffic	PTC (Base vs RVSM) (%)	PTC (Base vs CC) (%)
CEW	48	44	44	8.33	8.33
BTN	107	136	137	-27.10	-28.04
ABY	2	1	1	50.00	50.00
ASH	28	25	1	10.71	96.43
CDK	175	182	184	-4.00	-5.14
OCF	145	166	166	-14.48	-14.48
MAY	157	173	172	-10.19	-9.55
FPY	204	205	204	-0.49	0.00
TLH	41	41	41	0.00	0.00
AYS	35	31	31	11.43	11.43
NPT	137	135	134	1.46	2.19
GEN	103	114	112	-10.68	-8.74
SEM	186	132	126	29.03	32.26
LKE	110	172	173	-56.36	-57.27
SMV	137	143	143	-4.38	-4.38
GGE	38	44	45	-15.79	-18.42

Table B.2 ZJX Center Sector Traffic Patterns.

Sector	Baseline Traffic	RVSM Traffic	CC Traffic	PTC (Base vs RVSM) (%)	PTC (Base vs CC) (%)
MGR	138	143	130	-3.62	5.80
AMG	141	132	151	6.38	-7.09
CHS	44	24	25	45.45	43.18
SSI	27	23	24	14.81	11.11
JEK	3	1	1	66.67	66.67
SJS	181	138	139	23.76	23.20
SGJ	158	123	123	22.15	22.15
RWY	57	30	30	47.37	47.37
AIK	84	54	54	35.71	35.71
HUN	86	78	78	9.30	9.30
TBR	103	100	99	2.91	3.88
TOR	153	136	137	11.11	10.46
MTA	156	142	144	8.97	7.69
FLO	29	21	21	27.59	27.59
CAE	47	31	31	34.04	34.04
ALD	43	28	29	34.88	32.56
GCS	137	126	134	8.03	2.19
KEY	95	121	103	-27.37	-8.42
SSP	2	2	2	0.00	0.00
TAY	27	25	25	7.41	7.41
GNV	0	0	0	0	0

Table B.3 ZID Center Sector Traffic Patterns.

Sector	Baseline Traffic	RVSM Traffic	CC Traffic	PTC (Base vs RVSM) (%)	PTC (Base vs CC) (%)
EV	4	5	5	-25.00	-25.00
SD	40	63	63	-57.50	-57.50
LE	24	17	17	29.17	29.17
CV	437	302	302	30.89	30.89
CR	8	8	8	0.00	0.00
CM	115	108	108	6.09	6.09
DA	169	171	171	-1.18	-1.18
IN	48	55	55	-14.58	-14.58
HU	27	32	32	-18.52	-18.52
HT	4	4	4	0.00	0.00
25	9	9	9	0.00	0.00
26	103	64	64	37.86	37.86
35	83	93	94	-12.05	-13.25
80	226	205	204	9.29	9.73
99	53	158	138	-198.11	-160.38
81	215	211	216	1.86	-0.47
82	252	230	246	8.73	2.38
83	217	123	133	43.32	38.71
84	263	132	139	49.81	47.15
85	271	215	231	20.66	14.76
86	204	161	165	21.08	19.12
87	201	208	223	-3.48	-10.95
88	231	224	228	3.03	1.30
89	261	253	266	3.07	-1.92

Table B.3 ZID Center Sector Traffic Patterns.

Sector	Baseline Traffic	RVSM Traffic	CC Traffic	PTC (Base vs RVSM) (%)	PTC (Base vs CC) (%)
91	37	165	146	-345.95	-294.59
92	27	180	166	-566.67	-514.81
93	29	136	126	-368.97	-334.48
94	43	127	103	-195.35	-139.53
95	30	203	169	-576.67	-463.33
96	25	91	75	-264.00	-200.00
97	32	123	96	-284.38	-200.00
98	88	160	153	-81.82	-73.86

Table B.4 ZTL Center Sector Traffic Patterns.

Sector	Baseline Traffic	RVSM Traffic	CC Traffic	PTC (Base vs RVSM) (%)	PTC (Base vs CC) (%)
AT	704	324	324	53.98	53.98
BH	25	20	20	20.00	20.00
CH	0	0	0	0	0
CL	53	63	63	-18.87	-18.87
CS	3	3	3	0.00	0.00
GP	20	19	19	5.00	5.00
GO	38	31	31	18.42	18.42
MX	4	4	4	0.00	0.00
TR	1	1	1	0.00	0.00
TY	13	12	12	7.69	7.69
MA	3	3	3	0.00	0.00

Table B.4 ZTL Center Sector Traffic Patterns.

Sector	Baseline Traffic	RVSM Traffic	CC Traffic	PTC (Base vs RVSM) (%)	PTC (Base vs CC) (%)
AG	2	0	0	100.00	100.00
AV	2	2	2	0.00	0.00
1	0	0	0	0	0
2	79	113	111	-43.04	-40.51
3	181	142	146	21.55	19.34
4	113	81	81	28.32	28.32
16	111	42	42	62.16	62.16
5	112	78	78	30.36	30.36
6	98	145	150	-47.96	-53.06
8	35	43	44	-22.86	-25.71
9	105	51	51	51.43	51.43
10	114	55	56	51.75	50.88
11	92	71	73	22.83	20.65
12	39	29	29	25.64	25.64
13	18	15	15	16.67	16.67
14	36	22	24	38.89	33.33
15	72	82	74	-13.89	-2.78
17	37	62	62	-67.57	-67.57
18	52	39	39	25.00	25.00
19	55	27	27	50.91	50.91
20	174	119	120	31.61	31.03
21	152	61	61	59.87	59.87
22	286	186	187	34.97	34.62
23	202	178	166	11.88	17.82
24	103	66	65	35.92	36.89
28	60	37	36	38.33	40.00

Table B.4 ZTL Center Sector Traffic Patterns.

Sector	Baseline Traffic	RVSM Traffic	CC Traffic	PTC (Base vs RVSM) (%)	PTC (Base vs CC) (%)
29	42	28	28	33.33	33.33
30	103	92	92	10.68	10.68
31	138	116	116	15.94	15.94
32	208	135	142	35.10	31.73
33	227	167	167	26.43	26.43
34	224	163	158	27.23	29.46
36	39	115	113	-194.87	-189.74
37	68	105	110	-54.41	-61.76
38	81	29	29	64.20	64.20
39	161	133	130	17.39	19.25
40	30	83	84	-176.67	-180.00
41	4	1	1	75.00	75.00
42	113	93	95	17.70	15.93
43	113	92	97	18.58	14.16
44	21	38	38	-80.95	-80.95
45	18	13	13	27.78	27.78
46	5	5	5	0.00	0.00
47	34	31	31	8.82	8.82
48	15	15	15	0.00	0.00
49	122	37	37	69.67	69.67
50	137	97	105	29.20	23.36