Evolving complexity towards risk : a massive scenario generation approach for evaluating advanced air traffic management concepts

Author:
Alam, Sameer

Publication Date:
2008

DOI:
https://doi.org/10.26190/unsworks/18249

Evolving Complexity towards Risk: A Massive Scenario Generation Approach for Evaluating Advanced Air Traffic Management Concepts

Sameer Alam

M.Tech. (Computer Sc.) Birla Institute of Technology, India

M.A. (Economics) University of Rajasthan, India

B.Sc. (Maths) University of Rajasthan, India

A thesis submitted in partial fulfillment of the requirements

for the degree of Doctor of Philosophy at the

School of Information Technology & Electrical Engineering

University of New South Wales

Australian Defence Force Academy

Abstract

Present day air traffic control is reaching its operational limits and accommodating future traffic growth will be a challenging task for air traffic service providers and airline operators. Free Flight is a proposed transition from a highly-structured and centrally-controlled air traffic system to a self-optimized and highly-distributed system. In Free Flight, pilots will have the flexibility of real-time trajectory planning and dynamic route optimization given airspace constraints (traffic, weather etc.). A variety of advanced air traffic management (ATM) concepts are proposed as enabling technologies for the realization of Free Flight. Since these concepts can be exposed to unforeseen and challenging scenarios in Free Flight, they need to be validated and evaluated in order to implement the most effective systems in the field.

Evaluation of advanced ATM concepts is a challenging task due to the limitations in the existing scenario generation methodologies and limited availability of a common platform (air traffic simulator) where diverse ATM concepts can be modeled and evaluated. Their rigorous evaluation on safety metrics, in a variety of complex scenarios, can provide an insight into their performance, which can help improve upon them while developing new ones.

In this thesis, I propose a non-propriety, non-commercial air traffic simulation system, with a novel representation of airspace, which can prototype advanced ATM concepts such as conflict detection and resolution, airborne weather avoidance and cockpit display of traffic information. I then propose a novel evolutionary computation methodology to algorithmically generate a massive number of conflict scenarios of increasing complexity in order to evaluate conflict detection algorithms.

I illustrate the methodology in detail by quantitative evaluation of three conflict detection algorithms, from the literature, on safety metrics. I then propose the use of data mining

techniques for the discovery of interesting relationships, that may exist implicitly, in the algorithm's performance data. The data mining techniques formulate the conflict characteristics, which may lead to algorithm failure, using if-then rules.

Using the rule sets for each algorithm, I propose an ensemble of conflict detection algorithms which uses a switch mechanism to direct the subsequent conflict probes to an algorithm which is less vulnerable to failure in a given conflict scenario. The objective is to form a predictive model for algorithm's vulnerability which can then be included in an ensemble that can minimize the overall vulnerability of the system.

In summary, the contributions of this thesis are:

1. A non-propriety, non-commercial air traffic simulation system with a novel representation of airspace for efficient modeling of advanced ATM concepts.

2. An Ant-based dynamic weather avoidance algorithm for traffic-constrained enroute airspace.

3. A novel representation of 4D air traffic scenario that allows the use of an evolutionary computation methodology to evolve complex conflict scenarios for the evaluation of conflict detection algorithms.

4. An evaluation framework where scenario generation, scenario evaluation and scenario evolution processes can be carried out in an integrated manner for rigorous evaluation of advanced ATM concepts.

5. A methodology for forming an intelligent ensemble of conflict detection algorithms by data mining the scenario space.

keywords

Free Flight, Air Traffic Management, Scenario Planning, Conflict Detection Algorithms, Learning Classifier Systems, Air Traffic Simulation, Weather Avoidance, Genetic Algorithms

Acknowledgement

No words of gratitude are enough to thank my supervisor Prof. Hussein A. Abbass, whose untiring support, motivation and supervision made this thesis, see the light of the day.

I also wish to acknowledge my co-supervisors, Dr. Michael Barlow (UNSW@ADFA) and Prof. Peter Lindsay (UQ), for their guidance and support throughout my thesis. I owe my sincere thanks to my colleagues at the Artificial Life & Adaptive Robotics lab, Kamran Shafi, Dr. Minh Ha Nguyen, Dr. Lam T. Bui and Jiangjun Tang with whom I enjoyed my research collaboration.

Thanks are due to Dr. Mohammed Juman (CAA, Bahrain), Dr. Praveen Dhyani (BITIC, UAE), Dr. Jimmy Krozel (Metron Aviation, US) and Dr. Vu Doung (Eurocontrol, FR) for their comments, motivation and valuable discussions. Thanks are also due to Bob Peake, Dr. Robert Porteous, Nick King and Greg McDonald of Air Services Australia for providing air traffic data and valuable discussions. I would like to extend my hearty appreciation to the anonymous reviewers for my journal and conference papers whose valuable suggestions were instrumental in improving the research work.

I would like to acknowledge, with gratitude and appreciation, financial support from the Australian Research Council's Center for Complex System and the University of New South Wales at the Australian Defense Force Academy, Canberra.

Finally, to my parents and my wife Hina, for their constant encouragement, support and love during this endeavor.

Sameer Alam

Australia, 2008

Certificate of Originality

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person, nor material which to a substantial extent has been accepted for the award of any other degree or diploma at UNSW or any other educational institution, except where due acknowledgement is made in the thesis. Any contribution made to the research by colleagues, with whom I have worked at UNSW or elsewhere, during my candidature, is fully acknowledged.

I also declare that the intellectual content of this thesis is the product of my own work, except to the extent that assistance from others in the project's design and conception or in style, presentation and linguistic expression is acknowledged.

Sameer Alam

Contents

List of Figures

List of Tables

List of Acronyms

ATM	Air Traffic Management
ACO	Ant Colony Optimization
ADS-B	Automatic Dependent Surveillance - Broadcast
AFTN	Aeronautical Fixed Telecommunication Network
ASA	Airborne Separation Assurance
ATCs	Air Traffic Controller/s
ATOMS	Air Traffic Operations & Management Simulator
BADA	Base of Aircraft Data
CD&R	Conflict Detection and Resolution
CNS	Communication Navigation and Surveillance
CPA	Closest Point of Approach
CPDLC	Controller-Pilot Data Link Communications
EC	Evolutionary Computation
ETA	Estimated Time of Arrival
ETD	Estimated Time of Departure
FAA	Federal Aviation Adminstration
FIR	Flight Information Region
FMS	Flight Management System
GA	Genetic Algorithms
GCR	Great Circle Route
ICAO	International Civil Aviation Organization
ISA	International Standard Atmospheric
IFR	Instrument Flying Rules
RNP	Required Navigation Performance
ROAD	Rate of Accelerate and De-accelerate
ROCD	Rate of Climb Descent
ROHC	Rate of Heading Change
RTCA	Radio Technical Commission on Avionics
SID	Standard Instrument Departure Procedure

List of Publications

Peer-reviewed publications arising from research work conducted in this thesis are listed chronologically below (lates to earliest):

Journal Publications

1. Alam, S., Shafi, K., Abbass, H. A., and Barlow, M. Forming an Intelligent Ensemble of Conflict Detection Algorithms in Free Flight by Data Mining the Scenario Space, Journal of Transportation Research Part-C: Emerging Technologies, Elsevier Science, Conditionally accepted : 12/09/2008

2. Alam, S., Abbass, H. A., and Barlow, M. Air Traffic Operations and Management Simulator: ATOMS, IEEE Transactions on Intelligent Transportation Systems, vol 9 (2), pp. 209-225, ISSN: 1524-9050, 2008

Conference Publications

1. Alam, S., Shafi, K., Abbass, H. A. and Barlow, M. Evolving Air Traffic Scenarios for the Evaluation of Conflict Detection Models, 6th Eurocontrol Innovative Research Workshop and Exhibition, pp. 237-245, Eurocontrol Experimental Center, Paris, France, Dec, 2007

2. Ngugen, M. H., Alam, S. and Abbass, H. A. Dynamic Weather Avoidance in a Traffic Constrained Enroute Airspace, 6th Eurocontrol Innovative Research Workshop and Exhibition, pp. 205-212, Eurocontrol Experimental Center, Paris, France, Dec, 2007

3. Alam, S., Ngugen, M. H., Abbass, H. A. and Barlow, M. Ants Guide Future Pilots, 3rd Australian Conference on Artificial Life (ACAL07), LNAI-4828, Springer-Verlag Berlin Heidelberg, pp. 36-48, Gold Coast, Australia, Dec, 2007

4. Alam, S., Ngugen, M. H., Abbass, H. A. and Barlow, M. On Architecture, Design and Validation of an Air Traffic Simulator, SIMTecT-07, Brisbane, Australia, Jun, 2007

5. Alam, S., Lam, T. B., Abbass, H. A. and Barlow, M. Pareto Meta-Heuristics for Generating Safe Flight Trajectories Under Weather Hazards, 6th International Conference on Simulated Evolution and Learning (SEAL-06), LNCS-4247, Springer Berlin Heidelberg, pp. 829-836, Hefei, China, Oct, 2006

6. Alam, S., Abbass, H. A., Barlow, M. and Lindsay, P. Mapping Lessons from Ants to Free Flight: An Ants based Weather Avoidance Algorithm for Free Flight, SPIE-06, Complex Systems, Vol. 6039, pp. 205-213, Brisbane, Australia, Jan, 2006

7. Alam, S., McPartland, M., Abbass, H. A., Barlow, M. and Lindsay, P. Neuro-Evolution for Conflict Detection & Resolution in a 2D Environment, 2nd Australian Conference on Artificial Life (ACAL05), Recent Advances in Artificial Life, vol. 3, World Scientific Publishing, pp. 13-28, Sydney, Australia, Dec, 2005

Chapter 1

Introduction

"Heavier-than-air flying machines are impossible". (Lord Kelvin, British Mathematician and Physicist, President of the British Royal Society, 1895)

1.1 Overview

Air transport continues to change and expand both in volume and in the areas of the world it serves. While recognizing that the current systems and procedures of air traffic system have served the international civil aviation successfully and safely for the past 60 years, ICAO [1] felt that these systems and procedures are reaching their limits, as evidenced by the currently experienced delays at major airports and in congested airspaces during peak hours (ICAO, 2002). Fundamental changes in the present day air traffic control system are required to accommodate future air traffic growth. Free Flight (RTCA, 1995) is one such concept which is being actively pursued by industry, government and research organizations. In Free Flight, pilots will have the flexibility to choose their trajectory in real time and to modify the flight plan without an approval from the air traffic controllers (ATCs). The concept changes the air traffic control structure in such a fundamental way that one can think of it as a paradigm shift (Erzberger, 2004).

Automation is central to Free Flight (Eurocontrol, 2005) and several enabling technologies known as advanced air traffic management (ATM) concepts are proposed for its realization (Celio *et al.*, 2000). To achieve this, major challenges in the design, develop-

[1] International Civil Aviation Organization: ICAO was established in 1947 to develop principles and techniques of international air navigation and to support planning and development of international flight transportation

ment and evaluation of reliable and complex safety critical software systems need to be overcome (Krozel, 2000). Advanced ATM concepts, envisioned in Free Flight, must be investigated before they are operationally implemented, without which it will be a risky and expensive exercise (Bilimoria *et al.*, 2001). Evaluation of advanced ATM concepts have now become a key area of research in ATM and computer simulation models have become a preferred choice to prototype and evaluate them (Andrews *et al.*, 2006).

Since these concepts will eventually be deployed in an operational environment they need to be evaluated on field data. Air traffic scenarios thus form a core component in the evaluation process (Harvey *et al.*, 2003) and are of high importance in providing a robust feedback on whether a concept can be useful in the operational environment. However, the proprietary nature of field data and limitations in the existing scenario generation methodologies limit their rigorous evaluation (Signor *et al.*, 2004).

A key technological consideration for evaluating advanced ATM concepts is the requirement for a high fidelity air traffic simulation environment in order to prototype the concepts. Another vital technological issue involves the generation of complex air traffic scenarios. Moreover, different methodologies employed by advanced ATM concepts pose a challenge in their performance evaluation (Kopardekar *et al.*, 2002).

This thesis addresses the development of a methodology for evolving air traffic conflict scenarios to evaluate conflict detection (CD) algorithms and to gain an understanding of their performance under failure conditions. The objective is to apply this understanding to improve upon the overall conflict detection performance by forming an intelligent ensemble of algorithms. Further, this thesis also looks into the design and development of a Free Flight air traffic simulation environment with an efficient airspace representation for reducing the search space during the CD process.

In particular, we are interested in how conflict scenarios can be evolved in-silico towards a sufficient level of realistic complexity to evaluate advanced ATM concepts. We believe that the resultant methodology will be a benchmark methodology for evaluation of other advanced ATM concepts using scenario evolution.

1.2 Motivation

Conflict detection is a key requirement for Free Flight operations (Bilimoria, 2001). Timely and accurate detection of potential conflicts in Free Flight will not only help the flight crew to take early decisions but will also reduce the risk of loss of separation due to missed detects (MD) and unnecessary maneuver due to false alarms (FA).

A variety of conflict detection and resolution (CD&R) algorithms are proposed in the literature and detailed surveys of them can be found in (Kuchar and Yang, 2000; Watkins and Lygeros, 2002; Kopardekar *et al.*, 2002). Some of these algorithms are tested in simulation environments (Hoekstra, 2001; Eby and Kelly, 1999; Prandini *et al.*, 2000) and a few in operational environments (McNally *et al.*, 1999; Gool and Schrter, 1999). Most algorithms are evaluated on an individual level by using safety, efficiency and stability as metrics. These algorithms employ different methodologies for detecting conflicts which are likely to produce differences in their detection strategies and effectiveness. However, there is no detailed competitive analysis that demonstrates their efficiency and effectiveness (Kopardekar *et al.*, 2002). It is also found that most of the existing algorithms are inadequate for handling complex conflict scenarios, and their performance under non-nominal circumstances is unknown (Kuchar and Yang, 2000).

These shortcomings are primarily due to the lack of availability of a common platform where diverse CD algorithms can be modeled and evaluated on common traffic samples and secondly, due to limitations in current scenario generation methodologies, they are not evaluated rigorously for a wide range of conflict scenarios.

The distributed nature of Free Flight and the infinite number of possible conflict geometries makes it hard to estimate the actual safety level achieved by CD algorithms (Hoekstra, 2001). The evaluation exercises fall short of answering the question ***when will an algorithm fail in detecting a conflict in Free Flight?***

Given the inherent uncertainties in the air traffic environment no CD algorithm can guarantee a 100% success rate. This makes the choice of which CD algorithm to be placed in the cockpit a challenging task for the designers of future ATM systems.

Our motivation for this thesis stems from the fact that a better understanding of the performance of CD algorithms in a variety of complex conflict scenarios can pave the way for better design and improved performance of CD algorithms in Free Flight.

1.3 Research Questions and Hypothesis

CD algorithms can be exposed to diverse operational modes and conflict conditions in Free Flight. As "*the proof of the pudding lies in the eating*", how good a CD algorithm is, depends on how correctly it identifies a potential conflict. Missing a potential conflict will lead to a MD and an eventual loss of separation, while wrongly identifying a conflict will lead to a FA. However, this depends upon the type of conflict scenario a CD algorithm is exposed to. For example, a head-on conflict with both the aircraft flying at the same altitude might be easy to detect than an in-trail conflict with two aircraft in transition mode (climb/descent), which is at the boundary of separation standards (ICAO, 1996). It is a complex interplay of conflict characteristics between two or more aircraft that determine the effectiveness of a CD algorithm.

To evaluate CD algorithms, a variety of air traffic scenarios can be generated using recorded field data but the pre-scripted nature of conflict events (if any) makes it a less effective approach. Conflict events should ideally unfold dynamically as the scenario progresses and move to the next step from simple to complex using some kind of feedback from the evaluation process. This will lead to a more directed and incremental evaluation of CD algorithm's performance and also to an understanding of their performance in complex conflict scenarios.

In this thesis, we wish to specifically answer the following research question:

Is it possible to evolve complex air traffic scenarios in-silico for the evaluation of advanced ATM concepts?

Our hypothesis is that an evolutionary scenario planning methodology will be suitable for assessing and evaluating advanced ATM concepts, specifically CD algorithms. Hence, our main research objective is to prove or disprove that evolutionary computation techniques can evolve complex air traffic scenarios; or in other words that the evolutionary process can drive an algorithm towards the problem search space where it is vulnerable.

To address this question we will be evaluating CD algorithms, on quantitative performance metrics, for a large number of computer generated scenarios. We will then use the efficient set of solutions generated by the CD algorithms as a guiding mechanism for search in the evolutionary process of scenario generation.

In order to answer the main research question, a number of other related sub-questions will need to be investigated as well:

1. **What is an appropriate methodology to represent airspace in Free Flight for efficient modeling of advanced ATM Concepts?**

   Rigorous evaluation of advanced ATM concepts requires a fast time simulation environment with repetition and replication of hundreds of air traffic scenarios. A search process for detecting conflict with N aircraft in airspace will increase exponentially in space and time as the number of flights increases. Therefore, an efficient data structure and an appropriate search mechanism may lead to computationally feasible evaluation of advanced ATM concepts.

2. **What is an appropriate methodology to generate conflict scenarios algorithmically such that they meet desired conflict parameters?**

   Recorded air traffic data does not contain conflicts and other non-nominal scenarios since any such situations are already resolved by ATCs. The rare recorded air traffic accidents are in the order of 10^{-10} per flight hour (ICAO, 1998) making it almost impossible for the existing fast time simulators to assess such rare events in recorded air traffic data. A methodology to generate conflict events with desired conflict parameters without the need for field data will not only help in maintaining a common traffic sample for comparative evaluation purposes but may also be a step towards standardization in air traffic scenario planning.

3. **What is an appropriate methodology to evolve complex conflict scenarios?**

   In ATM concept validation, it is a common practice, for scenario generation, to focus on pre-scripted scenarios where the designer pre-selects the parameters (mainly conflicts) and then generates scenarios (Harvey *et al.*, 2003), which are then fed into the main simulator for execution. These two processes are usually separate with no feedback mechanism between the two. Moreover, such scenarios are static in nature, which limits the rigorous evaluation of any advanced ATM concept. It is desirable to have an integrated methodology where scenarios can be dynamically generated with events unfolding as scenario progresses by taking feedback from the evaluation process.

4. **How can we learn from the evolutionary process in order to overcome the shortcomings in the performance of CD algorithms?**

It is important to understand the failure patterns in a CD algorithm and the associated conflict characteristics when they occur. We would like to anticipate a relationship between the algorithm's failures and the corresponding conflict parameters, and use that relationship as a guide to direct future conflict probes. Given the large amount of conflict probes, it is difficult to perform this analysis manually to identify patterns of interest. Data mining techniques may provide an answer to this question. The objective is to implicitly learn a generalized concept from the given data which can then be applied to future cases with high predictability.

1.4 Organization of the Thesis

This thesis has eight chapters and is organized as follows:
In Chapter 1, an introduction to the thesis is presented. It first provides an overview of the research field, followed by the motivation and research questions raised in the thesis. An outline of the thesis is then given and the chapter concludes with a list of scientific contributions stemming from this research work.

In Chapter 2, a background is provided for research conducted on Free Flight ATM. First, a background to Free Flight is provided and the need for an air traffic simulation environment to evaluate Free Flight concepts is emphasized. CD methodologies for Free Flight are surveyed along with some evaluation approaches and performance measures. Emergent questions arising from the literature survey are then presented.

In Chapter 3, the design and development of a Free Flight air traffic simulation environment is explained along with its validation. Firstly, a description of the air traffic simulator architecture and airspace model is given, followed by the geographic reference used and the trajectory generation model. Atmospheric and wind grid model implementations are discussed next. Then the computation of aircraft's performance characteristics are provided, followed by navigation and guidance. The implementation of a Flight Management System (FMS) and the autopilot are discussed next followed by the modeling of the approach and landing phases in the simulator. The fuel flow methodology is discussed next and finally the graphical interface of the simulator along with the simulated ATC communication interface is presented.

In Chapter 4, the data structure design for modeling Free Flight airspace is presented along with the modeling of two advanced ATM concepts: CD&R and airborne weather avoidance. The advantages of the airspace design in reducing the search space, for the two ATM concepts that are being modeled, are discussed. Preliminary experiments with a variety of CD&R algorithms with simplistic conflict scenarios are presented and evaluated on safety and economic indicators. A meta-heuristic algorithm is developed, using Ant Colony Optimization and A-Star, for airborne weather avoidance in a traffic constrained airspace. The weather avoidance trajectories are validated on aircraft's safety parameters in the simulation environment. A prototype for cockpit display of traffic information is also developed and presented as an aid for airborne conflict resolution and weather avoidance.

In Chapter 5, we present an algorithmic approach for generating air traffic scenarios by using conflict parameters at the closest point of approach (CPA) between two aircraft. A methodology is developed to generate flight plans of many pairs of aircraft with desired conflict parameters. The proposed methodology for conflict scenario generation is then validated for a variety of conflict parameters such as conflict geometry, horizontal and vertical separations.

In Chapter 6, we break away from the traditional methods of pre-scripting conflict events in a scenario and explore the possibility of evolving complex conflict scenarios by using evolutionary computation techniques. The objective of the evolutionary process is to find the problem subspace where an algorithm is vulnerable. We present a novel mechanism of encoding conflict parameters into chromosomes and use algorithm evaluation metrics as the fitness functions to evolve them. Our approach is to play *Devil's Advocate* where complex conflict scenarios are deemed fitter individuals having increased likelihood of survival to the next generation in the evolutionary process of the Genetic Algorithm. These scenarios are fed into a fast time air traffic simulator for execution and evaluation the performance of CD algorithms. Three conflict detection algorithms are prototyped from the literature.

In Chapter 7, we use data mining techniques to anticipate a relationship between the failure patterns in a CD algorithm and associated conflict characteristics, and use it as a guide to direct future conflict probes. Using these relationships, we form a predictive model for algorithm's vulnerability which is then included in an ensemble that minimizes the overall vulnerability of the system. Pareto optimal curves are used for comparative analysis of the proposed ensemble approach vis-a-vis individual performance of the algorithms.

In Chapter 8, the main findings from this thesis are summarized. The chapter concludes the thesis with a discussion of possible future research directions.

1.5 Original Contributions

A list of the scientific contributions arising from this thesis is given below:

- A non-propriety, non-commercial air traffic simulation system with a novel representation of airspace is presented for efficient modeling of Free Flight concepts (Chapter 3). Although other air traffic simulation systems are present in the literature, no study to our knowledge has attempted to model Free Flight airspace in a manner that can reduce the search space to enable the directed evaluation of Free Flight concepts in a fast time simulation environment. This simulation environment may provide more insight into complex interaction of various ATM subsystems and pave the way for future investigations into Free Flight concepts.

- A meta-heuristic based weather avoidance algorithm is developed for traffic constrained enroute airspace (Chapter 4). As compared to previous approaches in weather avoidance, which focused on using a single objective for optimizing a weather free trajectory, a multi-objective optimization approach is used to offer flexibility of route choices in a constrained airspace. With more flexibility given to pilots in Free Flight, providing route choices in a constrained airspace may significantly improve decision making and help achieve flight and system level objectives.

- A mechanism for generating air traffic conflict scenarios and an evolutionary computation based methodology for evolving air traffic scenarios for the evaluation of CD algorithms are presented (Chapter 5 and Chapter 6). This approach illustrates that by evolving the problem landscape, a problem subspace can be found where an algorithm is vulnerable/prone to failure. The evaluation of advanced ATM concepts, specially conflict detection, has always been a challenging area given the distributed nature and possibility of an infinite number of conflict geometries in Free Flight. An entirely different perspective towards the evaluation of Free Flight concepts can be achieved by taking a scenario based approach. The evolution of air traffic scenarios towards more complexity can open up different avenues in the Free Flight concept development and evaluation. We show that scenario evolution can be carried out in a simple and practical manner using genetic algorithms (GAs). This highly accessible method of capturing scenario complexity may have significant implications not only within the scope of Free Flight concepts but across the much wider spectrum of the ATM research field.

- A methodology for forming an intelligent ensemble of CD algorithms by data mining the scenario space is presented (Chapter 7). Data mining techniques are employed to learn implicit patterns in the performance of the CD algorithms for a variety of increasingly complex scenarios, which are formulated as if-then rules. It is shown that these simple rules can be used to direct conflict probes to an algorithm which is less prone to failure in a given a conflict scenario. The advantage of this methodology is that, once the rule set is formulated, any number of CD algorithms can be added into the ensemble, making the CD process more robust. This approach also strikes a balance between various CD mechanisms by choosing the most appropriate CD algorithm in a given a conflict scenario.

Chapter 2

Background

2.1 The Present Day Air Traffic Control System

The present day Air Traffic Control (ATC) is a highly structured and centralized system which has evolved over the years with technological advances. In earlier times, pilots navigated the aircraft using ground based features such as highways and rivers. Maintaining separation was not a problem as the aircraft were slow and flew at a low altitude. To maintain a safe distance, pilots used "rules of the air" to decide which one has the right to way. With the advances in aircraft technologies and invention of radar, aircraft were able to fly higher and faster. This led to radio beacons being placed all over the countries, where aircraft can navigate from one beacon to another. This created virtual highways in the sky termed as airways and navigation points termed as waypoints. Modern aircraft use a combination of navigation equipment and no longer rely on ground based beacons, but the airway structure still remains. The task of maintaining safe separation between aircraft and managing the traffic flow now rest with ground based ATCs. This system, which is comprised of a variety of communication, navigation and surveillance (CNS) sub-systems, ensures safety through multiple redundancies and inefficient procedures. In this fixed route structure, the traffic is artificially increased by concentrating aircraft's movements on the airways with the cost of an increased potential of conflicts at crossing points. Human cognitive limitations also pose a bound on how many aircraft ATCs can safely manage at any particular point of time.

2.1.1 A Need for Improvement

With the continuing growth in air traffic worldwide there is a substantial increase in delays and congestion. One of the major challenges facing air traffic planners is how to increase airspace capacity and flight efficiency while still maintaining the current set of safety standards (Blom *et al.*, 2001). As pointed out by Erzberger:

"The next-generation air traffic control system must be designed to safely and efficiently accommodate the large growth of traffic expected in the near future" (Erzberger, 2004)

The majority of the U.S.A. and European airports are already operating at or above their operational capacity leading to considerable delays of more than 15 minutes (Donohue and Laska, 2000). In the U.S.A. and Australia, airport congestion is a major problem whereas in European airspace, due to its unique geographical features, enroute congestion is a major problem (Eurocontrol, 2007). The inadequate capacity in European airspace is reported so critical that a 1% increase in traffic typically results in a 7% increase in enroute delays (Eurocontrol, 2003b). The Eurocontrol Performance Review Report for 2006 states that air traffic delays are expected to increase further, as capacity plans do not match traffic growth. In a press bulletin released by Eurocontrol in July 2006, it is stated that from 2005 to 2006 the traffic growth was just 3.8%, but the average en-route delay per flight in minutes had increased by 16.9% (Eurocontrol, 2006). Some Civil Aviation Organizations estimated that if the current increase in air traffic and its accident rate continues, the industry could face a major accident every week by 2010 (Transport Canada, 2006). Therefore, an overall improvements in the ATC system is required to handle the delays and to accommodate future air traffic growth.

Several improvement efforts during the past decade have been undertaken worldwide to increase the capacity and efficiency of the air traffic system, in both ground based as well as airborne, within the constraints of the existing system. We review some of these briefly in the next section.

2.1.2 Improvement Efforts in ATC Systems

In ground based systems, research efforts are mainly aiding the ATCs, through the use of automation tools, to efficiently regulate the air traffic flow. The two most comprehensive projects in this direction are Center-TARCON Automated System (CTAS) by NASA Ames research center (Erzberger *et al.*, 1993) and the User Request Evaluation Tool (UERT) undertaken by MITRE Corp (MITRE, 2007). The CTAS system assists controllers in handling arriving aircraft by using automated tools for spacing, merging and arrival sequencing. UERT processes real-time flight plans and radar track data to predict trajectories to detect potential aircraft conflicts up to 20 minutes in future. It not only provides ATCs with strategic conflict alerts but also provides them with a “what if” scenario capability for determining the implications if the trajectory of an aircraft is modified.

In airborne systems, the research is mainly focused on the development and improvement of avionics namely the FMS and the autopilot. These advances enable the pilot to navigate an aircraft precisely on its desired flight path and provide improved traffic management by down linking the flight data to the ground based system (Lee *et al.*, 2003; Prevot *et al.*, 2005).

In Europe, a number of initiatives are underway under a common strategy for air-ground improvements known as Eurocontrol ATM 2000+ Strategy (Eurocontrol, 2003a). This strategy has four key operational areas which include Strategic Flow and Capacity Planning, Optimized Capacity Management, Tactical Flow and Capacity Management and Collaboration with ATM Partners. The aim of this strategy is to evolve a “Single Sky” ATM for Europe. One of the most successful implementation of this strategy is the European Air Traffic Control Harmonization and Integration Programme (EATCHIP) (Vink *et al.*, 1997), which has undertaken a series of technology based advancements for airborne communication, airborne separation assurance and air-ground integration projects.

In light of the above improvements, the ICAO has adopted the “Global Air Navigation Plan for CNS-ATM Systems” to develop a..

“seamless, globally coordinated system of air navigation services that will cope with worldwide growth in air traffic demand while improving upon the present levels of safety and improving upon the overall efficiency and capacity of airspace and airports.” (ICAO, 2002)

This concept requires the use of data link communication, satellite based navigation systems and use of automatic dependent surveillance broadcast system (ADS-B). However, due to the constraints in today’s air traffic systems in terms of rules and procedures and legacy systems, the current ATM system is not capable of exploiting these advancements.

ATM researchers soon realized that an overall improvement in air traffic systems is required to accommodate future air traffic growth. The posed restrictions by the route structure lead to less efficient routing, and benefits of advancements in CNS technology cannot be fully utilized. A radical departure from the present fixed route structure is now under investigation.

In the U.S.A., NASA under its Distributed Air/Ground Traffic Management (DAG-TM) program is evaluating the feasibility and the benefits of user preferred trajectories and self separation. Under DAG-TM

"Appropriately equipped aircraft accept the responsibility to maintain separation from other aircraft, while exercising the authority to freely maneuver in en route airspace in order to establish a new user-preferred trajectory that conforms to any active local traffic flow management (TFM) constraints." (DAG-TM, 2002)

In Europe, the Programme for Harmonized ATM Research in Eurocontrol (PHARE) was initiated as a collaborative research programme within Europe to investigate a future ATM concept (Gool and Schrter, 1999). The objective of PHARE was to organize, co-ordinate and conduct studies and experiments to demonstrate the feasibility and the benefits of a future air-ground integrated ATM system in all phases of flight.

In Australia, AirServices Australia is actively pursuing its Flexitrack program under Australian Organized Track Structure (AUSOTS) (Air Service Australia, 2006) where wind and traffic optimized trajectories are generated on a daily basis such that they enhance air traffic control separation assurance. It simultaneously allows airlines to obtain benefits from forecast wind conditions that cannot be obtained using the current fixed route structure.

2.2 The Free Flight Concept

In 1995, the Radio Technical Commission on Avionics (RTCA), which constitutes airline operators, ATCs, government and aviation research companies, proposed the Free Flight concept as the means to increase the capacity of air transport. Free Flight is formally defined (RTCA, 1995):

"a safe and efficient flight operation capability under IFR (Instrument Flying Rules) in which the operators have the freedom to select their speed and path in real time. Air traffic restrictions are only imposed to ensure separation, to preclude existing airport capacity, to prevent unauthorized flight through SUA (Special Use Airspace), and to ensure safety of flight Restriction are limited in extent and duration to correct the identified flight. Any activity which removes restriction represents a move towards Free Flight."

The RTCA definition highlights the fact that the major ATC constraint is the safe separation between the aircraft. However, the report fails to answer several key questions such as how the transition to Free Flight will happen and what is the feasibility of Free Flight.
After more than a decade into active Free Flight research, its feasibility and implications, three broad research areas have emerged (Krozel, 2000):

- CD&R algorithms, intent information propagation, resolution coordination, and system implementation (Kuchar and Yang, 2000; Krozel *et al.*, 2001; Duong and Hoffman, 1997; Bilimoria *et al.*, 2000).
- Risk Assessment and safety analysis of Free Flight concepts (Prandini and Watkins, 2005; Blom *et al.*, 2003; Brooker, 2002).
- Pilots and Controllers Workload Issues (Kopardekar and Magyarits, 2002; Alexander and Wickens, 2001; Johnson *et al.*, 1999).

Various studies in these research areas have reaffirmed the feasibility of Free Flight (Hoekstra *et al.*, 2002; Warren *et al.*, 1997; Celio *et al.*, 2000; Krozel, 2000; Ratcliffe, 2001). As noted by Krozel:

"Free Flight will enable greater traffic volume and complexity by distributing some of the functionality, including conflict detection and resolution, to airborne systems (including pilots)". (Krozel, 2000)

The European Commissioner for Aviation Research foresees that by 2020, Europe will manage to

"create a seamless system of air traffic management that copes with up to three times more aircraft movements than today. The development of Free Flight has made this possible". (Lumsden, 2001)

Free Flight may provides long term resolution benefits in terms of increased capacity and flexibility. By providing for more efficient routes, Free Flight will reduce users' operating costs. Free Flight will allow the user's aircraft to reach its destination at the prescribed time. These improvements will result in air quality benefits through reductions in fuel burn (Carlier *et al.*, 2004). Free Flight will also enable ATCs to accommodate future air traffic growth through decision support systems (Krozel, 2000).

However, safety analysis of Free Flight Concepts is one challenging open question to address. As pointed by Hoekstra..

"Free Flight appears to be one such promising concept, which deserves further safety analysis." (Hoekstra *et al.*, 2002)

2.3 Free Flight Research in the USA and Europe

Research on Free Flight is being actively done through various programs coordinated by government, universities, non-government and research organizations. Some of the research initiatives which have reported considerable progress on the realization of Free Flight concepts are discussed below.

2.3.1 U.S. Free Flight research programs

- The Distributed Air Ground Air Traffic Management project (DAG-TM) (NASA, 1999; Battiste *et al.*, 2002) : This is a NASA initiative under the Advanced Air Transportation Technologies (AATT) program. Various prototype technologies for Free Flight are being developed under this project. The DAG-TM concept offers two different perspectives for the development of new automation technologies. One is a ground based approach where automation tools will be developed to aid the controller on the ground and the other is an air based approach where automation tools will be developed to aid pilots on board aircraft. The ground perspective is being explored by researchers at NASA Ames Research Center while the air perspective is being explored by researchers at the NASA Langley Research Center. Under the air based approach, NASA Langley is currently focusing on CD& for enroute airspace, and self-spacing and self-merging in terminal area airspace. They have also developed a medium fidelity flight simulator to aid in the development of new technologies.

- The FAA-National Airspace System (NAS) Architectures v5.0 (FAA, 2002): The NAS Architecture V5.0 document describes the plans for the modernization of the National Airspace through 2015. Many technologies described in it are essential for the success of various Free Flight initiatives

- SafeFlight 21 project (MITRE, 2002): The Safe Flight 21 program was a joint government (FAA and MITRE) - industry (airlines, Aviation User groups etc.) initiative which was overseen by the RTCA. It was designed to demonstrate and validate, in a real-world environment, the capabilities of advanced surveillance systems and air traffic procedures associated with Free Flight. It used Automatic Dependent Surveillance - Broadcast (ADS-B) and Traffic Information Services -Broadcast (TIS-B) as enabling technologies.

- The Aviation Safety program (AvSP) (Foyle *et al.*, 2003): The AvSP program is a NASA partnership with the FAA, the Department of Defense and the aviation industry to reduce the aircraft fatal accident rate by a factor of 10 by 2020. It aims at not only reducing the aircraft accidents but also at reducing fatalities when accidents do occur. The AvSP program uses safety as its main driving factors in the development of better technology for aviation. It is devoting efforts to the adaptation of new Free Flight technologies such as Cockpit Display of Traffic Information for weather and traffic display and the investigation of human factors issues, so that they are at least as safe as the current setup.

2.3.2 European Free Flight research programs

The crowded airspace and small sectors over the European airspace are making air traffic problems worse. The establishment of Eurocontrol[1] is seen by the aviation community as a major milestone in ATM research, especially in the field of Free Flight. Some of the key European programs are:

- The Program for Harmonized ATM Research (PHARE) (Gool and Schrter, 1999):

[1]Eurocontrol is the European organization for the safety of air navigation. It currently numbers 31 member states. Eurocontrol's primary objective is to develop a seamless, pan-European ATM system for civil and military users.

This program is managed by Eurocontrol on behalf of the European Civil Aviation Conference (ECAC). In this four phase program, the goal of the harmonization and integration of existing European ATC systems is to define a European ATM system that should provide Europe with the means to manage air traffic effectively by the year 2015. Within PHARE, advanced ATM tools were developed in support of a line of advanced operational concepts. The feasibility and merits of these concepts were validated through a series of large-scale real-time simulations, called PHARE Demonstrations (PD). First, demonstrations PD/1 and PD/2 were performed in the UK and Germany, investigating en-route issues for the 2005 time frame. The last demonstration was PD/3, which is planned to investigate multi-sector, multi-center and Terminal Area planning issues for the 2005 to 2015 time frame.

- The NLR (Netherlands) Program: NLR has investigated the Free Flight concept in collaboration with NASA, the FAA and the RLD (Dutch Civil Aviation Authorities). The study started in 1997 and consisted of a number of key areas in Free Flight such as

  - Conceptual Design
  - Safety Analysis
  - Scenario analysis and generation
  - Man-in-the-loop experiment phase I & II
  - Avionics requirements and reliability
  - Critical conflict geometries

  In the final report published in Nov. 2001 (Hoekstra, 2001), it was emphasized that the proposed concepts and tools should be developed, validated in high fidelity and realistic scenarios based on operational factors. Further experiments and trials (Man-in-Loop) are underway, with encouraging initial results regarding the feasibility of Free Flight.

- German Aerospace Center (DLR) program (Gredes, 1997): Researchers at German Aerospace Research Establishment, Institute for Flight Guidance in Braunschweig have developed a tool for controllers for the construction of Conflict-Free routes for aircraft in case of free routing using evolutionary computational techniques. The research work produced a theoretical description and comparison of the algorithms used to calculate a set of aircraft trajectories as well as the steps to obtain a conflict-free one. It also provides a statistical estimation of a route's average length increase when applying these algorithms. Recently DLR and the NLR have come together and formed a common legal entity called AT-One EEIG (AT-One European Economic Interest Grouping) to intensify research on ATM.

2.3.3 New paradigm

In new paradigms, SESAR is Europes "Single European Sky Air traffic Research system" (SESAR Consortium, 2007) and NextGen is the USAs "Next Generation Air Transport System" (Planning and Office, 2007). Both SESAR and NextGen are developments targeted post 2020. The common SESAR and NextGen vision is to integrate and implement new technologies to improve ATM performance. SESAR and NextGen combine increased automation with new procedures to achieve safety, economic, capacity, environmental, and security benefits.

In NextGen, the basis for planning and executing system operations is an aircraft 4D trajectory, from gate-to-gate, including the path along the ground at the airport. Data on the planned and actual trajectories are exchanged between air and ground. All communication is through a satellite based constellation. Both Gate-to-gate 4-D trajectories and voice communications are broadcast. Thus, Communications, Navigation and Surveillance functions are much less ground-based than the current system. In SESAR the operations are based on better forecasting. It is a change from reactive ATM to anticipatory ATM with an objective to reduce operational pressure on human operators by better anticipation of problems and collaborative decision-making procedures. It merges the different trajectory representations into a single one established by the airborne automation system. It relies on accurate monitoring of the scheduled trajectory by means of extremely accurate satellite navigation and on efficient telecommunications network (ground-to-air data links) to enable accurate trajectory information exchanges. In SESAR, all stakeholders (ANSP, AOC, etc.) will have effective and simultaneous access to flight information status. In SESAR, there will be increased automation of air traffic control tools to assist operators to share workload between the air traffic controller on the ground and the pilot and trajectory negotiation planning and support tools in the cockpit to visualize surrounding traffic.

Both SESAR and NextGen concept manages the air traffic based on data transmission, sharing, and utilization. The data-link makes the ATM as an information network to achieve the flight planning, trajectory operations, weather avoidance, self-separation, airport operations, dynamic sectorisation and all other airspace management operations.

2.4 The Present Day Air Traffic Control System

The present day air traffic control system is a highly structured route based system where aircraft are constantly under the control of ground based ATCs. Any aircraft flying under instrument flying rule (IFR) must file a flight plan through appropriate channels to the air traffic service provider.

This flight plan is a contract between the ATCs and the pilot, where ATCs can modify it as and when the need arises, before or after the aircraft flies. On modern aircraft this flight plan is also loaded into its FMS.
Half an hour before the flight commences the pilot advises the control tower confirming the flight plan and intention to fly. The tower controller then advises the pilot to taxi and take off following a standard instrument departure procedure (SID). The tower controller is usually responsible for climb up to 3000ft, from there onwards, i.e. 3000ft until mixing the flight with jet routes, is the responsibility of the terminal area controller. Once an aircraft enters the sector airspace, it comes under the control of enroute ATCs who then navigate the aircraft sector to sector, FIR to FIR, on jet routes using standard hands-off procedures (Nolan, 2004). Figure 2.1 shows a schematic of air traffic services for a typical flight.

A variety of speed/altitude restrictions and procedures are used by the ATCs to maintain an orderly and safe flow of air traffic. Usually around 125nm the aircraft starts its descent to the destination airport (depending upon the aircraft performance). Approx. 25nm from and approx. 10,000-8,000ft to touchdown, the aircraft comes under control of terminal airspace controllers and starts its approach phase where it is assigned a standard terminal arrival route (STAR) which merges an aircraft into an arrival path to the runway. ATCs can apply hold patterns to maintain traffic flow and safe separation between aircraft (FAA, 2003). This system, though inefficient, is necessary for maintaining an extra margin of safety because of the inherent shortcomings in the CNS system which forms the backbone of the present day air traffic control system (ICAO, 2002).

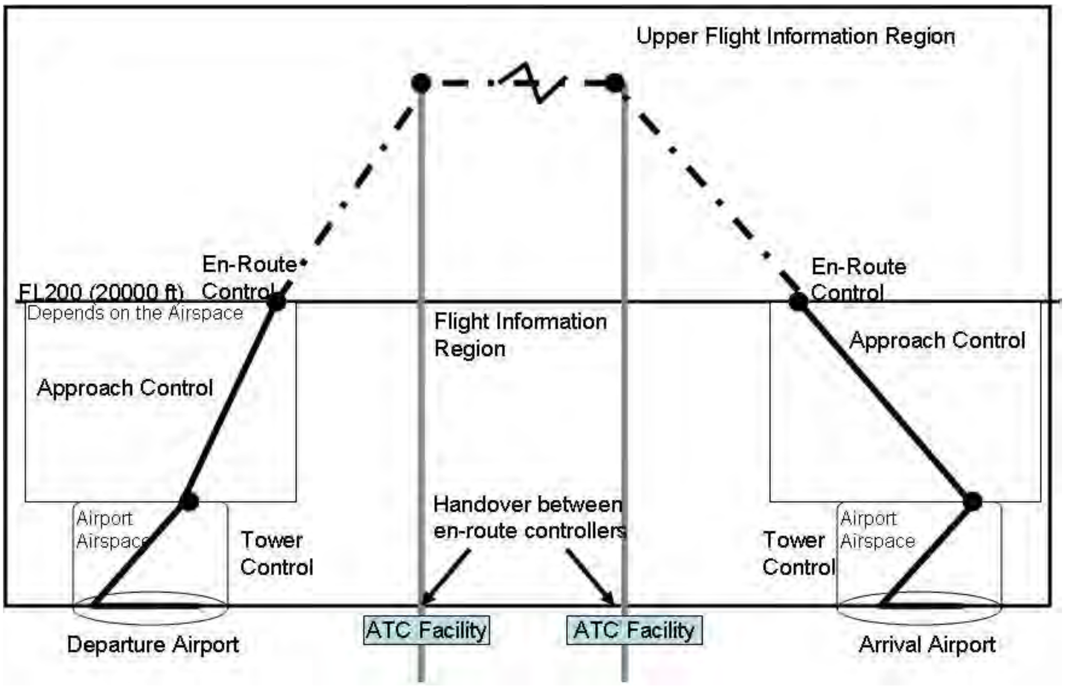


Figure 2.1: Schematic of air traffic services for a typical flight.

In the current ATC system, all the CNS systems are ground based. They offer a variety of information and features to the Air Traffic Service Providers (ATSPs), airline crew and the Airlines Operation Centers (AOCs) for safe and economic management of flights and air traffic. However, there is very limited amount of information sharing between them, for example the flight track data is available to ATCs but not to AOC. A major limitation of the existing CNS system is that these users have access to different set of information making any form of collaboration difficult.

The shortcomings in the present day CNS system and problems faced due to it by the civil aviation worldwide were highlighted a decade ago in the opening remarks by J. Howell, Director of the Air Navigation Bureau of the ICAO at the official opening of the world-wide CNS/ATM Systems Implementation Conference in 1998:

"The primary limitation of the present ground-based system is its restriction to line-of-sight usage...Secondly, we communicate primarily using voice. They do not permit high rates of transmission of data, and they take up a great deal of valuable and diminishing frequency spectrum.....add to all of this a lack of automation, and we find ourselves unable to handle and transfer information and unable to improve and speed up the decision-making process of people in the air and on the ground."

In the last decade, the limitations in the ground based CNS systems, the constraints posed by the fixed route structure and the capacity constraints of present air traffic system have lead to inefficient use of airspace capacity. We will now discuss the existing CNS system followed by the recent advances in it to leverage Free Flight capabilities.

2.4.1 Communication

Air traffic communication is typically divided into two parts: air-to-ground and ground-to-ground. For ground-to-ground voice communication, telephone lines are commonly used. For ground-to-ground data exchange, the primary means is the Aeronautical Fixed Telecommunication Network (AFTN). AFTN can provide ICAO format flight plan data to almost all the airports supporting AFTN hardware (ICAO, 2001); however smaller airports have old AFTN hardware which cannot handle the efficient exchange of data required by present air traffic service providers. Primary users of AFTN are the air traffic service providers and area traffic controllers. In Europe besides AFTN, the '*Socit Internationale de Tlcommunications Aronautiques*' (SITA) network (Chretien, 1974) is also used for ground-to-ground data exchange, mainly by AOCs. In the U.S.A., airlines operation centers often use AviNet by ARINC Inc., which provides advanced data communications such as message switching and content services. An integrated communication system can greatly benefit both ATSP and AOCs in the form of collaborative decision making.
Air-to-ground/ground-to-air voice communication involves HF (high frequency), VHF (very high frequency) and UHF (ultra high frequency) using both analogue and digital signals (Nolan, 2004). The VHF spectrum is primarily used by civilian air traffic, whereas HF spectrum is mainly used by long distance oceanic flights. The UHF spectrum is reserved for military aircraft. As an aircraft flies over different air traffic service providers, it must tune to the appropriate frequency. In modern aircraft with the FMS systems, it can automatically select and tune to appropriate frequency. All communications over this frequency can be heard by any pilot who tunes to it. This provides pilots with a major source of situational awareness regarding neighboring traffic. A standard set of communication procedure is followed between the ATCs and pilots to avoid any misunderstanding or confusion (Nolan, 2004). The only means of data link between the air-to ground and ground-to-ground network is ACARS (Airborne Communication and Addressing System), which uses VHF radio and the satellite communication link (Coppenbarger *et al.*, 2001). However, it is not yet approved for flight safety critical messages.

2.4.2 Navigation

Aircraft use ground based radio beacons and inertial navigation systems to navigate. With advances in satellites navigation, in-flight GPS is fast becoming an important means of navigation in the cockpit and a primary means of navigation in oceanic and remote areas (ICAO, 2007a). GPS provides not only higher accuracy but also greater redundancy. The present navigation system consists of airways and a variety of ground based navigation systems such as VHF Omnidirectional Range (VOR), Distance Measuring Equipment (DME), LOng RAnge Navigation (LORAN) and Tactical Air Navigation (TACAN) to name a few (Nolan, 2004). The airways depending upon their altitude are named as "victory" airways (up to 18,000 ft MSL) and "Jet" airways (18,000 ft to 45,000 ft MSL). In the U.S.A. they are known as Federal Airways and in Europe and Australia they are known as designated airways. At the moment VOR/DME is the ICAO approved primary means of navigation (ICAO, 2007a). For departure control and arrival management, a set of published routes and procedures known as Standard Integrated Departures (SIDs) and Standard Terminal Arrival Rules (STARs) are used.

2.4.3 Surveillance

In present day ATM systems, surveillance is performed by an array of radars which consists of primary surveillance radars (PSR), and secondary surveillance radars (SSR) (Nolan, 2004). The PSR provides information about bearing and distance of the aircraft and the SSR provides information about aircraft identification and its altitude. PSR radars are further sub-divided into long-range air route surveillance radar (ARSR), which scans a wide area (generally a 250-mile radius) and airport surveillance radar (ASR) which uses a shorter-range and scans a narrower area (generally a 60-mile radius). SSR radars have a variety of modes called A, C and S. Mode C and S provides pressure-altitude referenced to 29.92in of mercury in 100ft increments from 1,000ft to 126,700ft. SSR Mode S is the current standard for civil aircraft (ICAO, 1996). The PSR and SSR radar data are processed by the radar data processor to assess the quality and integrity of the data. The data is further processed to identify aircraft, calculates their positions, track their movements, transform the data to display coordinates, and display the resulting information along with maps on the controllers' plan view displays.

2.5 The Future ATM Environment

From the previous section, we have come to the conclusion that the present day ATM system is reaching its operational limits due to inherent shortcomings in CNS systems viz. propagation limitation of line of sight systems, limitations of voice communications and lack of digital data link. Advances in CNS technologies need to be incorporated in the ATM environment for safe and efficient management of growing air traffic. In this section, we will discuss the technological advances in the avionics and CNS systems that can aid in Free Flight

2.5.1 Flight Management System

The role of FMS is to guide the aircraft on its flight plan accurately and efficiently by choosing fuel efficient altitudes given the airspace constraints (Fishbein, 1995). Modern FMS also manages the 4D navigation by ensuring that the required time of arrival at various waypoints in the flight plan are met. The FMS contains a static database of location and frequencies of navigation aids and dynamic information such as the flight plan, weather and wind information. The FMS integrates data from avionics systems to acquire position, heading and velocity vectors and generate altitude, heading and throttle commands to the guidance system. Modern FMS are designed with a data link interface that allows it to transmit its current position, velocity, wind and weather data and to receive updated flight plans from ground control. Given the technological gap between the air and ground systems, the capabilities of FMS remains under utilized. With the advent of digital technologies the CNS functionalities are becoming overlapped and blurred, FMS is considered as a key integrator among the CNS systems.

2.5.2 Communication

The ATN (Aeronautical telecommunication Network) is the proposed communication system for both air-to-air and air-to-ground. ATN utilizes International Standard Organization's (ISO) Open-system Interconnection (OSI) 7-layer protocol architecture for data link communication between various ATM subsystems. This network will provide pilots, ATCs, and AOCs the same set of information with efficiency and reliability. The airborne component of ATN will provide airborne ADS (automatic dependent surveillance)

broadcast data which will include aircraft position and velocity data, and possibly intent information, to ATCs and AOC. The technological implementation of ADS is not yet finalized in Europe, however in the U.S.A. FAA has endorsed the use of 1090ES (Mode-s Extended Squitter) and UAT (Universal Access Transceiver) technology for ADS (Scardina, 2002).
ATN will also include voice communication in the form of controller-pilot data link communications (CPDLC) which will provide a second communications channel for use by pilots and controllers (Hawthorne *et al.*, 1999). It will augment the current voice communications capability. Initial experiments have found that CPDLC can reduce a pilot's workload and minimize errors in air-ground communication (Snyder *et al.*, 2003). However, high workload and time pressure scenarios may result in increased transmission time and more instances of missed messages (Lee *et al.*, 2004; Smith *et al.*, 2001).

Another critical issue in communication is of data link failure and ensuring the integrity/timeliness of data. In this direction RTCA SC-214 (Standards for Air Traffic Data Communication Services) (RTCA, 2007) are established at the request of the FAA to develop standards to define the safety, performance and interoperability requirements for Air Traffic Services (ATS) supported by data communications in support of the NextGen Air Transportation System (Planning and Office, 2007) and Single European Sky ATM Research (SESAR) initiatives (SESAR Consortium, 2007).

It is proposed in the RTCA Standards for Air Traffic Data Communication Services (RTCA, 2007) that, on operational level, for uplink messages which require a crew response, the ground CPDLC application should have an operational response timer installed which waits for the appropriate crew response. If the timer expires, the ATS Facility can initiate another message or contact the aircraft via voice. Invalid uplinks are recognized by the airborne application. If the error is found during the processing (e.g., a CRC failure), no response is sent and the ground views this as an outstanding message. If the error is found during the CPDLC decoding, a CPDLC error response is sent back via the network. Again, ATS can choose to uplink another message or initiate voice contact. On planning level (ICAO, 2007b) it is proposed that the data integrity standards for participating aircraft will be strict, and that the algorithms will have been rigorously tested. Aircraft with lower levels of broadcast reliability and integrity may be able to participate in ATN, only when ATC radar surveillance is available as a mitigation whenever there is a missing or erroneous information. ATC may receive the broadcast information and intervene, when the data is correct and the failure is in an aircraft receiving or display equipments.

2.5.3 Navigation

The navigation systems in the future will be based on satellite derived positions known as Global Navigation Satellite System (GNSS) (ICAO, 2002). In the U.S.A. the system is known as Global Positioning System (GPS) which is maintained by the U.S. department of defence, and in Russia is known as GLONASS. The European union is also coming up with its own satellite navigation system “Galileo” which consist of 12 low earth orbit satellites. These systems are supposed to augment each other providing coverage of up to 99.9% of the earth (Ochieng *et al.*, 2001). In the U.S.A. it is proposed that GPS becomes the standard for enroute and approach navigation (FAA, 2008). However, the satellite position data need to be accurate and have greater integrity for wider acceptability in the aviation community. A proposed Wide Area Augmentation System (WAAS) by the FAA is an augmentation of GPS that include integrity broadcasts, differential corrections and additional ranging signals with improved accuracy for up to 7.6 meters (Grewal *et al.*, 2002).

Due to its accuracy and worldwide availability, GNSS has been designated by the ICAO as the future navigation system for all civil aviation needs (ICAO, 2002). Another aspect of improvements is in the way navigation accuracy is measured. Required Navigation Performance (RNP) is a new concept which is a measure of lateral navigational accuracy of an aircraft in an airspace (ICAO, 2007a). RNP will allow aircraft operators to select which technologies they wish to utilize for the enroute and terminal phases of flight. An RNP certified aircraft will have the ability to maintain specified navigational accuracy during the flight. RNP has been accepted by the FAA as the future means of defining enroute and approach accuracy.

2.5.4 Surveillance

Surveillance coverage and accuracy will be enhanced in the future by integrating the GNSS derived position information with the information provided by the PSR and SSR radars. This information can be translated into a 4D trajectory[2] and made available

[2] A 4D trajectory defines precise position of aircraft in 3D (latitude, longitude and altitude) with a precise ”required time of arrival” constraint at each position.

both to the ATC and airline crew for better and shared decision making. A new system known as automatic dependant surveillance (ADS) is proposed which will eventually replace ground based surveillance systems (RTCA, 1999). In ADS, an aircraft transmits its position based on onboard navigational instruments. There are two versions under development: ADS-A (addressable) also known as ADS-C (Contract) in Europe, and ADS-B (broadcast). The ADS-A system exchange information about specific aircraft and ATC on request. The ADS-B system broadcasts information periodically to all aircraft in the immediate vicinity and all ATM facilities in a specified areas. The primary objective of ADS-A and ADS-B is to improve the surveillance coverage in low or no radar coverage area. ADS will be a suitable medium for the transmission of FMS data to ground based controllers and AOC. The surveillance radar will be operated in parallel with ADS from 2008 to 2012 and then get phased out.

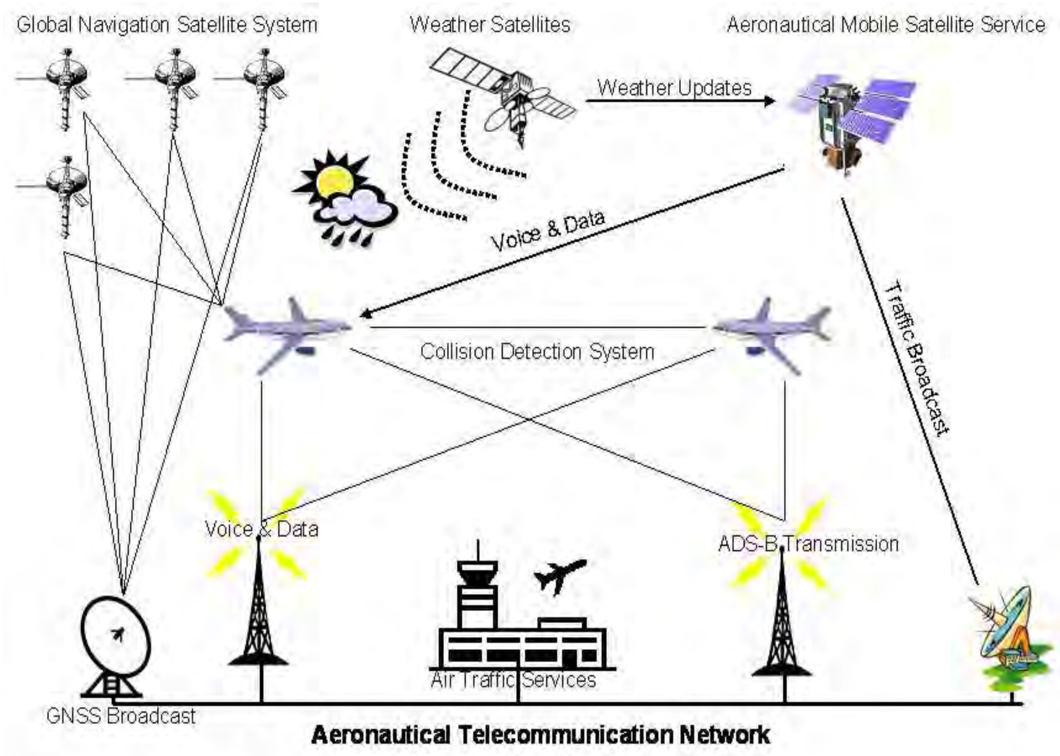


Figure 2.2: Role of CNS in the future ATM environment.

Figure 2.2 illustrates the role of CNS in the future ATM environment. It is evident that the advances in CNS are under-utilized in ATM. The process of moving the ATM legacy system to the state of art in CNS is naturally a slow process. Moreover, every new technology cannot be put in the cockpit without detailed operational and safety analysis.

Field testing of advanced ATM concepts and CNS capabilities is risky and expensive. Computer based models are used to evaluate and analyze these concepts in a simulated environments. Because of the importance of modeling and computer simulation for the evaluation of new ATM system concepts, a wide variety of Air Traffic Simulators are developed by the industry and government organizations for ATM research.

2.6 Air Traffic Simulation Models

A decade ago, sponsored by NASA Ames Research Center under the Advanced Air Transportation Technology (AATT) program, a group of scientists at MIT performed a detailed critical inspection and comparison of most of the existing ATM modeling capabilities (Odoni *et al.*, 1997). Most of the operationally successful models were divided into various categories according to their intended applications, such as capacity and delay models, CD&R models, human/automation models, cost/benefit models, and noise models. It was concluded that some serious deficiencies still remain. It was noted that surprisingly some of the best operational models suffer in several fundamental aspects including: lack of sufficient stochastic options, limited representation of weather and winds, inability to adopt to new ATM concepts, such as Free Flight, and the use of a large amount of resources (acquisition costs, training, input/scenario setting-up, and data collection).

We review here two commercially available models which represent some of the best modeling capabilities at the present time: Total Airspace & Airport Modeler (TAAM) (Datta and Schultz, 1996) and SiMulation MODel (SIMMOD) (Odoni *et al.*, 1997). TAAM is developed by the Preston group (TPG), which is a subsidiary of Boeing Australia. SIMMOD is developed by FAA. Both simulators are discrete event and can be run in both fast time and real time modes. They are widely used by the industry for airport and airspace research.

2.6.1 Commercial Air Traffic Simulation Models

TAAM is currently one of the most sophisticated simulation models that include a system-wide implementation to simulate the entire air traffic system. It processes 4D flight path simulation with interactive 3D graphics. It requires large data files for input and simulation setup and runs on propriety hardware. The major drawback is its high price. On the technological side it lacks stochastic options and does not cover all ATM components like hazardous weather, special use airspace etc, which cannot yet be modeled dynamically. Weather modeling was limited to winds aloft in sectors. The rule set it employs is fixed and inflexible to be used for testing new ATM algorithms/concepts. The CD capabilities are also limited. Because of its complexity, it requires a lot of resources and training to setup.

SIMMOD's main features are its airfield and terminal area airspace model. It uses a node-link system on which all the aircraft move. This underlying network model makes it inflexible to evaluate Free Flight concepts. It requires a high level of ATM expertise to use and its input data preparation process is tedious and time consuming. Moreover, SIMMOD is a one dimensional model, thus can check for conflicts along the longitudinal path only, with no possibility yet for checking lateral or vertical separation violation.

2.6.2 Non-Commercial Research based Simulation Models

Due to the inflexibility of commercial simulation models, high acquisition cost and licensing issues, researchers are independently developing air traffic simulation models for Free Flight concept evaluations. However, most of these models are difficult to obtain as they are developed for specific research needs and purposes. They are difficult to expand and port on different platforms. There is little or no documentation available except for some 8-10 page research papers presenting the findings when using them. Some notable efforts have been undertaken by NASA, Eurocontrol, DSNA and NLR. Some noteworthy models developed by these organizations are: Future ATM Concepts Evaluation Tool (FACET) (Bilimoria *et al.*, 2000) at NASA Ames Research Center, U.S., NLR Air Traffic Control Research Simulator (NARSIM) (Hoekstra *et al.*, 2002) at the National Aerospace Laboratory (NLR) Netherlands, Complete Air Traffic Simulator (CATS) (Alliot *et al.*, 1997) developed by the DSNA (French Air Navigation Service Department) which uses Free flight Autonomous and Coordinated Embarked Solver (FACES) (Alliot *et al.*, 2000) for airborne separation assurance and Recognized ATC Mathematical Simulator (RAMS) developed by the Eurocontrol (RAMS, 2008).

FACET is NASA's simulation tool for exploring advanced ATM concepts. It has a flexible environment for rapid prototyping of new ATM concepts and can be used for both interactive and off-line studies. It can interface with radar data (track and flight plan information) and has weather data processing capabilities. It can run on a single desktop computer and possesses 4D trajectory modeling capabilities. Besides the airborne separation assurance algorithm of NASA, it has capabilities of modeling space launch vehicle within the U.S.A's national airspace system.

NARSIM is an ATM research simulator and supports NLR's in-house ATM research. This includes evaluating new operational procedures, building, testing and evaluating new controller assistance tools and developing airborne decision support systems. It has a variety of in built tools which include trajectory prediction, short term conflict detection and cockpit display of traffic information, to name a few. It has an in-built scenario generation model and interactive interface for ATC as well as pilots.

CATS system is used by French Air Navigation Service Department. It is an enroute traffic simulation engine which is based on discrete, fixed, time-slice execution model wether the position and velocities of aircraft are computed at fixed time intervals (5 sec, 10 sec or 15 sec). It can take input from European Central Flow Management Unit (CFMU) and can simulate 25,000 flights in just 3 minutes. It has an in-built conflict solver FACES which can resolve conflict of up to 8 aircraft using genetic algorithms.

RAMS is a general purpose ATC modeling environment for en-route and terminal airspace as well as controller workloads. It is Eurocontrol's principal simulation tool for evaluating proposed changes to ATM. A large number of studies covering ATC workload, Free Flight concepts like airborne separation assurance and dynamic rerouting were conducted by Eurocontrol using RAMS (Majumdar *et al.*, 2002; Gool and Schrter, 1999; Ehrmanntraut, 2005). It has a highly flexible design to accommodate future ATM concepts.

These models have a few limitations such as, FACET can only simulate high altitude air traffic and cannot model departures/arrivals and transition airspace, moreover it is a simplified 3 degrees of freedom (DOF) model that derives aircraft's performance data from BADA tables. This limits its capabilities in modeling thrust, drag and other aerodynamics required for detailed ATM analysis. RAMS have limited capabilities of modeling landing and takeoff and NARSIM is a state based simulator, which limits its capability to model data linked airspace. Moreover, all of these models are closed architecture software and runs on proprietary hardware. They also require detailed input files for setup which are mostly proprietary in nature. Despite these limitations, air traffic simulation models have become an indispensable tool for prototyping and evaluating advanced ATM concepts for Free Flight.

2.7 Evaluation of Advanced ATM concepts

At this stage it is difficult to predict the final form of the Free Flight. Given the complexity of the ATM system, it will be constantly evolving with the development of new technologies. The only thing that can be said for certain is that over the next few decades, automation will become central to ATM. Advanced ATM concepts envisioned in Free Flight must be investigated and validated in simulation environments before they can be operationally tested in the field (Bilimoria *et al.*, 2001), without which it will be very risky and expensive. As Krozel stated in (Krozel, 2000) that advanced ATM concepts must be better explored for Free Flight scenarios otherwise they will become *"clumsy automation"* where they perform well in normal operating conditions but are ineffective in complex scenarios.
Given the complex system behavior of the ATM system, an integrated approach with an overall system perspective for prototyping is advocated by Erzberger:

"developmental software for these elements must be built and integrated into a test and evaluation system". (Erzberger, 2001)

On similar lines, NASA is developing FFSim (Free Flight Simulation) architecture to evaluate the efficacy of candidate Free Flight technologies and concepts from an overall system perspective (Miles *et al.*, 1999). However, the complex interaction between various air traffic sub-systems results in a chaotic movement of thousands of aircraft through airspace. Capturing all the possible interaction in simulation will be impossible. Therefore, each simulation has to be designed to cover those aspects that are relevant to answer particular research questions.

It is obvious that both simulations and field evaluations will be required in order to develop the final design specifications for the future ATM system. Butler (Butler *et al.*, 2001) states that standard techniques such as testing and simulation have serious limitations in new systems that are significantly *"more autonomous than the older ones"*. Therefore an entirely new approach for modeling and evaluation is required for the evaluation of advanced ATM concepts.

2.7.1 Conflict Detection Algorithms

Conflict detection between autonomous aircraft is a key requirement for Free Flight operations (Bilimoria, 2001; Andrews *et al.*, 2006). It is central to the Free Flight idea of maintaining self-separation by pilots. It is also the most important and widely investigated field in ATM. As stated in the ICAO FANS ATM report..

"Efficient conflict detection and resolution algorithms through decision support systems will provide for the generation of conflict-free trajectories, as well as offering the means to adapt quickly to changing traffic requirements......the ATM system will be better able to accommodate an aircrafts preferred flight profile and help aircraft operators to achieve reduced flight operating costs and delays." (ICAO, 2002)

Also highlighted in (Celio *et al.*, 2000)

"The objective of Free Flight are to provide greater flexibility and cost savings to the user without compromising safety, and conflict probe will provide safety, productivity and user cost saving benefits."

and also mentioned by (Warren *et al.*, 1997)

"Conflict detection is vital for Free Flight operations since traffic conflicts can occur anywhere in a sector with Free Flight."

A variety of CD algorithms are proposed in the literature (Kuchar and Yang, 2000). The key function of any CD algorithm is to project the aircraft position into future, based on state, intent or additional information and to check for potential conflicts. The typical process for CD&R algorithms surveyed is as follows:

1. Project the future 4D trajectory of each aircraft in a scenario for a given time horizon.

2. Apply a CD algorithm to estimate the criticality of the scenario.

3. If the CD algorithm finds a high level of criticality which is greater than a defined threshold for that algorithm, issue a conflict alert. If the algorithm has a conflict resolution component then resolution advisories are generated in this step.

4. Repeat the above steps at a periodic interval (typically at radar sample rate).

Based on how the aircraft position is projected into the future, (Kuchar and Yang, 2000) classified the CD algorithms into three detection methodologies:

- In the nominal method, an aircraft's position is extrapolated based on its current velocity vector. This method provides the best estimate of future aircraft position, for short distances, based on the current state information. However, it does not take into account the trajectory propagation error (Hoekstra *et al.*, 1998; Dowek *et al.*, 2001).

- In the probabilistic method, the uncertainty affecting the aircraft motion is modeled to take into consideration variation in an aircraft's future trajectory (Erzberger and Paielli, 1997; Isaacson and Erzberger, 1997). This is done either by adding a position error to the nominal trajectory or by using a probability density function (e.g. Gaussian). A conflict occurs when the conflict probability is larger than a given threshold.

- In the worst-case method, it is assumed that an aircraft can perform any range of maneuvers. If any one of these maneuvers could cause a conflict, then a conflict is predicted. To reduce the excess false alert rate, worst-case trajectory propagation is limited to a certain look-ahead projection time (Gazit, 1996).

The differences in their detection methodologies are likely to produce differences in their performance and overall effectiveness. Some of these algorithms have been tested in simulation environments (Hoekstra, 2001; Eby and Kelly, 1999; Prandini *et al.*, 2000) and a few in operational environments (McNally *et al.*, 1999; Gool and Schrter, 1999). However, there is no competitive analysis that demonstrates their efficiency and effectiveness (Kopardekar *et al.*, 2002).

