



The
University
Of
Sheffield.

Access to Electronic Thesis

Author: George W. Flathers III
Thesis title: An Airspace Simulator for Separation Management Research
Qualification: MPhil

This electronic thesis is protected by the Copyright, Designs and Patents Act 1988. No reproduction is permitted without consent of the author. It is also protected by the Creative Commons Licence allowing Attributions-Non-commercial-No derivatives.

If this electronic thesis has been edited by the author it will be indicated as such on the title page and in the text.

An Airspace Simulator for Separation Management Research

George W. Flathers III

April, 2011

Master of Philosophy Thesis

The University of Sheffield
Department of Automatic Control and
Systems Engineering

[Page intentionally left blank]

[Page intentionally left blank]

Abstract

Air Traffic Management (ATM) systems are undergoing a period of major transformation and modernisation, requiring and enabling new separation management (SM) methods. Many novel SM functions, roles and concepts are being explored using ATM simulators. Commercial simulators are capable, high-fidelity tools, but tend to be complex and inaccessible. The Airspace Simulator is a fast-time, discrete event simulator originally designed for exploratory ATM research. This thesis describes the redevelopment of the Airspace Simulator into a simulation platform better suited for researching and evaluating SM in future airspace. The Airspace Simulator-II has the advantage of new functionality and greater fidelity, while remaining high-speed, accessible and readily adaptable.

The simulator models FMS-like spherical earth navigation and autopilot flight control with an average cross track error of 0.05 nmi for waypoint-defined routes in variable wind-fields. Trajectories are computed using the BADA v3.8 tabulated database to model the performance of 318 aircraft types. The simulator was demonstrated with up to 4000 total aircraft, and trajectories for 300 simultaneous aircraft were computed over 900 times faster than real-time.

Datalink and radio-telephony communications are modelled between the air traffic and ATM systems. Surveillance is provided through ADS-B-like broadcasts, and an algorithm was developed to automatically merge instructions from conflict resolution systems with existing flight plans. Alternate communication, navigation, and separation modes were designed to permit the study of mixed-mode operations. Errors due to wind, navigational wander, communication latencies, and localised information states are modelled to facilitate research into the robustness of SM systems.

The simulator incorporates a traffic visualisation tool and was networked to conflict detection and resolution software through a TCP/IP connection. A scenario generator was designed to automatically prepare flight plans for a large variety of two-aircraft encounters to support stochastic SM experiments. The simulator, scenario generator, and resolver were used for the preliminary analysis of a novel concept for automated SM over radio-telephony using progressive track angle vectoring.

[Page intentionally left blank]

Acknowledgments

I would like to thank the many individuals who supported me and made this work possible. In particular, I wish to acknowledge Professor D.J. Allerton, Dr. Graham Spence, and Bill Flathers; the expertise they provided and the advice they gave was invaluable. Karen Feigh provided the foundation for this research, and I am glad she has allowed me to extend what she began. I am also grateful for the generosity of the Marshall Aid Commemoration Commission, the oversight of the Air Force Institute of Technology – Civilian Institutions Program, and the support of my family and The Crowded House – Refuge GC. Rah VA Mil!

Disclaimer

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the U.S. Government.

[Page intentionally left blank]

Table of Contents

Abstract	iv
Acknowledgments	vi
Disclaimer	vi
Table of Contents	viii
List of Tables	xiii
List of Figures	xiv
Notation	xvi
CHAPTER 1: Background and Motivation	1
1.1 Air Traffic Management	1
1.2 Air Traffic Management Modernisation	2
1.2.1 The Need and Direction of Modernisation	2
1.2.2 Trajectory Based Operations	3
1.2.3 Communication	5
1.2.4 Surveillance	6
1.3 Separation Management	6
1.3.1 The Current Separation Management Process	7
1.3.2 New Concepts for Separation Management	8
1.3.2.1 Automated Conflict Detection and Resolution Systems	8
1.3.2.2 Levels of Automation	9
1.3.2.3 Delegated Separation Responsibility	11
1.3.2.4 Mixed-Equipment Operations	12
1.4 Simulators for Separation Management Research	14
1.4.1 Commercial Simulation Platforms	15
1.4.1.1 TAAM	16
1.4.1.2 RAMS Plus	16
1.4.1.3 FACET	17
1.4.1.4 Limitations	18
1.4.2 Non-Commercial Simulation Platforms	19
1.4.3 The Airspace Simulator by K. Feigh	20
1.4.3.1 Case for Improving the Airspace Simulator	21
1.5 Research Aims	23
1.6 Organisation of Remaining Chapters	24
CHAPTER 2: Simulator Requirements and Design	25

