Integration of Reusable Launch Vehicles into Air Traffic Management

Phase II Final Report (VPI)

NEXTOR Research Report RR-98-15

Hanif D. Sherali (Principal Investigator)
Cole Smith (Graduate Research Assistant)

Department Industrial and Systems Engineering
Virginia Polytechnic Institute and State University
Blacksburg, VA 24061

Antonio A. Trani (Principal Investigator)
Srinivas Sale (Graduate Research Assistant)
Chuanwen Quan (Graduate Research Assistant)

Department of Civil Engineering
Virginia Polytechnic Institute and State University
Blacksburg, VA 24061

November 30, 1998

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 Commercial Space, and in particular, the guidance and leadership provided by Kelvin Coleman in funding this effort.

Table of Contents

List of Figures

List of Tables

CHAPTER 1 Introduction

According to recent Federal Aviation Administration (FAA) statistics the total number of operations handled by the Air Traffic Control (ATC) system in the U.S. continues to grow at a modest pace of 2% per year (FAA, 1997). By the year 2010, it is expected that 35 million flights will be handled by the FAA at various Enroute (ARTCC) and Terminal Area Traffic Control Centers (TRACON). Paralleling this growth is the number of space transportation actitivies which is anticipated to increase significantly into the next millennium. Some predictions foresee up to 1,200 satellite launches into the next decade to replace, upgrade, and improve weather, telecommunications and military assets in space (Aviation Week and Space Technology, 1998). As many as 50% of these projected launches could be carried from the Continental U.S., thus causing small to moderate disruptions to National Airspace System (NAS) commercial and General Aviation (GA) flight operations. The purpose of this research effort is to develop tools and techniques to quantify and minimize the effect of spaceport operations to commercial and GA flights.

This report summarizes the results of a preliminary task given to the National Center of Excellence for Aviation Operations Research to develop a framework to integrate Reusable Launch Vehicles (RLV) into Air Traffic Management (ATM). Some of the models developed as part of this research effort constitute a flexible toolset to quantify airspace operations around spaceports. The study is divided into two tasks: a) to examine various modes of RLV operation (a study performed by the Massachusetts Institute of Technology - MIT), and b) to study the economic impacts of RLV operations as they affect commercial and GA traffic around spaceports (Virginia Polytechnic Institute and). State Ur

This report deals with the development of three computer models to study airspace sector occupancy, predict aircraft encounters that would impose a measure of cost to the FAA, and an optimization model to minimize the cost of detours around Special Use Airspace (SUA) regions due to RLV operations. As part of the methodology proposed in this report we also include a cost analysis of aircraft operations, and the estimation of delays associated with affected airport operations. This report includes research activities performed at Virginia Polytechnic Institute and State University over the period October, 1997 through October, 1998.

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 FL 240 (i.e. flight level 24,000 ft.). 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. The relevance of all these facts for RLV operations is that under current NAS operations, ATC controllers are generally isolated from RLV actions that entail the establishment of large volumes of protected airspace around the atmospheric phases (i.e., reentry and ascent) of operating spacecraft. While SUA regions provide a simple and effective way to “sterilize” the airspace around a spaceport, they also increase the detour and delay costs of commercial and GA flights in the region. For example, a typical commercial flight traveling from Washington or Boston to Miami and traversing the airspace near the Cape Canaveral Warning Area 497B (see Figure 1.1) would suffer an optimal detour of 7-8[1] minutes if the SUA region around the Kennedy Space Center is activated as a result of a space launch. While this fact might seem insignificant at first glance, such delay induced disruptions at hub airports in the South can amplify costs via a domino-effect into more significant magnitudes.

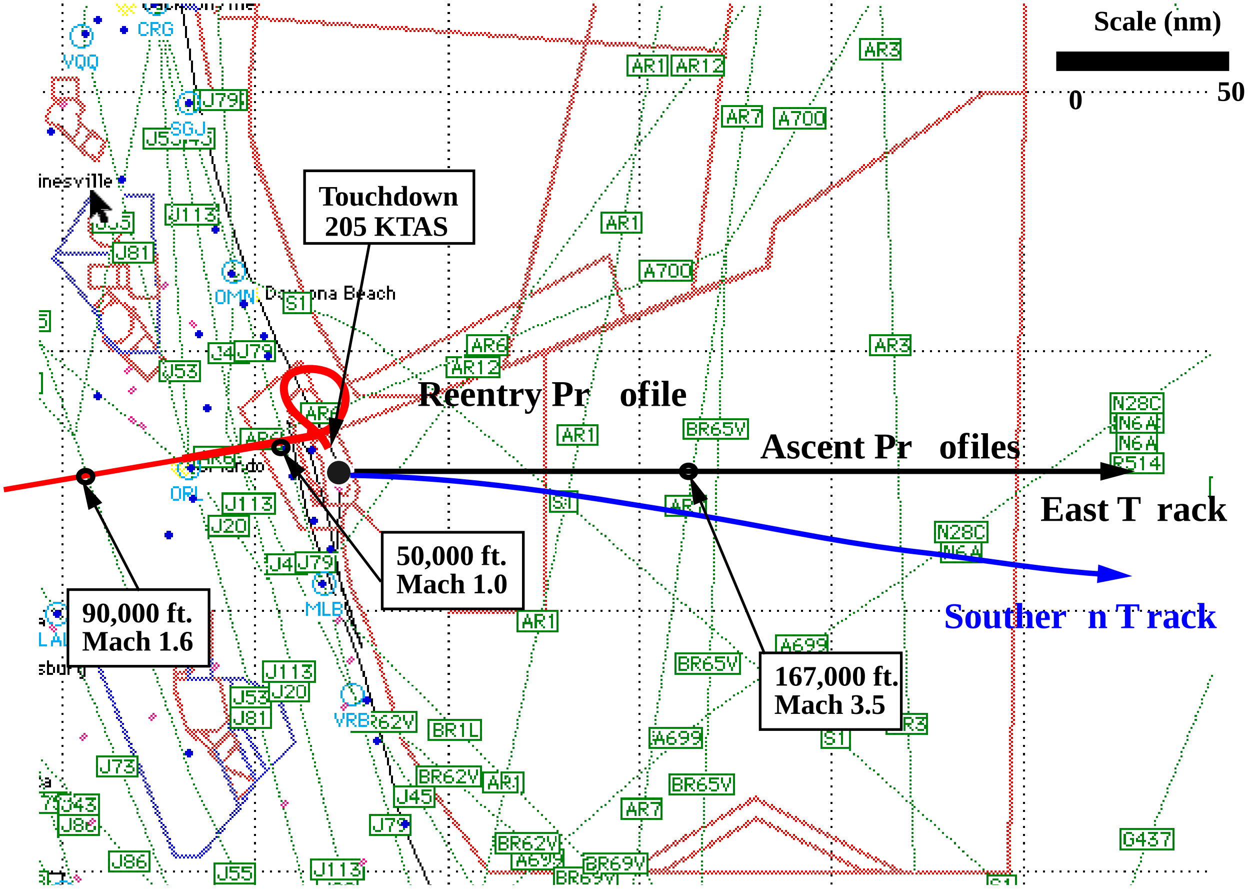


Figure 1.1 Typical Space Shuttle Transportation System Ascent and Reentry Tracks and Cape Canaveral Warning Areas.

[1] Using an MD-80 aircraft as sample vehicle.

A recent study by the MITRE Corporation for two Delta vehicle launches in the Canaveral Area indicates that 10.1 minutes of airborne delay are associated with typical flights affected by these launches. The same study indicates that up to $1.040 million dollars in user costs could potentially be associated with a heavy launch manifest of two launches per week over a year (MITRE, 1997). While this analysis provides an indication of the airspace user costs associated with spaceport operations, it only addresses one aspect of the RLV operations for the following reasons:

a) the number of launches investigated is quite small to represent a solid database;

b) the estimation provides an estimate of costs under conditions that are likely to change under Free Flight operations, and

c) non-user and service provider costs are not included in the analysis.

MITRE is currently conducting an evaluation of the spaceport operations from Kodiak Island. The analysis proposed in this report represents a complementary approach to the MITRE analysis and provides a methodology to reduce user and service costs in a rational basis. The approach proposed here, including the use of the models developed, equally applies to current or future concept of operations in NAS.

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 scenario of operations is intended to provide a decentralized air traffic control service that is more cost effective from an overall FAA/ATC and airspace users viewpoint. In a critical situation, however 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. Free Flight will certainly have an effect on RLV operations. In general, as individual aircraft adopt user-optimal flight plans, the degree of interaction between RLV and commercial and GA flights is expected to increase. Two reasons are responsible for this: a) normal growth in airspace operations as Free Flight matures and b) more even distribution of flights across various sectors of NAS due to increased freedom to select lateral and vertical trajectories optimized for predicted wind conditions. Figures 1.2 and 1.3 illustrate this point. In both figures, the same 700 random flight plans originating or ending in Miami are selected and plotted to show the intended flight plan trajectories. The data used as this baseline scenario has been gathered from the 1996 ETMS database. Figure 1.2 shows a well defined route structure using standard jet airways. Figure 1.3 illustrates a more random flight plan pattern as a result of wind-optimized trajectories, where waypoints are not dictated by ground Navigational Aids (NAVAIDS). This flexibility in flight planning will be responsible for more interactions between RLV spaceports and commercial and GA aircraft. Chapter 6 of this report provides further details of this analysis.

1.3 Research Scope, Objective and Approach

In the future Air Traffic Management System, it is imperative to have a set of models to understand aircraft flows across regions of congested airspace. This is necessary to reduce the costs imposed on airspace users and service providers. Such models may serve as an advisory tool to: 1) approve flight plans that offer minimum interaction with other flights or RLV vehicles if they are integrated as high priority flights, and 2) reschedule flights around Special Use Airspace (SUA) areas such as in the event of spaceport launches at a minimum cost to users, service providers, and space launch customers.

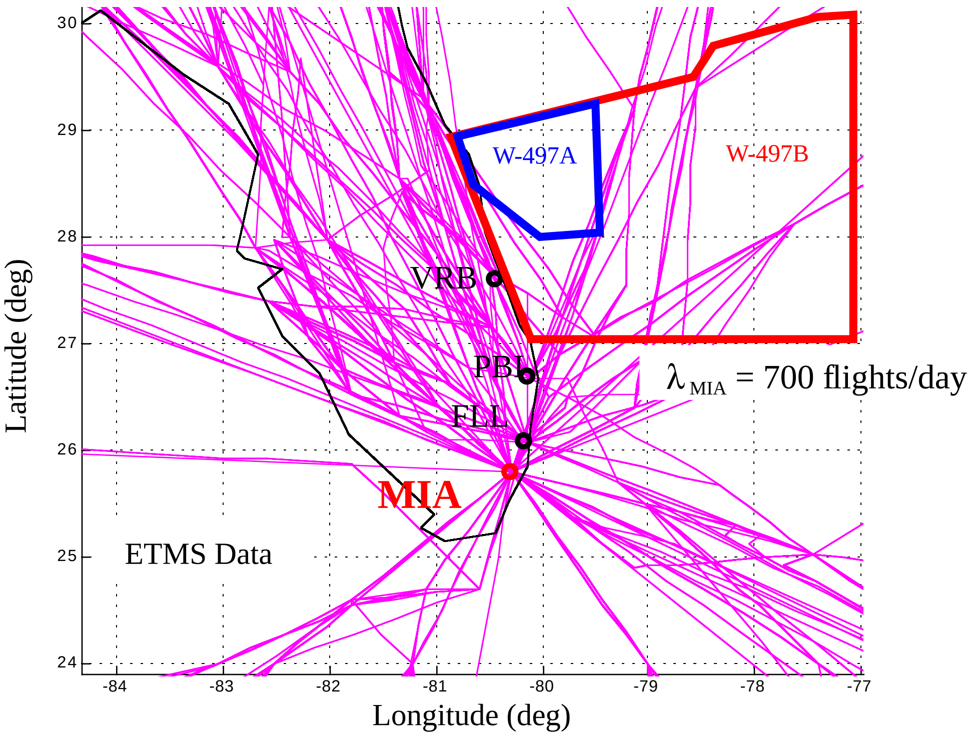


Figure 1.2 Partial View of Filed Flight Plans Across the Airspace in Southern Florida (Current NAS Operations).

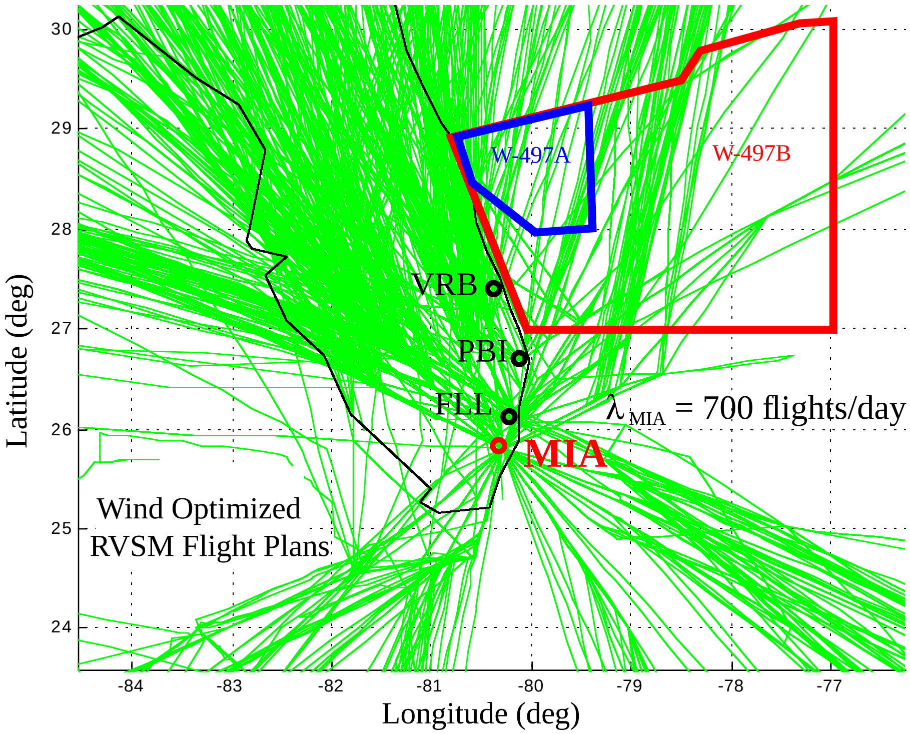


Figure 1.3 Partial View of Filed Flight Plans Across the Airspace in Southern Florida (Future NAS Operations Using Wind-Optimized RVSM Flight Plans).

The specific goals of this research project are:

1) to determine the impacts of RLV operations around spaceports, and

2) to develop tools and methods to mitigate impacts of RLV integration into the Air Traffic Management system.

Three 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), 2) the Airspace Encounter Model (AEM) and 3) an Airspace Planning Model (APM). AOM determines sector loads given flight plans or flight tracks. AEM estimates airspace conflicts, and their severity and geometries, if all these flight plans are considered collectively. Finally, APM uses the outputs of AOM and AEM to determine the best set of flight plans among submitted alternatives that would produce minimum disruptions around the SUA region of interest while considering equity among flights affected and workload balance for the service provider. All models can be executed on any Windows 95/NT compatible PC, Macintosh, or Unix Workstation without modifications.

In summary, the overall model APM (which encompasses the submodels APM and AEM) could be utilized in one of two ways.

(1) Generator of a suitable mix of flight plans for a set of flights operating in the vicinity of a spaceport: In this role, the model can be coordinated with other tools such as RAMS or even used in standalone mode to evaluate in more detail the airspace operations around an activated SUA.

(2) Policy Evaluator of various what-if scenarios proposed by policy/decision makers in determining operational guidelines around SUA induced by space operations.

Hence, the model can be used, both, in a tactical decision-making mode, as well as for generating strategic plans to detour flights around SUA regions. The framework of the modeling approach is illustrated in Figure 1.4. Here the module labeled “Modes of Operation” includes a detailed analysis of various RLV modes of operation being studied by MIT (Khan and Kuchar, 1998). AOM and AEM generate inputs for APM, which in turn, considers alternative flight plans for each flight plan among them based on safety, cost, equity, and workload considerations. In Figure 1.4 we advocate the use of standard airspace simulation models to estimate the impacts of flight detours without any optimization features in place. This is useful because the FAA and space transportation decision makers need to gage the benefit of the optimization model developed. In other words, by comparing the results of simulation models such as RAMS and SIMMOD with the outcome of the optimization model (APM), one can judge the savings between the status-quo and an optimized strategy, under future NAS operations around spaceports.

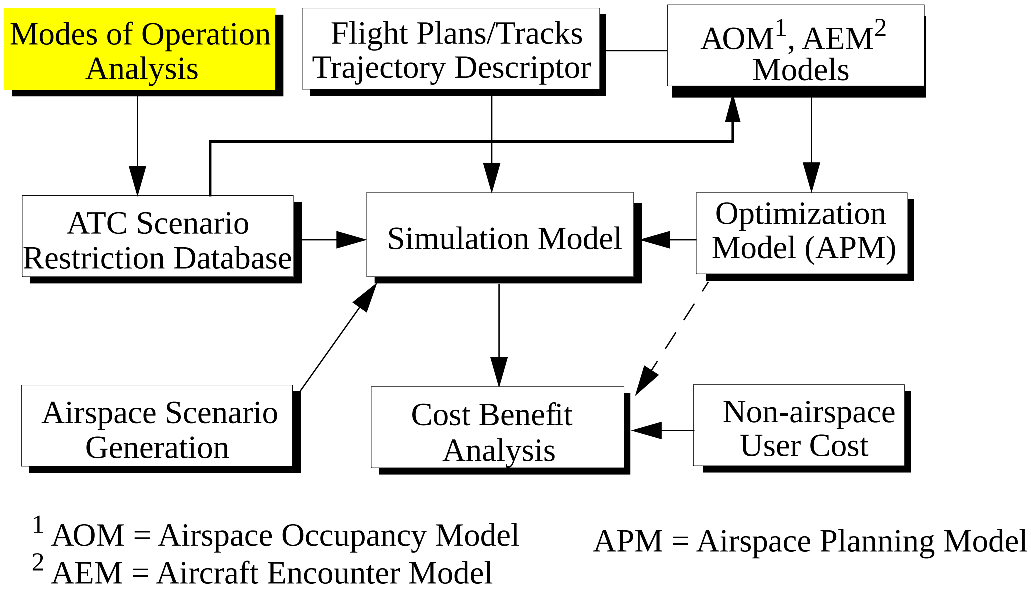

Modes of Operation Analysis
Flight Plans/Tracks Trajectory Descriptor
AOM[1], AEM[2] Models
ATC Scenario Restriction Database
Simulation Model
Optimization Model (APM)
Airspace Scenario Generation
Cost Benefit Analysis
Non-airspace User Cost
[1] AOM = Airspace Occupancy Model
[2] AEM = Aircraft Encounter Model
APM = Airspace Planning Model

Figure 1.4 Framework to Assess the Impact of RLV Operations in NAS.

The remainder of this research report is organized as follows. This chapter has merely introduced the reader to the justification of this research project. Chapter 2 includes a review of pertinent literature on material relevant to RLV analysis and impacts. Chapter 3 describes in detail the Airspace Occupancy Model (AOM) developed as part of this effort. This model estimates aircraft occupancies in any mathematically defined region of airspace. Chapter 4 describes the Airspace Encounter Model (AEM) used to estimate the level of interactions between flights traversing a defined region of airspace. Chapter 5 describes the principles of the mathematical optimization model developed to schedule a set of detour flights around regions of airspace affected by RLV operations. Chapter 6 illustrates some of the analysis performed with these three models including some calibration of the models to verify their accuracy. Finally, Chapter 7 discussed preliminary results using these models, and concludes with further research recommendations for Phase III of this research effort.

CHAPTER 2 Literature Review

This chapter reviews various airspace analysis models and tools that provide some capability for quantifying traffic density, airspace delays and RLV operations. We briefly mention some modeling tools that have overlapping capabilities with respect to AOM, AEM, and APM in order to justify their development. Several simulation models have been used by the research team that have prompted the development of AOM, AEM and APM and thus are also described here. Other models dealing with allocation of resources are also addressed here for the sake of completeness.

2.1 Airspace Models

There are numerous computer simulation models that estimate aircraft behavior in the airspace. Examples of these are SIMMOD - the FAA airspace and airfield simulation model, RAMS - Eurocontrol's Reorganized Mathematical Simulator model, and TAAM - a privately funded airspace model developed by the Preston Group to name a few. Some characteristics of these models and their possible use in RLV integration are described below.

SIMMOD

SIMMOD is a fast-time simulation model used to estimate airspace and airfield delays in complex network structures. This model, developed under the auspices of FAA since the 1970's, has been used in various sectorization studies in this country and abroad to determine the efficiency of airspace operations around airports. The model uses a discrete-event simulation doctrine to estimate aircraft delays for individual vehicles moving along a predefined airspace-airfield network structure. The model was originally developed as a fuel consumption prediction tool, but over the years evolved as a fast-time simulator to predict operational benefits of airport improvements and airspace modifications. Ironically, today, the model is seldom used for fuel consumption estimation due to a limited aircraft fuel consumption model database. This model has been coded in the popular SIMSCRIPT II.5 simulation language (CACI, 1996), and various versions of the model exist. The model can be executed on PCs and workstations (UNIX running earlier versions of the Solaris operating system). A new Graphic User Interface (GUI) has been developed by the ATAC Corporation using the Java computing language.

As it pertrains to our analysis, SIMMOD is a useful model as it provides a mechanism for estimating aircraft delays in the air and on the ground that are associated with spaceport operations. This package requires the modeler to include some definition of the airfield networks to assess possible interactions between departure and arrival events due to spaceport launches and airspace operations, subject to imposed SUA restrictions. On the negative side, this model demands a substantial input information including network definition, demand patterns, routes, airspace operational restrictions, etc.

RAMS

RAMS is a fast-time simulation model developed by Eurocontrol for quantifying airspace interactions, including workload measures. This model involves an explicit modeling of the airspace network structure, using airports as sinks and sources, and routes as arcs. The model has been developed in MODSIM, an object-oriented language developed by CACI in the 1980's. RAMS uses a discrete-event simulation doctrine to estimate aircraft interactions in the airspace, including a user-defined conflict detection and resolution maneuvers predicated on a rule-based mechanism. RAMS also has a simulation mode where the user can interactively participate in the resolution of conflicts for the purpose of evaluating new ATC procedures. The primary motivation in the development of this model has been to measure the workload imposed on airspace sectors. The model treats airspace sectors in good detail and derives measures of workload based on the number of conflicts in a sector over time.

Find Crossings

This model has been developed by CSSI Incorporated to identify sector piercing points as aircraft traverse airspace sectors over the National Airspace System. This model has been coded in the C language and runs under HP-UX 10.2 and Solaris 2.6 operating systems. The model takes the output from OPGEN (another model developed by CSSI in support of NARIM and described later), or flight plans derived from ETMS data, and finds sector crossing points including crossing times. The functionality of Find Crossings has certain overlaps with that of the AOM model described in Chapter 3 of this report. This functionality was necessary because the VPI research team did not have a compiled version of Find Crossings until January of 1998. The Office of Investment Analysis at the FAA offered Find Crossings to the NEXTOR research team last summer (1997) but due to the strict requirements in computer platform (at that time the model only ran on a Hewlett Packard HP-UX 9.0 equipment) and operating systems we could not execute the model and it was decided to generate an independent tool that would be more versatile. AOM offers some advantages over Find Crossings. For example, the model can be executed on any platform. The VPI research team has executed this model on various versions of the Mac OS (8.0 and 8.5), Windows 95/NT PC workstations and at least three flavors of UNIX systems (HP-UX-10.2, Irix 6.5 on Silicon Graphics O2 and Origin 2000 systems, and Solaris 2.5 on a Sun HyperSparc120). As previously stated, all models developed in this research effort use standard computer packages such as Matlab and CPLEX, thus making the code quite portable without modifications. On the negative side, AOM executes slower than Find Crossings due to its interpretive nature. As of this writing, we have not been able to run any comparative benchmark scenarios as yet, but the new release of Find Crossings is due in the month of January, 1999 and we hope to further validate the accuracy of the results of both models.

Total Traffic Tool

This is another interesting model developed by CSSI in support of ASD’s mission to execute airspace analysis under various concepts of operation. This model has some overlapping capabilities with AEM described in Chapter 4 of this report. In essence, the model basically takes outputs from Find Crossings and estimates traffic loads on airspace sectors. Some of the mathematical algorithms used by the model are not known and we are not aware of its flexibility to accommodate various protection envelopes around an aircraft to detect possible conflicts. In contrast, AEM provides details on the severity of conflicts as well as conflict geometries, using both detailed and aggregated metrics.The Total Traffic Tool has been coded in the C language and runs using the HP-UX 9.0 operating system, but apparently has been recently ported to run in Solaris 2.6.

CHAPTER 3 Airspace Occupancy Model

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 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 in 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.

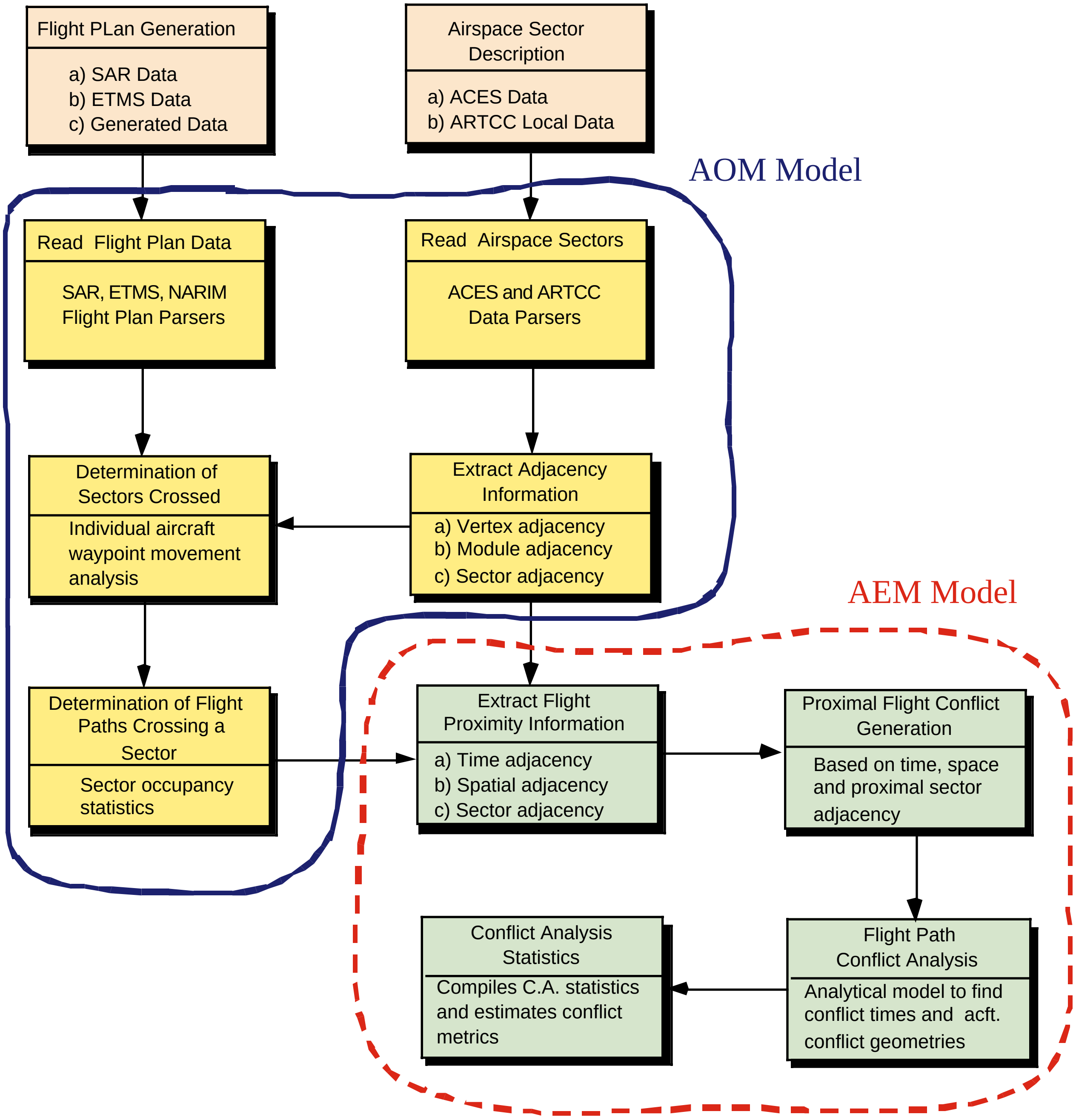
3.3 Airspace Sector Description

Sectors are well-defined airspace regions specified by the FAA for regulating air traffic. Each sector is comprised of Fixed Posting Airspace units (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.