2.7.2 Evaluation of CD Algorithms

Evaluation is a crucial step in the overall design, development and implementation process of CD algorithms. To evaluate the performance of a CD algorithm its ability to cope with the most safety critical situations and complex scenarios has to be demonstrated. A rigorous evaluation exercise can establish whether an algorithm will work in an operational environment or not. Evaluation exercises also provide valuable insights into the algorithm performance and can expose design flaws which can be corrected. Since a

CD algorithm uses a variety of parameters, an evaluation exercise can also serve as a fine tuning mechanism for optimal performance of an algorithm. As stressed by Kuchar & Yang:

"...CD & R algorithms need to be examined using a common traffic sample to measure their performance more accurately. The characteristics of the common traffic samples must include different conflict geometries, a variety of traffic density, aircraft type mix, weather and SUA presence, arrival, departures and sector layout." (Kuchar and Yang, 2000)

However, the types of situations and scenarios that need to be investigated can be very difficult to predict for the future Free Flight. The airspace structures and procedures may be completely different or even absent in Free Flight. The types of problems that an algorithm encounter may therefore be difficult to imagine. As noted by Hokestra:

..*"the distributed nature and the infinite number of conflict geometries make it very hard to estimate the actual safety level compared to a centralized system. Because of the certification criteria as currently used by the safety-conscious aviation community, this proof might be required before the introduction of Free Flight".* (Hoekstra *et al.*, 2002)

Three approaches are widely used in the literature to evaluate CD algorithms:

- Theoretical evaluation: In theoretical evaluation, theoretical arguments are developed regarding the performance of the algorithm based on simplistic assumptions about the system. Several automated and semi-automated theoretical tools such as model-checker and theorem proven systems are used in this area (Butler *et al.*, 2001).

- Evaluation on air traffic scenarios: Algorithms are evaluated by testing their performance on either field data (e.g. radar track data) (Bilimoria, 2001) or synthetic data (e.g. generated by a simulator) (Peters, 1998). Since the algorithms will have to be evaluated in field trials involving ATCs this approach has its own merits. In some of the evaluation exercises using field data, a conflict has to be induced by using altitude shifting or by increasing separation standards as any conflict events are already resolved by the controllers. Another alternative is to evaluate the algorithms on synthetic data. By simulating aircraft trajectories, and airspace constraints, air traffic data can be generated with the desired properties.

- Evaluation using standard metrics: Algorithms are evaluated using standard metrics such as ICAO's metric of collision risk (Campos and Marques, 2002; Blom *et al.*, 2003) also known as the Reich collision model. It assumes that the physical shape of each aircraft is a box, having a fixed x,y,z orientation, and the collision risk between two boxes is approximated by integrating the incrossing rate over the time period in which these boxes may be close to each other.

All three approaches have their limitations, for example, theoretical evaluations are challenged on the ground that the CD algorithms have a variety of parameters to optimize therefore it is more than the algorithm, such as the complex interaction between two conflicting aircraft, that determines the performance of a CD algorithm (Watkins and Lygeros, 2002).

When evaluating using field data, the proprietary nature of flight track data makes it difficult to obtain them. Moreover, it does not contain conflicts as they are already resolved by the controllers. Conflicts are therefore induced in field data and how these inducement methods can affect the nature of air traffic data is unclear.

When evaluating using standard metrics, since collisions occur very infrequently, they are not encountered very often, and direct Monte Carlo simulations may not produce significant results (Prandini and Watkins, 2005).

Another option is to use the synthetic air traffic data generated by an air traffic simulator. Large number of conflict scenarios with desired conflict characteristics can then be generated. The challenge lies in: how to develop a realistic model of aircraft movements as well as the uncertainty that enters into the process, such that it results in a conflict?

2.7.3 Performance Measures

To evaluate the performance of CD algorithms, both quantitative and qualitative methodologies are proposed in the literature. In quantitative methodologies, the algorithm is examined with real time traffic and human-in-loop (HIL) in a high fidelity simulation environment mainly looking at implementation and human factors issues (Gool and Schrter, 1999). In quantitative methodologies, they are examined using a range of scenarios in a fast time simulator mainly looking at the core algorithm (Bilimoria, 2001).

Cale (Cale *et al.*, 1998) proposed a generic set of metrics that can be used to assess the accuracy of any conflict probe. For conflict prediction accuracy, they proposed FA, where an alert is predicted but a conflict does not actually occur, and MD, where a conflict occurs which a conflict probe does not predict. It is proposed that the probability of a false alert given an alert outcome is a better measure of the performance, since it is not directly dependent on the size of the sample set. Thus, when the probability of MD is decreased, the probability of FA is increased. Similarly, the probability of FA can be decreased at the expense of increasing the probability of MD. It was recommended that both metrics should be considered when examining the performance of a CD algorithm. It was also shown that the MD and FA are influenced by many factors, e.g., horizontal and vertical separation between aircraft, encounter angles between aircraft, aircraft types in the encounters, and warning time. In the authors's own words..

"...these factors may distinguish the strengths and weaknesses of a particular conflict prediction algorithm as they may affect the performance of each conflict probe differently."

Billimoria (Bilimoria, 2001), evaluated two different CD&R approaches, geometric and force field, in a fast time simulation environment (FACET). Realistic air traffic scenarios were constructed using initial conditions from field data. Three metrics were utilized for safety, efficiency and stability. In safety, MD and FA were used as primary indicators. Several other investigations on quantitative evaluations of CD algorithms have established "MD" and "FA" as the primary metrics to quantify the reliability of a CD algorithm (Krozel *et al.*, 2001). Billimoria, using FACET, also investigated the properties, nature and degree of interaction in conflict data obtained from conflict scenarios generated form realistic air traffic data for Free Flight and structured routing (Bilimoria and Lee, 2001). Encounter angle, horizontal and vertical separations and conflict time and altitude transition between two aircraft were identified as key conflict parameters. It was found that a near linear relationship exists between the number of conflicts and density of aircraft in the airspace in Free Flight.

One of the key requirements of CD&R algorithms is to estimate the criticality of a conflict situation . The System Operatic Characteristic (SOC) approach proposed in (Kuchar, 1995) measures the performance of the algorithms in terms of trade-of between successful alerts and false alerts. Monte-Carlo simulation is used, where the algorithm is applied to a large number of simulations of a particular conflict situation. Parameters which can optimize the trade-off between successful alerts and FA are chosen to compare the performance of different algorithms. However, the SOC curve is based on conflicts defined in terms of violation of minimum horizontal and vertical separation standards only. Whether this definition can capture the criticality of the conflict is unclear (Watkins and Lygeros, 2002).

2.8 The Emergent Questions

From the literature survey, it is evident that existing air traffic simulators are incapable of modeling and evaluating advanced ATM concepts. NASA and Eurocontrol have therefore developed their own air traffic research simulators, which are proprietary in nature and not available to the research community at large. The ATM investigations carried out by these organizations are highly specific to their research needs. Therefore, there is a need for a non-commercial, non-proprietary and more accessible air traffic simulator which can provide an evaluation environment for Free Flight concepts. Such an environment must provide for quantitative evaluation using hundreds of scenarios running in fast time as well as qualitative evaluation with human-in-the-loop in real time. We address this challenge in chapter 3.

In Free Flight, the loss of airway structures may complicate the process of the timely detection of conflicts between aircraft, and their resolution and recovery thereafter. Since, each aircraft flies its desired trajectory based on great circle distance between entry and exit points of the airspace, some form of reference system, to accurately define aircraft's positions, is necessary. Moreover, an efficient representation of the airspace and search procedures are required for rigorous evaluation of any ATM concept. For example, if there are n aircraft in airspace, the number of computations needed, in a straightforward implementation, for predicting conflict between n aircraft at one instant of time is $n(n-1)/2$ and this has to be repeated in every prediction cycle. In chapter 4, we address the challenge of reducing the number of computations required for predicting conflict between n aircrafts, implement some well known CD&R algorithms from the literature and develop a weather avoidance algorithm for traffic constrained enroute airspace.

Current methodologies of generating/developing air traffic conflict scenarios are highly tedious and time consuming. Recorded air traffic data do not contain conflicts, since any such situation is already resolved by the controllers. The rare recorded air traffic accidents are in the order of 10^{-10} per flight hour making it almost impossible for the existing fast time simulators to capture such an event in recorded air traffic data. Inducing conflicts may achieve the desired conflict situation but may affect the traffic flow nature. These approaches might also not be sufficient as the conflict scenarios that need to be investigated can be very difficult to predict for future Free Flight airspace. The types of conflict scenarios that an algorithm encounters in Free Flight may therefore be difficult to imagine. Thus, an algorithmic approach to generate conflicts with desired properties is highly desirable. We address this challenge in chapter 5.

Existing methodologies on conflict scenario generation focus on pre-scripted encounters where the designer pre-selects the conflict parameters and then generates scenarios, which are then fed into the main simulator for execution and analysis. This approach limits the rigorous evaluation of advanced ATM concepts. Ideally, conflict events should unfold dynamically as the scenario progresses. The conflict scenario should evolve over time (i.e. become more complex) by taking constant feedback from the evaluation process. We address this challenge in chapter 6.

The distributed nature of Free Flight and the possibility of an infinite number of conflict geometries make it very hard to estimate the actual safety level achieved by CD algorithms (Hoekstra, 2001). The evaluation exercises fall short of answering the question: ***under what circumstance an algorithm can fail to detect a conflict in Free Flight?*** Given the inherent uncertainties in air traffic environments no CD algorithm can guarantee a 100% success rate. This makes the choice of which CD algorithm to be placed in the cockpit a challenging task for the designers of future ATM systems. We address this challenge in chapter 7 where we apply data mining techniques to learn failure patterns in algorithm performance for given conflict characteristics, and applying this learned knowledge to form an intelligent ensemble of CD algorithms.

Chapter 3

Air Traffic Operations and Management Simulator

Work in this chapter has been partially published in following papers:

1. Alam, S., Abbass, H. A., and Barlow, M. Air Traffic Operations and Management Simulator: ATOMS, IEEE Transactions on Intelligent Transportation Systems. Accepted : 9/08/2007, In Press, 2008

2. Alam, S., Ngugen, M. H. and Abbass, H. A. and Barlow, M. On Architecture, Design and Validation of an Air Traffic Simulator, (SIMTecT-07) Simulation Conference and Exhibition, Brisbane, Australia, June, 2007

In this chapter we introduce ATOMS (Air Traffic Operations & Management Simulator) which is an air traffic simulation system, developed as a part of this thesis, to prototype and evaluate advanced ATM concepts. This chapter describes its architecture, design, functionality and validation. ATOMS can simulate end-to-end airspace operations and air navigation procedures for conventional air traffic as well as for Free Flight. The Free Flight airspace is a continuous space modeled as discretized equal sized hyper-rectangular cells which can maintain intent reference points[1] to speed up the search. ATOMS uses a multi-agent based modeling paradigm for modular design of various air traffic sub-systems. A variety of advanced ATM concepts envisioned in Free Flight are prototyped in ATOMS including airborne separation assurance (ASA), airborne weather avoidance and cockpit display of traffic information (CDTI).

[1] The future aircraft trajectory expressed as a 4-D profile until destination.

3.1 Overview of ATOMS

The key objectives for developing ATOMS are to:

- provide a modular design to model present and future Air Traffic procedures.
- accommodate changes to airspace definitions and ATM infrastructure flexibly.
- be able to closely model the operation of aircraft in the current and future (Free Fight) ATM environments.
- scale well and accommodate different CD&R algorithms.

No third party or any proprietary software libraries are used in order to have the desired flexibility of future extension and development.

The fundamental goal underlying ATOMS is to develop a simulation and modeling environment where Free Flight concepts can be rapidly prototyped and evaluated. The main simulation engine of ATOMS models the basic air traffic and navigation features (e.g. airspace, waypoints, airways, atmosphere, aircraft and trajectory generation) that are essential for evaluating any air traffic concept. All other modules which implement present and future air traffic concepts are built around the core engine.

ATOMS models enroute airspace over a contiguous region, defined by a latitude and longitude pair, through a graphical user interface. It uses a variety of worldwide standard databases such as: Official Airlines Guide (OAG) (OAG, 2001) for the flight schedules, Jeppessen Navdata for the airspace structure database, BADA (Eurocontrol, 2004) for the aircraft performance database and the World Area Forecast System (WAFS) (National Climatic Data Center, 2003) data structure for the atmospheric data representation and International Standard Atmospheric (ISA) (ICAO, 2004) model for the atmospheric data representation. It uses a local-level right handed orthogonal earth-fixed geographic coordinate system where the x-axis points east, the y-axis points north and the z-axis points up. Since the axes of the coordinate system rotate with the earth, therefore to account for the rotation of the coordinate frame, the law of Coriolis (French, 1971) is used in trajectory computation.

As recommended by the ICAO for air traffic visualization, ATOMS utilizes the Lambert conformal conical (LCC) (Bugayevskiy and Snyder, 1995) projection for displaying airspace features and air traffic movements on the graphical user interface. The LCC projection offers the advantage that the shapes of small areas are maintained (no shearing), tearing only occurs around the edges, amount of distortion of areas is minimal near lines of tangency (compression) and distances are correct along the lines of tangency.

To model 4D aircraft trajectories, ATOMS uses the equations of motion to calculate the velocity and position data of the aircraft over a predefined time interval in a geographic coordinate frame. Distance and track calculations are based on great circle route (GCR) (Bugayevskiy and Snyder, 1995) calculations in an elliptical earth model. The motion equations are first order integrated to compute the new aircraft position at the end of the discrete time interval. The effect of the earth's rotation and the wind are included in the equations of motion. Aircraft performance parameters are computed through aerodynamic equations and performance parameters defined in BADA files licensed from Eurocontrol. ATOMS can be run in real time, fast time, or slow time with various other options described in the subsequent sections.

3.2 Architecture and Design

This section discusses the architecture and design principles of ATOMS including aircraft modeling, atmosphere modeling, trajectory generation, airspace modeling and it's graphical user interface.

3.2.1 Architecture

We adopted an agent-based approach to model and simulate the various entities of ATOMS. This approach helped us in the decomposition of agents (e.g. Aircraft agent) as shown in Figure 3.1, into activities based on the logical decomposition of roles (communication, navigation and surveillance) and interactions (conflict detection, weather avoidance, etc.) with the environment. The interactions between the agents is implemented using a 4D discretized airspace data structure where each agent has access to intent information of the other agents within a predefined area. It is assumed that each aircraft broadcasts its current state and intent information via data link and perfect information is available to all other aircraft within the broadcast range which can be relaxed if needed. The simulation model is discrete-event and assumes that all the necessary data is available

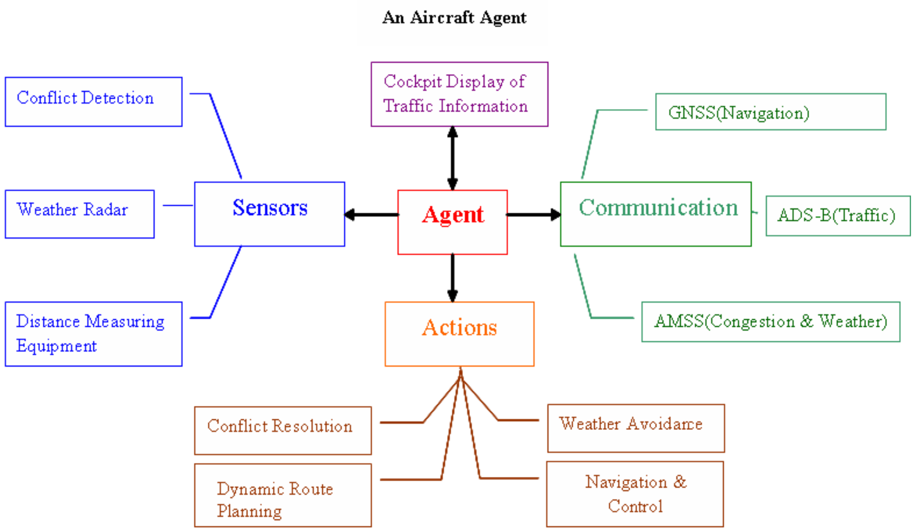


Figure 3.1: An agent based model of an aircraft in ATOMS with its sensors, actions and communication modeled around Free Flight concepts.

without any errors, this assumption can be relaxed if needed because in a real air traffic scenario the data is highly susceptible to corruption due to electronic interference and atmospheric conditions. The software is written using the C++ programming language and the GUI is written in Microsoft© Visual C++ version 6.0. The GUI can be detached from the model which then can be run as a console application on a UNIX/LINUX platform.

ATOMS is a 4D simulator, i.e. it works with a latitude-longitude pair for each flight with an associated altitude and time for each of these pairs. For Free Flight the airspace is considered in 3D which is approximated with grid cells in a hyper-rectangular discrete space as shown in Figure 3.2. The size of these 3D grid cells is based on the current set of separation standards by the ICAO ICAO (1996). In Free Flight, the aircraft trajectory is generated as a series of points in the airspace using these cells as reference. These cells also act as a repository of the airspace containing information such as: weather, atmospheric properties and intent information of aircraft. The fourth dimension is added to these cells by assigning them estimated time of arrival (ETA) for each aircraft passing through them in the near or immediate future. Looking at a 2-D version of an airspace cell block each cell can be defined by its start (latitude, longitude) and end (latitude, longitude). Using these two points, the corner co-ordinates for each cell in the whole airspace can be calculated and thus the extremes of the airspace can be defined using the geodetic earth co-ordinate system. This enables us to uniquely identify each block of the discretized airspace by (i,j,k) coordinates as shown in Figure 3.2. Such a discretization

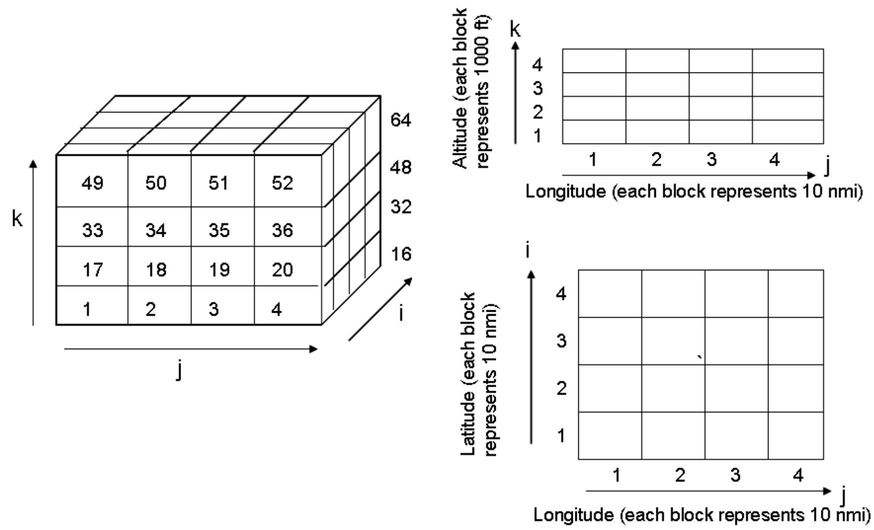


Figure 3.2: Airspace modeled as hyper-rectangular discrete space. Block 1 is (1,1,1) in (i,j,k) coordinates, block 64 is (4,4,4) in (i,j,k) coordinates. On right side the labelling scheme in a 2D representation can be seen.

of airspace helps maintain reference points in the space in the absence of waypoints and airways for trajectory simulation and intent information propagation, and secondly it limits the airspace search volume for conflict detection. This is explained in more detail in the next chapter.

ATOMS uses the flight plans (origin-waypoints-destination) when simulating the present air traffic system and uses great circle route (origin-destination) when simulating the Free Flight. Flight plans are then used to generate the arrival and departure schedules for the airports in the airspace under consideration or the arrival times at the fixes on the boundary of the airspace. Then using each flight plan, a flight trajectory is developed using the aircraft performance data, the navigational data and the atmospheric data from the relevant database files. The simulator triggers flights based on their estimated time of departure (ETD) from airports within the designated airspace and for overflying flights, they are activated based on the ETA at a boundary fix. The flight profiling is done for each flight to determine the top of climb (TOC) and top of decent (TOD), these points are then inserted in the flight route generated by the FMS. A conceptual representation of simulator modules and data flow is shown in Figure 3.3.

A simulated time clock is used to step the discrete event simulator through the sequence of time units using a fixed increment time advance. The simulation clock is

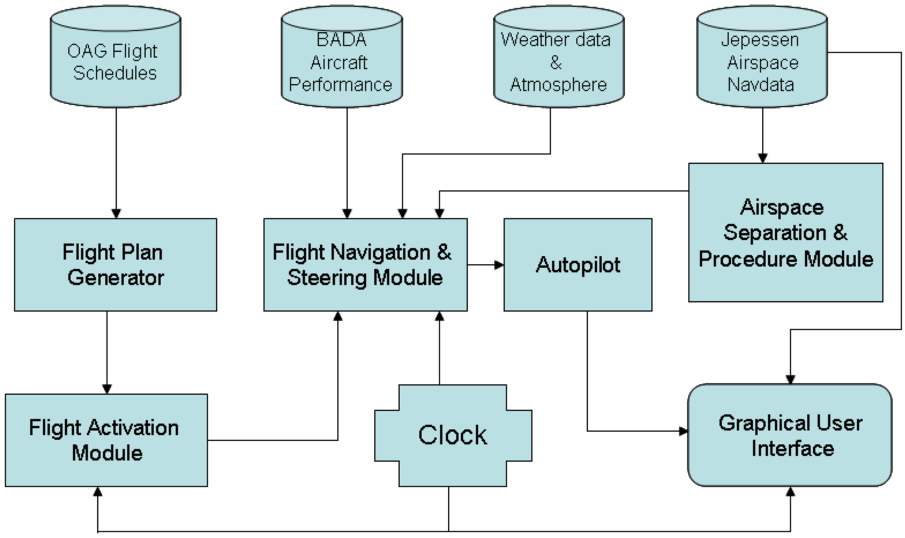


Figure 3.3: A conceptual representation of ATOMS's core modules.

advanced dT units and the aircraft are considered to move at the end of the dT which is set at one second (real time mode) as default. However, dT can be changed for slow time and fast time modes using the GUI interface.

3.2.2 Airspace Modeling

ATOMS is capable of modelings different airspaces around the world. So far it can model the Gulf airspace and Australian airspace and modeling of other airspaces (South East Asia) is underway. ATOMS uses airspace data (airways, waypoints, airports, navigational fixes, arrival fixes, jet routes) from the designated airspace handbooks by the air traffic service providers of the region or from the independent sources (Jeppessen) which provides the basis for the airspace modeling. The data is pre-processed to remove unwanted fields and records. The airspace is built as a set of defined points. These points are loaded from the airspace definition file and the airspace volume is built. The waypoints and other navigational aids that are loaded are within the airspace and those that are immediately next to the boundary of the airspace under consideration.

Airports and airspace pre-boundary fixes are simulated as queues. These queues are used to simulate airport and airspace capacity constraints by imposing time restrictions on the soon to be active flights. For an airport, when a flight departs from it based on its ETD, it is put under busy status for a certain duration of time based on aircraft category (light, medium, heavy), so if there is any other flight scheduled for departure in the next time window, then it will have to update its ETD accordingly and have its flight profile regenerated. Similarly for an overflying flight when it becomes active at airspace boundary waypoints based on their ETA and flight level, the waypoint for that flight level is put under busy status for ceratin time based on its wake category. If there is another flight which needs to be activated at that waypoint at the same flight level then it will search the next two lower flight levels for availability, if it finds a flight level available it gets activated else it is put on a hold pattern until the flight level is available.

3.2.3 Atmospheric Modeling

Since aircraft motion, altitude and speed measurements are highly dependent on temperature, pressure, and other atmospheric properties, modeling of these parameters is necessary for accurate computation of aircraft performance parameters and prediction of aircraft trajectory. ATOMS uses the ISA model (ICAO, 2004) equations for atmospheric modeling, where atmosphere is considered as a perfect gas in equilibrium on a non-rotating flat earth. ATOMS can also account for deviation from the ISA model. ISA defines air temperature, pressure and temperature as well as other atmospheric parameters as functions of altitude. Computation of atmospheric properties is based on the following assumptions and computations:

Temperature: In ISA conditions,

$(T_0)_{ISA} = 288.15K$

As the altitude increases the temperature decreases, initially at a constant rate, up to an altitude of approximately 11,000 meters. At non-ISA condition, if $\Delta(T)_{ISA}$is the temperature difference from ISA, then the temperature at sea level, T_0, is defined as:

$T_0 = (T_0)_{ISA} + \Delta(T)_{ISA}$

Below tropopause, the temperature is computed as a function of altitude h (meters):

$T = T_0 - 6.5 \times h/1000$

Figure 3.4 shows the variation in temperature at different altitude in ATOMS.

Density: In ISA conditions, the air density is:

$(\rho_0)_{ISA} = 1.225kg/m^3$. The air becomes less dense as altitude increases. In non-ISA condition, the air density at sea level, ρ_0, is computed as:

$(\rho_0) = \frac{(\rho_0)_{ISA}(T_0)_{ISA}}{T_0}$

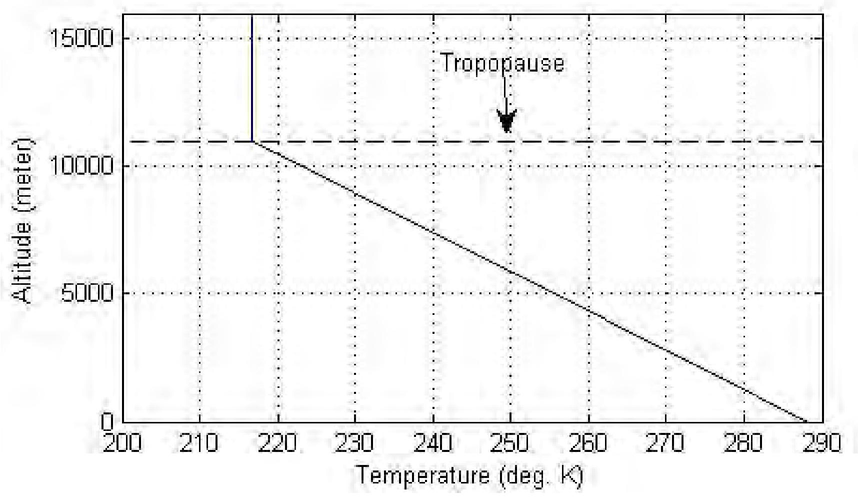


Figure 3.4: Variation in temperature at different altitude in the atmospheric module of ATOMS.

Below the tropopause, the air density, (ρ_0)(kg/m3) , is computed according to the temperature as follows:

$\rho = \rho_0[\frac{T}{T_0}]^{-\frac{g}{K_T R}-1}$

where $-\frac{g}{K_T R} - 1 \approx 4.25864$

R is the real gas constant for air ($287.04 m^2/Ks^2$), g is the gravitational acceleration ($9.81 m/s^2$), and K_T is the ISA temperature gradient with altitude below the tropopause ($-0.0065^0 K/m$). In ISA condition, and tropopause altitude, h_{Trop}=11000m,and $\rho_{Trop} \approx 0.3636 kg/m^3$

Above the tropopause, the air density at altitude h (meters) is computed as:

$\rho = \rho_{Trop} e^{-[\frac{g}{RT_{Trop}}]\times(h-h_{Trop})}$

Figure 3.5 shows the variation in air density at various altitude in ATOMS.

Pressure: At sea level, the standard pressure of the air is 2,116.22 lb/ft2 ($1.01325 \times 105 N/m^2$). The pressure at any point in a stationary fluid is determined by the weight of the fluid above that point. When the altitude increases from sea level the pressure slowly decreases throughout both atmospheric regions. At the end of the troposphere the pressure is equal to only 22% of the pressure at sea level. The rate of change of pressure is associated with the rate of change of density. Figure 3.6 shows the variation in pressure with altitude in ATOMS.

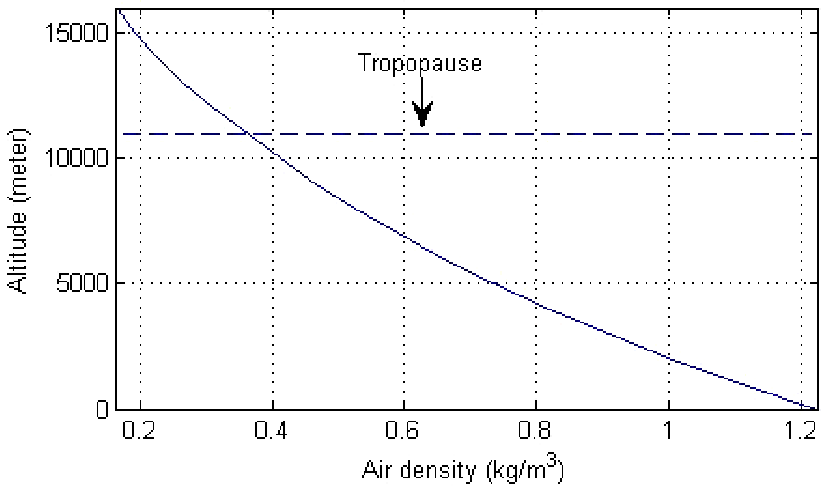


Figure 3.5: Variation in air density at various altitude in the atmospheric module of ATOMS.

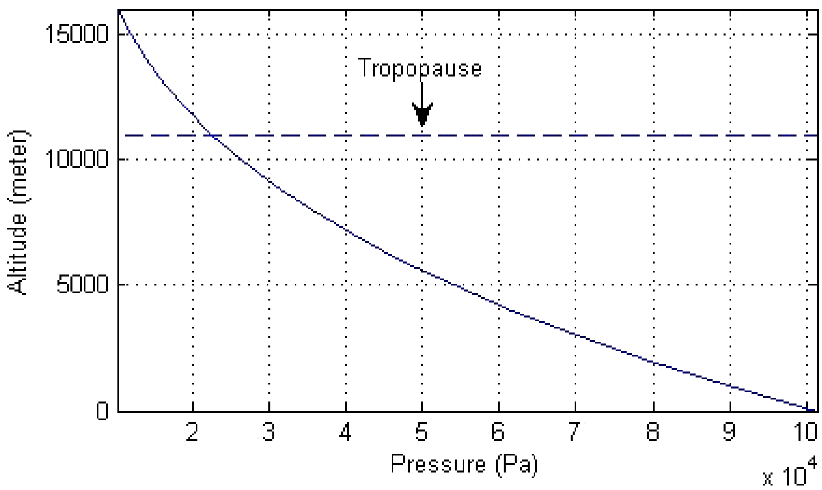


Figure 3.6: Variation in pressure with altitude in the atmospheric module of ATOMS

3.2.4 Wind Modeling

Wind is the single most important external factor that affects aircraft trajectory. A grid based wind model is used in ATOMS. The wind forecast data is provided by the Bureau of Meteorology in a grid format. As shown in Figure 3.7, it is an image file which shows the forecast for atmospheric and wind information in a grid format for the Australia wide region. The grid start points are 103 degrees latitude and 100 degrees longitude and end points are 160 degrees latitude and 150 degrees longitude. Each grid covers 50 degrees of latitude and 5 degrees of longitude. There are six altitude levels specified starting from 16500 ft up to 44500 ft, and it is assumed that there is a linear variation between these levels. Therefore the entire region is covered by a grid of dimension 8 X 12 X 45. The data is in the format "ddfffTT". "dd" is the true wind direction, to the nearest 10 degrees, and "fff" is the wind speed in whole knots, except wind speeds less than 5 knots are shown as 99000. "TT" is the temperature in whole Celsius degrees. All temperature values are negative. This forms a grid over latitude, longitude and altitude. The computation required during the simulation run are minimized by using a set of uniform grids and a simple linear interpolation algorithm. The data is stored in a matrix of dimension 8 X 12 X 45.

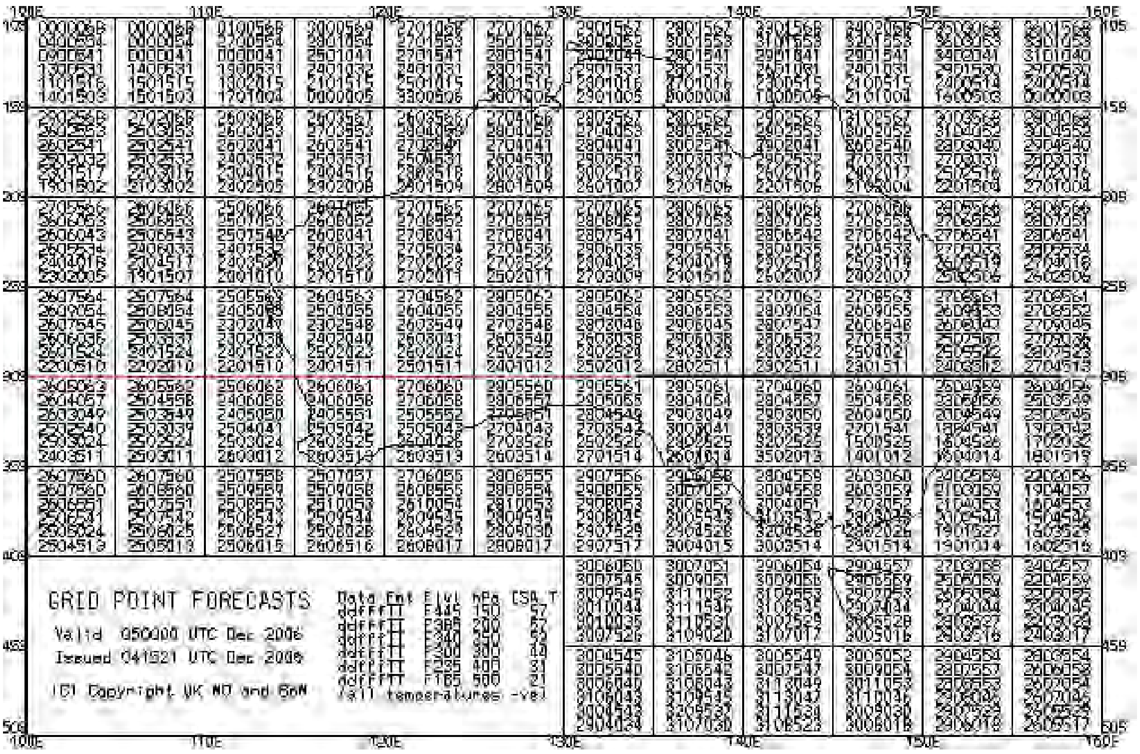


Figure 3.7: The Grid point forecast file from Bureau of Meteorology. The wind direction, speed and temperature are shown in each grid for five representative altitudes.

The wind data is then profiled by interpolating two known wind quantities at two different points and across altitudes to ensure no discontinuity in the wind calculation. The relative wind is assumed to apply to all points of the aircraft. Given the aircraft's location (latitude/longitude) and altitude and interpolating between the nearest grid points and levels respectively (assuming linear characteristics of the atmosphere between these points in space) a representative value for wind aloft, temperature and density is calculated. All these atmospheric elements are set up in the airspace data structure that contains the temperature, northerly/easterly wind component and the pressure at the given altitude. This gives flexibility to use either the ISA atmospheric model with Zero wind condition or actual atmospheric and wind data.

3.2.5 Air Traffic

To generate air traffic, a flight demand generator is developed that generates flights at preset points in the airspace using different traffic patterns and densities. The flight plans for all flights, on Jet routes, in the Australian airspace, for the seven days of the week for the first week of each month from January 2006 till August 2006 are recorded. These figures are used as a baseline. No VFR (Visual Flight Rules)[2] flights, military or general aviation flights are recorded and are substituted by increasing the flights by a factor of 25%. The traffic generator uses the scheduled flights and generates traffic based on the airport and/or pre boundary waypoint traffic density. The point of activation is the airport of origin if flight is in the airspace or it is the first pre-airspace boundary waypoint if it is an overflying flight. These flights are assigned an aircraft type based on the route database which contains the typical aircraft used on various routes. The relevant aircraft true air speed and the requested flight level is added to the generated flight record using aircraft performance database. The airway-waypoint route is assigned based on the city-pair and the STAR profile is based on approach airport.

For Free Flight air traffic, only the origin-destination pair is used and enroute waypoints are not computed. The point of activation for a flight is set to the airport of origin if it is in the airspace or it is set to the first airspace boundary waypoint if the airport of origin is outside the airspace. Similarly the point of deactivation for a flight is either the destination airport if it is within the airspace or the last waypoint point on the airspace boundary if its destination airport is outside the airspace.

3.2.6 Aircraft Performance Characteristics

For aircraft performance modeling, ATOMS uses BADA Eurocontrol (2004) which provides aerodynamic equation and performance tables for a variety of aircraft. It is maintained by the Eurocontrol Experimental Center and is designed to simulate aircraft

[2] Pilots flying under VFR assume responsibility for the separation of their aircraft from all other aircraft

movements in ATM environments and can be used to calculate a variety of flight parameters including fuel. The latest version 3.6 includes detailed information on 91 supported aircraft gathered from flight operating manuals and test beds. The database consists of ASCII files containing performance and operating parameters for all the supported aircraft. The fidelity of the BADA aircraft performance model offers the advantage that the model accuracy can be tuned through data and secondly, the approach facilitates agreement with other research projects. The core data is stored in the following files:

- Operations Performance Files (*.OPF) including aircraft-specific performance parameters,
- Airline Procedure Files (*.APF) with aircraft-specific operational data,
- Performance Table Files (*.PTF) with a summary of an aircrafts performance.

The *.PFT files provide look-up tables for cruise, climb and descent performance at different flight levels. For detailed performance calculations, on the other hand, only the *.OPF and *.APF files are required. The *.OPF files include a total of 51 parameters per aircraft which specify the aircrafts mass and flight envelope together with its aerodynamic and engine capabilities as shown in Table 3.1.

The *.APF files supplement the data by providing typical speeds or mach numbers for climb, cruise and descent conditions. This information can be used to calculate a flights speed schedule. Furthermore, a Global Parameter File (BADA.GPF) is provided containing non-aircraft-specific parameters like maximum accelerations, holding speeds and speed coefficients. The BADA data in combination with the underlying performance model can be used to calculate lift and drag as well as thrust and fuel flow in all flight phases. The model is believed to be most accurate for cruise conditions. The aircraft model in BADA is a point-mass model which balances the rate of work done by forces acting on the aircraft and the rate of increase in potential and kinetic energy. This approach, referred to as a Total Energy Model (TEM), is represented by the following equation (Eurocontrol, 2004):

$$(T - D) \times v_{TAS} = m \times g \times \frac{dh}{dt} + m \times v_{TAS} \times \frac{dv_{TAS}}{dt} \tag{3.1}$$

where : T = thrust[N] D = aerodynamic drag[N]
M = aircraft mass [kg] g = gravitational acceleration[m/s^2]

Table 3.1: BADA parameters and their description.

Category	Parameter and Description
aircraft type	n_{eng} number of engines [-] engine type Jet/Turboprop/Piston wake category Heavy/Medium/Light
mass	m_{ref} reference mass [t] m_{min} minimum mass [t] m_{max} maximum mass [t] m_{pyld} maximum payload [t]
flight envelope	v_{MO} max. operating speed [kt] M_{MO} max. operating Mach number [-] h_{MO} max. operating altitude [ft] h_{max} max. altitude at MTOW and ISA [ft] G_W weight gradient on max. altitude [ft/kg] G_t temp. gradient on max. altitude [ft/C]
aerodynamics	S reference wing surface area [m] $C_{D0,CR}$ parasitic drag coefficient (cruise) [-] $C_{D2,CR}$ induced drag coefficient (cruise) [-] $C_{D0,AP}$ parasitic drag coefficient (approach) [-] $C_{D2,AP}$ induced drag coefficient (approach) [-] $C_{D0,LD}$ parasitic drag coefficient (landing) [-] $C_{D2,LD}$ induced drag coefficient (landing) [-] $C_{D0,\Delta LDG}$ parasitic drag coef. (landing gear) [-] C_{M16} Mach drag coefficient [-] $(V_{stall})i$ stall speeds for TO,IC,CR,AP,LD [kt] $C_{Lbo(M=0)}$ Buffet onset lift coef. [-] *jets only* K Buffeting gradient [1/M] *jets only*

Category	Parameter and Description
engine thrust	$C_{Tc,1}$ 1st max. climb thrust coefficient [N] $C_{Tc,2}$ 2nd max. climb thrust coefficient [ft] $C_{Tc,3}$ 3rd max. climb thrust coefficient $[1/ft^2]$ $C_{Tc,4}$ 1st thrust temperature coefficient $[C]$ $C_{Tc,5}$ 2nd thrust temperature coefficient $[1/C]$ $C_{Tdes,low}$ low alt. descent thrust coefficient [-] $C_{Tdes,high}$ high altitude descent thrust coef. [-] h_{des} transition altitude [ft] $C_{Tdes,app}$ approach thrust coefficient [-] $C_{Tdes,ld}$ landing thrust coefficient [-] $V_{des,ref}$ reference descent speed [kt] $M_{des,ref}$ reference descent Mach number [-]
fuel flow	C_{f1} 1st TSFC coefficient [kg/min/kN] C_{f2} 2nd TSFC coefficient [kt] C_{f3} 1st descent fuel flow coefficient [kg/min] C_f4 2nd descent fuel flow coefficient [ft] C_{fer} cruise fuel flow correction coefficient [-]
ground operation	TOL take-off length [m] LDL landing length [m] span wingspan [m] length aircraft length [m]

v_{TAS} = true air speed[m/s] h = altitude[m]

Equation 3.1 includes three independent variables which represent typical aircraft control inputs: thrust T, true airspeed v_{TAS} and rate-of-climb (or descent) dh/dt. When modeling a cruise flight segment, the Total Energy equation 3.1 can be used to calculate thrust and in case a constant altitude cruise is assumed, the rate-of-climb becomes zero.

The ISA standard atmosphere is typically assumed for BADA calculations, although a temperature deviation from ISA could be specified. Air temperature and density vary with altitude and can be calculated from ISA assumptions as discussed in previous section. Mach numbers from the BADA speed schedule can be converted to true airspeeds by the following equation:

$$v_{TAS} = M \times a = M \times \sqrt{\gamma \times R \times T^*} \tag{3.2}$$

where γ = isentropic expansion coefficient for air a = local speed of sound$[m/s^2]$
R = universal gas constant for air $[m^2/Ks^2]T^*$ = local temperature[K]

Since the aerodynamic drag is required in equation 3.1, lift and drag coefficients C_L and C_D as well as the respective forces are calculated using the following equations:

$$C_L = \frac{2 \times m \times g}{\rho \times V_{TAS}^2 \times S \times cos\phi} \tag{3.3}$$

$$C_D = C_{D0,CR} + C_{D2,CR} \times C_L^2 \tag{3.4}$$

$$L = \frac{1}{2} \times C_L \times \rho \times V_{TAS}^2 \times S \tag{3.5}$$

$$D = \frac{1}{2} \times C_D \times \rho \times V_{TAS}^2 \times S \tag{3.6}$$

where: ρ = air density[kg/m^3] $C_{D0,CR}$ parasitic drag coefficient[-]
ϕ = bank angle[degrees] $C_{D2,CR}$ = induced drag coefficient[-]
S = reference wing surface area[m^2]

Wing area and drag coefficients are given from the BADA *.OPF file.

3.2.6.1 Cruise Phase Performance Characteristics

During cruise phase (level flight), the flight path angle ψ is assumed to be zero. The engine thrust is set equal to aerodynamic drag (T = D), and lift equals aircraft weight (L=W). Combining the lift equation above with the level flight condition of L = W = mg, the lift coefficient, C_L can be determined as (Eurocontrol, 2004)

$$C_L = \frac{2mg}{\rho V_{TAS}^2 S} \tag{3.7}$$

where m (kg) is aircraft mass, g is the gravitational acceleration ($9.81m^2/s$), ρ is the air density, V_{TAS} is the true airspeed and S is the wing reference area.

For a jet aircraft, the drag coefficient C_D is expressed as a parabolic function of the lift coefficient C_L, also called the parabolic drag polar, as follows:

$$C_D = C_{D0,CR} + C_{D2,CR} \times (C_L)^2 \tag{3.8}$$

where $C_{D0,CR}$ is the parasitic drag coefficient, i.e. the drag coefficient when the lift coefficient is zero, and $C_{D2,CR}$ is the induced drag coefficient, which presents the portion of drag due to lift.

3.2.6.2 Climb Phase Performance Characteristics

In the ISA condition, BADA estimates the maximum climb thrust T_{climb} as a quadratic function of altitude h:

$$T_{climb,ISA} = C_{Tc1} \times (1 - \frac{h}{C_{Tc2}} + C_{Tc3} \times h^2) \tag{3.9}$$

where C_{Tc1} (Newton), C_{Tc2} (feet), C_{Tc3} $(1/feet^2)$, are the max climb thrust coefficients. In other atmospheric condition, where temperature deviations from the standard atmosphere is ΔT_{ISA}, the corrected climb thrust is defined as:

$$T_{climb} = T_{climb,ISA} \times (1 - C_{Tc5} \times (\Delta T_{ISA})_{eff}) \tag{3.10}$$

where $(\Delta T_{ISA})_{eff} = \Delta T_{ISA} - C_{Tc4}$, C_{Tc4} (deg. Celsius), C_{Tc5} (1/deg. C) are the thrust temperature coefficients and $0 \leq (\Delta T_{ISA})_{eff} \times C_{Tc5} \leq 0.4$

3.2.6.3 Descent Phase Performance Characteristics

Descent Thrust is calculated using different correction factors for high and low altitude, based on the transition altitude h_{des} for calculation of descent thrust, and approach and landing configurations.

- High altitude descent, i.e. top of descent (TOD) altitude is higher than the transition altitude h_{des} $(h > h_{des})$

$$T_{des,high} = C_{Tdes,high} \times T_{climb} \tag{3.11}$$

- Low altitude descent, i.e. TOD altitude is lower than the transition altitude h_{des} $(h < h_{des}$

$$T_{des,low} = C_{Tdes,low} \times T_{climb} \tag{3.12}$$

- Approach: once the aircraft has descended below 8000 ft (h < 8000ft) and the airspeed falls below a certain threshold ($V_{TAS} < V_{min,cruise}$ + 10kts), the approach flap setting and thrust setting are used.

$$T_{des,app} = C_{Tdes,app} \times T_{climb} \tag{3.13}$$

- Landing: once the aircraft has descended below 3000 ft (h < 3000ft) and the airspeed falls below a threshold ($V_{TAS} < V_{min,approach}$ + 10kts), the landing flap setting and thrust setting are used.

$$T_{des,ld} = C_{Tdes,ld} \times T_{climb} \tag{3.14}$$

where $C_{Tdes,high}$ and $C_{Tdes,low}$ are the high and low altitude descent thrust coefficients respectively. $C_{Tdes,app}$ and $C_{Tdes,ld}$ are the approach and landing thrust coefficients respectively. T_{climb} is the maximum climb thrust.

The minimum speeds for the aircraft is specified as follows:

$$V_{min,cruise} = C_{V_{min}} \times V_{stall,CR} V_{min,approach} = C_{V_{min}} \times V_{stall,AP} \tag{3.15}$$

where C_{Vmin} is the minimum speed coefficient and set to 1.3 for all aircraft. $V_{stall,CR}$ and $V_{stall,AP}$ are the cruise and approach stall speeds.

The simulated aircraft navigates from one point to another based on a predefined flight plan or Free Flight trajectory. The aircraft's FMS algorithm plans 4D trajectories, determines the respective control inputs for the aircraft to follow simulated trajectories and models the dynamic aircraft reaction to these control inputs as shown in Figure 3.8. The aircraft state and position is computed at a sampling rate of 1 second. The subsequent integration of the aircraft state results in the desired flight profile. The aircraft class contains four key modules:

- Flight Module: Contains flight plan and generic data about flight.
- FMS: Generates 4D flight profile from the flight plan data.
- AutoPilot: Controls the navigation and steering of the aircraft based on the flight profile generated by the FMS or from the navigation commands generated by the Command Generator.
- Navigation Command Generator: Generates appropriate navigation commands based

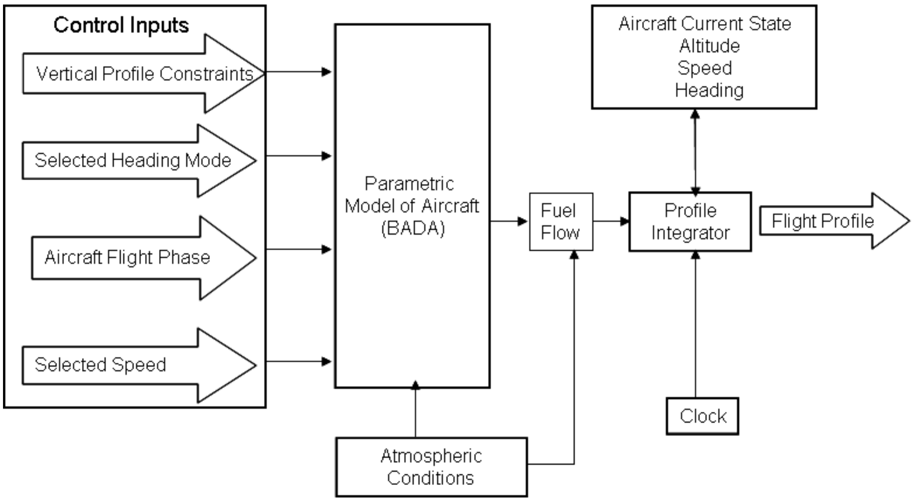


Figure 3.8: Parametric model of an aircraft and its interaction with other modules to generate flight profile.

on the flight profile and sends to autopilot for execution.

For computation of aircraft performance parameters from the BADA database, the phase of flight is required and is defined as shown in Figure 3.9. These flight phases are updated by the auto-pilot module based on altitude and flight profile data. For the GUI, appropriate color codes (blue-take off & climb, green-cruise, grey-descent, red-approach & land) are assigned, which makes it easy to understand which phase an aircraft is in at a given point of time.

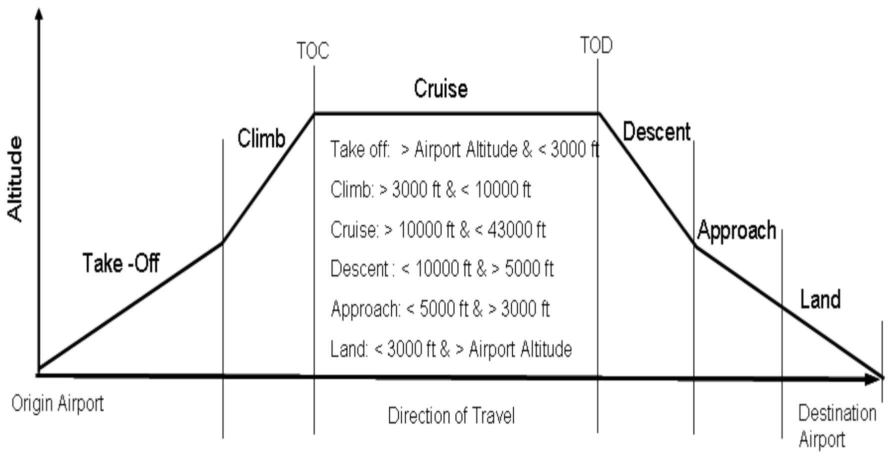


Figure 3.9: Typical phases of flight in ATOMS.

3.2.7 Aircraft Navigation and Steering

In this section we discuss the navigation and guidance equations used to move (fly) the aircraft in the simulation. The motion equations are used to calculate the velocity and position data of the aircraft over a predefined time interval in a geographic coordinate frame. Distance and heading calculations are based on Great Circle route (GCR) and Rhumb Line (RL) calculations in an ellipsoidal earth model. The velocity equations are integrated to compute the new aircraft position at the end of the time interval using a first order Euler integration method. The effect of the earth's rotation and the wind are included in the equations of motion.

3.2.7.1 Geographic Reference

Figure 3.10 shows the right handed orthogonal frame centered at the center of the earth. In this coordinate frame, the position of an aircraft is expressed in terms of latitude (ϕ), longitude (λ). The origin of the system coincides with the aircraft as it moves above the earth. The z-axis is oriented following the descending direction of gravitational attraction located on the origin. For most terrestrial applications, it is more convenient

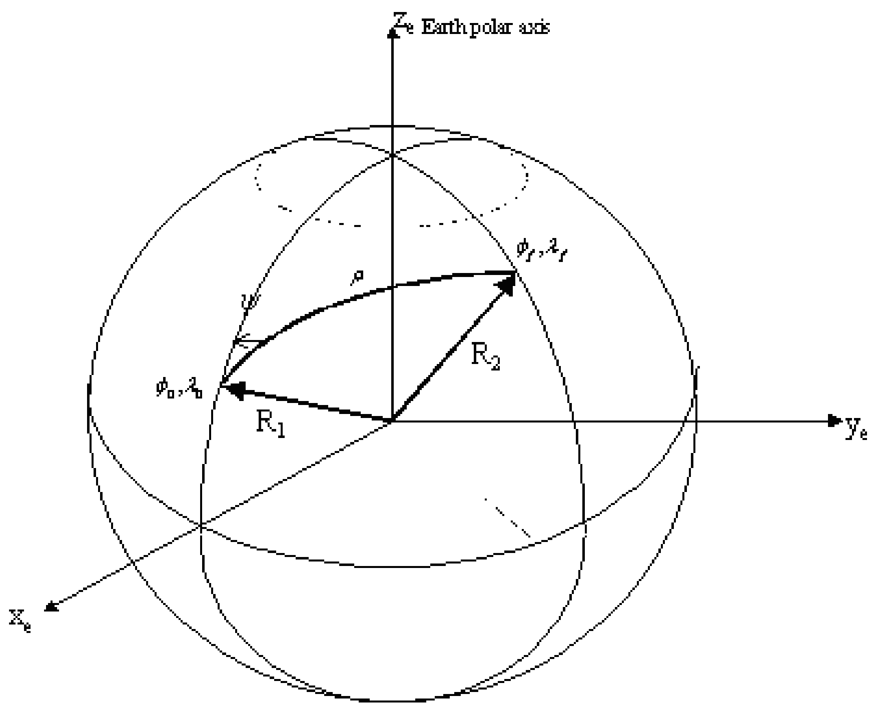


Figure 3.10: The Earth's reference frame

to refer to the position and velocity of the vehicle to an earth-fixed Cartesian coordinate frame which rotates with the earth. In such a coordinate frame, the equations of motion must account for the rotation of the coordinate frame. This is accomplished by using the Law of Coriolis.

The angular velocity ω of the geographic frame in inertial space is the sum of the angular velocity of the geographic frame with respect to earth and the angular velocity of the earth with respect to inertial space Ω where:

$$\Omega_{ie} = \Omega_{ie}cos\phi j + \Omega_{ie}sin\phi k \tag{3.16}$$

where i is the unit vector along x-axis, j is the unit vector along the y-axis and k is the unit vector along the z-axis. We can assume that the upward component is compensated for by the aircraft autopilot system.

$$North_{(y)} = \dot{y}t - \omega\dot{x}t^2 sin(\lambda) \tag{3.17}$$

$$East_{(x)} = (1/3)\omega g t^3 cos(\lambda) - \omega t^2(\dot{z_0}cos)\lambda) - \dot{y_0}sin(\lambda)) + \dot{x_0}t \tag{3.18}$$

where λ = latitude, g = earth's gravitational acceleration ($9.8m/sec^2$) and $\omega = 7.3 \times 10^{-5}$ is the angular speed of the earth's rotation.

3.2.7.2 Navigation Mechanization Equations

The objective of navigation mechanization is to generate position and velocity by computing the distance between two points on the surface of the earth. In this thesis, the navigation mechanization equations are based on the geographic implementation which calculates the latitude and longitude rate directly as functions of computed north and east velocity components. These components are integrated over the time interval to calculate the new latitude, longitude and altitude of the aircraft that is, its new position. The equations assume an elliptical earth model and will not work near poles.

The speed used is the ground speed rather than airspeed for locating the aircraft in an earth fixed co-ordinate frame. To be able to determine the groundspeed, the wind speed and direction are required in addition to airspeed. The wind triangle in Figure 3.11 shows the relationship between groundspeed and the wind speed.

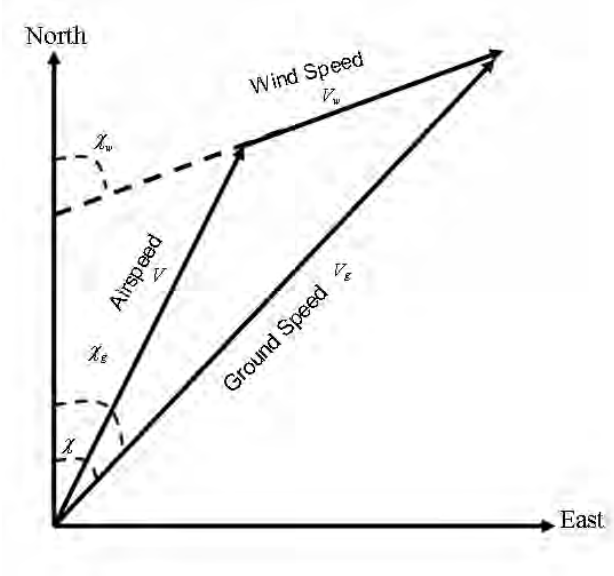


Figure 3.11: The wind triangle showing the vector representation of true airspeed, wind speed and ground speed.

In ATOMS, trajectory calculations are performed within the context of great circle en-route flight. The great circle distance is defined by the dot product of the origin and the departure and arrival points:

$$\ell = R \times cos^{-1}\{(sin\phi_i \times sin\phi_f + [cos(\tau_f - \tau_i)] \times cos\lambda_i \times cos(\lambda_f\} \tag{3.19}$$

where (λ_i, τ_i) and (λ_f, τ_f) are the co-ordinates of the departure and arrival points specified in terms of their respective latitudes and longitudes and R is the radius of the earth. To calculate the track of the aircraft, two terms are defined. Course angle, which is the planned path and track angle, which is the actual path as shown in Figure 3.11. It is assumed that the aircraft's autopilot corrects its track and there is no deviation from its planned track. Thus χ_g is taken as the track angle. Using the present aircraft position (λ, τ) and the destination position (λ_f, τ_f), an equation is developed where a new great circle course is set from the present position to destination as follows:

$$\chi_g = tan^{-1}\{\frac{sin(\tau_f - \tau) \times cos\lambda_f}{sin\lambda_f cos\lambda - sin\lambda cos\lambda_f \times cos(\tau_f - \tau)} \tag{3.20}$$

Using the heading and velocities, the kinematics equations of motion of a point-mass model given below (3.21, 3.22, 3.23) are calculated and then integrated

$$\lambda = \frac{1}{R} \times V_g \times cos\chi_g \tag{3.21}$$

$$\tau = \frac{1}{Rcos\lambda} \times V_g \times sin\chi_g \tag{3.22}$$

$$h = V_{climb} \tag{3.23}$$

where, λ is latitude, τ is longitude, h is geometric altitude, R is the mean radius of the Earth, V_g is the groundspeed, χ_g is the track angle of the ground relative velocity vector defined with respect to local north and V_{climb} is the climb rate.
The ground velocity is resultant of the horizontal components of the air mass-relative velocity and wind velocity. The magnitude of the ground velocity vector is the groundspeed V_g. The magnitude of the air speed is V, and χ is the heading angle. The magnitude of the horizontal components of the wind velocity is given by W_n (northerly component: $V_w cos\chi_w$) and W_e (easterly component: $V_w sin\chi_w$), where V_w is wind speed and χ_w is the wind heading angle .

Thus, to predict the location of the aircraft between current time t_0 and a future time t_p, the following equations are used:

$$\lambda(t_p) = \lambda(t_0) + \frac{1}{R} \int_{t_0}^{t_p} V_g cos\chi_g dt \tag{3.24}$$

$$\tau(t_p) = \tau(t_0) + \frac{1}{R} \int_{t_0}^{t_p} \frac{V_g sin\chi_g}{cos\lambda} dt \tag{3.25}$$

$$h(t_p) = h(t_0) + \int_{t_0}^{t_p} V_{climb} dt \tag{3.26}$$

Equations 3.24 and 3.25 require the estimation of V_g. Taking the wind correction angle χ_c and the track relative wind heading $\chi_{wg} = \chi_w - \chi_g$ into account, the ground velocity can be calculated as follows:

$$V_g = V cos\{sin^{-1}[\frac{V_w}{V} sin\chi_{wg}]\} + V_w cos\chi_{wg} \tag{3.27}$$

The first order Euler integration method used gives us an adequate approximation. The time interval (1 second) for integration is small enough to yield approximations that are appropriate to simulation.

3.2.8 Trajectory Modeling

ATOMS models the 4D trajectory of the aircraft from its initial position, using either flight plan based routing or Great Circle routing (GCR) for Free Flight. In GCR the FMS extracts the origin point or the activation point and the destination or the deactivation point of the flight, then using the airspace database calculates the position coordinates of these points. Whereas in flight plan based routing the FMS obtains the entire route which consists of waypoint/navigational fixes, and then calculates the position coordinates for each of them forming an ordered pair of coordinate points representing the flight trajectory. By using Great Circle equations the course angle between these points is determined and the aircraft is navigated from one waypoint to another.

Thus Free Flight can be seen as a special case of flight plan route flying, where all the middle waypoints are eliminated and the aircraft flies from origin to destination as a great circle route subject to aircraft performance parameters. While generating aircraft trajectory based on great circle routing, every cell of the discretized airspace through which the aircraft traverses is recorded along with the ETA in the flight plan. In case of any deviation from trajectory this cell plan is regenerated. This is done to model the intent information and the propagation of future trajectory change points to neighboring aircraft.

The aircraft trajectory is generated as follows:

1. The predicted position is set to the current position $\lambda = \lambda_f$, $\tau = \tau_f$ and $h = h_f$, where (λ,τ,h) are current position lat, lon and alt and $(\lambda_f,\tau_f,h_f\ )$ are predicted position lat,lon and alt.

2. The time is set $t = t_0$.

3. The wind components $W_n(\lambda,\tau,h)$, $W_e(\lambda,\tau,h)$ are obtained from the wind meteorological data where W_n and W_e are northerly and easterly components of the wind

respectively.

The wind speed V_w and wind direction χ_w are computed using the following equations respectively,

$$V_w = (W_n^2 + W_e^2)^{0.5} \tag{3.28}$$

$$\chi_w = \arctan(W_e, W_n) \tag{3.29}$$

4. Then the great circle heading χ_g is computed by the great circle track equation

$$\chi_g = \arctan\{\frac{\sin(\tau_f - \tau)\cos\lambda_f}{\sin\lambda_f\cos\lambda - \sin\lambda\cos\lambda_f\cos(\tau_f - \tau)}\} \tag{3.30}$$

5. Then the ground velocity is computed as:

$$V_g = V\cos\{\arcsin[(V_w/V)\sin(\chi_{wg})]\} + V_w\cos(\chi_{wg}) \tag{3.31}$$

where track relative wind heading $\chi_{wg} = \chi_w$ - χ_g and V is magnitude of the air speed.

6. By substituting these components into kinematic equation of motion given by:

$$\lambda = \frac{1}{R} \times V_g\cos\chi_g \tag{3.32}$$

$$\tau = \frac{1}{R\cos\lambda} \times V_g\sin\chi_g \tag{3.33}$$

$$h = V_{climb} \tag{3.34}$$

and integrating them using first order Euler integration for a time step of $t = 1$s , the new latitude, longitude and altitude are computed.

7. This process is repeated till the entire trajectory of the aircraft is generated.

3.2.8.1 Effect of wind on groundspeed

Ground speed is the speed of an aircraft relative to the ground. It is the sum of the aircraft's true airspeed and the current wind and weather conditions. A positive headwind component causes the ground speed to decrease, and a negative headwind component (or tail wind component) causes the ground speed to increase; while the cross wind component always causes the ground speed to decrease no matter the cross wind component comes from left or right (Eshelby, 2000). However, the reduction of ground speed due to the crosswind component is much smaller than that due to headwind component. An aircraft's ground speed is computed based on the true airspeed and the wind speed using the vector representation with corrections made for Coriolis effect. The two vectors, true airspeed and wind speed, determine the resultant ground speed vector as shown in Figure 3.11.

The ground speed is computed as follows (algorithm 1): First the direction of rotation of earth (Omega direction) with reference to aircraft heading is determined. If heading of the aircraft is greater than 180 degrees and less than 360 degrees then omega direction is positive else it is negative. Then the Coriolis effect is computed by multiplying the rotation rate of earth Ω with positive or negative magnitude depending upon the omega direction. Then the ground speed is compute by taking into consideration the northerly and easterly component and correcting for Coriolis effect. A long distance Perth-Darwin

Require: Aircraft true air Speed(TAS), altitude, heading, latitude, Wind Speed(WS)(easterly and northerly components)

if Heading $\geqslant 180^0$ AND Heading $\leqslant 360^0$ **then**

  $\Omega_{direction}$ = 1

else

  $\Omega_{direction}$ = -1

end if

Coriolis = $\Omega_{direction} \times \Omega$

$TAS_{North} = TAS \times cos(heading)$

$TAS_{East} = TAS \times sin(heading)$

$GroundSpeed_{North} = TAS_{North} + WS_{North}$

$GroundSpeed_{East} = TAS_{East} + WS_{East} + Corriollis \times cos(latitude)$

$GroundSpeed = (\sqrt{(GroundSpeed^2_{North}) + GroundSpeed^2_{East}})$

Algorithm 1: Computing the ground speed of the aircraft

flight with B747-400 aircraft is simulated to see the effect of wind on the ground speed. The result is reported in validation section.

3.2.8.2 Steering and Guidance

Steering involves determination of the change in aircraft motion required to achieve a desired result. The steering algorithm in ATOMS provides steering commands and related information to the mechanization equations. The steering algorithm performs the following operations:

- Convert the navigation equations output into course referenced quantities.

  1. Convert aircraft's velocity and position of the system into a North-East-Up right hand orthogonal system.
  2. Compute the desired parameters for the defined course.
  3. Calculates distance to go, time to go, cross track error and track angle error.

- Provide the steering parameters to the aircraft mechanization equations.

- Compute intercept steering parameters that account for anticipated coordinated turns and subsequent course changes.

The mechanization equations are used to generate trajectory data in the geographic coordinate frame. Great Circle Steering is accomplished by using the mechanization equations to generate trajectory data in the geographic coordinate frame. To steer aircraft between waypoints, a method is implemented for course (track) alteration upon the arrival of the aircraft at a waypoint and is ready to turn towards the next waypoint. The method used is the great circle direct steering method where the aircraft is steered to a way point using the great circle track and distance calculations. This means that the aircraft flies a great circle path directly from its current position (waypoint 1) to its destination (waypoint 2) as shown in Figure 3.13.

3.2.9 Flight Management System and the Autopilot

The objective of the FMS module is to guide the aircraft along the desired flight path. In general the FMS performs three specific functions in ATOMS. The first is to is to generate the flight profile based on the flight plan, this include the cruise level altitude, computation of TOC, TOD, ETA at each waypoint, heading and speed at each waypoint etc. The second is to accurately guide the aircraft along a specific path through a series of 2D or 3D waypoints. The third is to meet the required time of arrival at each waypoint. The FMS holds a database of information such as location of airports, navaids, waypoints, aircraft performance parameters, wind and atmospheric information. The FMS integrates avionics data to acquire current position, velocity and altitude, and generates heading and altitude commands to the autopilot module.

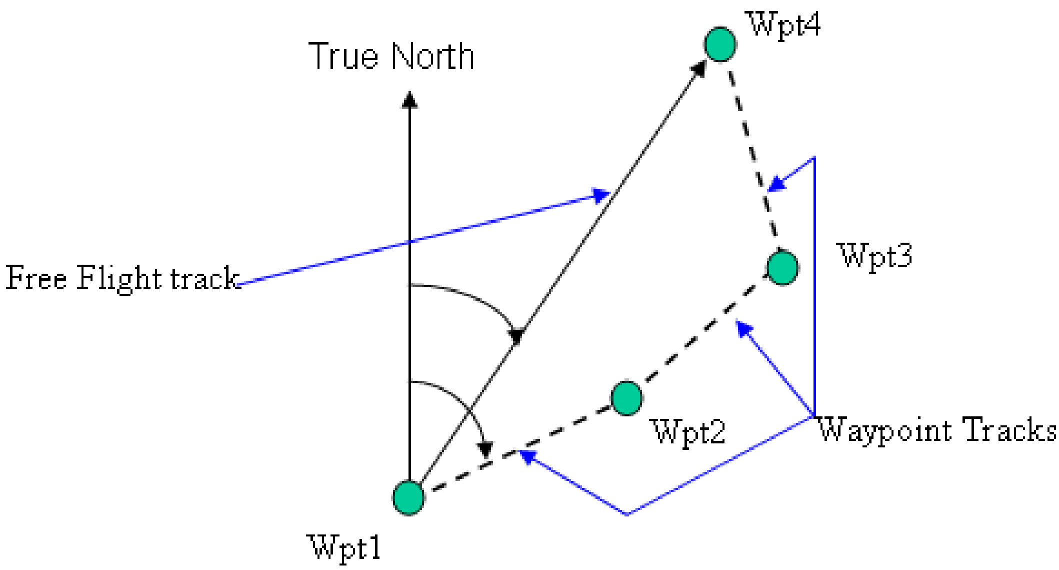


Figure 3.12: Great Circle Steering between multiple waypoints

At the start of simulation, for a given flight, a flight plan is loaded by specifying the origin, destination, airways, waypoints, altitudes and estimated en route times (with wind speed and Coriolis calculations) as well as an activation airport or waypoint. The departure and arrival waypoints have pre-specified airspeed and altitude constraints as part of the navigation requirements. Crossing speeds, altitudes, and times for en route way points are then computed using the aircraft performance database.

Given a flight plan, the FMS computes the flight profiles with pre-specified arrival times at the destination airport or deactivation point in the airspace in 4D navigation from one airport to another and then the autopilot flies the aircraft along that path. To achieve this, the FMS selects the most economic speed schedules for each phase of the flight (take-off, climb, cruise, descent and approach), predicts the complex vertical and horizontal profile that the aircraft would fly and when connected with the autopilot of the aircraft, controls the aircraft along the 3D flight plan. For the time dimension, the FMS computes the speed schedules and the flight path based on a required time of arrival at a selected point along the flight path. TOC and TOD are calculated by forward integrating from the origin and reverse integrating from the destination respectively using the aircraft equations of motion and the appropriate aircraft model. These two points are used to highlight the start and end of the cruise phase.

FMS provides the autopilot the necessary heading vectors, speed constraints, and climb/descent instructions. The FMS also interprets and implements the instructions received from the ATCs via the simulated communication link.

Autopilot is initialized by setting the heading, speed and altitude as follows, if the aircraft is originating from outside the airspace boundary then current position is set as the activation waypoint at the airspace boundary. If the aircraft is originating from within the airspace then the departure airport coordinates are set as the current position. Based on the location of the next waypoint or TOC the heading is initialized. The phase of the flight (take-off, climb, cruise, descent, approach) is set based on its current position. Then using altitude and phase of flight, airspeed is deduced from the aircraft performance database. Altitude is initialized as departure airport altitude if the flight is originating from within the airspace, else to the cruising level of the flight plan. Lateral navigation (LNAV) and vertical navigation (VNAV) modes are set, next altitude is initialized to the altitude of the next waypoint in the flight plan or to TOC (Free Flight). In the cruising phase as the flight gets close to a waypoint the autopilot starts checking the bearing of the waypoint with respect to the aircraft. If the bearing of the waypoint from the aircraft is greater than 90° and less than 270°, then the waypoint is deleted from the flight plan, and the heading is updated with the next waypoint heading in the FMS.