2.1 Simulation Requirements	25
2.1.1 Capabilities	25
2.1.2 Constraints	27
2.2 Programming Language and Operating System	28
2.3 Simulator Organisation	29
2.3.1 Scenario Generation Module	30
2.3.2 Air Traffic Module	31
2.3.3 Navigation and Trajectory Module	32
2.3.4 Wind Field Module	32
2.3.5 Communications Module	32
2.4 Operational Data Structures	33
2.4.1 The Master Array	33
2.4.2 The Aircraft Array	34
2.4.3 The Data Block	36
2.5 Logical Design	37
2.6 Synchronisation with External Systems	40
2.7 Simulator Outputs	44
2.7.1 Permanent Record	44
2.7.2 Traffic Visualisation	46
2.8 Summary of Changes Made to the Airspace Simulator	47
CHAPTER 3: Navigation and Trajectory Module	49
3.1 Requirements for Modelling Navigation and Trajectories	49
3.2 Trajectory Modelling Overview	51
3.2.1 Flight Plans	53
3.2.2 ATC Instructions	54
3.3 Performance Modelling	55
3.4 The Flight Management System	56
3.4.1 Lateral Guidance	56
3.4.1.1 Great Circle Navigation	57
3.4.1.2 Turn Anticipation	59
3.4.1.3 Lateral Offsets	60
3.4.2 Longitudinal Guidance	61
3.4.3 Vertical Guidance	62
3.5 The Autopilot/Flight Director System	62
3.5.1 Lateral Control	62
3.5.2 Longitudinal Control	64

3.5.3 Vertical Control	64
3.6 Equations of Motion	66
3.7 Navigation Error and Uncertainty Modelling	67
3.7.1 Flight Technical Error	67
3.7.2 State Estimation Noise	68
3.8 Verification and Evaluation of the Module	69
3.8.1 Verification of the BADA v3.8 Database Implementation	69
3.8.2 Evaluation of the Flight Control System	70
3.8.3 Evaluation of the Flight Management System	73
3.8.3.1 Evaluation of Navigation Accuracy	74
3.8.3.2 Verification of Flight Plan and ATC Instruction Following	75
3.8.4 Verification of the Flight Technical Error Model	79
3.8.5 Evaluation of Module Execution Speed	80
3.9 Summary	81
CHAPTER 4: Communications Module	82
4.1 Requirements for Modelling Communications	83
4.2 The Communications Process	84
4.2.1 The Message Set	85
4.2.2 The Message Queue	86
4.2.3 Incorporating Flight Plan Modifications	87
4.3 Latency Modelling	91
4.3.1 Datalink Latencies	92
4.3.2 Radio-Telephone Latencies	93
4.3.3 ASAS Resolutions	96
4.4 Surveillance Broadcasting	96
4.5 Evaluation of the Module	98
4.5.1 Verification of Surveillance Broadcast Failure Model	98
4.5.2 Verification of Latency Modelling	98
4.5.3 Verification of Trajectory Exchange	100
4.6 Summary	102
CHAPTER 5: Scenario Generation	103
5.1 Motivation	103
5.2 Pair-Wise Conflict Scenario Generator	105
5.2.1 Aircraft Type Selection	106
5.2.2 Mode Selection	107
5.2.3 Flight Plan Generation	108

5.2.4 Start Time Control	113
5.3 Verification and Discussion of Scenario Generator	113
5.3.1 Distribution Goodness-of-Fit	114
5.3.2 Scenario Geometry	114
5.3.3 Fast-Time Performance	117
5.3.4 Discussion	117
5.4 Verification of Fast-Time Performance	118
5.5 Summary	120
CHAPTER 6: Research Application – Vector Navigation	122
6.1 Automation Support for Mixed-Equipage Traffic	122
6.1.1 The Need for Improved Automation Support	123
6.2 Design of an RMAT System	127
6.2.1 Initial Processing Module	130
6.2.2 Resolution List Manager	132
6.2.3 Manoeuvre Modelling	134
6.2.3.1 Placement of the Alert Trigger	134
6.2.3.2 Placement of the Transmission Trigger	135
6.2.4 Implementation in the Airspace Simulator II	137
6.3 Initial Evaluation of RMAT	138
6.3.1 Comparison of RMAT to Datalink	138
6.3.2 Effects of Communication Timing Uncertainty on CPA	141
6.4 Chapter Summary	145
CHAPTER 7: Discussion and Conclusion	147
7.1 Discussion	147
7.1.1 Execution Speed	147
7.1.2 Fidelity	149
7.1.3 Functionality	150
7.2 Conclusion	151
7.2.1 Satisfaction of Requirements	152
7.2.2 Suggestions for Further Work on the Airspace Simulator – II	156
7.2.2.1 Expanding the FMS Functionality	156
7.2.2.2 Expanding the Scenario Generation Method	156
7.2.2.3 Incorporating a Native CD&R System	157
7.2.3 Suggestions for Further Work on Automated Vectoring	157
Glossary	159
Appendix A: Simulator Configuration File	161