Flight PLan Generation
a) SAR Data
b) ETMS Data
c) Generated Data
Airspace Sector Description
a) ACES Data
b) ARTCC Local Data
AOM Model
Read Flight Plan Data
SAR, ETMS, NARIM Flight Plan Parsers
Read Airspace Sectors
ACES and ARTCC Data Parsers
Determination of Sectors Crossed
Individual aircraft waypoint movement analysis
Extract Adjacency Information
a) Vertex adjacency
b) Module adjacency
c) Sector adjacency
AEM Model
Determination of Flight Paths Crossing a Sector
Sector occupancy statistics
Extract Flight Proximity Information
a) Time adjacency
b) Spatial adjacency
c) Sector adjacency
Proximal Flight Conflict Generation
Based on time, space and proximal sector adjacency
Conflict Analysis Statistics
Compiles C.A. statistics and estimates conflict metrics
Flight Path Conflict Analysis
Analytical model to find conflict times and acft. conflict geometries

FIigure 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.

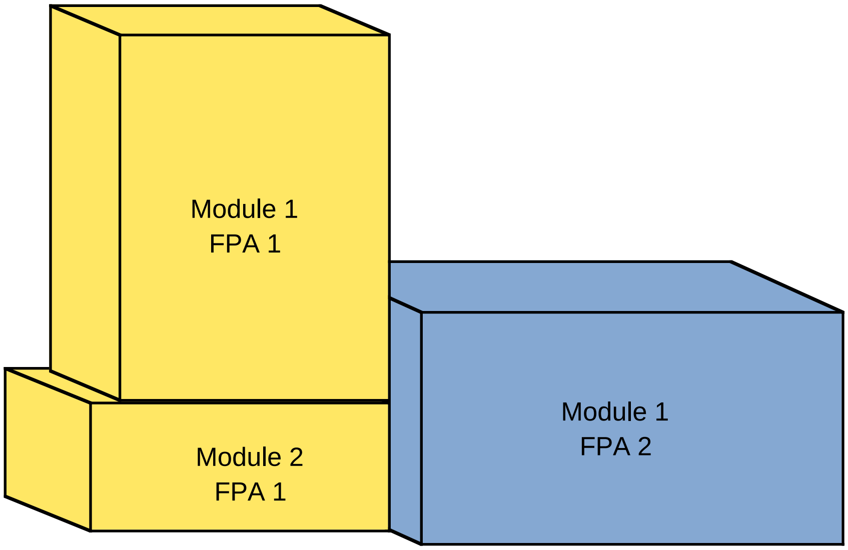


Figure 3.2 Typical Sector Geometry (showing a sector made up of 2 FPAs).

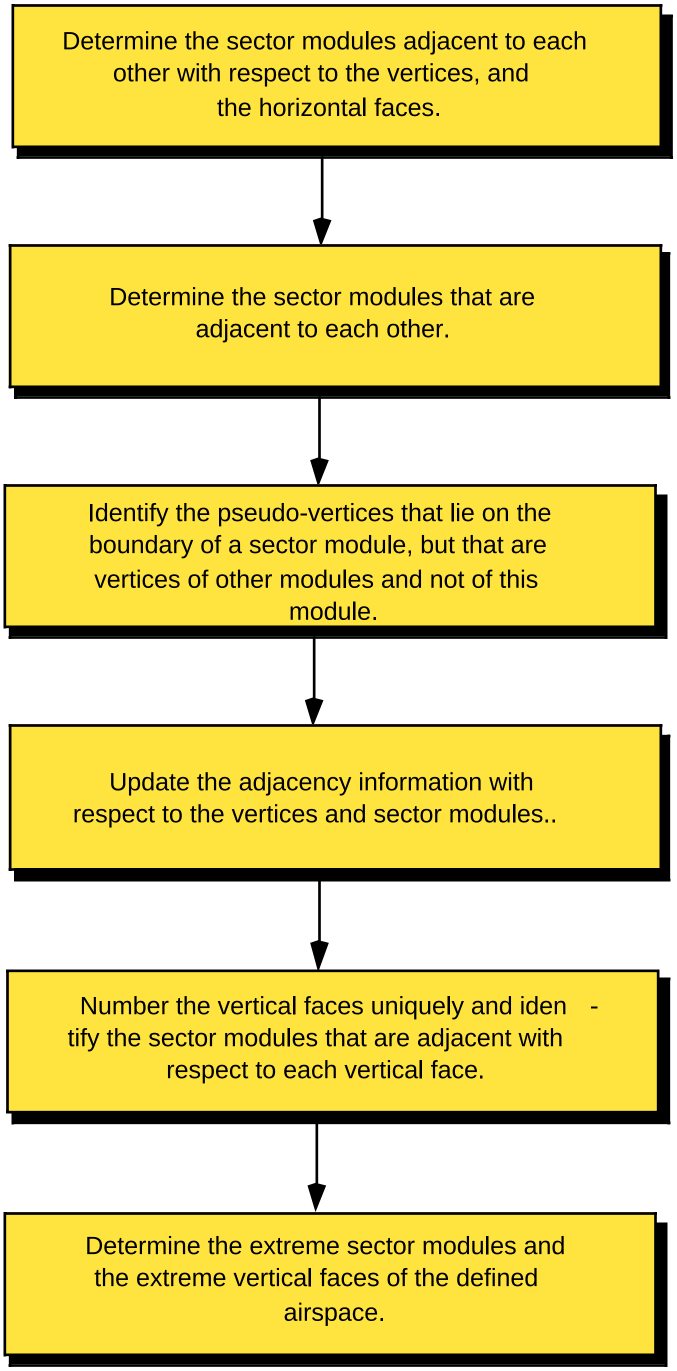

Determine the sector modules adjacent to each other with respect to the vertices, and the horizontal faces.
Determine the sector modules that are adjacent to each other.
Identify the pseudo-vertices that lie on the boundary of a sector module, but that are vertices of other modules and not of this module.
Update the adjacency information with respect to the vertices and sector modules..
Number the vertical faces uniquely and iden - tify the sector modules that are adjacent with respect to each vertical face.
Determine the extreme sector modules and the extreme vertical faces of the defined airspace.

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 $a \bullet X - c = 0$, 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.

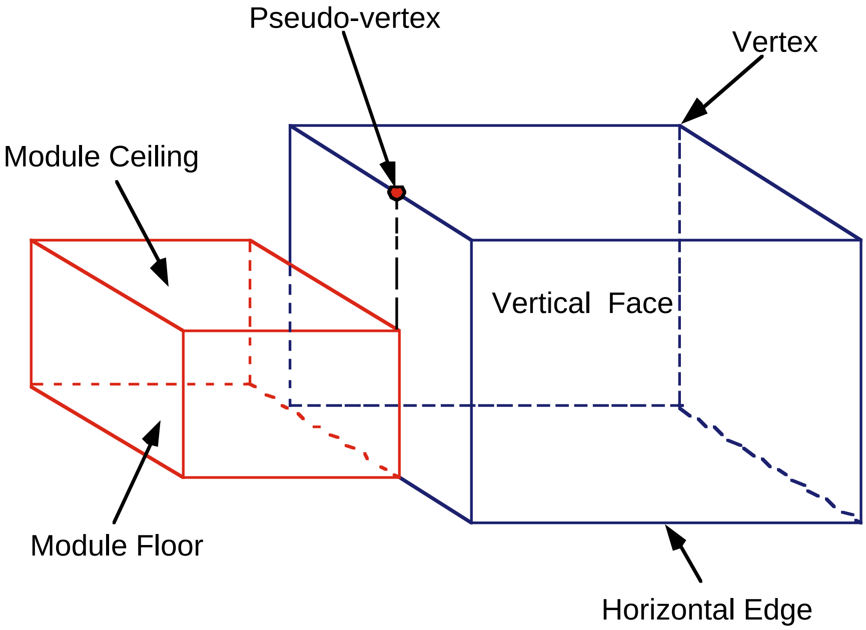


Figure 3.4 Geometric Components of a Sector Module.

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 $F_{ps} = [d_{p1}, -d_{p2}]$.

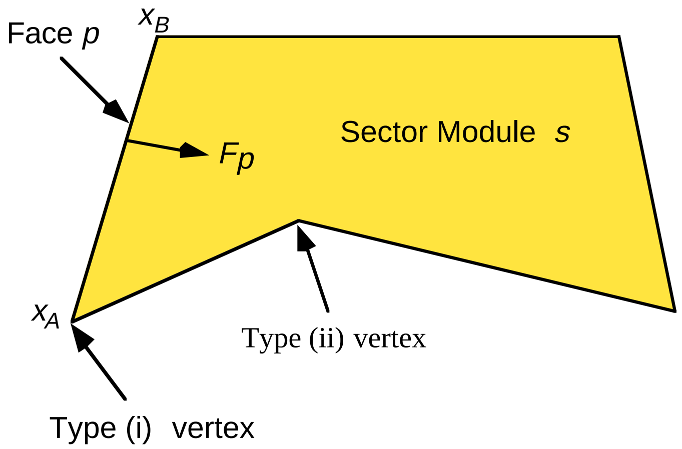


Figure 3.5 Basic Face and Vertex Definitions Inside a Sector Module.

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 the reader 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 $F_{qs} \bullet d \geq 0$.

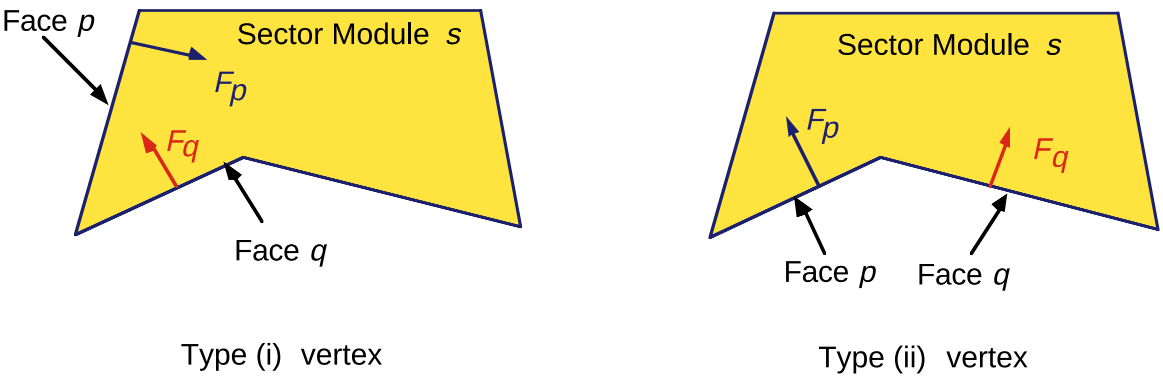


Figure 3.6 Types of Vertices.

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.

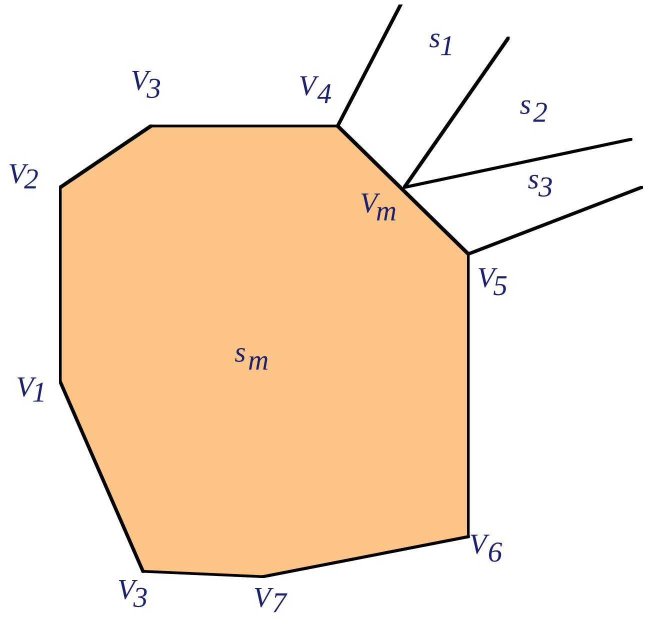


Figure 3.7 Adjacency with Respect to a Vertex.

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.

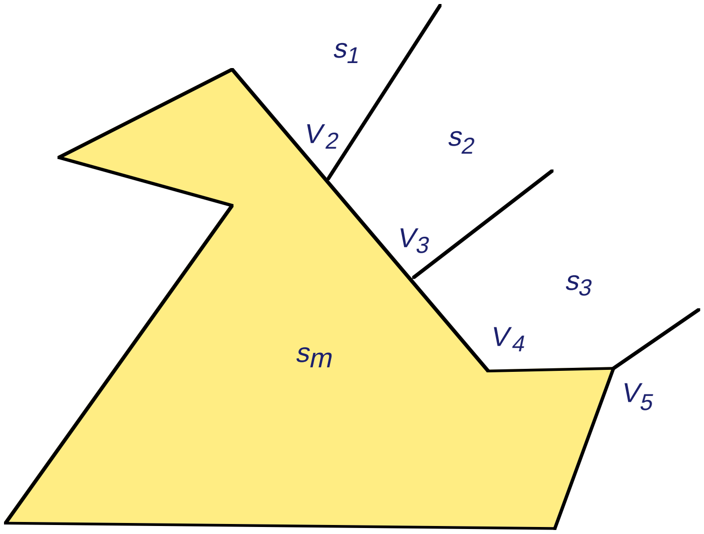


Figure 3.8 Adjacency with Respect to a Vertex.

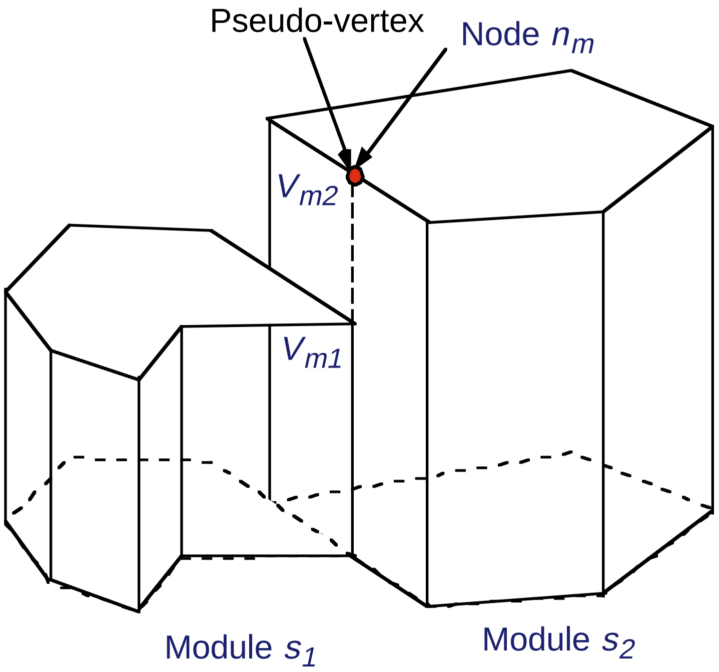


Figure 3.9 Pseudo-Vertices on a Face.

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,..., wp_i,..,.wp_n]. Let any linear segment of the trajectory be defined as for where .

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 $x_{new} = x_0 + \lambda_{new} d$.

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		◆	◆		◆

CHAPTER 4 Aircraft Encounter Model

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.

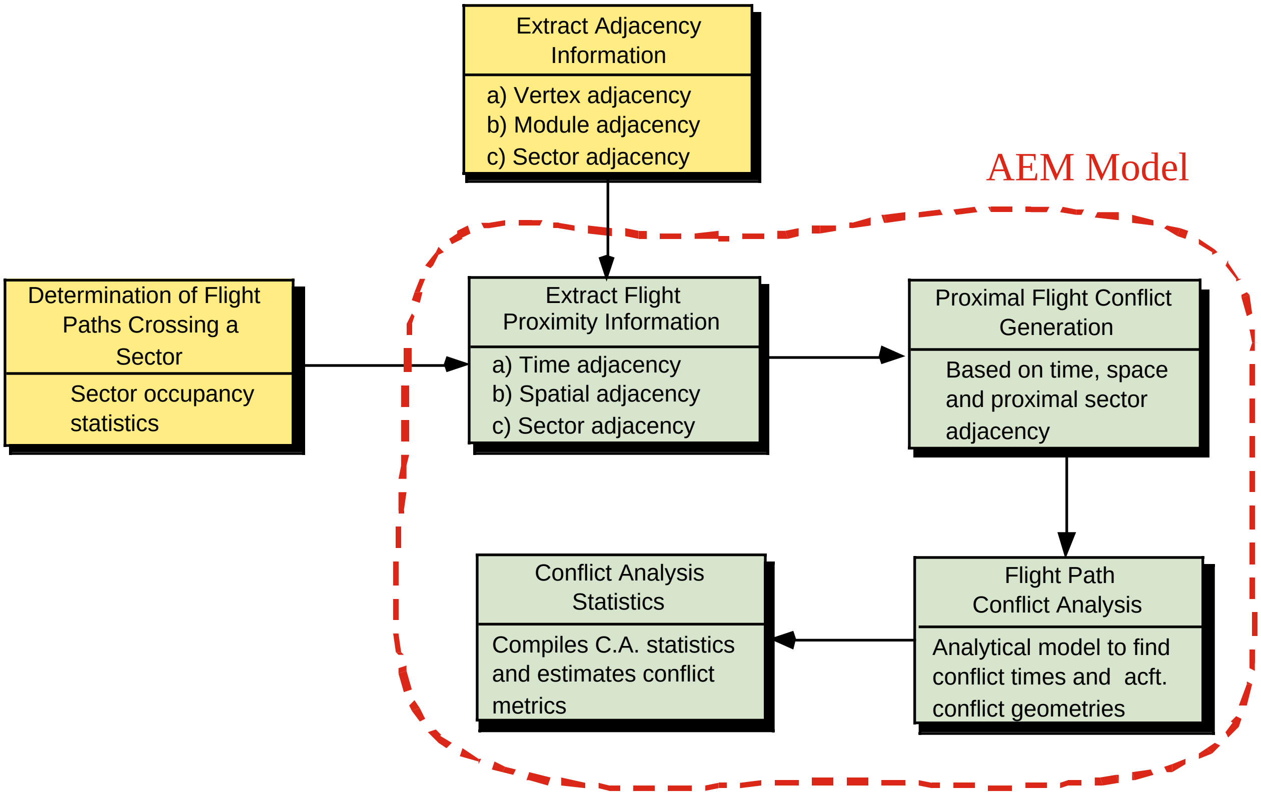

Extract Adjacency Information
a) Vertex adjacency
b) Module adjacency
c) Sector adjacency
AEM Model
Determination of Flight Paths Crossing a Sector
Sector occupancy statistics
Extract Flight Proximity Information
a) Time adjacency
b) Spatial adjacency
c) Sector adjacency
Proximal Flight Conflict Generation
Based on time, space and proximal sector adjacency
Conflict Analysis Statistics
Compiles C.A. statistics and estimates conflict metrics
Flight Path Conflict Analysis
Analytical model to find conflict times and acft. conflict geometries

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 $--90^0 \le \alpha \le 90^0$ (being -90^0 at the south pole and 90^0 at the north pole) and where $0^0 \le \beta \le 360^0$, 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

$$(\alpha, \beta)^0 \to R(\cos\alpha,\ \cos\beta,\ \cos\alpha\sin\beta,\ \sin\alpha) \tag{4.1}$$

Remark 1. If an aircraft is located at $(\alpha, \beta)^0$ 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.,

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

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

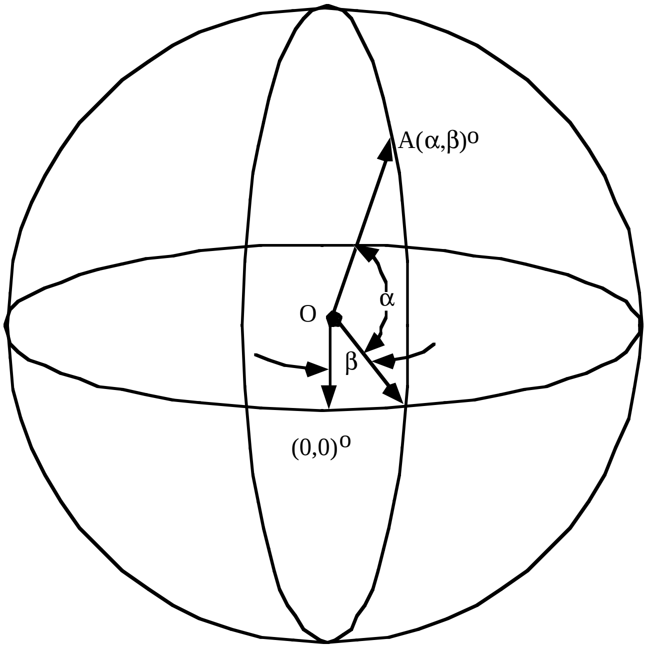


Figure 4.2 Definition of Latitudes and Longitudes.

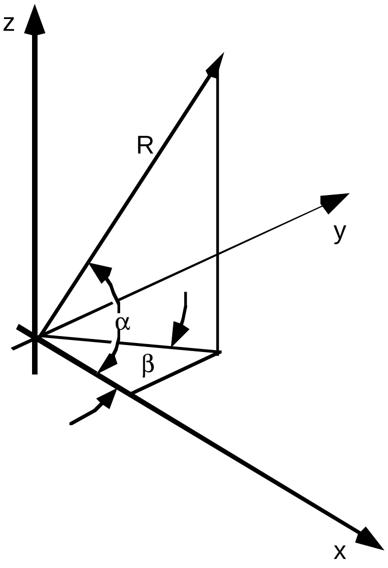


Figure 4.3 Polar To Cartesian Coordinate Transformation.

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

$$D^2 = 2R^2(1 - \cos\theta) \tag{4.3}$$

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

$$x^A = \bar{x}^A + \lambda d^A \text{ for } 0 \le \lambda \le 1 \tag{4.4a}$$

and

$$x^B = \bar{x}^B + \lambda d^B \text{ for } 0 \le \lambda \le 1 \tag{4.4b}$$

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 aircraft A and B satisfies

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

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

$$dA, B) \equiv \left\|x^A - x^B\right\| \leq r_A + r_B \tag{4.6}$$

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

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

and

$$\left\|X^A\right\| - \left\|X^B\right\| \leq S_3 \Rightarrow \text{there exists a conflict.} \tag{4.7}$$

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 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, respectively, denote the standard in-trail (along track), lateral (across track), and vertical separation parameters, and ${D_1}^A$, ${D_2}^A$ and ${D_3}^A$ denote the proximity shell based separation requirements in these three respective dimensions for some particular (type of) 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 inner box, this represents an *untenable conflict* that must be resolved. (A *fatal* conflict could be described via another innermost circumscribing box around the aircraft, if necessary.) When an intruding aircraft lies within the proximity shell, there exists a *conflict risk*. (The lattermost type of conflict is later designated as being of severity 1, an untenable conflict is designated as being of severity 2, while a fatal conflict is labeled as being of severity 3). The intensity of any conflict 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 envelope and the proximity 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

$$x = \bar{x}^A + Qy \tag{4.8}$$

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

$$Q = \left[\frac{Q_1}{\|Q_1\|}\frac{Q_2}{\|Q_2\|}\frac{Q_3}{\|Q_3\|}\right]$$

where,

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

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

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, and d^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 noncolinear (or else the aircraft would be moving vertically with respect to the earth's surface).

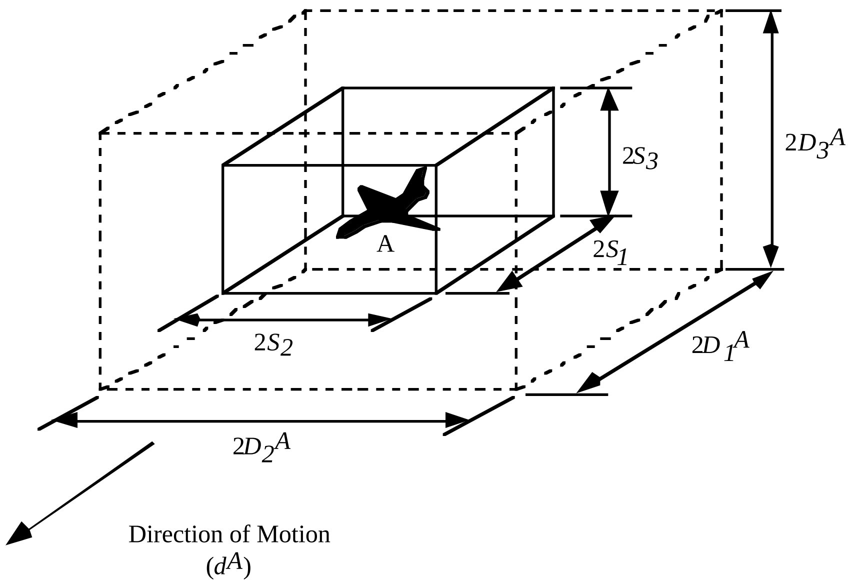


Figure 4.4 Standard Envelope and Proximity Shell for Aircraft A.

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

$$y^B = Q^t(\bar{x}^B - \bar{x}^A) + \lambda d^t d^B \text{, for } 0 \le \lambda \le 1 \tag{4.10}$$

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 standard separation criteria, and $\delta \equiv D$ if we are considering the proximity shell. 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

. $-\delta \le y^B - y^A \le \delta$, i.e.,

$$-\delta \le Q^t(\bar{x}^B - \bar{x}^A) + \lambda Q^t(d^B - d^A) \le \delta \tag{4.11}$$

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

$$0 \le \lambda_1 \le \lambda \le \lambda_2 \le 1 \tag{ 4.12}$$

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

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

where B represents the intruding aircraft, and where the different entities in (4.13) are determined as follows. 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

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

The duration of intrusion over this time segment is given by

$$\tau = T(\lambda_2 - \lambda_1) \tag{4.15}$$

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$A$ and$B$ B over this duration segment. From (4.4), we have $\|x^B - x^A\| = \|\bar{x}^{BA} + \lambda d^{BA}\|$, where ${}^{\lambda} \equiv \bar{x}^B - \bar{x}^A and\ d^{BA} \equiv d^B - .$

Hence, $d_{min} \equiv x^{-\boldsymbol{B}A}$ if $d^{BA} = 0$, and otherwise, we have that $\|x^B - x^A\|^2$ is minimized when

$$\lambda = \bar{\lambda} \equiv \frac{-x^{-BA} \bullet d^{BA}}{\|d^{BA}\|^2} \tag{4.16}$$

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

$$d_{min} = \begin{cases} \bar{x}^{BA} & if \quad d^{BA} = 0 \\ \left\| \bar{x}^{BA} + \lambda' d^{BA} \right\| & otherwise \end{cases} \tag{4.17a}$$

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

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

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

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

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

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

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

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

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

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

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

$$-\frac{1}{2\sqrt{2}} \le \frac{1}{\sqrt{2}} - 2\sqrt{2}\lambda \le \frac{1}{2\sqrt{2}} \text{ or } \frac{1}{8} \le \lambda \le \frac{3}{8}. \tag{4.22}$$

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.