3.2.10 Aircraft Maneuver Modeling

3.2.10.1 Aircraft Acceleration Model

For all aircraft, a standard acceleration model is used where aircraft assume a linear acceleration/deceleration profile. The maximum longitudinal acceleration shall be taken as 2.0 fps^2 or 0.67 m/s^2 and the maximum normal acceleration, which deals with rate of climb and descent, shall be taken as $5.0 fps^2$ or $1.67 m/s^2$.

3.2.10.2 Aircraft Bank and Turn Model

For all aircraft, a standard bank model is used. There are two levels of bank, one is a standard bank and the other is a maximum bank angle. During all turns, the aircraft shall maintain standard rate of turn unless the FMS requires the aircraft to perform a maximum bank angle. The nominal bank angle for civil aircraft in all phases of flight except take off and landing is 35 deg. The maximum bank angles for civil aircraft during all phases of flight except take off, landing and hold is 45 deg. The maximum bank angle for civil aircraft during a hold is 30 deg (Eurocontrol, 2004). As shown in Figure 3.13

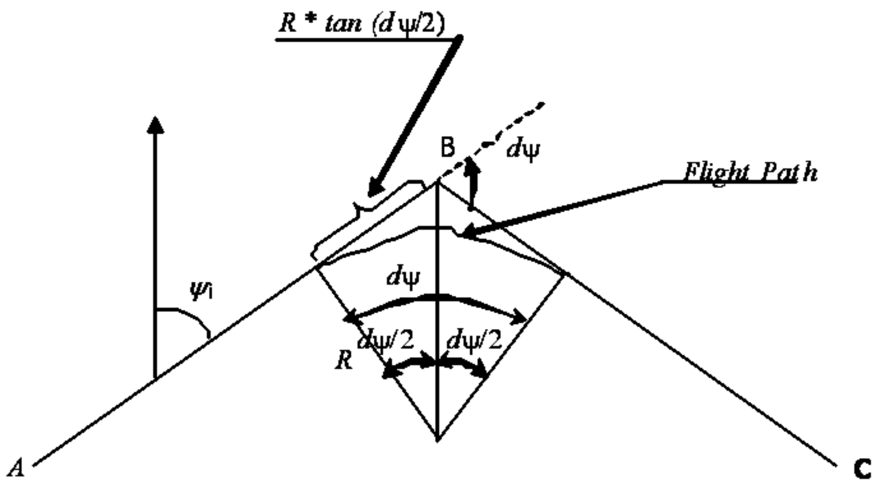


Figure 3.13: Aircraft turn maneuver.

The formulas in polar coordinates for the calculation of the radius of turn is given by

$$R = V \times \frac{d\psi}{dt} \tag{3.35}$$

where R is the radius of turn and V is the velocity(m/s) and ψ (radians/s) is the rate of change in track.

3.2.11 Approach and Landing

The approach and landing phases are very complex as aircraft fly with different throttle and flap settings. For landing, the aerodynamic model for the drag coefficient is extended to cover the effect of Mach number. This is based on the drag polar as formulated in BADA and is extended with drag polar coefficients for landing and approach flap settings and a drag coefficient increase due to undercarriage and speed brakes. For the idle thrust level a simplified linear model is used, which depends on Mach number, altitude and temperature correction. The aircraft descends to touchdown with a glide slope angle of 3 degrees along an optimized lateral flight path as defined by the Standard Terminal Arrival Rules (STARs). This horizontal segment in approach phase is maintained at 3000 ft. For final approach speed computations the minimum landing speed, V_{min}, is calculated from the following BADA equation:

$$V_{min} = 1.3 \times V_{stall} \frac{m}{m_{ref}} + 10 \quad (3.36)$$

where 1.3 is a factor recommended by the BADA manual for all aircraft operations, V_{stall} is the stall speed at the reference mass m_{ref}, and m is the simulated aircraft mass.

For deceleration rate in the approach phase, the BADA manual recommends a maximum longitudinal acceleration/deceleration of $2.0ft/s^2$ (118 kt/sec). This value was implemented in ATOMS as the maximum deceleration regardless of flight-path angle. To compensate for a descending aircraft, γ was subtracted from the maximum longitudinal acceleration, where g is the acceleration due to gravity in kt/sec and γ is the flight path angle in radians; for a 3 degrees flight path angle, the $g \times \gamma$ term is approximately equal to 1.0 kt/sec. The deceleration rates are also affected by the flap configuration. Since the landing flap configuration creates more drag than the cruise configuration, thus higher deceleration rates are expected. The BADA flap configuration for approach and landing is used. Simulations are performed for a B737-400 aircraft approaching Sydney (Kings-Ford Smith) via MANFA ONE arrival, runway 16R from BOREE fix and the results are reported in the experiment and validation section.

3.2.12 Fuel Flow Modeling

The fuel burn of an aircraft primarily depends on airframe drag, engine specific fuel consumption, distance of the route to be flown, vertical flight path and aircraft weight (Eshelby, 2000). Thrust specific fuel consumption (TSFC), or simply the specific fuel consumption (SFC) is a measure of engine efficiency, defined as fuel flow rate in pounds per hour divided by engine thrust in pounds of force (lb/hr/lb). Hence, the lower the ratio the lower the fuel flow rate for a given thrust the lower the fuel consumption and more efficient the engine. Two methods of fuel flow computations are implemented. In first method, thrust is calculated using aerodynamic equations and in the other, fuel flow is computed with dynamic weight reduction. The BADA performance table fuel flow values are used as primary means of fuel flow validation.

3.2.12.1 Aerodynamic Fuel Consumption Modeling

ATOMS uses BADA thrust specific fuel consumption model where the thrust specific fuel consumption, in kg/minute/kN is specified as a function of true airspeed V_{TAS}:

$$\eta = C_{f1} \times (1 + \frac{V_{TAS}}{C_{f2}}) \tag{3.37}$$

The nominal fuel flow, f (kg/minute), is calculated using the thrust T based on aerodynamic parameters as computed in previous section:

- Fuel Flow for Climb and Descent/Idle phases is computed as

$$f \equiv -\frac{dW}{dt} = \eta \times T \tag{3.38}$$

- Fuel Flow for Cruise phase is given by

$$f = \eta \times T \times C_{fcr} \tag{3.39}$$

where C_{fcr} is the cruise fuel flow correction coefficient.

3.2.12.2 Fuel Flow with Weight Reduction

Reduction of aircraft weight due to fuel burn is quite significant. The BADA performance table provides three fuel flow values according to the low, nominal and high mass levels for cruise phase and single fuel flow values for the climb and descent phases for each flight level. By interpolating the Fuel Flow values at different flight and different mass levels, and continuously reducing the aircraft mass by the fuel consumption amount at each time step, we can achieve higher fidelity fuel consumption model which takes into account the dynamic weight changes due to fuel burn. This dynamic fuel model is implemented using interpolation as follows:

$$W(t+1) = W(t) - W_f t \tag{3.40}$$

where $W_f t = \frac{Interpolate_1(Alt, FuelFlow_{BADA})_{CL}}{Interpolate_2(W(t), Alt, FuelFlow_{BADA})_{CR}} Interpolate_1(Alt, FuelFlow_{BADA})_{DS}$

where W(t) and W(t+1) is the aircraft's weight at time step t and t+1, $W_f(t)$ is the aircraft's fuel flow at time t, Alt is the aircraft's altitude at time t, CR: cruise phase, CL:

climb phase, DS: descent phase, and $FuelFlow_{BADA}$ is the fuel values from the BADA performance table.

3.2.12.3 For climb and descent phases

For climb and descent phases, since only a single nominal fuel flow value is given for each flight level, the fuel flow at a particular altitude, Alt, is computed by interpolating the fuel values at the flight levels above and below Alt from BADA table regardless of the actual weight. The interpolation formulae is expressed as follows:

$$Interpolate_1(W_i, Alt, FuelFlow_{BADA}) = \alpha \times (FuelFlow(FL_A, W_i) - FuelFlow(FL_A, W_i)) \tag{3.41}$$

where the interpolation coefficient $\alpha = \frac{Alt-FL_B}{FL_A-FL_B}$ and B and A are the levels right below and above altitude Alt and FuelFlow(FL_i, W_j) is the BADA fuel value at flight level FL_i and mass level W_j, which is ignored during climb and descent phases.

3.2.12.4 For cruise phase

For cruise phase, fuel flow values at low, nominal and high mass are given for each flight level. Therefore, the fuel flow at a particular altitude Alt and weight W is computed by interpolating the fuel values, from BADA table, at the flight levels above and below Alt and mass levels lower and higher than the current mass. The interpolation formulae is expressed as follows:

$$Interpolate_2(W, Alt, FuelFlow_{BADA}) = \beta \times (FF(W_H) - FF(W_L) \tag{3.42}$$

where the interpolation coefficient $\beta = \frac{W-W_L}{W_H-W_L}$ and
$FF(W_i)$=Interpolate1(W_i, Alt, $FuelFlow_{BADA}$) is the interpolation of the BADA fuel values according to altitude, L and H are the mass level below and above the current weight W.

3.3 Graphical User Interface

ATOMS is a Graphical User Interface (GUI) driven simulator. It is also capable of running in console mode in a fast time. Through the GUI, the user can define the extent of the airspace to be simulated, generate a new set of flights with a varying set of traffic densities, can load and run a previously generated set of flights, and set the simulation run speed. All the desired controls, navigation and display properties of aircraft and ATCs are achieved through the GUI. Figure 3.14 shows the startup window of ATOMS. This

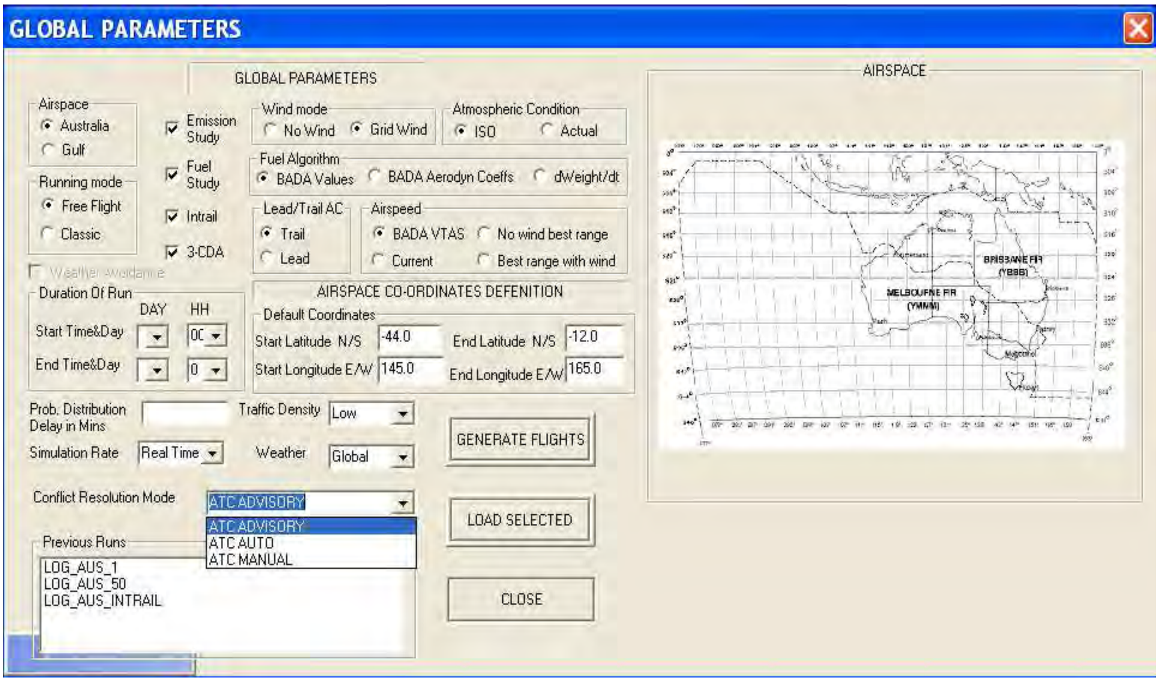


Figure 3.14: Parameter initialization and experiment setup window of ATOMS.

interface is used for setting the experiment and loading the traffic. It provides for running simulation in two different airspace viz Australia and the Gulf region. Experiments can be run for Free Flight as well as for Route based structures. The interface also provide for an experiment set ups for a variety of studies which are conducted in ATOMS such as In-Trail climb procedure, estimation of aviation emission etc. This window also provides control for running the simulation in real time or fast time. The main window of ATOMS consists of both airspace display and simulator operations control tool bar. Figure 3.15 shows the main simulator window and the airspace modeled, waypoints, airways, airports, aircraft and their intent track. It shows the Australian airspace modeled which extends from 48^0 south-75^0 east to 2^0 south-163^0 east. The airspace is divided into two Flight Information Regions (FIR) namely Brisbane (YBBB) FIR and Melbourne (YSSY) FIR. The vertical extent of the airspace modeled is 46,000 ft. Standard Terminal Arrival Rules (STARs) and Standard Instrument Departure (SID) procedures are used to model terminal area operations.

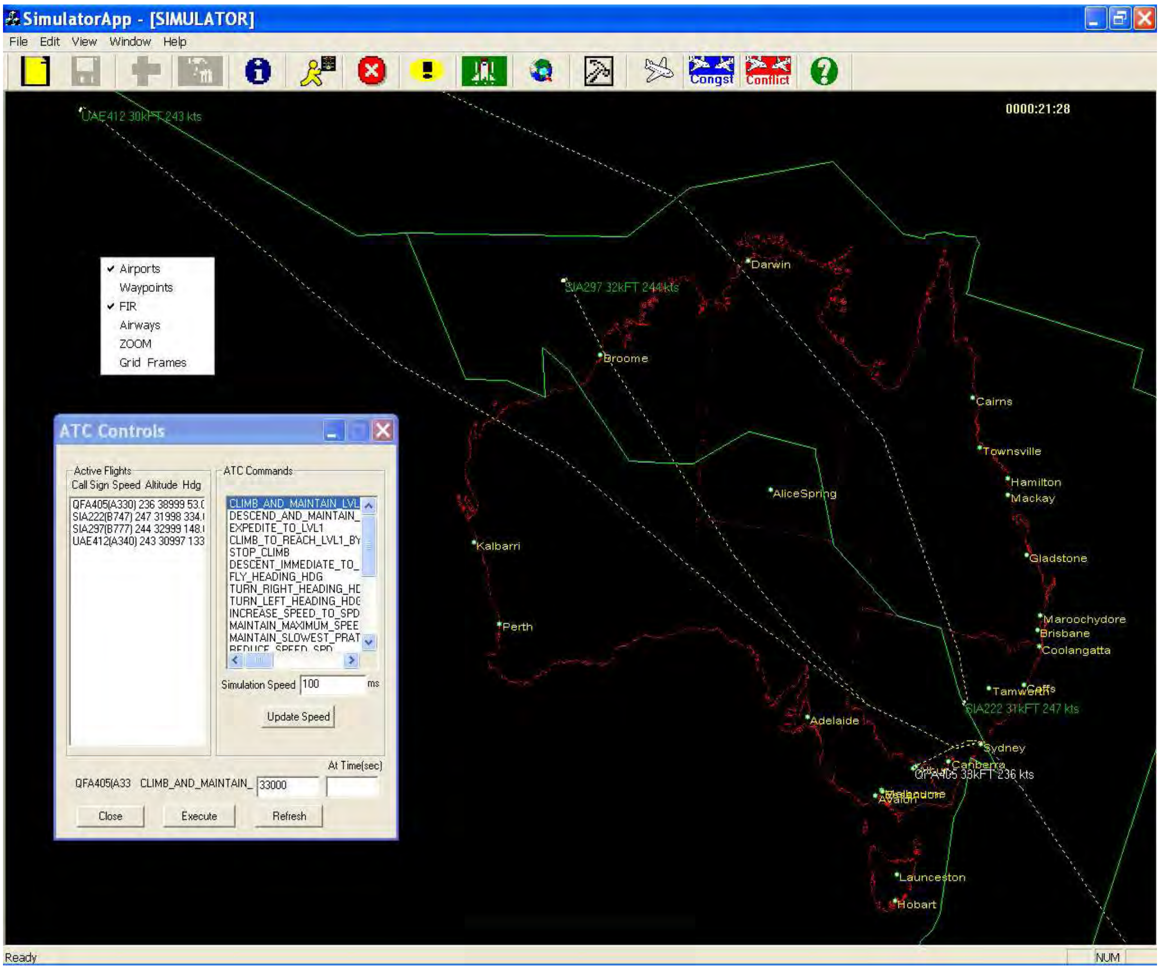


Figure 3.15: Main simulator window of ATOMS. ATC communication interface can be seen in the left side of the main window.

Four representative flights are also displayed with their call sign, speed and altitude information with color coded symbology for flight phase. The flights show their intended flight trajectory from current position to destination/deactivation point. The airspace display properties are controlled through user driven features. The user can pause, resume or terminate a simulation, airspace can zoom-in and zoom-out for any section and bad weather scenarios can also be created as described in a later section. Figure 3.16 top, shows the runway and ATC interface of ATOMS with flight QFA405 (B734) on the taxiway at Melbourne airport. Bottoms, the same flight landing at Sydney airport, runway 34L.

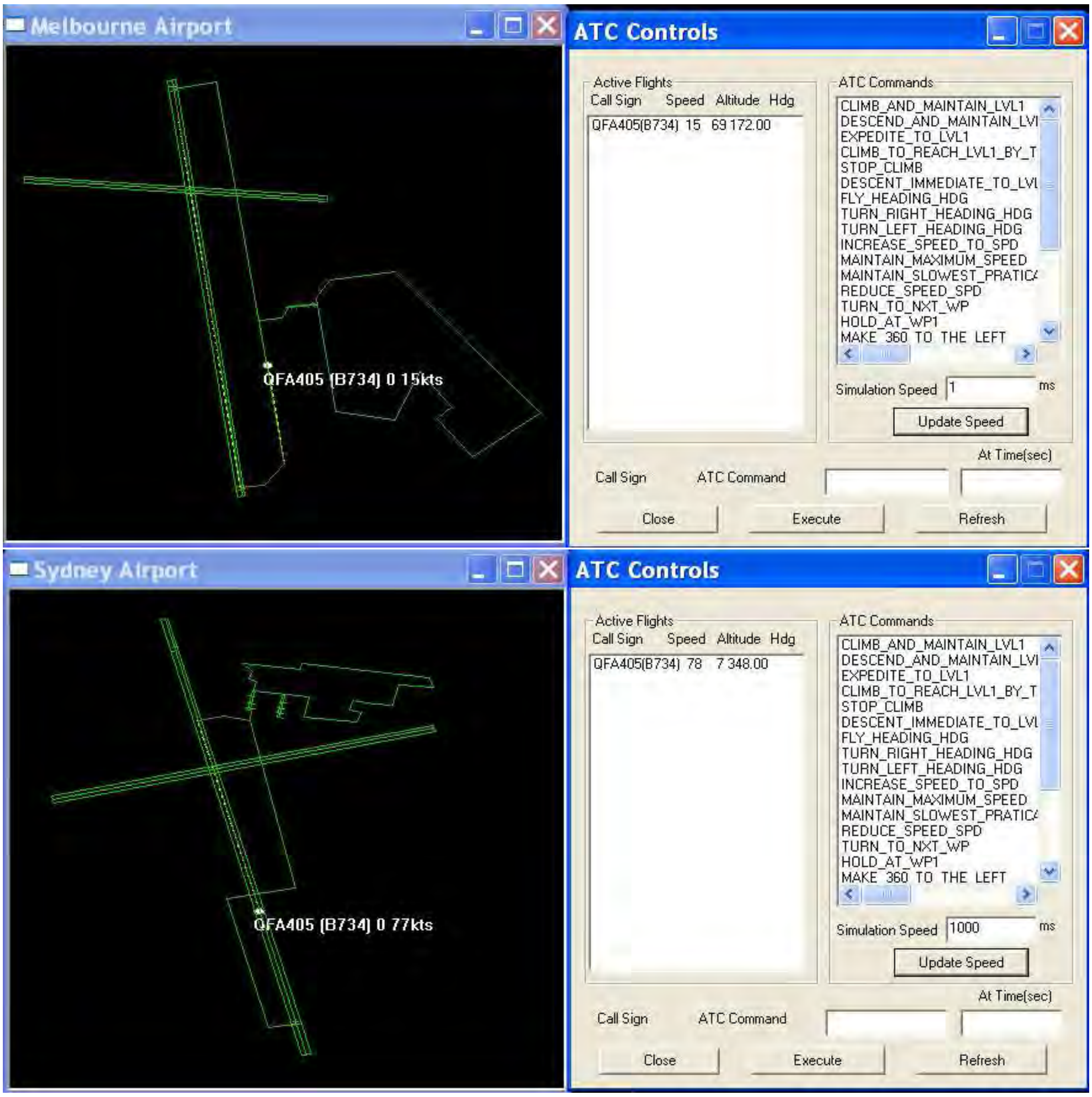


Figure 3.16: Runway interface in ATOMS showing departure from Melbourne Airport and the same flight landing at Sydney airport

3.3.1 Pilots and ATCs Interface

ATOMS provides a basic interface, for pilots and ATCs. Aircraft in ATOMS can be flown using autopilot mode as well as user driven mode. The interface for pilots can be seen in Figure 3.17 left, which provides the necessary features for navigation and autonomous control of aircraft. In the flight parameter section, pilots can see flight plan data, including origin, destination, flight level, speed etc. The autopilot section displays all the necessary autopilot information including fuel consumption and distance remaining to destination. It also provides for activation of weather radar, intent display and protected zone (5nmi) display. Pilots are also able to choose the CD&R algorithm depending upon the airspace complexity and their own preference, as well as being able to set the ADS-B range. The resolution advisory data is displayed as well as shown graphically in the CDTI interface to the pilots. Pilots can choose to resolve airborne conflicts and weather manually or through algorithm generated maneuvers.

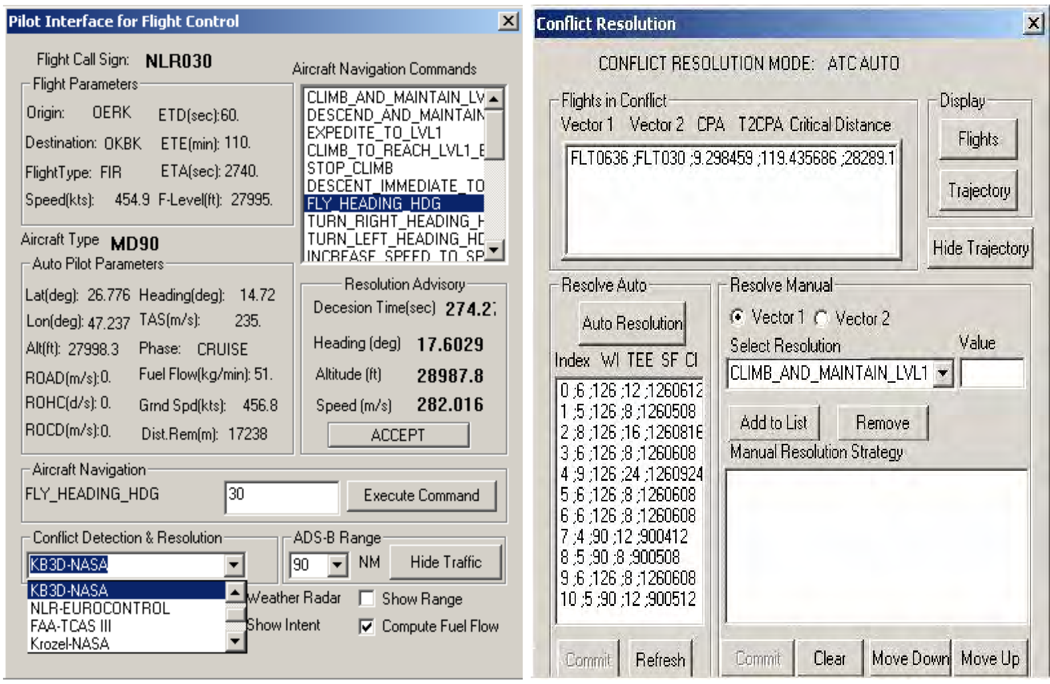


Figure 3.17: Pilots Interface (left) and ATCs interface (right) for decision support (CD&R) and navigation control

The interface for the ATCs comprise a simulated communication link as shown in Figure 3.15, left hand side and conflict resolution interface as shown in Figure 3.17 right. Through the simulated communication interface an ATC can monitor the active aircraft in the airspace and can perform the basic functionalities of ATCs. The aircraft responds to ATC directives received from the ATCs via this simulated communications interface. Once these directives are generated, they are interpreted by the FMS of the aircraft concerned, which then translates them into a series of autopilot actions depending on the type of directive, the actual aircraft state and the phase of flight. There is a simple short term collision avoidance (STCA) interface for the ATCs with conflict information data for two conflicting aircraft as well as various resolution maneuver options. The STCA logic is based on simple extrapolation of an aircraft's current state 20 minutes into future and checks for separation violation. The conflict resolution, which is rule based, can run in auto mode: where it can automatically transmit the resolution trajectories to the conflicting aircraft, advisory mode: where it generates resolution trajectories for the controllers to choose from, and manual mode: where ATCs can resolve the conflicts by issuing vectors. This interface is used for the controlled airspace simulation where pilots have a limited role in separation assurance.

3.4 Experiments & Validations

Validations have been performed empirically where we have compared the ATOMS generated data with the field radar data and with aircraft performance data from BADA. The airspace used for this study is the ICAO APAC region, namely Australian Airspace stretching in latitude from 2° to 48° south; and in longitude from 75° to 163° east (Air Services Australia, 2004). The simulation is set up by first loading the airspace under consideration as defined by the airspace boundaries, then loading of all the waypoints, airways, navaids and airports within the airspace, finally loading of flight plans where each flight is categorized as follows:

- For over-flying flights, the flight plan is activated at the first point on the airspace boundary at the flight's ETA at that point and deactivated at the last point of the airspace boundary and at the flight's ETA at that point (Type 0 Flight).
- If a flight originates outside the airspace and arrives at an airport within the airspace then the flight plan is activated at the first point on the airspace boundary at the flight's ETA at that point (Type 1 flight).
- If a flight is destined for an airport outside the airspace and originating at an airport

within the airspace then the flight plan is deactivated at the last point of the airspace boundary and at the flight's ETA at that point (Type 2 flight).

- If a flight originates and arrives from/at an airport within the area under consideration then the full flight plan is used (Type 3 Flight).

3.4.1 Validation of Atmospheric Model

The ISA atmospheric equations are programmed in C++ and the outputs is verified using the environment block of the MATLAB-Aeronautical Simulation Block Set (Aerosim) (Aerosim, 2005).

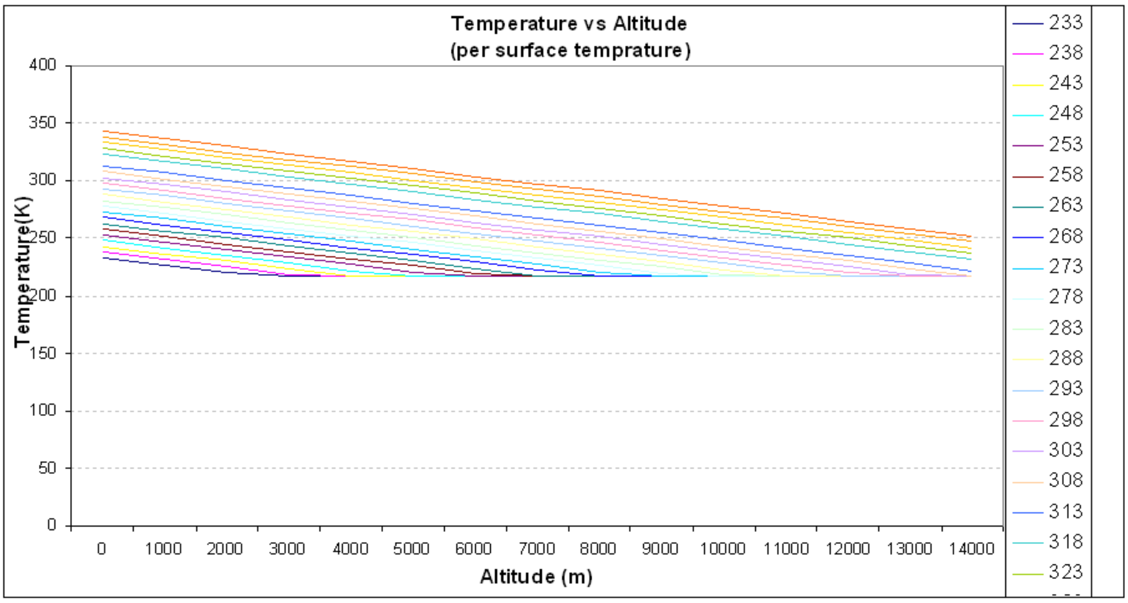


Figure 3.18: Temperature v/s Altitude plot per surface temperature

The atmospheric data for each flight level is stored in a 3D data structure, which can be accessed by other modules of the simulator. Figure 3.18, 3.19, 3.20, and 3.21 shows the temperature, density and pressure vs. altitude per surface temperature and variation in speed of sound vs. temperature generated by using ICAO standard atmosphere (ISA) (ICAO, 2004) equations modeled in ATOMS.

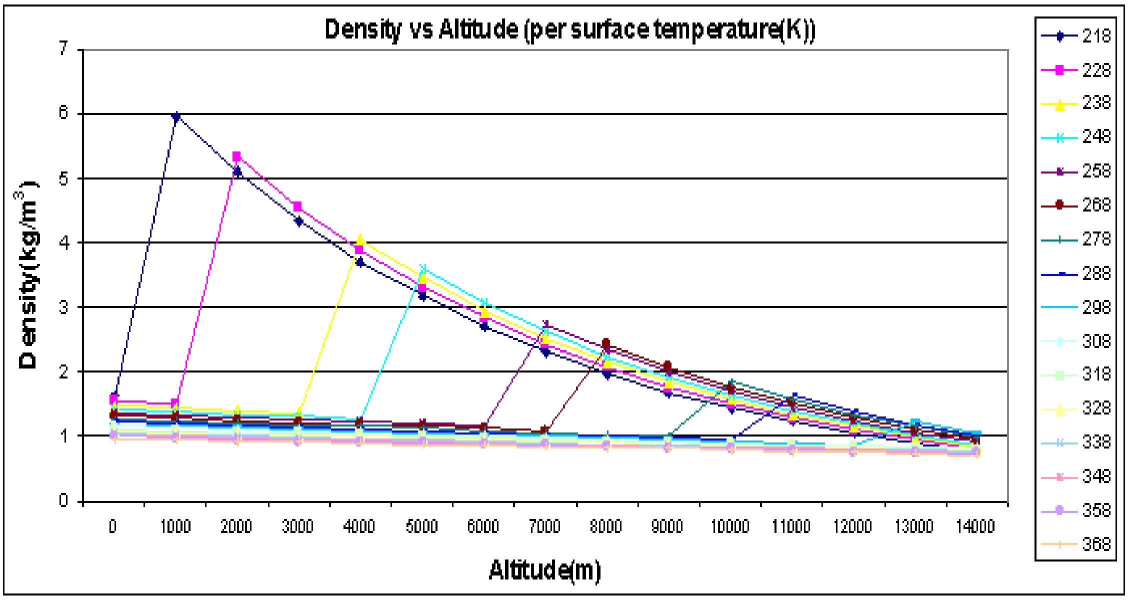


Figure 3.19: Air density v/s altitude per surface temperature.

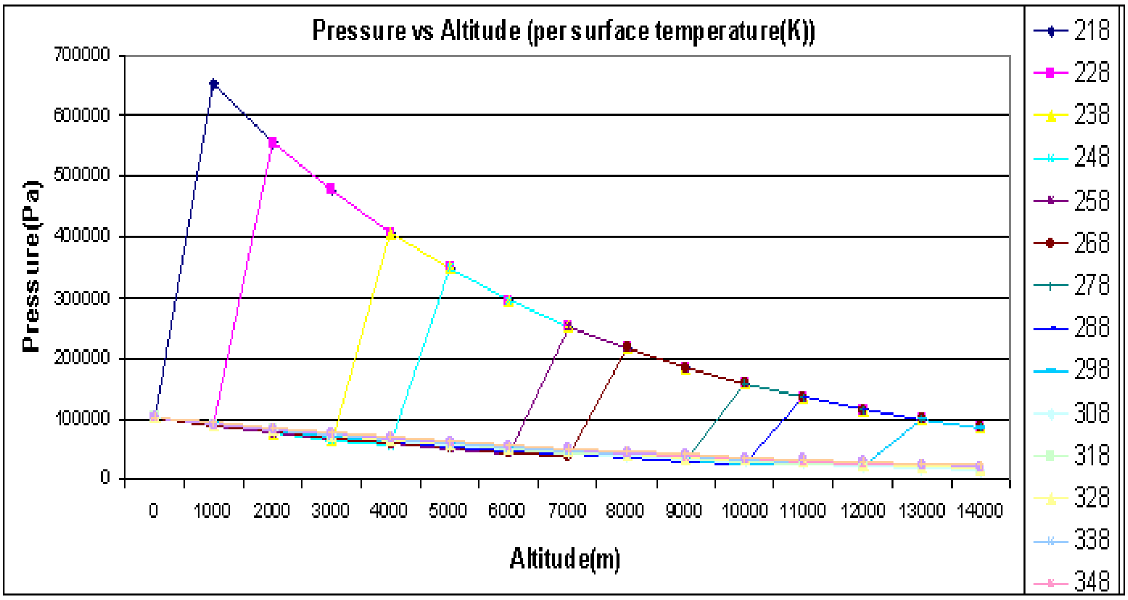


Figure 3.20: Pressure v/s altitude per surface temperature.

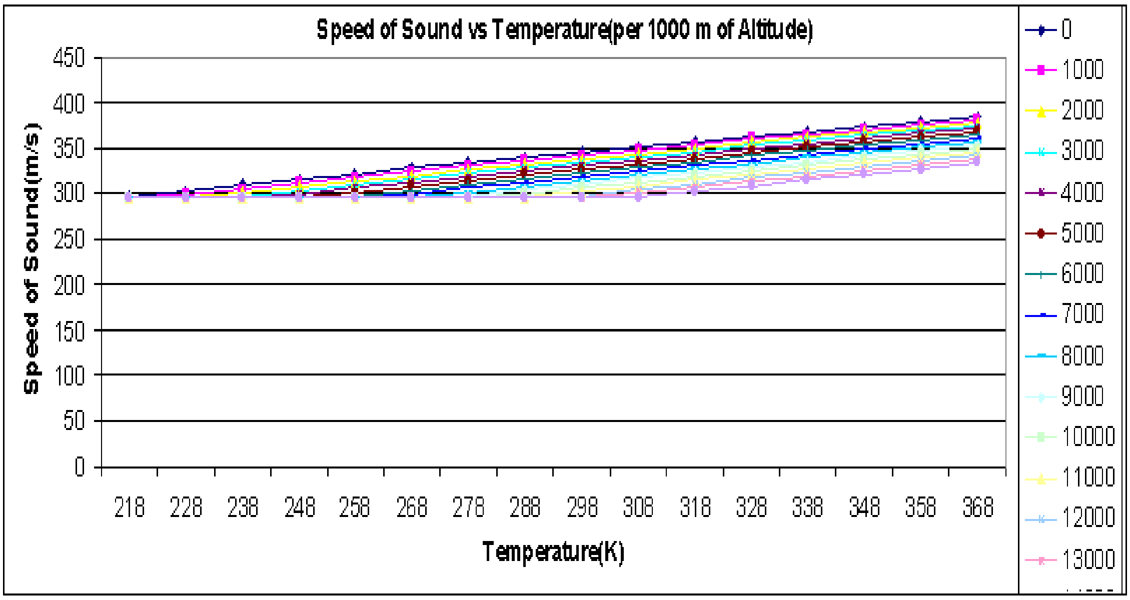


Figure 3.21: Variation in speed of sound vs temperature (per 1000m of altitude)

Table 3.2: Flight plans representing four different type of flights.

Call Sign	**SIA222**	**UAE412**	**SIA297**	**QFA405**
Origin	YSSY	OMDB	WSSS	YMML
Destination	WSSS	YSSY	NZCH	YSSY
Activation Point	YSSY	ELATI	ATMAP	YMML
Aircraft Type	B744	A340	B777	B734
Speed(kn)	480	473	474	458
Flight Level(00ft)	320	310	330	390
Est. Time Enroute(min)	466	779	515	61
Est. Time of Departure(s)	60	60	60	60
ETA at Destination(s)	26394	48908	34535	3673
Flight Type	2	1	0	3

3.4.2 Validation of FMS and Flight trajectory

For validation of the FMS and trajectory generated, flight plans of five hundred flights comprising of four different types of actual flights are selected and fed to the FMS module of ATOMS. As an example set, four typical representative flights are shown in Table 3.2 and their route element are shown in Table 3.3. The origin airport for the flights are YSSY (Sydney), OMDB (Dubai), WSSS (Singapore) and YMML (Melbourne). The destination airport are WSSS (Singapore), YSSY (Sydney), NZCH (ChristChurch-New Zealand) and YSSY (Sydney) respectively. Activation points ELATI and ATMAP are waypoints on the FIR boundary. The STAR routes are not included in the flight plan.

Table 3.3: The route element for each flight in the representative set comprising waypoints and navaids

Call Sign	Route(Waypoints and Navaids)
SIA222	RIC MDG WLG TAVEV TASHA LARAB KIKEM
UAE412	ELATI PIPOV EMVAS EGAVI NWN RUSAD UVUPU GTH CULIN
SIA297	ATMAP BRM PUGUT WR UVUPU GTH CULIN SY CAWLY PLUGA SULON
QFA405	HOPLA PUTTY DOSEL EBONY ARRAN NONUP CULIN TARAL RIVET TAMMI BOOGI

Table 3.4: Flight profile for a Type 3 flight

Call Sign:QFA405

Waypoint	Distance (m)	Hdg (rad)	Speed (m/s)	Lat (rad)	Lon (rad)	Alt (m)	ETE (sec)	ETA (sec)
YMML	15147.49	4.9598	115	-0.657698	2.528109	21	146	60
HOPLA	13715.70	5.9558	230	-0.657116	2.525201	2400	86	206
PUTTY	78910.11	0.9994	230	-0.655080	2.524328	4200	384	292
DOSEL	57133.78	0.9413	230	-0.648390	2.537418	9600	248	676
TOC	128819.57	0.9413	229	-0.643087	2.546484	11887	559	924
EBONY	63097.61	0.9421	230	-0.631227	2.566798	11887	274	1483
ARRAN	74904.26	0.9030	230	-0.625410	2.576688	11887	325	1757
NONUP	89093.13	0.9215	230	-0.618137	2.588032	11887	387	2082
TOD	36720.47	0.9215	229	-0.609691	2.601627	11887	159	2469
CULIN	56687.91	1.0948	228	-0.606211	2.607231	9600	248	2628
TARAL	51442.69	1.0415	197	-0.602139	2.616830	6700	261	2876
RIVET	41243.96	1.1042	172	-0.598066	2.625266	4000	239	3137
TAMMI	19285.51	1.0689	125	-0.595157	2.632247	2000	154	3376
BOOGI	17080.49	1.1213	119	-0.593703	2.635447	1100	143	3530
YSSY	0.00	0.000l	75	-0.592539	2.638356	7	0	3673

The FMS processes the flight plan and outputs a flight profile for the entire route for all the flights in the flight plan. A flight profile generated for a type 3 flight is shown in Table 3.4. In the tabulated data, distance implies distance to next waypoint from the current position, heading implies heading required to turn to next waypoint, speed implies speed at that particular waypoint, latitude (Lat) and longitude (Lon) of current waypoint, altitude (Alt) at the current waypoint, ETE implies estimated time enroute from current waypoint to next waypoint, ETA implies estimated time of arrival at current waypoint. Now to validate the trajectory generated, navigation and steering in ATOMS we plotted the trajectory flown by a simulated flight and validated it against the air traffic data provided by Air Services Australia which is obtained from their EUROCAT radar data processing system.

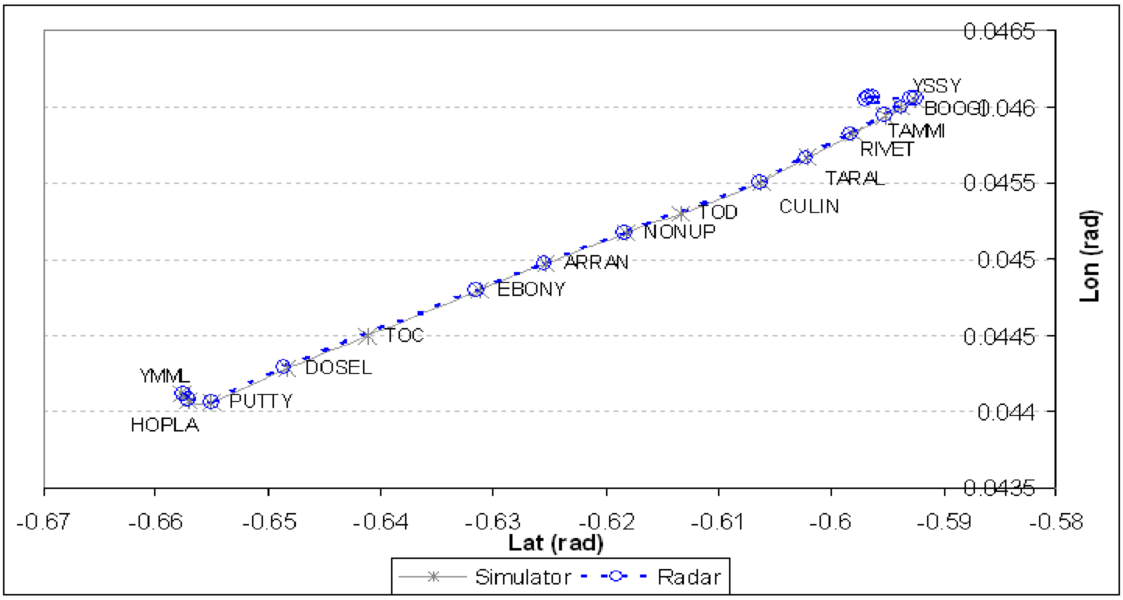


Figure 3.22: Trajectory of simulated flight (QFA405) in ATOMS compared with primary surveillance radar data for the same flight.

Table 3.5: FMS generated flight profile for a Type 3 flight in Free Flight

Call Sign:QFA405

Waypoint	**Distance** (m)	**Hdg** (rad)	**Speed** (m/s)	**Lat** (rad)	**Lon** (rad)	**Alt** (m)	**ETE** (sec)	**ETA** (sec)
YMML	330820.48	0.9250	115	-0.657698	2.528109	21	1434	60
TOC	190678.44	0.9609	236	-0.626427	2.579939	11887	807	1494
TOD	183514.51	0.9487	236	-0.609305	2.610006	11887	775	2301
YSSY	0	0.0001	74	-0.592539	2.638356	2	0	3076

Figure 3.22 shows, for flight QFA405, waypoints generated by the FMS and trajectory flown by the aircraft through autopilot control in ATOMS as compared to radar track data. Since the flight origin and destination are within the airspace, TOC and TOD can be seen along with the enroute waypoints and navaids, the autopilot flies the aircraft as per the FMS generated flight path. We then compared the altitude profile of flight QFA405, for ATOMS and radar data. As can be seen from Figure 3.23 that aircraft climbs correctly, as per its flight plan, to its cruise altitude, reaches its TOC, then enters into cruise phase following closely the actual radar data, upon reaching the TOD it transitions into descent phase, and finally into approach phase.

In Free Flight mode the FMS module of ATOMS eliminates the route data from the flight plan and performs the flight profiling using Great Circle as seen from flight QFA405 flight profile in Table 3.5, which has only origin, TOC, TOD and destination. Figure 3.24

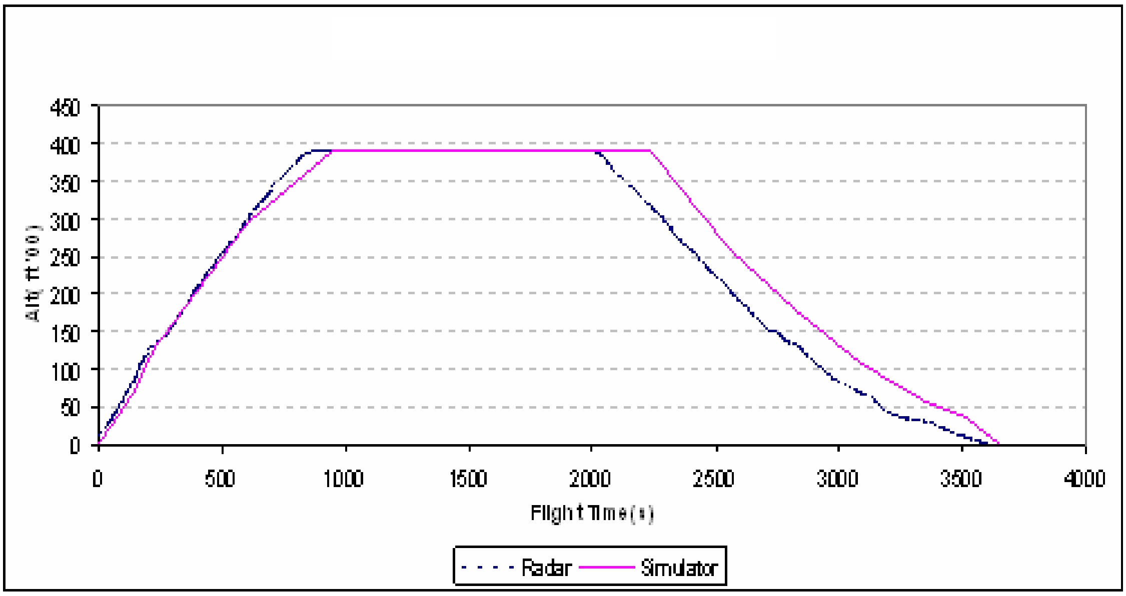


Figure 3.23: Altitude log of flight (QFA405) as recorded in ATOMS compared with the secondary surveillance radar data (altitude) for the same flight.

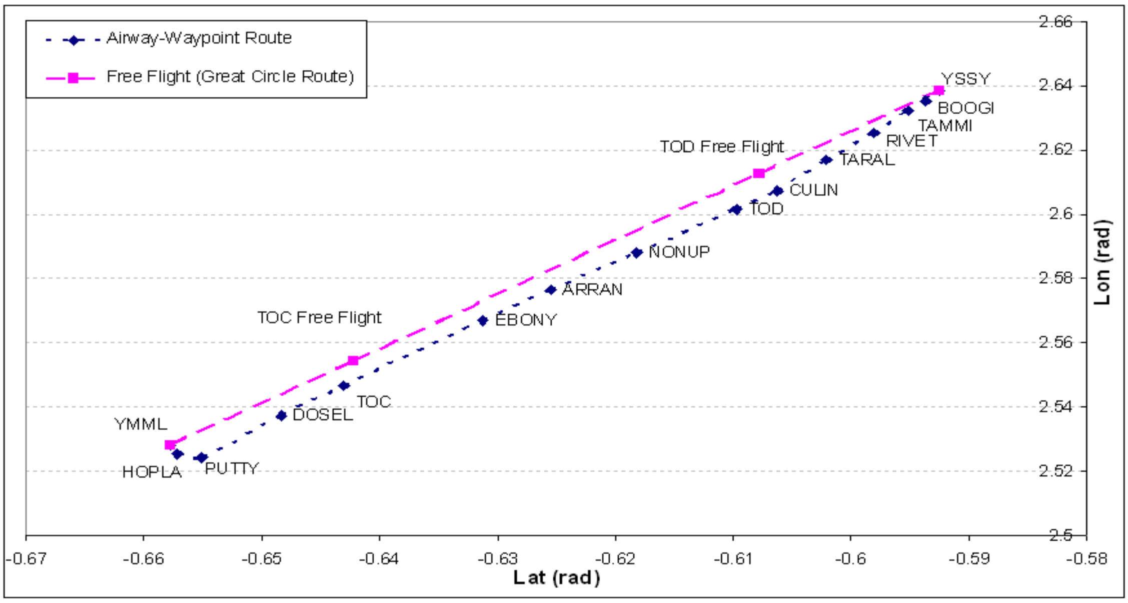


Figure 3.24: Free Flight route vs. Waypoint route for flight QFA405, Free Flight route can be seen as great circle route from origin to destination.

shows the aircraft route flown for flight QFA405 in a classic air traffic control environment and in Free Flight airspace. In Free Flight airspace the aircraft flies the great circle route from origin to destination, whereas in the classical model it follows the airways route, navigating waypoint to waypoint.

Figure 3.25shows the turn maneuver during the arrival maneuver for the given STAR route. It can be seen that ATOMS flight trajectory matches closely with the flight trajectory from the field data.

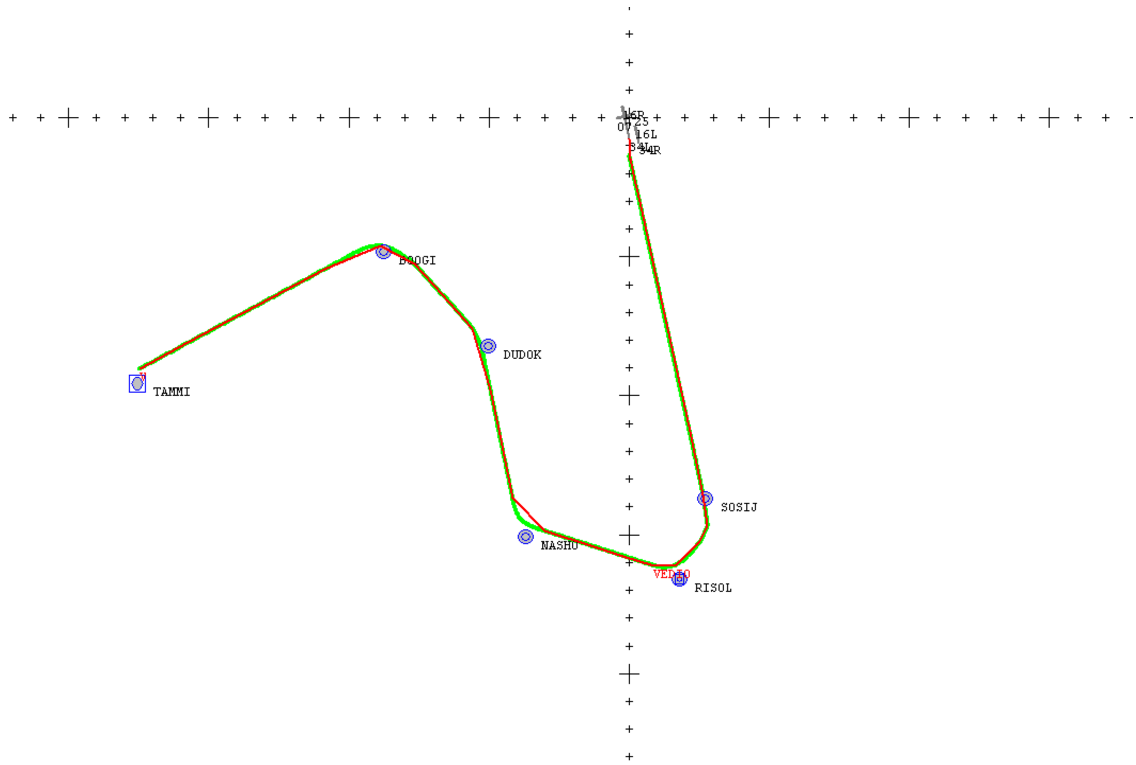


Figure 3.25: A turn maneuver validation during the approach and landing phase of flight. Green trajectory is from the actual flight field data and red trajectory is from ATOMS simulator. Flight landing at runway 34R, Sydney International Airport.

3.4.3 Flight Performance Characteristics in ATOMS

In the experiments concerning the flight performance check; a Sydney-Melbourne flight was simulated with a given operational flight plan as shown in Table 3.6. The aircraft has two CFM engines and has a cruise altitude of 35000 ft. The STAR route is included in the test flight.

Figure 3.26, top right, shows thrust profile for the entire flight. As can be seen from Figure 3.26, top right, that the thrust is maximum in the take off phase at around 150000 Newton. Then it gradually decreases to to 55000 Newton as aircraft climbs to its cruise phase. In cruise phase the thrust is 39000 Newton. In the descent phase the thrust drops to 2000 Newton, and then increase and decrease in the band of 8000 Newton to 40000

Table 3.6: Flight plan parameters for the test flight used in the experiment

Test Flight Parameters	
Origin	Melbourne
Destination	Sydney
Aircraft	B737-400
Engine type & code	CFM56-3C-1
No of engines	2
Waypoint Route	SBG AY MUSOP YAS BIK WELSH
STAR Route	BOREE BEROW OVILS MAJAR MANFA JAKLN LISHA SUZAN
Weight	Nominal (58000kg)
Cruise level speed	427kts
Cruise level altitude	35000ft
Est. Time Enroute	1 hrs 20 mins

Newton to maintain the speed and altitude profile. In the end due to reverse thrust it increases to 48000 Newton.

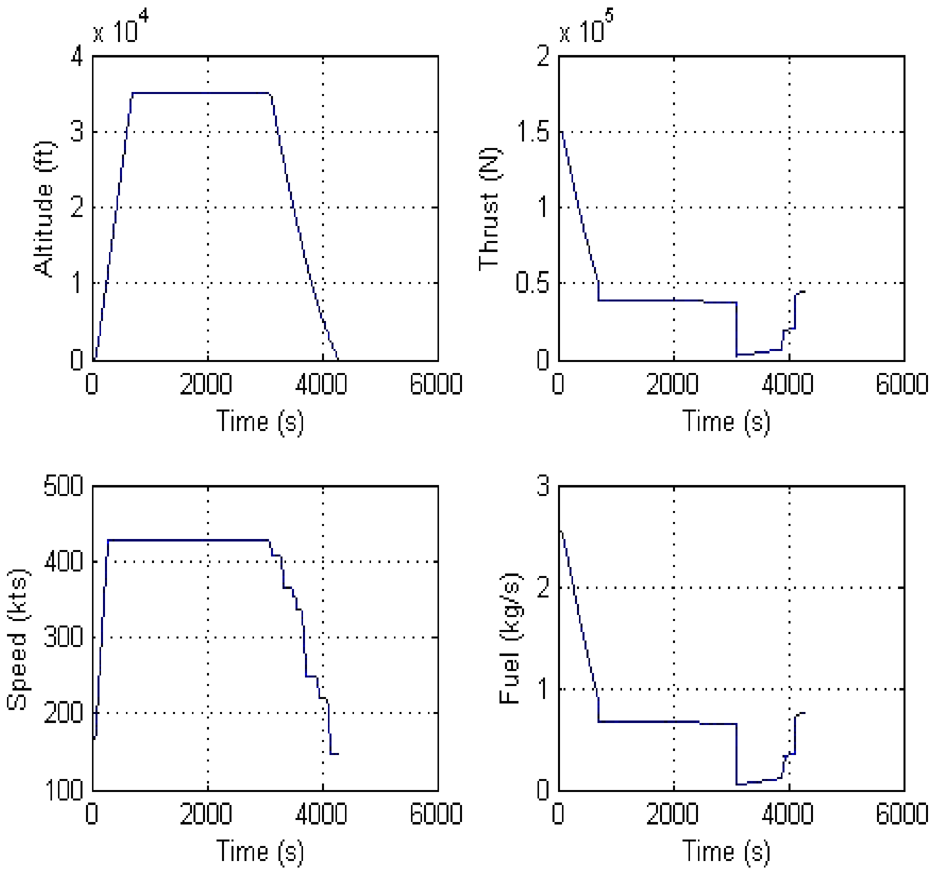


Figure 3.26: Fuel Flow, Engine thrust, True air speed, and altitude profile for the entire duration of flight.

Figure 3.26, bottom left, shows the speed profile of example flight. The cruise speed of aircraft as shown in figure is 430kts. As aircraft descents speed reduces to 310kts. In the landing phase the speed is 220kts and at touchdown the speed of the aircraft is 140kts.

Figure 3.26, bottom right, shows the total fuel consumption for the flight was 2621 kg, which is validated from the fuel data from airline operations office for the same flight. As can be seen from the figure the fuel flow is maximum in the take-off phase when the engine thrust is maximum, then in the cruise phase it is constant at 0.65kg/s. As aircraft starts to descent the fuel flow drops to 0.07kg/s. Then the fuel flow increase and decrease as thrust is applied to maintain the speed and altitude schedules. Just before the end of flight there is an increase in fuel flow due to application of reverse thrust at touchdown.

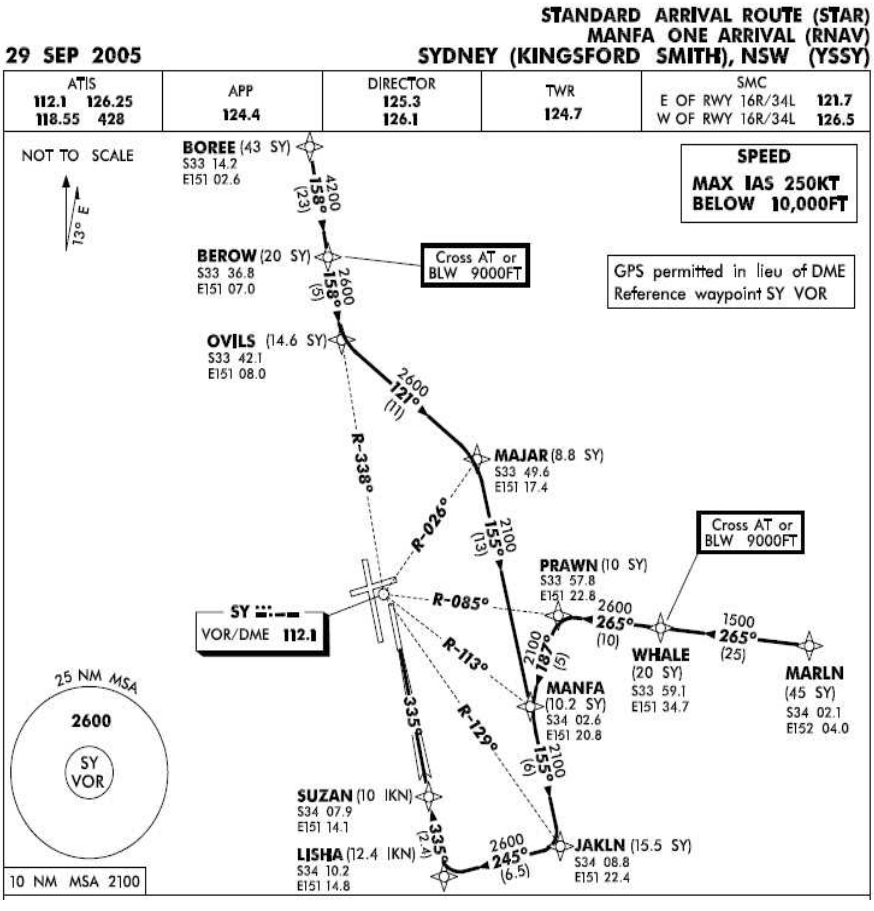


Figure 3.27: STAR chart of Sydney (Kings-Ford Smith) International Airport, MANFA ONE arrival (Air Services Australia, 2004)

For Approach and Landing phase, Figure 3.27 shows the STAR chart of Sydney (Kings-Ford Smith) International Airport, MANFA ONE arrival (Air Services Australia, 2004). This route (BOREE Fix) is used for approach and landing segment of the simulated flight. Figure 3.28 shows the horizontal flight segment of the simulated flight. It can be seen from figure that flight follows the arrival route closely during its approach and landing segments.

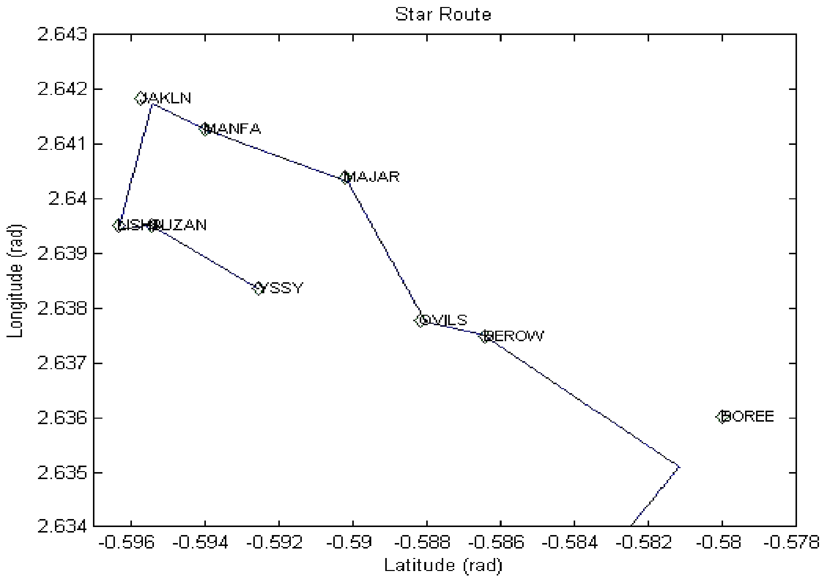


Figure 3.28: Approach route taken by aircraft for the given STAR route (BOREE Fix)

We then plotted distance wise plot (80nm to touchdown) of altitude, speed, thrust, acceleration/de-acceleration, flap configuration and fuel flow. As shown in Figure 3.29, the aircraft follows the vertical profile of the arrival route and speed is reduced accordingly. The thrust is applied to maintain the desired speed and is reflected in the fuel flow. Two flap configurations are used, first 15 degree flap at 18nm to touchdown and 30 degree flap at approx. 9nm to touchdown. The deceleration rate of -1.3kt/sec can be seen with the reduction in speed during descent.

3.4.4 Validation of Aircraft Maneuvers

For validation of aircraft maneuvers the same test flight is used with a *Boeing 737-400* aircraft. We plot rate of acceleration-deceleration (ROAD), rate of climb and descent (ROCD) and rate of heading change (ROHC). A wide range of maneuvers/navigations are performed for validation purpose in the cruise phase. Following are some of the representative manoeuvres used in the validation exercise which covers the three key

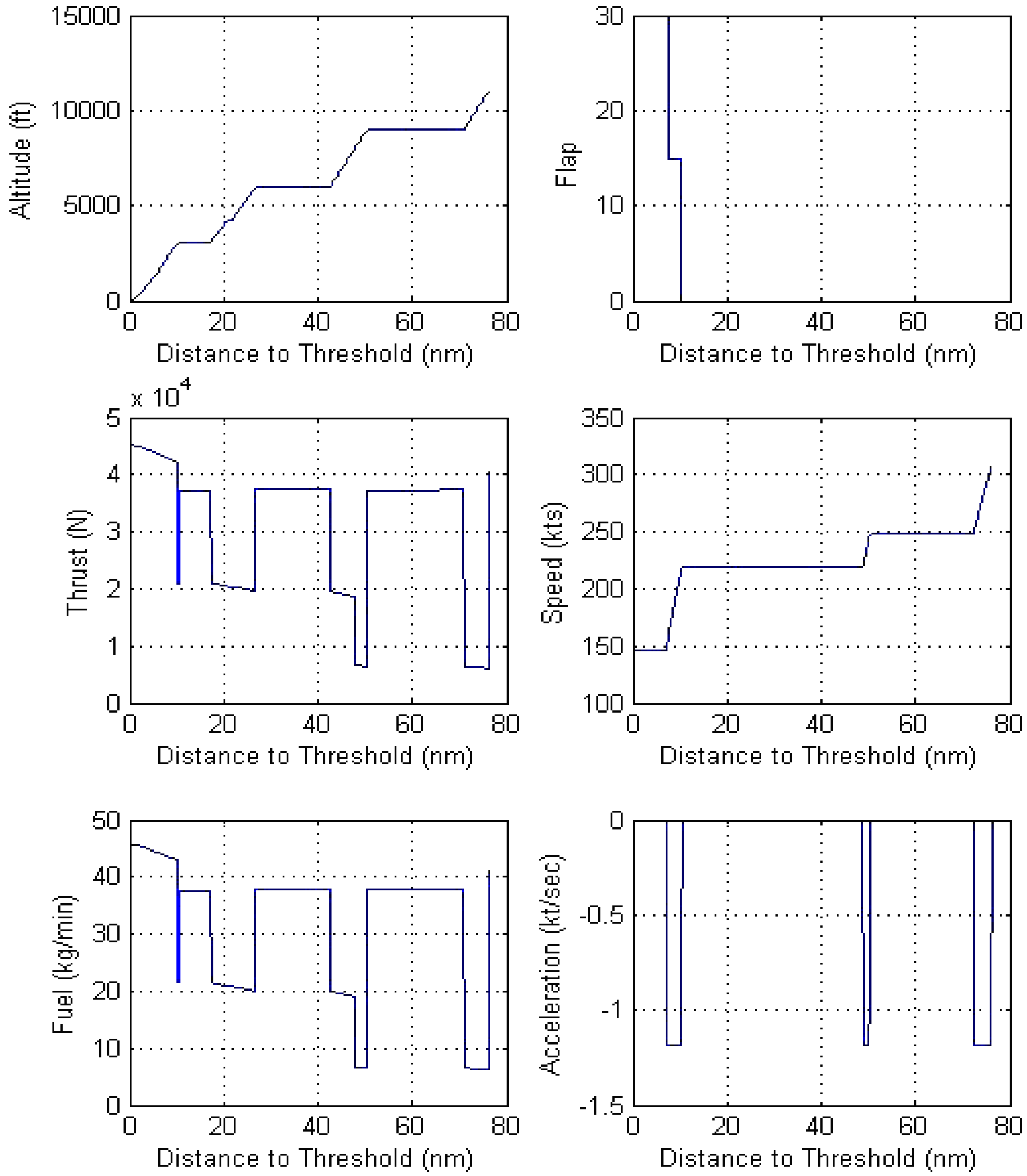


Figure 3.29: Distance to threshold plots for the approach and landing phase (80 nm to touchdown).

component of flight dynamic (Climb/Descent, Accelerate/De-accelerate and Turn heading (left and right).

- A climb maneuver for level flight from 35000ft to 37000ft
- A descent maneuver from 37,000ft to 35,000ft
- An increase speed from 428kts to 450kts
- A decrease speed from 450kts to 428kts
- Change of heading left by 45 degrees
- Change of heading right by 45 degrees

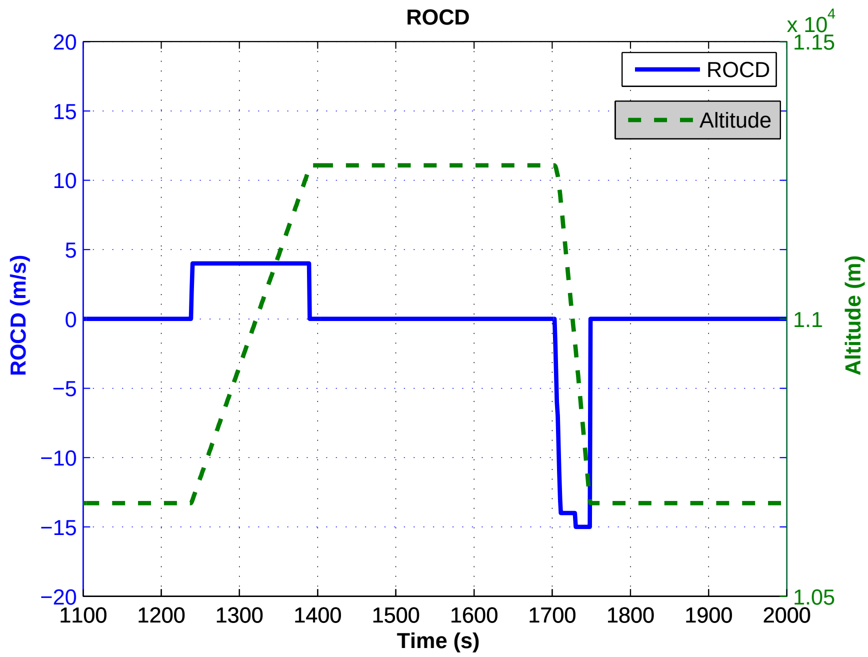


Figure 3.30: Rate of climb descent (ROCD) for a climb from 35000ft (10668m) to 37000ft (11277.6m) and then descent 37,000ft (11277.6m) to 35,000ft (10668m)

Figure 3.30 shows the rate of climb descent (ROCD) for a climb from 35000ft(10668m) to 37000ft(11277.6m) and then descent 37,000ft (11277.6) to 35,000ft(10668m).

Figure 3.31 shows the rate of accelerate and de-accelerate (ROAD) for increase in speed from 428kts(220m/s) to 450kts(231.5m/s) and then decrease in speed from 450kts to 428kts.