Appendix B: Output File Format	164
Appendix C: Flight Plan Format	167
Appendix D: Scenario Generator Configuration File	169
Bibliography	171

List of Tables

2-1: Summary of requirements.....	28
2-2: The Master Array.....	34
2-3: The Aircraft Array.....	36
3-1: Flight technical error parameters.....	68
3-2: Manoeuvre durations (seconds).....	70
3-3: Comparison of flight technical error parameters.....	80
4-1: The surveillance output data set.....	86
4-2: EUROCAE Effective Update Period performance requirements.....	97
4-3: Minimum allowable single message probabilities of reception.....	97
4-4: Sampled latency distributions.....	99
5-1: The ten most frequent aircraft types in European airspace.....	107
5-2: Distribution of encounter angles.....	110
5-3: Chi-Squared goodness-of-fit test results.....	114
5-4: Regression results of t_{LOS} standard deviation.....	116
5-5: Flight summaries for concurrent aircraft test.....	118
5-6: Concurrent aircraft test results.....	119
6-1: Linear regression results on datalink and RMAAT CPA.....	141
6-2: Radio-telephony latency settings and the resulting average error.....	142
B-1: Header data fields.....	164
B-2: Master Array data fields.....	166
B-3: Traffic state data fields.....	166
C-1: Leg segment data fields.....	168

List of Figures

2-1: Simulator organisation	30
2-2: Relationship between the Master and Aircraft Arrays.....	35
2-3: The simulator logical design	37
2-4: The setup phase.....	37
2-5: The outer simulation loop	38
2-6: The inner simulation loop.....	39
2-7: The shutdown phase.....	40
2-8: Speed-controlled synchronisation mode	41
2-9: The handshake synchronisation mode.....	43
2-10: Log-log plot of permanent record file size	45
2-11: Log-log time profile of data recording.....	45
2-12: TViz Screenshot.....	47
3-1: Illustration of lateral navigational accuracy bounds	49
3-2: Direct-to manoeuvre	50
3-3: Lateral offset manoeuvre	51
3-4: Path stretch manoeuvre	51
3-5: The navigation and trajectory modelling process	52
3-6: Fly-by waypoint (A), and fly-over waypoint (B)	53
3-7: Linked list data structure	54
3-8: Great Circle navigation geometry.....	58
3-9: Constant bank turn	59
3-10: Turn correction for lateral offsets	61
3-11: The lateral flight controller	63
3-12: Longitudinal flight controller	64
3-13: The vertical flight controller	65
3-14: Lateral controller response to 90 degree heading change	71
3-15: Vertical controller response to a 2000 ft climb	72
3-16: Longitudinal controller response to a 30 knot speed change	73
3-17: Sample points in navigation test	74
3-18: Radar plot of navigational accuracy	75
3-19: Test route.....	76
3-20: Segment transition test	76
3-21: Route modification test	77
3-22: Heading, track angle, and direct-to test	77
3-23: Lateral offset test.....	78
3-24: Test of timed altitude (A) and speed (B) instructions	78
3-25: Time series of flight technical error.....	79
3-26: Breakdown of simulator execution time	80
4-1: The communications input process.....	84
4-2: The communications output process	85
4-3: Data structure of message queue and type definitions	87