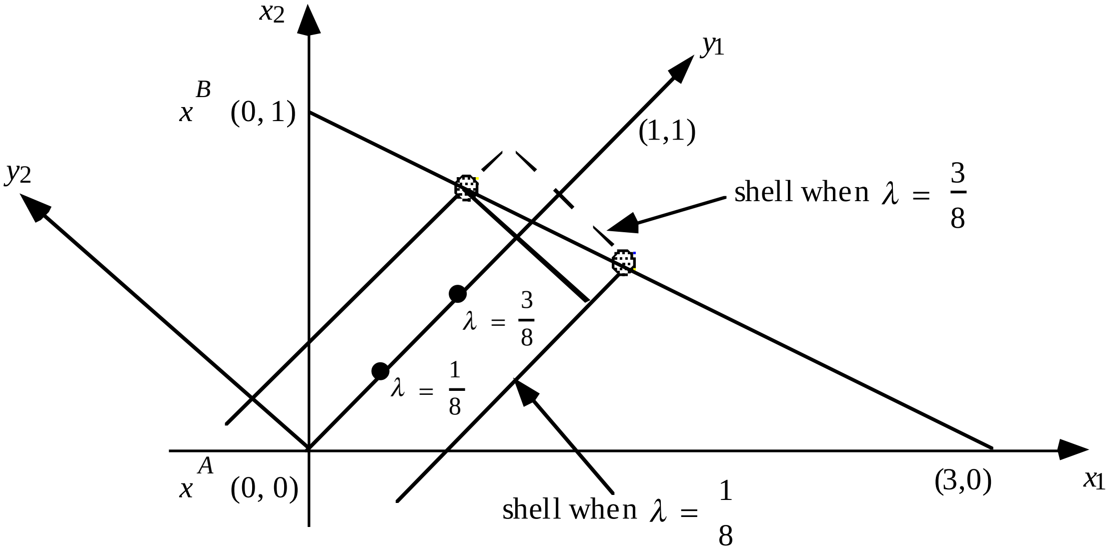


Figure 4.5 Trajectories of Aircraft A and B.

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 $0 \le \lambda \le 1$, 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,

$$-\delta \le Q(\lambda)^t[x^B(\lambda) - x^A(\lambda)] \le \delta \tag{4.23}$$

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:

$\bar{x}^A = (0, 0, \alpha)$ with $x^A(\lambda) = (\lambda, 0.5\lambda^2, \alpha)$, so that $d^A(\lambda) = \frac{d}{d\lambda}x^A(\lambda) = (1, \lambda, 0)$

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

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

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

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

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

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,

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

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:

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

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

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

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

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

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).

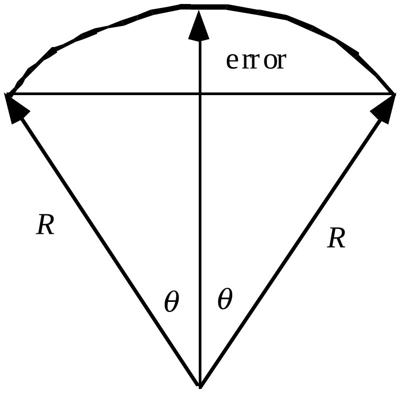


Figure 4.6 Circular Aircraft Trajectory Term Definitions.

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

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

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

.

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

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 compartamentalize workload as a series of time-space counts.

The metric for this study instead gives 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 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 conflicts 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, and L_k, is the sum of the durations for which a conflict of severity (at most) k occurs normalized byN_k . That is,

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

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 $a = i$ 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 $\dot{r}_{k(i,j)}(0) < 0$. To avoid canceling the effect of other $\dot{r}_{k(i,j)}(0)$ values, 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. 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

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

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 preprocessing steps.

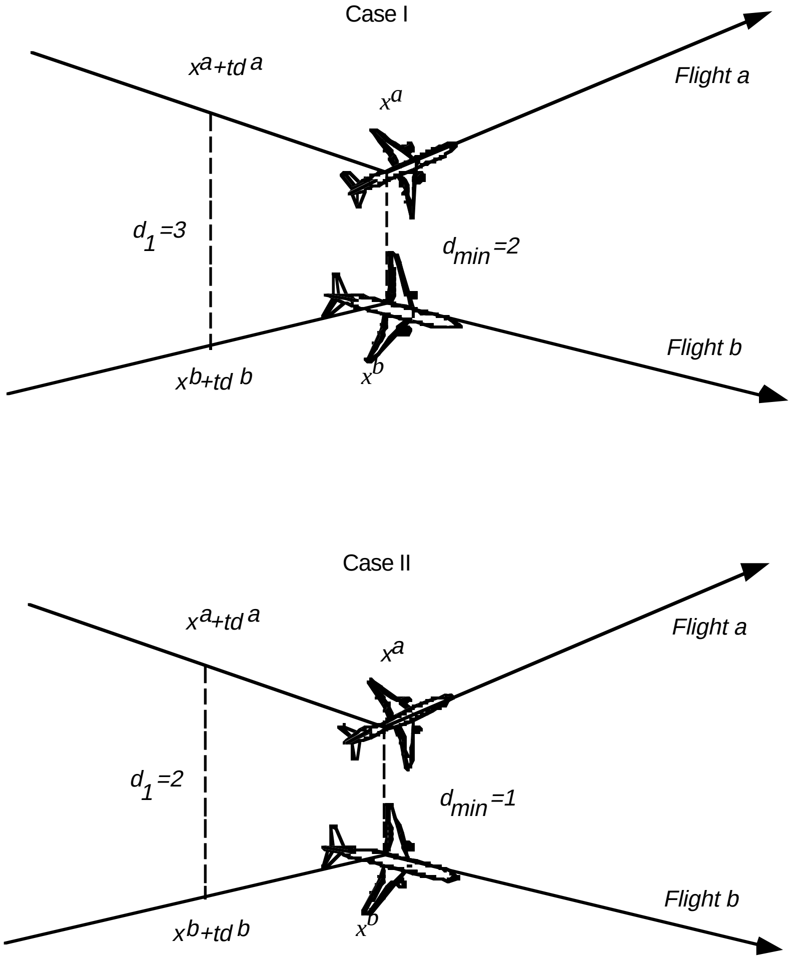


Figure 4.7 Example to Illustrate the Effect of Using Percent Rates.

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.

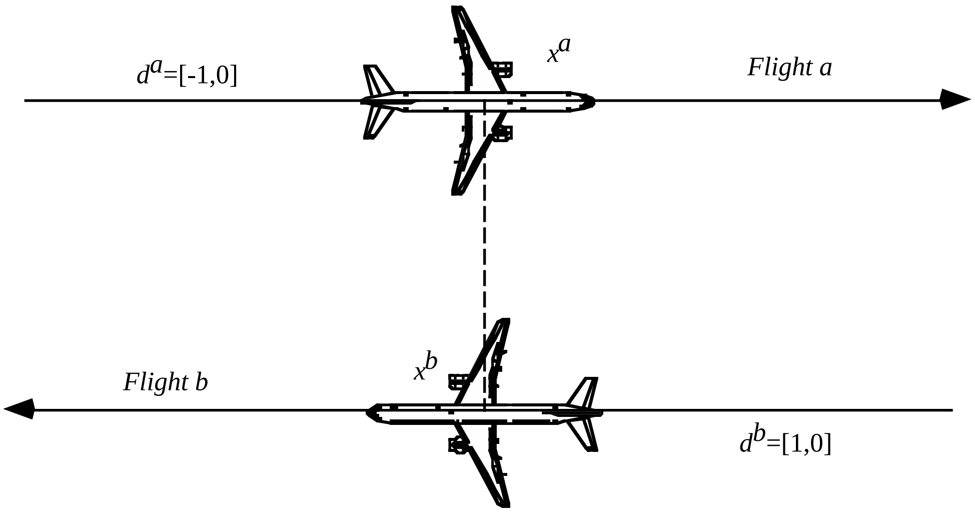


Figure 4.8 Example of Parallel Flights with Opposite Headings.

For each flight$i$ 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 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 $(a^s(i) < d^r(j))$, for each sector r equal to or neighboring sector s, 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 $s) = \{s \cup$ neighboring sector of
            For each sector $r$ in $N(s)$
                For each flight $j$ occupying $r$
                    If $a^r(j) \geq d^s(i)$ and $d^r(j) \leq a^s(i)$
                        $[max\{d^s(i), d^r(j)\}, min\{a^s(i), a^r($
                        Add $[i, j, C]$ to CA
                    end if
                 end for
            end for
        end for
end for
```

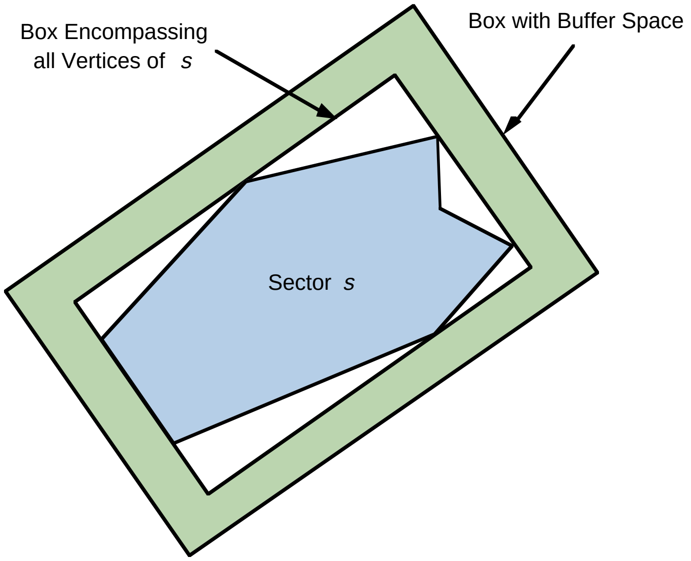


Figure 4.9 Illustration of 2-D Rectangle Created for Neighboring Sector Analysis.

Following this preprocessing, the list CA is passed to the conflict analysis routine developed in 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 $I^S(i)$ 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 computational 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.

CHAPTER 5

Airspace Planning Optimization Model

This chapter describes an optimization model to minimize the cost of detours around Special Use Airspace (SUA). The model assumes that airspace users will have identified various feasible alternative flight plans for each flight between an origin and a destination, thereby implying a colaborative decision making process between service providers (FAA) and users in real-time. In this process, the model considers delays, costs and airline equity issues, as governed the closure of a general shape airspace region due to RLV activity.

5.1 Problem Formulation

Consider a planning horizon H, and suppose that we are given a set of flights $i = \{1, \ldots, m\}$ covering this horizon that are relevant to interactions with a particular RLV spaceport. Let $M = \{1\ldots m\}$. Note that there might be other extraneous flights that are unaffected by this spaceport operation, but that interact between certain fixes with flights that are affected by such operations. We will assume that flight plans are known for such extraneous flights and are not part of the present decision process. (However, such flights do play a role in ascertaining workload and conflicts below.)

For each flight, $i \in M$, let

P_i = {set of possible flight plans p composed of departure times and flight levels between designated fixes along the route from the corresponding origin to the corresponding destination}.

Note that there will typically exist some preferred departure time for each flight, along with some alternative (discrete) departure schedules. For each departure time, a flight plan can be generated using some commercial package, such as JEPPESSEN Flite Star, for example. This plan would be naturally dependent on the spaceport operations during the journey of the flight. Possibly, more than one flight plan could be considered as an alternative for a given departure time. (We can also develop a new flight plan generator based on a time-space network representation of the airspace.)

Now, given any combination (i, p), $i \in M$, $p \in P_i$, we can compute a cost factor c_{ip} for adopting plan p for flight i . (AOM or RAMS, for example, can be used for this purpose.) This cost would reflect fuel expended, delay costs, as well as penalties or benefits (rewards or negative penalties) based on safety considerations and on the selection of a corresponding departure time.

Accordingly, defining the decision variables

$$x_{ip} = \begin{cases} 1 \text{ if plan } p \in P_i \text{ is adopted for flight } i \in M \\ 0 \text{ otherwise.} \end{cases} \quad \forall\ i \in M, p \in P, \tag{5.1}$$

We can formulate a total system-based objective function to

$$\text{Minimize} \sum_{i \in M} \sum_{p \in P_i} c_{ip} x_{ip}. \tag{5.2}$$

The constraints would include the selection of a plan for each flight as specified by

$$\sum_{p \in P_i} x_{ip} = 1 \ \forall \ i \in M, \tag{5.3}$$

as well as certain equity, workload, and conflict resolution restrictions as discussed next.

5.2 Equity Constraints

Suppose that there are some F airline firms involved in this study, indexed by $f = 1, \ldots, F$. In the process of selecting flight plans based on (5.2) and (5.3) (in addition to workload and conflict resolution constraints as described in the sequel), we would also like to achieve a degree of equity among the airline firms. For each firm $f = 1, \ldots, F$, let us define a measure of ineffectiveness as

$$M_f = \sum_{(i,\, p) \in A_f} \alpha_{ip} x_{ip} \tag{5.4a}$$

where

$$A_f \subseteq \bar{A}_f \equiv \{(i, p)\text{: flight } i \text{ belongs to firm } f\},\ f = 1, \ldots, F\ . \tag{5.4b}$$

Defining the variables x_l^e and x_u^e to represent the (variable) range for the ineffectiveness measures M_f, $f = 1, \ldots, F$, where the upper limit x_u^e of this range is restricted to be no more than some specified value , we can model equity via the following mechanism.

Include within constraints:

$$x_l^e \leq \sum_{(i,\,p) \in A_f} \alpha_{ip} x_{ip} \leq x_u^e \qquad \forall f = 1, \ldots, F \tag{5.5}$$

$$x_l^e \geq 0,\, x_u^e \leq n_e\,. \tag{5.6}$$

Include within the objective function:

$$\text{(Minimize)}\ \ldots + \mu_e(x_u^e - x_l^e) + \mu_u^e x_u^e \tag{5.7}$$

where μ_e is a (commensurate) penalty per unit of variation in the measures M_f ,$f = 1, \ldots, F$, and μ_u^e is a (commensurate) penalty for the maximum incurred measure of ineffectiveness. Note that if restricting $x_u^e \leq n_e$ is sufficient, we could take $\mu_u^e \equiv 0$. On the other hand, in order not to overly restrict the problem, n_e could be taken as the maximum tolerable limit on any M_f value, and then the penalty μ_u^e would serve to reduce x_u^e below n_e to the extent possible or desirable.

Remark 1. The variables x_l^e and x_u^e can be fixed at their respective bounds of 0 and n_e if so desired. In this case, each measure M_f is simply restricted to be no greater than n_e , for $f = 1, \ldots, F$, and the constraints $M_f \geq x_l^e \equiv 0$ in (5.5) may be omitted.

Special Cases. The following are some special cases of the equity modeling constraints (5.5)-(5.7). In each case, x_l^e and x_u^e can be treated as variables as in (5.5) and (5.6), or be fixed as mentioned above in Remark 1.

Case(i): $A_f = U_f$, where

$$U_f = \{(i, p) : \text{flight plan } (i, p) \text{ belongs to firm } f \text{ and is undesirable}\} \; \forall f = 1, \ldots, F, \quad (5.8)$$

and $\alpha_{ip} \equiv 1 \;\; \forall (i, p)$. This restricts the number of undesirable flight plans selected for each firm to be no more than n_e, and strikes a balance between the firms.

Case(ii) : $A_f = \bar{A}_f \;\; \forall f = 1, \ldots, F$, and α_{ip} = delay (d_{ip}, say) for flight plan (i, p). This case seeks an equity with respect to total delay. Note that in lieu of delay, any other "cost" measure could be used in this context.

Case(iii): Since the number of flights, n_f say, that belong to the various firms $f = 1, \ldots, F$ might differ quite widely, it is more appropriate to seek equity with respect to the average delay (or cost), as opposed to the total delay considered in Case (ii) above. Accordingly, we can use $A_f = \bar{A}_f$ $\forall f = 1, \ldots, F$, and

$$\alpha_{ip} = \frac{d_{ip}}{n_f} \text{ for each } (i, p) \in A_f, \; f = 1, \ldots, F. \qquad (5.9)$$

Case(iv): Similar to the normalization of Case (ii) via Case (iii), we can normalize Case (i) by considering equity among the fraction of flights selected for each firm that are undesirable. Accordingly, we can set

$$A_f = U_f \quad \forall f = 1, \ldots, F, \text{ and } \alpha_{ip} = \frac{1}{n_f} \text{ for each } (i, p) \in U_f, \quad f = 1, \ldots, F. \tag{5.10}$$

Case(v): The foregoing discussion pertains to "minisum" measures of ineffectiveness. Alternatively, we can consider a "minimax" strategy that attempts to minimize the maximum delay (or cost) d_{ip} incurred by any selected flight plan. Hence, we would fix $\mu_e = 0$ and $x_l^e = 0$ in this case, and replace (5.5) and (5.6) by

$$x_u^e \geq d_{ip}x_{ip} \quad \forall (i, p) \in \bar{A}_f, f = 1, \ldots, F. \tag{5.11}$$

Note that in this case, the overall objective function is reduced to a combination of a minisum and minimax objective.

Of these alternatives, we recommend the use of Case (iii) or Case (iv), embodied by Equations (5.9) or (5.10), respectively, along with Equations (5.5)-(5.7).

5.3 Workload Constraints

Consider the total collection of flight plans $\bigcup_{i \in M} P_i$. Jointly, these plans involve traversals between certain pairs of fixes, as well as free-flight cruises between designated pairs of fixes, at various specified levels. Let us consider a segmentation of the airspace into sectors as defined by FAA (generally nonconvex polygons, lifted into the third dimension). Define the workload for a sector at any point in time to be the number of aircraft in that sector at the given instant of time. Let

$$S = \{\text{set of all sectors involved with the collection of flight plans } \bigcup_{i \in M} P_i\}. \tag{5.12}$$

For each sector $s \in S$, we can now examine the occupancy durations of the flights $i \in M$ over the horizon H, in concert with the occupancy durations of any extraneous flights as described above. The

AOM model described in Chapter 3 provides this information. Note that whenever we have an overlap of such occupancy durations, we would have a potential increase in workload. In practice, Air Traffic Control (ATC) operators routinely handle several aircraft in their sectors successfully. Of course, when the workload becomes too high, a potentially dangerous or untenable situation can arise. Hence, let us define the following entities.

For each sector $s \in S$, let $k = 1, \ldots, K_s$ be a total collection of maximal overlapping sets of flight-plans (i, p), where an overlapping set of flight-plans is called maximal if it is not a strict subset of another overlapping set. For example, examining Figure 5.1, we have $\mathrm{K_s} = 4$ such maximal sets given by $\{(i_1, p_1), (i_2, p_2), (i_3, p_3)\}$,$\{(i_2, p_2), (i_3, p_3), (i_4, p_4)\}$,$\{(i_3, p_3), (i_4, p_4), (i_6, p_6)\}$,and $\{(i_5, p_5), (i_6, p_6)\}$. Let us denote these sets by C_{sk} for $k = 1, \ldots, K_s$. Hence,

$C_{sk} = \{(i, p)$: flight plan (i, p) belongs to the k^{th} maximal overlapping set for sector $s\}$,

$\forall\ k = 1, \ldots, K_s, s \in S$.(5.13)

An efficient algorithm for determining these sets is described in Sherali and Brown (1994). Note that it is possible that if (i_1, p_1) and $(i_2, p_2) \in C_{sk}$, then $i_1 = i_2$, i.e., this pair corresponds to the same flight, although in this case, the plans would be distinct. We would now like to impose that there be no more than some n_{sk} resident simultaneous aircraft in a sector from among the ones that appear in C_{sk} , i.e.,

$$\sum_{(i, p) \in C_{sk}} x_{ip} \le n_{sk} \qquad \forall\ k = 1, \ldots, K_s \text{ and } s \in S. \tag{5.14}$$

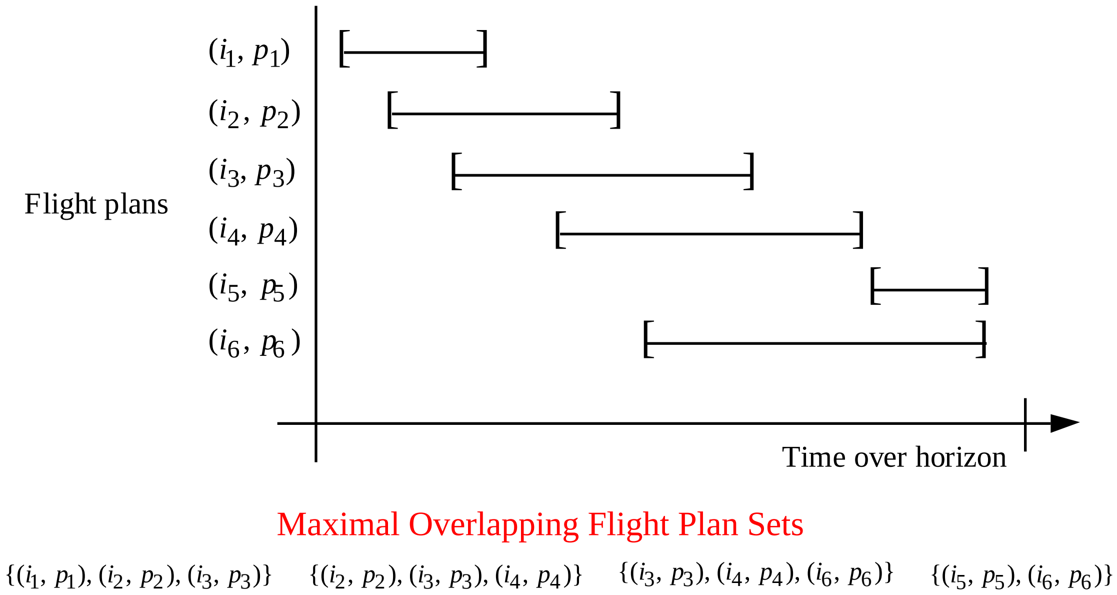


Figure 5.1 Gantt Chart for Formulating Airspace Workload (or Congestion Penalty) Constraints.

The parameters n_{sk} can be chosen by the user to be dependent on the particular sector and the nature (type and number) of the overlapping flights. Also, note that by virtue of (5.3), a given flight i would at most contribute a unit to the left-hand side of (5.14).

Remark 2. The reason for selecting maximal sets is to obtain a minimal nonredundant set of constraints in (5.14). For any overlapping set that is not explicitly represented in (5.14), this set must be a subset of some set that appears in (5.14). If the restriction on the permissible number of aircraft for this former set is at least as large as that for the latter (for some such case), then a constraint of type (5.14) based on this former set is redundant. Otherwise, we would need to explicitly include such a constraint within (5.14). Henceforth, we will assume that (5.14) includes all possible nonredundant workload constraints of this type.

A preliminary model based on the development thus far is stated below.

$$\text{Minimize} \sum_{i \in M} \sum_{p \in P_i} c_{ip} x_{ip} + \mu_e (x_u^e - x_l^e) + \mu_u^e x_u^e \tag{5.15a}$$

$$\text{subject to} \sum_{p \in P_i} x_{ip} = 1 \quad \forall i \in M \tag{5.15b}$$

$$\sum_{(i,\, p) \in C_{sk}} x_{ip} \le n_{sk} \qquad \forall\ k = 1, \ldots, K_s \text{ and } s \in S \tag{5.15c}$$

$$x_l^e \le \sum_{(i,\, p) \in A_f} \alpha_{ip} x_{ip} \le x_u^e \quad \forall f = 1, \ldots, F \tag{5.15d}$$

$$x \text{ binary}, \ \ x_l^e \ge 0, x_u^e \le n_e \ . \tag{5.15e}$$

Note that (5.15) is a mized-integer 0-1 programming problem that possesses special partitioning and generalized packing constraints described by (5.15b) and (5.15c), respectively. This structure can be exploited in devising special algorithmic schemes. (Alternatively, automatic reformulation techniques can be used to enhance the model itself before solving it via a commercial package such as CPLEX-MIP.)

Remark 3. In the foregoing model, we could let the maximum number of overlapping flights permitted within each sector $s \in S$ be a variable n_s bounded by the interval $[0, \bar{n}_s]$, say, and we could accordingly penalize its value in the objective function. If a linear penalty term is used, this would simply involve replacing the right-hand side in (15c) by n_s, where $0 \le n_s \le \bar{n}_s \ \forall \ s \in S$, and incorporating an appropriate linear cost term in the objective function. However, it might be more suitable to impose a penalty factor that increases nonlinearly in an appropriate fashion with an increase in workload. That is, if the maximum number of aircraft in a sector increases from one to three, the associated penalty should likely more than triple. Hence, let us define the binary variables

$$y_{sn} = \begin{cases} 1 \text{ if the maximum workload in sector } s \text{ is } n \\ 0 \text{ otherwise} \end{cases} \qquad \forall s \in S, n = 1, \ldots, \bar{n}_s$$

and let μ_{sn} be the associated penalty for having $y_{sn} = 1$. Then, Model (5.15) would be modified as follows.

$$\text{Minimize } \sum_{i \in M} \sum_{p \in P_i} c_{ip} x_{ip} + \sum_{s \in S} \sum_{n=1}^{\bar{n}_s} \mu_{sn} y_{sn} + \mu_e (x_u^e - x_l^e) + \mu_u{}^e x_u{}^e \qquad (5.15\text{'a})$$

$$\text{subject to } \sum_{p \in P_i} x_{ip} = 1 \quad \forall i \in M \qquad (5.15\text{'b})$$

$$\sum_{(i,\, p) \in C_{sk}} x_{ip} - n_s \le 0 \quad \forall k = 1, \ldots, K_s, s \in S \qquad (5.15\text{'c})$$

$$x_l^e \le \sum_{(ip) \in A_f} \alpha_{ip} x_{ip} \le x_u^e \quad \forall f = 1, \ldots, F \qquad (5.15\text{'d})$$

$$n_s = \sum_{n=1}^{\bar{n}_s} n y_{sn} \forall s \in S \qquad (5.15\text{'e})$$

$$\sum_{n=1}^{\bar{n}_s} y_{sn} \le 1 \forall s \in S \qquad (5.15\text{'f})$$

$$(x, y) \text{ binary, } x_l^e \ge 0, x_u^e \le n_e \ . \qquad (5.15\text{'g})$$

Note that n_s can be treated as a continuous variable in (5.15'), and its bounding and integrality restrictions are implied by (5.15'e-g).

Remark 4. Observe that for each flight plan combination (i, p), we can examine the number of times x_{ip} appears in the constraint set (5.15c) in order to assess the degree of workload being generated by this flight-plan. This indicator could be used to prompt the generation of alternative plans for a given flight, based on the degrees of workload associated with its current set P_i .

5.4 Aircraft Conflict Constraints.

For each sector, let us discretize the horizon into time slots whose durations depend on the amount of traffic normally present in the sector and its conflict resolution capability. For example, since air traffic is dense in New York, and an ATC controller handling a New York enroute sector is likely to be more exposed to relatively larger workloads, the duration of the time slots in the corresponding sectors would be relatively smaller. The conflict constraints developed below impose the restriction that for each sector, the maximum number of (permissible) conflicts that need to be resolved for each time period should not exceed 1. The practical implications of this assumption will be studied in Chapter 6 with a realistic example.

In order to develop these conflict and workload constraints, we must first be able to compare each flight trajectory against every other flight path, and record vital information regarding any potential conflicts. Furthermore, we must be able to determine the sector occupancy of each flight trajectory, and consequently, determine the sectors over which the models AOM and AEM described in Chapters 3 and 4, respectively, provide this information.

If any detected conflict is fatal (as defined in Chapter 4) we would immediately impose a constraint that permits the selection of at most one such flight plan. Denoting FC as the set of such "fatally conflicting" pairs of flight plans P and Q, we begin by stipulating that

$$x_P + x_Q \leq 1 \text{ for all } (P, Q) \in FC. \tag{5.16}$$

Other conflicts must be resolved by the ATC in the particular sector in which they occur. The identification of such sectors can be made by examining the time at which these conflicts occur and locating the corresponding sectors via the sector occupancy list created in the sector occupancy routine for each aircraft.

Remark 5. Note that a particular flight plan that traverses through some sector s might be in conflict by the foregoing definition with another flight plan that occupies a different sector s' . Since this situation adds to the potential workload of both sectors s and s' , we include within the Gantt chart for each of these sectors the flight plan that belongs to the other sector for the duration over which this conflict persists. Hence, the workload constraints of the previous subsection accommodate such extraneous occupancy intervals as well. Observe that another alternative might be to extend the boundaries of each sector by an appropriate amount so that the sectors provide an overlapping coverage of the national airspace, and then to formulate the workload constraints in the usual fashion as described before. However, for now, we will adopt the former strategy because of the standard data files used by the FAA to record sector designations.

To formulate the conflict constraints, suppose that we construct a graph $G_t(N_t, A_t)$ for each given time period t, whereN_t is the set of nodes that represent all the flight plans (i, p), and A_t is the set of edges such that if flight plans P and Q are in conflict during this period t, then A_t includes an edge joining these corresponding nodes. Since we have explicitly excluded non-permissible conflicts via (5.16) above, we can restrict our attention to recording via A_t just the permissible conflicts, that is, conflicts that can be resolved by some defined measure. This graph would typically be a collection of (disjoint) components. For each sector s, let $G_{st}(N_{st}, A_{st})$ be a subgraph of G_t that is comprised of those components of G_t for which at least one of the nodes in this component is in sector s during time t. Alternatively, G_{st} can be defined as follows. Let N_{st} contain all the nodes that belong to sector s at time t, along with any adjacent nodes from the graph G_t . Then define A_{st} as the set of arcs from A_t that have both the end points included in N_{st}, i.e., G_{st} is the subgraph induced by N_{st} . The defined construction of G_{st} would need to compromise between effort versus representation, but the general concept behind G_{st} is to obtain a graph that represents conflicts between pairs of flight-plans that sector s needs to participate in resolving during time period t. Also, if for successive periods, the conflict graph remains unchanged, then these periods can be combined in the present constraint generation phase. We now impose the constraint that:

"No more than one permissible conflict should occur for each sector during each time period".

To model this constraint, for each sector, consider the edges in A_{st} taken two at a time, and for each pair k, let S_k be the set of nodes at which this pair of arcs is incident. $|S_k|$ equals three or four, depending on whether the pair of edges is adjacent or not. The imposed constraint would then be

$$\sum_{P \in S_k} x_P \le |S_k| - 1 \ . \tag{5.17}$$

Note that there would be $|A_{st}|(|A_{st}| - 1)/2$ inequalities of the type (5.17) for each sector s, for each time period t. (We assume that the index k runs contiguously over these constraints for all s, t.) Observe that there will likely be several redundant constraints established via this process. In particular, the following result holds true.

Proposition 1. Consider a pair of constraints of the type (5.17) for some sets S_1 and S_2, say, such that $S_1 \subseteq S_2$. Then (5.17) for S_2 is redundant (even in the continuous sense) and can therefore be deleted.

Proof. Let us show that (5.17) for S_1 implies that for S_2. Given that (5.17) holds for S_1, we have that

$$\sum_{P \in S_2} x_P = \sum_{P \in S_1} x_P + \sum_{P \in S_2 - S_1} x_P \le |S_1| - 1 + |S_2 - S_1| = |S_2| - 1 \ .$$

This completes the proof.

For example, if we had a conflict graph comprised of nodes 1,2,3, and 4 with edges (1,2), (2,3), (1,3), and (3,4), then the pair of edges (1,2) and (1,3) impose the constraint that

$$x_1 + x_2 + x_3 \le 2$$

while the edges (1,2) and (3,4) impose that

$$x_1 + x_2 + x_3 + x_4 \le 3$$

which is implied by the former. Procedure 1 below presents a scheme for directly generating only non-redundant elements of (5.17).

Overview of Procedure 1. First, constraints (5.17) are generated for all adjacent pairs of edges. These constraints may be duplicates if the set of nodes involved forms a clique. The procedure recognizes this structure to avoid generating a copy of a pre-existing constraint. For all remaining pairs of edges, a constraint is generated only if it is not already implied by the constraints generated from the adjacent pairs of edges.

Details for Procedure 1. Define an upper triangular node adjacency matrix E having elements $E(P, Q) = 1$ if nodes P and Q have a connecting edge, and 0 otherwise for each $P < Q$. We will let $E(P - Q)$ denote $E(P, Q)$ if $P < Q$, $E(Q, P)$ otherwise.

Step 1: Generate (5.17) for adjacent edges.