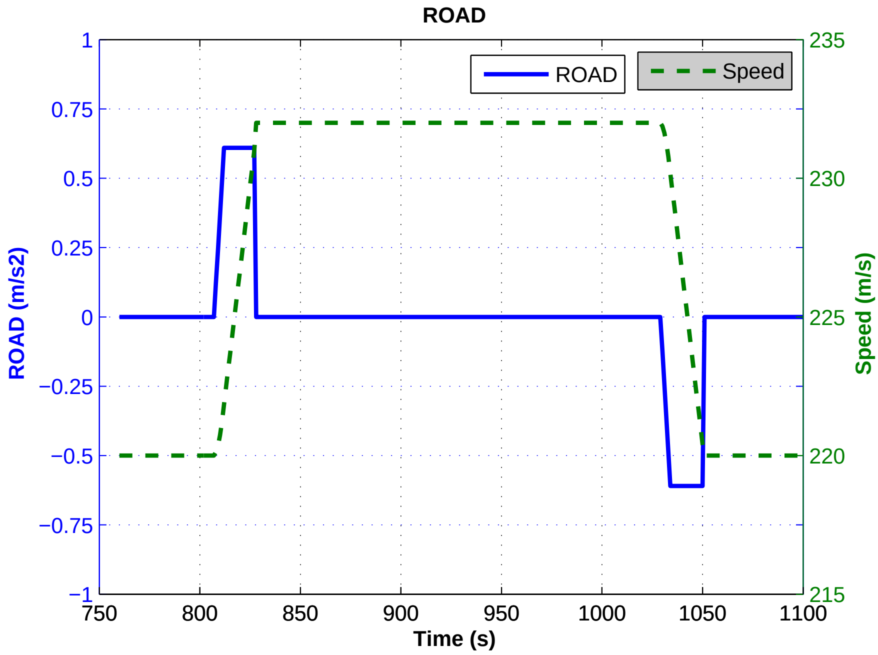


Figure 3.31: Rate of accelerate and de-accelerate (ROAD) for increase in speed from 428kts(220m/s) to 450kts(231.5m/s) and then decrease in speed from 450kts to 428kts

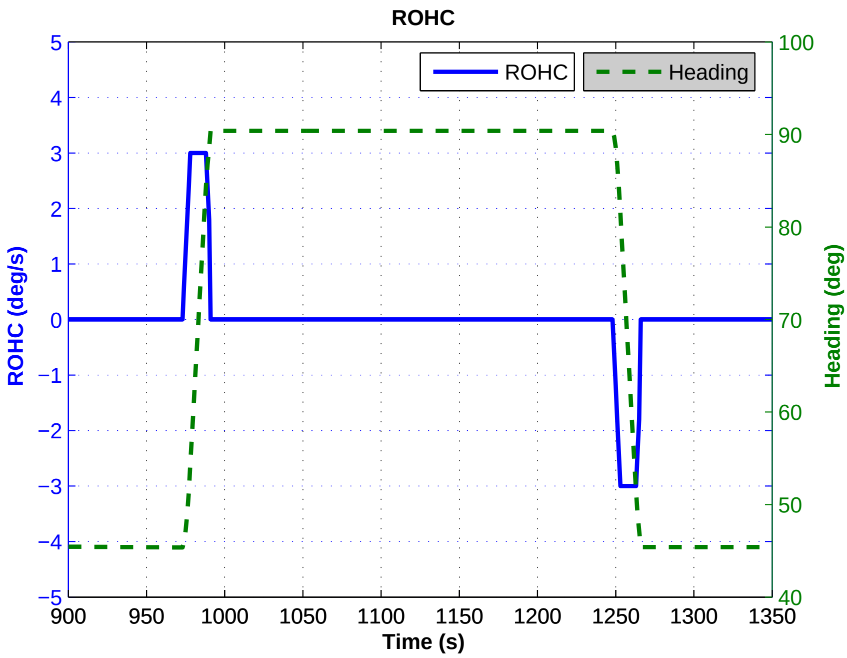


Figure 3.32: Rate of heading change (ROHC) with change in aircraft heading 45 degrees to the left then 45 degrees to the right. Maximum ROHC is ±3 degrees on either side for the turn maneuvers.

Figure 3.32 shows the rate of heading change (ROHC) with change in aircraft heading 45 degrees to the left then 45 degrees to the right. Maximum ROHC is ±3 degrees on either side for the turn maneuvers.

All these maneuvers are within the aircraft performance envelop for the given aircraft as validated from its respective BADA performance table (Eurocontrol, 2004).

3.4.5 Effect of Wind on the Ground Speed

To see the effect of wind on aircraft speed we simulated a long distance flight (Perth-Darwin) with a B747-400 aircraft. The cruise speed of the aircraft was 250m/s. The effect of wind on the ground speed can be seen in Figure 3.33.

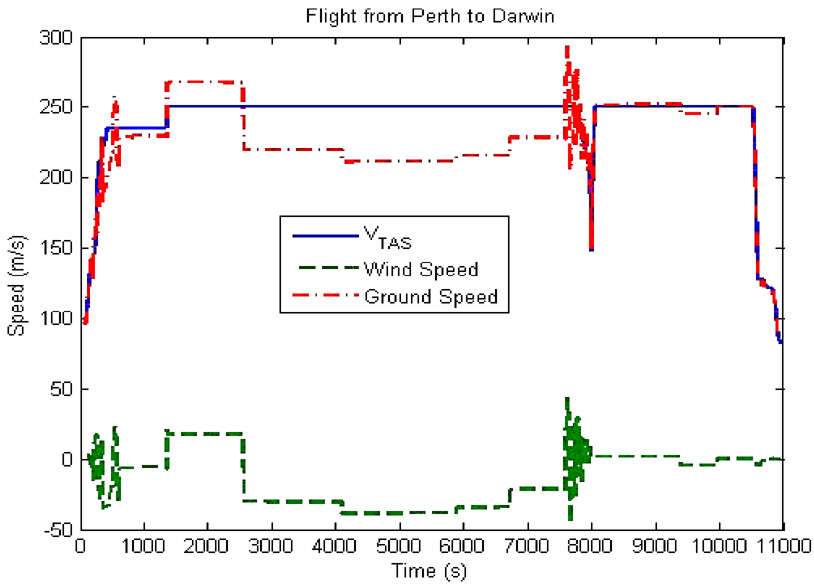


Figure 3.33: Effect of wind on the ground speed of the aircraft (B747-400).

3.4.6 Fuel flow Validation

Figure 3.34 present the fuel consumption (kg/s) and the aircraft's mass (kg) over time for different fuel computation approaches for fixed route and Free Flight scenarios respectively. The plots show that an aircraft's mass monotonously reduces according to the amount of fuel burnt, with the assumption that the only reduction factor for aircraft's mass is the fuel. The fuel consumption plots also illustrate that the fuel consumption is highest during take-off and gradually decreases during the climbing phase and levels off

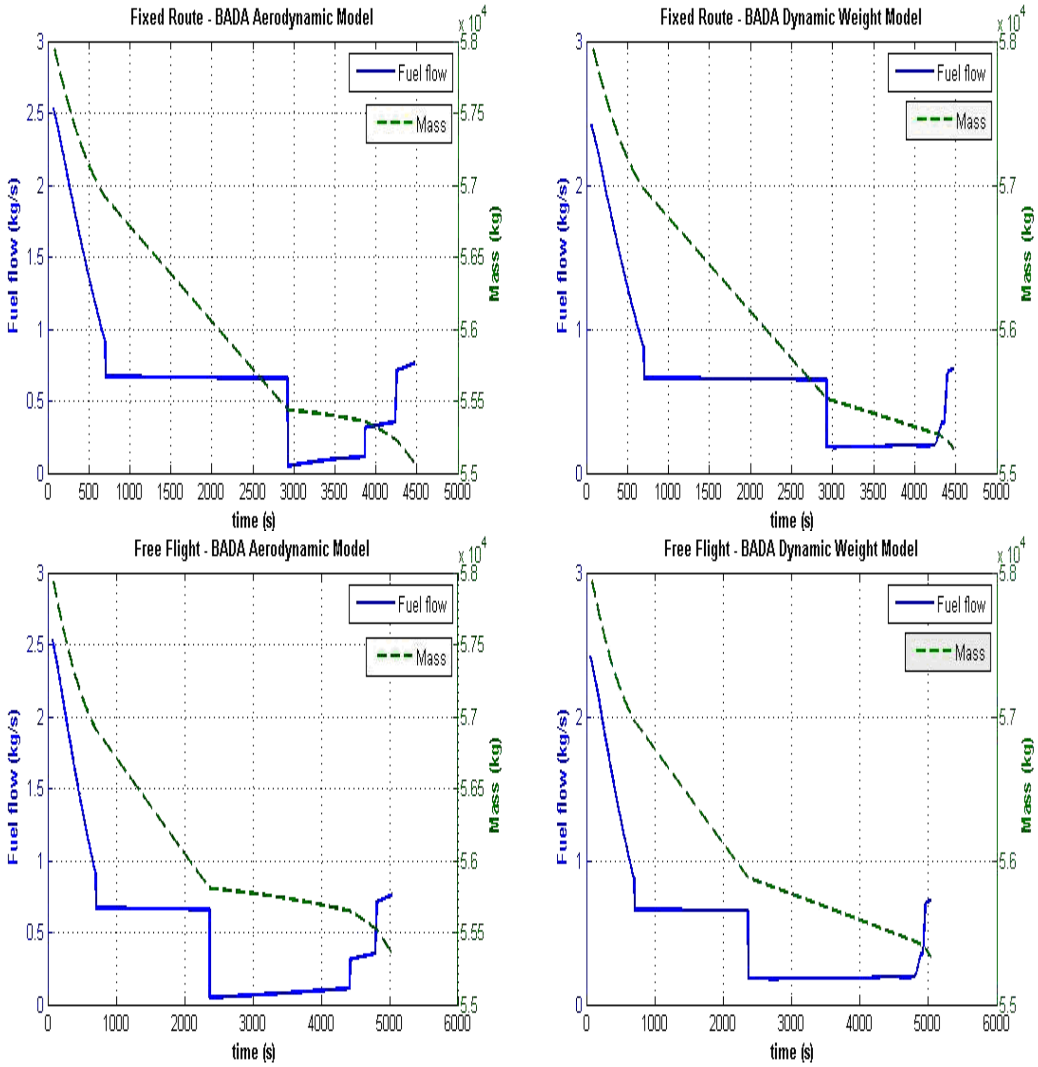


Figure 3.34: Fixed Route (top) and Free Flight (bottom) Fuel consumption and aircraft mass over time for BADA Coefficient and Dynamic Weight approaches

when the aircraft hits the cruise phase. During cruise, the fuel consumption decreases very slowly, i.e. reduction slope is almost zero, until the aircraft hits the TOD where the fuel consumption drops significantly due to fact that aircraft uses minimal power during descent and approach. These behaviors are similar for both fuel consumption models and both Fixed route and Free flight.

We then compare the three different fuel consumption methodologies used in ATOMS: fuel flow derived from BADA tables for a given speed and altitude, fuel flow with weight reduction and the fuel computation using BADA aerodynamic model. Figure 3.35 presents the fuel consumption results from three different models proposed in ATOMS for the given flight. The plots show that all three approaches yield similar results. The maxi-

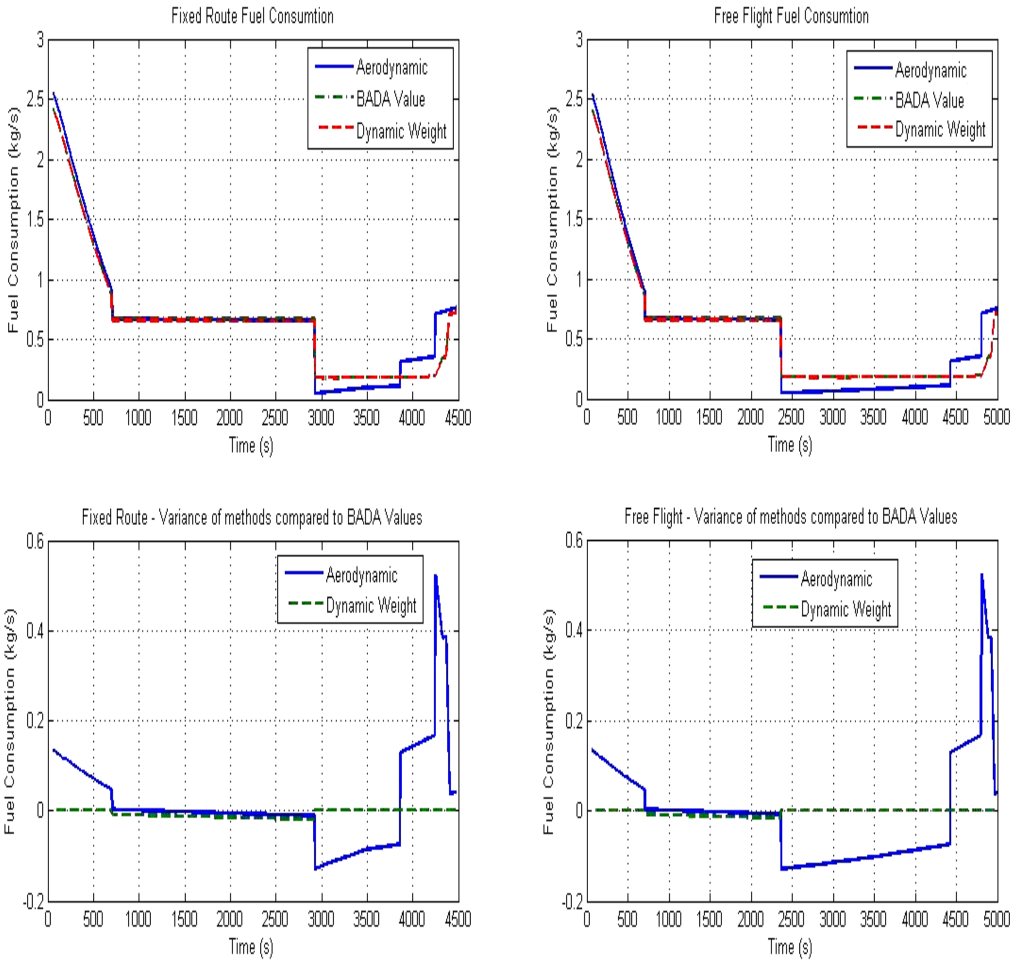


Figure 3.35: Fixed Route and Free Flight Fuel Consumption - Fuel consumption values of the BADA aerodynamic model, weight reduction model and BADA baseline fuel flow values (top), variance of fuel values comparing to the BADA fuel flow values (bottom)

mum differences between the aerodynamic model and the weight correction approaches comparing to the BADA fuel flow values are 0.5 kg/s and 0.01 kg/s respectively. Since the fuel burn calculations, based on BADA fuel flows are very close to real values (Jelinek, 2006), the small variances between the other two approaches comparing to the simple approach validate the fact that the other two approaches are also correct.

The fuel consumption model using weight correction is closer to the BADA values because it is computed based on interpolation of the fuel flow BADA values. Meanwhile the aerodynamic approach computes the fuel consumption based on a more complicated and higher fidelity formula of an aircraft's aerodynamic forces, hence the computed result might not come as close to the BADA fuel flow values. However, the aerodynamic method yields more realistic performance than the other two methods.

3.4.7 Fuel & time savings in Free Flight

To maintain orderly air traffic flow ATCs may allocate pilots an altitude that might not be optimum for the aircraft in terms of fuel consumption. In Free Flight pilots will be able to choose the most fuel efficient altitude depending upon the type of aircraft, thus offering immediate economic benefits to the industry in terms of lower fuel consumption. This mainly results from flying aircraft at their optimal altitude and time savings due to

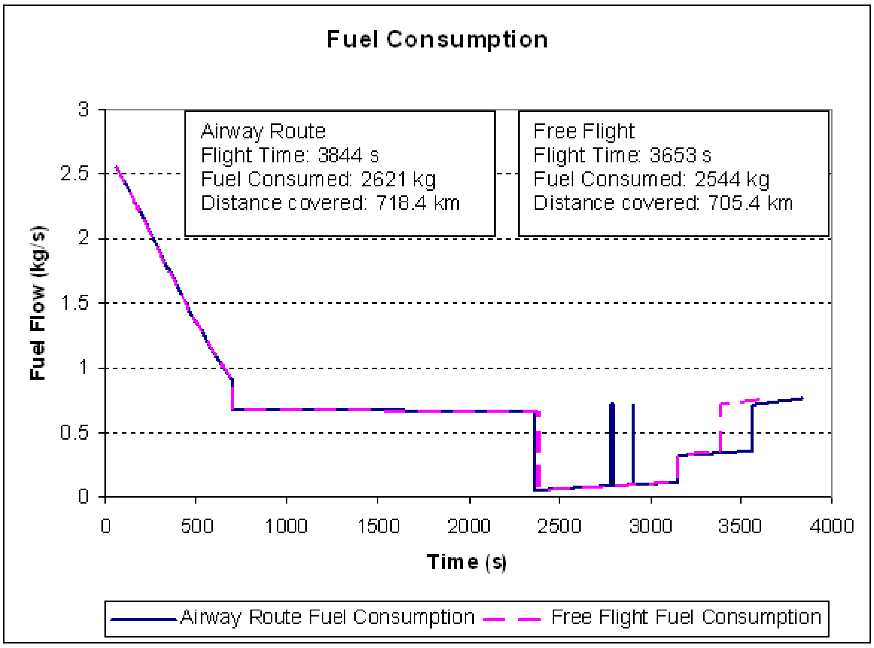


Figure 3.36: Fuel consumption and travel time for the test flight (B737-400 with two CFM56-3C-1 engines) in Free Flight and controlled airspace (No STAR routes).

direct route to destination, given the airspace constraints. This will lead to reduction in flight operating time which may result in more effective use of airline fleet and passenger comfort. As shown in figure 3.36 for the test flight, when simulated in Free Flight airspace, it consumed 77.0kg less fuel, saves 191 seconds of flying time and covers approx. 13km less distance than in controlled airspace. For simplification, in both the experiments STAR route is not considered and aircraft fly directly from the last waypoints in their flight plan to the destination airport. This was primarily due to the direct route flown by the aircraft in Free Flight airspace leading to few maneuvers in the absence of waypoints.

However, in Free Flight airspace, aircraft may not be able to fly direct routes due to various airspace constraints but even then, given the quantum of benefits, it will certainly have a significant economic impact.

3.5 Chapter Summary

In this chapter we have presented the design, architecture and various functionalities of Air Traffic Operations and Management Simulator (ATOMS). The GUI of ATOMS and various run options are also shown. Atmospheric and wind modeling is explained and validated. Field radar data is used to validate aircraft's trajectory and guidance model. Aircraft performance model is evaluated using BADA performance tables. Effect of wind on ground speed is illustrated. Fuel models implemented in ATOMS are compared with baseline BADA model. Finally fuel and time savings from flying great circle route in Free Flight is presented.

Chapter 4

Advanced ATM Concepts in ATOMS

This chapter is partially based on following publications:

1. Alam, S., Ngugen, M. H., Abbass, H. A. and Barlow, M. On Architecture, Design and Validation of an Air Traffic Simulator, Simulation Conference and Exhibition (SIMTecT-07), Brisbane, Australia, June, 2007

2. Ngugen, M. H., Alam, S. and Abbass, H. A. Dynamic Weather Avoidance in a Traffic Constrained En-route Airspace, 6th Eurocontrol Innovative Research Workshop and Exhibition, pp. 205-212, Eurocontrol Experimental Center, Paris, France, Dec, 2007

3. Alam, S., Ngugen, M. H., Abbass, H. A. and Barlow, M. Ants Guide Future Pilots, 3rd Australian Conference on Artificial Life (ACAL07), LNAI-4828, Springer-Verlag Berlin Heidelberg, pp. 36-48, Gold Coast, Australia, Dec, 2007

4. Alam, S., Lam, T. B., Abbass, H. A. and Barlow, M. Pareto Meta-Heuristics for Generating Safe Flight Trajectories Under Weather Hazards, 6th International Conference on Simulated Evolution and Learning (SEAL-06), LNCS-4247, Springer Berlin Heidelberg, pp. 829-836, Hefei, China, Oct, 2006

5. Alam, S., Abbass, H. A., Barlow, M. and Lindsay, P. Mapping Lessons from Ants to Free Flight: An Ants based Weather Avoidance Algorithm for Free Flight, SPIE-06, Complex Systems , Vol. 6039, pp. 205-213, Brisbane, Australia, Jan, 2006

In the previous chapter, we discussed ATOMS which can provide a flexible platform for rapid prototyping of advanced ATM concepts. In this chapter we develop and implement a representation for Free Flight airspace and discuss advanced ATM concepts implemented in ATOMS with their preliminary evaluation.

The research objectives of this chapter are:

- to design and develop an efficient airspace model for effective implementation of advanced ATM concepts.
- to design and implement simulated versions of advanced ATM concepts in ATOMS,
- to conduct pilot experiments to evaluate the prototyped ATM concepts on safety and economic metrics,
- to identify issues in evaluation and implementation of advanced ATM concepts.

The advanced ATM concepts which are prototyped and evaluated in ATOMS are:

- Airborne Separation Assurance: Four CD&R algorithms from the literature, based on a nominal projection methodology, are implemented in ATOMS. Preliminary experiments are conducted, on simplistic air traffic scenarios, to assess safety and economics.
- Airborne Weather Avoidance: A meta-heuristic algorithm is developed and implemented for airborne weather avoidance in a traffic constrained enroute airspace. A multi-objective approach is adopted to generate a set of maneuver choices which are inherently safe by design.
- Cockpit Display of Traffic Information (CDTI): CDTI enables pilots to observe the surrounding air traffic pattern and airspace constraints. It provides an enhanced situation awareness and "what if" capability in case of trajectory modification or conflict resolution. A prototype for CDTI is developed and implemented in ATOMS as an aid for CD&R and weather avoidance.

4.1 Modeling of the Free Flight Airspace

In Free Flight, the loss of waypoint-airway structure may complicate the process of the timely detection of conflicts between aircraft. Since each aircraft flies its desired trajectory based on great circle distance between entry and exit points of the airspace, some form of a reference system to accurately define an aircraft position is necessary. A 3-D reference grid system with discretized bins, representative of the airspace under consideration, is proposed and implemented.

4.1.1 Data Link Assumption

A standard assumption of Free Flight is that the aircraft will make use of a high speed data link with other aircraft, ground based services and ATCs for communication, navigation and surveillance. As discussed in the background chapter, (section: The future ATM Environment, subsection: Communication) the airborne component of the proposed Aeronautical Telecommunication Network will provide airborne ADS broadcast data which will include aircraft position and velocity data, and possibly intent information, to ATCs and AOC. Any future modeling and simulation of advanced ATM concepts requires some form of data link modeling. Therefore, in ATOMS all simulated aircraft are assumed to be data linked.

4.1.2 Airspace Modeling

Every aircraft in a data linked environment repeatedly broadcasts and receives state/intent information from other airspace and ground entities within a certain range (RTCA, 1998). Simulation of data linked enabled aircraft in a fast time simulation is quite challenging from both a computational (search) and representational (data structure) point of view. For example, if there are n aircraft in the airspace, the number of computations needed for predicting conflict between n aircraft at one instant of time is $n(n-1)/2$ and this has to be repeated in every prediction cycle. Thus an efficient representation of the airspace and search procedure is required for fast time evaluation of any ATM concept.

Isaacson and Erzberger (Isaacson and Erzberger, 1997), for their CD algorithm in CTAS, employed pair-wise trajectory pruning by altitude and time skipping when aircraft are separated by large distances to reduce the number of trajectory pairs to be evaluated by the computationally intensive 4D search algorithm. The computational performance of this algorithm still grows as $O(n^2)$, but the number of probe computations is proportionally reduced through the applied heuristics.

Shridhar and Chatterji (Sridhar and Chatterji, 1997) proposed a computationally efficient conflict detection method by transforming the planned trajectories of the aircraft into a sequence of 2D discrete spatial regions (grid cells). The positions of aircraft are stored in the grid cells by using a hashing function to convert the aircraft's coordinates into grid indices. The grid indices are then sorted to locate regions in airspace where conflict is possible. Efficient sorting algorithms (heap and quick sort) are used to locate the bin numbers with $nlog_2(n)$ computational performance.

Jardin (Jardin, 2005) extended this approach to a 4D grid of space and time to represent the airspace and presented algorithms for computationally efficient conflict detection. The potential challenges in implementing the conflict grid techniques is left as an open question. The potential applications, of grid based airspace, in weather avoidance is also suggested to be explored.

4.1.2.1 Introduction of the Bin Concept

We attempt to model the Free Flight airspace as grid and call each cell in the grid a bin. A bin referenced airspace is a 3-D volume of airspace that is subdivided into small hexahedral blocks. Looking at a 2-D version of an airspace cell block in Figure 4.1, right, each bin can be defined by its start (latitude, longitude) and end (latitude, longitude). Using these two points, the corner co-ordinates for each bin in the whole airspace can be calculated and thus the extremes of the airspace can be defined using the geodetic earth co-ordinate system. Extrapolating this to the third dimension, which is altitude, the result

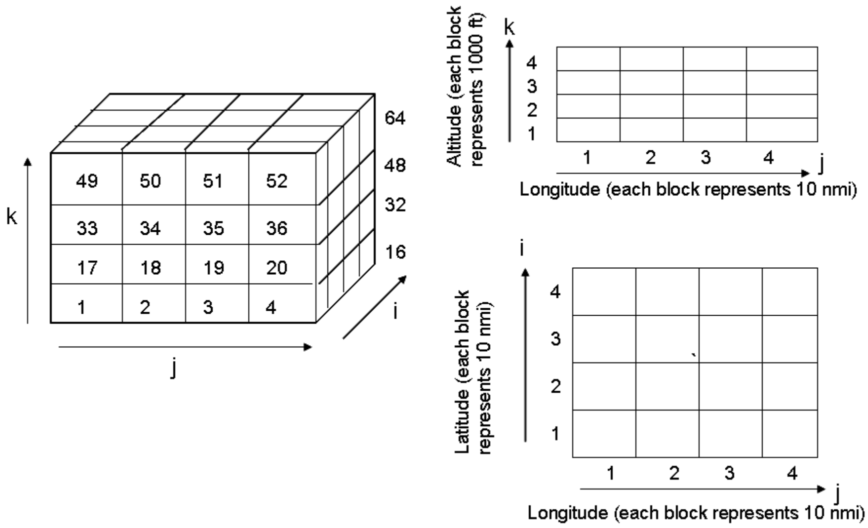


Figure 4.1: Airspace modeled as hyper-rectangular discrete space. Block 1 is (1,1,1) in (i,j,k) coordinates, block 64 is (4,4,4) in (i,j,k) coordinates. On right side the labelling scheme in a 2D representation can be seen.

is bins that have a defined start and end points and a base altitude with which the 3-D shape of the bin can be determined. Each bin is identified by a unique index representing the latitude, longitude, and altitude. As aircraft fly from one bin to the other, their entry and exit from the bin is defined as a latitude and longitude pair. Figure 4.1, left, shows the conceptual 3-D version of the bin.

By promulgating the concept across the airspace from the start latitude to the end latitude, the start longitude to the end longitude, and the start altitude to the end altitude, the whole airspace can be represented as a 3-D bin formation. The bins function as a reference grid, storage to flight details and a support for conflict detection process. Figure 4.2 shows an illustration of a bin modeled airspace.

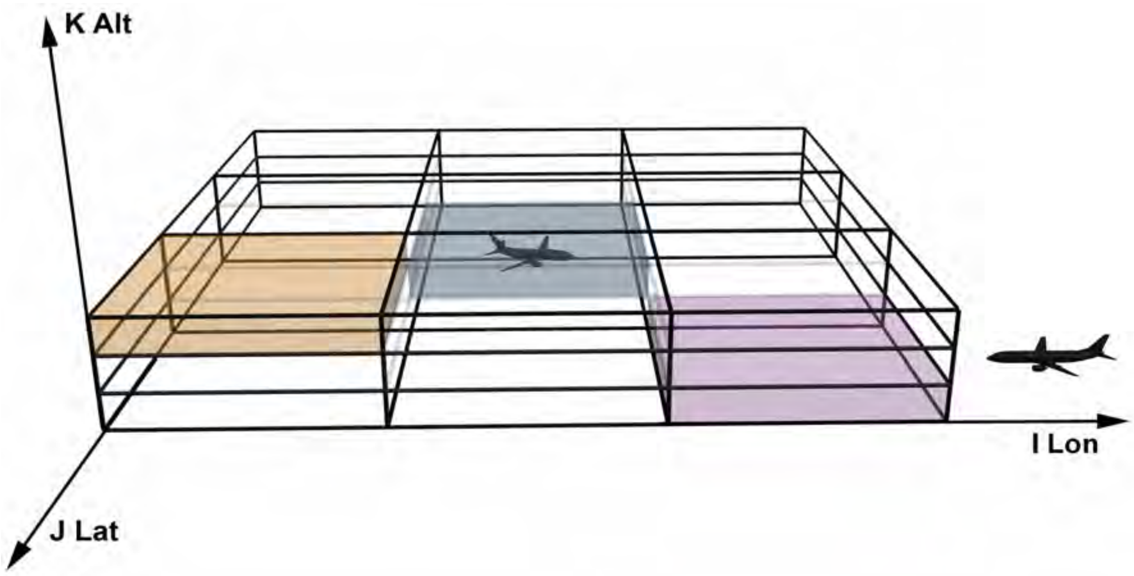


Figure 4.2: An illustration of flights in bin modeled Free Flight airspace

4.1.2.2 Determining the Bin Size

The size of the bin is based on minimum separation standards between two or more aircraft in Free Flight. As per the RTCA guidelines for Free Flight (RTCA, 1995), aircraft are surrounded by an imaginary cylindrical volume called the protected zone (PZ). The radius of the PZ is half the required separation standards (2.5 nm) and the height is equal to the vertical separation (1000 ft). The two aircraft are said to be in conflict if their PZ overlaps. An aircraft flies its intended trajectory by passing from one bin to the other. We have adopted a slightly modified definition as shown in Figure 4.4 where only the intruder is surrounded by the PZ whose dimensions are twice as defined in RTCA definition. The radius of the cylinder (PZ) is 5nm and height is 1000 ft. The conflict detection problem is to find in 4D (position and time) whether, given state information, a projected trajectory

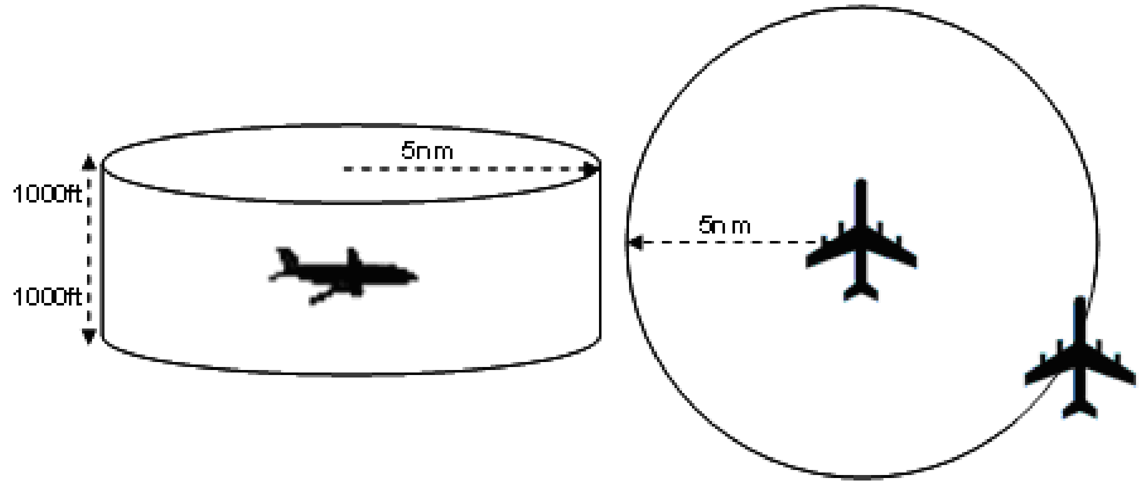


Figure 4.3: Left: The PZ around an aircraft, Right: a 2D view of a conflict situation

(of an ownship) intersects with a cylinder (PZ of an intruder) at a future time. This representation simplifies the implementation of the conflict concept. The bin dimensions are thus chosen to be 10 nm horizontally and 1000 ft vertically. In consideration of the volume of airspace around aircraft to be protected, this is the best fit model to use. Thus, the bin concept can be assumed to give a near optimal approximation of the protected zone in the horizontal and vertical case.

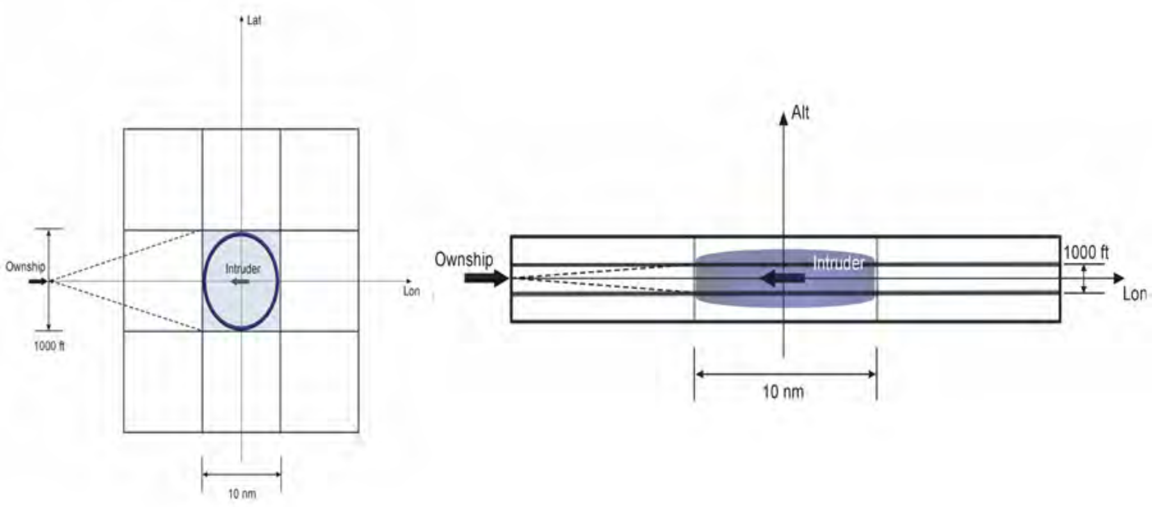


Figure 4.4: Top and vertical view of the bin based protected zone respectively

The bin protection zone is a fixed grid as opposed to the mobile cylindrical zone. A fixed grid can have a drawback as illustrated in Figure 4.5. The aircraft could be anywhere in the fixed bin as shown by the red dot. Therefore for any optimal resolution of a conflict, it may be necessary to keep track of where the aircraft is with respect to a bin. The overall protection area necessary, using the bin concept, is shown in Figure 4.5. The large square represents 30nm x 30nm x 3000 ft and form the bin cluster (27 bins in the cluster with ownship in the central bin) around the central bin. Since it is not known where the aircraft (red dot) is in the bin, all the neighboring 26 (shaded) bins would be considered as the maximum possible conflict search area.

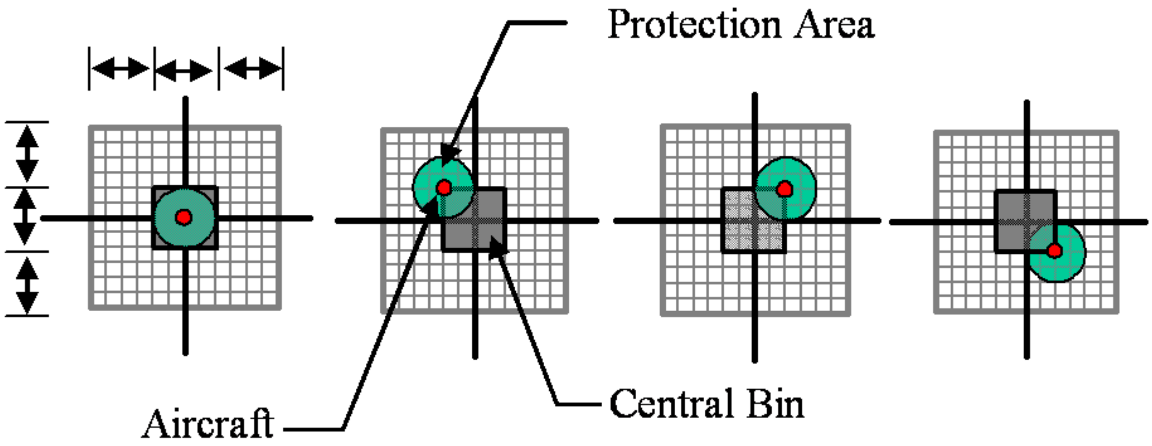


Figure 4.5: Conflict search area. The smaller grid represents a 2.5nm x 2.5nm grid. The large square represents 30nm x 30nm x 3000 ft (bin Cluster). The small circle is the aircraft and the large circle is its protected zone (5nm radius)

4.1.2.3 Referencing the Bins

Each reference must be unique to the bin and must support the objective of storing flight intent information of aircraft that pass through it and allowing efficient access to this information.

First, start latitude, start longitude, end latitude, and end longitude are identified for the airspace to be modeled as a grid. The vertical extent of the airspace to be modeled is also identified. Using start and end coordinates the airspace distance dimensions are computed. For example given some start and end degrees the airspace distance dimensions are 500 nm $\times$ 1000 nm. Let the altitude extent be 42,000ft. Then airspace dimensions are 500 nm $\times$ 1000 nm $\times$ 42000 ft. Using the bin dimensions this will transform to 50 $\times$ 100 $\times$ 42 number of bins. This airspace is represented as 3D array data structure where each element of the 3D array represents a bin. Each bin can now be uniquely identified by a 3D array index.

To define each bin we need to compute its start and end coordinates as well as start altitude. We know that at the equator the degree-distance relationship is given by 1 deg latitude = 60 nm. As we go towards the poles this relationship between degree and distance changes and is given by 1 deg latitude = 60 × cos (latitude)nm. To build bins with 10 nm X 10 nm dimension, the delta latitude change for each 10 nm is given by $1/6.0 \times cos(latitude)nm$

The degree-distance relationship along the longitude remains constant. The algorithm to compute the bin dimensions, their unique index, and assignment into airspace 3D array is presented in Algorithm 2. The unique index for each bin is: (LatIndex, LonIndex, AltIndex). Similarly given a bin index, its start coordinates in the airspace can be also computed.

```
Require: :(Start Latitude, StartLongitude, EndLatitude , EndLongitude, EndAltitude)
  3D Airspace Grid G with AltIndex × LatIndex × LonIndex dimensions
  EndAltitude = EndAltitude / 1000.0
for AltIndex= 0 ; AltIndex ≤ EndAltitude; AltIndex = AltIndex + 1 do
    for LatIndex= 0 , DLat = StartLatitude ; DLat = EndLatitude; LatIndex =
    LatIndex + 1 do
        for LonIndex = 0 , DLon = StartLongitude ; DLon = EndLongitude;
        LonIndex = ILonIndex + 1 do
            BINStartLat = DLat
            BINStartLon = DLon
            BINEndLat = DLat + 1/6.0 × cos(DLat)
            BINEndLon = DLon + 1/6.0
            BINBaseAlt = AltIndex * 1000.0
            G[AltIndex][LatIndex][LonIndex] = bin
            DLon = DLon + 1/6.0
        end for
        DLat = DLat + 1/6.0 × cos(DLat)
    end for
end for
return Grid G
```

Algorithm 2: The pseudo code for index generation and bin computation

As discussed in the previous chapter a flight's trajectory in the Free Flight airspace is either defined by its origin-destination pair, where the aircraft follows the great circle route or by its activation and deactivation points in the airspace (any other trajectory change point (TCP) between them).

By forward integrating the aircraft position and speed (derived from BADA) a flight trajectory can be built which consists of a set of bins which the aircraft will traverse during its flight. The sequence of bins that make up the trajectory of the flight is called its bin plan. Each bin in the airspace grid, have now a list of pointers to the flights which will pass through it.

Each bin thus stores the following data: 1.) The start and end latitudes and longitudes of the bin, 2.) The base altitude of the bin, 3.) A list of flights, sorted on their ETA, that shall be entering the bin some time in future, 4.) The forecast wind and atmospheric information of that bin, 5.) The status of the bin i.e. 'Active' if there are flights traversing through it within a certain time window otherwise 'Inactive'.

4.1.2.4 Conflict Probing in Bin Modeled Air Space

When a flight moves in the airspace, its bin plan is updated based on its current position (lat, lon and alt). When it exits a bin, and proceeds to the next, that previous bin is deleted from its bin plan. Therefore the first bin in the aircraft's bin plan always indicates its current reference position in the airspace grid. Given the look ahead time (T) parameter of the CD algorithm used by the aircraft, its future bin position b_{ijk} is identified and based on it the bin cluster B_{ijk} surrounding it is identified (Figure 4.6). For each flight in the active bins of the bin cluster, its ETA is checked with the intruder's ETA. If the difference in their ETA (ΔETA) is within a certain time window then the CD algorithm is invoked to do the conflict probe. The same process is repeated for all active flights in the airspace until all are checked. The search process can be represented as:

$$S^T = \sum_{n=1}^{N} \sum_{q=1}^{27} \sum_{r=1}^{N_{B_q}} \Delta ETA(n, r) \tag{4.1}$$

where $n \neq r$, q is the bin counter for the bins in the cluster B_{ijk} around the bin b_{ijk} which contains the aircraft a_n at time t. r represents the number of aircraft in a bin B_q, which is one of the bins in the bin cluster B_{ijk}, at time t and where N_{B_q} represents the maximum number of aircraft in the bin B_q. Thus, the inner most loop checks each aircraft in the bin B_q against all the other aircraft in B_q's 27 bin cluster B_{ijk}. This process is repeated for each aircraft "n". The search is O(27 x N). To fine tune the search even more, the status of the bin, 'Active' or 'Inactive', is used to indicate whether to check it for flights or not. The search requires checking all the adjacent 26 bins in the worst case.

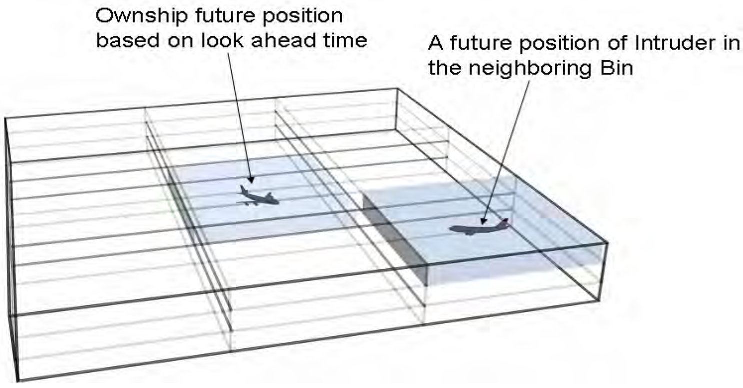


Figure 4.6: Position of ownship in the center of a bin cluster at a certain look ahead time. One of the bins in the bin cluster stores the future position of the intruder.

This search is flight centric it requires iteration through each flight in the active flight list data structure at every discrete time step. To make this search bin centric (i.e. to directly know the state of future conflicts in a bin) we used an M-way tree data structure (Sedgewick, 1998). The M-way tree's node structure is designed to possess quick access to the neighboring bins in the search process. The links to the active bins in the 3D airspace are stored in an M-way (3 nodes) tree data structure which is self balanced. The degree (M) of the tree is 3, based on the index (LatIndex, LonIndex, AltIndex). The data structure design is shown in Figure 4.7. Each node in the 3 way search tree has three keys stored in the order AltIndex, LatIndex, and LonIndex. On the left hand side of Figure 4.7 (search by Tree) only the search node keys are shown.

The values in a node are stored in ascending order, and the subtrees are placed between adjacent values, with one additional subtree at each end. We can thus associate with each value a 'left' and 'right' subtree. All keys in a sub-tree to the left of a key are smaller than it. All keys in the node between two keys are between those two keys. All keys in a sub-tree to the right of a key are greater than it.The algorithm for searching for a value in an M-way search tree is the generalization of the algorithm for searching in a binary search tree.

As shown in Figure 4.7, to access bin (35,42,6), the value 35 is recursively searched in the tree's left subtree if AltIndex is less than parent node's AltIndex, else in the right subtree until the node is found. Then from that node a recursive search procedure of LatIndex and then LonIndex is performed. Once the desired node is found, we have

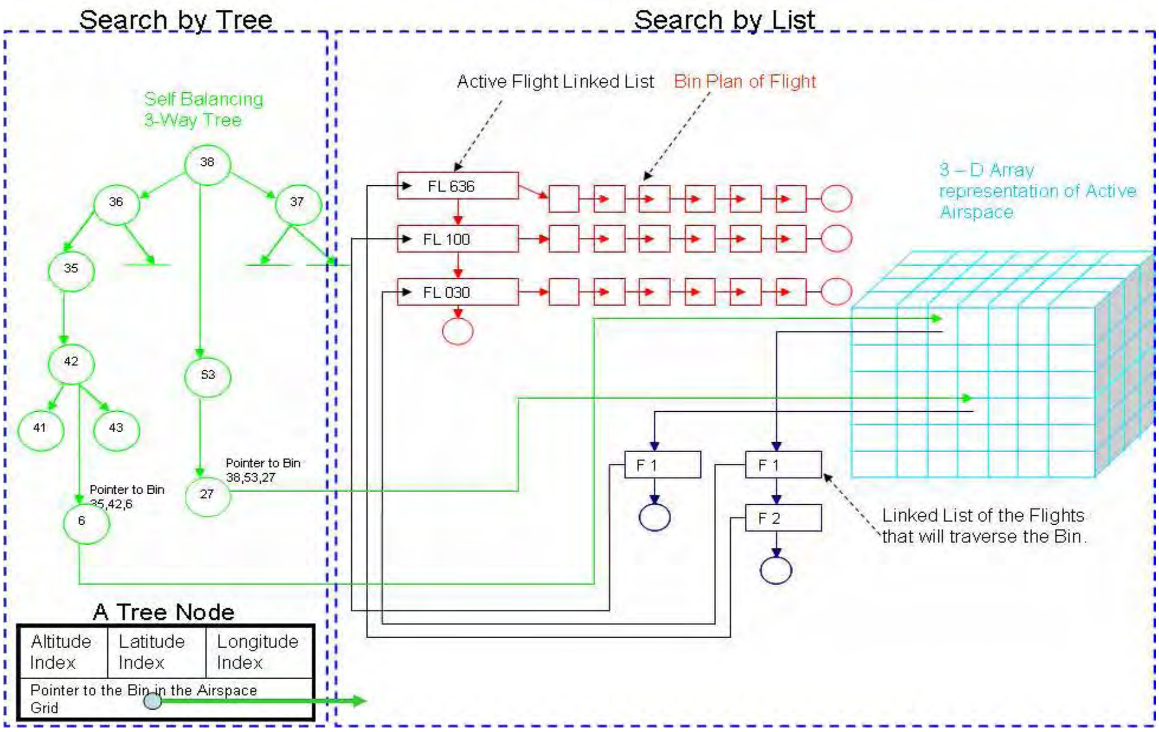


Figure 4.7: Data Structure implementation of grid airspace using bins. In the bottom left, a representation of the tree node is shown with the three keys and a link to the bin

direct access to its bin pointer in the 3D array. The Tree links are updated based on the flight bin position in the 3D airspace. The tree is height balanced with every insertion (when new flights are activated in the airspace and their bin plan is loaded in the 3D airspace) and deletion (when the flight moves to the next bin, the link (node) to that bin in the Tree is deleted). This bin centric search process requires one complete parse through the 3-Way Tree. The search time for an m-way tree is O(n), where the number of nodes visited in O(n/M). For each node visited, search must look at M values, searching in O(log_2(M)) time, yielding a best case of O(n/M $\times$ log_2(M)).

4.2 Conflict Detection and Resolution

The objective of a CD&R system is to maintain safe separation with neighboring traffic. It is an essential component of ATM. The objective of a CD& R system is to identify a potential loss of separation with one or more neighboring aircraft and to provide resolution to avoid that potential conflict. The CD&R algorithms use sensor data as input to predict conflicts and provides resolution advisory either in the form of heading, speed and altitude vectors or a combination of them. Traffic Alert and Collision Avoidance

System (TCAS) (Ford, 1986) is a preliminary form of a CD&R system that is available on most of the modern jetliners. However, it is a collision avoidance system not a collision detection system and is seen as a last line of defence. It provides a collision alert warning with a 20-30 seconds window. It only provides vertical resolution advisory which is neither optimized nor takes into consideration the aircraft's performance envelop. With the advances in automation and availability of technologies such as data link, more sophisticated models have been developed and tested in the field with the objective of improving traffic flow and to implement advanced air traffic procedures to increase airspace capacity and safety. In this section, we summaries the nominal projection conflict detection methodology which forms the basis of CD&R algorithms implemented in ATOMS. Three simplistic scenarios consisting of head-on, cross-over and in-trail conflicts are simulated and the algorithm is evaluated on safety and economic measures.

4.2.1 CD&R process

There is a variety of methodologies for detecting conflicts. We however, focussed only on a nominal conflict detection methodology. The reason being, there is sufficient details available in the literature to prototype the nominal approach based CD&R algorithms and overall the methodology is straight forward to implement.

4.2.1.1 Nominal Projection Methodology

The nominal projection method provides a best estimate of where the aircraft will be, based on the current state information. In situations where aircraft's trajectories are very predictable (projecting a few seconds into the future), a nominal trajectory model may be quite accurate. Nominal projections, however, do not directly account for the possibility that an aircraft may not behave as expected - a factor that is especially important in longer-term conflict detection as shown in Figure 4.8.

Generally, this uncertainty is managed by introducing a safety buffer, minimum miss distance, or time to CPA threshold at which point a conflict will be detected. We first give a simple illustration of 2D conflict detection using this methodology, followed by 3D case as approached in (Dowek *et al.*, 2001).

In 2D, given two aircraft (own ship and intruder) whose position and speed vectors are $\vec{a_o}, \vec{u_o}$ and $\vec{a_i}, \vec{u_i}$ respectively at time t. Assuming their ground speed is constant,

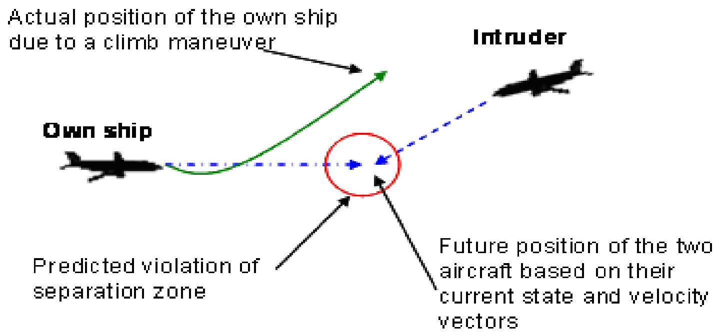


Figure 4.8: A false alarm generated due to climb maneuver by own ship

separation is lost between two at time t, if the projected distance between the two at time t is strictly less than D (horizontal separation).

$$|(\vec{a_o} - \vec{a_i} + t(\vec{u_o} - \vec{u_i})|^2 < D \tag{4.2}$$

Solving this equation for t, we can have two solutions $t1$ and $t2$, such that $t1 \neq t2$ and $t1 < t2$, then $t1$ represents the time of start of separation and $t2$ represents the time of end of separation. If $t1 = t2$ then there is no conflict.

In 3D, the relative position and speed vectors of the own ship are computed with respect to intruder referential $\vec{a} = \vec{a_o} - \vec{a_i}$ and $\vec{u} = \vec{u_o} - \vec{u_i}$, then the relative own ship trajectory is given by

$$L = \{\vec{a} + t\vec{u} : \quad t > 0\} \tag{4.3}$$

A conflict is detected between an ownship and intruder, if at a given time t, the relative trajectory L intersects with the protected zone C, where $\vec{a}$ is the relative position and $\vec{u}$ is the relative speed of the ownship.

$$\vec{a} \in C \quad where \quad C = \{(x_r, y_r, z_r) : x_r^2 + y_r^2 < D^2 and - H < z_r < H)\} \tag{4.4}$$

Where x_r and y_r are the relative horizontal positions of intruder with respect to ownship and z_r is the vertical distance between the two. D is horizontal separation, and H is vertical separation. This cylindrical zone C consists of two parts, the lateral surface around the cylinder $P1$ defined as:

$$C1 = \{(x_r, y_r, z_r) : x_r^2 + y_r^2 = D^2 \quad and \quad - H <= z_r <= H)\} \tag{4.5}$$

and the top and bottom bases

$$C2 = \{(x_r, y_r, z_r) : x_r^2 + y_r^2 < D^2 \quad and \quad |z_r| = H)\} \tag{4.6}$$

A variety of conflict situations are presented:

1. -H $< a_z$ and $-a_x >$ D, as shown in Figure 4.9, where a_x is the the horizontal distance and a_z is the vertical distance between two aircraft. In this case, the boundary of the cylinder surface that may be intersected by the intruder trajectory is given by:

$$B1 = (x_r, y_r, z_r) : |x_r| = \frac{D^2}{a_x}, \quad or \quad |z_r| = H, x_r < \frac{D^2}{a_x} \tag{4.7}$$

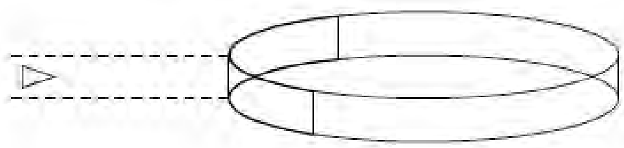


Figure 4.9: Conflict Geometry in Case 1

2. $a_z \leq$ -H and $-a_x \geq$ D as shown in Figure 4.10. In this case the boundary of the cylinder surface that may be intersected by the intruder trajectory is given by:

$$B2 = (x_r, y_r, z_r) : |x_r| = \frac{D^2}{a_x}, \quad or \quad |z_r| = H, x_r < \frac{D^2}{a_x} \quad or \quad z_r = -H, x_r > \frac{D^2}{a_x} \tag{4.8}$$

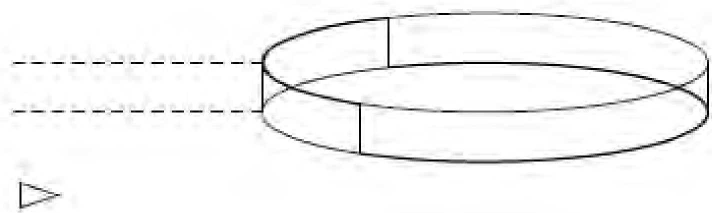


Figure 4.10: Conflict Geometry in Case 2

3. H $< a_z$ and $-a_x \geq$ D as shown in Figure 4.11. In this case the boundary of the cylinder surface that may be intersected by the intruder trajectory is given by:

$$B3 = (x_r, y_r, z_r) : |z_r| = -H \tag{4.9}$$

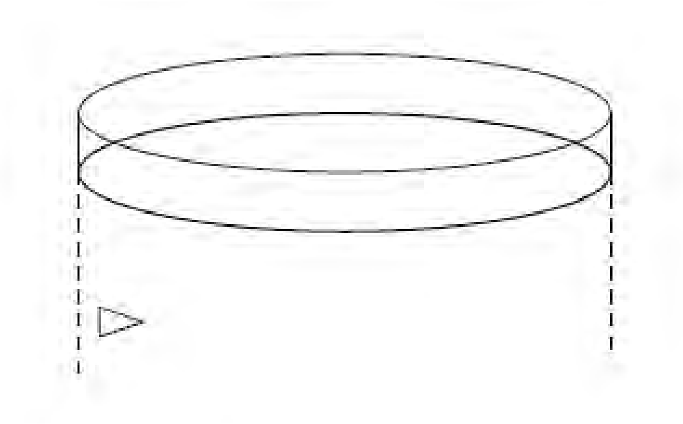


Figure 4.11: Conflict Geometry in Case 3

In (Dowek *et al.*, 2001), for each of the above cases, a geometric conflict test is developed and a conflict resolution set is identified.

In order to resolve the conflict, the relative speed vector $\vec{u}$ to $\vec{u^{`}}$ is modified such that the half line

$$L^{`} = \vec{a} + t\vec{u^{`}} : \quad t > 0 \tag{4.10}$$

does not intersect with the protected zone. The resolution vector needs to be optimal such that $L^{`}$ touches the boundary of P. Thus the target point for the half line L must be either on the lateral surface $P_1$ or on the base $P_2$, and must satisfy two conditions: (1) it is on the boundary of the protected zone and (2) the half line $L^{`}$ passing through it must not intersect with the protected zone. Condition (1) can be represented as $x^2 + y^2 = D^2$ and $-H \leq z \leq H$ and condition (2) as, for all time t > 0, $(a_x, tv_x, tv_y, a_z + tv_z)$ ϵ P. This can however generate an infinite set of target points. Therefore constrained solutions, where only one parameter of the ownship speed vector (ground speed, ground track or vertical speed) is modified, are obtained for resolution.

4.2.2 Candidate CD&R Algorithms

We selected four CD&R algorithms using the nominal projection methodology for conflict detection and three different approaches for resolution, and prototyped them in ATOMS. The first approach is Force Field (Eby, 1994) in which aircraft are treated as electrically charged particles, and uses modified electrostatic equations to generate resolution maneuvers. For resolution maneuvers this approach uses the repulsive forces between these charged particles (aircraft).

Second, a geometric approach (Bilimoria, 2000; Dowek *et al.*, 2001) which utilizes current position and velocity vectors to determine miss distance and time-to-closest-approach as the conflict detection mechanism, and for resolution uses a combination of the vertical speed and heading changes to resolve conflicts while minimizing the magnitude of the velocity change.

The third is an advanced version of the present day traffic collision avoidance system (TCAS) known as FAA-TCASIII (Gazit, 1996), and uses range rate measurers and an altitude threshold for detecting conflicts (tau criterion). For resolution it selects maneuvers that maximize the range vector for the time to CPA or CPA between two aircraft.

Given a conflict probe, the CD&R algorithms processes the state and flight information of the ownship and intruder and predict conflicts (if any) in the future.

The following four CD&R algorithms are modeled in ATOMS:

1. CD&R algorithm (KB3D) by G. Dowek, C. Munoz, and A. Geser, ICASE, NASA Langley Research Center (Dowek *et al.*, 2001)(Geometric).

2. CD&R algorithm (Billimoria) by K.D. Billimoria, NASA Ames Research Center (Bilimoria, 2000)(Geometric).

3. CD&R algorithm (Rockwell) by T.W. Rand, Rockwell Collins and M.S. Eby, Source Code Systems (Rand and Eby, 2004)(Force Field).

4. CD&R algorithm (FAA TACS-III) by R.Y. Gazit, Department of Aeronautics and Astronautics, Stanford University (Gazit, 1996) (Tau criterion).

Table 4.1: Flight plans of conflicting flights for three scenarios

Scenario	**Call Sign**	**Origin**	**Dest.**	**AC Type**	**Speed (kts)**	**Alt (00ft)**	**ETD (s)**
HeadOn	QFA405	YMML	YSSY	A330	458	390	60
	VBB222	YSSY	YMML	A330	458	390	60
InTrail	QFA405	YMML	YSSY	A330	297	390	60
	ZRW444	YMML	YSSY	A330	458	390	1080
CrossOver	SIA222	YSSY	WSSS	B747	480	320	1680
	XCC999	YPAD	YBBN	B747	480	320	60

4.2.3 Conflict Scenarios

The modeled algorithms are evaluated using Free Flight scenarios (Great circle routes). Three different air traffic scenarios in Australian airspace are simulated for the following conflict geometry:

- Head-On conflict: Two aircraft flying at the same altitude, having bearings in opposite directions.
- Cross-Over conflict: Two aircraft flying at the same altitude having a crossing at a point in airspace at the same time.
- In-Trail conflict: Two aircraft flying at the same altitude, with the same heading, where the trailing aircraft has a higher speed and comes in conflict with the leading one in the future.

Sample flight plans for conflict simulation have their estimated time of departure and cruise altitude adjusted accordingly to have the desired conflict scenario as shown in Table 4.1. For the in-trail conflict scenario, two flights departed from the same airport with a certain time difference, the flight which is ahead is assigned a slower speed than the flight which is behind it, causing in-trail conflict when the flight behind catches up.

4.2.3.1 Metrics

We evaluated the algorithms on the following two performance metrics suggested by (Krozel *et al.*, 2001)

- Safety: This metric records the loss of separation, if any, between the conflicting aircraft.

- Efficiency: This metric records the operational cost of conflict resolution by each algorithm in terms of extra fuel burn and flight path distance travelled during a maneuver and the extra time required to execute a maneuver and return back to track.

For resolution, we have assumed instantaneous heading change in vectoring aircraft to resolve the conflict.

For parallel head-on conflict when resolution is done through a heading change maneuver, the horizontal displacement will extend the flight path proportional to the radius of the protected zone R_p (Figure 4.12) as given by (Suchkov, 2001):

$$\delta = 2 \times R_p \times (\frac{1}{sin\psi} - \frac{1}{tan\psi}) \tag{4.11}$$

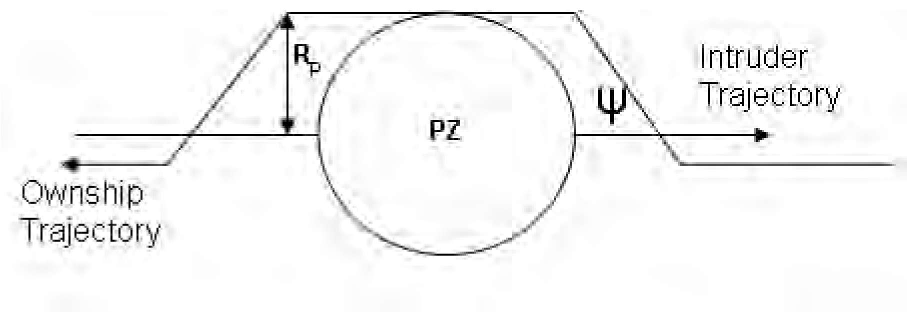


Figure 4.12: Conflict Resolution through a heading change maneuver.

For the given flight with the nominal fuel flow f_{nom} at a speed V, this displacement will incur an additional fuel burn of:

$$\Delta_f = 2 \times f_{nom} \times (\frac{1}{sin\psi} - \frac{1}{tan\psi}) \times \frac{R_p}{V} \tag{4.12}$$

If the two flights are in a crossover conflict, then the additional distance for vectoring aircraft is given by:

$$\delta = 2 \times R_p \times (\frac{1 - cos\psi}{1 - cos\psi + sin\psi}) \tag{4.13}$$

where nominal fuel flow f_{nom} and cruise thrust is calculated from the BADA model as described in chapter 3.

To compute the cost of altitude maneuvers we consider the additional fuel burnt due to change in thrust and time in vertical maneuver as follows:

During climb the following are the force equations on an aircraft (Eshelby, 2000)

$$T = D + W \times sin(\gamma) \tag{4.14}$$

and

$$L = W \times cos\gamma \tag{4.15}$$

then

$$sin(\gamma) = \frac{T - D}{W} \tag{4.16}$$

where γ is the flight path angle or angle of climb. We may assume that γ is sufficiently small so that: $cos(\gamma) \approx 1$

then,

$$L = W \tag{4.17}$$

and rate of climb RC is given by

$$RC = V \times sin\gamma = \frac{V \times (T - D)}{W} \tag{4.18}$$

This assumes that the climb is performed at a constant true airspeed, where as practically the climb is performed at a constant Mach number. Climb thrust for resolution can then be computed as:

$$T = D + W \times sin(\gamma) + \frac{W}{g} \times \frac{dV}{dt} \tag{4.19}$$

4.2.3.2 Experiments & Results

For all the CD&R algorithms, the lateral separation distance is set to 9200m (5nmi) and the vertical separation is set to 305m (1000ft) based on current set of separation standards (ICAO, 1996). The separation time is set to 300s (5min look ahead time) as it provides enough time for pilots to execute a resolution maneuver (Hoekstra *et al.*, 2002). A conflict search area of 80nmi is defined for an aircraft based on current ADS-B technology limitations (RTCA, 2003).

Each flight plan for a given geometry is run independently for each CD&R algorithm. The resolution advisory generated by the algorithms may consist of a heading change maneuver, an altitude change maneuver, a speed change maneuver or any combination of them. For each conflicting flight pair in the flight plan, one is treated as own ship (the aircraft which detects the conflict and initiates a resolution) and the other as the intruder (aircraft which does not resolve conflict and maintains course). In KB3D and FAA-TCASIII, resolution vectors are independent of each other such that implementing even one of them can resolve a conflict, whereas in the Billimoria and Rockwell-Collins case they require a combination of heading and speed change for a resolution maneuver. Conflicts are resolved by ownship in a non-cooperative manner. Figure 4.13 shows one of the conflict resolution maneuvers for a head on conflict. Note that after resolving a conflict, the ownship resumes back its own navigation by performing course corrections. Resolution maneuver vectors generated by the algorithms along with fuel and time cost are

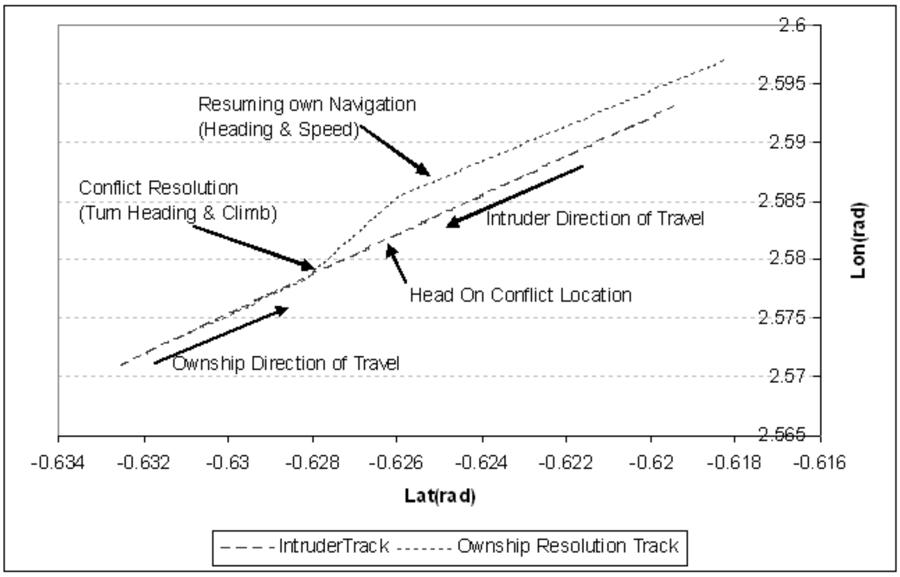


Figure 4.13: A conflict resolution maneuver for a head-on conflict scenario.

shown in Table 4.2. For all the three conflicting geometries simulated, all four algorithms successfully resolved the conflicts and no loss of separation is recorded. Metric data for the four algorithms as shown in Figure 4.14 suggest that the Force Field algorithm is very expensive in terms of fuel consumption. For cross over scenarios, the application of speed controls as part of the resolution vectors, proved the most expensive, followed by heading and altitude based maneuvers. The rule based approach (FAA-TCASIII) in general gives expensive maneuvers both in terms of fuel cost and time consumed which is due to lack of maneuver optimization in the algorithm, however it always gives resolution vectors within the aircraft's performance envelop. The Force Field algorithm suggested time consuming maneuvers and recorded high on the extra time required to execute a maneuver. The Geometric approach based algorithm provided a good CD&R mechanism

Table 4.2: Conflict resolution vectors generated for different conflict geometries along with extra fuel and time incurred.

Head-On Conflict						
Ownship:QFA405		Resolution	KB3D	Billimoria	Rockwell	TCASIII
Bearing(deg):	54.84	Heading(deg)	44.12	47.59	63.10	25.01
Altitude(kft):	39.00	Altitude(kft)	40.00	–	–	31.50
Speed(kn):	456.80	Speed(kn)	–	455.72	274.69	–
		Extra Fuel(kg)	30.8	17.4	64.3	66.28
		Extra Time(s)	27.6	20.0	94.0	111.0
Cross-Over Conflict						
Ownship:XCC999		Resolution	KB3D	Billimoria	Rockwell	TCASIII
Bearing(deg):	58.33	Heading(deg)	56.06	56.97	75.57	28.33
Altitude(kft):	32.00	Altitude(kft)	33.00	–	–	24.50
Speed(kn):	497.62	Speed(kn)	524.86	506.41	386.59	–
		Extra Fuel(kg)	23.23	2.04	45.6	145.6
		Extra Time(s)	27	60.0	55	167.4
Intrail Conflict						
Ownship:ZRW444		Resolution	KB3D	Billimoria	Rockwell	TCASIII
Bearing(deg):	54.35	Heading(deg)	59.02	61.40	60.59	84.21
Altitude(kft):	39.00	Altitude(kft)	40.00	–	–	31.50
Speed(kn):	456.80	Speed(kn)	305.03	437.37	239.10	–
		Extra Fuel(kg)	28.2	15.6	108	113.53
		Extra Time(s)	24.6	18.0	276.0	127.8

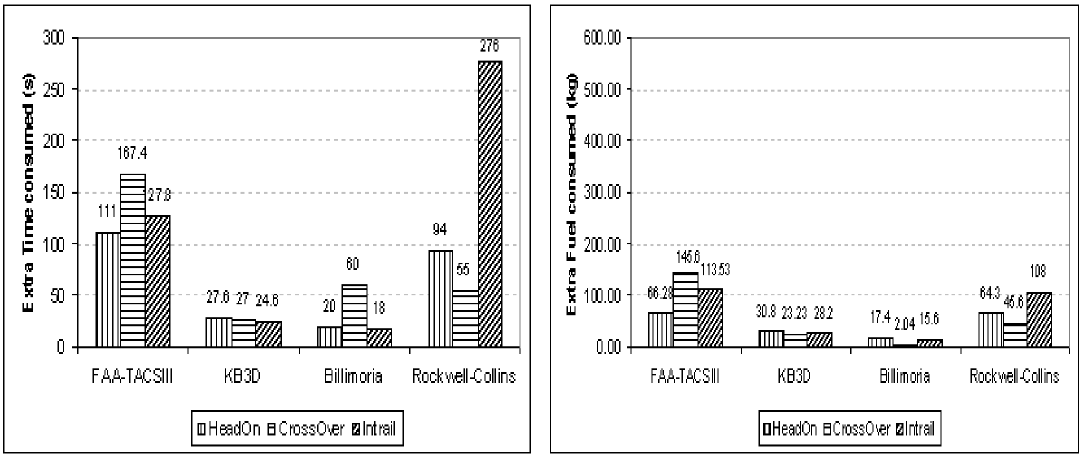


Figure 4.14: Extra fuel and extra time cost for the resolution vectors generated by the CD&R algorithms.

for all three geometries examined. However the set of resolution vectors it provided are sometimes beyond the aircraft's performance limits, this requires an extra function call to the BADA database to eliminate infeasible solutions. It can be deduced from the data that speed maneuvers proved to be very costly, followed by altitude maneuvers, heading change maneuvers resulted in less fuel and extra time consumed.