4-4: Scenario requiring merging of resolution waypoints	88
4-5: Resulting ground tracks of two incorrect merges	89
4-6: Wayline for fly-by and for fly-over transitions	89
4-7: An automated algorithm for merging route modifications.....	90
4-8: Resulting ground tracks after merge	91
4-9: Pseudo-code model of transmission delay.....	95
4-10: Transmission delays due to channel occupation.....	95
4-11: The effect of radio-telephony latency	100
4-12: TViz screenshot before and after conflict resolution	101
5-1: A pair-wise conflict scenario at initialisation	106
5-2: A pair-wise conflict scenario at initialisation	108
5-3: Velocity triangle geometry	111
5-4: A pair-wise conflict scenario at initialisation	112
5-5: A screenshot of a pair-wise conflict scenario at tLOS.	114
5-6: Scatter plots of tLOS verses encounter angle and speed differences.....	115
5-7: CPA histogram	115
5-8: Execution time per concurrent aircraft	119
5-9: Fast-time gain per concurrent aircraft	120
6-1: Mixed-equipage operations using an RMAT system	127
6-2: RMAT system architecture	129
6-3: Initial Processing Module control flow	130
6-4: Location of alert and transmission trigger points	132
6-5: Resolution List Manager control flow	133
6-6: Placement of the alert trigger	134
6-7: Effective waypoint and transmission trigger point geometry.....	136
6-8: RMAT Implementation	137
6-9: Histogram of the closest points of approach	139
6-10: Summary of CPA statistics by run.....	140
6-11: Effect of communication timing errors on the RMAT flight path	142
6-12: CPA statistics by simulation run	143
6-13: Difference of RMAT and Datalink CPAs, plotted by scenario CPA	144
6-14: The absolute value of CPA Difference statistics by simulation run.....	144
6-15: Illustration of cross track error from along track error	145
7-1: Functional decomposition of the APFDS	148
A-1: Example configuration file.....	162
B-1: File organisation	164
C-1: Example flight plan	167
D-1: Example configuration file	169

Notation

Name	Symbol
Along-Track Error	ATE
Altitude	h
Bank Angle	ϕ
Cross-Track Error	XTE
Degrees	deg
Estimated or Predicted Value	\hat{x}
Feet	ft
Flight Level	FL
Flight Path Angle	γ
Ground Speed	v_{gnd}
Heading	ψ
Knots	kts
Latitude	ϕ
Longitude	λ
Metres	m
Nautical Miles	nmi
Radians	Rad
Seconds	s
Time Rate of Change	\dot{x}
Track Angle	θ
Track Angle Change	α
True Airspeed	v_{tas}
Wind Direction	θ_{wnd}
Wind Speed	v_{wnd}

Chapter 1

Introduction and Motivation

1.1 Air Traffic Management

In 1956, a Trans World Airlines Super Constellation and a United Airlines DC-7 collided over the Grand Canyon, Arizona, killing all on board. The tragedy highlighted the need for a formal system of Air Traffic Control (ATC) in the United States and led to the creation of the Federal Aviation Agency (FAA) (Nolan, 2004). Preventing collisions between aircraft operating in the system remains the primary purpose of ATC (FAA, 2010). However, ATC has evolved globally as air transportation became more popular and as new technologies were developed, and ATC responsibilities have grown to include the task of organizing and expediting the flow of traffic, as well as providing other support to aircraft on a capacity-available basis. ATC is now often referred to as ATM (Air Traffic Management), reflecting the shift from strictly controlling traffic to the more encompassing task of managing traffic.

The domains of ATM activity are often categorised as communications, surveillance, and navigation. Air traffic controllers currently use VHF AM radios to maintain voice communications with aircraft under their control. Digital data communication (datalink) is commonly used by commercial aircraft to communicate with their airline operation centres. ATM use of datalink has been introduced in various airspace regions under the Controller-Pilot Datalink Communications (CPDLC) program, automating routine communications such as frequency and transponder assignments, a limited set of ATC clearances, and microphone checks – tasks that can occupy up to 50% of controller activity (Gonda, et al., 2005, EUROCONTROL, 2010b).

Traffic surveillance is primarily accomplished through the ATC Radio Beacon System (Nolan, 2004). Ground-based secondary surveillance radars (SSR) periodically interrogate all aircraft within range. Aircraft equipped with transponders are capable of replying to the interrogation, providing range and bearing, as well as encoded data

such as identify and altitude. However, in some remote and oceanic areas, radar surveillance is limited or non-existent. In these cases, verbal position-reports and flight plan tracking are used in lieu of radar surveillance.