```
for each row $P$,
    for each entry $E(P, Q) = 1$, $Q > P$
        Generate (5.17) corresponding to edges $(P, Q)$ and $(P, K)$, where $E(P, K) = 1$ and $Q < K$.
            Generate (5.17) corresponding to edges $(P, Q)$ and $(H, Q)$, where $E(H, Q) = 1$,
              $P < H < Q$, and $E(P, H) \neq 1$.
            Generate (5.17) corresponding to edges $(P, Q)$ and $Q, M$, where $(Q, M) = $ ,
                $P < Q < M$, and $E(P, M) \neq 1$.
    end
end
```

Step 2: Generate (5.17) for non-adjacent edges .

```
for each row $P$,
    for each entry $E(P, Q) = 1$, $Q > P$
        for each row $M > P$, $M \neq Q$, $E(P, M) \neq 1$,
            for each column $N > M$, such that $E(M, N) = 1$, $N \neq Q$, $E(P, N) \neq 1$
                if $E(Q - M) \neq 1$ and $E(Q - N) \neq 1$
                    Generate (5.17) corresponding to edges $(P, Q)$ and $(M, N)$
                end
            end
        end
    end
end
```

Remark 6. Note that depending on the given durations of each time slot for a sector relative to the horizon, a discrete event simulation process that advances aircraft one interval at a time in order to check for potential pairwise conflicts could be prohibitive. Instead, we can examine straight line paths between designated way-points (or breakpoints) in a pairwise fashion and identify conflicts when they occur, placing each identified conflict in the appropriate graph G_{st} . Also, a preprocessing can be conducted to a priori identify possibly conflicting situations for which a more detailed conflict analysis needs to be performed. This is essentially the procedure adopted by the model AEM of Chapter 4.

An airspace planning model, AP1, that incorporates the workload and the conflict constraints, along with suitable costs in the objective function, can now be constructed as follows.

$$\text{AP1: Minimize} \quad \sum_{i \in M} \sum_{p \in P_i} c_{ip} x_{ip} + \sum_{s \in S} \sum_{n=1}^{\bar{n}_s} \mu_{sn} y_{sn} + \mu_e (x_u^e - x_l^e) + \mu_u^e x_u^e \qquad (5.18a)$$

$$\text{subject to } \sum_{p \in P_i} x_{ip} = 1 \ \ \forall \ \mathrm{i} \in M \tag{5.18b}$$

$$\sum_{(i,\, p) \in C_{sk}} x_{ip} - n_s \le 0 \quad \forall \ k = 1, \ldots, K_s, s \in S \tag{5.18c}$$

$$n_s = \sum_{n=1}^{\bar{n}_s} n y_{sn} \quad \forall s \in S \tag{5.18d}$$

$$\sum_{n=1}^{\bar{n}_s} y_{sn} \le 1 \quad \forall s \in S \tag{5.18e}$$

$$x_l^e \le \sum_{(i,\, p) \in A_f} \alpha_{ip} x_{ip} \le x_u^e \qquad \forall \ f = 1, \ldots, F \tag{5.18f}$$

$$x_P + x_Q \le 1 \ \text{ for all } \ (P, Q) \in FC \tag{5.18g}$$

$$\sum_{p \in S_k} x_P \le |S_k| - 1 \ \text{ for each arc pair } \ k \in A_{st}, \forall t \in H, s \in S \tag{5.18h}$$

$$(x, y) \ \text{binary}, \ x_l^e \ge 0, x_u^e \le n_e \ . \tag{5.18i}$$

Remark 7. We can alternatively model the conflict constraints by defining a variable z_{PQ} for each edge (P, Q) in the conflict graph, $P < Q$, which takes on a value of 1 if this conflict is permitted and 0 otherwise. Then, we would have a single conflict constraint for each sector s in period t that requires the sum of z_{PQ} over (P, Q) in A_{st} to be no more than 1. These z variables would then need to be related to the x-variables via the following constraints:

$$z_{PQ} \le x_P \,,\ z_{PQ} \le x_Q \,,\ z_{PQ} \ge 0 \,,\ z_{PQ} \ge x_P + x_Q - 1 \,.$$

Note that this in effect would create a linearized version of essentially a quadratic model based on $z_{PQ} = x_P \bullet x_Q$, but it would permit the penalizing of different types of conflicts differently in the objective function. However, its LP relaxation will likely be weaker, although reformulation techniques could be used to strengthen it. A formulation for this alternative model AP2 is given below.

AP2: Minimize $$\sum_{i \in M} \sum_{p \in P_i} c_{ip} x_{ip} + \sum_{s \in S} \sum_{n = 1}^{\overline{n}_s} \mu_{sn} y_{sn} + \mu_e (x_u^e - x_l^e) + \mu_u^e x_u^e \tag{5.19a}$$

subject to $$\sum_{p \varepsilon P_i} x_{ip} = 1 \quad \forall \ \mathrm{i} \in M \tag{5.19b}$$

$$\sum_{(i,\, p) \in C_{sk}} x_{ip} - n_s \le 0 \ \forall \ k = 1, \ldots, K_s, s \in S \tag{5.19c}$$

$$n_s = \sum_{n = 1}^{\bar{n}_s} n y_{sn} \qquad \forall s \in S \tag{5.19d}$$

$$\sum_{n = l}^{\bar{n}_s} y_{sn} \le 1 \qquad \forall s \in S \tag{5.19e}$$

$$x_l^e \le \sum_{(i,\, p) \in A_f} \alpha_{ip} x_{ip} \le x_u^e \ \forall f = 1, \ldots, F \tag{5.19f)}$$

$$x_p + x_Q \le 1 \text{ for all } (P, Q) \in FC \tag{5.19g}$$

$$\sum_{(P,Q) \in A_{st}} z_{PQ} \le 1 \qquad \forall t \in H, \forall s \in S \tag{5.19h}$$

$$z_{PQ} \le x_P, z_{PQ} \le x_Q, z_{PQ} \ge 0, z_{PQ} \ge x_P + x_Q - 1 \tag{5.19i}$$
$$\forall (P, Q) \in A_{st}, \forall t \in H, x \in S$$

$$(x, y) \text{ binary, } x_l^e \ge 0, x_u^e \le n_e \text{ .} \tag{5.19j}$$

Remark 8. For sectors that are unable to handle even one conflict over the defined duration of a single time slot for that sector, we can either directly expand the duration of the time slot for that sector, or examine the union of the graphs over more than one time slot, as necessary, and impose conflict constraints with respect to the resulting graph that represents conflicts over the expanded duration. These methods can accommodate the capabilities of different sectors differently, if necessary.

5.5 Implementation Issues

The principal value of this model would arise in providing insights into the problem situation via various what-if scenario investigations. For example, the following types of investigations can be considered.

(a) Alternative restrictions on the cordoning of airspace around the RLV spaceport during launches could be evaluated with this model. Different airspace restrictions would yield different values of cost coefficients in (5.15) based on fuel and delay computations. In addition, one might develop certain measures of safety, and incorporate appropriate penalties in the objective cost coefficients to reflect the relative safety of trajectories with respect to RLV operations. This can be particularly accomplished when treating n_{sk} as variables n_s as in (5.15’).

(b) The effect of various ATC policies can be evaluated with respect to their influence on the parameters n_{sk} in constraints (5.15c). ATC workload restrictions (and/or costs) can also be reflected via the parameters n_{sk} along with their associated costs (when treated as variables in (5.15')).

(c) The effect of alternative flight plans can also be evaluated using this model. In fact, this model can itself serve to evaluate the efficacy of various flight plan generation programs. Also, this model can be used in conjunction with SIMMOD or RAMS, which are large-scale simulation models for analyzing airspace operations related to a given set of flight plans. Hence, RAMS can be used to provide a more detailed evaluation of a solution prescribed by our model.

(d) Similar to (b), different regulations imposed by FAA might yield different interpretations on what poses a "conflict." These policies could be evaluated by translating them into appropriate constraints of the type (5.15c) and examining their effect on the model solution.

In summary, this model can be utilized in one of two ways.

(1) Generator of a suitable mix of flight plans for a set of flights operating in the vicinity of a spaceport: In this role, the model can be coordinated with RAMS by using the latter simulation package to evaluate in more detail the airport operations related to the prescribed solution suggested by the model.

(2) Policy Evaluator: Various what-if scenarios can be evaluated by policy/decision makers in determining operational guidelines.

Hence, the model can be used, both, in a tactical decision-making mode, as well as for generating strategic plans to detour flights around SUA regions.

Figure 5.2 demonstrates the implementation sequence of the Airspace Planning Model (APM). AOM evaluates sector occupancies over time for the regions surrounding the RLV operational airspace. The output from AOM is fed into AEM to generate aircraft encounter sequences, and to determine measures of workload and congestion for each sector. This result is then translated into constraint equations necessary to solve the Airspace Planning model using a mixed-integer programming packages. CPLEX, a standard mathematical programming solver, is used in this context. The net result of this procedure is an Air Traffic Management policy for selecting a set of flights that are cleared around the RLV induced SUA. A sample scenario and its corresponding solution is illustrated in Chapter 6.