4.2.4 Discussion

CD algorithms can be exposed to complex conflict scenarios in Free Flight. Thus, they should be tested rigorously for robustness, and assessed on safety measures. Mathematical proofs are not sufficient because CD algorithms have many degrees of freedom to optimize. Therefore, it is highly desirable to have a methodology to generate complex conflict scenarios for evaluation of CD algorithms on a common platform. Apart from rigorous evaluation another advantage of such a methodology would be to examine the conflict characteristics data and provide insights into the anomalies in the algorithm or complex conflict geometries where a CD algorithm fail (i.e. produce FA or MD). This will be returned to in Chapter 6.

4.3 Airborne Weather Avoidance

4.3.1 Introduction

Weather is the single largest contributor to delay in the air traffic control system and is a major factor in aircraft safety incidents and accidents (Nolan, 2004; NTSB, 2001). Weather disturbances pose a serious threat for aviation on nearly a daily basis worldwide and they account for approximately 70% of all delays in U.S. National Airspace alone (National Research Council, 2003). Even a small weather disturbance like rain fall can lead to water on runways which further lead to longer braking distance required by aircraft. This further propagates to low acceptance rate of aircraft by airport approach ATCs, forcing the en-route ATCs to hold flights in the air and delaying taking on flights from neighboring airspace. All these lead to delay propagation thousands of miles away resulting in grounded planes, cancelled flights, higher fuel burns etc. (Krozel *et al.*, 2003). Weather disturbances can also severely damage the airframe of the aircraft. They can potentially damage the navigational and electronic equipments leading to pilot's loss of control. During the enroute phase convective weather may appear as a cluster of convective weather cells and may block a jet route. This leads to aircraft being shifted from one jet route to another causing congestion and reduced airspace capacity. However, narrow corridors still exist through which aircraft can pass.

In Free Flight the pilots will have a larger role in real time route planning given airspace constraints such as weather and traffic (RTCA, 1995). Enroute aircraft require a weather avoidance algorithm that takes into consideration multiple objectives besides avoiding weather cells of high severity. These objectives can be minimizing changes in heading and altitude while negotiating around weather cells and minimizing deviation from the planned trajectory. Further, in the absence of jet routes in Free Flight, the weather avoidance algorithm must also account for neighboring traffic such that the avoidance trajectory must not lead to possible collisions. The relatively fast changing nature of convective weather cells and neighboring traffic makes this a challenging problem.

4.3.2 Background

Given the far reaching benefits to flight safety, fuel savings and capacity enhancement resulting from efficient weather avoidance systems, there has been a number of research and industry initiatives in this field.

The most significant of them in terms of research output and trials are Cockpit Weather Information (CWIN) and User Request Evaluation Tool (URET). CWIN (Tu, 1996) is a NASA and Honeywell Laboratory's joint initiative currently under development to distribute graphical weather information to the cockpit in near-real time. In an initial evaluation done for a prototype CWIN weather avoidance tool by Dorneich (Dorneich *et al.*, 2003), it is found that with the aid of a weather avoidance system there is a considerable increase in safety of a flight as well as significant fuel efficiency which is largely attributed to optimized weather avoidance routes. The MITRE Corporation's Center for Advanced Aviation System Development (CAASD) and the U.S. Federal Aviation Administration (FAA) are currently developing a set of enhancements to the URET (Heagy and Kirk, 2003) to support the controller in severe weather situations and to assist the pilots in negotiating bad weather. The weather avoidance system utilizes Dijkstra's Algorithm to compute the shortest path around the buffered severe weather polygons. This enhancement is in the form of a severe weather display used in conjunction with Graphic Trail Planning (GTP) allowing controllers to enter vector maneuvers for severe weather avoidance into the Host computer, thus improving the quality of flight trajectories and increasing the likelihood that aircraft will receive the most efficient routes possible in severe weather situations.

Recently Krozel (Krozel *et al.*, 2004) investigated the problem of synthesizing weather avoidance routes in the transition airspace using an algorithm based on the grid search method. They formalized the problem as a weighted regions problem in which routes obey Snell's Law of Refraction as a local optimality criterion. The generated weather avoidance paths are compared to three alternatives: variations of the Standard Arrival Routes, a geometric optimization solution synthesizing multiple non-intersecting routes, and two different Free Flight approaches in which aircraft fly weather avoidance routes using a greedy prioritization method. It is shown that an increase in airspace capacity is possible using efficient weather avoidance algorithms. However, this approach takes into consideration static weather cells only without any traffic. Moreover, the problem approach is single objective i.e. avoiding convective weather cells with no consideration of other flight optimization parameters.

The use of heuristics algorithms such as A-Star for weather avoidance in General Aviation is investigated by (Bokadia and Valasek, 2001). The algorithm successfully computes the safest (in terms of weather) and shortest flight path around hazardous weather cells. However, the solutions mechanism does not undertake the optimality of solution and aircraft safety constraints and operates in a two dimensions abstract environment.

Prete and Mitchell (Prete and Mitchell, 2004) designed a Flow-Based Route Planner (FBRP), which computes routes for multiple flows of aircraft in the transition airspace using the A-Star algorithm. The planner allows for a variety of constraints on the computed routes based on type of aircraft, including hazardous weather patterns, turn and curvature constraints, and the horizontal separation standard. The algorithm searches for optimal routes within specified constraints using a search within a discrete network that models the geometry of the airspace. One of the key areas for extending the work, as highlighted by the authors, is "having a set of optimal solutions to choose from, and to be able to search multiple routes in the presence of multiple airspace constraints".

Nilim (Nilim *et al.*, 2004) tackled the problem of dynamic convective weather avoidance for multiple aircraft flow management by using a stochastic dynamic programming algorithm, where the evolution of the weather is modeled as a stationary Markov chain. The approach provided a dynamic routing strategy for multiple aircraft that minimized the expected delay of the overall system while taking into consideration the sector capacity and traffic constraints. However, their system is limited to abstract modeling where the airspace is presented in 2–D and no physical constraints of an aircraft, such as heading changes, acceleration/deceleration limit, speed, etc, are considered. Moreover, the approach could only minimize a single objective, that of the distance traveled, without considering other flight optimization objectives such as altitude and heading changes.

Recently, Pannequin (Pannequin *et al.*, 2007) developed a model predictive control based algorithm, which used the Hamilton-Jacobi equation to find the fastest trajectory for each aircraft subject to weather conditions, traffic constraints and wind profile. The algorithm could generate trajectories that avoid bad weather cells as well as potential collisions between aircraft. However, their algorithm is limited to static weather avoidance, which did not take into account the dynamic changing nature of weather. Moreover, their system is centralized, in which information from all the aircraft involving in the situation had to be fed into the algorithm so that the system could compute their trajectories. This centralized design suffered from limited scalability and poor robustness, and is subjected to equipment failure at the Air Route Traffic Control Center level.

Most existing weather avoidance systems/algorithms do not take into account the dynamic nature of convective weather pattern in the high-fidelity realistic flight environment. They also lack a consideration of the flight performance envelop in trajectory generation and have limited capability of accommodating multiple flight optimization objectives. As identified by Krozel,

"...a key evolution of future air traffic management system in weather avoidance is to dynamically assign routing to aircraft and adjusting the routes to the size and shape of weather constraints as they change." (Krozel *et al.*, 2006),

The key objectives that we identified in developing a weather avoidance algorithm for enroute traffic constrained airspace are:

- to search multiple routes in the presence of multiple airspace constraints which include relatively fast moving convective weather cells and neighboring air traffic.
- to accommodate flight optimization objectives such as minimizing changes in heading, altitude and deviation from planned trajectory.
- to take into consideration aircraft's performance parameters.
- to be able to generate weather free trajectories such that there is no loss of separation with neighboring traffic.

- to provide the set of optimal solutions to choose from.

In this chapter, we present a weather avoidance algorithm which is conceptually based on space-time search using a combination of two heuristic search mechanism viz. Ant Colony Optimization (Dorigo and Stutzle, 2004), a meta heuristic search technique inspired by functioning of real ants and A-Star (Russell and Norvig, 1995) which is an informed heuristic search technique. In a later section we describe them in detail and how we adopted them to suit our problem domain.

In the next section we explain the weather avoidance problem and state space pre-processing mechanism followed by the design of the algorithm using ACO and A-Star. We then explain the experimental design and conclude with the results and discussions.

4.3.3 Algorithm Design

Finding weather avoidance trajectories can be seen as path planning in four dimensions (i.e. time and the physical 3-D space). In ATM this problem attains unique dimensions due to safety and airspace constraints posed on it. Apart from the hard safety constraints like aircraft's performance envelope (turn and climb) and potential conflicts, the objectives are to minimize severe weather cells impact, minimize changes in heading, minimize changes in altitude (climb and descent), and minimize the deviation from the planned trajectory. We will first formulate the problem for avoiding dynamic weather cells and then extend it to traffic avoidance.

4.3.3.1 Problem Search Space

Weather cells can spread in an area extending up to 1000 square nautical mile (Bureau of Meteorology, 2003). Finding feasible solutions in such a large search space can be demanding from a safety and time perspective. Hypothetically, if the number of states in the search space can be reduced without compromising the solution quality, the reduction of state space size will result in faster convergence.

By pre-processing infeasible transitions in the search space, the search is guaranteed to produce feasible solutions. We eliminate the state space recursively starting from an entry point in the 4-D grid and performing forward recursion on different layers. At each layer we eliminate states which violate the problem constraints before moving to the following layer. Thus, we can guarantee that the resultant state space contains feasible transitions only.

4.3.3.2 Problem Constraints

The following safety constraints are considered in the algorithm design:

1. Performance constraints: These constraints include restrictions based primarily on the operating limitations of the aircraft. Restrictions, such as maximum operating altitude, climb/descent rate, and maximum permissible turn, govern the degrees of freedom available for weather avoidance maneuvers.

2. Airspace Hazard Constraints: These hazards are present in the airspace when a particular region is inaccessible for maneuvering, either reserved for Special Use (SUA) or that a particular maneuver may lead into conflict situation with other neighboring aircraft in the airspace. In this work, neighboring traffic is considered as a hazard constraint.

4.3.3.3 Objectives

The following objectives are considered in the algorithm design:

1. Minimize weather cells encountered: The prime objective of the algorithm is to avoid weather cells, especially those with high precipitation (radar reflectivity). If there is no possible path through the bad weather cells within the search bounds, the algorithm finds the trajectory which passes through weather cell(s) with least precipitation.

2. Minimize deviation from the flight plan: Another objective of the algorithm is to find paths which satisfy the constraints and have the least deviation from the original planned trajectory.

3. Minimize heading changes: Solution trajectories that involve too many heading changes will cause higher fuel cost as well as discomfort to the passengers. This may lead to degradation in aircraft performance as the aircraft may not be able to follow sharp turn angles due to ”start of turn distance” limitation.

4. Minimize climb and descent maneuvers: In their mature stage, weather cells shows a high tendency of upward and downward drafts with lightening. These drafts sometimes reach high wind speeds of 8000ft to 10,000 ft per minute(Bureau of Meteorology, 2003). Flying above or below a thunderstorm cell by climb or descent maneuver is never advisable(Peter, 1995). Thus solutions should minimize changes in altitude.

4.3.3.4 Problem definition

The problem can then be stated as follows: Given a 4-D grid of dimensions i (latitude) x j (longitude) x k (altitude) and time t, an entry point x (start manoeuvre point) and an exit point y (end manoeuvre point), locations of weather cells and their severity in space and time in this grid, find the set of routes between x and y on the given objectives satisfying the problem constraints.

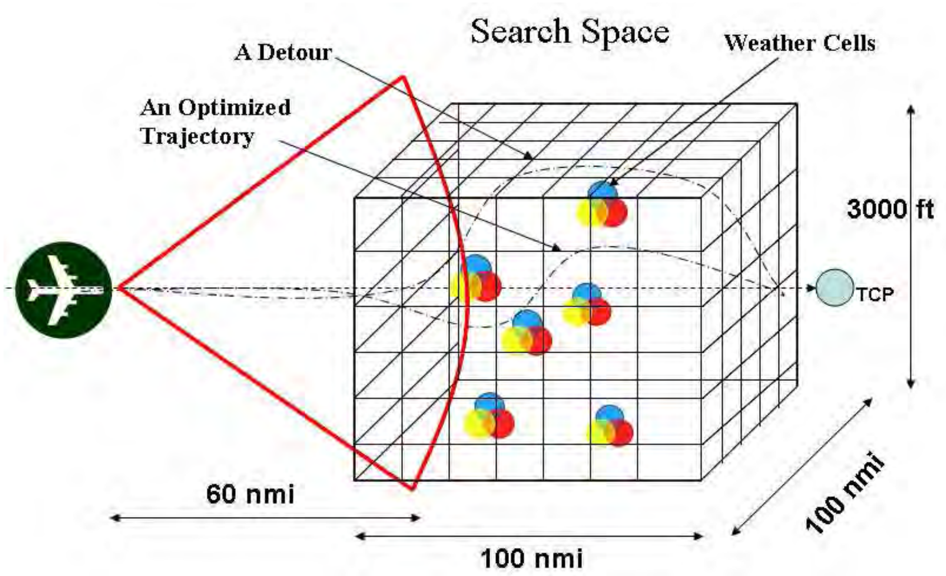


Figure 4.15: Conceptual representation of weather avoidance algorithm design.

Weather cells are simulated in the enroute airspace by assigning a radar reflectivity (a measure of thunderstorm intensity) value between 5 and 50 by using a uniform distribution random number generator. All weather cells are dynamic in nature and possess a speed and direction. Each cluster of bad weather cells comprises of 6 to 12 thunderstorm cells of dimension 10 nm X 10 nm X 3000 ft (Peter, 1995). As shown in Figure 4.15, the algorithm upon detection of weather cells at a distance of 60nm, (weather radar range) generates a three dimension volume around them. This airspace volume is of dimension 100 nm X 100 nm X 3000 ft.

This search space is then discretized in a grid of 10 x 10 x 3 (300 nodes), which gives enough volume (1000 sq nautical mile) to cover the entire weather pattern. Each cell in the grid forms a state in the graph. The arcs of the graph represent possible transitions from one state to another. The state space is then processed for aircraft performance constraints. This performance constraint data for the given aircraft is derived from the Eurocontrol's Base of Aircraft Data (BADA) (Eurocontrol, 2004) aerodynamic model.

To take into account the dynamic characteristic of the weather, the 3-D grid is extended to 4-D by including time as the fourth dimension. The simplest way to encode the time dimension is to take a snapshot, i.e. aircraft positions and weather cells' positions and intensities, of the discretized grid at every time step. However, since the rate of change Δt of the grid is limited by the minimum time that takes either a weather cell to move from a node i to the next node j or an aircraft to move from i to j, snapshots of the 3-D grid at time intervals Δt are recorded. The time $t_W = \frac{d_{ij}}{V_W}$ for a weather cell to move from node i to node j is limited by the speed V_W of the weather cell and the distance d_{ij} between nodes i and j. Similarly, the time $t_A = \frac{d_{ij}}{V_A}$ for an aircraft to move from node i to node j is limited by the speed V_A of the aircraft and the distance d_{ij} between the two nodes.

$$\Delta t = min(t_W, t_A) = min(\frac{d_{ij}}{V_W}, \frac{d_{ij}}{V_A}) \tag{4.20}$$

Require: Aircraft Performance Database D :(Rate of Heading Change, Rate of Climb & Descent, Altitude Ceiling, Max. Bank Angle, Max. Turn Angle, Max. Vertical Speed)
Grid G with i × j × k dimensions
Exit node position in the grid(sink)

for Node in G **do**
  Compute the turn angle, altitude change, rate of heading change require, rate of altitude change required, vertical speed required from each node (xyz) to the grid point (ijk) in G
  if Transition from node xyz to grid point ijk violates D **then**
    Eliminate the link
  else
    Retain the link
  end if
end for
return Grid G with links within safety envelop

Algorithm 3: The pseudo code for state space processing

The resultant 4-D grid is stored in a 4-D array data structure as an enumerated state space where each element of the array represents a point in the 4-D grid. Every array element stores the information about its snapshot index and position (latitude, longitude and altitude) and all the immediate next links (which do not violate aircraft's performance envelope) from that point in the grid. Furthermore for each link, the array element stores the heading change required, altitude change required, distance between the two nodes and the distance to exit point from that link. The search algorithm performs search on this pre-processed state space.

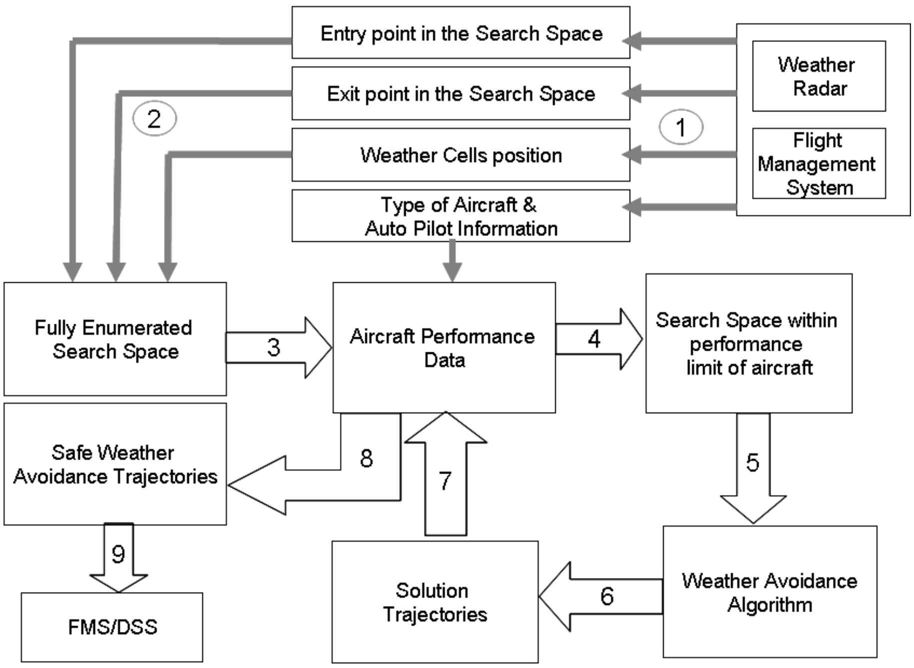


Figure 4.16: Data flow in weather avoidance system.

The data flow for the weather avoidance algorithm is shown in Figure 4.16, the numbered arrows indicate the sequence of flow of information between various modules of the weather avoidance system

4.3.3.5 Traffic Constraints

The concept presented above is extended to take into account the neighboring traffic that might have potential conflicting trajectories with the weather avoidance routes. The 3-D airspace in ATOMS is divided into a hyper rectangular 4-D grid structure as discussed in an earlier section: Modeling of the Free Flight Airspace. Every cell in the grid acts as a repository of information for the flights that pass through it. The information apart from the call-sign includes the estimated entry time and estimated exit time of an aircraft in that cell. When a flight is initialized (flight plan loaded into the system's memory), based on its intended trajectory, all the grid cells through which the trajectory passes are updated with the flight information. This information is updated/deleted wherever the aircraft changes its trajectory or passes by a cell in the airspace grid. It is assumed that all the aircraft are ADS-B (RTCA, 2003) equipped where they broadcast/receive state and intent information to/from the neighboring traffic.

Once the weather free routes are computed, they are converted into grid-cell plans which will be traversed if a particular route is chosen. If the ETA of the weather avoiding aircraft and of the traffic aircraft, at a given cell common in their trajectories, is within five minutes (based on average speed of transport aircraft of 300nm, and 10 nm distance between two neighboring cells in the grid), a collision is signaled and the route is discarded from the solution set.

As conceptually shown in Figure 4.17, the solution set contains two possible paths which avoid weather cells. The ETA of each path and the traffic passing through the weather grid are checked. The solution path that consists of cells where its ETA and the ETA of the other traffic is less than 5 minutes is discarded.

4.3.4 Search Algorithm

We have combined features of meta heuristic search techniques (ACO), to leverage their exploration nature, and informed heuristic search techniques (A-Star), to leverage their exploitation nature. These heuristic search algorithms are employed as they have shown several desirable properties for application in the transportation domain (Teodorovic, 2003)

4.3.4.1 A-Star

The A-Star algorithm is an informed heuristic search technique which minimizes the estimated path cost to a goal state (destination). At a given node n, the A-Star algorithm will choose the next state which minimizes the function $\Upsilon(n) = g(n) + h(n)$, where $g(n)$ gives the path cost from the current node i to the next node j, and $h(n)$ is the estimated cost from the next node j to the destination node (exit point).

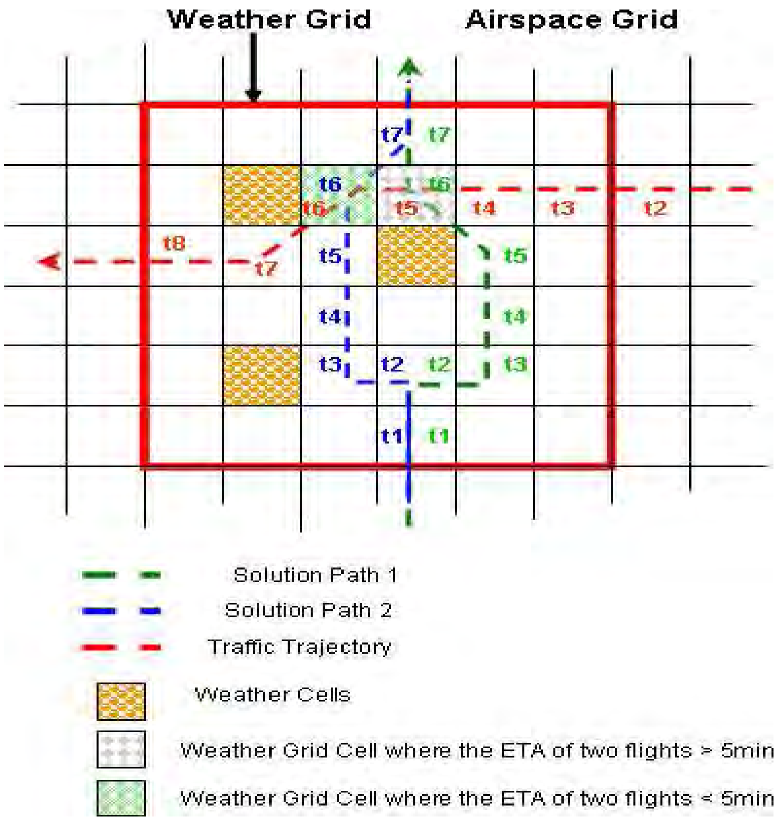


Figure 4.17: Possible paths with ETA of each path and traffic passing through the weather grid

We define the function $g(n)$ as the estimated cost on weather severity (WF), heading change (HF) and altitude change (AF) from current node to next node and $h(n)$ as the estimated distance (DF) from the next node j to the exit point in the search grid. The next-node distance cost is not incorporated in $g(n)$ as all neighboring nodes will be equidistant from current node given the grid structure.

$$\Upsilon(n) = (w_1 WF + w_2 HF + w_3 AF) + (w_4 DF) \tag{4.21}$$

where w_1, w_2, w_3, w_4 are dynamically initialized polar weights on the surface of a unit sphere.

4.3.4.2 Ant Colony Optimization

The ACO algorithm (Dorigo and Stutzle, 2004) is a meta-heuristic search technique for finding solutions to hard combinatorial optimization problems. The paradigm is based on the foraging mechanism employed by real ants attempting to find a shortest path to a food source. Ants use indirect communication via the environment by employing pheromone trails. In ACO, the transition rule which is the probability for an ant k on node i to choose next node j while building its path is given according to the following rule:

$$j = \begin{cases} arg\max_{u \in J_i^k}[\tau_{iu}(t)] \times [\eta_{iu}]^\beta & \text{if } q \leq q_0 \\ J & \text{if } q > q_0 \end{cases} \tag{4.22}$$

where $J \in J_i^k$ is a node that is randomly selected according to the following probability

$$p_{ij}^k(t) = \frac{[\tau_{ij}(t)] \times [\eta_{ij}]^\beta}{\Sigma_{l \in J_i^k}[\tau_{ik}(t)] \times [\eta_{ik}]^\beta} \tag{4.23}$$

where τ_{ij} is the pheromone value between the two nodes i and j, β controls the relative weight of η_{ij}, and q is a random variable uniformly distributed over $[0, 1]$.

Parameter q_0 in equation 2 is a tunable parameter ($0 \leqslant q_0 \leqslant 1$), where $q \leqslant q_0$ corresponds to an exploitation of the heuristic information of given objectives and the learnt knowledge memorized in terms of pheromone trails, whereas $q > q_0$ favors more exploration of the search space. We tune this parameter in the interval [0 1], to evolve different search behavior.

The visibility parameter η_{ij} in equation (3) represents the heuristics desirability of choosing node j when at node i, it can be used to direct the search behavior of ants by tuning β in the interval [0 1]. We have incorporated the inverse (since it is a minimization problem) of the A-Star evaluation function $\Upsilon(n)$ for the visibility parameter η_{ij} of the ACO.

$$\eta_{ij} = \frac{1.0}{\Upsilon(n)} \tag{4.24}$$

For the management of pheromone evaporation, pheromone trails and pheromone limits we have followed the Max-Min Ant System (MMAS) (Stutzle and Holger, 2000) methodology, which sets the initial pheromone to a maximum value and ensures that pheromone information remains in a defined bound, preventing local optima and early convergence of solutions. Algorithm

To evolve different search behavior we assigned weights iteratively for the A* objective function where for each objective a dynamic weight is assigned. In every iteration the weights are re-initialized using the hypercube rejection method Gentle (2003) where weights are spread on the surface of a unit sphere. This results in different weights (rela-

Require: Grid G with i $\times$ j $\times$ k dimensions
Entry node position in the grid(source)
Exit node position in the grid(sink)
initialize pheromone values $\tau_{i,j} \mapsto \tau_0$
repeat
for each ant i ϵ 1,...,m **do**
    initialize selection set S $\mapsto$ 0,1,...n-1
    let ant i construct solution π_i
end for
for all (i,j) **do**
    $\tau_{i,j} \mapsto$ (1-ρ)$\cdot$ $\tau_{i,j}$
end for
determine best solution for iteration π^+
if π^+ better that current best π^* then **then**
    set $\pi^* = \pi^+$
end if
for all (i,j) ϵ π^+ **do**
    $\tau_{i,j} \mapsto \tau_{i,j} + \rho/2$
end for
for all (i,j) ϵ π^* **do**
    $\tau_{i,j} \mapsto \tau_{i,j} + \rho/2$
end for
Untill condition of termination is meet

Algorithm 4: High level pseudo code for ACO + A^* algorithm

tive importance) assigned to the objectives in each iteration resulting in diverse solution paths.

The heuristics information represented by the visibility parameter η_{ij} is not static as it is in the case of ACO algorithm, it changes due to the dynamic weight initialization of A-Star objectives in every iteration. This parameter is also tuned in the interval of 0 to 1. A value of 1 will result in a typical A-Star behavior, and a value of 0 is indicative of the use of pheromone information only.

4.3.4.3 Different search behavior resulting from different pheromone update mechanism

We have investigated two different pheromone update strategies:

Pheromone update based on dominance : All valid solutions N from the current set P and archive set A (which stores the non-dominated solution obtained so far) are allowed to update the pheromone matrix. For pheromone update, each individual solution i in the archive A and the current set P is assigned a strength value $S(i)$, representing the number of solutions it dominates:

$$S(i) = \mid j \mid j \epsilon P_t + A_t \wedge i \succ j \mid \tag{4.25}$$

where $| \cdot |$ denotes the cardinality of a set, $+$ stands for multi set union and the symbol $\succ$ corresponds to the Pareto dominance relation extended to individuals ($i \succ j$ if the vector encoded by i dominates the vector encoded by j). Based on the S value, the raw fitness $R(i)$ of an individual i is calculated by summing the strengths of its dominators in both archive and current population.

$$R(i) = \sum_{j \epsilon P_t + A_t, j \succ i} S(j) \tag{4.26}$$

$R(i) = 0$ corresponds to a non-dominated individual,while a high $R(i)$ value means that i is dominated by many individuals. We then update the pheromone quantity as:

$$\forall \quad N \in [P, A] : \tau_{xy}(t) \leftarrow (1 - \rho) \cdot \tau_{xy}(t) + \rho \cdot (1 - R(i)) \tag{4.27}$$

where ρ is the pheromone decay parameter, and (x, y) represent the links in the solution path.We call this approach ACO-D.

Pheromone update based on scalarization: All the valid solutions N in the current population P are allowed to update the pheromone matrix based on the quality of the solution as:

$$\forall \quad N \in [P] : \tau_{xy}(t) \leftarrow (1 - \rho) \cdot \tau_{xy}(t) + \rho \cdot (1 - \psi(n)) \tag{4.28}$$

The quality of the solution is determined in terms of the scalarization $\psi(n)$ of n objectives. This mechanism is analogous to real world ants which deposits a higher quantity of pheromone returning from a rich food source. We call this approach ACO-S.

The solution trajectories generated by the search algorithm are converted into actual flight routes with artificial waypoints (Figure 4.18). These optimal trajectories can then be prioritized by the FMS based on pilot's preference. The flight plan is then amended where the modified section of the route based on the chosen solution trajectory is fed into the FMS which then flies the aircraft in auto-pilot mode and follow the updated route.

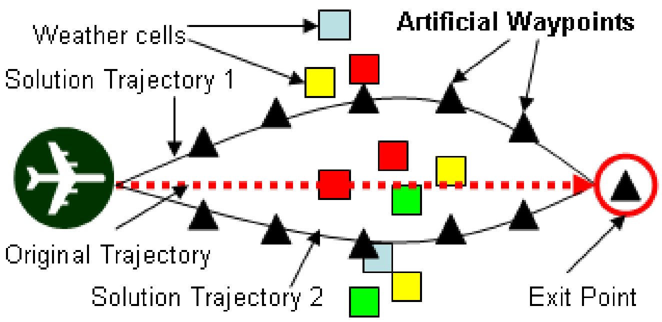


Figure 4.18: Flight routes with artificial waypoints generated by the ACO-based Weather Avoidance algorithm.

4.3.5 Experiment Configuration & Results

ATOMS is employed to generate weather patterns and traffic scenarios. Middle-East airspace (OBBI FIR) is used for weather modeling and flight simulation. Two flights in the opposite direction are simulated with great circle routing between their origin and destination. The cruising altitude is set to 27 kft for the flights. Two enroute weather patterns are simulated: clustered weather cells and distributed sparse weather cells (Figure 4.19). The weather cells are generated with a speed of 60kts (30m/s) and a direction of 2 degrees from the true North. The weather patterns are then activated at a time such that they obstruct the trajectory of the flights.

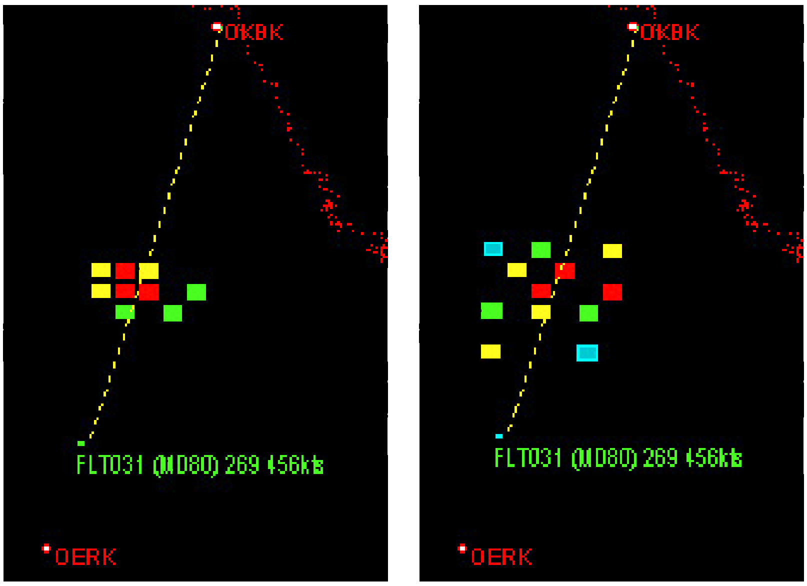


Figure 4.19: Snap shot of the different weather scenarios in the flight path of an aircraft. Left: clustered weather cells, right: distributed sparse weather cells.

We tested the algorithm in two scenarios: (A) an aircraft flies through bad weather cells that are dynamically changing through time, and (B) two aircraft, with a future head-on conflict, fly through dynamically changing bad weather cells. In the first situation, the algorithm has to find a set of solution trajectories that avoid any severe weather cell impact. In the second situation, the algorithm has to find a set of solution trajectories, which not only avoid the bad weather cells impact but also avoid conflicts between aircraft. All the generated flight paths have also to satisfy the aircraft's performance constraints.

To evolve different search behaviors and understand performance of ACO algorithm with heuristic search under various parameter configurations, we have examined the following combinations:

1. Use of different combinations of values of the exploration-exploitation parameter (q_0) and heuristic desirability parameter(β).
2. Pheromone update mechanism based on *Dominance* v.s. pheromone update mechanism based on *Scalarization*.

We performed preliminary experiments to determine both the size of the ant colony, and the number of iteration required for convergence to a solution. We found that 30 ants with 300 iteration provides a good solution convergence. Based on MMAS rules we set $\rho = 0.9$ for the pheromone evaporation. τ_{max} was set to a theoretically largest value (Stutzle and Holger, 2000). Combinations of the following values of parameter β=$\{0.1, 0.3, 0.5, 0.7, 0.9\}$ and parameter q_0=$\{0.1, 0.3, 0.5, 0.7, 0.9\}$ were examined to understand their effect on ACO with heuristic search performance.

To measure the performance of these aspects we have used the Median Attainment Surface (Zitzler *et al.*, 2003), which quantifies how much an algorithm A is better (strictly dominates) than algorithm B. We also used ANOVAN analysis which performs multi-way (n-way) analysis of variance (ANOVA) for testing the effects of multiple factors (grouping variables) on the mean of the given vector.

4.3.5.1 Dynamic weather avoidance without traffic constraint

The algorithm generated *322* possible routes which are weather impact free in the given scenario. All the routes are optimum on the given objectives and within the aircraft's performance parameters. Figures 4.20 and 4.21 present a series of snapshots, in terms of time, of flight FLT031 (OERK to OKBK) through two different weather scenarios. The snapshots show one of the optimized weather-free trajectory generated by the algorithm through bad weather cells when the cells are moving dynamically. The search space are drawn as the dotted grid around the weather cells.

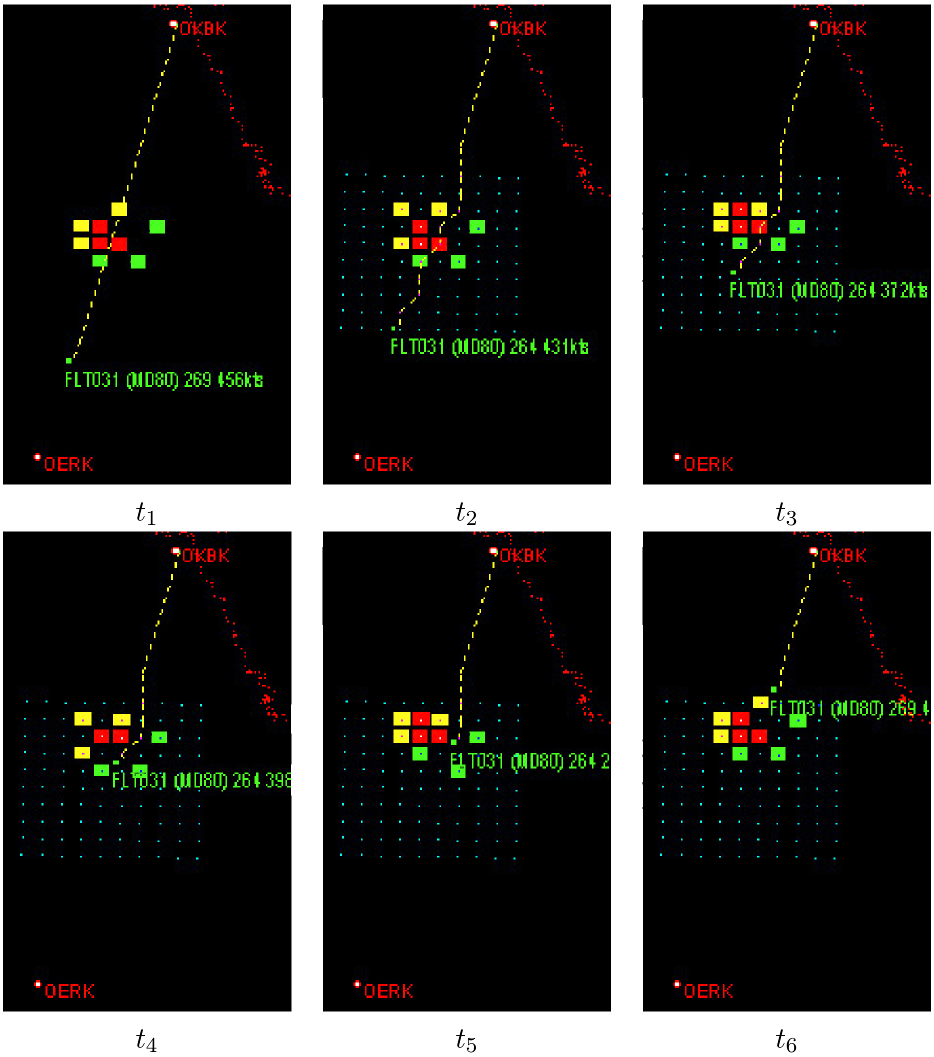
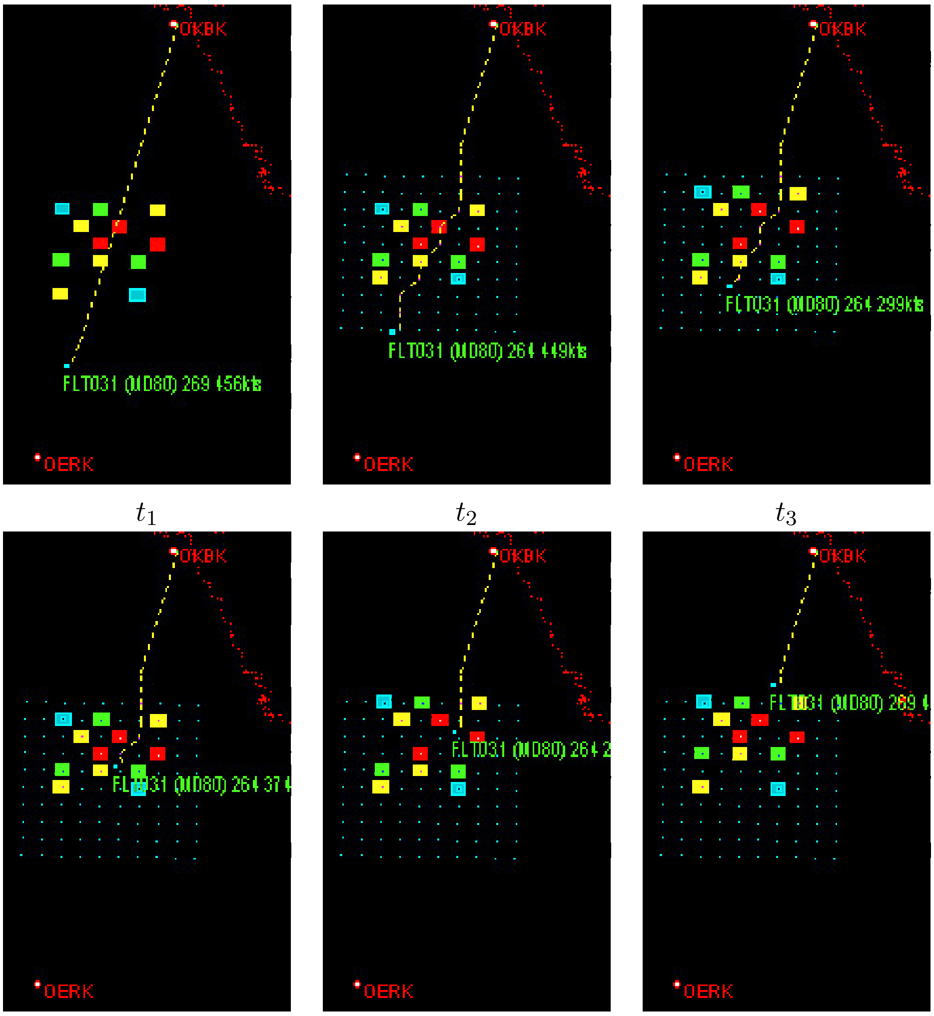


Figure 4.20: Clustered Weather Scenario: Snapshots (time progression $t_1 \rightarrow t_6$) of flight FLT031 through dynamically moving bad weather region.

Figure 4.22 presents all the optimal routes generated by the algorithm which avoids the bad weather cells and are optimum on the given objectives. As shown in Figure 4.22, the algorithm finds not only one but many possible routes that avoid the weather cells, and are optimal on the given objectives. The thicken line presents the selected weather avoidance trajectory which is then executed by the FMS.



t_1 t_2 t_3

t_4 t_5 t_6

Figure 4.21: Distributed Weather Scenario: Snapshots (time progression $t_1 \rightarrow t_6$) of flight FLT031 through dynamically moving bad weather region.

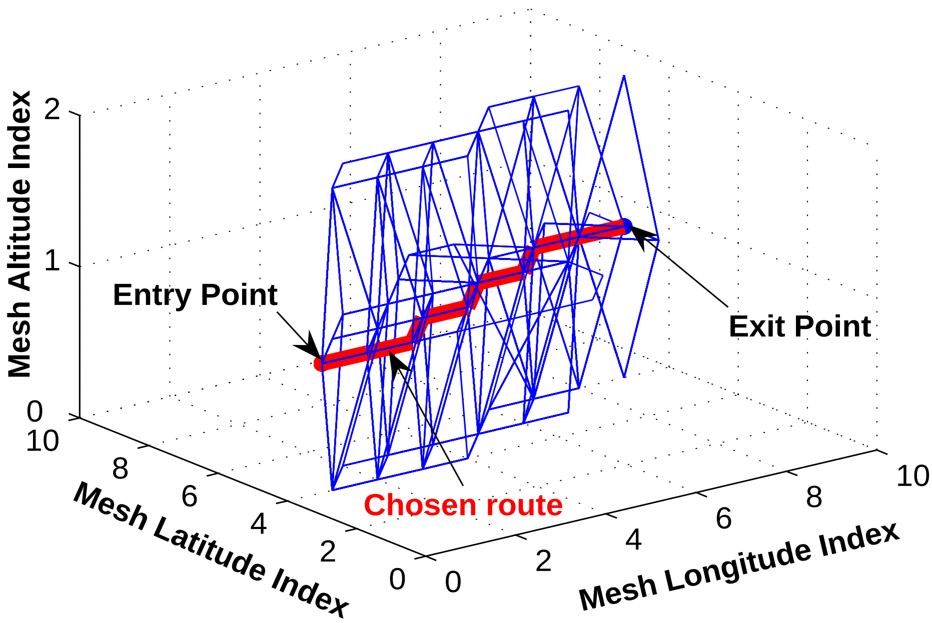


Figure 4.22: 322 weather-free routes generated by ACO. The thickened line represents the chosen flight route.

4.3.5.2 Aircraft Performance Constraints

For the selected trajectory which is flown by the aircraft we examined the performance constraints from a safety perspective. For the given *MacDonnell Douglas-80* aircraft (MD80) we plotted its rate of accelerate-decelerate (ROAD), rate of climb and descent (ROCD) and rate of heading change (ROHC) for the given maneuver.

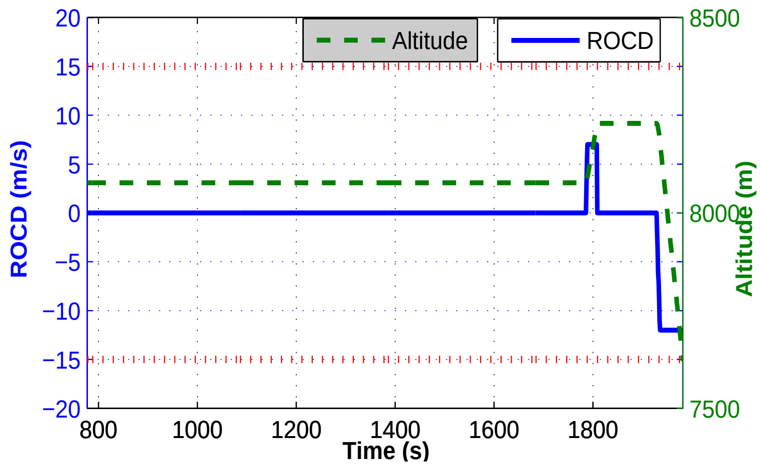


Figure 4.23: Altitude and Rate of Climb/Descent of FLT031 as it passes through dynamically moving bad weather cells. Boundary lines represent the aircraft performance envelop $(max/min\, ROCD = \pm 15m/s)$

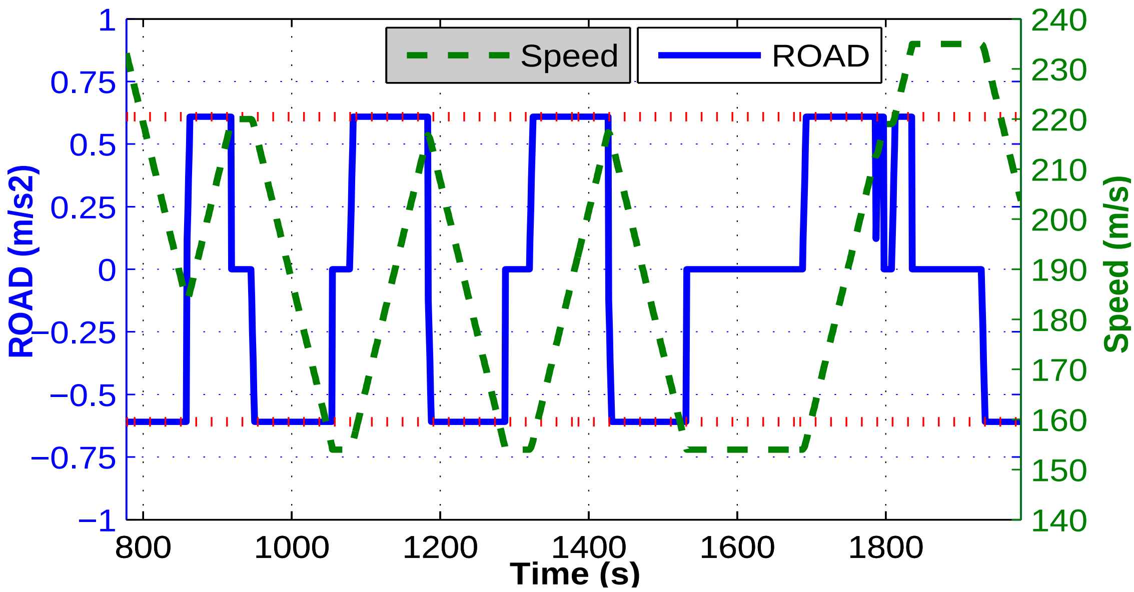


Figure 4.24: Speed profile and Rate of Acceleration/Deceleration of FLT031 as it passes through dynamically moving bad weather cells. Boundary lines represent the aircraft performance envelop ($max/min\, ROAD = \pm 0.7 m/s^2$)

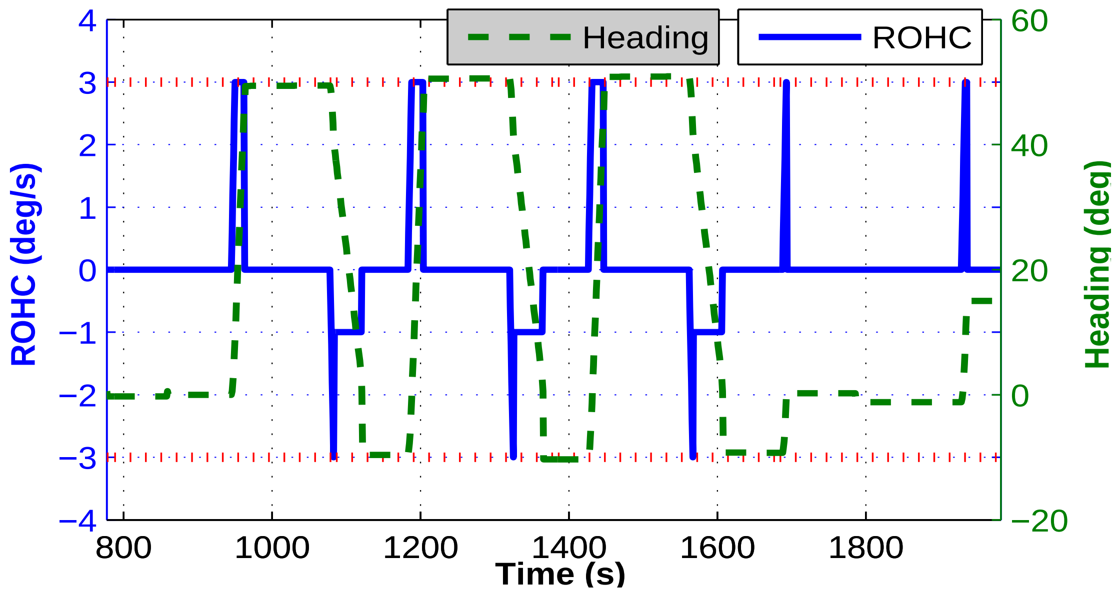


Figure 4.25: Heading and Rate of Heading Change of FLT031 as it passes through dynamically moving bad weather cells. Boundary lines represent the aircraft performance envelop ($max/min\, ROHC = \pm 3 degree/s$)

Figures 4.23, 4.24, 4.25 shows that aircraft maneuvers are within the bounds of its performance envelop.

4.3.5.3 Dynamic weather avoidance with traffic constraint

Figure 4.26 presents snapshots of two potentially head-on conflicting flights, both passing through the bad weather region. The algorithm can generate solution trajectories that avoids bad weather cells as well as maintain safe separation distance between the aircraft. The sequence of events of detection and avoidance happens as follows:

1. FLT031 detects the weather cells and uses the weather cell information and intent information of FL032 to generate the weather and traffic avoidance trajectory.
2. FLT031 updates its flight plan and propagates intent information.
3. FLT032 detects the weather cells and uses the weather cell information and updated intent information of FLT031 to generate the weather and traffic free route.
4. FLT031 and FLT032 fly through the weather cells without encountering one another and maintains safe separation from each other.

Figure 4.27 shows the horizontal and vertical separation between the two aircraft during their maneuvers. As shown, the minimum horizontal separation between the two flights during the maneuver is 21.44 nm. In other words, the algorithm can generate weather avoidance and collision free trajectories in a traffic constrained airspace.

The solution set contains trajectories with different trade offs for the given objective. For all the given combinations of q_0 (exploration-exploitation) and β (heuristic desirability) parameter, performance measures were computed, based on the outcome we plotted color map which displays the solution quality over other configurations, where darker shade indicates a good performance of the strategy parameter combination for the implied configuration in relation to all other combinations, and a lighter shade shows bad performance.

4.3.6 Median attainment surface

We computed the *median attainment surface*(MAS) for the non-dominated set of solutions generated by the ACO with informed heuristics search for the two approaches

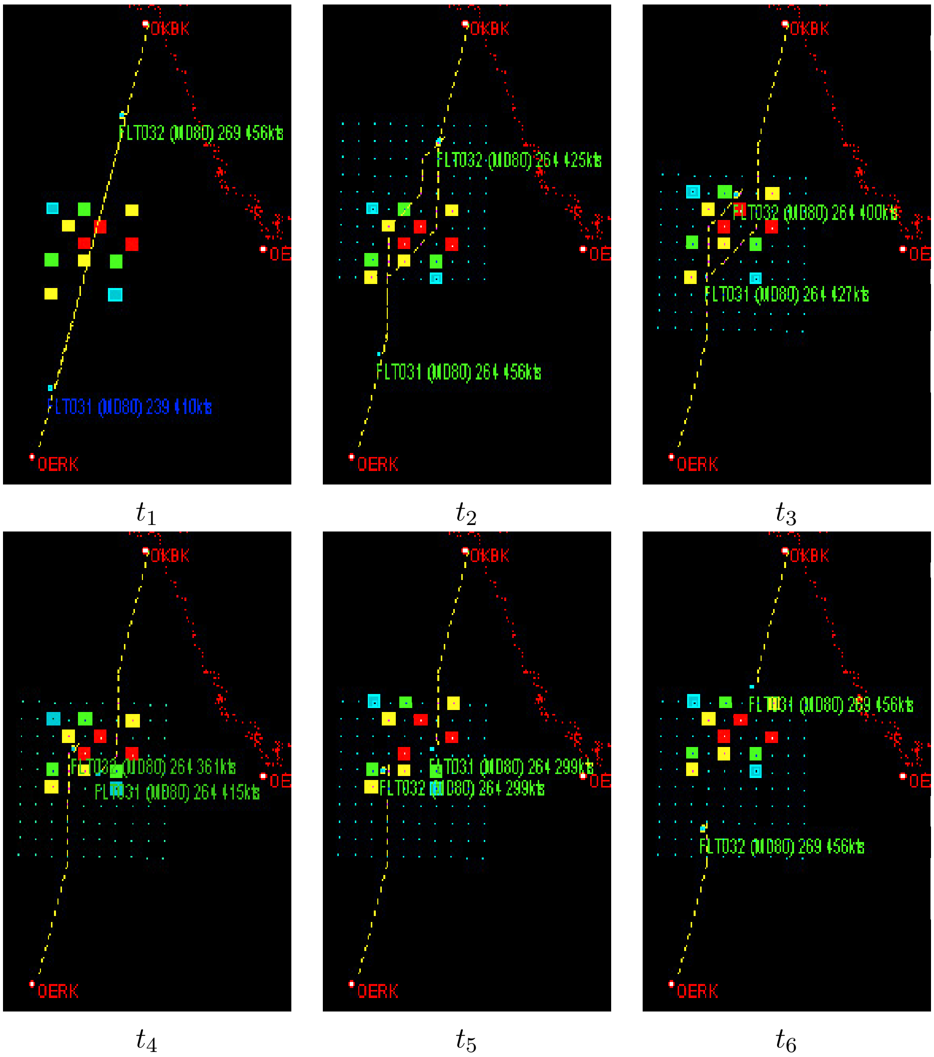


Figure 4.26: Distributed Weather Scenario: Snapshots (time progression $t_1 \rightarrow t_6$) of potentially conflicting flights FLT031 & FLT032 via dynamic bad weather region.

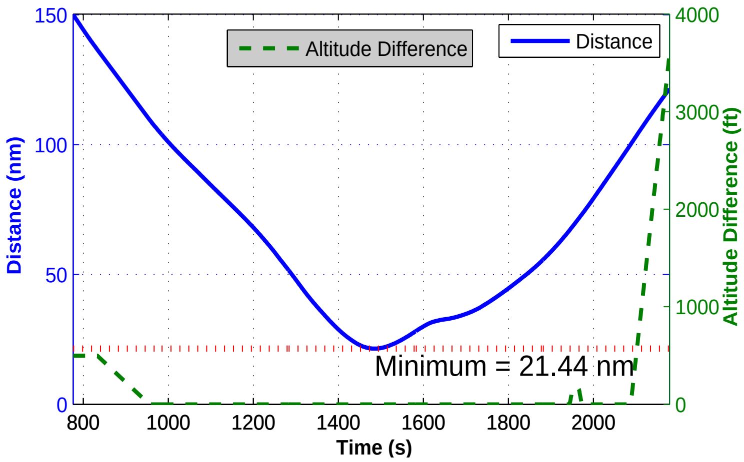


Figure 4.27: Horizontal and vertical separations between two aircraft, FLT031 & FLT032 avoiding the bad weather cells. Minimum horizontal separation is 21.44nm

and for different parameter combinations. As shown in figure 4.28 and judging from the

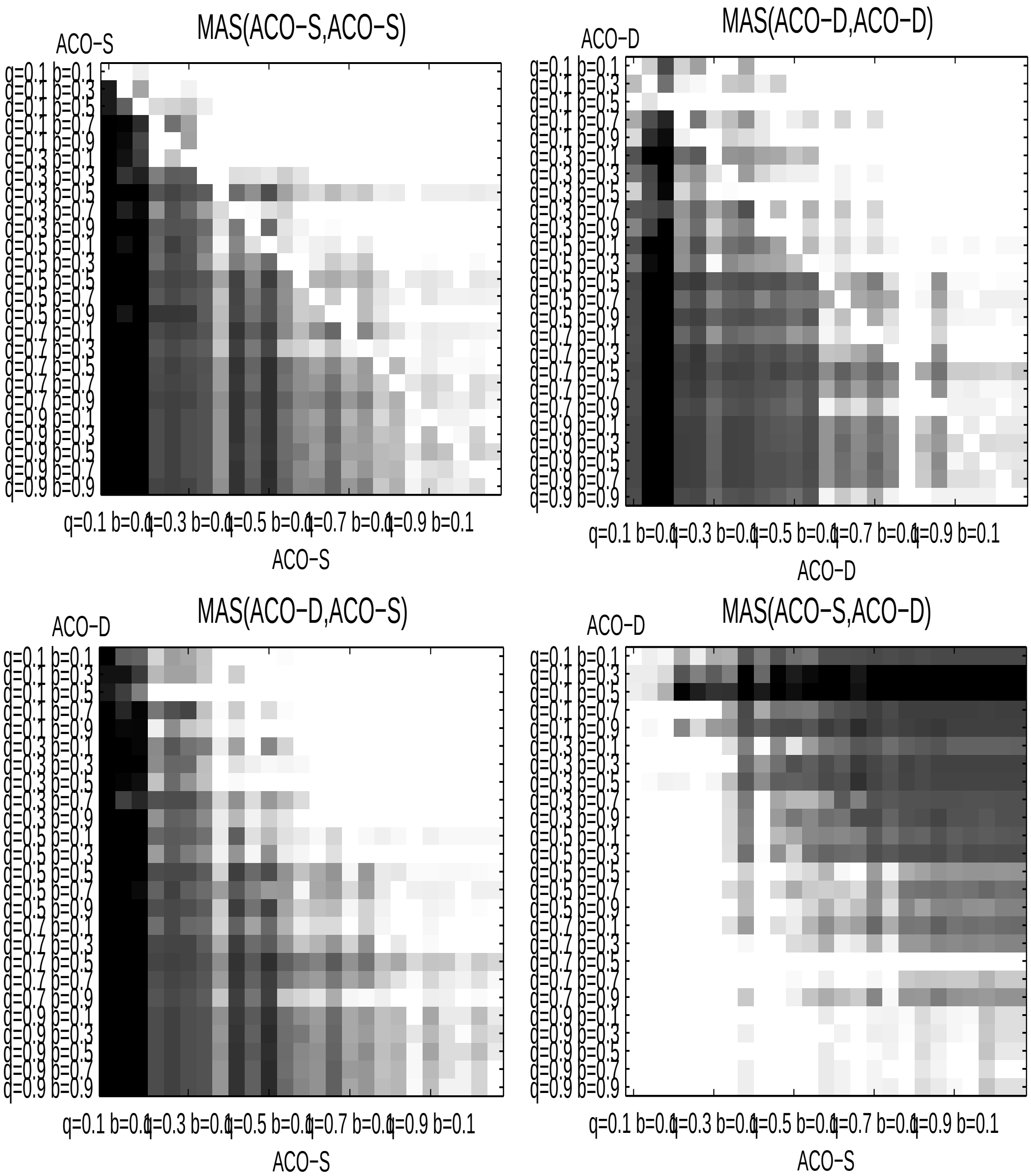


Figure 4.28: MAS performance of ACO-S with in its own set (top-left), ACO-D with in its own set (top-right), ACO-D (y-axis) compared with ACO-S (x-axis) (bottom-left) and ACO-S (x-axis) compared with ACO-D (y-axis) (bottom-right). Darker shade indicates a good performance of the strategy parameter combination for the implied configuration in relation to all other combinations, and a lighter shade shows bad performance.

average darkness, some interesting observations can be made out of them. In dominance based pheromone update mechanism, we can see that when q_0 is low (q_0=0.1 and q_0=0.3) then high exploration (β=0.7) of search space gives good results. When q_0 is medium (q_0=0.5) then high to very high exploration($\beta \geq$0.7) is results in good solutions. When

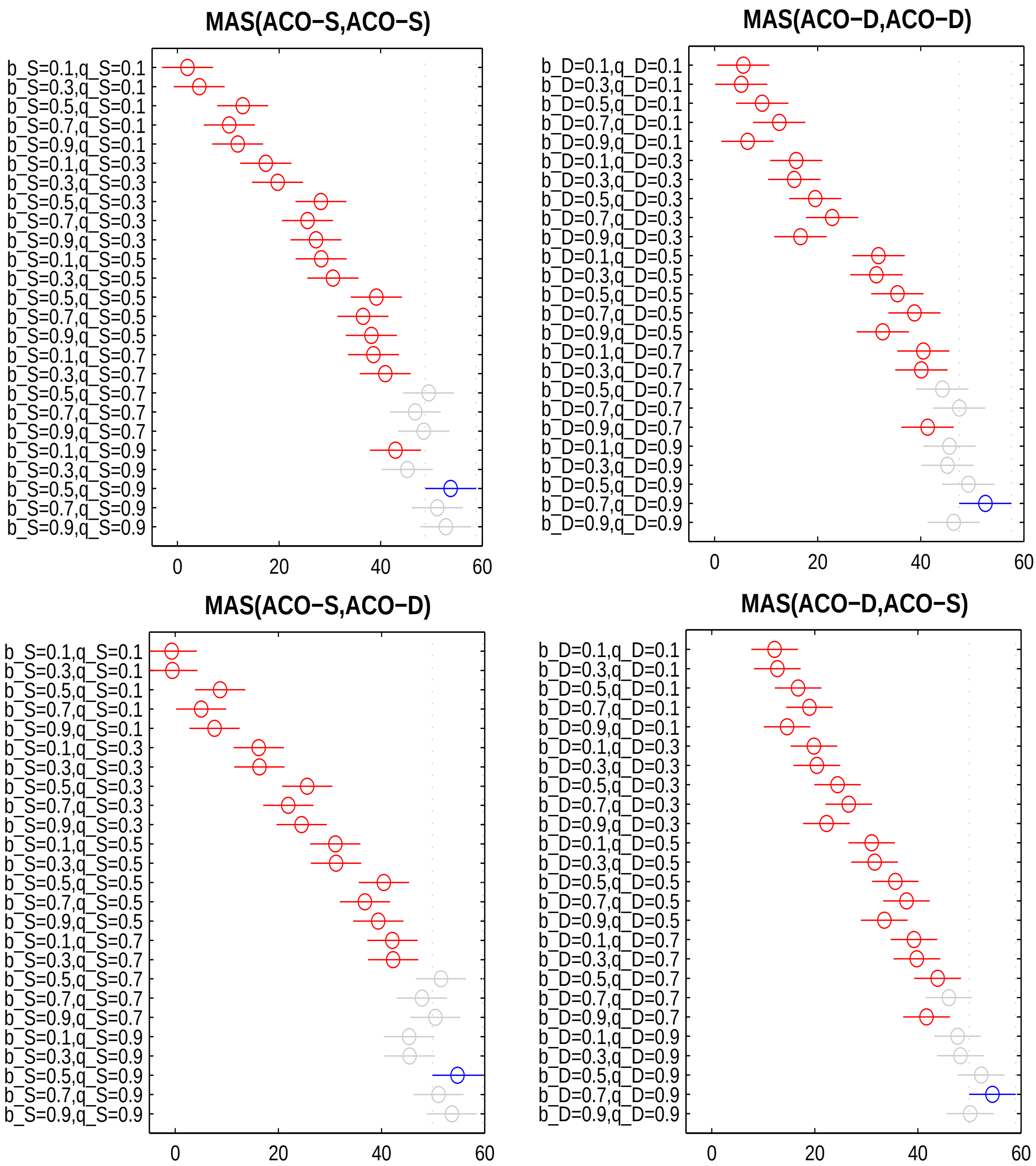


Figure 4.29: ANOVAN analysis (showing means and 95% confidence interval) grouped on different set of q_0 and β parameters for the two approaches.

q_0 is high to very high ($q_0 \geq 0.7$) then very high exploration (β=0.9) is undesirable. Best Solution were obtained for parameter combination of q_0=0.7 and β=0.5. In scalarization based pheromone update mechanism, we can see that with increasing value of q_0 we get very good solutions and β has very marginal effect on the solution quality. However, for all the good solutions we can see that parameter β has a value of 0.5. In ACO-D over ACO-S, the color map indicates that high to very high value of exploitation ($q_0 \geq 0.7$) in ACO-D gives good solution quality over ACO-S and similarly in ACO-S over ACO-

D, with high exploitation ACO-S gives good solution quality over ACO-D. This indicates that very high exploitation of learned knowledge in the search space coupled with medium to high use of heuristic information can lead to good quality solutions for this kind of problems.

To get a more conclusive picture we then used ANOVAN analysis on MAS data. Figure 4.29 shows the ANOVAN analysis of ACO-S and ACO-D with different factors q_0 and β; each sub–figure presents the means and 95% confidence interval of the means of each data groups. Figure 4.29 top left and right shows ANOVAN for ACO-S, ACO-D for various parameter combination. It can be seen that the best strategy in both the approaches is high exploitation of embedded information in the environment. with increased exploitation we get good results, but if too much exploitation of heuristic is done the solution quality decreases. In ACO-S this however depends on medium favor for heuristic information available about the system. Whereas in ACO-D this depends on high favor for heuristic information available about the system. Thus in dominance based approach heuristic information plays a greater role than in scalarization based approach. It is concluded that very high exploitation of learned knowledge in the later part of search coupled with low to medium exploitation of heuristic information available about the system in the early part of search leads to good quality solution.

4.3.7 Discussion

A safety inherent design utilizing intent information can provide a good framework for the weather avoidance problem in an enroute traffic environment. Pre-processing the state space before search helps in reducing the search space and making the search manageable. Heuristic search provides a good mechanism for incorporating multiple objectives and combining the virtues of exploration of environment and exploitation of information available about the environment. The algorithm generates a set of solution trajectories in different weather patterns in a traffic constrained airspace successfully. All the trajectories are optimal on the given objectives. Moreover, the trajectories generated are all flyable, such that they do not violate the performance parameter's range of the aircraft. The solution quality of the trajectories is strongly affected by exploitation of the embedded information in the environment rather than the available heuristics. High heuristic desirability leads to poor quality solution indicating early convergence to sub-optimal solutions due to more weight on the local information. Overall the best strategy is to exploit more what is learned about the search space and use average local information.

On the downside, since the algorithm takes into consideration the performance parameters of an aircraft, which restricts the turn angle for heading change and bank angle for vertical maneuver, it may not generate optimum avoidance trajectories if the aircraft is very close to weather cells at the time of detection. Finally, the algorithm performance is based on the assumption that the traffic intent information and weather cells information, which is provided through ADS-B transmission and on-board Doppler radar respectively, is available and error free.

4.4 Cockpit Display of Traffic Information

In this section, we describe the design and development of a prototype for cockpit display of traffic information (CDTI) which is implemented in ATOMS to investigate various design issues including the amount of traffic information to be displayed, level of intent information, and constrained airspace information display. Operational features and applications of the CDTI prototype developed are briefly discussed.

4.4.1 Introduction

As being mentioned in the background section, in Free Flight the pilots will play an active role in real time trajectory planning given airspace constrains which include traffic, weather, SUAs etc. Cockpit Display of Traffic Information (CDTI) technology (Alexander and Wickens, 2001) will provide pilots with a traffic display that will be more global than that in the current TCAS situation display. It will allow pilots to determine the state and intent information of those aircraft in close proximity to themselves and may also provide other useful and time critical information such as convective weather cells, special use airspace, and help in path replanning (Johnson *et al.*, 1999). This technology is a key driver towards the Free Flight regime as a good design and functionalities of CDTI can improve the operational effectiveness of pilots and crew members (RTCA, 2000).

Currently the TCAS system provides a limited traffic display on most modern jetliners. However, due to limited bandwidth, short range and use of the ground based navigation systems to obtain traffic information, it has limited functionality and usually acts as a last line of defence in the case of a conflict situation. Moreover, it is not designed to harness the automation proposed in Free Flight. Proposed avionics systems will derive highly accurate positional data from satellites and will use high bandwidth to send and receive data without any ground support at a larger distance. Therefore, several new issues must be addressed in designing CDTI for Free Flight.

These issues include an answer to each of the following questions: how to manage the clutter due to increased number of aircraft on display resulting from longer radar range? How to display the state and intent information of traffic effectively for efficient decision making? How to integrate the display of a variety of airspace constraints such as weather and SUAs apart from traffic. In Free Flight, the flight crews apart from evaluating and verifying the conflict resolution would also like to oversee and coordinate such resolution with other aircraft or air traffic service providers. CDTI displays must provide information for displaying traffic, conflict alert information and resolution advisory. The objective of this work is to develop a display that will enhance the situation awareness of the flight crew and realize the Free Flight objectives.