Navigation services are provided to aircraft through an extensive network of ground-based navigation aids, such as VOR and DME stations (Nolan, 2004). Traffic routes were developed to correspond with these navigation aids in order to effect organised and efficient flight operations and to ensure navigational coverage. Aircraft navigation capabilities have improved dramatically with the advent of advanced on-board navigation systems such as precise inertial navigation systems (INS) and Global Navigational Satellite Systems (GNSS).

1.2 Air Traffic Management Modernisation

However, Air Traffic Management (ATM) systems throughout the world are undergoing a period of major transformation and modernisation, driven by the limitations of the current ATM system and the growth in air traffic demand.

1.2.1 The Need and Direction of Modernisation

The national airspace system in the United States is approaching maximum capacity. In 2007, the FAA reported record levels of air traffic delays and predicted they would continue to grow until the system becomes gridlocked around the year 2015 unless action was taken (FAA, 2007). Despite the global economic downturn and the resulting decrease in the number of commercial flights in 2009, the FAA forecasts that air traffic will grow 19% over the next 8 years (FAA, 2010). The Joint Development and Planning Office (JPDO) – the organisation responsible for overseeing ATM modernisation in the United States – warns that “The current method of handling traffic flow will not be able to adapt to the higher volume and density demanded of it in the future, even if twice as many or more resources are devoted to it” (JPDO, 2004). The JPDO has developed a long-term plan, called NextGen, to guide the development

of these new methods of handling traffic with the goal of increasing safety, security, and capacity.

Europe faces a similar challenge; on a peak day in 2005, Europe's ATM system was responsible for nearly 30,000 commercial flights (SESAR, 2006). EUROCONTROL forecasts that traffic growth will recover by 2011 to a rate of 3% per year, resulting in a nearly 22% increase in IFR traffic between 2009 and 2016 (EUROCONTROL, 2010a). A public-private consortium responsible for Europe's ATM modernization, called the Single European Sky ATM Research (SESAR) Programme, has concluded that "There is a need for a paradigm shift in today's concept of operation to break through the capacity barrier predicted to occur around 2015" (SESAR, 2006).

In order to meet this demand, while simultaneously improving the safety and efficiency of flight operations, both SESAR and NextGen foresee a future airspace system transformed by the concept of trajectory based operations, and leveraging new technologies in the communications and surveillance domains.

1.2.2 Trajectory Based Operations

Trajectory based operations (TBO) represent a shift away from the broad, static directives and the fragmented airspace that characterise the current air traffic control system. Instead, precisely defined flight trajectories will provide the basis for planning and executing all flight operations. Unlike current-day flight plans, these trajectories will be described by a series of Earth-referenced waypoints that define the centreline of the flight path (including position uncertainty), unconstrained by the current route structures and ground-based navigation aids (SESAR, 2007). While full-TBO involves precise management of an aircraft's 4D trajectory in time and space, the concept is scalable to accommodate 2D (lateral) or 3D (lateral and vertical) trajectories which are appropriate for lesser equipped aircraft.

The NextGen Concept of Operations notes that with TBO, "The traditional responsibilities and practices of pilots/controllers will evolve due to the increase in automation, support, and integration inherent to trajectory management" (JPDO,

2009). Under TBO, desired trajectories will be negotiated between the aircraft operators and the Air Navigation Service Providers (ANSP), tailored to individual flight preferences and airspace constraints. Aircraft will then be contracted to fly this trajectory within a required navigation performance, and any subsequent trajectory modifications will be renegotiated.

Trajectory-based operations are expected to increase the capacity, efficiency, and safety of the global air transportation system by permitting optimised, flexible routing from gate-to-gate (Funabiki, et al., 2003; Prevôt, et al., 2003; Wichman, et al., 2007). A major limitation to efficient flight routing in the current ATM system is the fixed route structures which were designed around ground-based navigation aids. Nolan notes that “During any given day, pilots using the low-altitude victor airway system add approximately 125,000 miles of extra distance to their flight plans as a result of preferred routes” (2004). However, the evolution of advanced Flight Management Systems (FMS) and improvements in navigation technologies such as Global Navigational Satellite Systems (GNSS) and multi-sensor data fusion have made it possible to navigate precisely apart from the fixed route structure – this capability is known as area navigation (RNAV). RNAV procedures have already shown significant reductions in fuel burn, emissions and flight time (Jha & Crook, 2009; Sprong & Mayer, 2007). TBO will permit aircraft to take full advantage of RNAV through unique, dynamic RNAV routes.