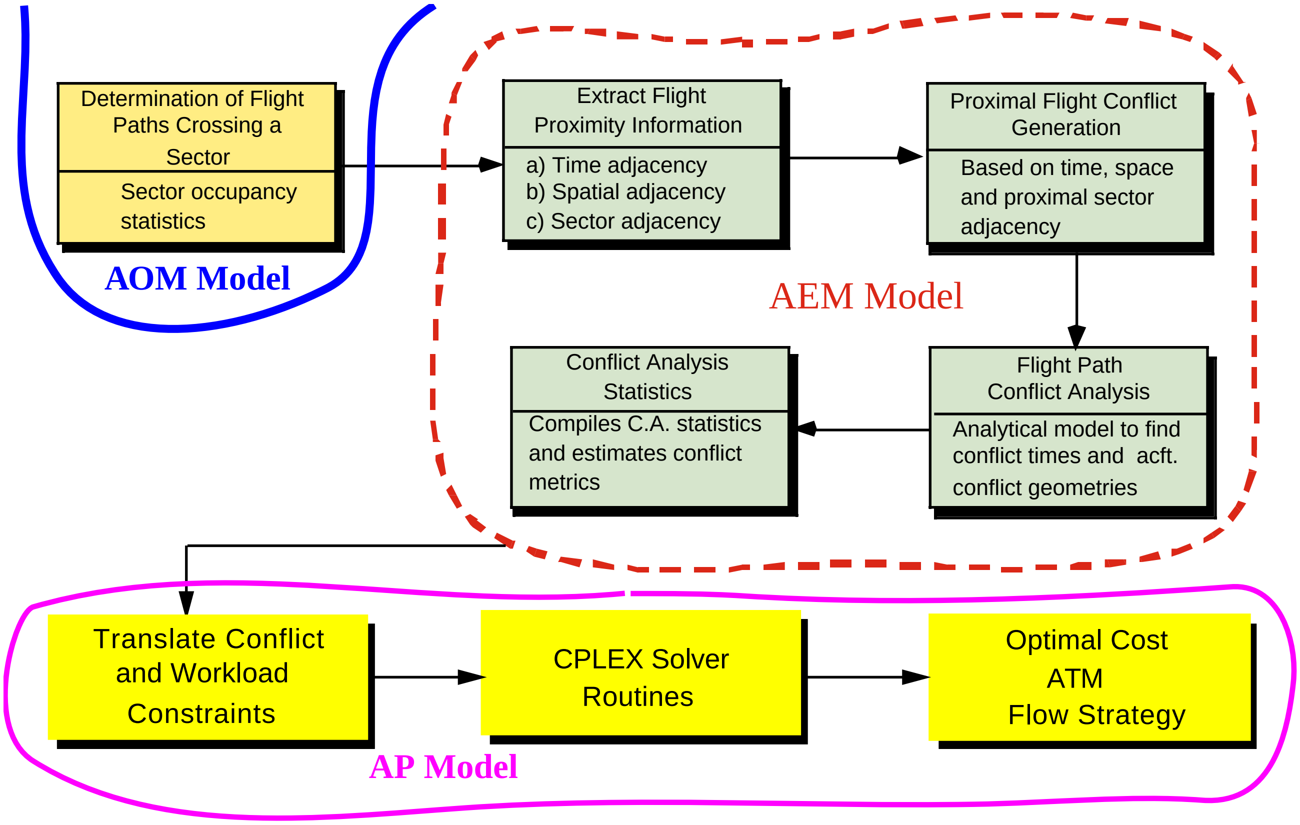

Determination of Flight Paths Crossing a Sector
Sector occupancy statistics
AOM Model
Extract Flight Proximity Information
a) Time adjacency
b) Spatial adjacency
c) Sector adjacency
Proximal Flight Conflict Generation
Based on time, space and proximal sector adjacency
AEM Model
Conflict Analysis Statistics
Compiles C.A. statistics and estimates conflict metrics
Flight Path Conflict Analysis
Analytical model to find conflict times and acft. conflict geometries
Translate Conflict and Workload Constraints
CPLEX Solver Routines
Optimal Cost ATM Flow Strategy
AP Model

Figure 5.2 Implementation Sequence of the Optimization Model.

CHAPTER 6

Model Analysis

The models AOM and AEM are an integral part of the economic analysis of aircraft operations around spaceports. The accuracy of results in these two models is of paramount importance to let FAA decision makers judge important questions such as traffic demands in airspace boundaries over time under current and future Free Flight conditions, possible encroachment of traffic as it detours around SUA regions and even to judge collision risk between RLV and aircraft traffic traversing regions of airspace over a given time horizon.

To test the accuracy of 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 report 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 research team is reviewing these scenarios to assess the economic impact of RLV ob ATM in future years.

The NAS traffic demand scenario database represents typical NAS conditions for five days of the year using 1996 ETMS traffic data as a baseline. Each scenario or operational concept as defined in the NA-

RIM 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 OPGEN[1]. 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 terminal 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 relates to the optimization mode used. Flights longer than 1000 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 study the transition to the concept of Free Flight. The following paragraphs summarize the ATC rules and wind conditions considered for each NAS traffic demand scenario.

6.1 Model Validation

SAR data derived from SDAT was used to validate AOM and AEM. In this validation study, 4320 flights traversing the Miami and Jacksonville Air Route Control Centers on August 18, 1997 (between 15:00 and 24:00 Zulu) were used as a test case for AEM. A subset of this data is shown in Figure 5.1, illustrating all flights arriving and departing the Miami International Airport over 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

[1] OPGEN is a copyrighted by CSSI, Inc.

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.

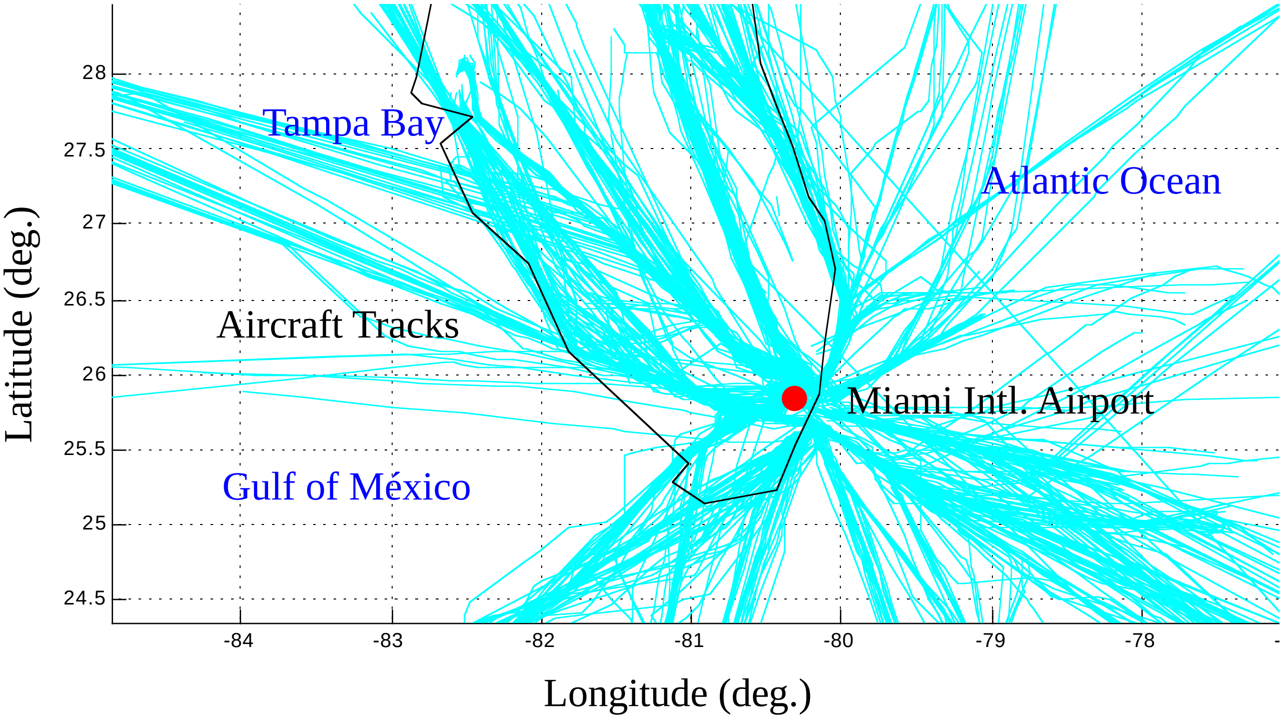


Figure 6.1 Partial ZMA-ZJX Traffic Data used for Model Validation (August 18, 1997).

In the validation of AEM, all 4320 flights were used to test the number of conflicts in the airspace. In this study, the size of the'collision proximity shell' around the aircraft was chosen as a 3D rectangular box with dimensions s_1, s_2, and s_3 equal to 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 FL 180, and 1700 ft. above FL180. These thresholds were selected after multiple runs of AEM revealed that these choices 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 1700 ft., respectively) vertical separation parameters were used in composing the heights of the proximity shell box. Results from this analysis are shown in Table 6.2

6.2 Airspace Scenarios

6.2.1 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).

6.2.2 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
Above FL290 at intervals of 4,000 ft beginning at FL310	Above FL290 at intervals of 4,000 ft beginning at FL290

6.2.3 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

6.2.4 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

6.2.5 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

6.2.6 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);

c) Wind-Optimized Profiles without Hemispherical Rules (Cruise-climb).

The typical 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 flights were used as representative of the conditions expected at various southern 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 OD airport pairs. This is important in the study of sector occupancies to maintain a homogeneous flight data structure to derive valid traffic pattern conclusions. Two ARTCC Centers were selected for this study to restrict the number of runs to a reasonable level. Table 6.1 illustrates the case studies scrutinized in this study. In order to expedite runs ZMA and ZJX were executed in batch mode simultaneously using both AOM and AEM.

Table 6.1 Scenarios Used in the Preliminary Model Study.

	Concept of Operations		
ARTCC Center	Baseline (1996 Traffic)	RVSM (1996 Traffic)	Cruise Climb (1996 Traffic)
ZMA[a]	✓	✓	✓
ZJX	✓	✓	✓

[a] .8000 flights used of 18,000 daily flights

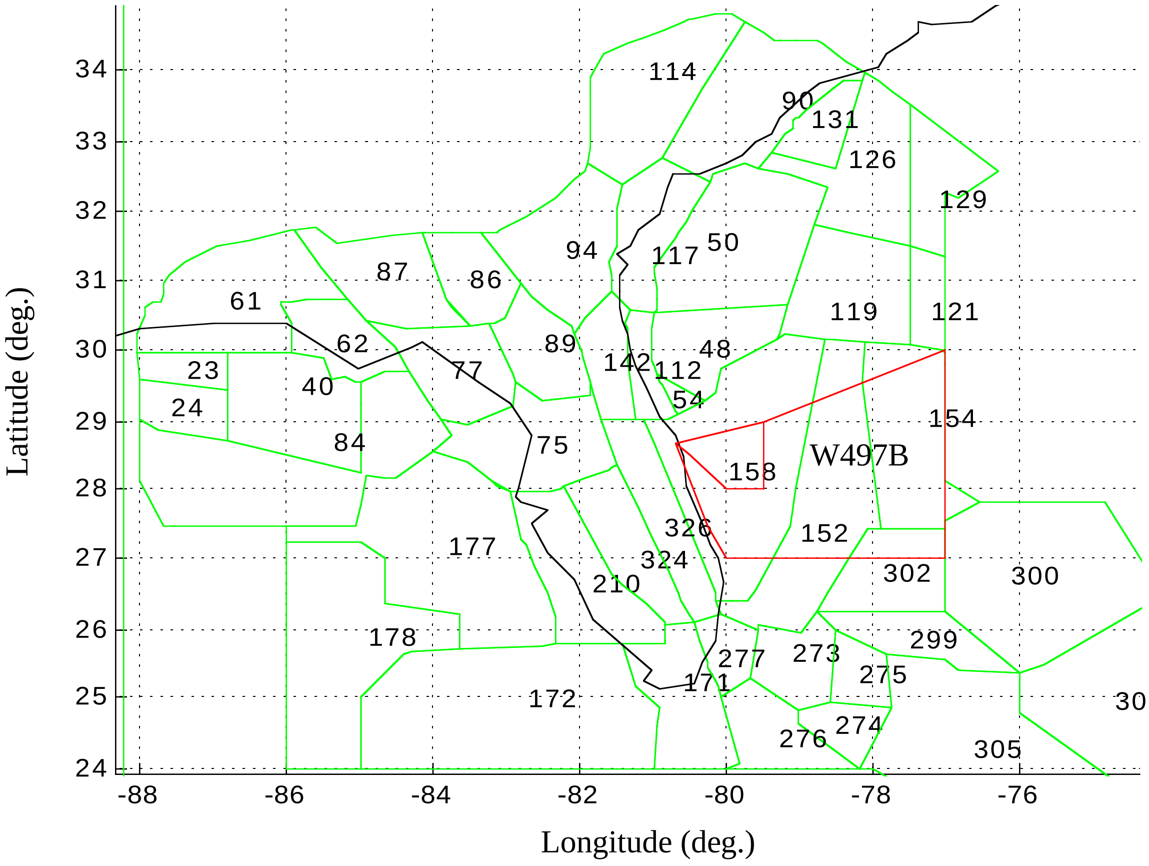


Figure 6.2 AOM View of ZMA and ZJX Sectors at FL 300 (includes CCAS warning areas).

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 6.2. A sampling rate limitation of the data is obvious from these results. It would be rather remarkable that two conflicts of severity class 2 occur over these two ARTCC Centers over an 8 hour time span. 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.

Table 6.2 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

6.3 RLV Application 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) around the KSC SUA region. 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 as NAS moves towards Free Flight.

6.3.1 Traffic Flow Patterns

Traffic flow patterns in an airspace sector are important to assess the impact of RLV integration into the Air Traffic Management system. Sector traffic volumes are used to estimate the best strategy to route flights affected by SUA regions during RLV operations. Figures 6.3 through 6.7 illustrate the changes in traffic patterns expected around the Canaveral warning areas 497A and 497B. This figures illustrate that moderate changes are expected in traffic flows as NAS moves towards Free Flight operations. Note that these results have assumed **no SUA region activation** . They merely represent the collective flight plan activity over the SUA region.

If the activation of the SUA takes place various sectors in ZMA that are parallel to the coast line (sector 77 for example) experience large increments in traffic of up to 200% (depending upon the assumptions foverning the new flight plans) in relatively short periods of time. This is the result of detours for traffic that normally uses the airways AR-1, AR-3, AR-7 and A699 off the coast.

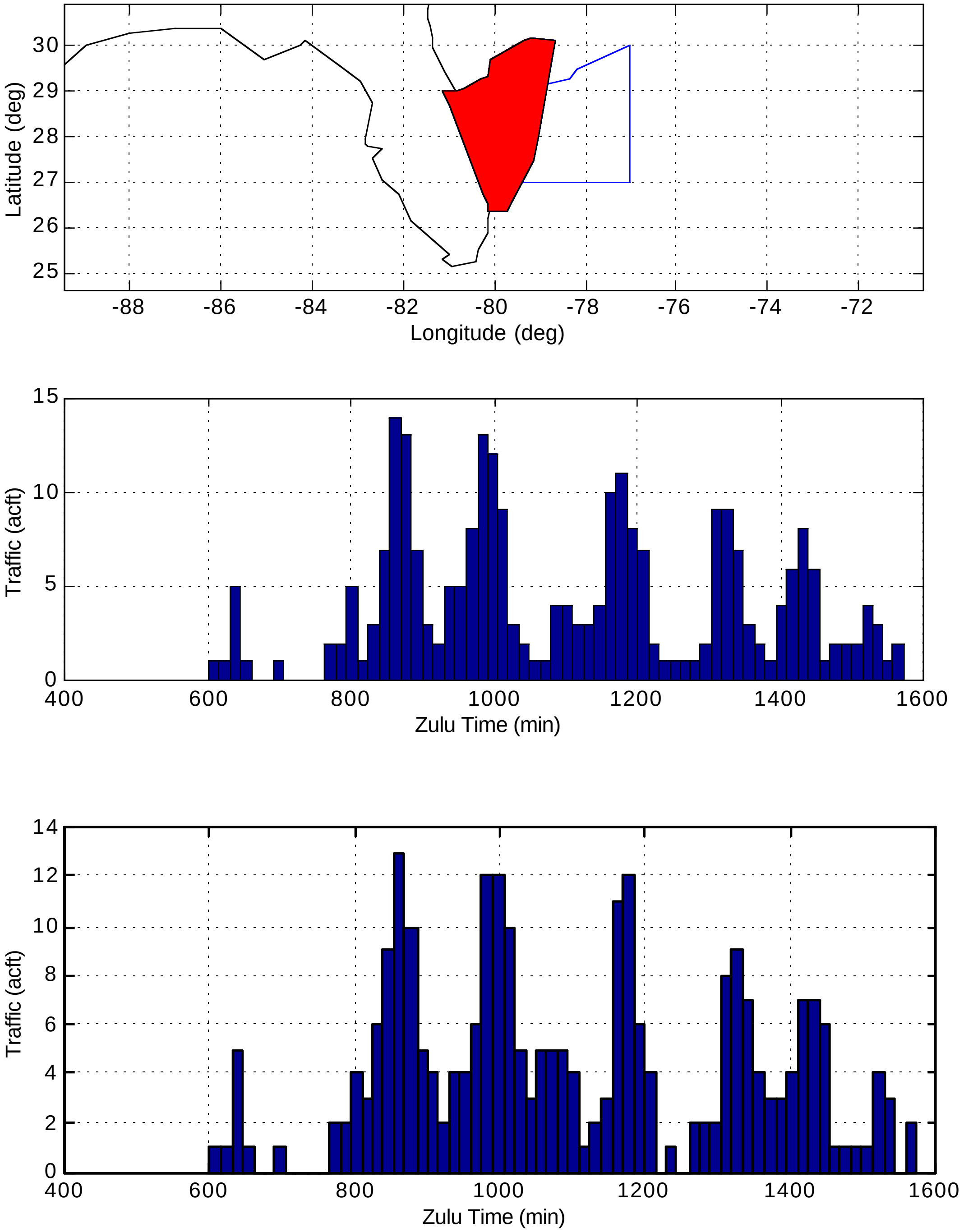


Figure 6.3 AOM Sector Traffic Results (Sector 40 ZMA, NAS Baseline Conditions).

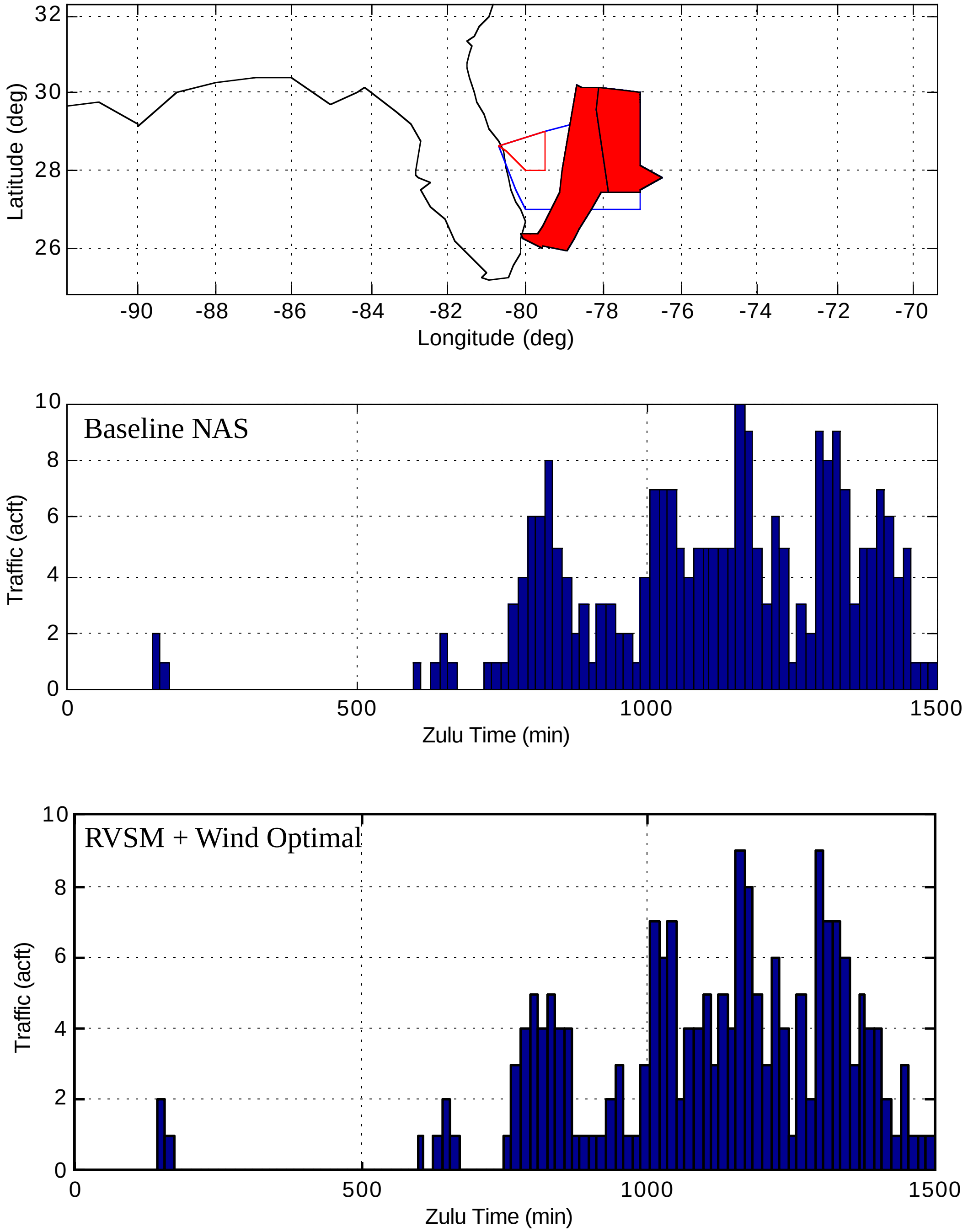


Figure 6.4 AOM Sector Traffic Results (Sector 39 ZMA, NAS Baseline and RVSM Conditions).

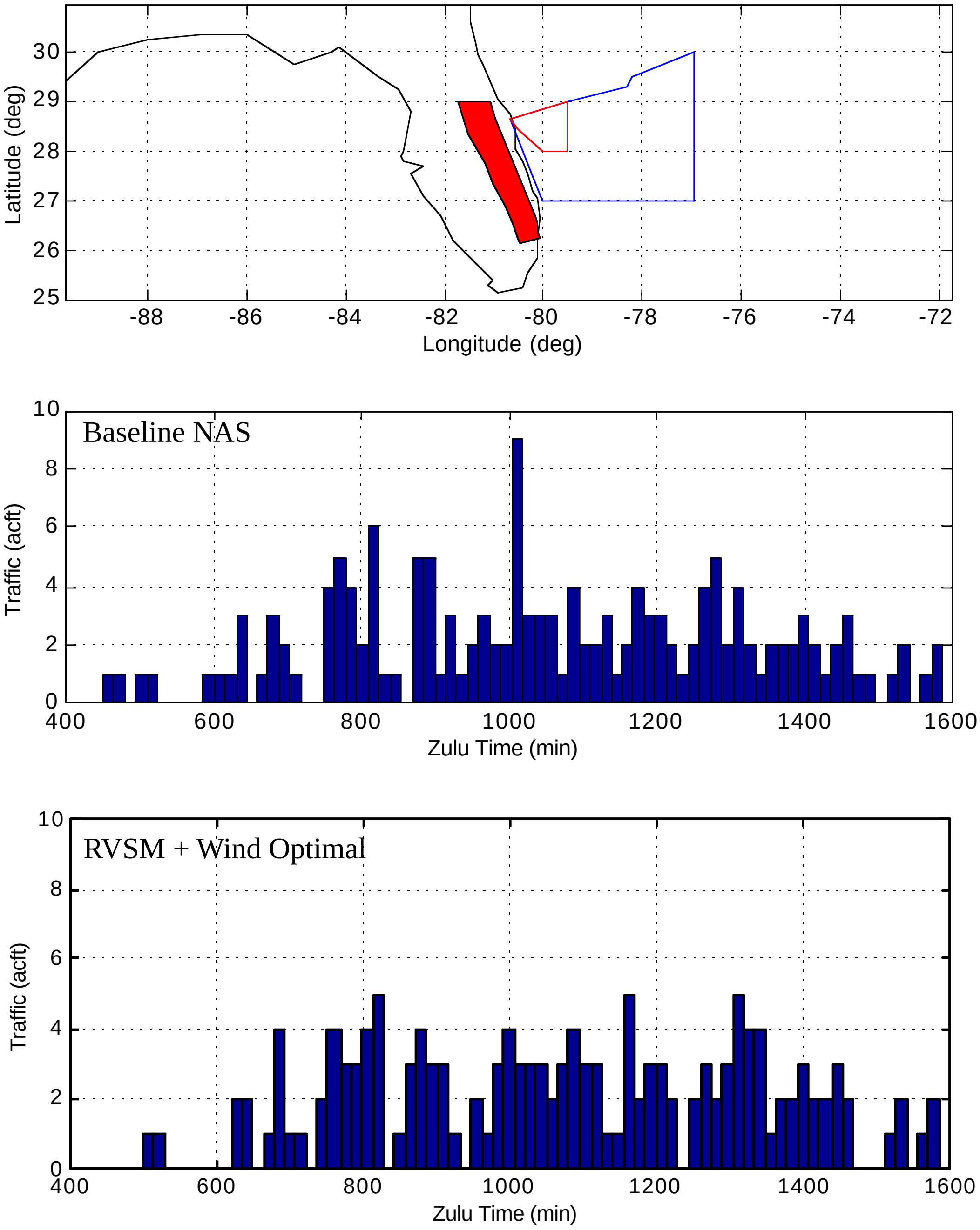


Figure 6.5 AOM Sector Traffic Results (Sector 77 ZMA, NAS Baseline Conditions).

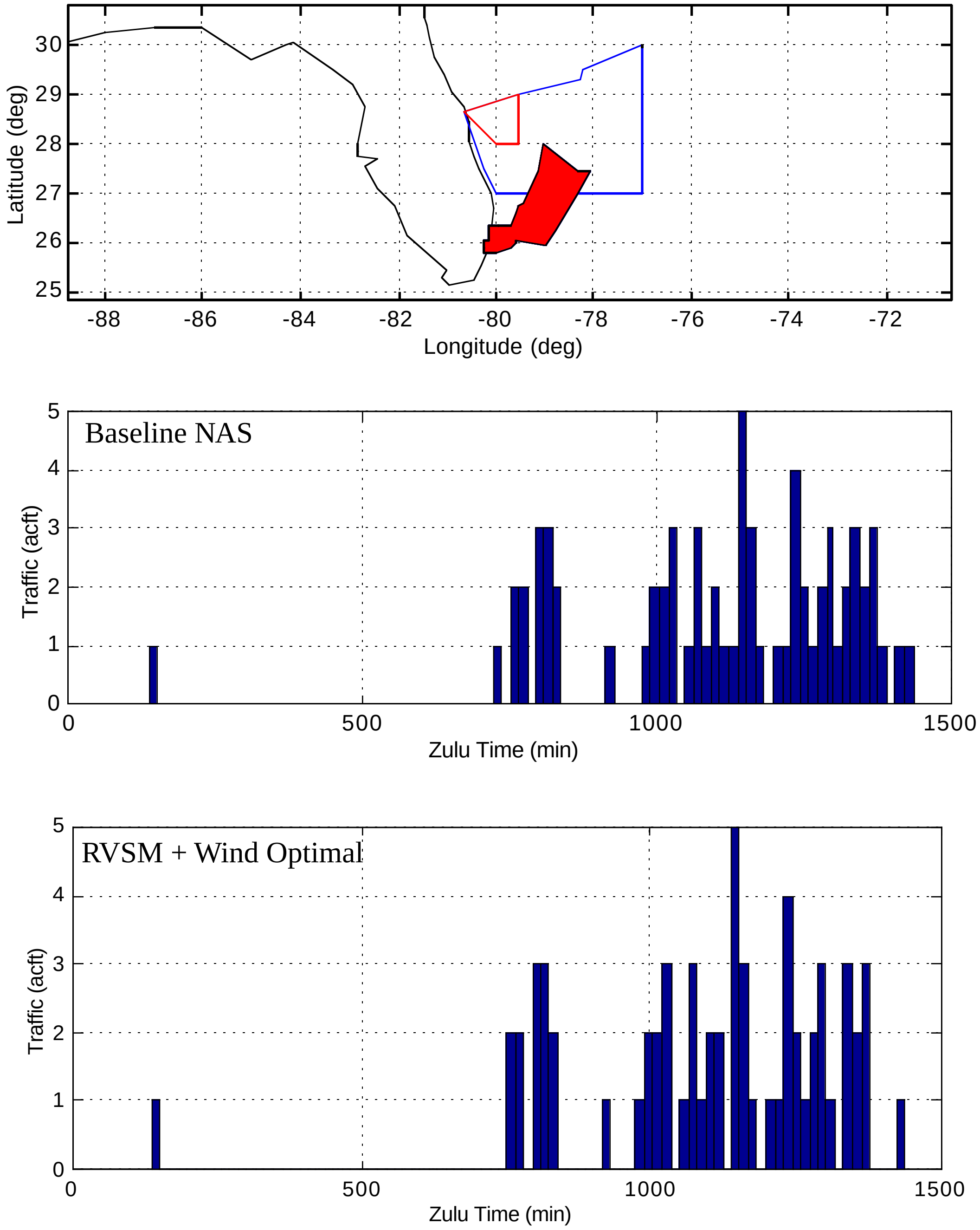


Figure 6.6 AOM Sector Traffic Results (Sector 50 ZMA, NAS Baseline Conditions).

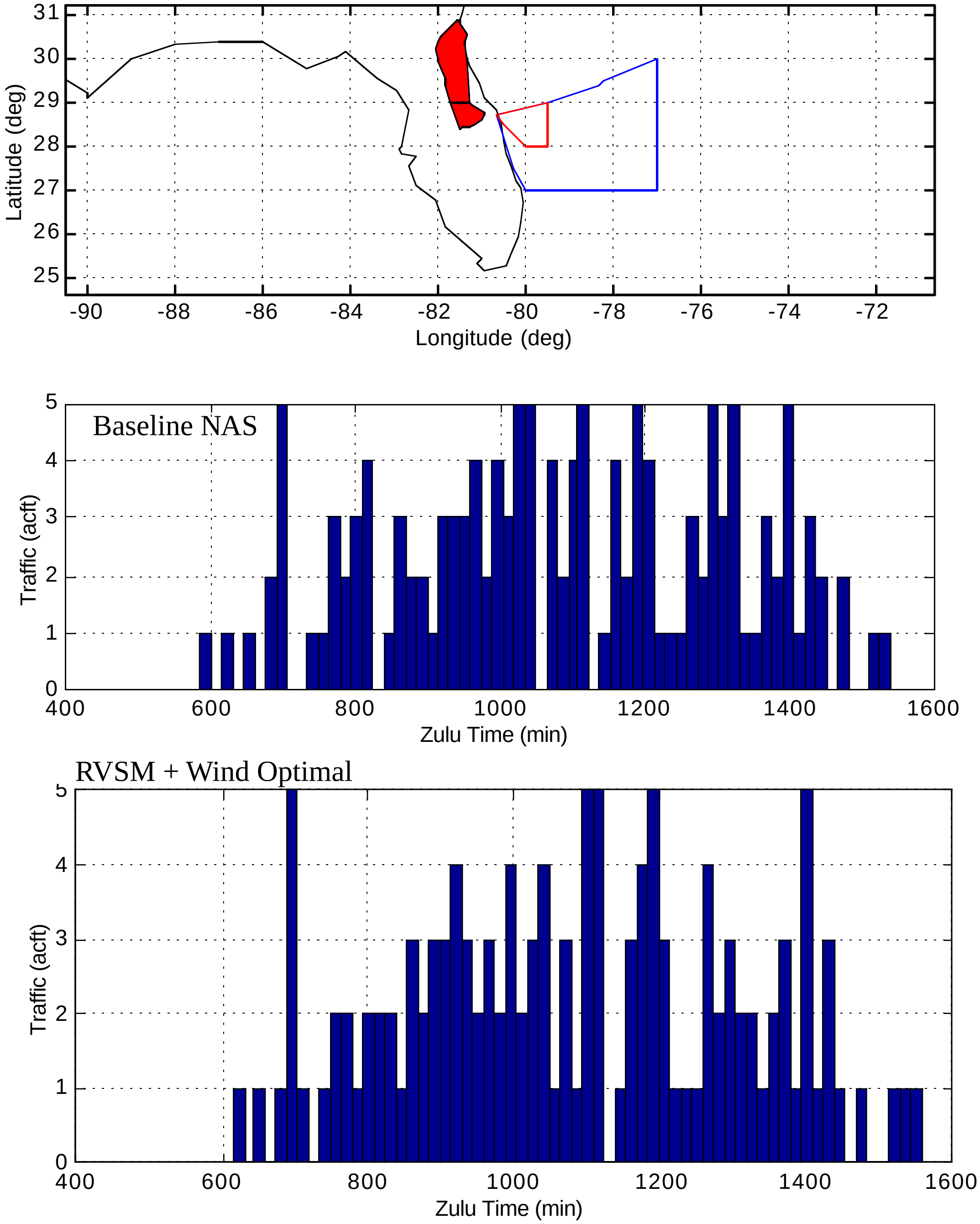


Figure 6.7 AOM Sector Traffic Results (Sector 34 ZJX, 1996 NAS Traffic Conditions).

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. In this analysis, the sequence of the 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 data. In our analysis about one third of the total flights over a 24 hour time span was used.

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 6.4 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 6.3 demonstrates that both the centers located in the Florida Peninsula show less variations in traffic flows across sectors than those in the mainland portion of CONUS. This result was expected since flight plans in Florida are well organized in a North-South direction whereas RVSM and CC flights across central ARTCC centers show significantly greater 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. This fact also implies that sectors in Florida are more prone to congestion effects under

Table 6.3 Statistical Analysis of Sector Traffic Flows (Wilcoxon Rank Sum Test α=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[a]	58	5/5	35.2 / 34.9
ZID[b]	32	10/8	102.5 / 86.3

[a] Given for comparison purposes only

b. Using 6000 baseline flights