4.4.2 CDTI Design

The CDTI design provides for an integrated display of traffic, their intent, conflict alerts, weather alerts and resolution advisories. The design is based on the basic format and size of the *Navigation Display Map Mode* in the Boeing 747-300. The following changes/enhances are made to the display: (1) ownship is represented by letter $\times$, (2) traffic aircraft are represented by a $\star$ (3) range ring selection. The following features are added: (1) the intent information of ownship and traffic are presented in the form of their flight plan (2) The representation of an aircraft using color coded altitude information (not visible in black and white), (3) clutter management, (4) conflict alerts information, (5) convective weather alert information, and (6) resolution advisory for traffic and weather. In the pilot interface, an option is provided for selection of radar range for CDTI display, the pilot can select the coverage area, of which the traffic information is displayed. It also provides display on/off for intent information as well as for limited display of flight information of traffic. The CDTI also displays key autopilot information such as ground speed, true air speed, heading and altitude. It also shows graphically the track angle, original track and current track. Waypoints are displayed in a light green color. Flight

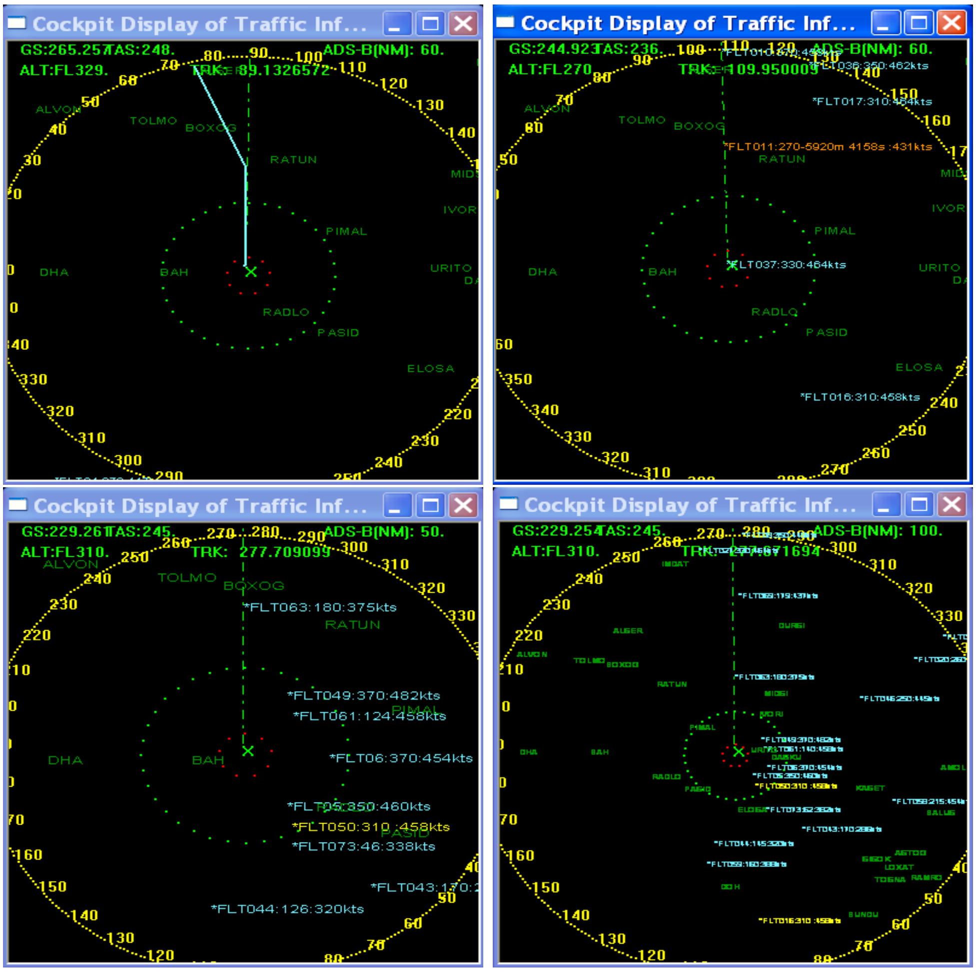


Figure 4.30: Basic view of CDTI interface with ownship at the center and a 20 nm range ring(green) and 5 nm protected zone ring (red). Top Left: Ownship with intent, Top Right: Selective view of neighboring traffic, Bottom Left: Traffic within 50 nm radius, Bottom Right: Traffic within 100 nm radius

call-signs are displayed with a “FLT” pre-fix (Figure 4.30) when simulating artificial air traffic and with real call-signs when simulating field data.

4.4.3 Traffic and Intent Information

The traffic information display is designed to provide a quick “bird’s eye view” of the surrounding traffic. Based on the selected radar range the traffic is displayed with ownship centered in the middle of the CDTI display. Traffic information,which is user selectable, consists of call sign, altitude, speed, and aircraft type. The traffic intent information is displayed as a 3D flight plan, selected by a checkbox on the pilot’s interface. The ownship route is displayed as a light blue line for the next two trajectory change points(Figure 4.30

top left).

4.4.4 Altitude Information

The altitude information which is displayed in the standard 2D map display for ATC as well in aircraft's TCAS consist of textual information which requires focal attention to read and can increase the workload (Alexander and Wickens, 2001). Since altitude information is not graphically rendered, it is difficult to get a common picture of traffic altitude information. Previous attempts of using a 3D map display and coplanar altitude display have their own merits and demerits (Alexander and Wickens, 2001). To address these issues, we have used a color coded scheme to display textual information for relative altitude. Color is used to distinguish the traffic into three groups, first flying at the same altitude, second flying above the ownship altitude and the third flying below. The altitude information is displayed next to the traffic symbol in 100's of feet. Traffic at the same altitude as ownship is color coded yellow, if it is at a higher or lower altitude than the ownship, then it is color coded light blue.

4.4.5 Clutter Management

To manage clutter we used two strategies, first the radar range is user selectable so that pilots can select the area and hence the traffic in that area, which will be displayed in the CDTI map. Selecting a large range will increase the display clutter and a small range will increase the risk of late visual indication of collision alert from neighboring traffic. Secondly, the traffic within 3000 ft below or above of the ownship altitude are displayed in full brightness, other traffic is dimmed. Apart from this, options are provided to switch on/off traffic intent information, traffic tags and flight plans to manage the display clutter.

4.4.6 Conflict Alerts

The conflict alert mechanism is linked to the CD&R algorithm in use. The CD algorithm detects a potential conflict and computes a variety of conflict characteristics including the CPA and time to CPA between ownship and intruder. The information, in a text tag format, is displayed next to the intruder aircraft. Based on the time to CPA, there are two levels of alerts defined: Level 1 alert, when the time to CPA is more than 5 minutes and there is low probability of conflict then the intruder is color coded orange, and Level 2 alert, where the time to CPA is less than 5 minutes and there is a high probability of conflict then the intruder is color coded red and flashing (Figure 4.31). This information is continuously updated as the simulation progresses. The display also shows the flight path trajectories of the two conflicting aircraft. The CDTI interface also provide for range ring display around ownship of 5nm, 20nm, 50nm and 80nm. This, along with color coded altitude information, provides pilots with a better visual understanding of complex conflict scenarios.

4.4.7 Resolution Advisory

The resolution advisory is displayed in textual form on the pilot interface as shown in Figure 4.31, middle. There are three conflict resolution options available by change in heading, by change in altitude and by change in speed. The resolution advisory can be chosen such that only one parameter is used or a combination depending upon the algorithm's capabilities. The resolution advisory is updated continuously as the ownship and intruder gets closer to the conflict position.

4.4.8 Weather Alerts

The CDTI interface displays textual warning of weather cells ahead on the intended trajectory. It also displays the weather cell's heading and speed. When the weather cells are within the CDTI display range then thunderstorm cells are displayed along with their color coded intensity (based on radar reflectivity). The Weather avoidance module does not have a decision support interface on CDTI, therefore when the algorithm generates a set of weather avoidance trajectories, they are filtered based on pre-defined pilot preference for route choice. The final trajectory is displayed as modified flight plan on CDTI, which aircraft fly to avoid thunderstorm cell impact (Figure 4.32) .

4.4.9 Discussion

It is expected that the CDTI display will provide pilots with desired flexibility of route choices in Free Flight. An integrated display for traffic, weather, and SUA will help in effective decision making and optimum route choices while maintaining safety of

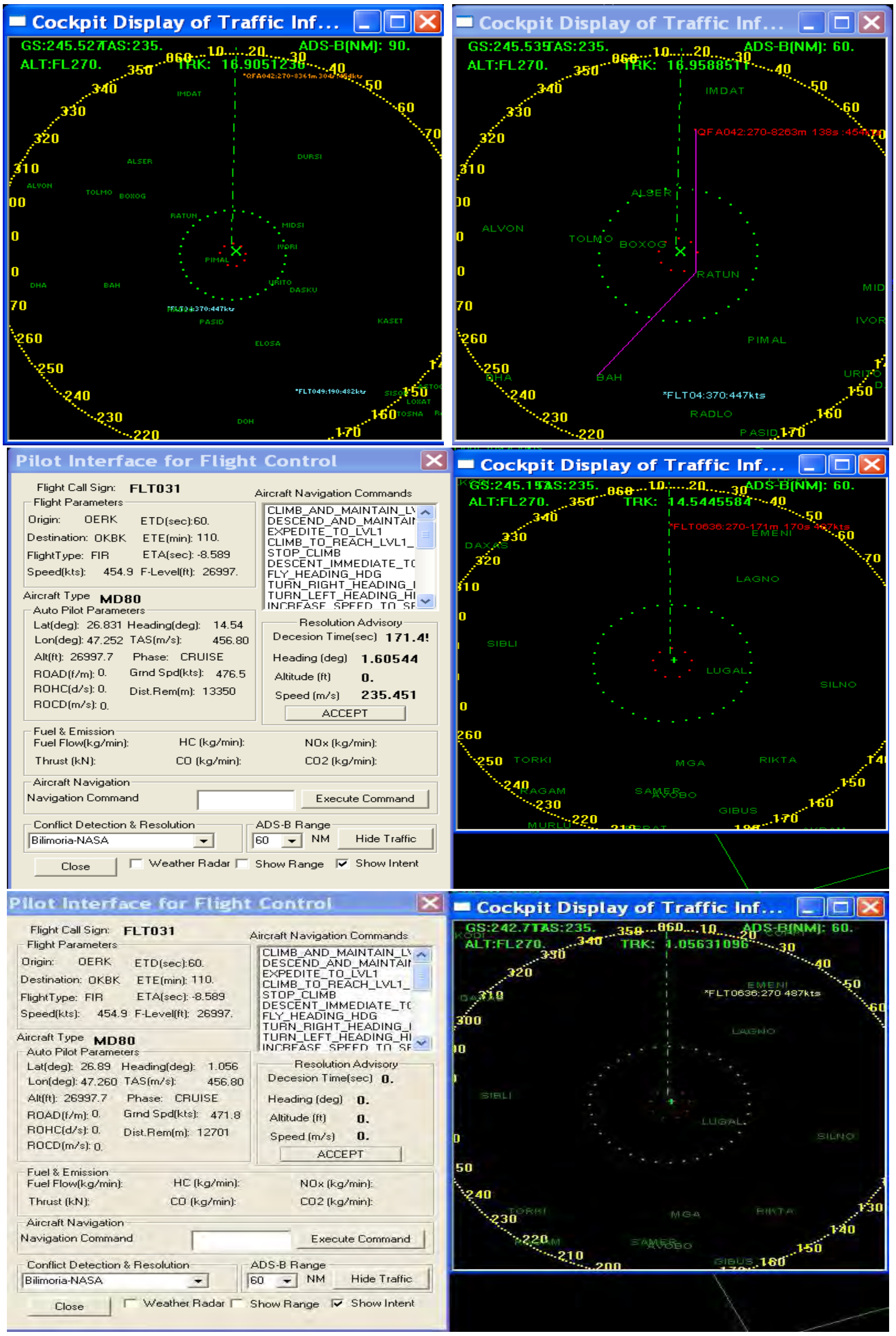


Figure 4.31: Conflict Display Management in CDTI, Top Left: Conflict Alert (intruder in orange), time to CPA > 300 sec. Top Right: Conflict Alert (intruder in red) time to CPA < 300 sec. Middle: CD&R algorithm suggest resolution advisory. Bottom: conflict resolved (no alert)

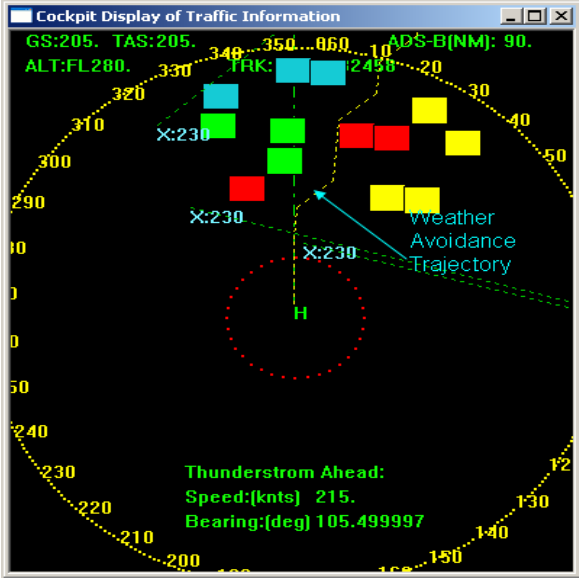


Figure 4.32: Weather alerting and resolution advisory display in CDTI

flights. Pilots acceptability and comfort level when dealing with decision support system will be a key issue in the final outcome of the CDTI design.

4.5 Chapter Summary

Advanced ATM concepts which are under various stages of implementation in ATOMS are presented. Initial experiments are conducted to establish the feasibility of ATOMS in prototyping diverse ATM concepts. A weather avoidance algorithm for traffic constrained enroute airspace is developed and implemented in ATOMS, which besides being inherently safe by design, can provide pilots flexibility of routes choices. Several CD&R algorithms are prototyped and evaluated. A rigorous evaluation of CD&R algorithms is required by exposing them to complex conflict scenario for understating the algorithm's behavior. This can help in identifying anomalies in their design/methodology and help find ways to improve their performance. CDTI can provide for an integrated conflict resolution approach, which is needed to simultaneously account for traffic, weather and other airspace constraints so that complex Free Flight scenarios can be handled.

Chapter 5

Scenario planning for Air Traffic

This chapter is partially based on following publication:

Alam, S., Shafi, K., Abbass, H. A. and Barlow, M. Evolving Air Traffic Scenarios for the Evaluation of Conflict Detection Models, 6th Eurocontrol Innovative Research Workshop and Exhibition, pp. 237-245, Eurocontrol Experimental Center, Paris, France, Dec, 2007

The advanced ATM concepts discussed in the previous chapter need to be better explored for a variety of scenarios. Since field testing them is highly expensive and risky, computer simulations have become a natural choice for their modeling and evaluation. The advanced ATM concepts can be evaluated in a simulation environment by planning hundreds of air traffic scenarios with varying degree of complexities. However, the present methodologies of scenario generation are time and resource intensive which limits their rigorous evaluation.

In this chapter we propose a methodology for generating conflict scenarios algorithmically. The objective is to increase the efficiency of generating air traffic scenarios and to use them in a fast time simulation environment to evaluate advanced ATM concepts.

5.1 Introduction

"A scenario is a documented description of a sequence of events and actors involving one or more sub-components of the ATM system." (Harvey *et al.*, 2003)

Scenarios are core components in evaluating advanced ATM concepts as well as in assessing the impact of any proposed change in existing air traffic procedures. They are of high importance in providing a robust feedback on whether or not an ATM concept can be useful in the operational environment.
They are extensively used in the state-of-the-art cockpit simulators which include: 1) advanced motion-based cockpit simulators such as: a) the Advanced Concepts Flight Simulator (ACFS) at NASA Ames Research Center and b) Aircraft Simulation for Traffic Operations Research (ASTOR) at NASA Langley Research Center (Raytheon, 2004), 2) State-of-the-art ATC simulations which includes: a) NASA's Airspace Operations Laboratory (AOL), b) NASAs Center TRACON Automation System (CTAS) Verification and Validation (V&V) Laboratory, and c) FAAs Target Generation Facility (TGF) (Peters, 1998) and d) Traffic and Experiment Manager (TEM/TMX) ATM simulator at National Aerospace Laboratory (NLR), Netherlands, where they are used in multi-aircraft simulation over a variety of airspaces as well as the setting up of scripted background air traffic.

NASA is currently working on the Virtual Airspace Modeling and Simulation (VAMS) project (Swenson, 2005), which will be used to assess promising technological and conceptual approaches in order to meet these future capacity needs. It will utilize multiple air traffic scenarios for the future airspace in real-time and fast-time simulations as a primary method of evaluating advanced ATM concepts.

5.1.1 Uses of Scenario Planning in ATM

Scenarios can provide a basis for exploratory studies in ATM, concept demonstration, pre-operational testing and evaluation exercises (Signor *et al.*, 2004; Whittle *et al.*, 2003). In the initial concept development level, they help designers to understand the potential of the concept, to understand it better, to define the scope, form basic design features, eliminate poor design choices and refine the concept further; Followed by proof of concepts, performance of concept under non-nominal events and finally advanced design issues. In these concept development stages, scenarios help in evolving a stable and mature design with proof of concepts. Scenarios also help in setting the design limits and procedures for the concept to enable the designers to set up a stable simulation environment for pre-implementation phase as is done in the Programme for Harmonised ATM Research

(PHARE) in Eurocontrol (Gool and Schrter, 1999).

In the final stages of concept development the scenario focuses on pre-operational trials and concept integration in a high fidelity simulation environment. The mature concepts are tested in a rigorous high fidelity simulation environment (Bussink *et al.*, 2005). Large and detailed scenarios are developed to determine the system wide performance assessment. In this stage the scenarios are designed with specific indicator and performance metrics. The scenarios are run repeatedly to gather detailed performance data and often replicated with different scenario configuration parameters.

5.1.2 Scenario Classifications

In ATM, scenarios are broadly classified in two categories: operational concept scenarios and validation scenarios (Harvey *et al.*, 2003). An operational concept scenario mainly focuses upon a specific ATM concept or procedure and is used when a concept is in the initial development phase. It allows for investigating "what if" scenarios for an operational concept in a variety of instances. Such operational scenarios are generic in nature not directed towards any specific detail of the ATM environment (Prandini and Watkins, 2005; Bilimoria, 2000).
A validation scenario is the next logical step after using the operational scenarios. They look at performance issues of operational concepts for a variety of scenario configurations under stress conditions and constraints such as weather, flow management, conflicts etc (Pannequin *et al.*, 2007; Heagy and Kirk, 2003; Prete and Mitchell, 2004; Davidson *et al.*, 2004). Their aim is to test the assumption and design issues in the proposed concept. Therefore validation scenarios, when executed in a simulation environment, can provide robust feedback on whether a proposed concept can be useful in an operational scenario or not.

These scenarios are usually executed in a high fidelity simulation environment (Bilimoria *et al.*, 2001; Peters, 1998; Bussink *et al.*, 2005) and thus requires a detailed configuration file for traffic, airspace and its constraints and scripted details for events to be executed.

5.1.3 The Scenario Planning Process

Following key elements in the scenario planning process are identified by (Harvey *et al.*, 2003)

- Research question and objectives, which drive the scenario design and development process.
- Different aspects of scenarios such as concepts (procedures), environmental (airspace configuration, traffic volume) and events (non-nominal).
- Air traffic samples (if used), which forms the basis of scenario.
- Peaks and troughs in traffic samples, which should be driven by the research objectives and directed towards evaluation metrics.
- The aircraft performance model and aircraft type/mix, which must be validated and identified.

Since covering all the possible interactions in a scenario is impossible. Therefore, each scenario is to be designed to cover those aspects that are relevant to answer particular research questions.

5.1.4 Scenario Execution

Scenarios can be operated in real time and in fast time mode. A real time scenario operation focuses on human-machine interaction and workload issues (Shakarian and Haraldsdottir, 2001; Alexander and Wickens, 2001) where they consist of generic or specific airspace configurations, traffic samples and specifically scripted events such as a conflict. Whereas, fast time scenario operations are used in testing whether a concept works as designed in an operational environment under a variety of constraints (Blom *et al.*, 2003) where they are used to identify the potential gains such as capacity increase and economic benefits associated with new concepts. They are also useful in identifying safety issues and system level properties such as the overall operating cost of a concept. However, a combination of both real time and fast time scenario operation is required for thorough analysis of any concept/procedure.

5.1.5 A typical Scenario Generation and Evaluation Process

A typical scenario generation exercise, when using field data, is highly time consuming due to the lack of standard data processing techniques to handle dispersed air traffic data which have limited standardization (Signor *et al.*, 2004). A typical scenario generation and evaluation process is illustrated in Figure 5.1. As shown in Figure 5.1, the scenario

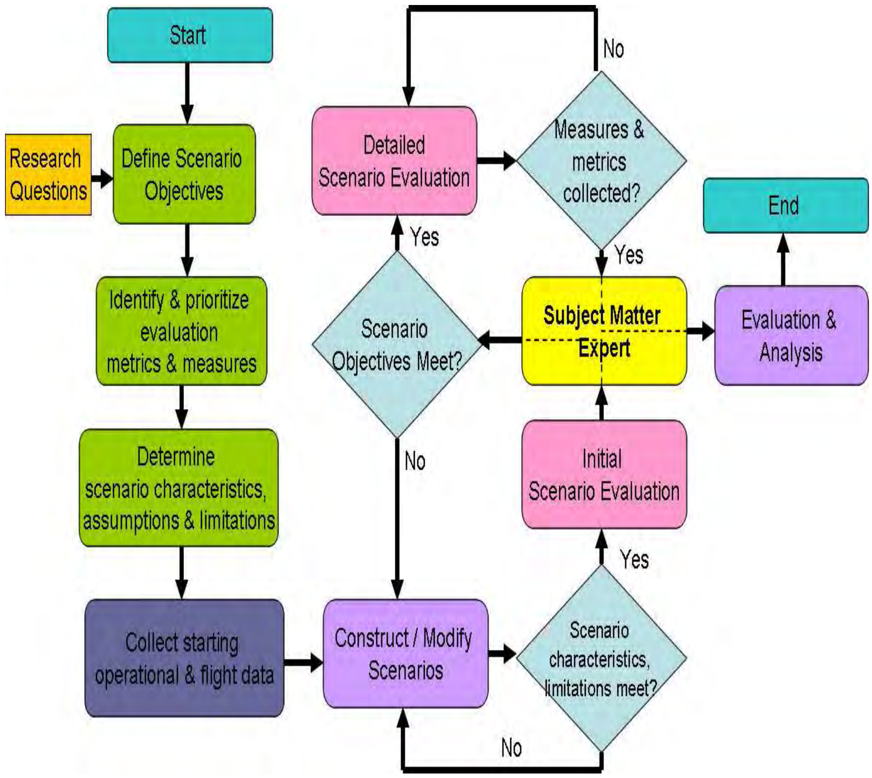


Figure 5.1: A typical scenario generation and evaluation process

generation and evaluation process starts with defining the scenario objectives which are driven by the research questions. While defining the scenario objectives, the concept level of maturity and operational characteristics are taken into consideration. After that, the performance metrics and measures are identified and prioritized in consultation with subject matter experts (SMEs) and stakeholder.

The next step is to determine the assumptions and limitations to map out potential scenario characteristics. The data collection exercise starts after that which is a time consuming exercise due to proprietary nature of air traffic data and its diverse sources which have little standardization. Once the initial data is at hand then sophisticated data processing tools and algorithms are required to modify and construct the scenarios out of them with the desired conflict parameters. It is an iterative and time consuming process. Only after that, a detailed scenario evaluation is conducted with repetition and replication preferably in a fast time simulation environment in order to obtain generalized results on the measures and metrics. SMEs and concept designers then perform detailed analysis of the results.

5.2 Existing Methodologies for Scenario Generation

Existing methodologies on scenario generation focus on pre-scripted scenarios where the designer pre-selects the scenario characteristics (mainly conflict) and then generates the scenarios, which are then fed into the main simulator for execution and analysis (Harvey *et al.*, 2003). In these methodologies, the scenario generation process is separate from the scenario execution process (air traffic simulator) making the whole process complex and time consuming. Moreover, it limits the rigorous evaluation of concepts due to the segregation of two processes. Examples of existing systems are, NASA Ames's *Trajectory-Centered Simulator* (TCSim) (Callantine, 2004) which is a fast-time simulator that contains automated scenario generation capabilities designed specifically for use in conjunction with its *Airspace Operations Lab* (AOL). Another tool called *AvScenario* (Signor *et al.*, 2004) is being developed by Seagull technologies under contract from NASA for its V*irtual Airspace Simulation Technologies Real-Time* (VAST-RT) system which is part of the VAMS project. FAA at the William J. Hughes Technical Center have developed a tool called *AWSIM*, to assist subject matter experts (SME) in the development of scenarios from scratch and from existing data for use in the simulators located at the FAA Technical Center.

In these systems, conflict scenario generation methodologies mainly rely on recorded field data since they preserve real world errors that are likely to affect the performance of a CD algorithm when tested in the field (Paglione *et al.*, 2003). However, this is a challenging task as recorded air traffic data does not contain conflicts and other non-nominal scenarios since any such situations are already resolved by the controllers. The rare recorded air traffic accidents are in the order of 10^{-10} per flight hour (ICAO, 1998) making it almost impossible for the existing fast time simulators to asses such events in recorded air traffic data. Thus the recorded data needs to be modified to induce conflict situations. Conflict situations are achieved in recorded field data either by incorporating pseudo-conflicts where the current separation standards are increased to induce

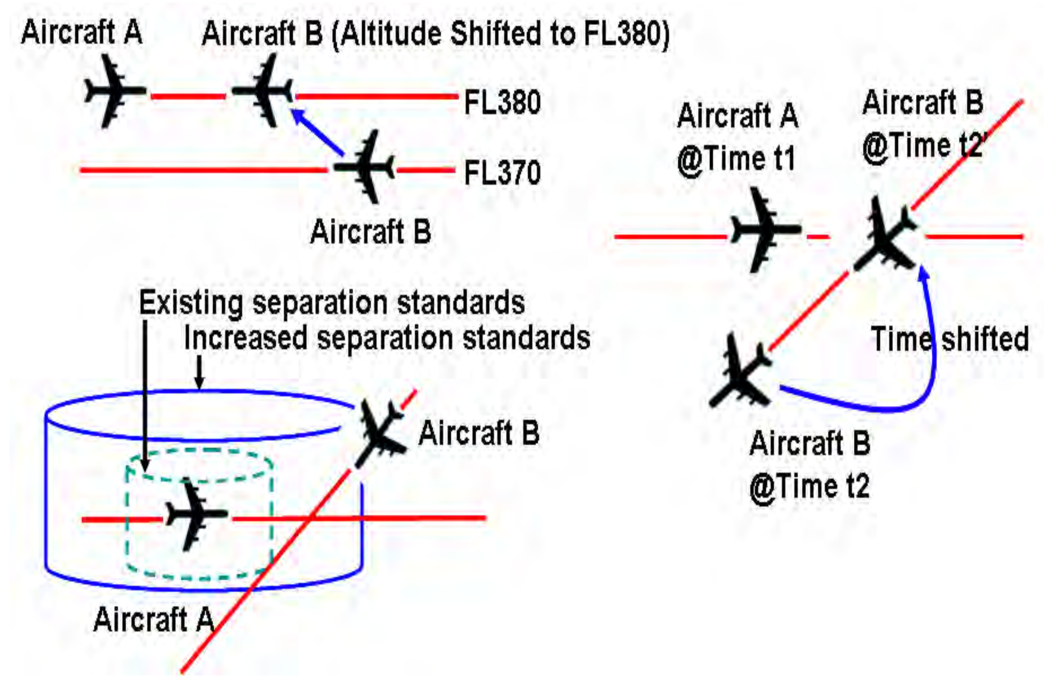


Figure 5.2: Different methodologies of inducting conflict situations in recorded field data. Top Left: Altitude Shift method, Bottom Left: Increase Separation Standards method, Right: Time Shift method

conflicts (Niedringhaus, 1998) (Figure 5.2: bottom left) or by altitude shift where the altitude of an aircraft is manipulated to generate conflict situations (Paielli, 1998) (Figure 5.2: top left). Another methodology, which is developed by the FAA, involves the use of GAs to time-shift flights taken from recorded data to produce the desired conflict parameters (Paglione *et al.*, 2003) (Figure 5.2: right).

Another way is to hand design scenarios which requires extensive subject matter expert input and complex computations to obtain the desired conflict parameters. When field data is not available due to their proprietary nature or when certain complex geometries are difficult to induce in the field data, this is a handy but tedious option. For example, conflict-wall scenario in NLR TMX/TEX simulations (Bussink *et al.*, 2005), where a wall of aircraft is in conflict with another aircraft traveling in opposite direction, and super conflict scenarios, where multiple aircraft are converging towards a common point.

5.2.1 Limitations of Existing Methodologies

Automated scenario generation methodologies have the edge over hand design scenarios in terms of being less tedious and preserving real world errors. On the downside, they can induce conflict situations in recorded field data but the structured nature of traffic flow is still retained in the resulting scenario. In the future Free Flight airspace structure might be completely different (or even absent) rendering these methodologies less effective for future ATM concept evaluation.

Secondly, since an infinite number of conflict geometries are possible due to the decentralization of ATM in the future, field data from linearly designed airspace may fall short in capturing the complexity of future air traffic scenarios. The static nature of scenarios and pre-scripted events makes the evaluation process predictable and less rigorous. These limitations can be overcome by repetition, replication and evolution of scenarios, by generating scenarios of varying degree of complexity and by unfolding events dynamically as a scenario progress in a simulation.

5.3 The Proposed Methodology

In this section, we discuss the proposed methodology for scenario generation with desired conflict parameters. The proposed methodology does not require field data but rather uses conflict parameters at the CPA between two conflicting aircraft to compute their trajectories in the past. This occurs, such that, when the two aircraft are activated at given "birth" points and at a given time, they arrive at a CPA with the pre-specified conflict parameters. For generating scenarios, aircraft's performance parameters are computed from the BADA aerodynamic model under ISA conditions. Zero wind conditions are assumed with no trajectory propagation error to achieve the exact conflict at CPA properties.

The proposed methodology is capable of generating a wide variety of conflict scenarios. The scenario generation process is integrated with ATOMS to facilitate replication and repetition for rigorous evaluation of CD algorithms. Real world errors can be introduced by adding noise in trajectory generation and in the computation of aerodynamic parameters. The broad objective is to generate complex conflict scenarios to evaluate the quantitative performance of various CD algorithms and to understand their behavior in failure circumstances.

5.3.1 Conflict Scenario Parameters

The performance of a CD algorithm depends upon the characteristics of the conflict scenario (Bilimoria and Lee, 2001). A head-on conflict between two cruise level aircraft may be easy to detect as compared to an in-trail conflict between two descending aircraft. Based on (Campos and Marques, 2002), we determined that the following conflict parameters at the CPA are sufficient to generate initial birth points of the two conflicting aircraft. These parameters are:

- Horizontal separation (HS): The horizontal distance between two aircraft at the CPA.
- Vertical separation (VS): The vertical distance between two aircraft at the CPA.
- Conflict geometry Intruder (CGI): The phase of intruder aircraft at the CPA. This can be climb, cruise or descent.
- Conflict geometry Ownship (CGO): The phase of ownship aircraft at the CPA. This can be climb, cruise or descent.
- Conflict angle (CA): The relative conflict angle between the two aircraft at the CPA.

The conflict angle between the two aircraft is computed as follows (Bilimoria, 2000): Given two aircraft A and B, with ψ_a and ψ_b as their heading, V_a and V_b as their speed and v_a and v_b as their velocity vectors, then the conflict angle ψ_r between the two can be computed as:

$$\psi_r = tan^{-1}(N/D) \tag{5.1}$$

where N = $V_a sin(\psi_a)$ - $V_b sin(\psi_b)$ and D = $V_a cos(\psi_a)$ - $V_b cos(\psi_b)$

5.3.2 Assumptions

The following assumptions are made for the scenario generation:

- Zero wind conditions: International standard atmospheric and zero wind conditions are assumed for the experiments.

Table 5.1: Scenario Parameters

Parameter	Value
Start Latitude	S32.0
End Latitude	S38.0
Start Longitude	E142.0
End Longitude	E150.0
Maximum Altitude	38000 ft
Minimum Altitude	10000 ft
Minimum Speed	300 kts
Maximum Speed	550 kts
Activation Time	≥ 1 sec ≤ 600 sec

- No turn Maneuver: Aircraft follow their flight plans which do not consist of any turn maneuver. Since all the algorithms evaluated are state based, having frequent turns would only lead to a large number of FA.
- En-route phase flights only: No TOC or TOD are computed. Flights activate at their birth points in the designated airspace area and deactivate at designated locations.
- No conflict resolutions: Since the objective is to evaluate the CD component of the algorithms, the resolution component of the model is not activated and aircraft are able to fly without resolving conflicts.

The east coast of Australian flight information region is used as research airspace in the simulation. The bounds of various scenario parameters are set as shown in Table 5.1: The following set of aircraft is used while generating flight plans for the conflicting pair of aircraft: [A300 A310 A320 A330 A340 B707 B727 B737 B747 B757 B767 B777]. Ownship and intruder aircraft are chosen from this set randomly using uniform distribution. Figure 5.3 shows the outline of the scenario generation process.

5.3.3 Conflict Scenario Generation Mechanism

This section proposes a mechanism (conflict generator) to determine the initial activation point of the two conflicting aircraft given the conflict parameters at their CPA. The outline of the mechanism is as follows:

1. The ownship position and heading is randomly generated within the bounds of the research airspace. This becomes the ownship position at the CPA.

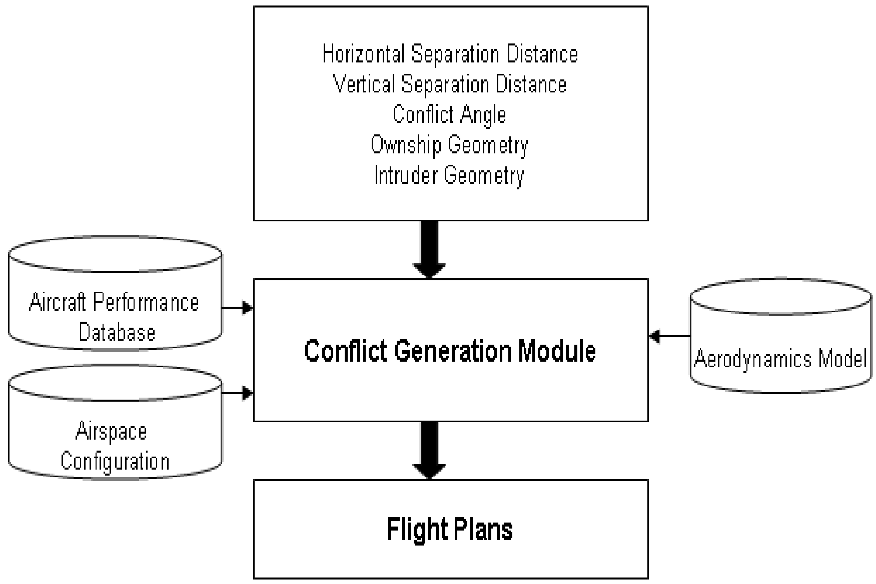


Figure 5.3: Outline of flight plan generation from chromosomes

2. Based on conflict angle between ownship and intruder, the relative heading between the two is worked out.

3. Using the relative heading, ownhsip position data and horizontal separation at CPA the intruder position at CPA is computed.

4. Using the vertical separation distance, the altitude at the CPA is computed for the intruder.

5. Based on the ROCD of the aircraft the activation and deactivation altitude are computed such that the aircraft is in a desired geometry when it crosses the other aircraft at the CPA.

Figure 5.4 shows the position of the intruder and the ownship in conflict positioned at the CPA cylinder.

Following notations are used in the conflict generation process:

A: Research Airspace defined by Start Latitude, Start Longitude, End Latitude, End Longitude

AC: Set of aircraft used in simulation

O_AC: Ownship Aircraft

I_AC: Intruder Aircraft

OD_{lat}: Latitude of Ownship

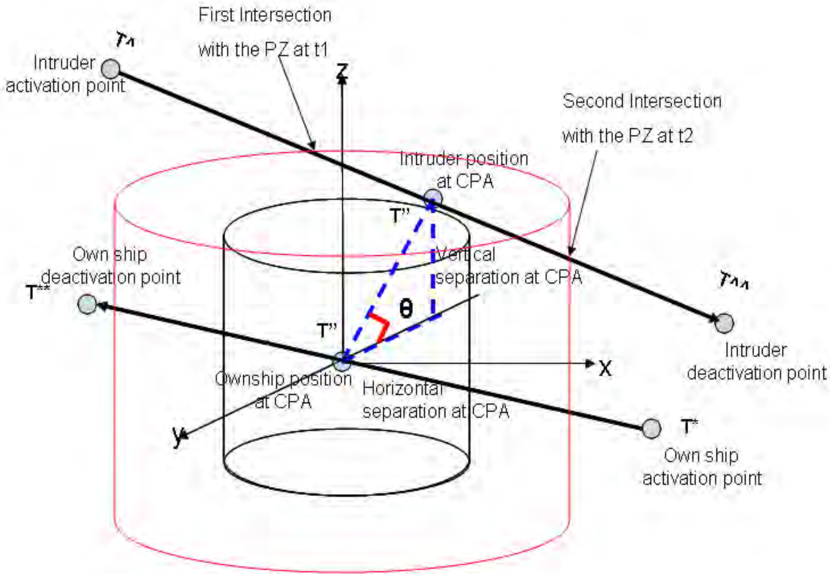


Figure 5.4: Intruder and ownship at the CPA cylinder

OD_{lon}: Longitude of Ownship

OC_{lat}:Latitude of ownship at the CPA

OC_{lon}:Longitude of ownship at the CPA

OC_{alt}:Altitude of ownship at the CPA

OC_{hdg}:Heading of ownship at the CPA

O_{speed}:Altitude of the ownship at the CPA

OA_{lat}:Latitude of ownship at the activation point

OA_{lon}:Longitude of ownship at the the activation point

OA_{alt}:Altitude of ownship at the the activation point

OA_{hdg}:Heading of ownship at the the activation point

OA_{speed}:Altitude of the ownship the activation point

IC_{lat}:Latitude of intruder at the CPA

IC_{lon}:Longitude of intruder at the CPA

IC_{alt}:Altitude of intruder at the CPA

IC_{hdg}:Heading of intruder at the CPA

I_{speed}:Altitude of the intruder at the loss of separation

$T2CPA_i$:Time to closest point approach for intruder

$T2CPA_o$:Time to closest point approach for ownship

I_{AT}: Intruder activation time

O_{AT}: Ownship activation time

SIM_S: Simulation Start time

SIM_E: Simulation End time

O_{speed}:Ownship Air Speed

I_{speed}:Intruder Air Speed

CA: Conflict angle between two aircraft at the CPA

HS: Horizontal separation distance at the CPA

VS: Vertical separation at the CPA

Now we detail the steps for generating initial conditions for two aircraft having a loss of separation at CPA characterized by HS, VS, CA, CG-I, CG-O.

1. Randomly assign a conflict position to the Ownship:

   (a) Randomly select own ship aircraft from the set of available aircraft
   $O_AC \leftarrow rand[AC]$

   (b) Randomly assign a conflict position within the airspace bound to the aircraft
   $OC_{lat} \leftarrow rand[A_{startlat} - A_{endlat}]$
   $OC_{lon} \leftarrow rand[A_{startlon} - A_{endlon}]$
   $OC_{alt} \leftarrow rand[15000ft - 33000ft]$

   (c) Randomly assign a heading for the ownship aircraft
   $OC_{hdg} \leftarrow rand[0.0 - 2 \times PI]$

   (d) For the given conflict altitude obtain the true air speed from the aircraft performance database
   $O_{speed} \leftarrow Speed(OC_{alt})$

2. Compute the relative angle of conflict between the two aircraft:
   $if((OC_{hdg} + CA) > 2.0 \times PI)$ then

$$RA = ((OC_{hdg} + CA) - 2.0 \times PI) \tag{5.2}$$

else

$$RA = (OC_{hdg} + CA) \tag{5.3}$$

3. Compute the conflict position of the Intruder:

   (a) Based on the relative heading compute the heading of intruder

   $$IC_{hdg} = RA + PI \tag{5.4}$$

   (b) check if $IC_{hdg} > 2.0 \times PI$ then

   $$IC_{hdg} = IC_{hdg} - PI \tag{5.5}$$

   (c) Using ownship state data and horizontal separation find the position of the intruder aircraft at the CPA.

   $$IC_{lat} = asin(sin(OC_{lat}) \times cos(HS) + cos(OC_{lat}) \times sin(d) \times cos(RA)) \tag{5.6}$$

   $$IC_{lon} = mod(OC_{lon} - asin(sin(RA) \times sin(HS)/cos(IC_{lat})) + PI, 2 \times PI) - PI \tag{5.7}$$

   (d) Assign altitude to the intruder by adding the vertical separation to the ownship altitude

   $$IC_{alt} = OC_{alt} + VS \tag{5.8}$$

   (e) Randomly assign the aircraft type to the intruder
   $I_AC \leftarrow rand[AC]$

   (f) Get the aircraft speed given the altitude
   $I_{speed} \leftarrow Speed(IC_{alt})$

The above mentioned steps define the position of the two aircraft at the time of conflict. Now, we have to backout their trajectory in distance and time to define their start position.

4. Compute ownship data at the activation point:

(a) Compute ownship heading at the activation point

$$OA_{hdg} = OC_Hdg + PI \tag{5.9}$$

check if $OA_{hdg} => 2.0 \times PI$ then

$$OA_{hdg} = OA_{hdg} - PI \tag{5.10}$$

(b) For Ownship randomly select a distance between range 80 nm to 120 nm and assign this as distance from activation point and point of closest approach $O_{cd} \leftarrow random(120.0 - 80.0)$

(c) Find the ownship latitude and longitude at the activation point

$$OA_{lat} = asin(sin(OC_{lat}) \times cos(O_{cd}) + cos(OC_{lat}) \times sin(O_{cd}) \times cos(OA_{hdg})) \tag{5.11}$$

$$OA_{lon} = mod(OC_{lon} - asin(sin(OA_{hdg}) \times sin(O_{cd})/cos(OA_{lat})) + PI, 2 \times PI) - PI \tag{5.12}$$

(d) Based on the conflict geometry of the ownship assign the altitude
If Climb then $OA_{alt} = OC_{alt} - AltFactor(ft)$
If Descend then $OA_{alt} = OC_{alt} + AltFactor(ft)$
If Cruise then $OA_{alt} = OC_{alt}$
where AltFactor is computed based on ROCD of the aircraft.

(e) Compute the time for ownship to cover the distance O_{cd}

$$O_{t_cd} = O_{cd}/OA_{speed} \tag{5.13}$$

5. Compute the deactivation point of the ownship:
For Ownship select a distance (80nm) from point of closest approach to the deactivation point.
Based on the heading of the ownship at the point of closest approach find the coordinates of the deactivation points (latitude and longitude) using equations 5.11 and 5.12.

6. Repeat the above step to compute the activation and deactivation point for intruder and time to travel from activation point to the point of loss of separation.

7. Randomly assign an activation time for the intruder

$$I_{AT} \leftarrow random(SIM_S, SIM_E) \tag{5.14}$$

8. Compute the activation time for ownship and intruder
   If $T2CPA_i > T2CPA_o$ then
   $O_{AT} = I_{AT} + (T2CPA_i - T2CPA_o)$
   $I_{AT} = I_{AT}$
   Else If $T2CPA_o > T2CPA_i$ then
   $I_{AT} = I_{AT} + (T2CPA_o - T2CPA_i)$
   $O_{AT} = I_{AT}$

9. Form the flight plan.

If a combination of conflict parameters is generated such that conflict is not possible, then that combination is discarded. If the conflict point is generated outside the research airspace then the conflict position is regenerated. Flight plans generated by the conflict generator consists of a unique call sign, prefixed with "FFF" and then a numeric code which is generated following a sequential order. The number of flights is always kept even so that there can be a pairwise conflict between two flights.

The activation coordinates, conflict location coordinates, and deactivation coordinates are specified in latitude (in degrees south) and longitude (in degrees est). Aircraft type denotes the type of aircraft used; activation flight level is the altitude where the flight is activated; target flight level is the altitude which the flight climb, descent or maintains depending upon the conflict geometry; speed is the initial speed at the activation flight level; and activation time is when the aircraft is activated since the start of the simulation.

5.4 Generated Conflict scenarios

In this section we present validations carried out for conflict scenario generation. Validations are carried out for conflict angle, horizontal separation, vertical separation and conflict geometry. Different conflict geometry combinations for ownship and intruder are used and other conflict parameters are generated at random within the given bounds. The flight plan for the conflicting flights are shown along with the conflict parameters.

Each flight pair is simulated in ATOMS and their trajectories are recorded. The activation, conflict and deactivation points are shown in latitude (degree South) and longitude (degree East). The speed is in knots and is the activation speed at the activation flight level for the given aircraft type derived from BADA performance tables. Activation and target altitude are given in '00 feet and activation time is in minutes and seconds. Figure 5.5 to Figure 5.13 shows the conflict parameters and the resulting flight plans of the two flights. It also shows the flight trajectories resulting from the flight plan execution. At the CPA, the horizontal, vertical separation and conflict angle between two conflicting flight is shown. The following combinations for conflict geometry are presented:

1. CLIMB-CLIMB: At CPA, ownship is climbing and intruder is also climbing.
2. CLIMB-CRUISE: At CPA, ownship is climbing and intruder is cruising.
3. CLIMB-DESCENT: At CPA, ownship is climbing and intruder is descending.
4. DESCENT-CLIMB: At CPA, ownship is descending and intruder is climbing.
5. DESCENT-CRUISE: At CPA, ownship is descending and intruder is cruising
6. DESCENT-DESCENT: At CPA, ownship is descending and intruder is also descending.
7. CRUISE-CLIMB: At CPA, ownship is cruising and intruder is climbing.
8. CRUISE-CRUISE: At CPA, ownship is cruising and intruder is also cruising.
9. CRUISE-DESCENT: At CPA, ownship is cruising and intruder is descending.

5.5 Distribution of Conflict Parameters

To see the distribution of conflict parameters in a scenario, a random sample of 100 conflict pairs comprising 200 flights is generated. Plots of frequency distribution for each conflict parameter is generated as shown:

- Distribution of flight density over simulation time: Figure 5.14 shows how many flights are active in the research airspace as the simulation is progressing over time.

Horz. Sep	Vert. Sep	Conflict Angle	Ownhip Geo	Intruder Geo
0.14 nm	183 ft	152.77 deg	CLIMB	CLIMB

CallSign	Activation Point	Conflict Point	Deactivation Point	Aircraft Type	Speed (kts)	Activation Alt (000ft)	Deactivation Alt (000ft)	Activate Time(mm.ss)
FFF39	S36.48E147	S36.19E147.38	S34.67E146.33	B727	480	239	309	3.00
FFF40	S35.81E147	S36.12E147.34	S37.63E148.21	B747	498	244	314	3.05

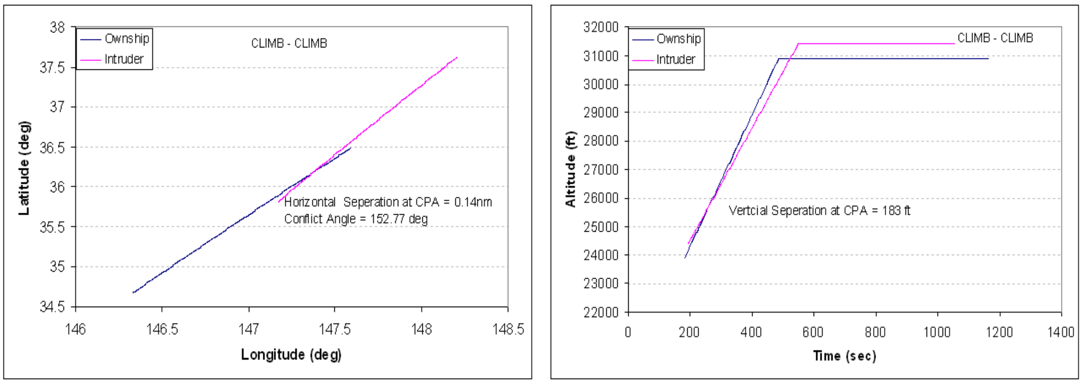


Figure 5.5: A CLIMB-CLIMB conflict scenario. Top: Conflict characteristic, next from the top, flight plans of the conflicting flight pair, bottom left: planar view of the conflict in simulation, bottom right: vertical view of the conflict in simulation

Horz. Sep	Vert. Sep	Conflict Angle	Ownship	Intruder
3.87 nm	290 ft	70.02 deg	CLIMB	DESCENT

CallSign	Activation Point	Conflict Point	Deactivation Point	Aircraft Type	Speed (kts)	Activation Alt (00 ft)	Target Alt (00 ft)	Activation Time(mm:ss)
FFF61	S33.82E142.81	S33.50E142.88	S31.93E143.32	A330	447	145	215	16:17
FFF62	S33.83E143.19	S33.56E142.95	S32.24E141.73	B757	404	276	206	16:00

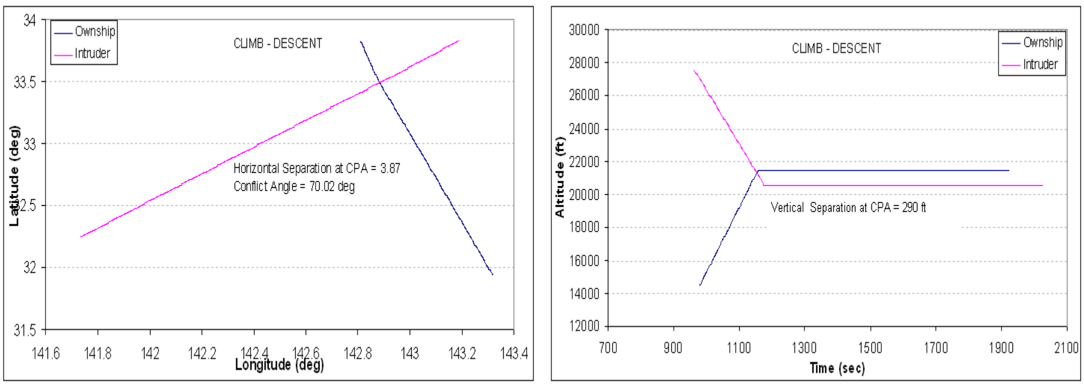


Figure 5.6: A CLIMB-DESCENT conflict scenario. Top: Conflict characteristic, next from the top, flight plans of the conflicting flight pair, bottom left: planar view of the conflict in simulation, bottom right: vertical view of the conflict in simulation

Horz. Sep	Vert. Sep	Conflict Angle	Ownship	Intruder
2.57 nm	925 ft	27.43 deg	CLIMB	CRUISE

CallSign	Activation Point	Conflict Point	Deactivation Point	Aircraft Type	Speed (kts)	Activation Alt (00 ft)	Target Alt (00 ft)	Activation Time(mm:ss)
FFF49	S33.62E144.79	S33.57E145.18	S33.22E147.11	A320	463	207	277	3:01
FFF50	S33.19E144.99	S33.50E145.15	S35.04E145.93	A310	458	339	269	3:00

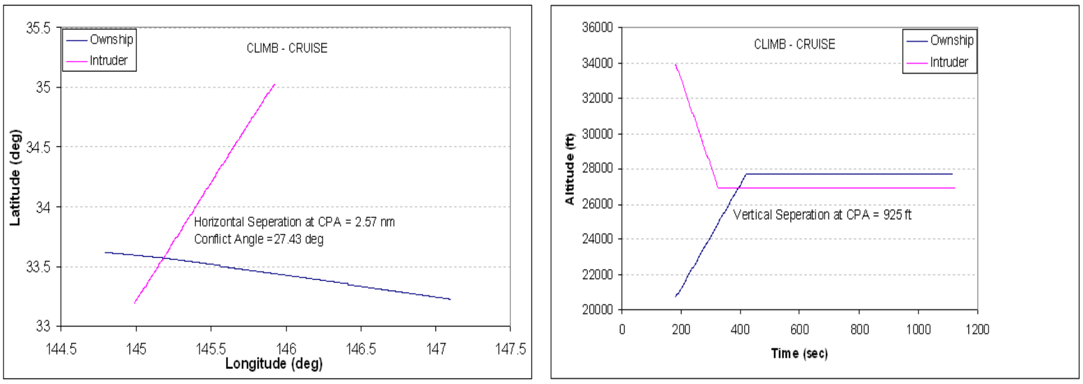


Figure 5.7: A CLIMB-CRUISE conflict scenario. Top: Conflict characteristic, next from the top, flight plans of the conflicting flight pair, bottom left: planar view of the conflict in simulation, bottom right: vertical view of the conflict in simulation

Horz. Sep	Vert. Sep	Conflict Angle	Ownship	Intruder
3.13 nm	400.26 ft	31.01 deg	DESCENT	CLIMB

CallSign	Activation Point	Conflict Point	Deactivation Point	Aircraft Type	Speed (kts)	Activation Alt (00 ft)	Target Alt (00 ft)	Activation Time(mm:ss)
FFF3	S32.73E147.00	S33.00E147.25	S34.29E148.44	B727	492	306	256	13:07
FFF4	S32.68E146.75	S32.98E147.21	S33.99E148.80	A330	479	213	263	12:00

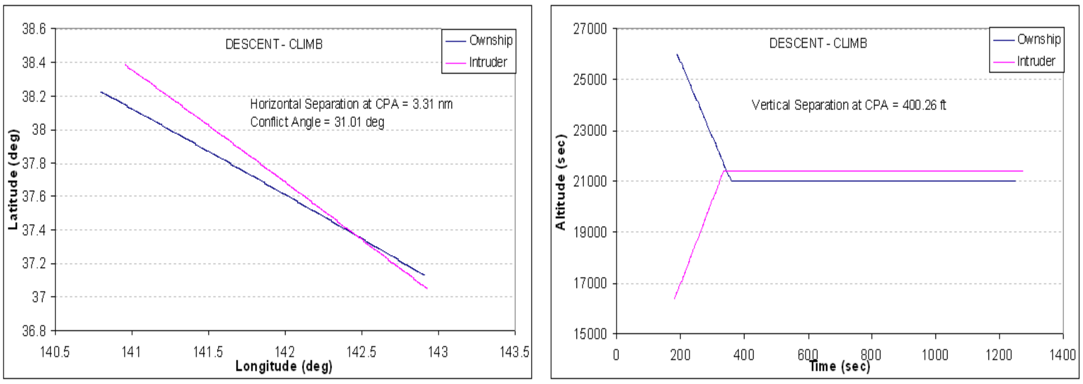
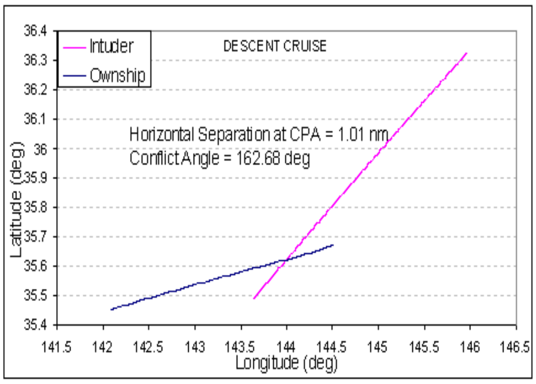


Figure 5.8: A DESCENT-CLIMB conflict scenario. Top: Conflict characteristic, next from the top, flight plans of the conflicting flight pair, bottom left: planar view of the conflict in simulation, bottom right: vertical view of the conflict in simulation

Horz. Sep	Vert. Sep	Conflict Angle	Ownship	Intruder
1.01 nm	498.68 ft	162.68 deg	DESCENT	CRUISE

CallSign	Activation Point	Conflict Point	Deactivation Point	Aircraft Type	Speed (kts)	Activation Alt (00 ft)	Target Alt (00 ft)	Activation Time(mm:ss)
FFF77	S35.49E143.64	S35.63E144.02	S36.33E145.98	A300	378	92	162	1:00
FFF78	S35.67E144.51	S35.63E144.10	S35.45E142.07	B727	428	229	159	1:22



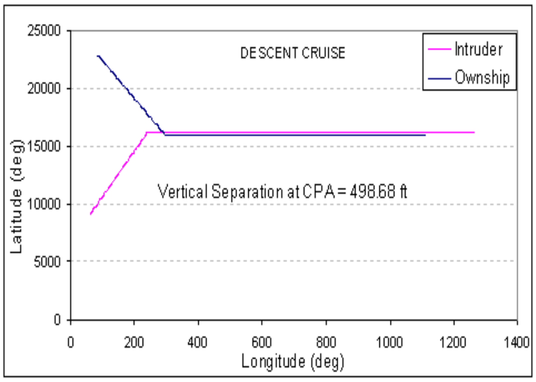
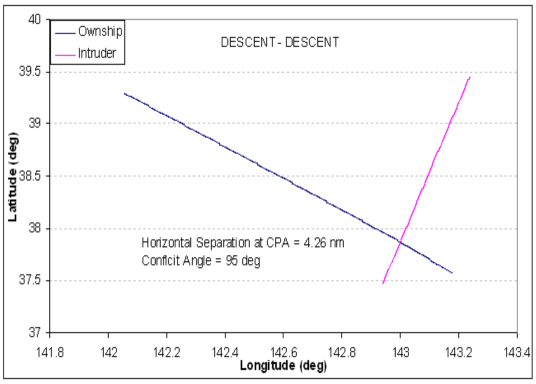


Figure 5.9: A DESCENT-CRUISE conflict scenario. Top: Conflict characteristic, next from the top, flight plans of the conflicting flight pair, bottom left: planar view of the conflict in simulation, bottom right: vertical view of the conflict in simulation

Horz. Sep	Vert. Sep	Conflict Angle	Ownship	Intruder
4.26 nm	290.0 ft	95.0 deg	DESCENT	DESCENT

CallSign	Activation Point	Conflict Point	Deactivation Point	Aircraft Type	Speed (kts)	Activation Alt (00 ft)	Target Alt (00 ft)	Activation Time(mm:ss)
FFF35	S37.57E143.18	S37.87E143.00	S39.30E142.05	B777	446	321	251	4:04
FFF36	S37.47E142.94	S37.80E142.99	S39.46E143.24	B757	434	324	254	4:00



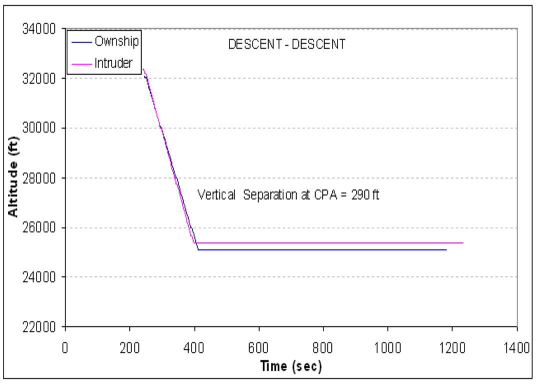


Figure 5.10: A DESCENT-DESCENT conflict scenario. Top: Conflict characteristic, next from the top, flight plans of the conflicting flight pair, bottom left: planar view of the conflict in simulation, bottom right: vertical view of the conflict in simulation

Horz. Sep	Vert. Sep	Conflict Angle	Ownship	Intruder
1.95 nm	751.3 ft	86.12 deg	CRUISE	CLIMB

CallSign	Activation Point	Conflict Point	Deactivation Point	Aircraft Type	Speed (kts)	Activation Alt (00 ft)	Target Alt (00 ft)	Activation Time(mm:ss)
FFF167	S33.44E141.91	S33.73E142.11	S35.12E143.19	B777	393	97	167	2:00
FFF168	S33.44E142.46	S33.68E142.17	S34.84E140.73	B747	426	161	161	2:14

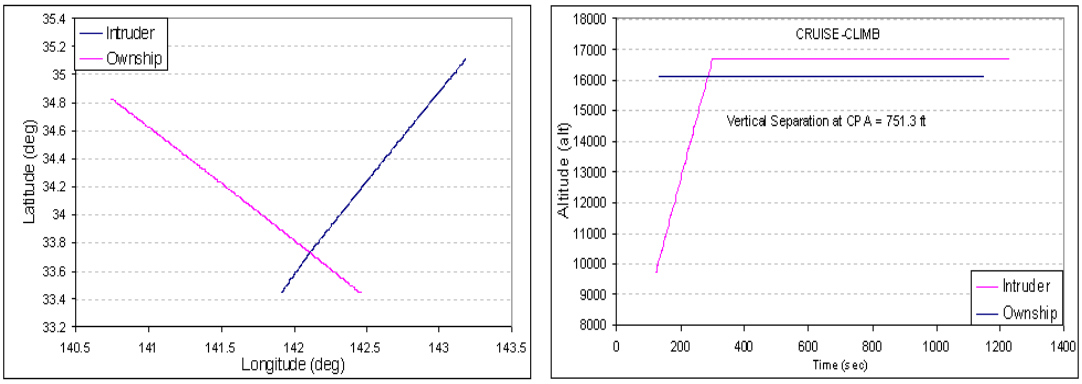


Figure 5.11: A CRUISE-CLIMB conflict scenario. Top: Conflict characteristic, next from the top, flight plans of the conflicting flight pair, bottom left: planar view of the conflict in simulation, bottom right: vertical view of the conflict in simulation

Horz. Sep	Vert. Sep	Conflict Angle	Ownship	Intruder
4.12 nm	600.39 ft	104.13 deg	CRUISE	CRUISE

CallSign	Activation Point	Conflict Point	Deactivation Point	Aircraft Type	Speed (kts)	Activation Alt (00 ft)	Target Alt (00 ft)	Activation Time(mm:ss)
FFF19	S37.37E148.18	S37.30E147.70	S36.97E145.63	A340	415	226	226	1:48
FFF20	S37.56E147.16	S37.30E147.68	S36.40E149.44	B757	420	232	232	1:00

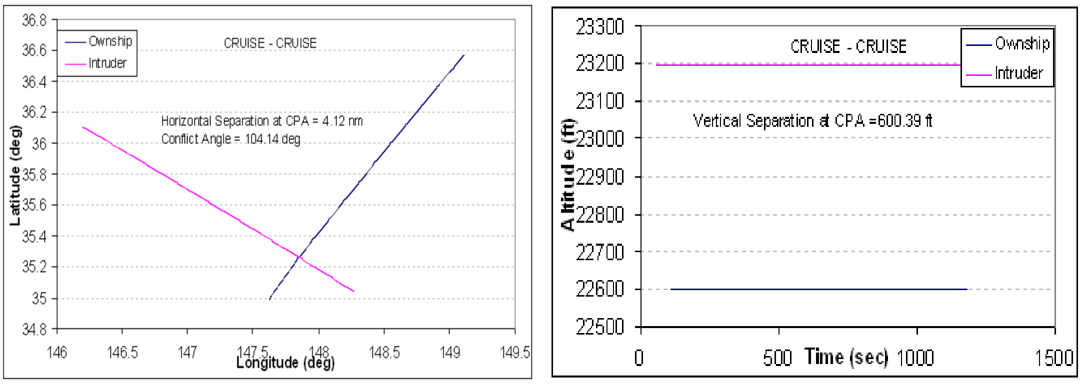


Figure 5.12: A CRUISE-CRUISE conflict scenario. Top: Conflict characteristic, next from the top, flight plans of the conflicting flight pair, bottom left: planar view of the conflict in simulation, bottom right: vertical view of the conflict in simulation

Horz. Sep	Vert. Sep	Conflict Angle	Ownship	Intruder
2.57 nm	78.0 ft	96.19 deg	CRUISE	DESCENT

CallSign	Activation Point	Conflict Point	Deactivation Point	Aircraft Type	Speed (kts)	Activation Alt (00 ft)	Target Alt (00 ft)	Activation Time(mm:ss)
FFF119	S37.67E148.77	S37.95E148.55	S39.31E147.38	A330	402	88	158	1:29
FFF120	S37.61E148.27	S37.89E148.50	S39.29E149.67	B757	344	221	151	1:00

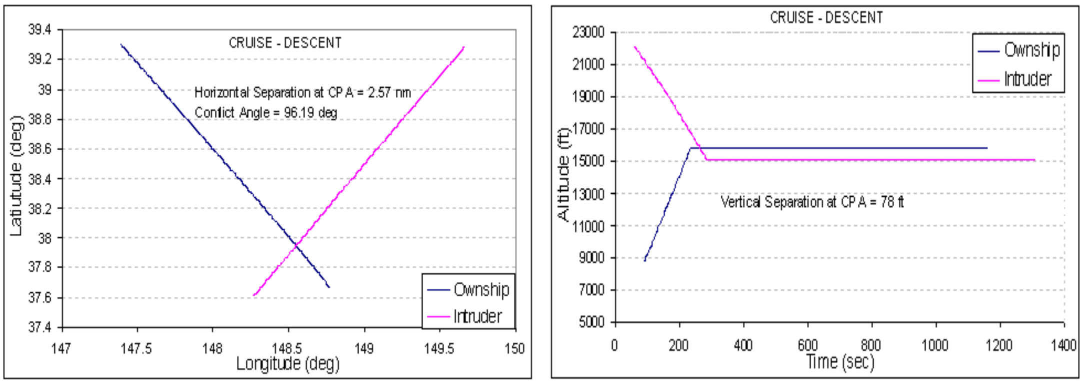


Figure 5.13: A CRUISE-DESCENT conflict scenario. Top: Conflict characteristic, next from the top, flight plans of the conflicting flight pair, bottom left: planar view of the conflict in simulation, bottom right: vertical view of the conflict in simulation

Traffic slowly builds up and reaches its peak at about 950 seconds in simulator, and then tapers off.

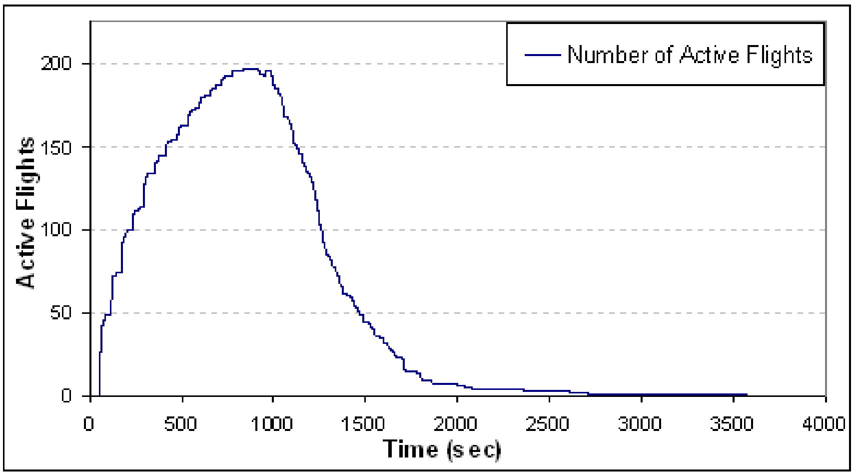


Figure 5.14: Distribution of flight density over simulation time

- Distribution of conflict's locations over the research airspace: Figure 5.15 shows how the conflicts are distributed in the research airspace. Conflict locations largely cover the airspace and are well distributed.

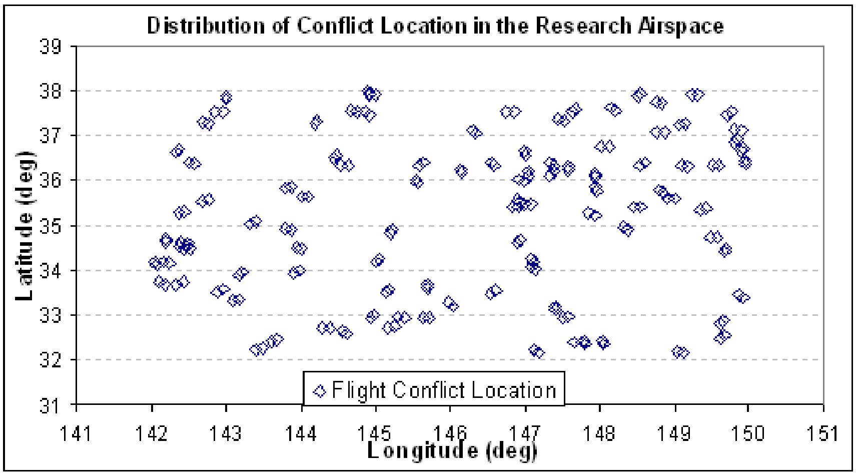


Figure 5.15: Distribution of conflicts locations over the research airspace

- Frequency distribution of flight conflict geometry: Figure 5.16 shows the flight phase (climb, cruise, descent) at their CPA. Cruise-Cruise conflicts are maximum, covering more than 40% of the scenario geometries generated, which is acceptable as in Free Flight the aircraft will spend less time in transition and more time in cruise (Bilimoria and Lee, 2001).

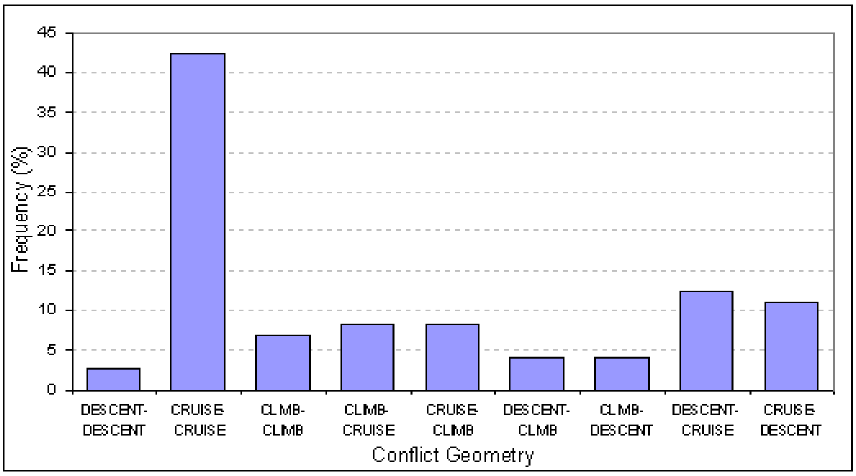


Figure 5.16: Distribution of conflicts locations over the research airspace

- Frequency distribution of horizontal separation (nm) at CPA: Figure 5.17 shows the distribution of the horizontal separation between two conflicting flights at their CPA.

- Frequency distribution of vertical separation (ft) at CPA: Figure 5.18 shows the distribution of the vertical separation between two conflicting flights at their CPA.

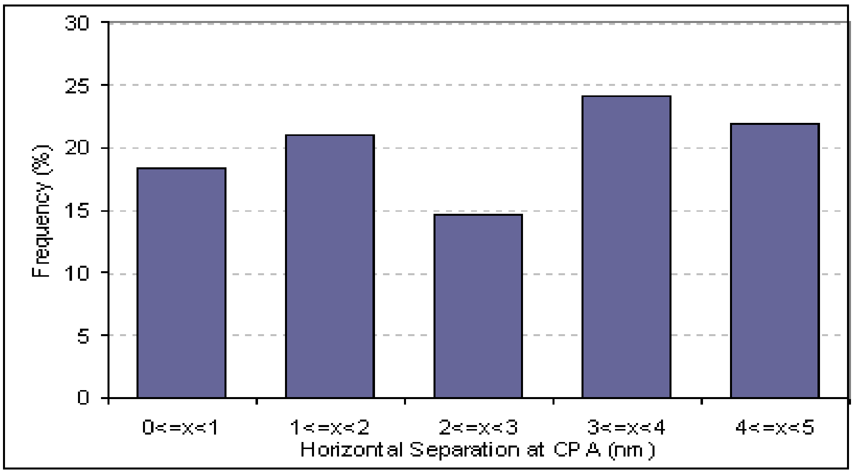


Figure 5.17: Frequency distribution of horizontal separation (nm) at CPA

The band of 800ft to 1000ft has the majority of conflicts.