Additionally, TBO intends to reduce navigational uncertainty through widespread application of the Required Navigation Performance concept (RNP). Under RNP, aircraft are required to navigate along the designated route of flight to a given accuracy and precision, with on-board performance monitoring and alerting. Constraining the navigational uncertainty in this way increases airspace capacity by permitting reduced separation standards and more closely spaced traffic flows (FAA, 2006).

1.2.3 Communication

A second fundamental shift from the current ATM system is that the digital data exchange of trajectories will become the primary mode of communication between an ANSP and flight operators, replacing the verbal delivery of ATC clearances (SESAR, 2007; JPDO, 2009). Although voice communications using the legacy VHF radio-telephony (R/T) systems will remain, their role will likely change to being used for non-routine communications and to provide a back-up means of communication in the event of datalink failure.

Datalink communications enable complex message sets (such as the uplink and downlink of detailed trajectories) to be transmitted. Trajectory exchange between ground ATM systems and airborne FMS using datalink has been successfully demonstrated both in hardware-in-the-loop simulation studies and in flight trials (van Gool & Schröter, 1999; Jones & Schleicher, 2001; Mueller, 2007). Furthermore, from flight trials in a NASA Boeing-737 test aircraft, Knox and Scanlon have shown that datalink can significantly reduce communications errors between pilots and controllers compared to conventional radio-telephony procedures (1991). One concern about datalink communications, however, is the possibility of reduced situational awareness due to the lack of the 'party-line' effect with datalink (Fan & Kuchar, 2000). New avionics, Cockpit Display of Traffic Information (CDTI) systems, have been introduced as a solution, greatly improving flight crew situational awareness (SESAR, 2007; JPDO, 2009).

Frequency congestion and channel saturation are a further incentive for the transition to datalink communications (DLBST, 1996). Although VHF channel spacing has already been reduced in Europe from 25 kHz to 8.33 kHz to increase the number of available channels, frequency congestion remains an issue. A study of over 60 hours of radio traffic in French en-route sectors found that on average, the frequency was congested more than 40% of the time (Graglia, 2002). The expansion of CPDLC is intended to address this concern.

1.2.4 Surveillance

The accuracy, extent, and availability of surveillance information will also be improved in future ATM systems by integrating airborne and ground-based sources (SESAR, 2007; JPDO, 2009). In the Automatic Dependent Surveillance – Broadcast (ADS-B) concept, all aircraft will periodically broadcast position, velocity, and intent data to other traffic and the ANSP. ADS-B can provide surveillance in remote areas not currently covered by radar, as well as an air-to-air surveillance capability (Lester & Hansman, 2007). This detailed knowledge of both the current aircraft state and the intended flight trajectory can help improve the accuracy of trajectory predictions made by ATM decision support tools (DST) and automation. Better trajectory predictions, in turn, can facilitate more accurate conflict detection, traffic flow management, and strategic airspace planning (Mondoloni, 2006; ECC, 2009).

ADS-B can also be augmented by both legacy ground-based systems such as secondary surveillance radar (SSR), as well as emerging technologies such as Wide Area Multilateration (EUROCONTROL, 2005). Surveillance information gathered by the ANSP will be shared with all traffic through Traffic Information System – Broadcast (TIS-B) and displayed by CDTI systems (SESAR, 2007; JPDO, 2009).

Technology and architecture independent standards are being established to define the high-level system performance requirements for surveillance applications. These Required Surveillance Performance (RSP) standards will define the accuracy, availability, integrity, latency, update rate, and continuity required of surveillance systems, allowing airspace designers to set safe separation minimums to enable more efficient airspace usage (Thompson, et al., 2006).

1.3 Separation Management

Assuring safe aircraft separation is the highest priority of Air Navigation Service Providers (FAA, 2010). The growing traffic levels and the modernisation plans

described above have significant implications for the separation management process – both requiring and enabling new separation management methods.

1.3.1 The Current Separation Management Process

Currently, separation management (SM) is primarily a manual process. That is, air traffic controllers must make cognitive operational judgements to identify and resolve possible conflicts on the basis of information from surveillance radars, flight progress strips, and experience (Prevot, et al., 2005). Ensuring safe separation takes place in conjunction with other controller tasks such as:

- managing communications
- coordinating with other controllers
- responding to pilot requests
- monitoring convective weather and other airspace hazards.