RLV operations due to inherently larger traffic densities prevalent at these two centers.

6.3.2 Conflict Analysis and Results

Three concepts of operations were applied to two enroute control centers to assess the number of expected conflicts in the enroute airspace and during vertical transitions. Results of this analysis are shown in Tables 6.4 through 6.6. The analysis was carried out for a pair of centers in order to improve the validity of results over a larger geographical area around a known SUA region. 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 6.4 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

Table 6.5 ZMA and ZJX ARTCC Conflict Statistics (RVSM).

Conflict Type	Vertical Transition Conflicts	Enroute Conflicts
Severity 1	104	8

Table 6.5 ZMA and ZJX ARTCC Conflict Statistics (RVSM).

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

Table 6.6 ZMA and ZJX ARTCC Conflict Statistics (Cruise Climb).

Severity1	110	15
Severity2	64	13
Severity3	10	4
Total	184	32

Table 6.7 Statistical Analysis of 15-Minute ARTCC Center Conflicts.

ARTCC Center	Scenario	P Values ($\alpha = 0.05$)
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

Table 6.7 Statistical Analysis of 15-Minute ARTCC Center Conflicts.

ARTCC Center	Scenario	P Values ($\alpha = 0.05$)
ZMA/ZJX	RVSM vs. CC (enroute)	0.056
ZMA/ZJX	RVSM vs. CC(transition)	0.954

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 6.7. Only in one instance of six comparisons (RVSM vs CC at ZMA and ZJX) the number of conflicts observed is judged to be the same (statistically speaking). Another generalization obtained after a close examination of Tables 6.4 through 6.7 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 6.4 through 6.7) ATC controllers would experience less workload and thus the airspace system capacity would be judged as better 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 conflict 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 instance occurring in a more automated environment is a challenging issue that warrants further investigation.

Other important statistics gathered in this analysis and consistent with the descriptions of Chapter 4 are: a) the average relative heading of each blind conflict and its standard deviation, b) times in conflict between pairs of flights, c) relative speeds in conflict among many others. A quick summary of this analysis is presented below.

a) Both baseline and RVSM scenarios have similar relative conflict geometries whereas the cruise climb scenario shows substantial differences across the ZMA and ZJX ARTCC Centers.

b) Some uniformity exists among RVSM and CC scenarios for conflict times (shorter conflicts). Longer conflict times for the baseline conditions were observed. This result is expected because the baseline scenario uses a 2000 ft. vertical separation criteria above FL 290, thus making conflicts last longer.

c) The Closest Point of Approach (CPA) between two desired aircraft trajectories could perhaps be used as a measure of complexity that complements workload. The means of CPA are expected to be slightly higher under Free Flight conditions due to the better distribution of flights in the three dimensional airspace.

The reader should be reminded that as a result of the RLV activity at spaceports the ensuing distribution of aircraft traffic in confined volumes of airspace results in larger shares of workload for some sectors. This fact is precisely the motivation behind the optimization model described in Chapter 5 of this report.

6.4 Modeling User and Service Provider Costs

User costs are currently being investigated using various sources of information including the European BADA database, RAMS and analytic queueing models to model airspace operations. The estimation of user costs is done at two levels: a) individual flight path parameters and b) interactions of each aircraft with others in airspace sectors. In practice, these two levels of analysis are related since a single performance cost function to model the cost of an aircraft traveling from *i* to *j* is ultimately dependent upon the interactions between the aircraft and others traversing the same airspace. In fact, the cost performance function is quite complex when one considers the capacity limitations of airspace sectors and airports.

The first level addresses the need to have good quality trip cost models based on travel time and Direct

Operating Costs (DOC). In this process, the Eurocontrol BADA 3.0 database is being used to ascertain the cost of individual flights. To address the second level (i.e., aircraft interactions in the airspace and on the ground), sector capacity and airport capacity functions are currently being developed using the Reorganized Mathematical Simulator (RAMS) and using standard airport capacity delay functions derived from analytic queueing models whenever possible.

An example single aircraft performance cost function is shown in Figure 6.8 for stage lengths of 1000 nm. In this figure, an SUA having infinitesimal depth and semi distance R restricts the aircraft path to follow a near triangular trajectory following a detour decision point once the SUA is encountered.The plot depicts has parametric values for normalized cost (i.e., normalized travel time) as a function of decision distance from the SUA boundary. Other SUA shapes can be analyzed, and information on normalized costs can be similarly derived. For example, Figure 6.9 shows a typical cost function obtained for random flights from New York to Miami avoiding the Canaveral warning areas.

6.4.1 Fuel Consumption Model (Eurocontrol BADA Database)

A quantification of airspace user costs requires a bookkeeping activity to estimate Direct (DOC) and Indirect Costs (IOC) of operations. Aircraft cost functions are usually nonlinear and vary according to relevant parameters such as cruise speed, altitude, payload, wind conditions, etc. These conditions can be ascertained from the output files of AOM or from other relevant simulation models used in the analysis (e.g., RAMS). A simple, yet reliable, estimation of the DOC cost of operations is the use of realistic database containing information on fuel flows as a function of speed, altitude and aircraft mass. In this process, the research team has been using the Eurocontrol BADA 3.0 database to assess fuel costs, hence, to estimate DOC penalties imposed by airspace constraints resulting from RLV operations.

The Eurocontrol database characterizes aircraft performance using a modified parabolic drag polar. This model is relatively simple to implement in our analysis for developing aircraft cost functions for practically any aircraft mix operating around SUA regions. Figure 6.10 illustrates a typical aircraft mix using warning areas 497A and 497B.

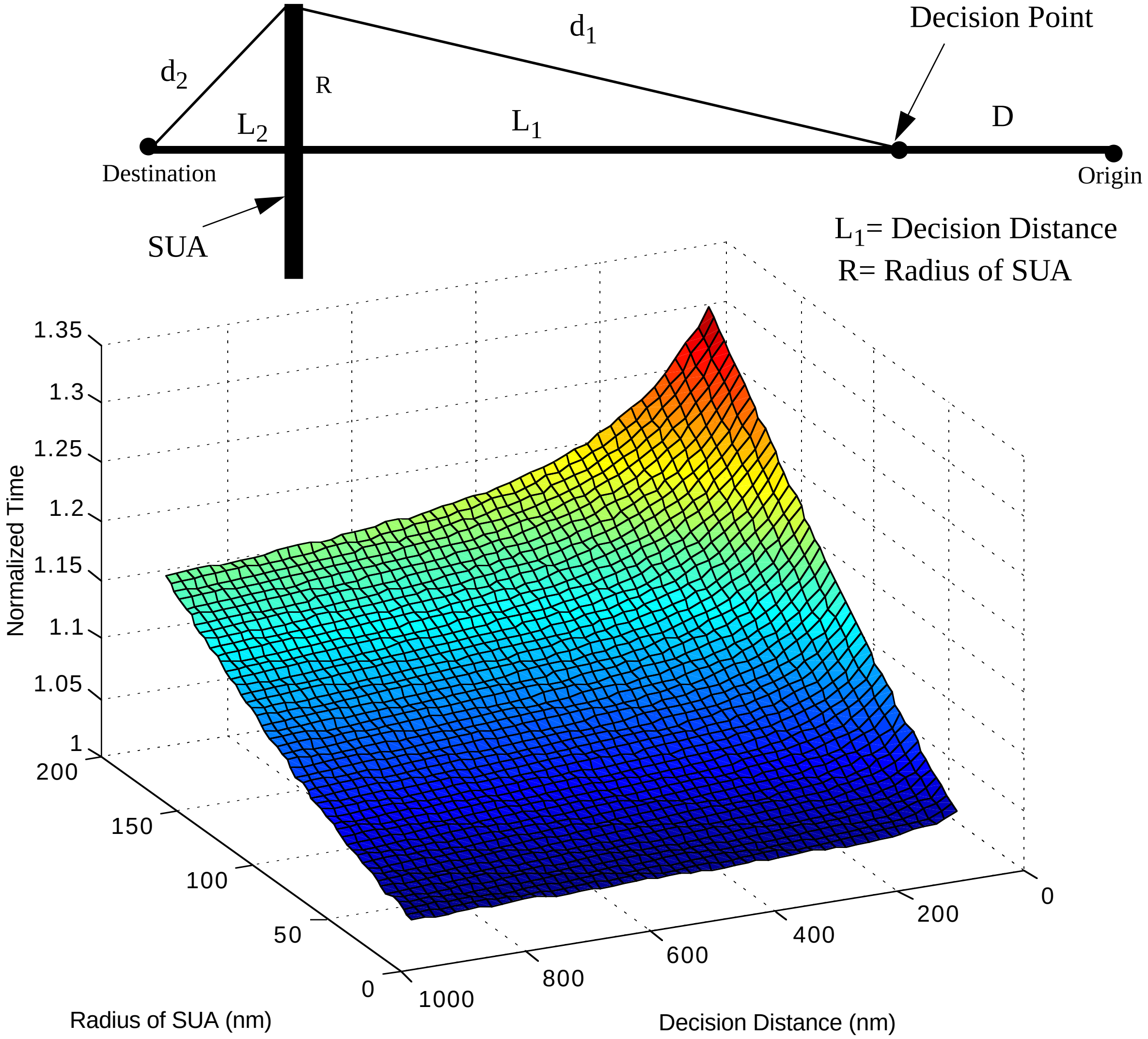


Figure 6.8 Sample Aircraft Performance Cost Function (1000 nm stage length)

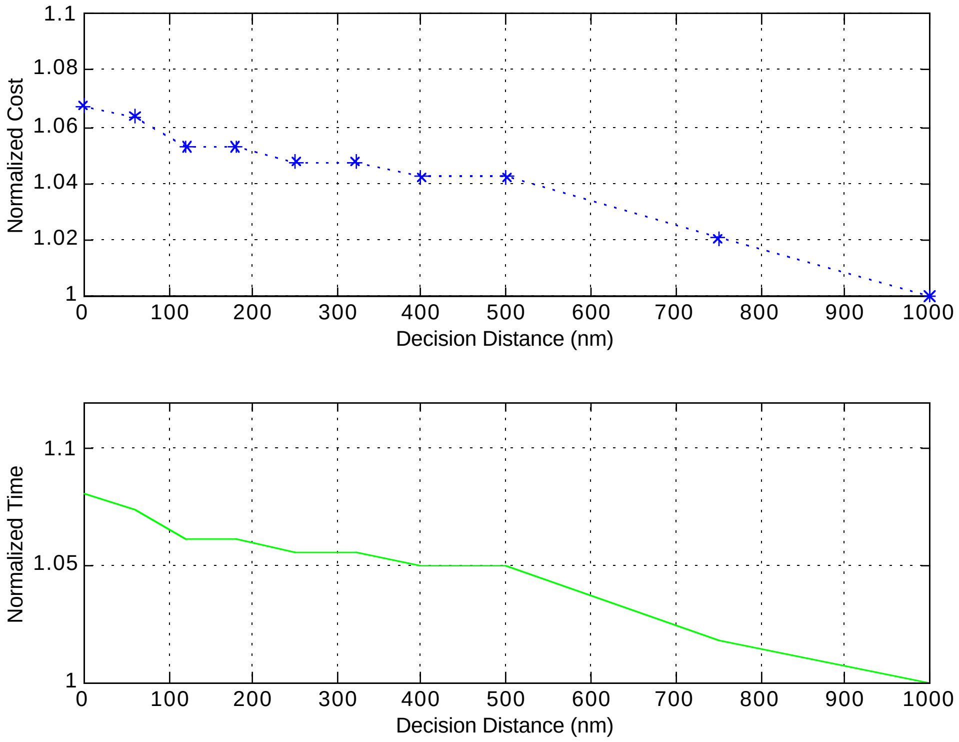


Figure 6.9 Normalized Cost and Time for Boeing 757-200 Traveling from New York to Miami.

The BADA model defines thrust specific fuel consumption (η) (in kg/min/kN) as follows (Eurocontrol, 1998),

$$\eta = C_{fl}(1 + V_{TAS}/C_{f2}) . \tag{6.1}$$

The cruise fuel flow (f_{cr}) is then given by

$$f_{cr} = \eta \times T \times C_{fcr} \tag{6.2}$$

where:

C_{fl} , C_{f2} , and C_{fcr} are aircraft specific coefficients

V_{TAS} is the true airspeed (knots)

T is the engine thrust (kN)

Using this model, fuel consumption functions can be derived for various operating conditions. Figure 6.12 illustrates the Specific Air Range (SAR) for a small transport aircraft. Note that in this case, SAR is also a function of wind conditions and aircraft Mach Number as represented in the diagram. At this stage, a simple relationship between fuel cost and DOC is being used to estimate the cost of detours around spaceport regions.

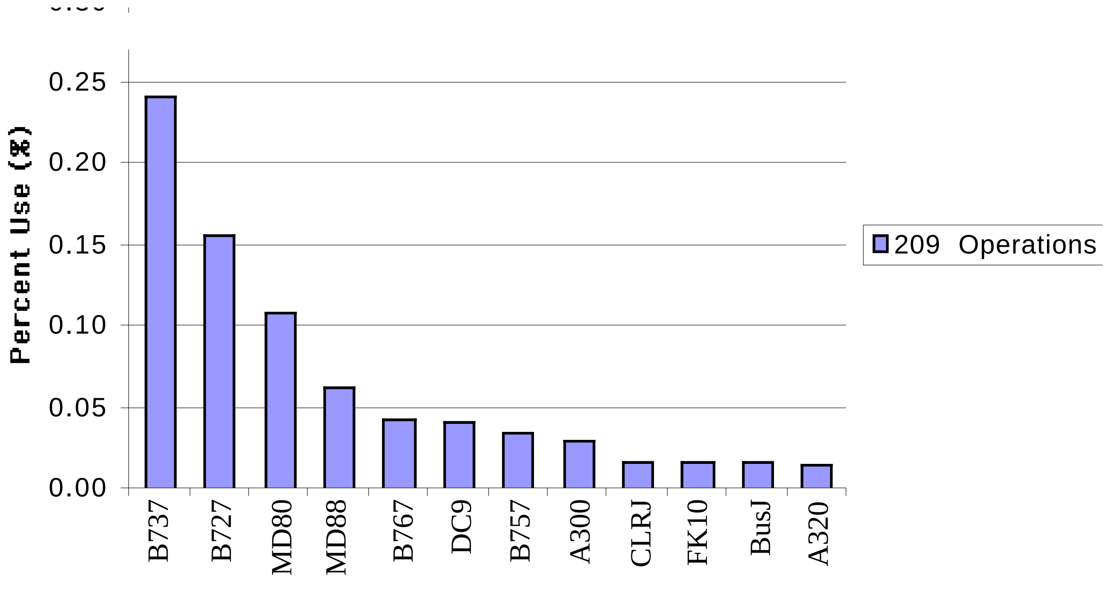


Figure 6.10 Typical Aircraft Mix Using Cape Canaveral Warning Areas (1996-1997 ETMS Traffic).

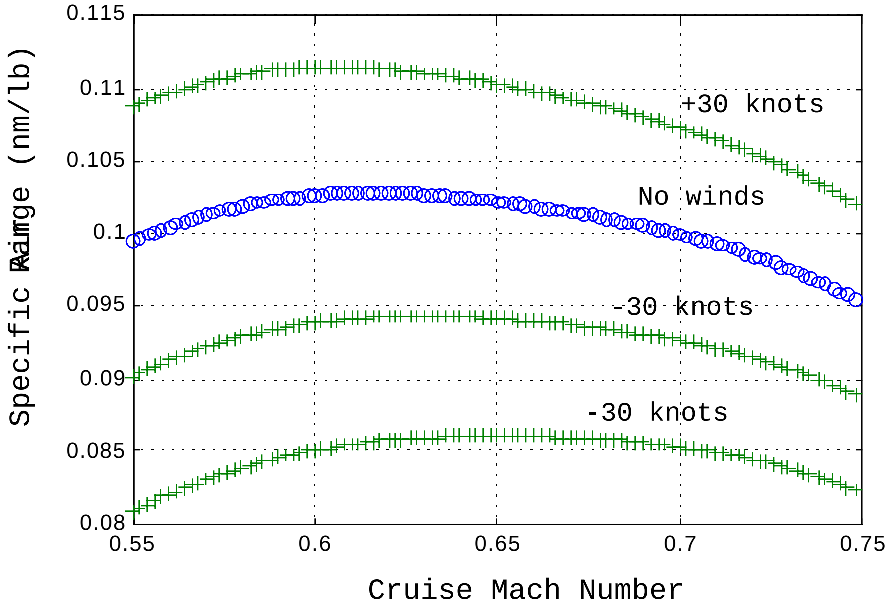


Figure 6.11 Specific Air Range Parameter (SAR) for the Fokker 100 Transport Aircraft.

6.5 Application of the Airspace Planning Model