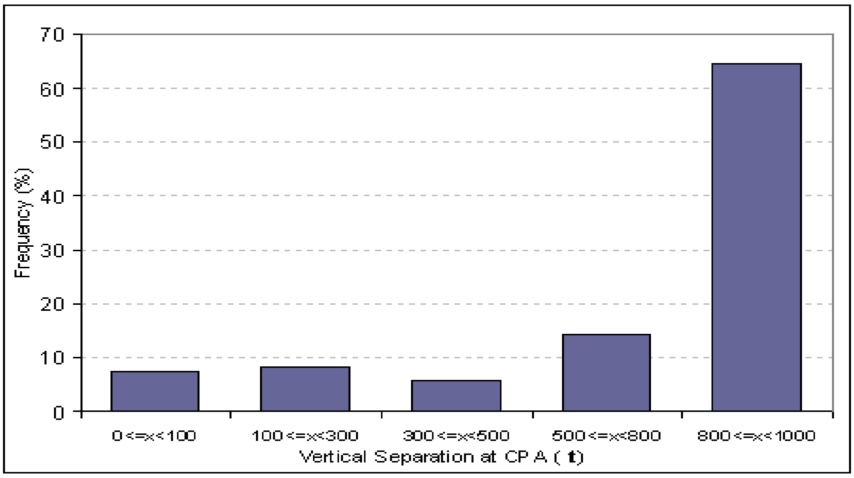


Figure 5.18: Frequency distribution of vertical separation (nm) at CPA

- Frequency distribution of conflict angle (deg) at CPA : Figure 5.18 shows the distribution of the conflict angle between two conflicting flights at their CPA.

5.6 Chapter Summary

A conflict scenario generation methodology is developed to meet the need of rapid encounter scenario simulation in a highly controlled experimental environment. The proposed methodology allows the definition of the planned horizontal and vertical separation at CPA, conflict angle at CPA, and the flight geometry at the CPA in a conflict scenario.

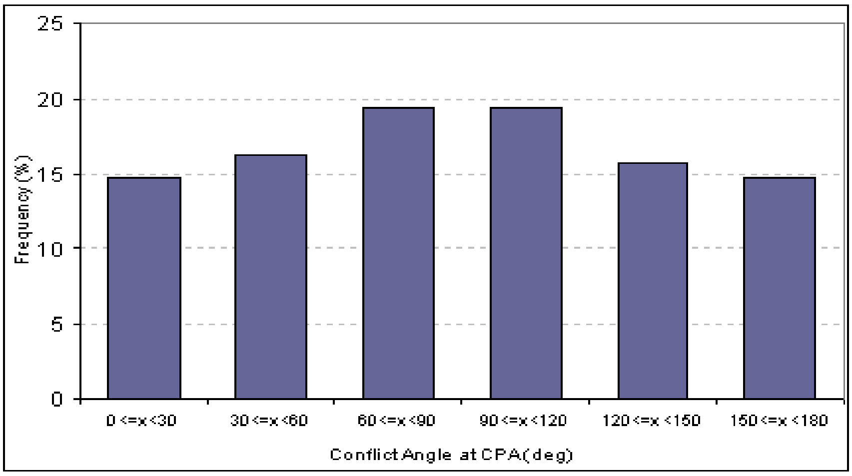


Figure 5.19: Frequency distribution of conflict angle (deg) at CPA

This provides the means for rapidly evaluating CD algorithm's performance across a wide variety of conflict geometries, crossing angles, and other experimental conditions. The proposed methodology also supports the reconstruction of encounter scenarios by identifying conflict parameters at CPA. The conflict scenario generation methodology does not account for wind errors and flight technical errors which can occur in real world air traffic conflict situations.

Chapter 6

Evolving Scenarios towards Complexity

This chapter is partially based on following publications:

1. Alam, S., Shafi, K., Abbass, H. A. and Barlow, M. Evolving Air Traffic Scenarios for the Evaluation of Conflict Detection Models, 6th Eurocontrol Innovative Research Workshop and Exhibition, pp. 237-245, Eurocontrol Experimental Center, Paris, France, Dec, 2007

2. Alam, S., Shafi, K., Abbass, H. A. Forming an Intelligent Ensemble of Conflict Detection Algorithms in Free Flight by Data Mining the Scenario Space, submitted to the Journal of Transportation Research Part-C: Emerging Technologies, Elsevier Science, Conditionally accepted : 29/02/2008

Air traffic scenarios are vital in the evaluation of any present and future ATM concepts. However, existing scenario generation methodologies have several limitations which render them ineffective to evaluate advanced ATM concepts. Moreover, the scenario generation processes are static in nature such that the designer knows in advance the nature of conflict events in the scenario. Ideally, the events should unfold dynamically as the scenario progresses. The conflict events should evolve to the next level, becoming more complex and challenge the CD algorithms to their limits. If we see a scenario as a problem landscape then this can be seen as evolving the problem landscape to find the problem subspace where an algorithm is vulnerable.

Since generating all the possible conflict scenarios is not possible due to the infinite number of possible conflict geometries in Free Flight, finding the vulnerable areas in problem space, where an algorithm may fail, is critical for algorithm evaluation as well as for performance improvement.

In this chapter, we break away from the classical approach of pre-scripting conflict events in air traffic scenarios and use Evolutionary Computation (EC) to evolve conflicts instead. Our approach is to play "Devil's Advocate" where complex conflict scenarios are deemed "fitter individuals" having increased likelihood of survival to the next generation in the evolutionary process of a genetic algorithm. The objective of the proposed methodology is to evolve increasingly complex conflict scenarios so that the CD algorithms can incur maximum failure (in terms of evaluation metrics).

6.1 Evolutionary Computation

The basic idea behind evolutionary computation techniques is to evolve a population of candidate solutions to a given problem, using naturally inspired genetic variations and natural selection (Holland, 1975).

Many complex ATM problem require searching through a large number of possible solutions. Such problems can be effectively dealt with by an intelligent strategy for choosing the direction of search. Further, these problems also require an algorithm to adaptively perform well in a continuously changing environments (for example weather avoidance trajectories). Some problems require algorithms to be innovative so as to construct something new out of the available information (for example landing sequence for aircraft). Some problems require complex solutions that are difficult to program by hand (for example air traffic scenarios).

The evolutionary process of natural selection provides a simple way of addressing these problems where evolutionary process is searching among a large number of possibilities.

6.1.1 Genetic Algorithms

GAs (Goldberg, 1989) are population-based, evolutionary computation techniques proposed by Holland (Holland, 1975). They are stochastic search algorithms and yet different from other approaches like Monte Carlo techniques as they combine elements of directed and stochastic search. They maintain a population of potential solutions as compared to other techniques which process a single point in the search space at a time.

GAs strength lies in performing a multi-directional search by maintaining a population of solutions and by encouraging the exchange of information among these directions. The population undergoes evolution where in each generation relatively good solutions reproduce whereas relatively bad solutions are eliminated. To evaluate the solutions an objective function is used that measures how far in the search space the solutions are from the optima. The following definition of GA by Mitchell is found suitable in the context of our work:

If the search space to be searched is large, is known not to be perfectly smooth and unimodal, or is not well understood, or if the fitness function is noisy, and if the task does not require a global optimum to be found, a GA will have a good chance of being competitive or surpassing other methods that do not use domain specific knowledge in their search procedure. (Mitchell, 1996)

GAs start with a randomly generated population of n chromosomes (candidate solutions to a problem). The suggested solution is encoded into the "genes" of the chromosome. It then calculates the fitness $f(x)$ of each chromosome x in the population which is a measure of how good a solution is relative to other solutions in the population. It then selects a pair of parent chromosomes from the current population, the probability of selection being an increasing function of fitness. New offspring are formed by using genetic operators like crossover and mutation. This process is repeated until a new population is generated. The same process is repeated. Each iteration of this process is called a generation and the entire set of generations is called a run. (Michalewiz, 1992)

A theoretical foundation of GA and their convergence to an optimal solution can be found in (Goldberg, 1989; Davis, 1991). Experiments have shown that the incorporation of problem specific knowledge generally improves GAs (Davis, 1991). In this chapter, attention will be paid to how specific ATM information have been incorporated in GAs.

GAs have found wide application in problem solving such as evolving cellular automata (Das *et al.*, 1994), evolving computer programs (Koza, 1994) and evolving Neural Networks (Whitely, 1993). In this chapter, we attempt to evolve air traffic scenarios by using GAs.

6.1.2 GAs in Air Traffic Management

Given the increasingly complex nature of ATM problems, traditional optimization techniques now have limited success in solving them. GAs have been successfully applied in air traffic planning (Delahaye *et al.*, 1999), conflict detection and resolution (Alliot *et al.*, 1996; Stephane and Sheila, 2001), scenario generation (Paglione *et al.*, 2003), and weather avoidance (Alam *et al.*, 2006).

In (Stephane and Sheila, 2001), Mondoloni proposed an airborne conflict resolution algorithm based on GA, both for traffic and for hazardous areas, which is capable of meeting constraints on flight plans. In the proposed approach, each gene in a chromosome represents a possible flight plan. These flight plans are first perturbed through genetic operators and then boundary constraints are imposed on them. Trajectories are generated through these flight plans and a conflict detection function is called. The fitness of each trajectory is evaluated based upon conflict information for conflicted flight plans. A flight rules function is used to accept conflict information and determine whether the aircraft should move according to the rules of flight. The newly evaluated flight plans are then combined with the best flight plans from the prior iteration and ranked according to fitness. In the selection process, flight plans are selected in proportion to their fitness, where higher fitness flight plans have higher probability of selection.

Alam (Alam *et al.*, 2006) used GA for path planning in a weather constrained Free Flight air traffic environment. The weather constrained airspace is discretized into a hyper-rectangular grid and a GA is used to find weather free routes while incorporating other flight optimization objectives such as minimize deviation from planned trajectories and reduce excessive climb and descent maneuvers. Each fixed length chromosomes represents a path, in grid coordinates, from a maneuver start point to a maneuver end point and transition from one point to the other. Paths are initially generated at random, then grid bounds and direction constrains are implied to eliminate infeasible solutions. The problem is formulated as a multi-objective optimization problem where the objectives are to minimize weather cell encountered along with other flight optimization objectives.

Delahaye (Delahaye *et al.*, 1999) applied GA for an air TFM problem where for each flight, a small set of possible routes and departure times is created. A simple GA algorithm is used to assign the departure time and the route for each flight in order to minimize the workload over all involved sectors. In the proposed approach, each chromosome is built as a matrix which encapsulates the new slot for the time of departure and the new route number for the flight path. The fitness function is defined as the ratio of the congestion associated with the initial distribution of the flight plans and the distribution given by the chromosome. Genetic operators are applied to add new information to the chromosome which indicates for each gene, the maximum level of sector congestion encountered during a flight.

In a closely related work, Pagolinie (Paglione *et al.*, 2003) applied GA for generating conflict scenarios by time shifting recorded air traffic data. GAs are utilized to determine the values of time shifts for the recorded air traffic data to obtain the desired conflict properties. The time shift of each flight is a gene on a chromosome that represents a vector of N time shifts for a set of N flights. Therefore, the number of genes in a chromosomes is equal to the number of flights in the scenario. The fitness function is based on the number of encounters which fall in the defined primary conflict properties bounds. A sigma scaling selection technique is used to favor those chromosomes with a fitness value close to the average fitness. Two point crossover and elitism are used. Convergence is achieved when a chromosome with a fitness function value of 1.0 is achieved.

Though the results show that, primary properties of conflict are of close match with the time shifted conflict data the proposed approach requires air traffic data which is difficult to obtain at times and to process due to lack of standardization in data sources. The approach also preserves the structured nature of traffic flow which is an artifact of the present day air traffic system. This is not suitable for evaluating CD algorithms for future Free Flight scenarios, where the structure of airspace might be entirely different or even absent.

Our proposed methodology differs in: firstly, the scenarios are generated algorithmically from scratch eliminating the need for field data (discussed in the last chapter) and secondly the scenario generation and evaluation process are linked together in a way that the latter provides feedback to the former. This feedback directs the evolutionary process of conflict scenarios towards complexity. Our approach has assumptions such as perfect world with zero wind condition, no flight technical errors and trajectory generation error. However, these assumptions are necessary to achieve the desired conflicts at a given time and location in space.

6.2 Evolutionary Scenario Planning

GA is used to evolve conflict characteristics further which are then transformed into complex conflict scenarios. This requires searching through a large number of possible solutions. Such a problem can be effectively dealt with by an intelligent strategy for choosing the direction of search with a population based approach. The evolutionary process of natural selection provides a simple way of addressing such problem where evolutionary process is searching among a large number of possibilities. GAs are stochastic search algorithms and yet different from other approaches like Monte Carlo techniques, Simulated annealing and Tabu search. GAs strength lies in performing a multi-directional search by maintaining a population of solutions and by encouraging the exchange of information among these directions. The population undergoes evolution where in each generation relatively good solutions (scenarios where CD algorithms gave large number of MDs and FAs) reproduce whereas relatively bad solutions (scenarios where CD algorithms gave less or zero MDs and FAs) are eliminated.

In the proposed evolutionary scenario planning methodology, every chromosome represents an air traffic scenario, where the pair of conflicting aircraft in the scenario are represented as a gene of the chromosome. Initially, random scenarios with conflicts are generated and then based on the fitness of a scenario and applying the genetic operators, complex conflict scenarios are evolved over generations. In the proposed approach, complex conflict scenarios are deemed "fitter individuals" having increased likelihood of survival to the next generation in the evolutionary process of a genetic algorithm. For the evolutionary process, we use the non-dominated sorting genetic algorithm (NSGA-II) (Deb *et al.*, 2002) to generate the set of efficient solutions on the given objectives.

6.2.1 Encoding Conflict parameters in Chromosomes

The primary properties of a conflict, identified in the last chapter, are the key genetic materials encoded in the chromosome. These are horizontal separation at CPA, vertical separation at CPA, ownship and intruder geometry (climb, cruise or descent) and the conflict angle between two conflicting flights. A real valued representation with linear chromosome structure is chosen to represent a conflict scenario. Every gene of the chromosome encodes the parameters of a conflict and represents a conflict between a pair of aircraft as shown in Figure 6.1. For example, if there are 50 genes in a chromosome, then the number of aircraft in the airspace will be 100 as each gene represents a unique pair-wise conflict. Thus a set of chromosomes can be seen as several air traffic scenarios, where the properties of conflict are characterized by the genetic encoded information in the respective chromosome.

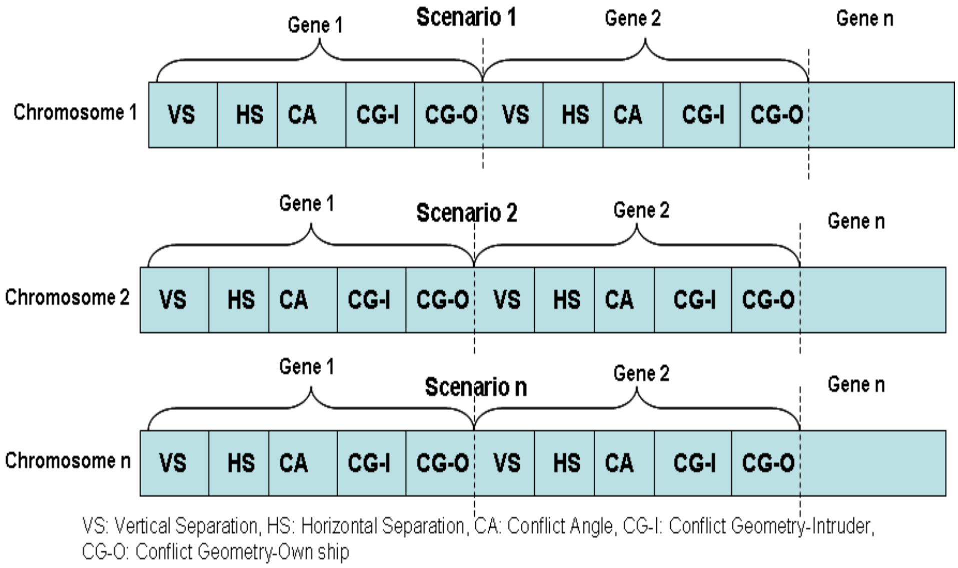


Figure 6.1: Encoding of conflict parameters in a linear chromosome structure

Before initializing the chromosomes, the following steps are performed:

- Decide on the airspace region (the extent of research airspace in 3D): It is defined by start and end latitudes, start and end longitudes and a vertical altitude range. All conflict events are simulated within this region.

- Initialize the simulation run time: It is the time window in which the aircraft are activated and deactivated. A scenario is executed until the last flight in the airspace is deactivated.

- Initialize the airspace density: The total number of aircraft to be executed in a scenario. The actual density, however, depends upon the activation time of flights.

- Initialize the minimum and maximum range for the variables: The upper and lower bounds for the conflict parameters to be encoded. For HS violation: {0.0 nm - 5.0 nm}, for VS violation: {0.0 ft - 1000 ft}, for CA :{0 degrees - 180 degrees}, for own ship and intruder geometry we randomly selected a floor value in the interval {1.0 and 3.0} so that 1 represents climb, 2 represents cruise and 3 represents descent

Then each allele in the gene is initialized randomly within the given bounds. A chromosome with its genes after initialization is illustrated in Figure 6.2.

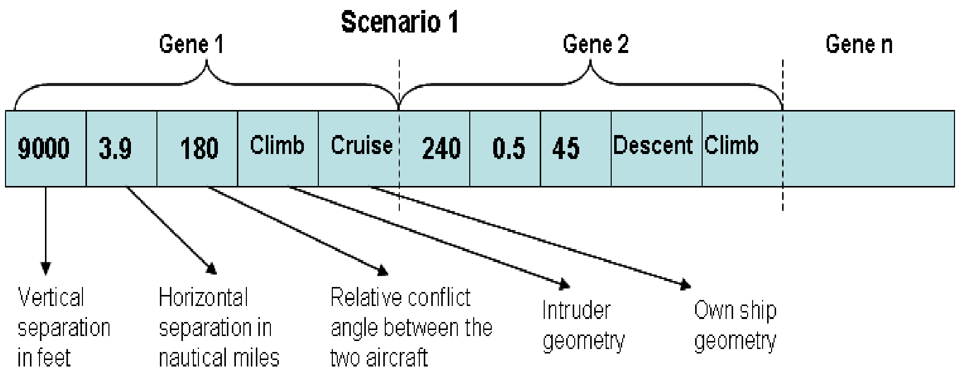


Figure 6.2: A chromosome with its genes after initialization.

6.2.2 Fitness Function

The fitness of a scenario is based on how a CD algorithm performed on it. If a CD algorithm performs well (correctly predicted conflicts) in a scenario then the scenario fitness is low; and if it performs poorly then the fitness is high. We use MD and FA as primary metrics for a CD algorithm's performance. They are defined as follows:

- Missed Detects (MD): This metric represents the number of potential conflicts which resulted in a separation violation but the CD algorithm failed to detect them.
- False Alarms (FA): This metric represents the number of conflict alerts that didn't actually materialize into a separation violation, but the CD algorithm labeled them as potential conflicts.

The higher the MD and FA the better the scenario is. Mathematically, we can formulate this as:
If S is a given scenario with N flights, A is the CD algorithm; D represents the actual CD function in the scenario, C represents the set of conflicts detected, and P represents the set of actual conflicts.

$$S = \{f_i : i = 1, 2, ...N\} \tag{6.1}$$

$$C = \{(f, f^{`}) : A(f, f^{`}) = 1 \quad \forall \quad f, f^{`} \in S, f \neq f^{`}\} \tag{6.2}$$

$$P = \{(f, f^{`}) : D(f, f^{`}) = 1 \quad \forall \quad f, f^{`} \in S, f \neq f^{`}\} \tag{6.3}$$

Table 6.1: Truth table for MD and FA

Event	**Conflict happened**	**Conflict does not happened**
Probe	CD predict conflict and it occurs(Correct Detect)	CD predict conflict and it does not occurs(False Alarm)
No Probe	CD does not predict conflict and it occurs(Missed Detect)	CD does not predict conflict and it does not occurs (No Call)
Total no. of Probe	Total no. of Conflicts	Total no. of non-conflicts

then FA are defined as:

$$FA = \{(f, f^{`}) : (f, f^{`}) \in C \wedge (f, f^{`}) \notin P\} \tag{6.4}$$

and MD as:

$$MD = \{(f, f^{`}) : (f, f^{`}) \in P \wedge (f, f^{`}) \notin C\} \tag{6.5}$$

Thus the objective functions can be defined as a maximization problem where the objectives of the evolutionary process is to maximize the events of MD and FA in a scenario.

$$MAX \begin{cases} f_1 = FA \\ f_2 = MD \end{cases} \tag{6.6}$$

The relationship between MD and FA is shown as a truth table in Table 6.1.

This can be seen as a multi-objective optimization problem where the two objectives to be maximized are conflicting in nature. If an algorithm generates a large number of FA this will lead to a lesser number of MD as the algorithm is raising conflict flag, for a large number of conflict probes and vice versa. Figure 6.3 shows a typical conflict detection process. Using the current and projected state information from the environment the probe characteristics are derived. Probe characteristic are metrics based on which a CD algorithm identifies a future conflict event. If a CD algorithm detect it as a conflict then the probe characteristics becomes conflict characteristics. We have use the term probe and conflict characteristics interchangeably depending upon the context in subsequent chapters.

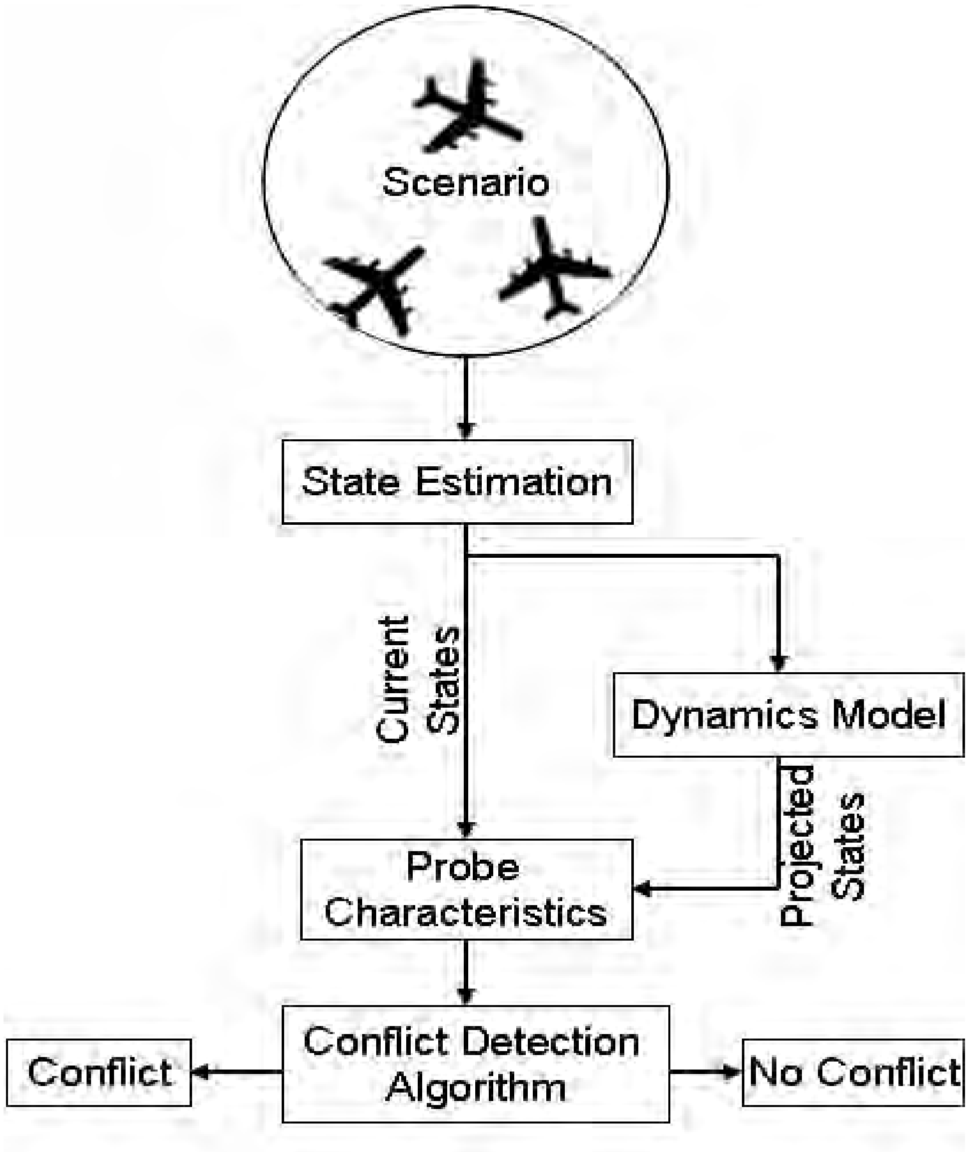


Figure 6.3: A conflict detection process.

6.3 Multi-Objective Optimization using GAs

Solutions to a such multi-objective optimization problem can be expressed in terms of non-dominated solutions. A solution dominates another, only if it has an equal and a better performance in at least one of the criterion. A solution is said to be Pareto-optimal if it cannot be dominated by any other available solution.

Pareto optimality is formally defined as: *"A solution $\vec{x_u}$ ϵ $\digamma$ (where $\digamma$ is the feasible region) is said to be Pareto optimal if and only if there is no $\vec{x_v}$ ϵ $\digamma$ for which $v = f(x_v) = (v_1, ..., v_k)$ dominates $u = f(x_u) = (u_1, ..., u_k)$, where k is the number of*

objectives, which means that $\vec{x_u}$ is Pareto optimal if there exists no feasible vector $\vec{x_v}$ which would decrease some criterion without causing a simultaneous increase in at least one other criterion in a minimization problem."
(Coello and Lamont, 2004)

The set of all Pareto-optimal decision vectors is referred to as the Pareto-optimal set of the problem. The corresponding set of objective vectors is called the non-dominated set, or Pareto front. The solutions on the Pareto front dominate all other possible solutions.

Due to the conflicting nature of objectives in the current problem none of the feasible solutions allow simultaneous optimal solutions for all objectives. For such problems, Pareto-based evolutionary optimization has now become an alternative to classical techniques such as weighted sum. It explicitly uses Pareto dominance in order to determine the reproduction probability of non-dominated individuals by assigning ranks to them. We will use the Pareto optimal front as a measure of algorithm performance.

In this work, we have used Non-dominated Sorting Genetic Algorithm NSGA-II proposed by Deb (Deb, 2001; Deb *et al.*, 2002). The main feature of NSGA-II lies in its elitism-preservation operation. NSGA-II does not use an explicit archive; a population is used to store both elitist and non-elitist solutions for the next generation. However, for consistency, it is still considered as an archive. Firstly, the archive size is set equal to the initial population size. The current archive is then determined based on the combination of the current population and the previous archive. To do this, NSGA-II uses dominance ranking to classify the population into a number of layers, such that the first layer is the non-dominated set in the population, the second layer is the non-dominated set in the population with the first layer removed, the third layer is the non-dominated set in the population with the first and second layers removed and so on. The archive is created based on the order of ranking layers: the best rank being selected first. If the number of individuals in the archive is smaller than the population size, the next layer will be taken into account and so forth. If adding a layer makes the number of individuals in the archive exceed the initial population size, a truncation operator is applied to that layer using *crowding distance*. The *crowding distance* D of a solution x is calculated as follows: the population is sorted according to each objective to find adjacent solutions to x; boundary solutions are assigned infinite values; the average of the differences between the adjacent solutions in each objective is calculated; the truncation operator removes the individual with the smallest *crowding distance*.

$$D(x) = \sum_{m=1}^{M} \frac{F_m^{I_x^m+1} - F_m^{I_x^m-1}}{F_m^{max} - F_m^{min}} \tag{6.7}$$

in which F is the vector of objective values, and I_x^m returns the sorted index of solution x, according to objective m^{th}. An offspring population of the same size as the initial population is then created from the archive, by using crowded tournament selection, crossover, and mutation operators. Crowded tournament selection is a traditional tournament selection method, but when two solutions have the same rank, it uses the crowding distance to break the tie.

6.3.1 Scenario Complexity

Air traffic complexity also known as "Dynamic Density" measures is a widely studied field (Kopardekar and Magyarits, 2002; Shridhar *et al.*, 1998). It is primarily a sector workload measure from an ATC perspective. In Free Flight, this measure might be critical as it may be used to define situations where the traffic is complex enough to revert back to centralized control from that place and time.

Currently, ATC system is the primarily means for the prevention of collision between aircraft in the air and on the ground, whilst expediting and maintaining an orderly flow of air traffic (ICAO, 1996). In Free Flight this task will be partially delegated to the CD algorithms. However, there is no complexity measure defined for CD algorithms which can identify the scenarios where an algorithm may become overloaded or will potentially fail.

For our research purpose, we use a Pareto view to define complexity as suggested in (Teo and Abbass, 2005). With a Pareto view of complexity for air traffic scenario we do not intend to introduce another measure of complexity that can overcome all previous limitations nor claiming that it is an all encompassing technique which will be able to calculate a definite complexity value for air traffic scenarios. Our objective is to propose and demonstrate that the Pareto set of solutions arising from EMO process can be useful for characterizing and comparing between the complexities of different air traffic scenario. The major advantage when using EMO based approach for capturing complexity is that it measures complexity of a particular system from an observer's point of view (Casti, 1986) and secondly since a Parteo set is the result of optimization across two or more objectives, the solution can be viewed as the result of a two way interaction that occurs between the different objectives during then optimization process. A complexity measure is said to be useful when it is able to capture what can be intuitively regarded as complex (Edmonds, 1999). As shown in Figure 6.5In our case a conflict scenario is complex from an algorithm perspective when it either fails to identify a future conflict or falsely identifies a conflict which never happens. Thus a Pareto set of solutions arising from an evolutionary process can be highly beneficial for characterizing and comparing between the complexity of different systems. Since a Pareto set is the result of optimization across two or more objectives, the solution can be viewed as a result of two way interactions that occur between the different objectives during the optimization process. Hence a Pareto approach provides a distinct advantage when used to capture complexity by generating a set of solutions that inherently exhibits properties of two way interaction.

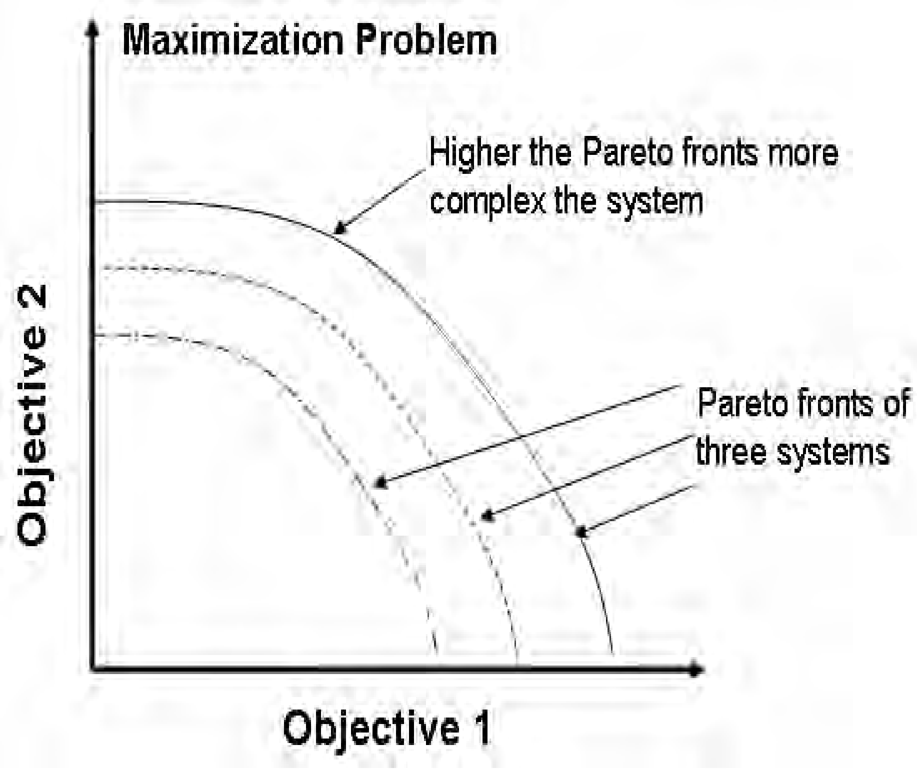


Figure 6.4: A Pareto View of Complexity for three systems (Scenarios).

The Pareto approach capturers complexity through an evolutionary optimization process that continually generates new solutions from modifications of previous solutions arising from testing and measuring of the system's performance within the given environment, which in turn is guided by the Pareto approach that imposes evolutionary pressures from multiple dimensions.

6.4 The Methodology

There are three core components of the proposed methodology: scenario generation, scenario evaluation and scenario evolution. Figure 6.5 shows the process flow in the proposed methodology and the interaction among the three components. The three components are integrated in a fast time simulation environment to facilitate repetition, replication and evolution.

- Scenario Generation: In the scenario generation component, the conflict generator module generates conflict scenarios based on the number of chromosomes (scenarios) selected. These initial scenarios which have randomly generated conflicts within the defined airspace bound forms the initial population set.
- Scenario Evaluation: In the scenario evaluation component, the generated scenarios are fed into ATOMS which then executes each of them in a fast time mode. CD algorithms are implemented in a distributed manner. Each aircraft independently checks for potential conflict using the CD algorithm under evaluation. Aircraft are flown in autopilot mode. Conflicting aircraft arrive at the CPA with the desired conflict parameters. At the end of each scenario evaluation, conflicts detected by the algorithm and actual conflicts occurred are compared to find out the MD and FA generated by the algorithm for that scenario.
- Scenario Evolution: Once all the scenarios in the initial population are evaluated, they are ranked, some of them are selected, get crossed over and mutated. These scenarios are again fed into the air traffic simulator and the process is repeated until the maximum number of generations is reached. This acts as a feedback mechanism to direct the search towards more complex conflict scenarios.

During the simulation run (executing a scenario), potential conflicts detected by the algorithm, as well as actual "conflict" events along with their conflict parameters are recorded. With each event, probe parameters are also recorded. The probe parameters considered for our research are: for two aircraft in proximity their relative angle, relative

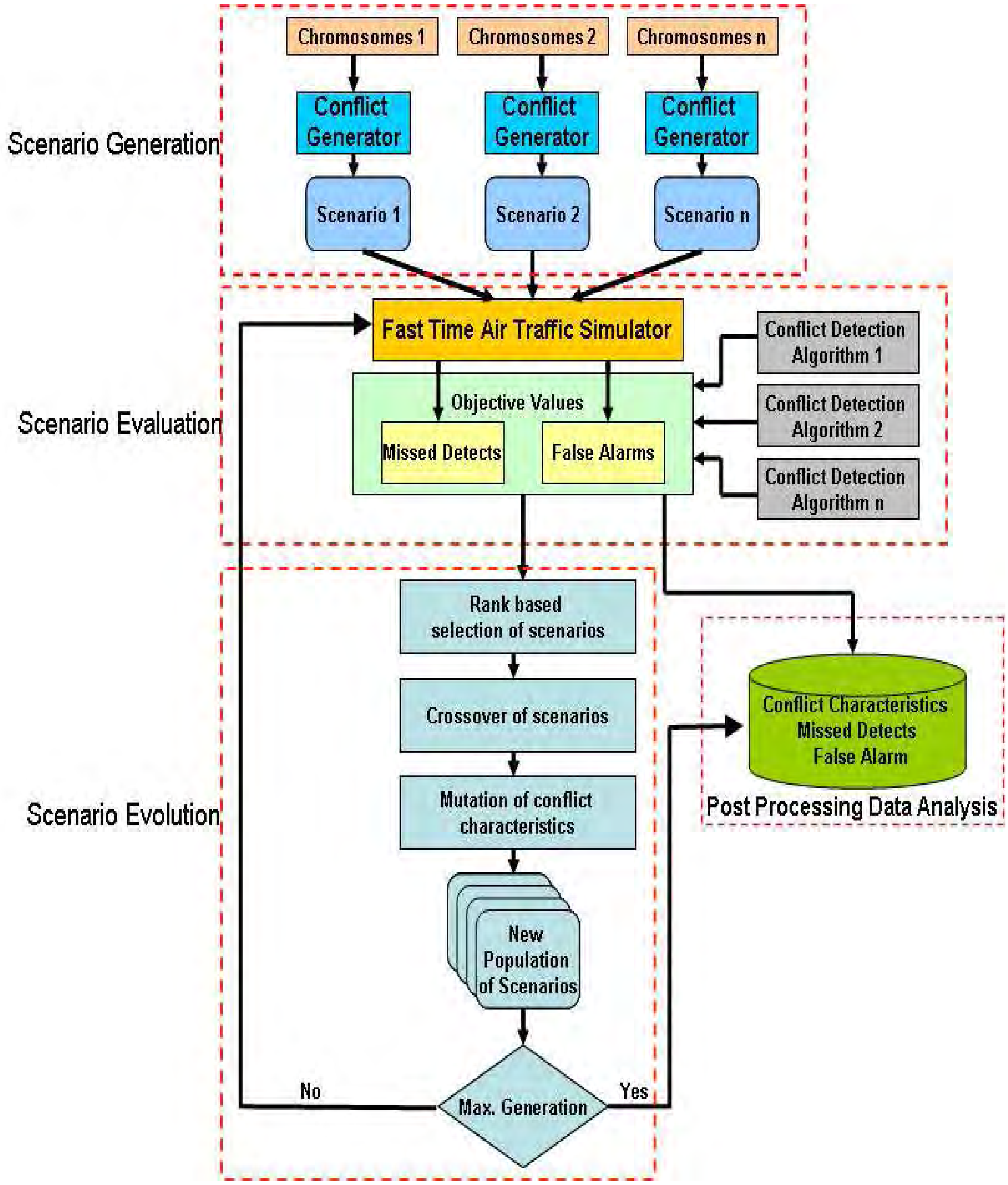


Figure 6.5: Process flow diagram of the proposed methodology showing the scenario generation, scenario evaluation and scenario evolution components.

velocity, azimuth angle, horizontal separation at the CPA, vertical separation at the CPA, ownship geometry and intruder geometry. These probe parameters are selected as they are cited as the main characteristics affecting the performance of a CD algorithm (Campos and Marques, 2002). At the end of every scenario execution, the two sets are processed to obtain the MD and FA. Each data set consists of a varying number of cases belonging to each of the two classes (i.e. FA and MD) and their associated conflict parameters. This is done to perform post data analysis on algorithm behavior in conflict scenarios. Once all the generations are over the NSGA-II generates the best non-dominated set of solution comprising of scenarios that achieved highest (non-dominated) number of MD and FA.

6.5 Evolving Scenarios with Algorithms-in-the-Loop

6.5.1 Candidate algorithms

Three CD algorithms, (discussed in chapter 2), are modeled in ATOMS for this research purposes. Dowek and Munoz (Dowek *et al.*, 2001) KB3D algorithm, Hoekstra's ASAS algorithm (Hoekstra *et al.*, 2002) and Gazit algorithm (Gazit, 1996). All three algorithms detect conflicts in the horizontal as well as vertical plane. For the three algorithms, as suggested in the literature, the look ahead time is set to 8 minutes, time to CPA: 5 minutes, probe frequency: 3 seconds, horizontal Separation: 5.0 nm, vertical separation: 1000 ft and probe range to 80 nm.

6.5.2 Experimental set up and GA Parameters

A generic airspace defined by [S32.0E142.0, S38.0E150.0] is used for air traffic simulation. Flight activation time is [1 second - 1200 second (20 minutes)]. No turn maneuvers are modeled, as having frequent turns would only lead to a large number of FA. Only the enroute phase is considered, so no TOC or TOD are computed. While TOC and TOD are not modeled, we recognize that these issues are challenging factors influencing MD and FA. Flights activate at their designated birth points and deactivates at an exit point. Only the CD component of each CD&R algorithm is implemented and aircraft fly without resolving conflicts.

A preliminary set of experiments were conducted [number of generations 20, 50, 100, 200 and population size: 20, 50, 75, 100]] to determine the convergence for the GA. Based on that, the number of generations is set to 100 and population size is set to 50. In each scenario there are 100 aircraft with at least 50 conflicts. More conflicts can also result due to overlapping of trajectories. Standard NSGA-II simulated binary crossover (SBX) operator and polynomial mutation are used for real-coded chromosomes. As suggested in NSGA-II algorithm, the probability of crossover is set to 0.9 and probability of mutation is set to 0.2 ($1/n$, where n is the number of decision variables, 5 in our case). Experiments are run on an SGI Altix 3700 Supercomputer. For each algorithm 5000 scenarios comprising of 500,000 conflicting flights are executed. The experiments are repeated for 10 different seeds. The same set of experiments are repeated for each CD algorithm. Pareto optimal curves are then generated for individual algorithms and get compared.

6.5.3 Evolution of Conflict Parameters

We first present the results of the scenario evolutionary process to see how conflict parameters are evolved over generations. The initial population represents the random conflict scenario generated initially and the final population represents the final set of conflicts evolved over generations.

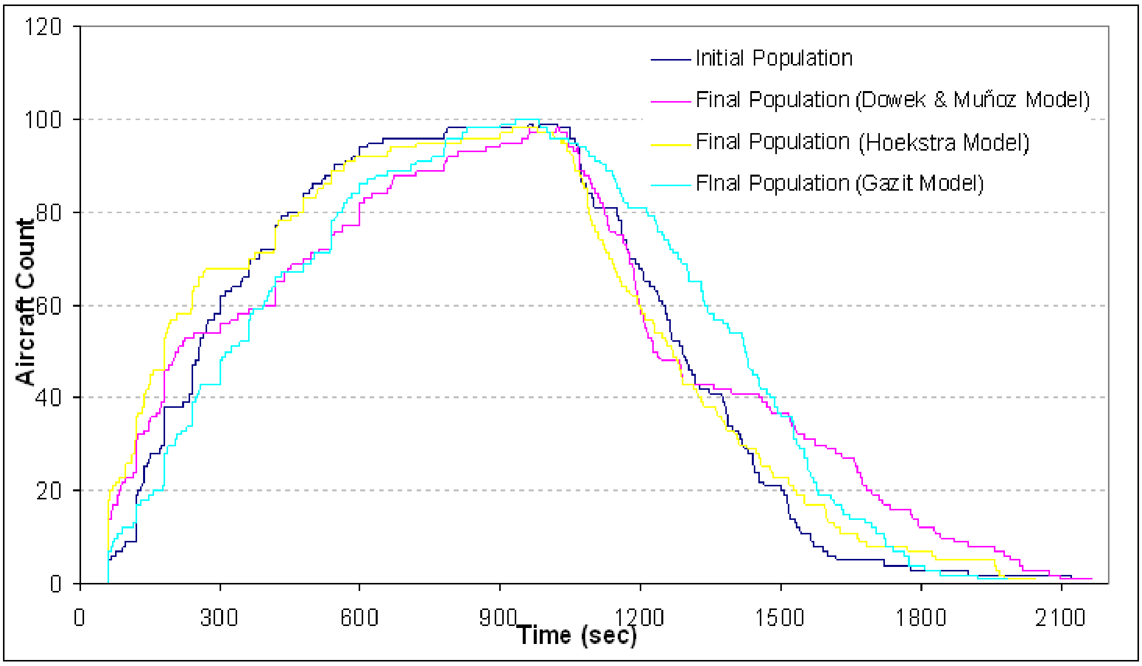


Figure 6.6: Active flights in the airspace over simulation time in the initial population and final population.

We first ascertain that the evolutionary process does not suffer from an early convergence, either by concentrating the flights in a narrow region within the airspace to generate complex conflicts or by "squeezing" the flight's activation time to a small time window to generate cascading conflicts. Figure 6.6 and Figure 6.7 show respectively that the flight's density over simulation time and conflict locations in the initial and final populations are well distributed.

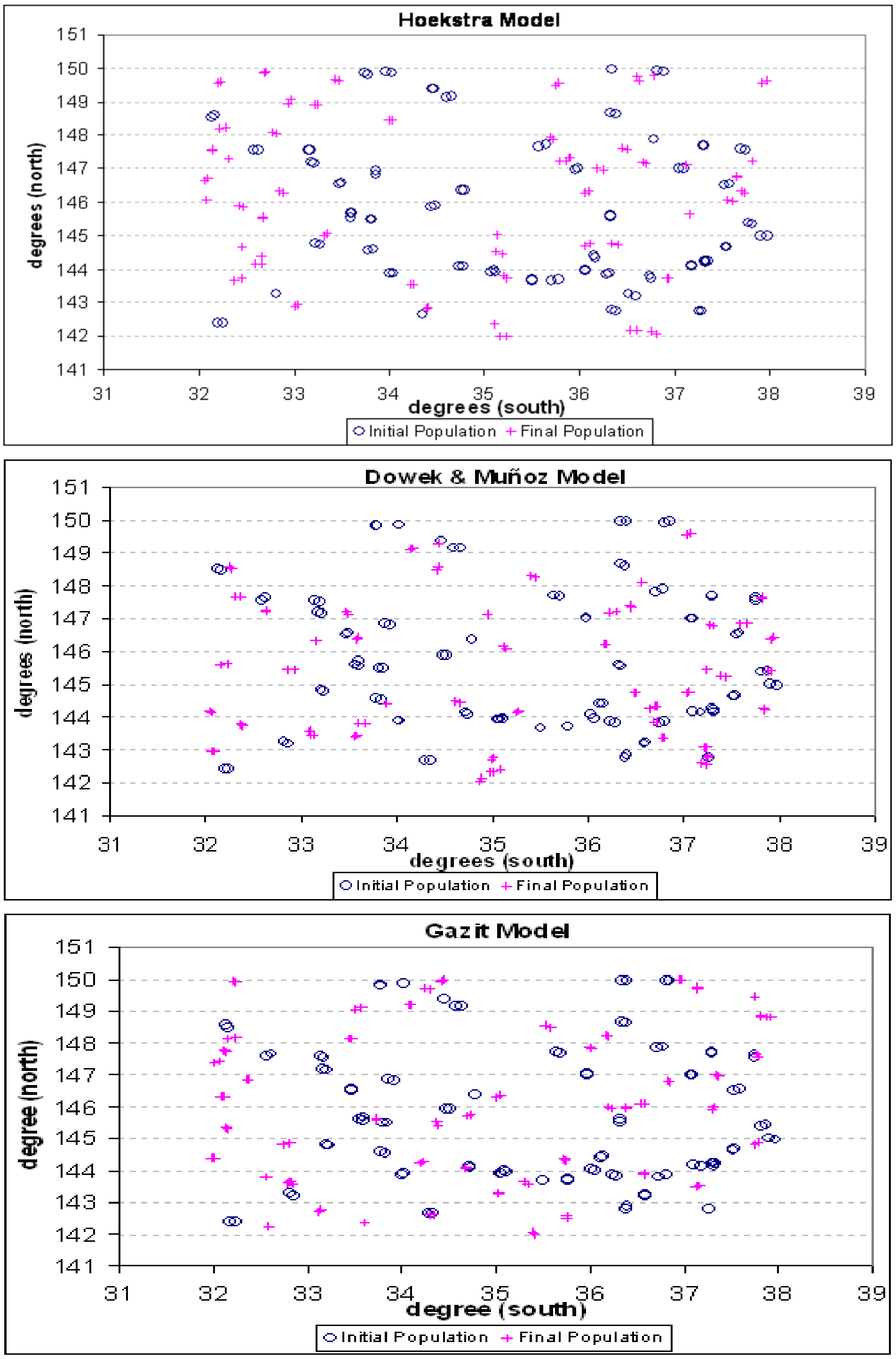


Figure 6.7: Distribution of conflicting flight CPA in the initial population and final population over the research airspace.

We then looked into how the individual conflict parameters are evolved over generations to see how the conflict parameters are driven by the evolutionary process to have maximum MD or FA generated by an algorithm. Figure 6.8 shows the frequency distribution of conflict geometry in the initial and final population during the evolutionary process. Hoekstra's algorithm is more likely to miss or falsely detect a Transition-Transition conflict. Dowek & Munoz's algorithm is somewhat stable to conflict geometry whereas Gazit's algorithm is more likely to under perform in both in Level-Level and Level-Transition conflicts.

Figure 6.9 shows the frequency of horizontal separation (nm) at CPA in the initial and final populations. Hoekstra's algorithm is susceptible to conflicts where the horizontal separation at CPA is between 3nm and 4nm. Dowek & Munoz's algorithm is susceptible to conflicts where horizontal separation at CPA is between 0nm and 2nm and Gazit's algorithm between 3nm and 4nm.

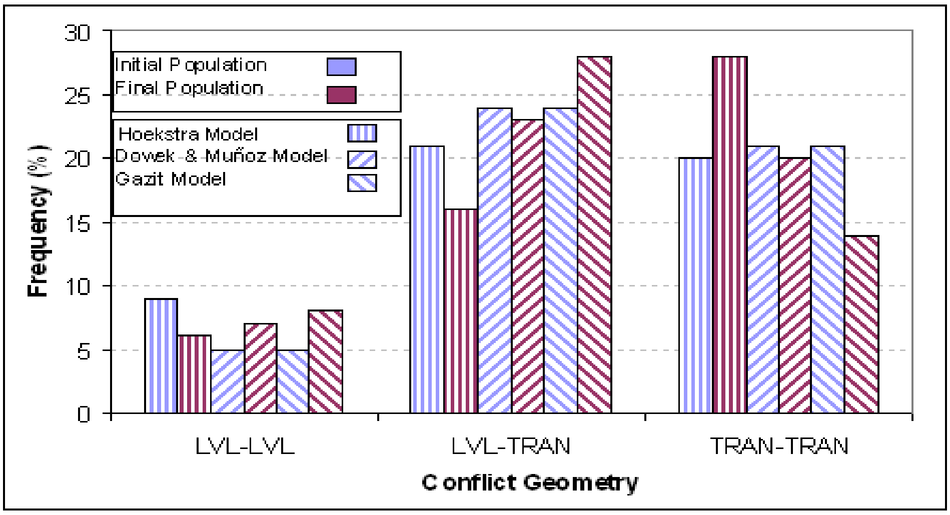


Figure 6.8: Frequency distribution of flight conflict geometry in the initial population and final population.

Similarly in Figure 6.10, which shows the frequency of vertical separation (ft) at CPA in the initial and final populations, the three algorithms perform well on the given range of vertical separation. However, Dowek & Munoz's algorithm is susceptible to low altitude separation (0ft to 100 ft).

Figure 6.11 shows the frequency distribution of conflict angle at CPA, for Hoekstra algorithm a conflict angle between 90 degrees and 120 degrees is more likely to generate MD or FA. Whereas, Dowek & Munoz algorithm is susceptible for conflicts whose angle is between 150 degrees and 180 degrees and Gazit algorithm between 60 degrees to 90 degrees. This explicit inference may not be enough when it comes to understand the behavior of a CD algorithm in a failure scenario. In the next chapter we try to derive

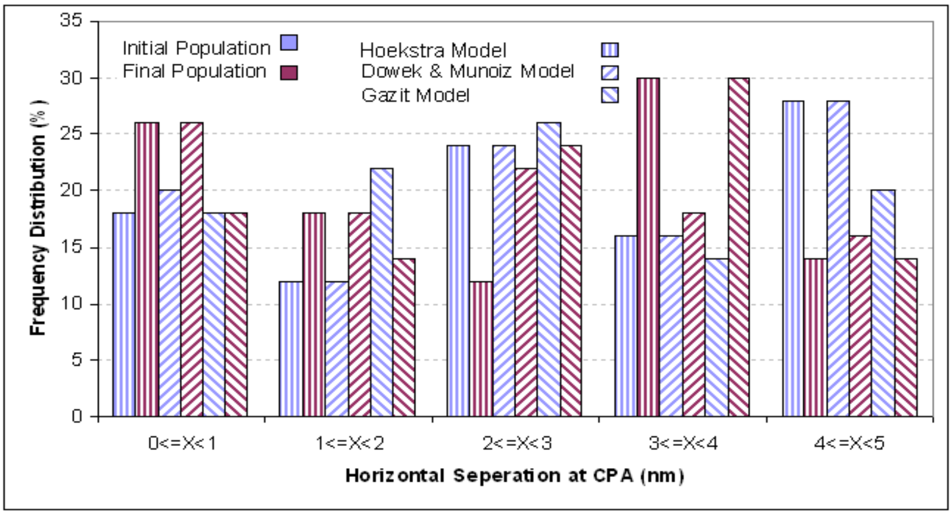


Figure 6.9: Frequency distribution of horizontal separation (nm) at CPA in the initial population and final population.

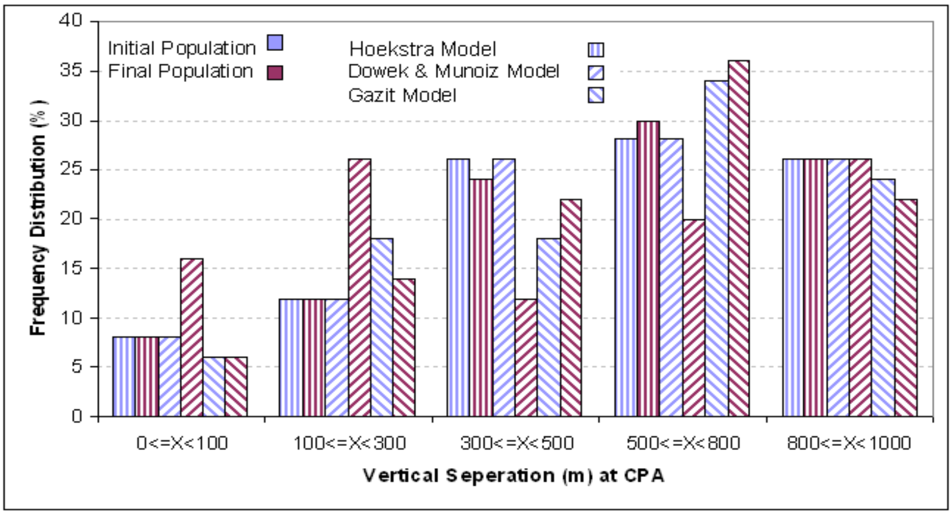


Figure 6.10: Frequency distribution of vertical separation (ft) at CPA in the initial population and final population.

implicit information from their performance patterns to gain a deeper insight into their behavior.

6.5.4 Pareto Optimal Front and Algorithm Performance

Figure 6.12 shows the overall performance of the three algorithms during the scenario evolutionary process. To reduce clutter only the initial generation and final generation are displayed.

Figure 6.12 also shows the Pareto optimal front obtained for the three algorithm. In

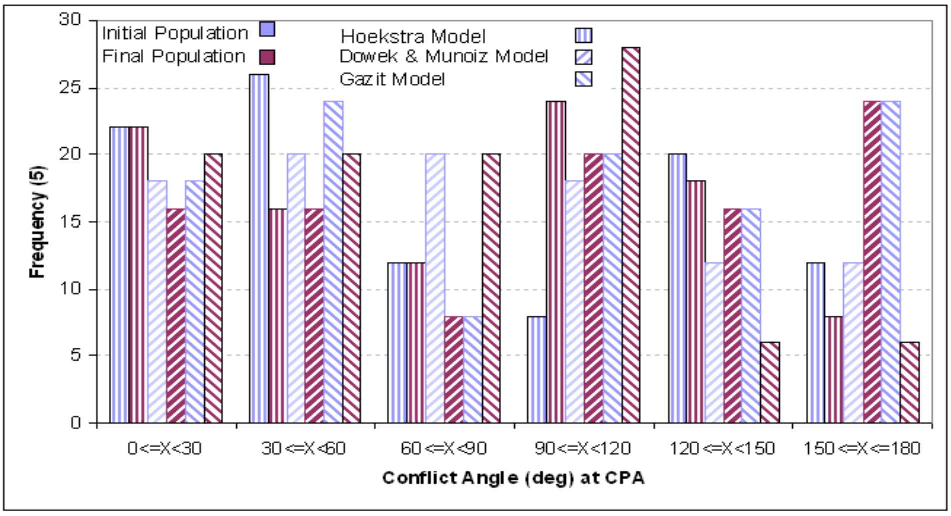


Figure 6.11: Frequency distribution of conflict angle (deg) at CPA in the initial population and final population.

the initial generations all three algorithms performed well by giving low number of MD and FA, but as the evolution progressed and complex conflict scenarios are evolved and the performance of each algorithm starts falling. In the final generation the number of MD and FA increased for all three algorithms.

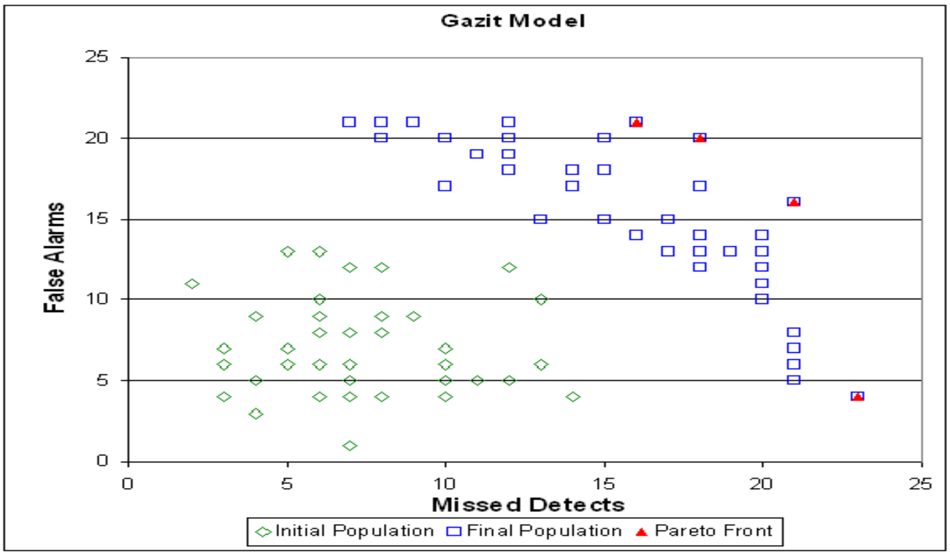
This shows that the evolutionary process is able to drive the initially generated random conflict scenarios towards highly complex conflict scenarios. The Pareto front shows the set of non-dominated solutions obtained and the trade-off between the MD and FA for each CD algorithm.

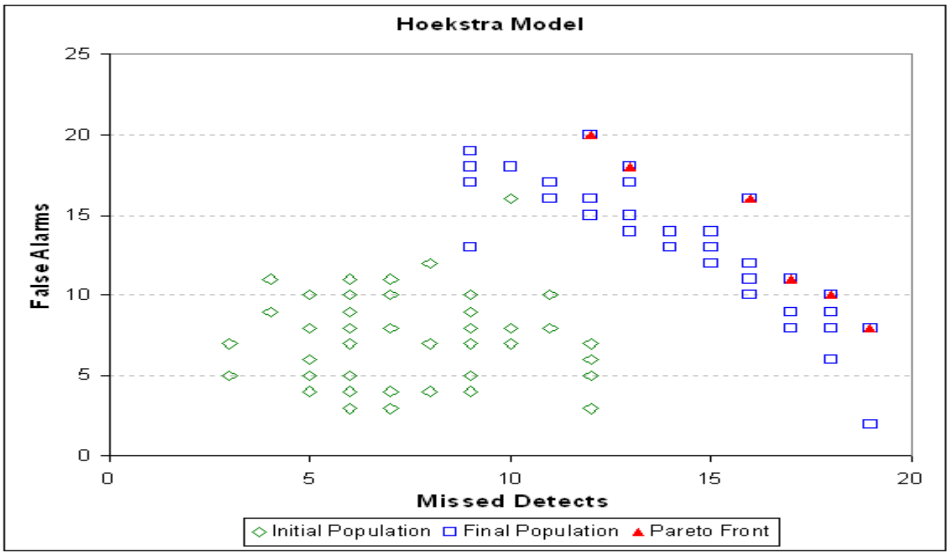
Table 7.1 show the total number of MD and FA generated by each algorithm in the scenario evolutionary process. In terms of the overall performance, for the evolved

Table 6.2: Number of Conflicts

	Dowek & Muoz Model		
	MD	FA	Overall
Number of Cases	4829	54240	59069
	Hoekstra Model		
Number of Cases	5051	48518	83344
	Gazit Model		
Number of Cases	6120	62610	68730

conflict data set, Dowek & Muoz Model generated less number of MD and Hoekstra Model generated less number of FA compared to the other two algorithms. Figure 6.13





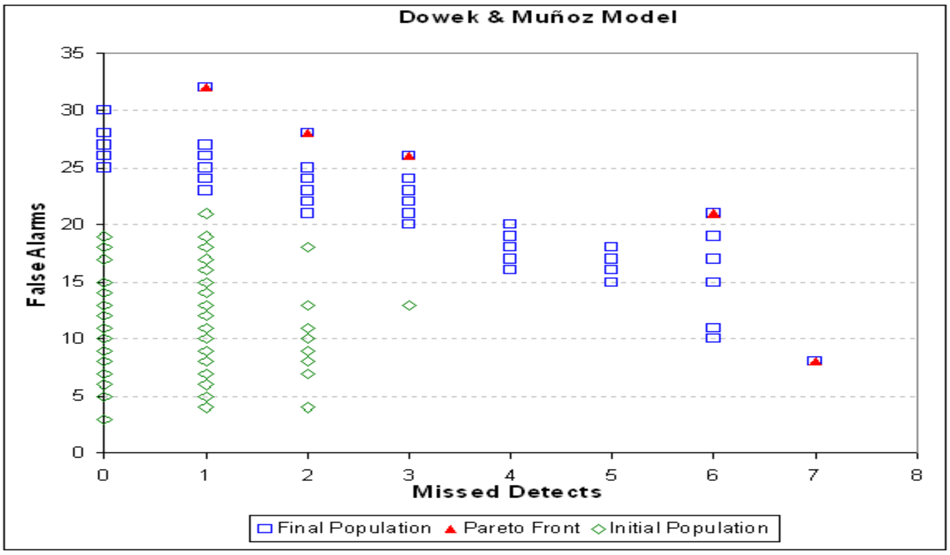


Figure 6.12: Higher number of MD and FA are generated by the CD algorithms as evolutionary process progress.

shows a snapshot of conflict parameters recorded for an MD and FA event in one of the algorithms.

Algorithm	Class	Horizontal Separation	Vertical Separation	Relative Velocity	Azimuth Angle	Conflict Angle	Ownship Geometry	Intruder Geometry
X	Missed Detect	4.98 nm	99.28 ft	778.52 kts	29.23 deg	47.67 deg	Cruise	Cruise
X	False Alarm	41.33 nm	492.12 ft	797.96 kts	136.42 deg	137.58 deg	Descent	Cruise

Figure 6.13: Conflict parameters recorded for MD and FA event generated by one of the algorithms

However, given the large amount of scenarios generated by the process it is difficult to perform any analysis manually to identify characteristics of interest. In the next chapter, we will seek help from data mining techniques to accomplish this task and apply the learned behavior towards improving the CD algorithms.

6.6 Chapter Summary

The proposed methodology of using evolutionary computation to evolve complex conflict scenarios successfully drove the algorithm towards its vulnerable areas. The evolutionary process did not suffer from early convergence and the flight density and conflict locations are well distributed in the airspace. Three CD algorithms are evaluated on evolving scenarios and their performance on given metrics is evaluated in terms of their Pareto optimal front. Integrating the scenario generation, scenario evaluation and scenario evolution processes in a fast time simulation environment provides a platform for rigorous evaluation of CD algorithms on a massive number of conflict scenarios. The evolved conflict parameters gave some indication, keeping other things equal, as to which conflict parameters can lead to algorithm's failure. However, the large amount of conflict data limits the manual analysis and any attempt to find meaningful patterns.

Chapter 7

Mining the Scenario Space

Work done in this chapter is partially based on the following publication:

1. Alam, S., Shafi, K., Abbass, H. A. and Barlow, M. Forming an Intelligent Ensemble of Conflict Detection Algorithms in Free Flight by Data Mining the Scenario Space, submitted to the Journal of Transportation Research Part-C: Emerging Technologies, Elsevier Science, Conditionally accepted : 29/02/2008

CD algorithms, due to differences in their mechanisms, differ in their effectiveness and performance. If patterns can be derived from their behavior, in a wide range of conflict scenarios, then that understanding can be used to improve their performance. In this chapter, we would like to anticipate a relationship between the algorithm failures and the conflict parameters and use it as a guide to direct future conflict probes. In the last chapter, we evaluated three CD algorithms on complex conflict scenarios and tried to find those conflict parameters where an algorithm can be vulnerable or fail. However, all three CD algorithms generated a large number of MD and FA. Given the large amount of algorithm performance data, it is difficult to perform any analysis manually to identify patterns of interest. Traditional techniques may be unsuitable due to enormity, high dimensionality, and heterogeneity nature of algorithm performance data. Thus, we used data mining techniques (Michalski *et al.*, 1998; Han and Kamber, 2001) for the discovery of interesting relationships and patterns that may exist implicitly in the algorithm performance data.

In data mining, we use a genetic based machine learning technique to learn performance patterns which cannot be induced directly from the algorithm performance data using traditional data mining techniques. Using the learned patterns, we propose an ensemble of CD algorithms which uses a selection (switch) algorithm for detecting conflicts in Free Flight. The objective is to form a predictive model for algorithm's vulnerability which can then be included in an ensemble that can minimize the overall vulnerability of the system.

7.1 Data Mining

Data mining is a non-trivial extraction of implicit, previously unknown and potentially useful information from data (Michalski *et al.*, 1998). It refers to exploration and analysis, by automatic or semi-automatic means, of large quantities of data in order to discover meaningful patterns. Data mining techniques are concerned with finding useful information or interesting patterns in large sets of data (Witten and Frank, 2000). They are part of a broader process of knowledge discovery in database. One of the task of data mining is classification of data, where given a collection of records (training set) each record contains a set of attributes, one of the attributes is the class, find a model for the class attribute as a function of the values of other attributes (Pang-Ning *et al.*, 2005). The objective is that the previously unseen records should assign a class as accurately as possible. A test set is used to determine the accuracy model. Usually, the given data set is divided into training and test sets, with the training set used to construct the model and test set used to validate it.

The objective of using Data mining techniques here is to find a generalized concept from the given data which can then be applied to future cases with high predictability. By applying such techniques, we hope to summarize and categorize the conflict parameters for each algorithm's failure. In this work, we have used a real time Learning Classifier System (LCS) as they combine a sequential covering algorithm with a GA based search technique to learn rules in a disjunctive normal form. The GA module empowers LCS to adaptively learn new rules as well as provide a generalization mechanism that enables the system to classify future cases. LCS are not merely rule learners but they also provide an interface with live environments where data can be received continuously and processed using the learnt knowledge of the system. Further, LCS are incremental and online rule learners, which means they update their classification model after seeing each data instance and without storing the instance in the memory. These powerful features make LCS an interesting approach to be used for discovering relationships and patterns in the algorithm performance data. We have used an LCS based signature extraction system

(UCSSE) as this can adaptively identifies and extracts the most generalized and accurate rules (Shafi *et al.*, 2007).

7.1.1 Learning Classifier System

LCS (Holland *et al.*, 2000) are rule based systems that are increasingly applied in various domains such as classification and data mining. Generally, there are two approaches to LCS; Pittsburgh style LCS (Smith, 1980) deals with a complete rule set as an individual classifier, whereas Michigan style LCS (Holland, 1986) considers each individual rule as a classifier. LCSs combine a sequential covering algorithm with a GA based search technique to learn rules in a disjunctive normal form. The GA module empowers LCS to learn rules which cannot be induced directly from the training data using traditional techniques. It also provides the generalization mechanism that enables the system to classify unseen cases or novel cases during a test phase. Further, LCSs are incremental and online rule learners, which mean they only need a single pass through the data and the classification model is updated after seeing each data instance.

These features provide interpretable representation of the classification model allowing an understanding of the learned rules.

7.1.2 XCS

XCS (Wilson 1995) is currently the most popular accuracy based LCS. XCS evolves a population of rules called classifiers. The rule part of a classifier has a simple syntax of *if* condition *then* action, where a condition is essentially a conjunction of predicates and the action corresponds to the class or label predicted by the classifier. Each classifier has few parameters associated with it e.g. prediction, accuracy and fitness, which are used in various decision processes in the system. The whole population of classifiers constitutes a complete solution to the problem by partitioning the problem space using an axis-parallel hyper rectangular representation. The system alternates between explore and exploit phases during each cycle of the algorithm.

During the explore phase an input example is presented in the form of a feature vector to the system and the system builds a matchset [M] of all the matching classifiers in the population. The input is covered if no matching classifier exists in the population, in which case a condition predicate is generated for each feature in the input vector. An ordinal, integral or real feature is covered using an interval representation, where each interval takes the form of (L_i, U_i) corresponding to upper and lower bounds of each interval (for example: the conflict angle between two aircraft has lower and upper bound of 0 and 180 degrees respectively). A classifier matches an input data instance if all of its feature values match with the corresponding predicate values. Several matching rules can participate in the process of deciding on a final class. A subset of the matching rules i.e. the rules participating in the decision making process form an action set [A].

XCS uses reinforcement learning to update classifier parameters in [A]. The system receives a scalar reward from the environment based on the predicted class. This feedback is first used to update classifiers' prediction. The accuracy is then calculated based on the updated prediction which in turn is used to update fitness.

The GA in XCS is responsible for introducing new rules into the population. During the selection process, two parents from the action set [A] are selected with probability proportional to their fitness. Two offspring are generated by reproducing, crossing-over, and mutating parents. Parents continue to stay in the population competing with their offsprings. If the population size is less than a certain number, offsprings are inserted into the population; otherwise, two of the most inaccurate classifiers are deleted from the population before the offspring can be inserted.

UCS (sUpervised Classifier System) is an LCS derived from XCS. Both use the same classifier's representation and a niche Genetic Algorithm (GA) as their search mechanism. The fitness of individuals (classifiers) in the population is based on their accuracy. UCS is specifically designed for classification tasks and benefits directly from the labels during training. In contrast, XCS uses a reinforcement learning approach and can be used in single or multi-step tasks. It receives an immediate or delayed reward from the environment upon predicting an action or label for an input state. Consequently, the classifier parameters in XCS and UCS are updated according to their respective learning schemes.