In order to illustrate the methodology to minimize user and service provider costs as explained in Chapter 5 of this report, we used a simple, yet realistic, example to examine the operational features of the Airspace Planning Model (APM). This example involves 15 flights converging over a narrow region of airspace West of the Canaveral Warning Areas 497A and 497B. The idea is to illustrate that various flight plans (or real flights for that matter) traversing the areas adjacent to an active SUA can be managed using the APM model described in Chapter 5 of this report. The example presented here is offered only as a proof-of-concept that Air Traffic services can manage the selection of flight plans to reduce the costs to users and service providers using an optimization model.

Figures 6.12 and 6.13 illustrate the traffic situation around the Cape Canaveral Air Station (CAAS). when SUA activation is in place. Figure 6.13 depicts one of several critical airspace sectors (sector 77) belonging to the Miami Air Route Traffic Control Center (MIA ARTCC). The traffic situation between

115 and 160 minutes indicates heavy use of this sector due to aircraft detours around the SUA. Figure 6.14 illustrates all possible flight paths of the15 flights affected in this example. Figure 6.14 includes four *surrogate* flight plans for each one of the nominal 15 flights affected by the SUA activation. These *surrogate* flights represent various *alternative* flight plans selected by the airline or by the crew on-board each aircraft.

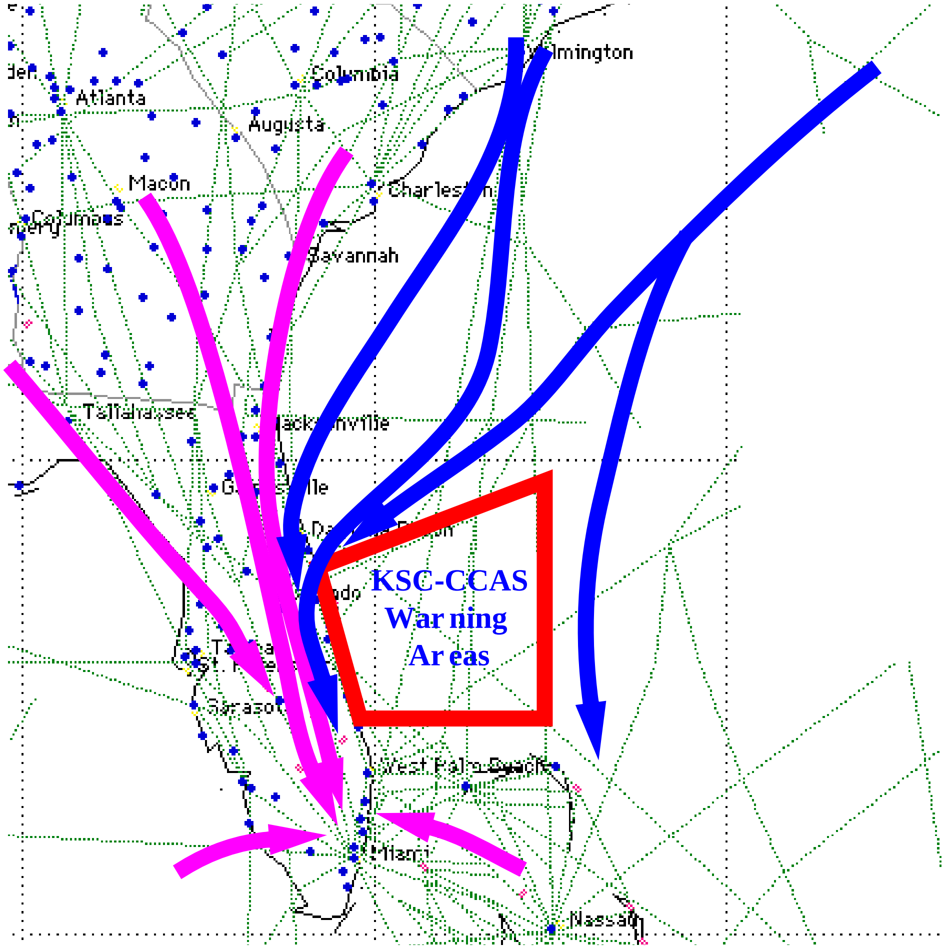
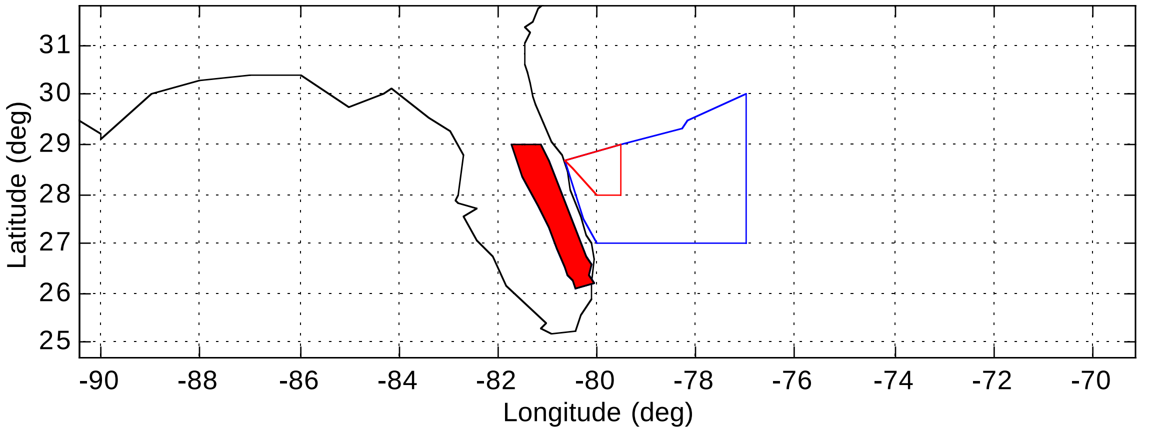


Figure 6.12 Primary Detour Patterns Considered in the Optimization Problem.



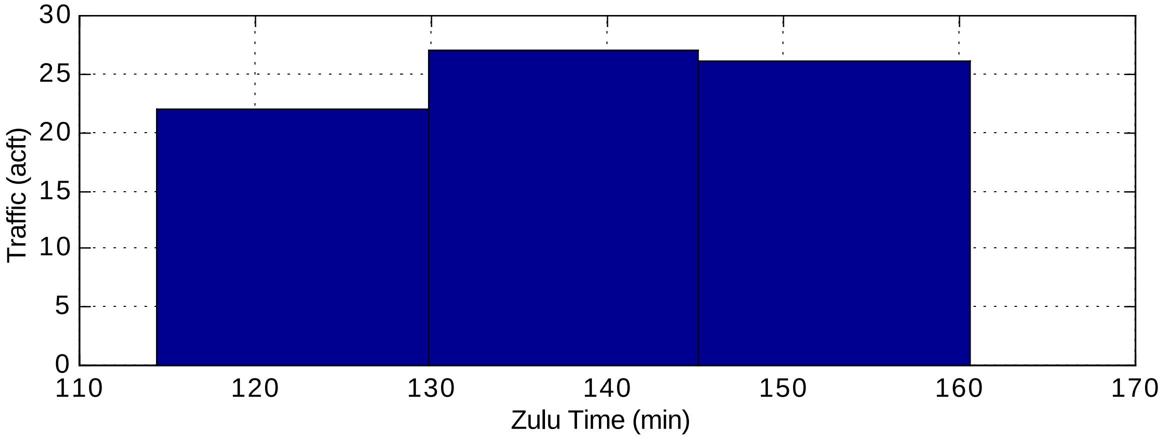


Figure 6.13 Sector 77 Occupancy Over Time.

In this illustrative example these flight plans have been generated manually, but in practice, optimal flight plan generators such as Jeppesen Flite Star^{TM} or OPGEN^{TM} can be employed. AOM and AEM have been developed to read the output data structures used in OPGEN, thus making the generation of such flight paths possible with ease. Figure 6.15 illustrates various surrogate flight plans for two aircraft to accommodate detours around an active SUA.

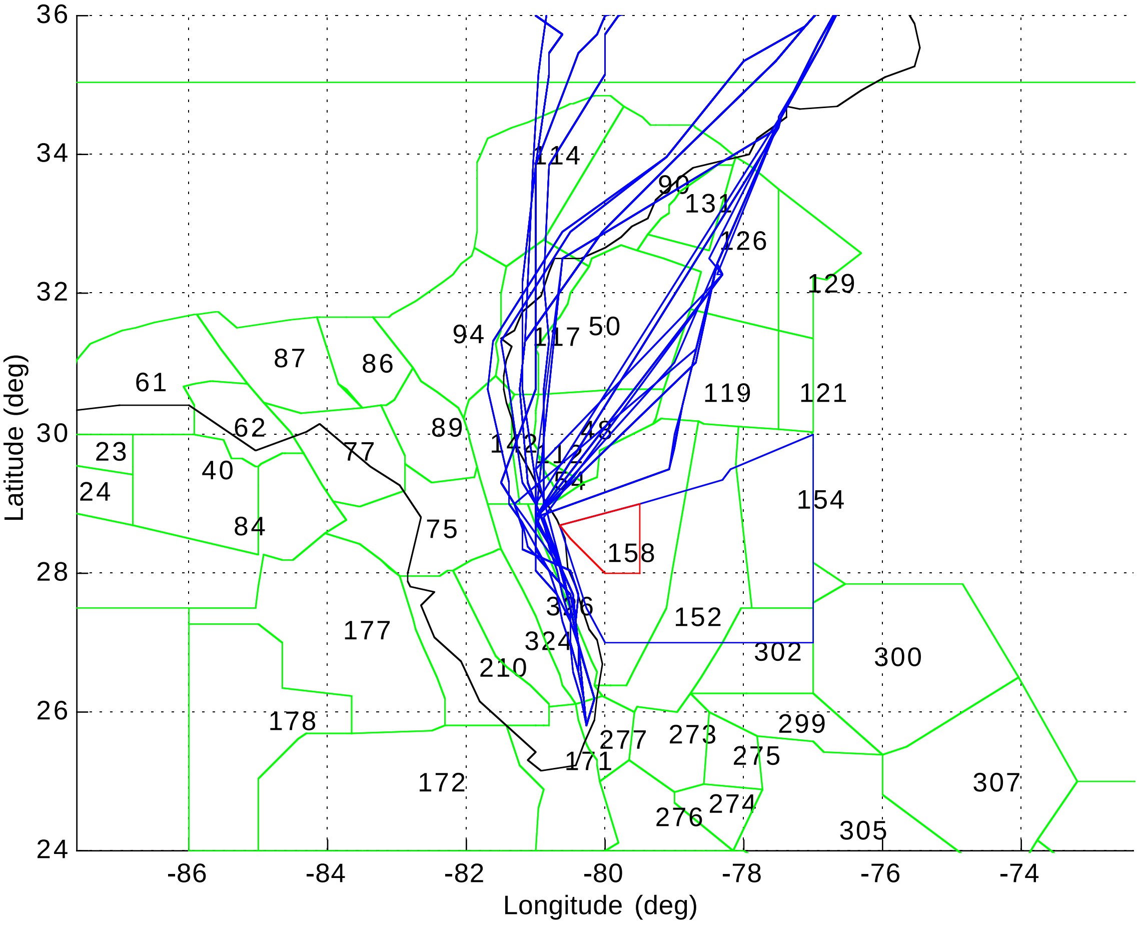


Figure 6.14 Sample Flight Paths for 15 Aircraft and Their Surrogate Flight Plans.

In practice surrogate flight plans can be generated before the aircraft leaves the departing airport or could also be generated in-flight using standard functions found in modern Flight Management Systems (FMS). The important point to realize is that aircraft detours around SUA can be planned using computer simulation and optimization models considering both users and service providers.

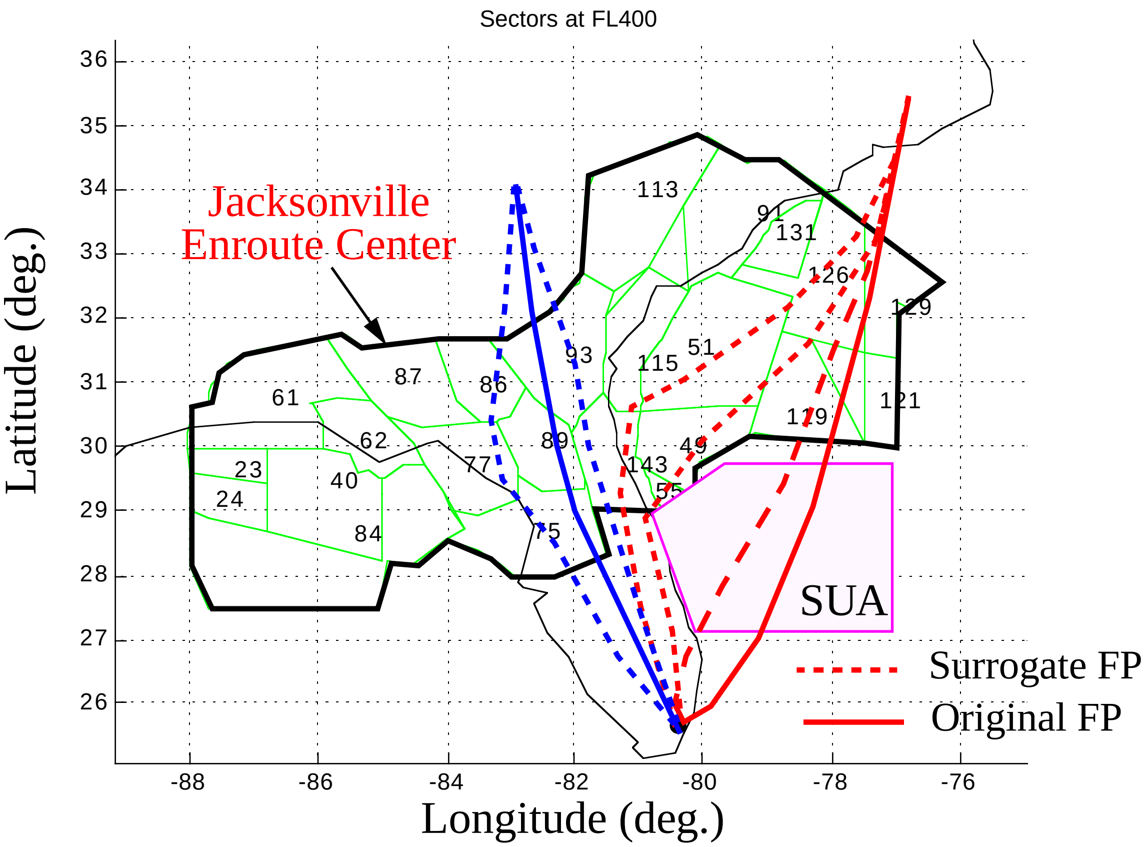


Figure 6.15 Notional Illustration of Optimization Model Detour Procedures.

Table 6.8 illustrates all nominal flight plans used in this numerical example. Note that these flight plans have deliberately being selected at small intervals from one another to produce numerous airspace conflicts and thus increase the complexity of the model solution. The reader will note that even though the number of flights considered in this example is quite small, the large overlap between flight paths results in several hundred possible conflicts in the model.

The mixed-integer program used to solve this optimization problem includes 15 constraints to enforce that exactly one flight plan is chosen for each flight, and 176 constraints (88*2) to define the workload variables in terms of binary variables. Two hundred constraints were needed to describe the overlapping flights in each sector (see computation of maximal overlapping subsets of flights). There were 88 ‘fatally’ conflicting flights, each of which generate a constraint. The remaining conflict constraints, which made up the bulk of the problem, consisted of 15,624 constraints. In addition, 7 constraints were used to set up the ineffectiveness measure, for a total of 16,110 constraints.

There are 75 binary variables corresponding to the selection of a flight plan for each flight, 880 binary variables and 88 continuous variables for the workload constraints, and 2 continuous variables for the ineffectiveness constraints. This gives a total of 955 binary variables and 90 continuous variables. Table 6.9 illustrates a baseline result for the solution to this problem using insignificant ineffectiveness costs coefficients in the model (10 units for maximum discrepancy in ineffectiveness, and 5 units for maximum ineffectiveness). This reflects a situation where airline equity is not a critical issue in the diversion of flights around SUA. Table 6.10 provides maximum the assumed sector occupancie parameters, used in the model. In this case the occupancy costs are factored at a rate three times the occupancy of the sector. Although these parematers are arbitrarly selected for the sake of illustration, in practice, some extra work is needed to understand the balance between user and service provider costs.

Table 6.11 illustrates a modified diversion strategy for the same problem using more significant ineffectiveness costs coefficients in the model (5000 units for maximum discrepancy in ineffectiveness and 200 units for maximum ineffectiveness). This illustrates a solution where airline equity is quite important in the diversion of flights around SUA. In this case, of the 15 flights selected a different flight plan is in the optimal solution..

Table 6.8 Nominal Flight Plans for Fifteen Aircraft Used in the Sample Problem.

Flight Number (Surrogate FPs)	Flight	Origin-Destination	ETD / ETA (UTC - min)	Aircraft Type	Cruise Altitude (FL)	Cost ($)
1 (4)	AA 1017	LGA-MIA	4.91 / 151.87	B727	310	11,022

Table 6.8 Nominal Flight Plans for Fifteen Aircraft Used in the Sample Problem.

Flight Number (Surrogate FPs)	Flight	Origin-Destination	ETD / ETA (UTC - min)	Aircraft Type	Cruise Altitude (FL)	Cost ($)
2 (4)	AA 1181	EWR-MIA	5.61 / 149.45	B727	310	10,825
3 (4)	AA 1291	BOS-MIA	6.18 / 174.53	B757	310	12,626
4 (4)	TWA 505	BOS-MIA	9.60 / 179.10	B727	310	12,712
5 (4)	TWA 500	LGA-MIA	7.91 / 154.40	B727	310	10,987
6 (4)	USA 53	LGA-MIA	7.11 / 151.45	B727	310	10,825
7 (4)	AA 1759	BOS-MIA	6.10 / 169.13	B757	310	12,227
8 (4)	USA 759	BOS-MIA	7.35 / 170.13	B757	310	12,300
9 (4)	USA 700	BOS-MIA	8.10 / 167. 12	B757	310	11,924
10 (4)	USA 1104	PIT-MIA	6.82 / 136.37	B757	310	9,735
11 (4)	AA 100	PIT-MIA	10.82 / 138.50	B757	240	9,825
12 (4)	USA 100	BOS-MIA	7.10 / 140.63	B757	310	12,300
13 (4)	AA 50	PIT-MIA	9.82 / 139.70	B757	310	9,745

Table 6.8 Nominal Flight Plans for Fifteen Aircraft Used in the Sample Problem.

Flight Number (Surrogate FPs)	Flight	Origin-Destination	ETD / ETA (UTC - min)	Aircraft Type	Cruise Altitude (FL)	Cost ($)
14 (4)	USA 11	PIT-MIA	12.82 / 142.32	B757	310	9,735
15 (4)	AA 25	PIT-MIA	11.80 / 141.60	B757	310	9,825

Table 6.9 Optimal Detour Plans for 15 Flights Affected by CCAS SUA Activation (Baseline[a]).

Flight	Selected Plan	Flight	Selected Plan
AA 1017	4	USA 700	1
AA 1181	4	USA 1104	1
AA 1291	3	AA 100	1
TWA 505	4	USA 100	1
TWA 500	4	AA 50	1
USA 53	1	USA 11	1
AA 1759	1	AA 25	4
USA 759	4		

[a] Insignificant airline ineffectiveness costs.

Table 6.10 Maximum Sector Occupancies for Sample Problem.

Sector	Max. Occupancy (aircraft)	Sector	Max. Occupancy (aircraft)
1	3	34	1
16	3	40	10
24	5	49	1
26	6	52	4
28	6	68	5
29	1	77	6
30	5	85	4

Table 6.11 Optimal Detour Plan for 15 Flights Affected by CCAS SUA Activation (Modified Scenario[a]).

Flight	Selected Plan	Flight	Selected Plan
AA 1017	1	USA 700	1
AA 1181	3	USA 1104	1
AA 1291	3	AA 100	5
TWA 505	4	USA 100	1
TWA 500	4	AA 50	1

Table 6.11 Optimal Detour Plan for 15 Flights Affected by CCAS SUA Activation (Modified Scenario[a]).

Flight	Selected Plan	Flight	Selected Plan
USA 53	4	USA 11	1
AA 1759	4	AA 25	2
USA 759	2		

[a] Substantial contribution of ineffectiveness costs.

6.6 Costs Component Model

FAA and Space Transportation System decision makers will face the difficult question on how to estimate costs to users, ATC, and RLV service providers. The optimization model presented in Chapter 5 of this report represents a first effort to mitigate the cost of RLV operations to airspace users and airspace service providers (FAA). The use of this model is considered to be of tactical benefit to the agency and users as it introduces microscopic modeling effects of airspace flows around the SUA in question. A general procedure to present cost information to decision makers and RLV service providers is detailed in this section. This procedure is currently being developed as part of the third phase of this project and will be discussed in more detail in the months ahead.