7.1.3 Real Time Signature Extraction Algorithm (UCSSE)

UCSSE (UCS with Real time Signature Extraction System) (Shafi *et al.*, 2007) is a LCS based signature extraction system. UCSSE automatically detects the presence of the optimal classifiers as they are discovered by UCS in real time. Figure 7.1 shows the block diagram of the real time signature extraction system.

The signature set [S] is essentially a subset of [P] consisting of optimal classifiers extracted during the operation of UCS. An input from the environment is first presented to [S] whereby a [M] is generated using current input label and the parameters of the signatures participating in the [M] are updated as usual. The discovery component of UCS i.e. the GA however is bypassed when system is run through [S]. The input is escalated to UCS only if no cover is available in [S], in which case standard UCS takes over and runs its performance and discovery components using [P] for certain number of trials. Meanwhile, the extraction process of accurate and experienced classifiers from

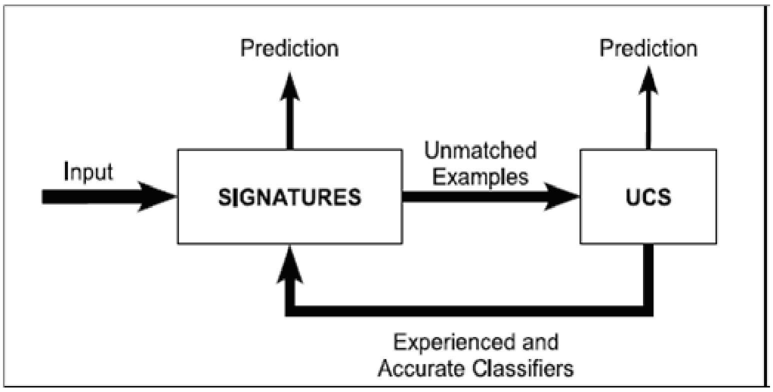


Figure 7.1: Block diagram of the real time signature extraction system. (Shafi *et al.*, 2007)

[P] to [S] is triggered in parallel periodically. Initially [S] is empty and the system runs mainly through [P] having enough exploration opportunities. The operation is shifted gradually to [S] as it starts becoming populated. The transition completes when the system discovers the best map of the input space and the control is transferred to [S] in which case the evolutionary search is completely halted and the system is made to run from [S]. A pruning step in [S] is carried out when the average experience of the signatures in [S] reaches a high value. In the pruning step all inaccurate and below average signatures are deleted from [S]. If the deletion causes a covering gap, the control is handed back to [P] and the process is repeated until the system stabilizes to run from [S] at which point the learning process can be stopped.

7.2 Mining the Scenario Space

As mentioned in chapter 6, during the simulation run (executing a scenario), potential conflicts detected by the algorithm, as well as actual "conflict happened" events along with their conflict parameters are recorded. At the end of every scenario execution the two sets are processed to obtain the MD and FA. Each data set consists of a large number of cases belonging to each of the two classes (i.e. MD and FA) and their conflict probe parameters.

Thus the problem of learning a model for the failure of CD algorithms can be framed as a 2-class problem of XCS. The goal is to extract the patterns or conflict parameters where CD algorithms produced FA and MD. Here we adopted the 2-class problem approach so that maximally non-overlapping patterns or rules can be found for each classes.

The data is fed into UCSSE in the form of feature vectors. Each feature vector, or an instance in our case, consists of the attribute values of a conflict event recorded with its corresponding probe parameters. The labels correspond to two possible outcomes of the prediction of a CD algorithm i.e. MD and FA. Since UCSSE is a supervised learning system the labels are provided along with the instances during the learning or training phase.

The system outputs its algorithm as a set of self-explanatory if-then rules. The accuracy of these rules is then evaluated using the data without labels. Figure 7.2 shows an illustration of the rule set generated for an algorithm.

Rule	Action	Accuracy
X & Y | Z	MD	89%
X | Y | Z	FA	79%
X & Y & Z	MD	88%
Z & Y	MD	76%

Figure 7.2: An illustration of the rule set generated for an algorithm. X, Y, Z represents the attribute value of the probe characteristic.

7.3 Rule Set for Algorithms

UCSSE is applied on the performance data (events and conflict parameters) obtained for the three CD algorithms. Table 7.1 summaries the classification model evolved by UCSSE for each of the CD algorithms. The number of cases shows the total number of MD and FA generated in the entire run for each algorithm. The number of rules show the compactness of the model whereas the model accuracy represents its correctness or the generalization ability.

7.3.1 Rule Sets Accuracy

From Table 7.1 we can see that each training data set consists of a large number of unique instances (59069 for Dowek & Muoz algorithm, 83344 for Hoekstra algorithm and 68730 for Gazit algorithm) but a much smaller number of rules are obtained that capture the concepts in the data. This in itself signifies an underlying relationship between the conflict parameters and the failure classes.

Table 7.1: Number of rules and test accuracy using UCSSE

	Dowek & Muoz algorithm		
	MD	FA	Overall
Number of Cases	4829	54240	59069
Number of Rules	22	25	47
Model Accuracy	79.33%	98.86%	89.09%
	Hoekstra algorithm		
Number of Cases	5051	48518	83344
Number of Rules	153	34	187
Model Accuracy	90.81%	91.24%	91.02%
	Gazit algorithm		
Number of Cases	6120	62610	68730
Number of Rules	88	186	274
Model Accuracy	77.34%	98.21%	87.77%

For Dowek & Muoz algorithm, UCSSE generated 22 rules for MD and 25 rules for FA. The accuracy of rules for FA is (98.86%) and the accuracy of rules for MD (79.33%). Whereas in Hoekstra algorithm, UCSSE generated 153 rules for MD and 34 rules for FA, with accuracies of 90.81% and 91.24% respectively. In Gazit algorithm , UCSSE generated 88 rules for MD and 186 rules for FA with accuracies of 77.34% and 98.21% respectively.

Large number of rules signifies that the problem (algorithm performance pattern) is hard to generalize. This is mainly due to the fact that the features are not sufficient to discriminate between the classes when they are overlapped or interleaved. Moreover, the number and representativeness of the available instances also influence the generalization ability of the classifier system. Despite the large number of rules generated, the overall accuracy of Hoekstra algorithm is highest (91.02%) which signifies the high overall correctness of the rules generated given the instances.

7.3.2 If-Then Rules and Interpretation

Table 7.2 provides some of the most accurate rules obtained by UCSSE that capture the conflict characteristics where the three CD algorithms produced FA and MD. HS is the horizontal separation in nm, VS is the vertical separation in feet, A_z is the relative azimuth angle in degrees, R_v is the relative velocity in knots, G_o is the ownship geometry, G_i is the intruder geometry and A_r is the conflict angle at CPA between two conflicting flights.

For each rule, its condition, the predicted class and accuracy are provided. A 100% accuracy means that the rule correctly predicts all cases it matches whereas a drop in accuracy occurs when a rule wrongly predicts some of the cases. The overlapping in rules is resolved using the rule accuracy.

The rules can be broadly interpreted as follows: For Dowek & Muoz algorithm, rule 3 states that, there is 93.4% possibility of a FA when the horizontal separation between two aircraft at their CPA is greater than 4.77 nm and less than 4.99 nm, vertical separation is less than 100.97 ft and greater than 210.45 ft, relative velocity is greater than 300.21 knots and ownship is climbing or descending and intruder is cruising.

Similarly For Hoekstra algorithm, rule 1 states that, there is a 96.0% possibility of a MD when the horizontal separation between two aircraft at their CPA is greater that 4.89 nm and less that 4.99 nm, vertical separation is less that 23.83 ft, relative azimuth angle is less than 20.43 degrees, conflict angle between the two is less than 45.38 and ownship is climbing and intruder is either descending or climbing.

For the Gazit algorithm, rule 2 states that, there is a 92.8% possibility of a MD when the horizontal separation between two aircraft at their CPA is less than 3.12 nm, azimuth angle is less than 21.37 degrees, relative velocity is greater than 614.7 knots, conflict angle is between 30.2 degrees and 90.2 degrees and ownship is descending.

Table 7.2: A subset of rules obtained using UCSSE.

No.	Condition	C	Acc
Dowek & Muoz algorithm			
1	$3.42 < HS < 3.79 \wedge 17.24 < A_z < 40.01 \wedge R_v < 95.56 \wedge A_r > 163.7 \wedge G_o$ is CR or DS $\wedge G_i$ is CR	MD	97.2
2	$HS < 0.51 \wedge 850.31 < VS < 992.12 \wedge R_v > 300.22 \wedge A_r > 132.47 \wedge G_o$ is DS $\wedge G_i$ is DS	MD	97.2
3	$4.77 < HS < 4.99 \wedge 100.97 < VS < 210.45 \wedge R_v > 300.21 \wedge G_o$ is CL or DS $\wedge G_i$ is CR	FA	93.4
4	$6.53 < HS < 7.11 \wedge A_z > 110.43 \wedge R_v < 36.0 \wedge A_r < 115.94 \wedge G_o$ is CR $\wedge G_i$ is CR or CL	FA	90.1
	...		
Hoekstra algorithm			
1	$4.89 < HS < 4.99 \wedge VS < 23.82 \wedge A_z < 20.43 \wedge R_v > 488.35 \wedge A_r < 45.38 \wedge G_o$ is CL $\wedge G_i$ is DS or CL	MD	96.0
2	$567.98 < VS < 854.64 \wedge 94.23 < A_z < 120.43 \wedge R_v < 100.67 \wedge 122.39 < A_r < 148.61 \wedge G_o$ is CL $\wedge$ G_i is CR or DS	MD	94.3
3	$5.45 < HS < 5.94 \wedge VS < 32.01 \wedge 120.26 < A_z < 162.17 \wedge R_v > 625.19 \wedge A_r > 175.35 \wedge G_o$ is DS or CL $\wedge$ G_i is CR or DS	FA	98.2
4	$804.24 < VS < 956.52 \wedge A_z > 171.61 \wedge R_v < 163.20 \wedge A_r < 7.12 \wedge G_o$ is CR or CL $\wedge G_i$ is DS	FA	92.5
	...		
Gazit algorithm			
1	$HS < 0.76 \wedge 12.12 < VS < 200.54 \wedge A_z < 30.46 \wedge 20.29 < R_v < 39.55 \wedge A_r > 116.25 \wedge G_o$ is DS $\wedge$ G_i is DS or CR	MD	98.1
2	$HS < 3.12 \wedge A_z < 21.37 \wedge R_v > 614.7 \wedge 100.2 < A_r > 110.2 \wedge G_o$ is DS $\wedge G_i$ is DS	MD	92.8
3	$3.53 < HS < 7.11 \wedge 122.42 < A_z < 164.83 \wedge R_v < 3.14 \wedge A_r > 172.4 \wedge G_o$ is CR $\wedge G_i$ is CR OR DS	FA	96.3
4	$VS < 11.10 \wedge A_z < 110.51 \wedge A_r < 13.46 \wedge$ G_o is DS or CL $\wedge G_i$ is DS	FA	92.0
	...		

For all three algorithms, the rules broadly suggest that, given a conflict probe, when ownship is descending the CD algorithm is more prone to generate FA. Similarly, when the relative conflict angle between the two aircraft is wide then the algorithms are likely to generate MD.

7.4 An Ensemble Approach for Conflict Detection

In this section we present the ensemble approach for conflict detection by leveraging the learned rules for each CD algorithm.

7.4.1 Forming an Intelligent Ensemble

The learned rules can be seen as conflict probe parameters for which an algorithm is likely to generate MD or FA. They provide an insight into the performance of a CD algorithm which is *learned* not derived from its past performance. An ensemble of the algorithms can now be formed with a rule set associated with each one of them. We can call this an intelligent ensemble as each algorithm not only has its own conflict detection mechanism but also an understanding of its weakness. This understanding (rule sets) is not simply derived from its past performance but learned from it. The future conflict probe parameters are matched against this set of rules to decide which algorithm will assess it. If probe parameters matches with an algorithm's rule set then the algorithm

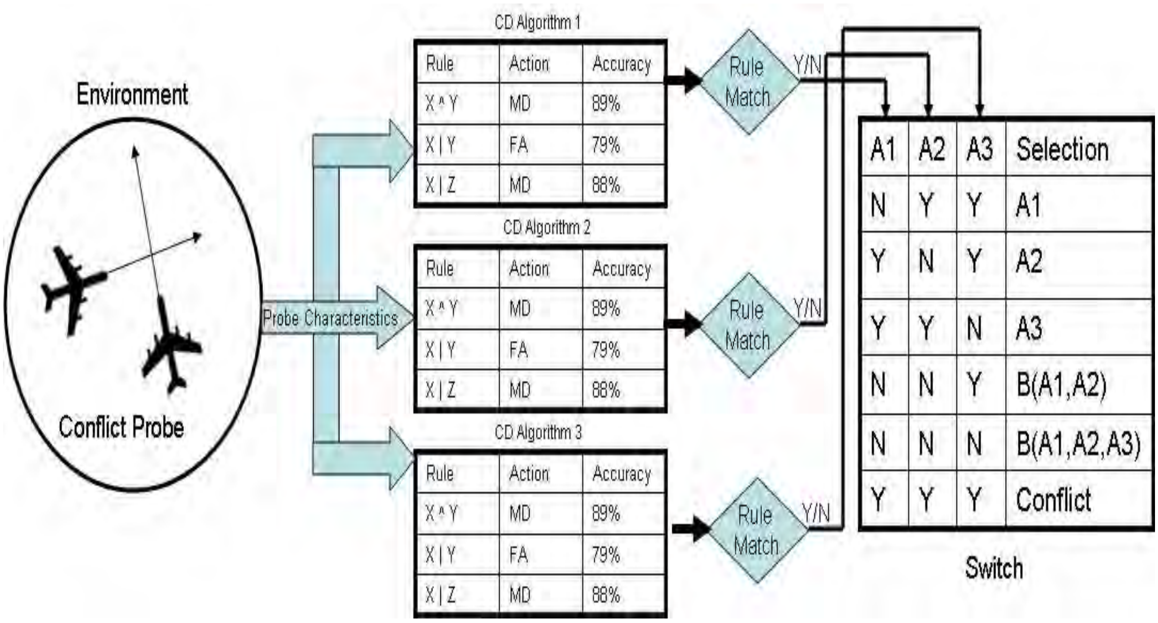


Figure 7.3: Conceptual representation of the ensemble approach. X, Y, Z notations in the rules represents individual probe feature. A1, A2, A3 represents the three CD algorithm. Y and N represent condition matched or not matched respectively. B(A1,A2,A3) means that the algorithm that has lower Pareto front is chosen.

is likely to generate MD or FA given the probe parameters. The rule match condition is "YES". If a probe parameters does not matches with an algorithm's rule set then this means that the algorithm is less likely to generate MD or FA given the probe parameters. The rule match condition is “NO”.

Thus in the proposed approach, a conflict probe is not directly assessed by a CD algorithm but a switch mechanism, based on rule match conditions, route it to an algorithm which is less likely to generate MD or FA. Figure 7.3 shows a conceptual implementation of the ensemble approach. The rule sets and the algorithms selection process using the switch now becomes a part of the overall conflict detection process.

7.4.2 A Switch Mechanism to Direct the Probe

For a given conflict probe, the rule sets for each algorithm returned either “YES” or “NO” as discussed in the previous section. As shown in Figure 7.3 (switch), there are three different possibilities:
First, when there is only one “NO” rule match and the rest are “YES” rule matches. In this case the algorithm whose rule set returned ”NO” is chosen for evaluating the probe for potential conflict. For example, as shown in Figure 7.3, in the 2^{nd} row, only algorithm A1 rule set doesn’t have a match, and the other two had a match. So A1 is selected for conflict probe evaluation.
Second, when there are two or more “NO” rule matches. In this case for the algorithms whose rule set returned “NO”, the one with lowest Pareto front obtained in the individual run is chosen for conflict probe evaluation. The 5^{th} and 6^{th} rows in the switch table shown in Figure 7.3 are examples of this possibility.
Third, when all of them are “YES” rule matches, last rows in the switch table. In this case all three algorithms are likely to either miss detect a conflict probe or falsely identify a probe as conflict. The performance improvement by using ensemble approach will largely depend upon how this condition is dealt with in the switch.

Algorithm 5 shows the logical implementation of the switch mechanism in the conflict detection module of ATOMS. Two possibilities can be considered for the third condition, first, label it as a conflict and assume it to be routed to flight crew which then take necessary measures to deal with it. Second, choose the algorithm with lowest accuracy for the rule set matched and direct the probe to it for evaluation. Since the first approach requires human-in-the-loop so from algorithmic quantitative evaluation perspective we have taken the first approach which labels such a case as a conflict event. If in such a

Require: Probe parameters P :(Horizontal separation, vertical separation, conflict angle, azimuth angle, relative speed, ownship geometry, intruder geometry)
Set of CD algorithm A (A_1, A_2, A_3)
Priority Rule B (based on the Pareto optimal fronts of A)
Rule set R for each algorithm (R_1, R_2, R_3)

if P $\notin R_1$ AND P $\in R_2$ AND P $\in R_3$ **then**
    Route the Probe to A_1
else if P $\in R_1$ AND P $\notin R_2$ AND P $\in R_3$ **then**
    Route the Probe to A_2
else if P $\in R_1$ AND P $\in R_2$ AND P $\notin R_3$ **then**
    Route the Probe to A_3
else if P $\notin R_1$ AND P $\notin R_2$ AND P $\in R_3$ **then**
    Route the Probe to B(A_1, A_2)
else if P $\notin R_1$ AND P $\notin R_2$ AND P $\notin R_3$ **then**
    Route the Probe to B(A_1, A_2, A_3)
else if P $\in R_1$ AND P $\in R_2$ AND P $\in R_3$ **then**
    return Conflict
end if

Algorithm 5: The pseudo code for switch to select the CD algorithms for probe evaluation

scenario when a probe is flagged as a conflict, and the probed aircraft didn't have a loss of separation then this will generate a FA.

7.5 Experiment & Results with the Ensemble of Algorithms

Based on the proposed ensemble approach, changes are made to the conflict detection module of ATOMS. Each aircraft in the simulation now has three CD algorithms, their rules set and the switch mechanism formed as an ensemble. Experiments, with different traffic samples are repeated, for 10 different seeds on the same platform (SGI Altix 3700 Supercomputer). For the ensemble approach, as with individual CD algorithm's evaluation, there are 100 generation with 50 scenario in each generation and 100 aircraft in each scenarios with at least 50 conflicts, making 5 million conflict events in total.

Scenarios are evolved and evaluated in fast time simulation mode with an ensemble of CD algorithms detecting the conflicts. After each run MD and FA are processed as detailed in chapter 6. At the end of 100 generations, Pareto optimal curves are generated for the ensemble approach and compared with the Pareto optimal curve of the individual CD algorithms (generated in chapter 6). The Figure 7.5 and 7.6 shows the average from 10 runs with different seeds.

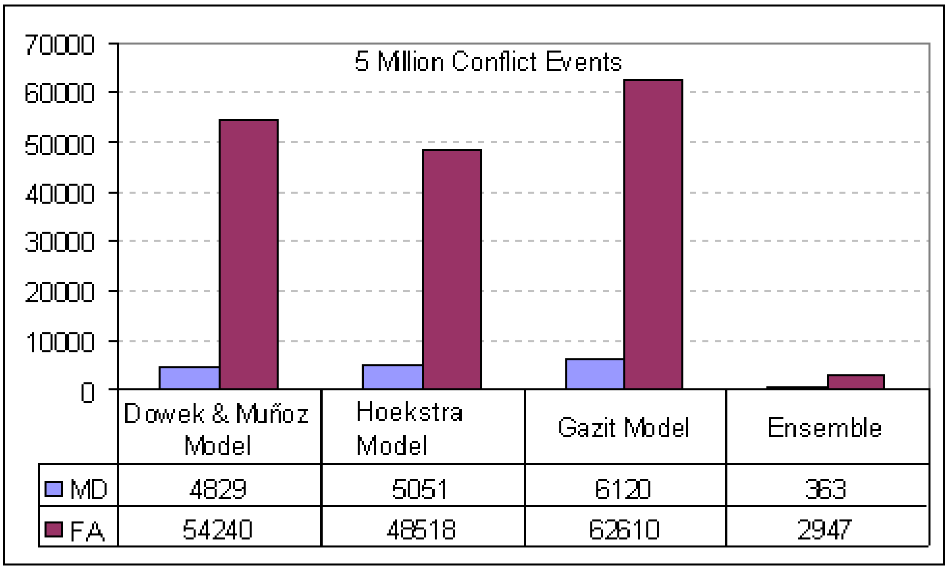


Figure 7.4: MD and FA with CD algorithms and ensemble approach

As shown in Figure 7.4, the ensemble approach generated 363 MD and 2947 FA events. There are 393 FA generated due to conflicts miss-detected by the switch i.e. when the probe matched with the rule set of the three CD algorithms, and 72 MD generated due to wrong choice of CD algorithm i.e. when the probe didn't match any of the three CD algorithms and selection is made based on the Pareto front.

7.5.1 Pareto Front of the Ensemble Approach

Figure 7.5 shows the initial and final population of solutions and the Pareto front of the solutions with the ensemble approach. With the ensemble approach there is no MD in the initial population. In the final population, the maximum number of MD is only two and the maximum number of FA is eight. With the ensemble approach the initial population have false alarms but no missed detects. This can be explained by the all "YES" rule-match condition of the switch where the algorithm with lowest accuracy for the rule set matched is used to evaluate the probe for conflict. A loss of separation may generate a false alarm but will not generate a missed detect which may lead to a loss of separation.

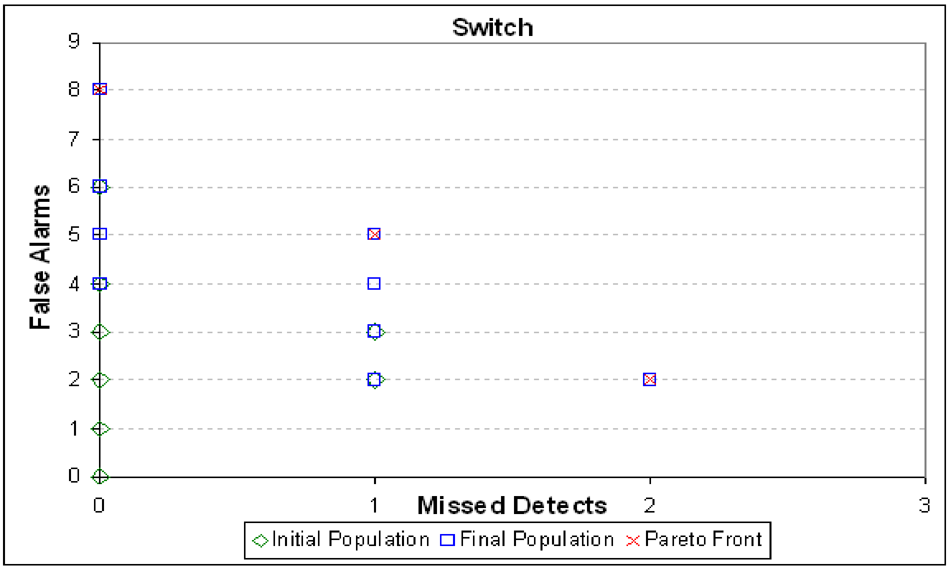


Figure 7.5: Pareto front of solutions obtained using the ensemble approach

7.5.2 Comparing the Pareto Fronts

We then compare the ensemble performance with the individual algorithm performance using their Pareto optimal fronts. Figure 7.6 shows that the Pareto front generated by the ensemble approach is lower than any of the three CD algorithms. In one of the Pareto Optimal solutions, the ensemble had zero MD which none of the algorithms reached in the evaluation process.

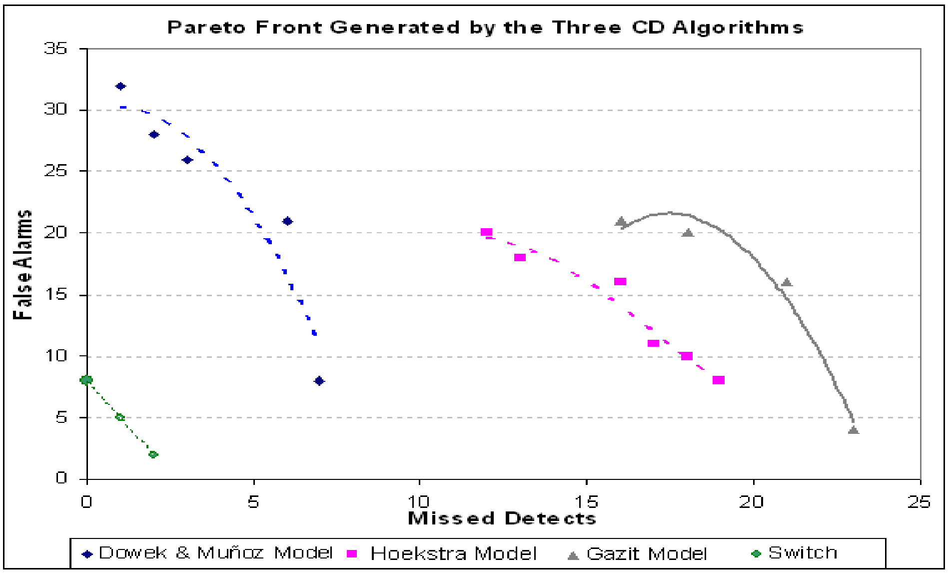


Figure 7.6: Pareto front of solutions for the three algorithms and ensemble approach

7.6 Discussions

The advantage of the ensemble approach is that, once their rule set are formulated, any number of CD algorithms can be added in the ensemble making the conflict detection process more robust.

This approach also strikes a balance between various conflict detection mechanisms by choosing the most appropriate algorithm given probe parameters. In this chapter, only one ensemble approach is tested, other possible ensemble approaches of clubbing the two algorithms together or using majority-will-select criterion need to be explored to identify the best possible ensemble. Moreover, the performance of a CD algorithm is sensitive to parameters setting such as look ahead time, time to CPA and separation standards. Since we have used the parameter's setting suggested in the literature, the performance may vary with different parameter settings and may impact the accuracy of the rule sets generated and the overall effectiveness of the ensemble approach.

The performance of the ensemble largely depends upon the accuracy of the rules and the traffic samples used to build the rule sets. It might not be possible to cover all the possible probe characteristics in Free Flight, but we demonstrated that the evolutionary approach successfully lead the algorithms to the problem sub-space where the vulnerabilities of the algorithms lies.

In future cockpits, redundancy in the conflict detection system will be necessary as it is usually done for other critical avionics. The proposed ensemble approach may utilize this redundancy, in a meaningful manner, to reduce the risk of MD or FA.

7.7 Chapter Summary

In this chapter, we proposed an ensemble approach for conflict detection in Free Flight. The ensemble consists of several CD algorithms, a rule set for each algorithm describing its likely behavior given probe parameters and a switch mechanism to choose the most appropriate algorithm. An evolutionary computation based approach is used to evolve complex conflict scenarios which are then evaluated in a fast time air traffic simulator. This allows for rigorous evaluation of CD algorithms on a variety of conflict scenarios. Data mining techniques are then employed to identify patterns in the probe characteristics where the CD algorithms missed or falsely identified a conflict. These patterns are formulated as rule sets for each algorithm and are then used by a switch mechanism to route a probe to an appropriate algorithm. The algorithm which is less likely to miss or falsely identify a conflict, given probe parameters, is selected to evaluate the probe for potential conflict.

The performance of the ensemble and of individual algorithms is evaluated by comparing the Pareto front of solutions generated by them. The ensemble approach significantly reduces the number of MD and FA as compared to individual algorithms. The proposed methodology is not only capable of accommodating new CD algorithms for Free Flight but can be extended to other advanced ATM concepts as well.

Chapter 8

Conclusion

8.1 Summary of Results

In this thesis, we presented a methodology for evolving complex conflict scenarios to evaluate CD algorithms and to discover patterns in their behavior. This "learned understanding" is transformed into a rule set with which an ensemble of CD algorithms applied for improving the overall performance of the conflict detection process.

An air traffic simulation system (ATOMS) is designed and developed to prototype and evaluate advanced ATM concepts such as, airborne separation assurance (ASA), cockpit display of traffic information (CDTI) and airborne weather avoidance. An efficient airspace representation is developed to reduce the search space in Free Flight concepts simulation. A conflict scenario generation methodology is developed to meet the need of rapid encounter scenario simulation in a highly controlled experimental environment. An integrated framework is developed to generate air traffic scenarios, evaluate CD algorithm and evolve further conflict scenarios using the feedback from the algorithm performance's metrics. Evolutionary computation techniques are used to evolve complex conflict scenarios and Pareto optimal front is used as a measure of algorithm performance with MD and FA as metrics.

The methodology is formulated in three stages, in the first stage CD algorithms are evaluated in a fast time air traffic simulation environment. This is done for a large number of increasingly complex conflict scenarios to gather data on algorithm performance vis-a-vis probe characteristics. In the second stage, a LCS is used to identify patterns in the data set and generate *if-then* rules to learn a concept description. The LCS based signature extraction system (UCSSE) adaptively identified and extracted the most generalized and accurate rules for each algorithm. In the final stage, the rules learned in the second stage are used to choose an algorithm from an ensemble of algorithms to assess a probe for potential conflicts. A switch mechanism is developed to route a conflict probe to an algorithm which is less prone to generate a MD or FA given probe characteristics.

The findings and summary of the work carried out in this thesis are as follows:

1. The proposed bin based airspace representation has a worst case complexity of O (M x N + 27 x N), where N is the number of aircraft and M is the number of active bins in airspace. The search complexity increases linearly with the increase in the number of aircraft.

2. In Free Flight weather avoidance, the proposed Ants-based heuristic search provides a methodology for incorporating multiple objectives and balances the virtues of exploration of environment and exploitation of information available about the airspace constraints. The quality of a solution is strongly affected by the exploitation of the embedded information in the environment rather than the available heuristics. High heuristic desirability leads to a poor quality of solution indicating early convergence to a sub-optimal solution due to more weight being placed on local information. Overall the best strategy was to exploit more of what is being learnt about the search space and uses less local information.

3. The proposed methodology of using evolutionary computation to evolve complex conflict scenarios drove each algorithm towards its vulnerable areas. The move towards higher complexity was depicted by the Pareto optimal curves in each generation of the evolutionary process. The evolutionary process did not suffer from early convergence and the flight density and conflict locations remained well distributed in the airspace. The evolved conflict characteristics gave indication as to which probe characteristics may lead to an algorithm failure.

4. A genetic-based machine learning technique is successfully employed to learn the implicit relationships in the probe characteristics where the three CD algorithms produced FA and MD. A small number of rules were obtained that captured the underlying concepts in the data with high accuracy, signifying an underlying rela-

tionship between the conflict characteristics and the categories of failure.

5. The ensemble approach reduced the number of MD and FA when compared to the individual algorithms performance. In the final generation, the maximum number of MD is only two and the maximum number of FA in one of the scenarios, reached eight.

8.2 Future Work

Various avenues of further research stem from the work carried out in this thesis. Some open research questions have already been highlighted in the respective chapters where they directly follow on from the work completed in the experiments. Here we outline more diverse future research directions.

The rapid prototyping environment of ATOMS can be used to model and evaluate other advanced ATM concepts. The environmental effect of Free Flight in terms of reduction in emission and noise foot prints for different air traffic procedures (continuous descent approaches) can be explored and validated. Investigations on “system” level properties of many aircraft flying in Free Flight with certain challenging constraints (weather, bottlenecks, flow or flow rate constraints, density constraints, complexity constraints, etc), would be another interesting avenue of research. Human in loop experiments for validation of the CDTI interface in ATOMS can provide an insight into the nature and amount of intent information needed by pilots for effective decision making in a constrained airspace. The bin based airspace design concept can also be extended to develop conflict resolution mechanisms.

In evolutionary scenario planning, the incorporation of several other probe characteristics (secondary conflict properties) while evaluating the CD algorithm may improve the performance of the machine learning algorithms. This may increase the accuracy of rule prediction and hence reduction in MD and FA when used in an ensemble. Another direction which will be interesting to pursue is, simultaneous evolution of conflict scenarios and algorithm parameters. Co-evolution of parameters and scenario characteristics can lead to real time optimization of GA parameters given probe characteristics, which may improve the performance of the individual algorithms and of the ensemble significantly.

In this thesis, only one ensemble approach is tested, other possible ensemble approaches of clubbing the two algorithms together or using majority-will-select criterion need to be explored to identify the best possible ensemble. Different parameter settings for CD algorithms can be explored which may impact the accuracy of the rule sets generated and the overall effectiveness of the ensemble approach.

In this work I have used classifier system (LCS) which is based on propositional logic thus can only capture simple relationship between scenarios and algorithms. By using representation such as First order Propositional logic may provide further details about the behavior of the chromosomes in the evolutionary process.

The methodology of evolving complex air traffic scenario can be extended for evaluation and validation of future as well as of existing ATM concepts. For example in airborne weather avoidance, weather patterns can be encoded into the chromosomes based data structure and can be evolved using GA to assess the performance of weather avoidance algorithms and to see how they perform as an ensemble.

8.3 Concluding Remarks

Performance evaluation of airborne CD algorithms is critical to the success of Free Flight. Given the inherent distributed nature and possibility of an infinite number of conflict geometries this will remain a challenging task. A significant progress has been made through numerous conflict scenario simulations and field trails. In this thesis, we have proposed the use of evolutionary scenario planning to this end. The evolution of air traffic scenarios towards higher complexity can open up different avenues in the Free Flight concept development and evaluation. We showed that scenario evolution can be carried out in a simple and practical manner using GAs. The highly accessible method of capturing scenario complexity has significant implications not only within the scope of Free Flight concepts but across a much wider spectrum of the ATM research field. This body of work should not be seen as merely an attempt to present a new methodology for the evaluation of CD algorithms, although it does give an insight into the performance of these algorithms, but as a new paradigm in which evolutionary computation can be used in a purposeful and powerful way in terms of evolving complex scenarios for the meaningful evaluation of advanced ATM concepts.

Appendix

The Australian Flight Information Region

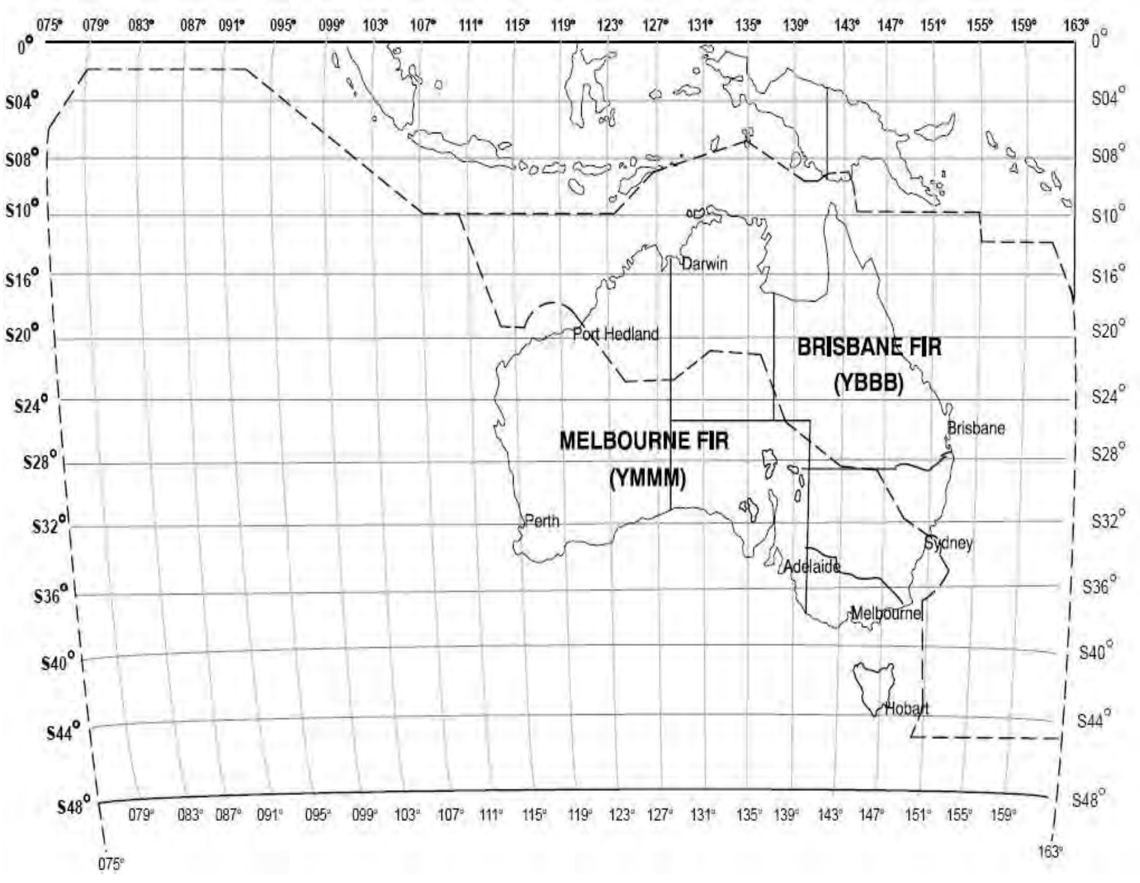


Figure 1: The Australian Flight Information Region

Bibliography

Aerosim (2005). Users guide, Aeronautical Simulation Block Set ver 1.1. http://www.u-dynamics.com, accessed 03/01/2008.

Air Service Australia (2006). Australian Organized Track Structure (AUSOTS). http://www.airservicesaustralia.com/customer/flextrack/default.asp, accessed 03/01/2008.

Air Services Australia (2004). *Designated Airsapce Handbook*. Air Services Australia, Canberra, Australia.

Alam, S., Lam, T. B., Abbass, H., and Barlow, M. (2006). Pareto meta-heuristics for generating safe flight trajectories under weather hazards. In *Proc. of the 6th Intl. Conf. on Simulated Evolution and Learning*, LNCS 4247, pages 829–836, Hefei, China. Springer-Verlag.

Alexander, A. L. and Wickens, C. D. (2001). Cockpit display of traffic information: The effects of traffic load, dimensionality, and vertical profile orientation. In *Proc. of the 45th Annual Meeting of the Human Factors and Ergonomics Society*, pages 68–72, Santa Monica, CA. Human Factors & Ergonomics Society.

Alliot, J., Durand, N., and Noailles, J. (1996). Automatic aircraft conflict resolution using genetic algorithms. In *Proc. of the 11th Annual ACM conference on Applied Computing*, pages 289–298. ACM Press.

Alliot, J., Bosc, J., Durand, N., and Maugis, L. (1997). CATS: A Complete Air Traffic Simulator. In *Proc. of 16th Digital Avionics Systems Conference*, Irvine, CA,.

Alliot, J., Durand, N., and Granger, G. (2000). FACES: a free flight autonomous and coordinated embarked solver. *Air Traffic Control Quarterly*, **8**(2), 109–130.

Andrews, J. W., Welch, J. D., and Erzberger, H. (2006). Safety analysis for advanced separation concepts. *Air Traffic Control Quarterly*, **14**(1), 5–24.

Battiste, V., Johnson, W. W., Kopardekar, P., Lozito, S., Mogford, R., and Palmer, E. A. (2002). Distributed Air/Ground Management - technology and concept demonstration report. In *Proc. of the AIAA Aircraft Technology, Integration, and Operations Technical Forum*, number AIAA-2002-5825, Los Angles, CA.

Bilimoria, K. (2000). A geometric optimization approach to aircraft conflict resolution. In *Proc. of the AIAA Guidance, Navigation, and Control Conference*, number AIAA-2000-4625, Denver, CO.

Bilimoria, K. (2001). A methodology for the performance evaluation of a conflict probe. *AIAA Journal of Guidance, Control, and Dynamics*, **24**(3), 444–451(8).

Bilimoria, K. and Lee, H. Q. (2001). Properties of air traffic conflicts for free and structured routing. In *Proc. of the AIAA Guidance, Navigation and Control Conf.*, number AIAA-2001-4051, Monteral, Canada.

Bilimoria, K., Lee, H. Q., Mao, Z.-H., and Feron, E. (2000). Comparison of centralized and decentralized conflict resolution strategies for multiple-aircraft problems. In *Proc. of the AIAA Guidance, Navigation, and Control Conference*, number AIAA-2000-4268, Denver, CO.

Bilimoria, K., Sridhar, B., Chatterji, G., Sheth, K. S., and Grabbe, S. (2001). FACET: Future ATM concepts evaluation tool. *Air Traffic Control Quarterly*, **9**(1), 203 229.

Blom, H., Bakker, G., Blanker, P., Daams, J., Everdij, M., and Klompstra, M. (2001). *Air Transportation Systems Engineering: Progress in Astronautics and Aeronautics*, volume 193, chapter Accident Risk Assessment for Advanced Air Traffic Management, pages 463–480. AIAA Press.

Blom, H., Bakker, G., Everdij, M., and Park, M. (2003). Collision risk modeling of air traffic. In *Proc. of the IFAC European Control Conference*, Cambridge, UK.

Bokadia, S. and Valasek, J. (2001). Severe weather avoidance using informed heuristic search. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, number AIAA-2001-4232, Montreal, Canada.

Brooker, P. (2002). Future air traffic management: Quantitative en route safety assessment part 1 review of present methods. *The Royal Institute of Navigation: Journal of Navigation*, **55**, 197–211.

Bugayevskiy, L. and Snyder, J. (1995). *Map Projections: A Reference manual*. Taylor & Francis, London.

Bureau of Meteorology (2003). *Manual of Aviation Meteorology*. AirServices Australia, Canberra, Australia.

Bussink, F., Hoekstra, J., and Heesbeen, B. (2005). Traffic manager: a flexible desktop simulation tool enabling future ATM research. In *Proc. of the 24th Digital Avionics Systems Conference*, volume 1, pages 3.B.4–31–10, Hampton, VA.

Butler, R., Carreno, V., Dowek, G., and Munoz, C. (2001). Formal verification of conflict detection algorithms. In T. Margaria and T. Melham, editors, *Proc. of the 11th Working Conference on Correct Hardware Design and Verification Methods*, number LNCS 2144, pages 403–417, Livingston, Scotland, UK. Springer-Verlag.

Cale, M. L., Paglione, M. M., Ryan, H. F., Oaks, R. D., and Summerill, J. S. (1998). Application of generic metrics to assess the accuracy of strategic conflict probes. In *Proc. of the 2nd USA/EUROPE ATM R&D Seminar*, Orlando, FL.

Callantine, T. J. (2004). Modeling and simulation of trajectory-oriented air traffic management concepts. In *Proc. of the AIAA Modeling and Simulation Conference and Exhibit*, number AIAA 2004-5260, Providence, Rhode Island.

Campos, L. M. B. C. and Marques, J. M. G. (2002). On safety metrics related to aircraft separation. *Journal of Navigation, The Royal Institute of Navigation*, **55**, 39–63.

Carlier, S., Lepinay, I., and Smith, J. (2004). Mediterranean Free Flight (MFF2) Project - Global Aviation Emission Study. Technical Report EEC/SEE/2004/006, Eurocontrol Experimental Centre, Bretigny Sur Orge, France.

Casti, J. (1986). *Complexity, Language and Life: Mathematical Approaches*, volume 6, chapter On System Complexity: Identification, measurement and management, pages 146 – 173. Springer-Verlag, Berlin.

Celio, J. C., Bowen, K. C., Winokur, D. J., Lindsay, K. S., Newberger, E. C., and Sicenavage, D. (2000). Free Flight Phase 1 Conflict Probe Operational Description. Technical Report MTR0W00000100, MITRE Corp, McLean, VA.

Chretien, G. (1974). The SITA Network. In R. L. Grimsdale and F. F. Kuo, editors, *Proc. of the NATO Advanced Study Institute on Computer Communication Networks*, Leyden, Holland. Noordhoff International Publishing Co.

Coello, C. and Lamont, G. (2004). *Applications of Multi-Objective Evolutionary Algorithms*. World Scientific, Singapore.

Coppenbarger, R. A., Kanning, G., and Salcido, R. (2001). Real-Time Data Link of Aircraft Parameters to the Center-TRACON Automation System (CTAS). In *Proc. of the 4th USA/Europe ATM R&D Seminar*, Santa Fe, NM.

DAG-TM (2002). DAG Concept Element 11 AATT Operational Concept For ATM. Technical Report Titan Systems Corporation - SRCD RTO-62:NAS2-98005, NASA Ames Research Center, Moffett Field, CA.

Das, R., Mitchell, M., and Crutchfield, J. (1994). A genetic algorithm discovers particle based computation in cellular automata. In *Parallel Problem Solving from Nature: PPSN III*, volume LNCS-866, pages 344 – 353. Springer-Verlag.

Datta, K. and Schultz, G. (1996). An evaluation of taam for free flight modelling:final report for advanced air transportation technology program. Technical Report NAS2-13767, NASA Ames Research Center, Moffett Field, CA.

Davidson, G., Krozel, J., Green, S., and Mueller, C. (2004). Strategic traffic flow management concept of operations. In *Proc. of the AIAA Aircraft tech, Integration and operation Conference*, Chicago, IL.

Davis, L. (1991). *Handbook of Genetic Algorithms*. Van Nostrand Reinhold.

Deb, K. (2001). *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons.

Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, **6**, 182–197.

Delahaye, D., Oussedik, S., and Schoenauer, M. (1999). Dynamic air traffic planning by genetic algorithms. In *Proc. of the IEEE Congress on Evolutionary Computation CEC-99*, volume 2, pages 11–17, Washington DC.

Donohue, G. and Laska, W. (2000). United States and European Airport Capacity Assessment using the GMU Macroscopic Capacity Model (MCM). In *Proc. of the 3rd USA/Europe ATM R&D Seminar*, Napoli, Italy.

Dorigo, M. and Stutzle, T. (2004). *Ant Colony Optimization*. MIT Press,, Cambridge, MA.

Dorneich, M., Olofinboba, O., Pratt, S., Osborne, D., Feyereisen, T., and Latorella, K. (2003). Evaluation of a dispatcher's route optimization decision aid to avoid aviation weather hazards. Technical Report NASA/TM-2003-212172, NASA Langley Research Center, Hampton, VA.

Dowek, G., Muoz, C., and Geser, A. (2001). Tactical conflict detection and resolution in a 3-d airspace. Technical Report NASA/CR-2001-210853, ICASE, NASA Langley Research Center, Hampton, VA.

Duong, V. N. and Hoffman, E. G. (1997). Conflict resolution advisory service in autonomous aircraft operations. In *Proc. 16th Digital Avionics Systems Conf.*, pages 9.3–10 – 9.3–17., Irvine, CA.

Eby, M. (1994). A self-organizational approach for resolving air traffic conflicts. *Lincoln Laboratory Journal*, **7**(2), 239 – 254.

Eby, M. and Kelly, W. (1999). Free flight separation assurance using distributed algorithms. In *Proc. of the IEEE Aerospace Conf.*, pages 429–441, Snowmass, CO.

Edmonds, B. (1999). *Syntactic Measures of Complexity*. Phd thesis, University of Manchester, England.

Ehrmanntraut, R. (2005). Performance parameters of speed control and the potential of lateral offset. Technical Report EEC Note No. 22/05, EUROCONTROL Experimental Centre, Brtigny-sur-Orge Cedex, FRANCE.

Erzberger, H. (2001). The Automated Airspace Concept. In *Proc. of the 4th USA/Europe ATM R&D Seminar*, Santa Fe, NM.

Erzberger, H. (2004). Transforming the NAS: The Next Generation Air Traffic Control System. Technical Report NASA/TP2004-212828, NASA Ames Research Center, Moffett Field, CA.

Erzberger, H. and Paielli, R. (1997). Conflict probability estimation for free flight. *AIAA Journal of Guidance, Control, and Dynamics*, **20**(3), 588–596.

Erzberger, H., Davis, T. J., and Green, S. (1993). Design of center-TRACON automation system. In *Proc. AGARD Guidance and Control Symp. Machine Intelligence in Air Traffic Management*, pages 11.1–11.12, Berlin, Germany.

Eshelby, M. (2000). *Aircraft Performance: Theory and Practice (AIAA Education Series)*. Butterworth-Heinemann.

Eurocontrol (2003a). ATM Strategy for the Years 2000+. Technical report, Eurocontrol, Brussels, Belgium.

Eurocontrol (2003b). Performance Review Report. Technical Report PRR2003, Eurocontrol, Brussels, Belgium.

Eurocontrol (2004). User manual for base of aircrfat data (BADA). Technical Report Rev No:3.6, Eurocontrol Experiment Center, Bretigny, France.

Eurocontrol (2005). Single European Sky ATM Research Programme: SESAR. www.eurocontrol.int.sesar/public/subsitehomepage/homepage.html accessed 03/01/2008.

Eurocontrol (2006). European air traffic situation. Press Bulletin: http://www.eurocontrol.int/corporate/gallery/content/ public/docs/pdf/pressreleases/2006/PressJuly2006Final.pdf accessed 03/01/2008.

Eurocontrol (2007). Performance Review Report for 2006. Technical Report PRR2006, Eurocontrol, Brussels, Belgium.

FAA (2002). National Airspace System Operational Evolution Plan (OEP): A foundation for capacity enhancement, version 5.0. Technical report, Federal Aviation Administration, Washington, DC.

FAA (2003). *Aeronautical Information Manual/Federal Aviation Regulation*. McGraw-Hill.

FAA (2008). Airborne supplemental navigation equipment using the global positioning system (gps). Technical Report TSO-C-129(a), Aircraft Certification Service, Department of Transportation, Federal Aviation Administration.

Fishbein, S. B. (1995). *Flight Management Systems: The Evolution of Avionics and Navigation Technology*. Praeger Publishers.

Ford, R. (1986). The conflict resolution process for TCAS-II and some simulation results. *Journal of Navigation, The Royal Institute of Navigation*, **40**, 283 – 303.

Foyle, D., Goodman, A., and Hooey, B. (2003). An Overview of the NASA Aviation Safety Program (AvSP) System-Wide Accident Prevention (SWAP) Human Performance Modeling (HPM) Element. In D. Foyle, A. Goodman, and B. Hooey, editors, *Proc. of the 2003 Conference on Human Performance Modeling of Approach and Landing with Augmented Displays*, number NASA/CP-2003-212267, pages 1–13, Moffett Field, CA.

French, A. P. (1971). *Newtonian Mechanics*. W. W. Norton, New York.

Gazit, R. (1996). *Aircraft Surveillance and Collision Avoidance Using GPS*. Ph.D. thesis, Stanford University, Stanford, CA.

Gentle, J. E. (2003). *Random Number Generation and Monte Carlo Methods*. Springer, New York, 2 edition.

Goldberg, D. (1989). *Genetic algorithms in search, optimization and machine learning*. Addison-Wesley Longman Pub. Co, Boston, MA, 1 edition.

Gool, M. and Schrter, H. (1999). PHARE Final Report. Technical Report DOC 99-70-09, Eurocontrol, Brussels, Belgium.

Gredes, I. (1997). Construction of conflict-free routes for aircraft in case of free-routing with genetic algorithms. In *Proc. of 1st USA/Europe ATM R&D Seminar*, Saclay, France.

Grewal, M., Weill, L., and Andrews, A. (2002). *Global Positioning Systems, Inertial Navigation, and Integration*. John Wiley & Sons, Inc.

Han, J. and Kamber, M. (2001). *Data mining: Concepts and Techniques*. Morgan Kaufmann.

Harvey, A., Mullan, C., Schwartz, A., and Magyarits, S. (2003). FAA/Eurocontrol Cooperative R&D: Action Plan 5, Operational Concept Validation, Appendix A. In *Joint Workshop Eurocontrol Research Center and FAA*, Rome, Italy.

Hawthorne, M., Skipper, S., and Hancock, T. (1999). CPDLC: A data communications evolution for ATM. *Journal of ATC*, pages 16–19.

Heagy, W. and Kirk, D. (2003). Description of URET enhancements to support severe weather avoidance. Technical report, The MITRE Corporation Center for Advanced Aviation System Development (CAASD), McLean, Virginia.

Hoekstra, J. (2001). Designing for safety the free flight air traffic management concept. Technical Report NLR-TP-2001-313, National Aerospace Laboratory(NLR), Amsterdam, Netherlands.

Hoekstra, J., Gent, R. v., and Ruigrok, R. (1998). Conceptual design of free flight with airborne separation assurance. In *Proc. AIAA Guidance, Navigation, and Control Conference*, number AIAA-98-4239, pages 807–817, Boston, MA.

Hoekstra, J., Gent, R., and Ruigrok, R. (2002). Designing for safety: the free flight air traffic management concept. *Reliability Engineering & System Safety*, **75**(2), 215–232.

Holland, J. (1986). *Machine learning: An artificial intelligence approach*, volume 2, chapter Escaping brittleness: The possibilities of general purpose learning algorithms applied to parallel rule-based systems., pages 229–257. Morgan Kaufmann, San Mateo, CA.

Holland, J., Booker, L., Colombetti, M., Dorigo, M., Goldberg, D., Forrest, S., Riolo, R., Smith, R., Lanzi, P., Stolzmann, W., *et al.* (2000). What is a learning classifier system. *Learning Classifier Systems: From Foundations to Applications*, **1813**, 3–32.

Holland, J. M. (1975). *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, MI.

ICAO (1996). Doc 4444-RAC/501 : Procedures for Air Navigation Services. Technical report, ICAO, Montreal, Canada.

ICAO (1998). Annex 11 air traffic services. Technical Report 12th edition, Monteral, Canada.

ICAO (2001). ANNEX 10 to the Convention on International Civil Aviation Aeronautical Telecommunications. Technical Report Volumes II and III, ICAO, Montreal, Canada.

ICAO (2002). *The ICAO Global air Navigation Plan for CNS/ATM systems*, volume 1. ICAO, Montreal, Canada.

ICAO (2004). *Manual of the ICAO Standard Atmosphere extended to 80 kilometers*. ICAO, Montreal, Canada, third edition.

ICAO (2007a). ICAO DOC 9613 Manual of Performance Based Navigation (PBN). Technical report, ICAO, Montreal, Canada.

ICAO (2007b). *Manual of Air Traffic Services Data Link Applications*. ICAO, Montreal, Canada, doc 9694-an/955 edition.

Isaacson, D. and Erzberger, H. (1997). Design of a Conflict Detection Algorithm for the Center/TRACON Automation System. In *Proc. of the 16th Digital Avionics Systems Conf.*, pages 9.3–1 – 9.3–9, Irvine,CA.

Jardin, M. (2005). Grid-based strategic air traffic conflict detection. In *Proc. of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, number AIAA-2005-5826, San Francisco, California.

Jelinek, F. (2006). Findings of the Validation of the Advanced Emission Model (AEM-III) version 1.5. Technical report, Society, Environment, Economy Research Area, Eurocontrol Experimental Center, France.

Johnson, W. W., Battiste, V., and Bochow, S. H. (1999). A Cockpit Display Designed to Enable Limited Flight Deck Separation Responsibility. In *Proc. of the American Society of Mechanical Engineers Wind Energy Symposium : Aerospace Sciences Meeting and Exhibit*, number AIAA-99-5567, Reno, NV.

Kopardekar, P. ., Sacco, N. ., and Mogford, R. (2002). Comparison of air and ground conflict detection and resolution algorithms and their implications. In *Proc. 21st AIAA/IEEE Digital Avionics Systems Conf.*, volume 1, pages 2B1–1– 2B1–8, Irvine, CA.

Kopardekar, P. and Magyarits, S. (2002). Dynamic density: measuring and predicting sector complexity. In *Proc. 21st AIAA Digital Avionics Systems Conference*, pages 2C4–1 – 2C4–9, Irvine, CA. AIAA Press.

Koza, J. (1994). *Genetic Programming II: Automatic Discovery of Reusable Programs*. MIT Press.

Krozel, J. (2000). Free flight research issues and literature search. Technical Report NAS2-98005, NASA Ames Research Center, Moffett Field, CA.

Krozel, J., Peters, M., Bilimoria, K., Lee, C., and Mitchell, J. (2001). System performance characteristics of centralized and decentralized air traffic separation strategies. In *Proc. of the 4th USA/Europe ATM R&D Seminar*, Santa Fe, USA.

Krozel, J., Andre, A., and Smith, S. (2003). The future national airspace system: Design requirements imposed by weather constraints. In *Proc. of the AIAA Guidance, Navigation, and Control (GNC) Conference, 2003*, number AIAA-2003-5769, Austin, TX.

Krozel, J., Penny, S., Prete, J., and Mitchell, J. (2004). Comparison of algorithms for synthesizing weather avoidance routes in transition airspace. In *Proc. of the AIAA Guidance, Navigation, and Control Conference, 2004*, number AIAA-2004-4790, Providence, RI.

Krozel, J., Andre, A., and Smith, P. (2006). Future air traffic management requirements for dynamic weather avoidance routing. In *25th Digital Avionics Systems Conference, 2006 IEEE/AIAA*, pages 1–9, Portland, OR.

Kuchar, J. (1995). *A unified methodology for the evaluation of hazard alerting systems.* Phd thesis, MIT.

Kuchar, J. and Yang, L. (2000). Survey of conflict detection and resolution modelling methods. *IEEE Transactions on Intelligent Transportation System*, **1**(4), 179–189.

Lee, P., Mercer, J., Martin, L., Prevot, T., Shelden, S., Verma, S., Smith, N., Battiste, V., Johnson, W., Mogford, R., and Palmer, E. (2003). Free maneuvering, trajectory negotiation, and self-spacing concepts in distributed air-ground traffic management. In *In Proc. 5th USA/Europe ATM R&D Seminar*, Budapest, Hungary.

Lee, P. U., D'Arcy, J. F., Mafera, P., Smith, N., V., B., Johnson, W., Mercer, J., Palmer, E. A., and Prevot, T. (2004). Trajectory negotiation via data link: Evaluation of human-in-the-loop simulation. In *Proc. of the International Conference on Human-Computer Interaction in Aeronautics*, pages 149–154, Toulouse, France. HCI-Aero 2004.

Lumsden, J. (2001). European Aeronautics: A vision for 2020. Technical report, Report of the Group of Personalities, Luxembourg.

Majumdar, A., Ochieng, W., and Polak, J. (2002). Estimation of european airspace capacity from a model of controller workload. *The Journal Of Navigation, The Royal Institute of Navigation*, **55**, 381–403.

McNally, B., Bach, R., and Chan, W. (1999). Field test evaluation of the CTAS conflict prediction and trial planning capabilities. In *Proc. AIAA Guidance, Navigation, and Control Conf.*, number AIAA-1998-4480, pages 1686–1697, Boston MA.

Michalewiz, Z. (1992). *Genetic Algorithms + Data Structures = Evolution Programs*. Springer Verlag.

Michalski, R., Bratko, I., and Bratko, A., editors (1998). *Machine Learning and Data Mining; Methods and Applications*. John Wiley & Sons, Inc., New York, NY, USA.

Miles, E., Davis, P., and Wing, D. (1999). Development of a free-flight simulation infrastructure. In *Proc. of the 22nd Digital Avionics Systems Conference*, number AIAA-99-4193, Indianapolis, IN.

Mitchell, M. (1996). *An Introduction to Genetic Algorithms*. MIT Press.

MITRE (2002). Safeflight 21 working group, safe flight 21 ver 3.0. Technical report, Mitre Corporation.

MITRE (2007). User request evaluation tool. http://www.mitrecaasd.org/work/projectdetails.cfm/ite accessed 03/01/2008.

NASA (1999). NASA Advanced Air Transportation Technologies Project: Concept Definition for Distributed Air/Ground Traffic Management (DAG-TM). Technical report, Advanced Air Transportation Technologies (AATT) Office NASA Ames Research Center, Moffett Field, CA.

National Climatic Data Center (2003). World Area Forecast System WAFS, DS-9939, NESDIS, NOAA, U.S. Department of Commerce. http://www.ncdc.noaa.gov, accessed:04/01/2008.

National Research Council (2003). Weather forecasting accuracy for FAA traffic flow management. Technical report, National Academic Press, Washington DC.

Niedringhaus, W. (1998). Solution complexity metrics for evaluation of dense air traffic control scenarios. In *Proc. AIAA Guidance, Navigation, and Control Conference*, number AIAA-1998-4104, pages 30–51, Boston, MA.

Nilim, A., Ghaoui, L., and Duong, V. (2004). Algorithms for multi-aircraft routing under uncertainty. In *The 2004 International Conference of Recherche Informatique Vietnam-Francophone*, pages 21–32, Hanoi, Vietnam.

Nolan, M. (2004). *Fundamentals of Air Traffic Contro*. Brooks/Cole-Thomson Learning, Belmont,CA, fourth edition.

NTSB (1991-2001). Aviation Statistical Reports. Technical report, National Transport Safety Board, Washington D.C.

OAG (2001). *Official Airline Guide*. OAG Worldwide Limited, Bedfordshire, LU5 4HB, United Kingdom.

Ochieng, W., Sauer, K., Cross, P., and Sheridan, K. (2001). Potential Performance Levels of a Combined Galileo/GPS Navigation System. *Journal of Navigation*, **54**(2), 185–197.

Odoni, A., Bowman, J., Delahaye, D., Deyst, J., Feron, E., J.R., H., Khan, K., Kuchar, K., Pijet, N., and Simpson, R. (1997). Existing and required modeling capabilities for evaluating ATM systems and concepts, final report to the NASA Advanced Air Transportation Technology project. Technical Report NASA-CR-204978, International Center for Air Transportation, Massachusetts Institute of Technology, Cambridge, Massachusetts.

Paglione, M. M., Oaks, R. D., and Bilimoria, K. (2003). Methodology for generating conflict scenarios by time shifting recorded traffic data. In *Proc. of the AIAA 3rd Annual Aviation Technology, Integration and Operations (ATIO) Conference*, number AIAA-2003-6724, Denver, CO.

Paielli, R. (1998). Empirical test of conflict probability estimation. In *Proc. of the 2nd USA/Europe ATM R&D Seminar*, Orlando, FL.

Pang-Ning, T., Michael, S., and Vipin, K. (2005). *Introduction to Data Mining*. Addison Wesley.

Pannequin, J. J., Bayen, A., Mitchell, I., Chung, H., and Sastry, S. (2007). Multiple aircraft de-conflicted path planning with weather avoidance constraints. In *Proc. of the*

AIAA Guidance, Navigation and Control Conference, number AIAA-2007-6588, Hilton Head, SC.

Peter, F. (1995). *Aviation Weather*. Jepesson Sanderson, Inc.

Peters, M. (1998). The Detailed Design and Software Implementation of the Aircraft Dynamics Model for the FAA Target Generation Facility. Technical report, Seagull Technology Inc., Los Gatos, CA.

Planning, J. and Office, D. (2007). Concept of operations for the next generation air transportation system ver 2.0.

Prandini, M. and Watkins, O. (2005). Probabilistic Aircraft Conflict Detection , Distributed Control and Stochastic Analysis of Hybrid Systems Supporting Safety Critical Real-Time Systems Design (HYBRIDGE). Technical Report IST-2001-32460 of European Commission.

Prandini, M., Hu, J., Lygeros, J., and Sastry, S. (2000). A probabilistic approach to aircraft conflict detection. *IEEE Transactions on Intelligent Transportation System*, **1**(4).

Prete, J. and Mitchell, J. (2004). Safe routing of multiple aircraft flows in the presence of time-varying weather data. In *Proc. of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, number AIAA-2004-4791, Province, Rhode Island.

Prevot, T., Lee, P., Smith, N., and Palmer, E. (2005). ATC technologies for controller-managed and autonomous flight operations. In *Proc. of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, number AIAA-2005-6043, San Francisco, CA.

RAMS (2008). Recognized atc mathematical simulator (rams). http://http://www.ramsplus.com/, accessed 03/02/2008.

Rand, T. and Eby, M. (2004). Algorithms for airborne conflict detection, prevention, and resolution. In *Proc. of the 23rd Digital Avionics Systems Conference*, volume 1, pages 17–31, Salt Lake City, UT.

Ratcliffe, S. (2001). Free Flight in Europe, Problem, and Solutions. *The Journal of Navigation*, **52**(2), 213–221.

Raytheon (2004). Air traffic management system development and integration (ATMSDI) acquisition. CTOD 16ASTOR Software Design Document, Prepared by Raytheon Company for NASA Ames.

RTCA (1995). Report of the RTCA board of director's select committee on Free Flight. Technical report, RTCA Inc., Washington, DC.

RTCA (1998). Minimum aviation system performance standards for automatic dependent surveillance broadcast (ADS-B). Technical Report RTCAAIO-242, Special Committee 186 (SC-186) RTCA Inc., Washington, DC.

RTCA (1999). Development and Implementation Planning Guide for Automatic Dependent Surveillance Broadcast (ADS-B) Applications. Technical Report RTCA/DO-249, Washington, DC.

RTCA (2000). Application descriptions for initial cockpit display of traffic information (CDTI) applications. Technical Report RTCA/DO-242, RTCA Inc., Washington, DC.

RTCA (2003). Minimum Operational Performance Standards for 1090 MHz Automatic Dependent SurveillanceBroadcast (ADS-B). Technical report, RTCA Inc., Washington, DC.

RTCA (2007). Future Air Navigation System 1/A (FANS 1/A) - Aeronautical Telecommunications Network (ATN) Interoperability Standard. Technical Report RTCA SC-189 DO-305, RTCA Inc., Washington D.C.

Russell, S. and Norvig, N. (1995). *Artificial Intelligence A Modern Approach*. Prentice Hall.

Scardina, J. (2002). The approach and basis for the FAA ADS-B link decision. Technical report, FAA, Washington, DC.

Sedgewick, R. (1998). *Algorithms in C++ Parts 1-4: Fundamentals, Data Structure, Sorting, Searching,*. Addison Wesley Professional, 3rd edition.

SESAR Consortium (2007). The atm target concept: Sesar definition phase, deliverable 3. Technical Report DLM-0612-001-02-00, Eurocontrol, Brussels.

Shafi, K., Abbass, H., and Zhu, W. (2007). Real Time Signature Extraction During Adaptive Rule Discovery Using UCS. In *Proc. of the IEEE Congress on Evolutionary Computation*, Singapore. IEEE Press.

Shakarian, A. and Haraldsdottir, A. (2001). Required total system performance and results of a short term conflict alert simulation study. In *Proc. of the 4th US/Europe ATM R&D Seminar*, Santa Fe, NM.

Shridhar, B., Seth, K. S., and Grabbe, S. (1998). Airspace complexity and its application in air traffic management. In *Proc. of the 2nd USA/Europe ATM R&D Seminar*, Orlando, FL.

Signor, D., Davis, P., Lozito, S., Andre, A., Sweet, D., and Wallace, E. (2004). Efficient air traffic scenario generation. In *AIAA 4th Aviation Technology, Integration and Operations (ATIO) Forum*, number AIAA-2004-6399, Chicago, IL.

Smith, N., Polson, P., Brown, J. A., and Moses, J. (2001). An assessment of flight crew experiences with fans-1 controller-pilot data link communication in the south pacific. In *Proc. of the 4th USA/Europe ATM R&D Seminar*, Santa Fe, NM.

Smith, S. F. (1980). *A Learning System Based on Genetic Adaptive Algorithms*. Phdthesis, University of Pittsburgh.

Snyder, K., Sokoloff, S., and Bearden, M. (2003). The cockpit associate: A "co-pilot in a box" for general aviation. In *Proc. of the 22nd Digital Avionics Systems Conference*, volume 2, pages 12.C.3– 121–12. IEEE Press.

Sridhar, B. and Chatterji, G. B. (1997). Computationally efficient conflict detection methods for air traffic management. In *Proc. of the American Control Conference (AACC)*, number 0-7803-3832-4/97, Albuquerque, NM.

Stephane, M. and Sheila, C. (2001). An airborne conflict resolution approach using a genetic algorithm. In *Proc. of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, number AIAA-2001-4054, Montreal, Canada.

Stutzle, T. and Holger, H. H. (2000). MAX - MIN Ant System. *Future Generation Computer Systems*, **16**, 889–914.

Suchkov, A. (2001). Strategic RVSM or conflict resolution utilization of additional reduced vertical separation minima (RVSM) flight levels as interim alternative to domestic RVSM implementation. Technical report, CSSI, Inc., Washington, DC.

Swenson, H. (2005). The Virtual Airspace Modeling and Simulation (VAMS) Project, "http://www.vams.arc.nasa.gov/"accessed 05/01/2008.

Teo, J. and Abbass, H. (2005). Multi-objectivity and complexity in embodied cognition. *IEEE Transactions on Evolutionary Computation*, **9**(4), 337–360.

Teodorovic, D. (2003). Transport modeling by multi-agent systems: a swarm intelligence approach. *Transportation Planning and Technology*, **26**(4), 289–312.

Transport Canada (2006). Flight 2010: A strategic plan for civil aviation. Technical Report TP 14469, Minister of Transport, Ottawa, Canada.

Tu, J. (1996). Cockpit weather information (CWIN) system. In *Proc. of the IEEE 1996 National Aerospace and Electronics Conference*, volume 1, pages 29–32, Dayton, OH.

Vink, A., Kauppinen, S., Beers, J., and Jong, K. (1997). Medium term conflict detection in EATCHIP Phase III. In *Proc. of the 16th AIAA/IEEE Digital Avionics Systems Conference*, Irvine, CA.

Warren, A., Schwab, R., Geels, T., and Shakarian, A. (1997). Conflict probe concept analysis in support of free flight. Technical Report NASA CR-201623, NASA Langley Research Center, Hampton, VA.

Watkins, O. and Lygeros, J. (2002). Safety relevant operational cases in Air Traffic Management, Distributed Control and Stochastic Analysis of Hybrid Systems Supporting Safety Critical Real-Time Systems Design (HYBRIDGE). Technical Report IST-2001-32460 of European Commission, University of Cambridge, UK.

Whitely, L. (1993). *Foundation of Genetic Algorithm 2*. Morgan Kaufmann.

Whittle, J., Saboo, J., and Kwan, R. (2003). From scenarios to code: An air traffic control case study. In *Proc. of the 25th International Conference on Software Engineering, ICSE-03*, Portland, OR. IEEE Computer Society.

Witten, I. H. and Frank, E. (2000). *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann.

Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. M., and da Fonseca, V. G. (2003). Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Trans. Evol. Comput.*, **7**(2), 174–188.