Figure 6.16 illustrates the process for the evaluation of economic impacts of RLV integration into ATM. Until now, the research team has developed modeling procedures in boxes 1 through 7 (including the MIT research on RLV modes of operation). Some of these procedures are currently being executed using standard off-the-shelf simulation models such as RAMS to assess general delay patterns. However, the general solutions to RLV integration require high-level mathematical modeling abstractions that usually go beyond the temporal realm of one day or few hours of simulation results. For this reason, a cost component model that synthesizes the outcomes of microscopic computer simulations and the outcomes of the optimization procedure explained in this report are needed. The approach to this cost model is more macroscopic because cost estimates need to be accrued over longer periods of time (i.e., over the life-cycle of the system).

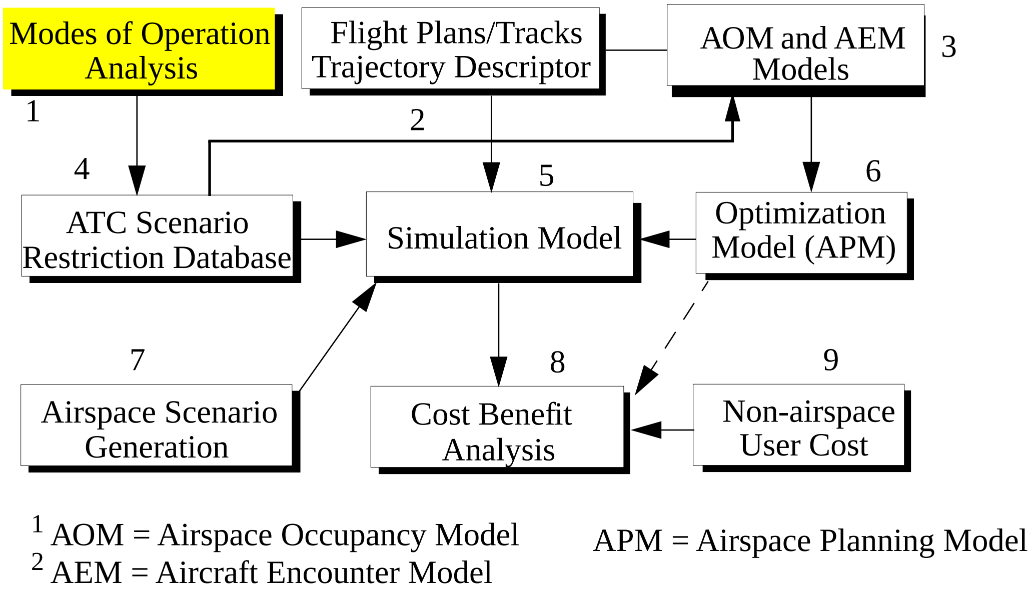

Modes of Operation Analysis
1
Flight Plans/Tracks Trajectory Descriptor
2
AOM and AEM Models
3
4
ATC Scenario Restriction Database
5
Simulation Model
6
Optimization Model (APM)
7
Airspace Scenario Generation
8
Cost Benefit Analysis
9
Non-airspace User Cost
1 AOM = Airspace Occupancy Model
2 AEM = Aircraft Encounter Model
APM = Airspace Planning Model

Figure 6.16 Framework to Assess the Impact of RLV Operations in NAS.

Figure 6.17 illustrates a prototype model to capture life-cycle impacts of RLV operations on users and service providers. The prototype model uses the Systems Dynamics methodology developed by Forrester (Forrester, 1967) and shows three state variables represented by the NAS traffic, regional airport and airspace capacities. Delay functions are quantified using microscopic models such as RAMS and SIMMOD and then included in the RLV cost model to project costs and benefits over time. Decision parameters in this model are shown in Figure 6.18. The RLV mode of operation dictates the SUA size which in turn influences the amount of traffic detoured around SUA and thus affected by the SUA activation period. Traffic is quantified using simulation models and delay parameters are extracted for a generic airspace configuration. Finally, delay costs are projected based on the aircraft mix and period of influence and thus total RLV impacts are quantified. This process is integrated forward in time to estimate life-cycle costs.This model is currently being refined as part of phase III of this project.

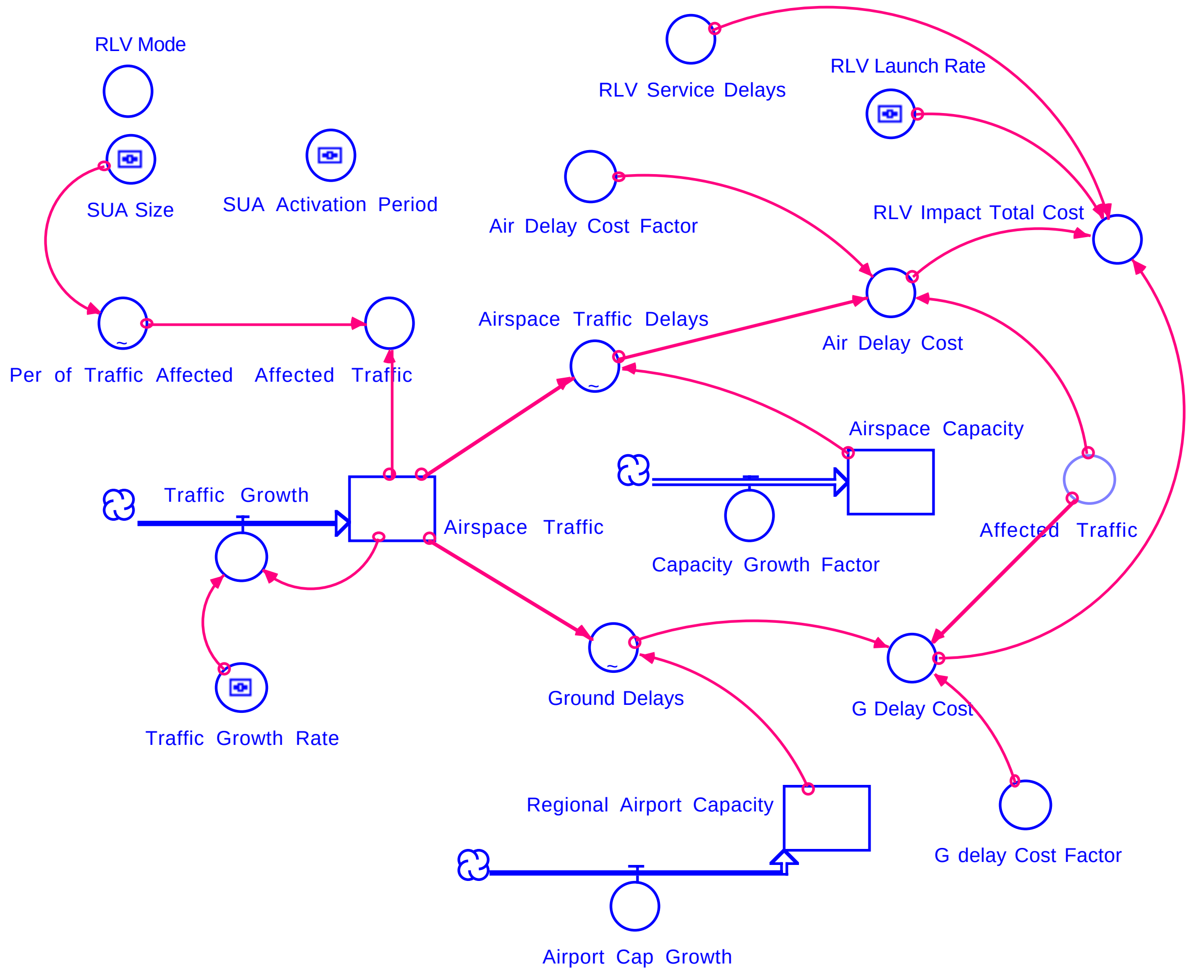
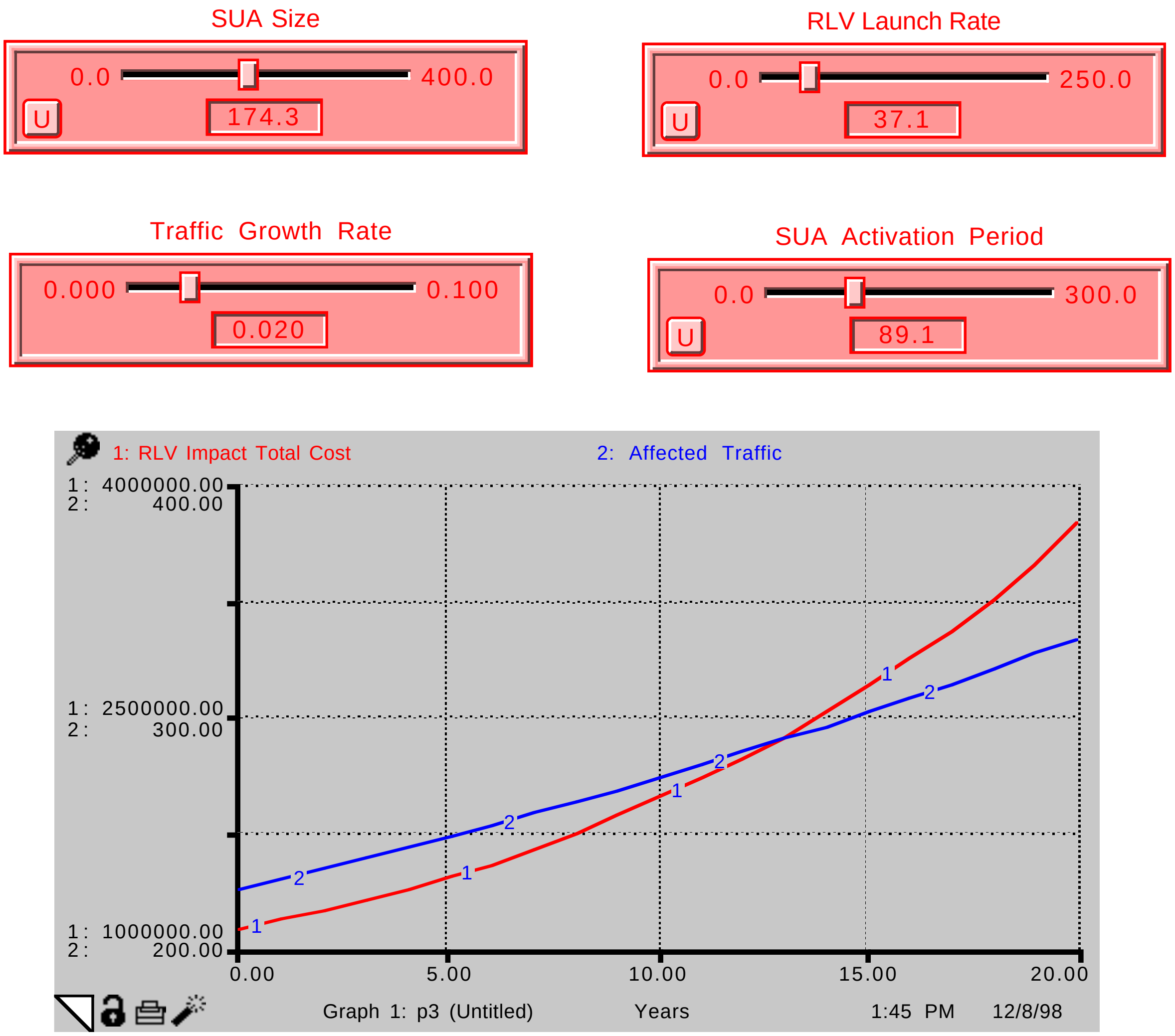


Figure 6.17 Top Level Cost Component Model Causal Diagram.



SUA Size: 0.0 – 400.0, 174.3

RLV Launch Rate: 0.0 – 250.0, 37.1

Traffic Growth Rate: 0.000 – 0.100, 0.020

SUA Activation Period: 0.0 – 300.0, 89.1

Figure 6.18 Notional Cost Curves for KSC-CCAS SUA.

CHAPTER 7 Concluding Remarks

This report has presented various modeling strategies for estimating the impacts of RLV operations on ATM. The focus has been in the development and demonstrated use of three computer models (AOM, AEM and APM) to predict traffic flows across well defined volumes of airspace, to identify potential conflicts in the airspace system, and to suggest optimal detours around activated SUA regions.

The models developed in this project have been coded in Matlab, a general engineering language, allowing their execution on any computer platform (PCs, PowerPC Macs, and UNIX workstations) without modifications. The optimization extensions of APM are coded in CPLEX, a general mathematical programming software pacakge available for PCs and UNIX workstations. The models presented here represent a toolset of algorithms that could be used in planning tactical or strategic resolution of flights around spaceports, where launch or reentry activity requires commercial and GA traffic to execute detours.

These models are currently being used in conjunction with standard simulation models to assess general impacts of various SUA regions representing various modes of operation. Until now the major effort behind our research has concentrated in developing tools and techniques to assess impacts. In the current final phase of the project, these models and techniques are being synthesized to derive quick estimates of RLV impacts for various levels of traffic and SUA envelopes.

From our analysis of current and predicted air traffic activity in NAS, the following conclusions are evident:

1) There would be moderate to substantial variations in traffic flow patterns across various ARTCC sectors in NAS. The introduction of flexible flight planning rules expected as a result of Free Flight would have different effects on the various ARTCC centers according to their geographical location . In this study the ZMA and ZJX centers had less variations in 15-minute traffic flows from the baseline case than those observed at ZID and ZTL.

2) While the magnitude of RLV impacts is currently being assessed using the models developed, the potential RLV impacts are likely to grow moderately with traffic growth. Recent data generated by the FAA and CSSI indicate that for the Florida region alone, traffic levels at the 25 busiest airports will increase 50% in the year 2010. This fact will result in longer interactions between flights to be detoured around SUA regions, and thus increase the RLV impacts to commercial and GA aviation.

3) Based on NARIM model data there would be substantial to moderate differences in the time and space distribution of traffic levels under the Free Flight Concept of Operations (i.e., RVSM and Cruise Climb scenarios). The effect of these distributions on ATC controller workload should be factored into detouring traffic around SUA regions. The APM model developed represents a tool to mitigate the impacts of these detours.

4) The models developed provide valuable insights on traffic flow patterns around spaceports including the identification of potential bottlenecks due to conflicting filed flight plans.

5) The optimization model APM has been successfully demonstrated with small-scale problems to predict flight detours around warning areas 497A and 497B. The real cost savings through the implementation of the prescribed flight plans is currently being investigated.

APPENDIX A

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 ith sector module and h(i, 2) will be the ceiling altitude of the ith 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 jth face of the ith 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∑(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 fight 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 extreme 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 airspace 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 airspace 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

KSC-CCAS Sector Traffic Loads

Traffic flow patterns in an airspace sector are important to assess the impact of RLV integration into the Air Traffic Management system. Sector traffic volumes are used to estimate the best strategy to route flights affected by SUA regions during RLV operations. Tables B.1 through B.10 in this appendix show graphic summaries of the traffic patterns around KSC-CCAS and comprise sectors at two enroute control centers (ZMA and ZJX) under three NAS operational concepts (Baseline, RVSM and Cruise Climb).

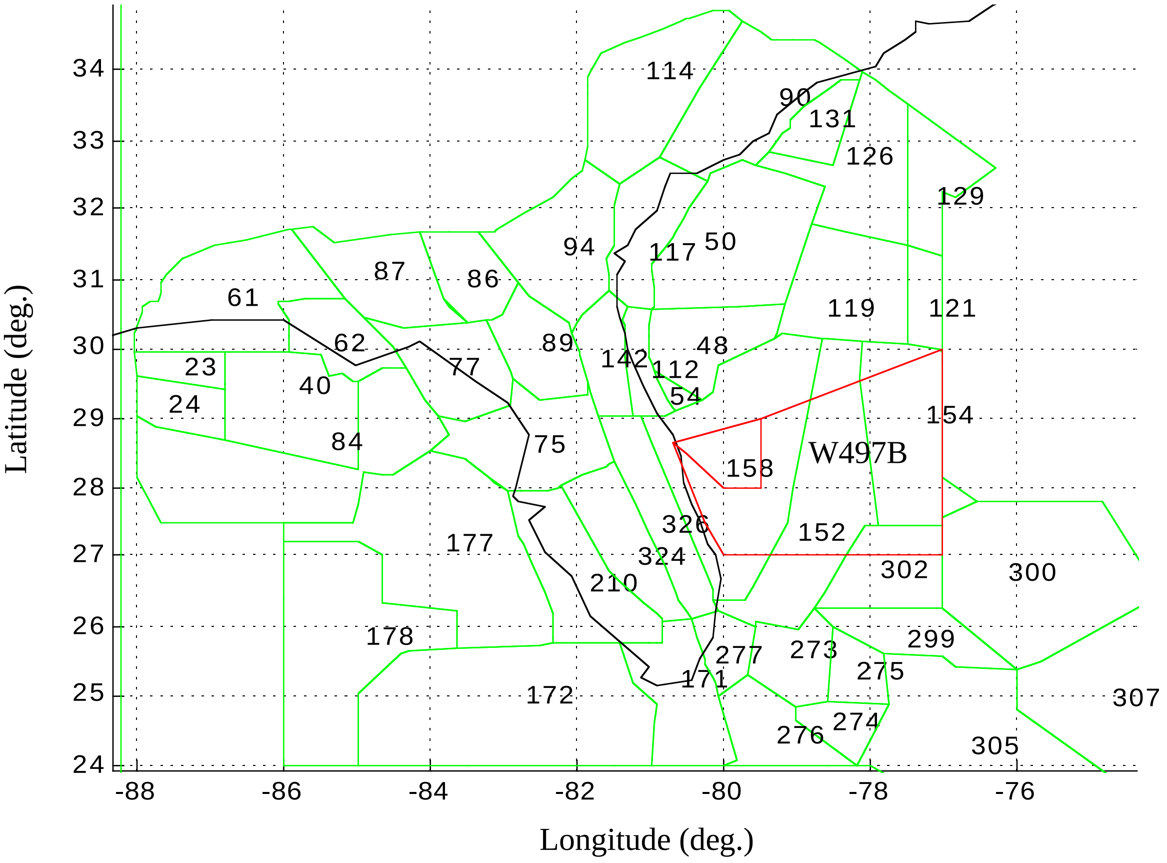


Figure B.1 AOM View of ZMA and ZJX Sector Modules at FL 300 (includes CCAS warning areas).

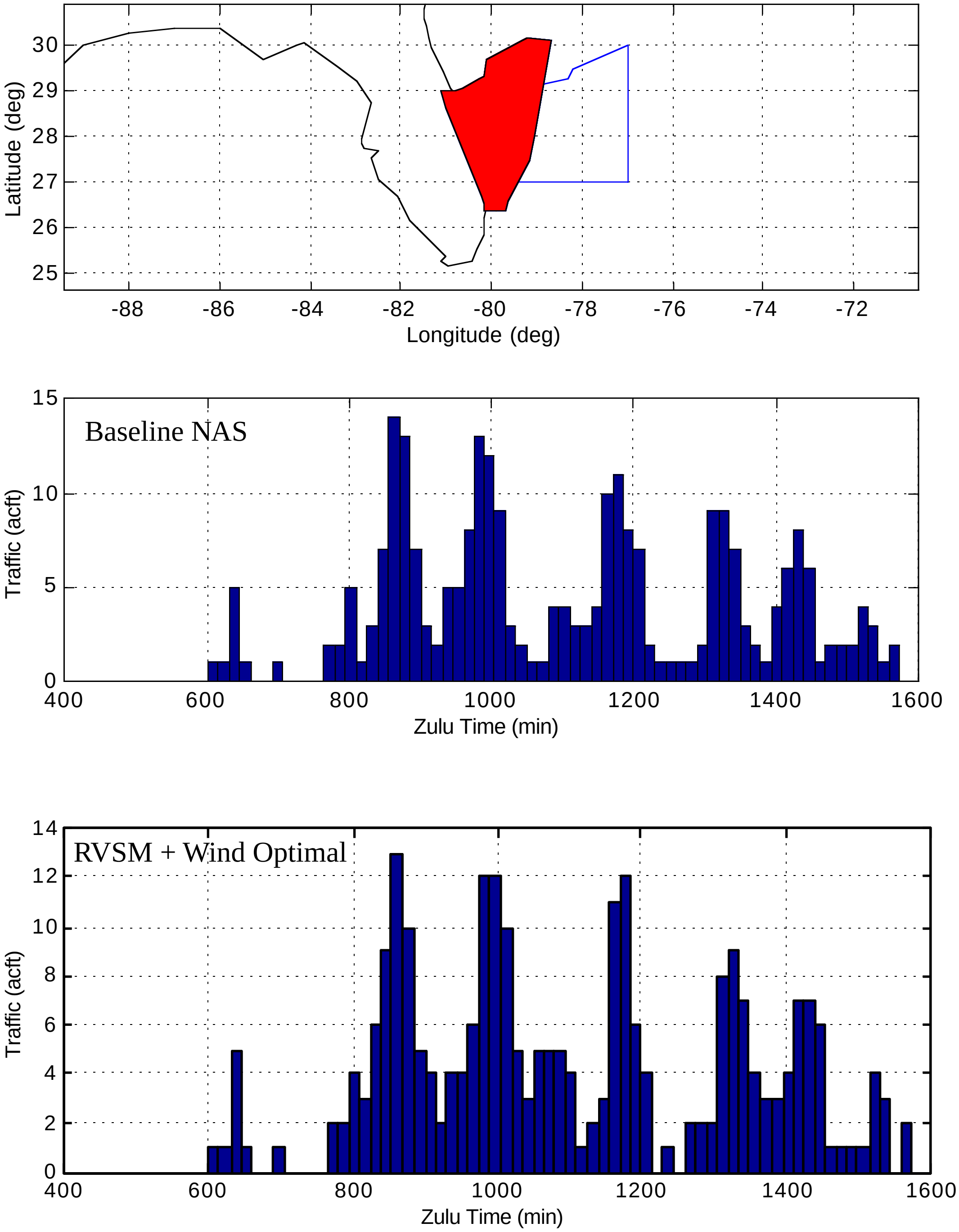


Figure B.2 AOM Sector Traffic Results (Sector 40 ZMA, 1996 NASTraffic Conditions).

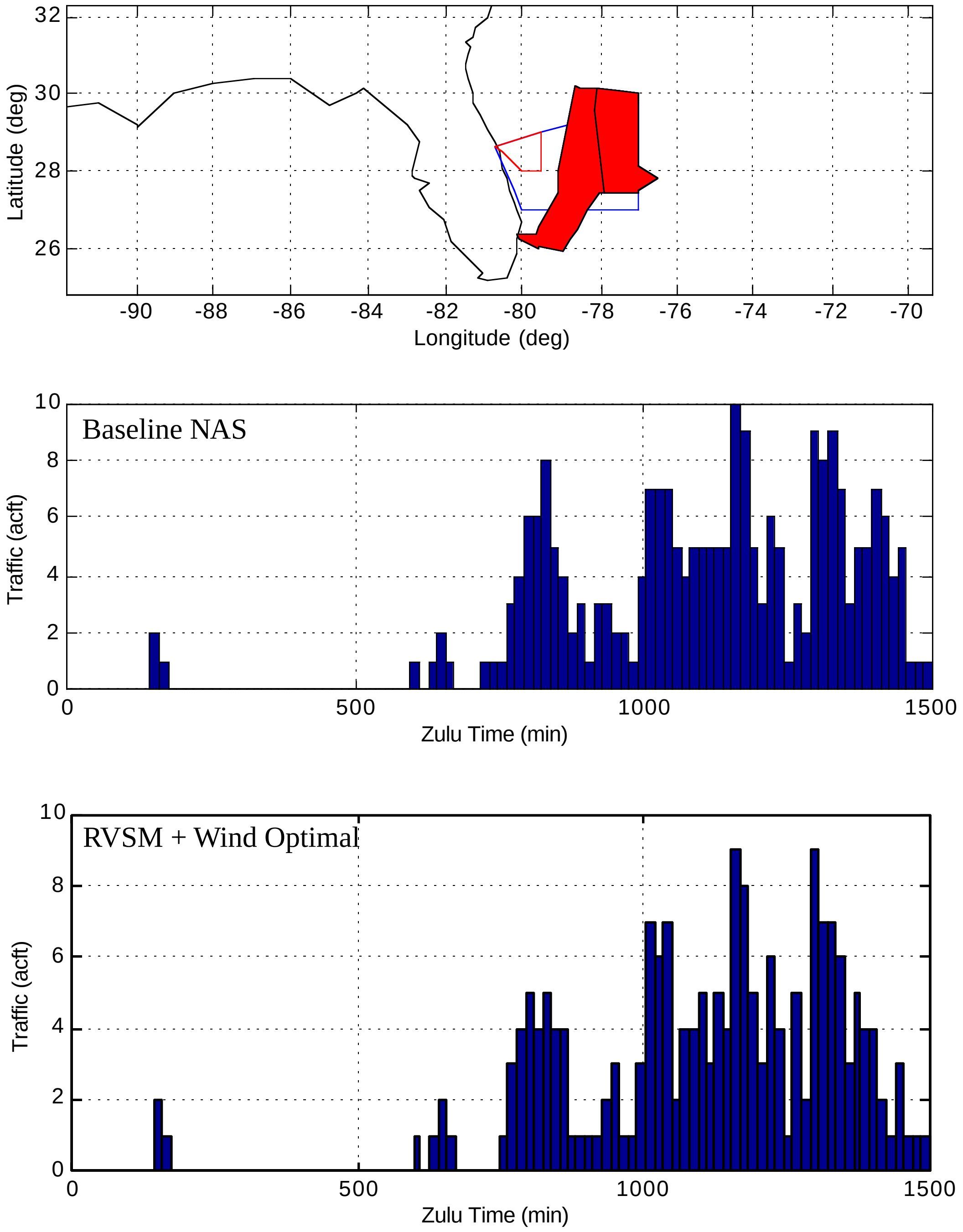


Figure B.3 AOM Sector Traffic Results (Sector 39 ZMA, 1996 NASTraffic Conditions).

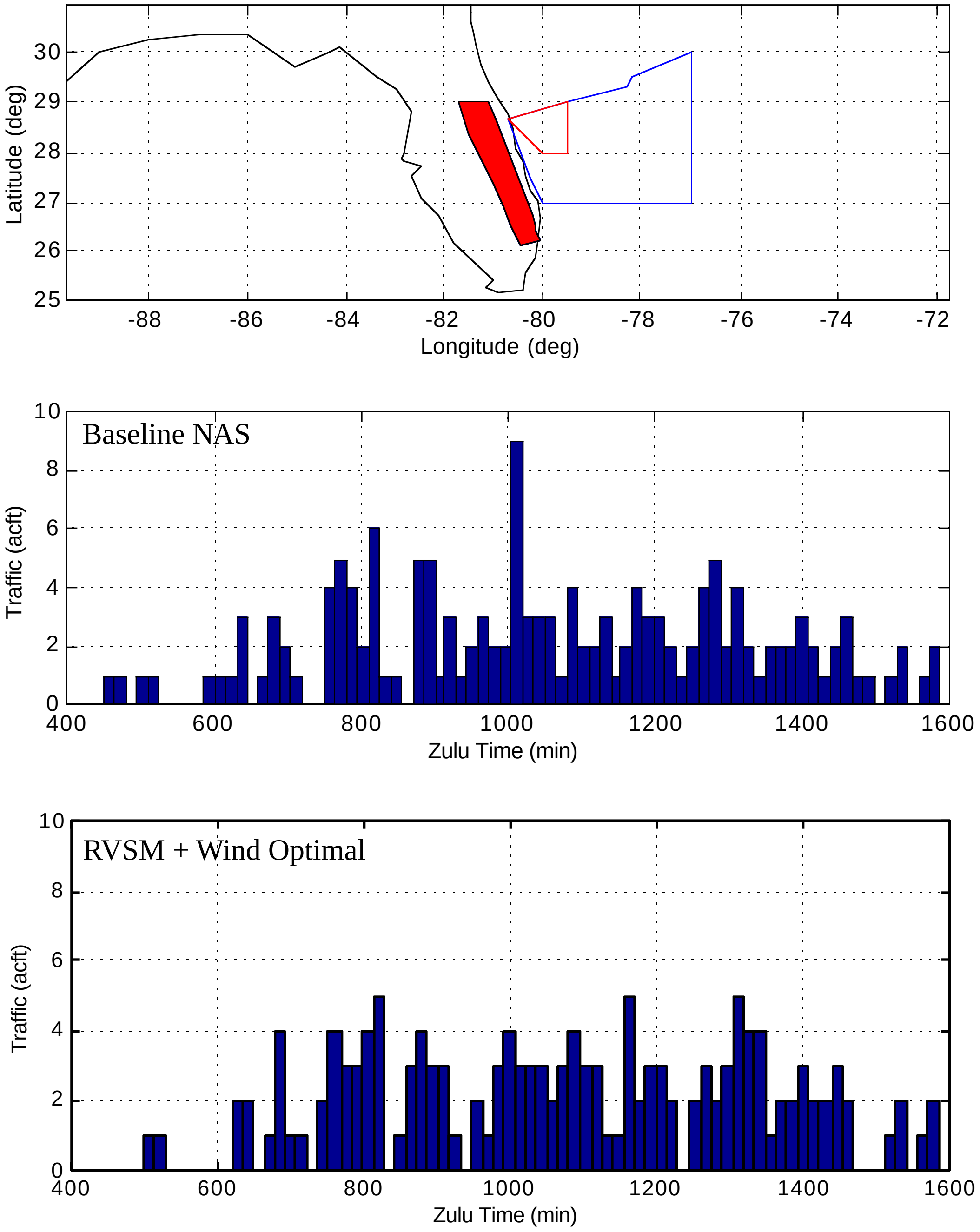


Figure B.4 AOM Sector Traffic Results (Sector 77 ZMA, 1996 NAS Traffic Conditions).

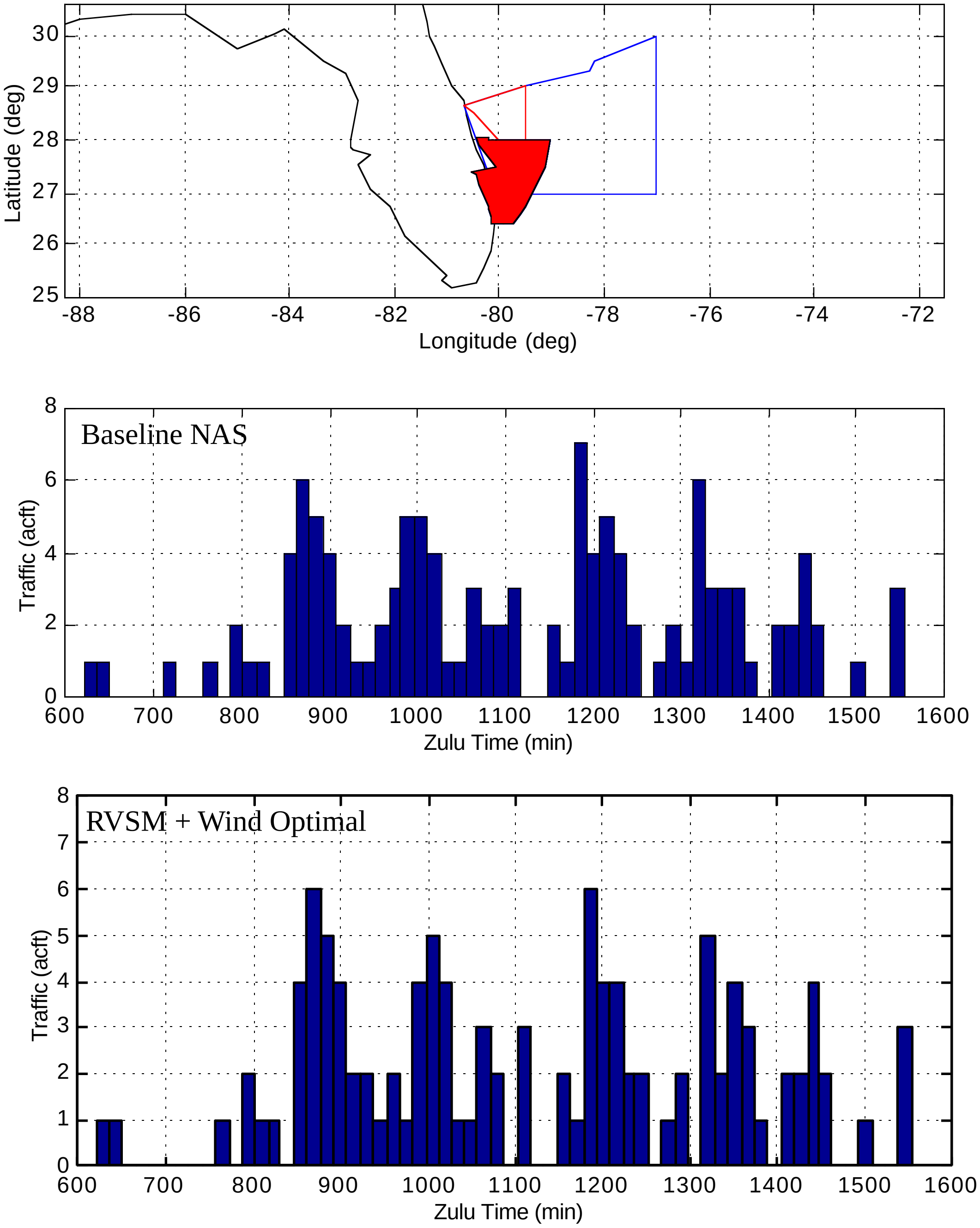


Figure B.5 AOM Sector Traffic Results (Sector 49 ZMA, 1996 NAS Traffic Conditions).

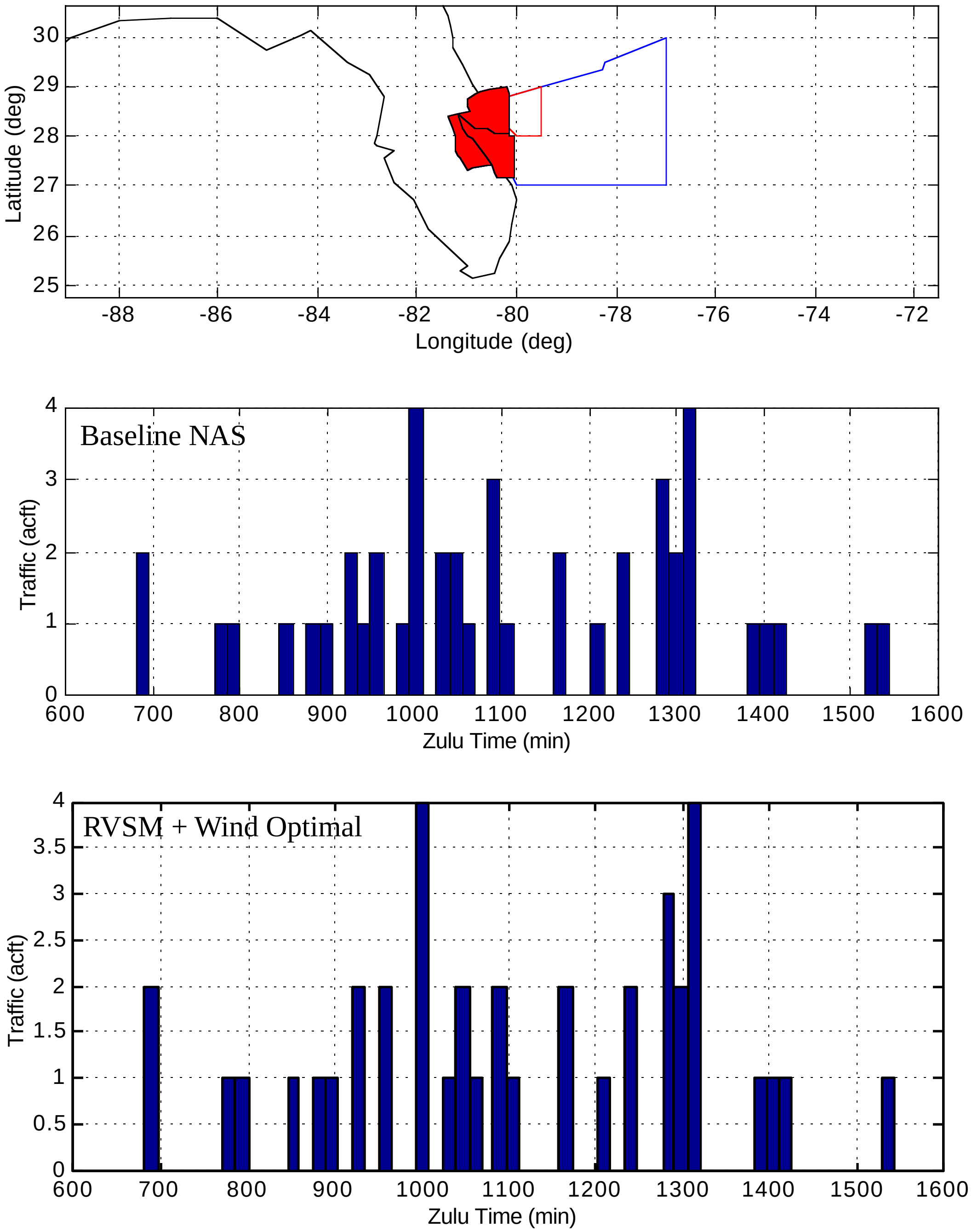


Figure B.6 AOM Sector Traffic Results (Sector 51 ZMA, 1996 NAS Traffic Conditions).

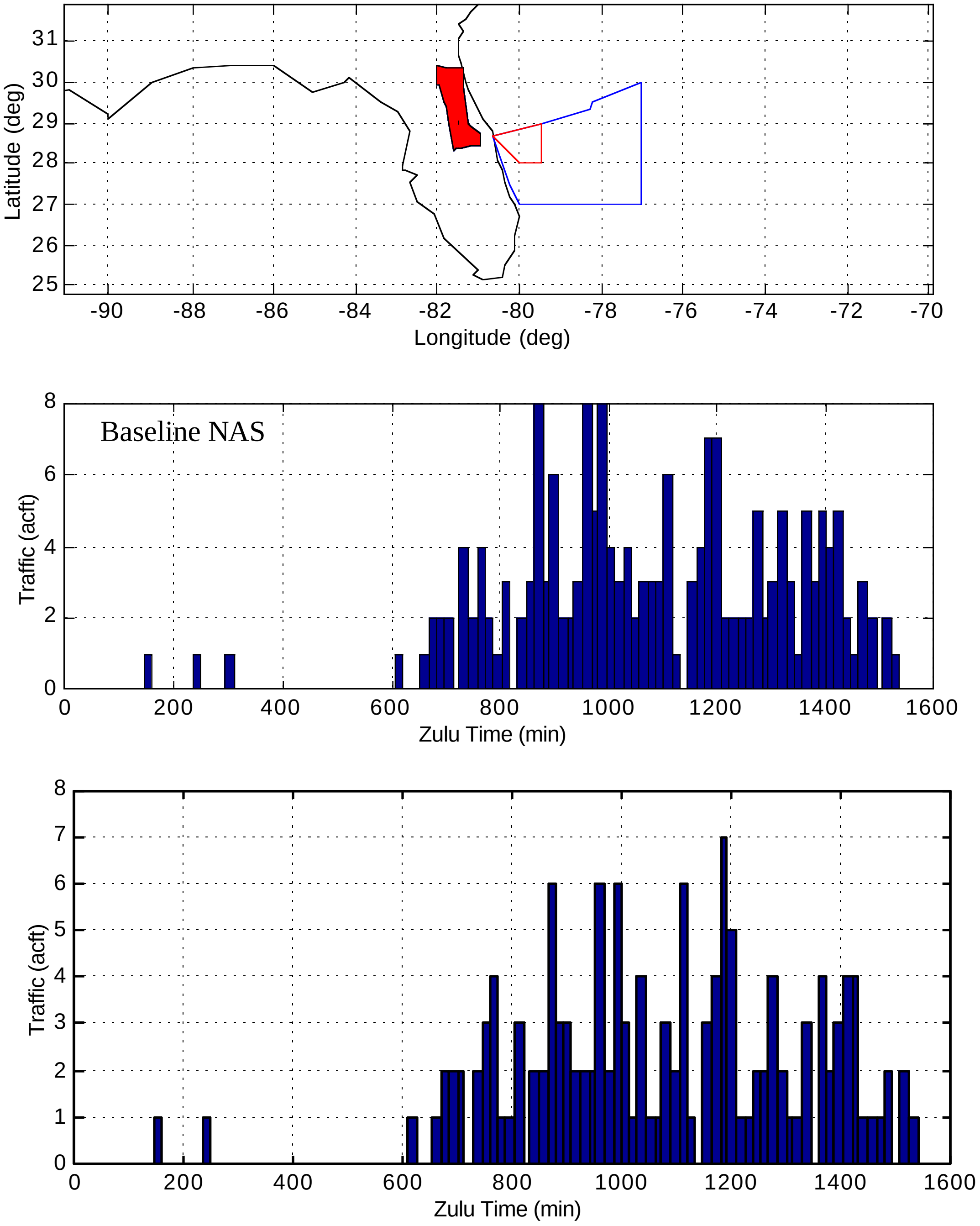


Figure B.7 AOM Sector Traffic Results (Sector 23 ZJX, 1996 NAS Traffic Conditions).

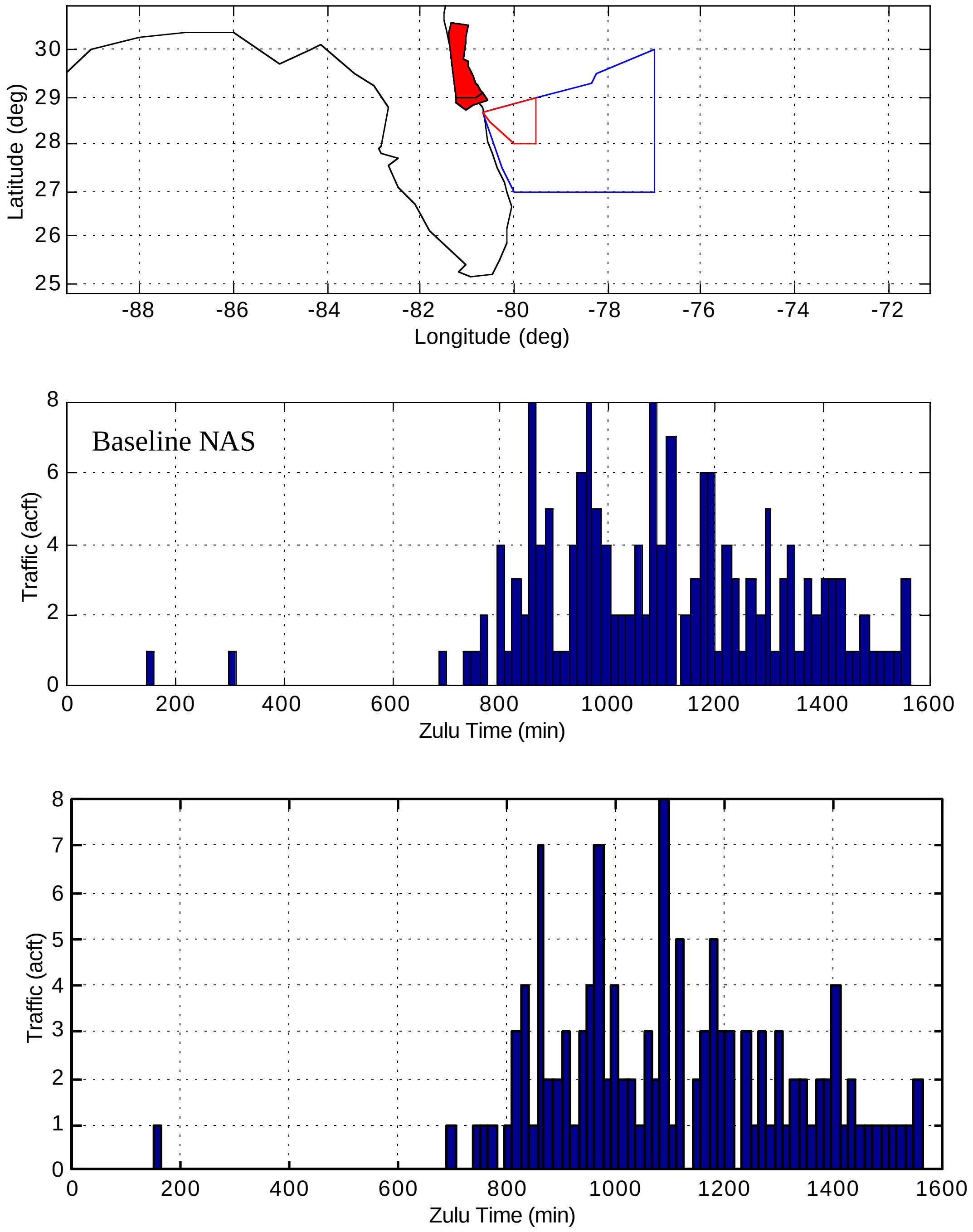


Figure B.8 AOM Sector Traffic Results (Sector 24 ZJX, 1996 NAS Traffic Conditions).

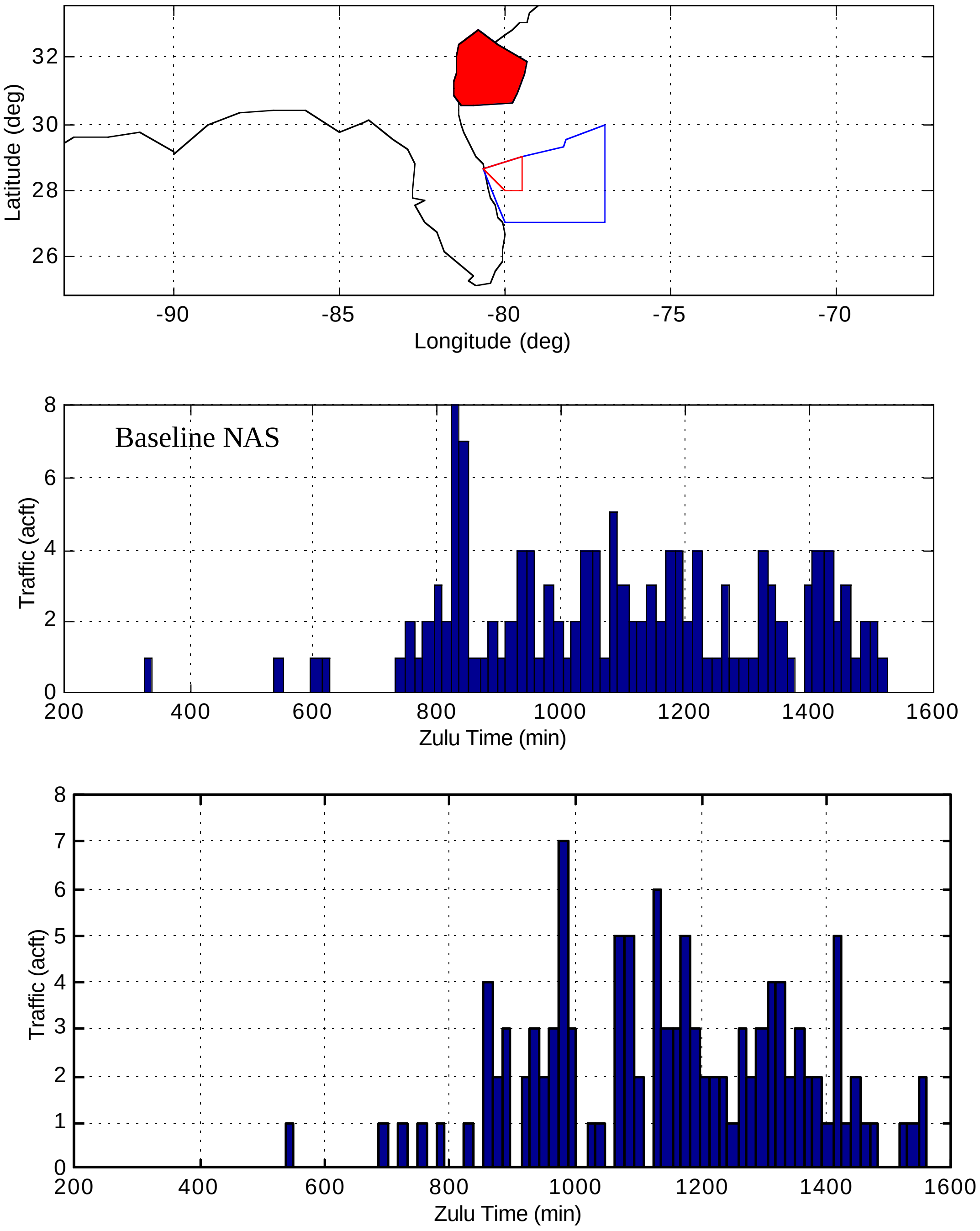


Figure B.9 AOM Sector Traffic Results (Sector 27 ZJX, 1996 Traffic Conditions).

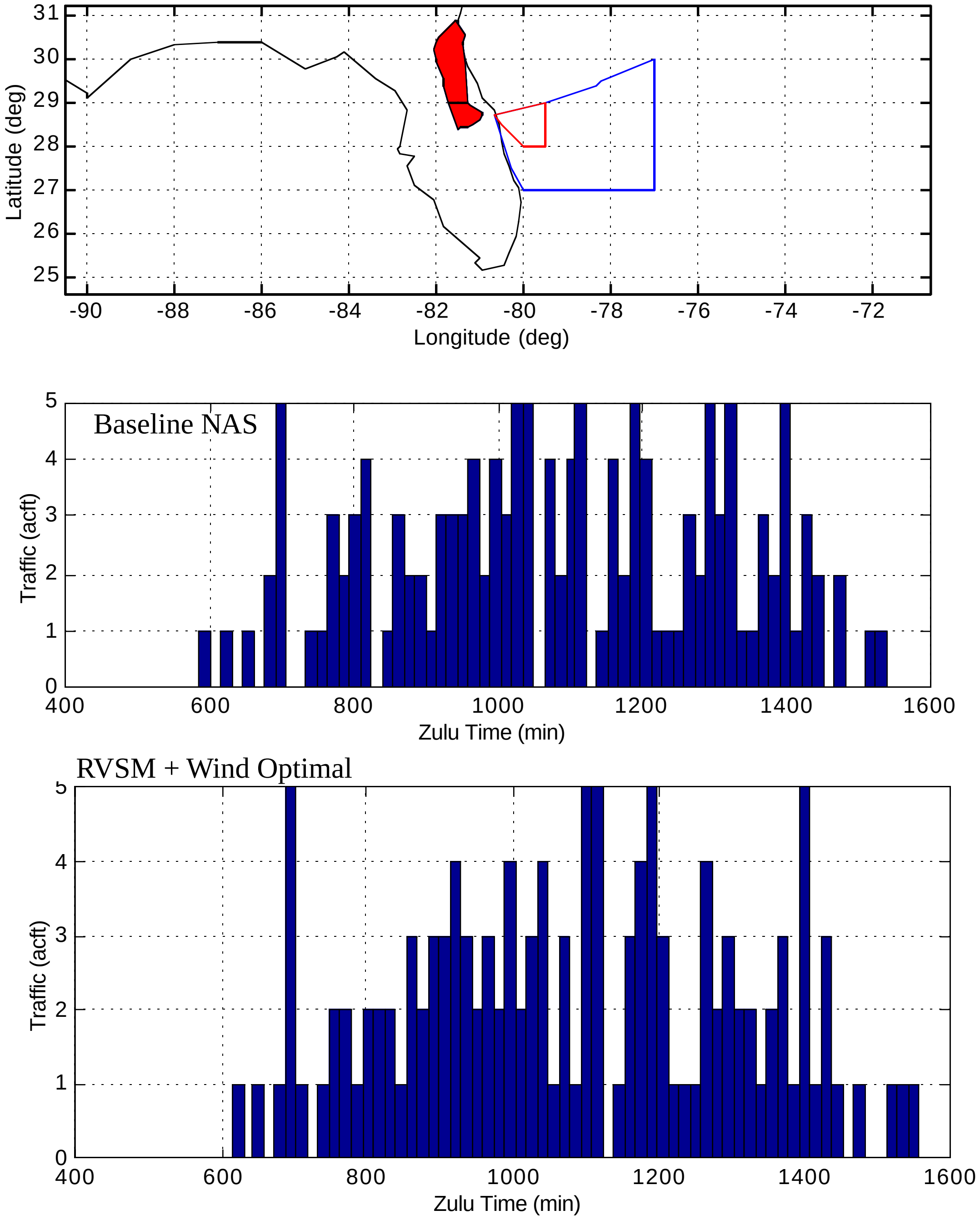


Figure B.10 AOM Sector Traffic Results (Sector 34 ZJX, 1996 NAS Traffic Conditions).