Airspace design, structured routing, and standard flight rules prevent many conflicts from forming in the first place. In addition, a limited range of decision support tools have been developed to assist with separation provision. For example, modern surveillance displays allow controllers to show range rings around selected aircraft to help visualise separation distances, as well as trajectory prediction lines from the radar-derived velocity vector to help visualise future aircraft positions (Prevot, et al., 2005). Conflict detection tools such as the Medium-Term Conflict Detection (MTCD) system can help alert controllers of potential conflicts up to 20 minutes ahead, showing the time of conflict and predicted minimum separation (Kauppinen, et al., 2002). Conflict probes such as the User Request Evaluation Tool (URET) can then be used to search for conflict free routes (Brudnicki, et al., 2007).

Despite these tools, air traffic controllers remain strictly responsible for conflict detection and resolution. As a result, they must maintain constant situational awareness of all aircraft under their control. To ensure the traffic load does not exceed the cognitive capabilities of the controllers managing a given sector, limits are placed on the number of aircraft that can operate safely in that sector (Prevot, et al., 2005).

These limits, however, do not reflect the true potential capacity of the airspace. For example, in an analysis of two US en-route sectors near the workload limit of the controllers, Andrews observed that less than half of the aircraft in the sector ever came within 20 nmi (Andrews, 2001). He further concluded that the airspace volume had the capacity to handle more than four times the number of aircraft allowed by the controller workload limit.

1.3.2 New Concepts for Separation Management

To overcome the limitations of the current SM process, new automation functions, roles, and operational concepts are being explored to help ensure safe and efficient air transportation for up to three times the current traffic levels.

1.3.2.1 Automated Conflict Detection and Resolution Systems

Initial research has suggested that more highly automated, trajectory-based separation management has the potential to maintain safe aircraft separation in high density, high complexity airspace (Erzberger & Paielli, 2002; Andrews, et al., 2006; Callantine, 2007; Gawinowski, et al., 2008). In future ATM systems, improved surveillance information and knowledge of aircraft intent will allow better automatic conflict detection with fewer false alarms and undetected conflicts. Conflict resolution algorithms can then be used to generate optimised, conflict-free trajectories. These resolutions can be uplinked directly to the aircraft flight guidance system, ready for flight crew approval, using digital datalinks coupled to the FMS. Consequently, NextGen foresees that SM in future TBO airspace will be based on intelligent automation, with a shift in roles for air traffic controllers from tactical separation between individual aircraft to the strategic management of traffic flows in high-density airspace (JPDO, 2009). Similarly, SESAR intends to use automation to support conflict/interaction detection, situation monitoring, and conflict resolution (SESAR, 2007).

Methods and algorithms for automatically detecting potential conflicts and generating resolutions have been discussed in the literature since at least 1973

(Flanagan, et al., 1973). An extensive review of conflict detection and resolution (CD&R) methods conducted in 2000 by Kuchar and Yang revealed 68 separate systems that were under development (2000). However, the design and evaluation of new and improved systems remains an active field of research. Numerous other CD&R systems have been published subsequently to the Kuchar and Yang review, including at least 10 new algorithms or methods in 2008 and 14 in 2009; for example Lei, et al., (2008), Archibald, et al., (2008), Vela, et al., (2009), and Karr, et al., (2009).

However, before any new CD&R system can be operationally implemented, the safety and efficiency performance must first be thoroughly evaluated. Safety is paramount to any CD&R system; if the SM process is to be more highly automated, then conflicts must be detected and resolved with a high degree of accuracy and integrity, even in complex, high density traffic scenarios (Erzberger & Paielli, 2002). The system must be shown to be robust against faults and uncertainties in the requisite navigation, communication, and surveillance systems. Efficiency is also important; CD&R systems should produce resolutions that minimise flight delays, fuel consumption, traffic disturbances, and the environmental impact. Many resolvers use a cost function to find an efficiency-optimised resolution, such as the fuel-optimal integer programming algorithm described by Vela, et al. (2010).

In addition to the design and analysis of novel CD&R systems, there is also considerable work underway to develop operational concepts that integrate these systems effectively into future airspace, particularly with a focus on acceptable levels of automation, the delegation of separation responsibility to flight crews, and mixed-mode operations.

1.3.2.2 Levels of Automation

Although SM will be based on “intelligent automation,” the acceptable and appropriate level of automation is being debated (Kirwan & Flynn, 2002; Zemrowski, 2008; Dwyer & Landry, 2009). Ideas for levels of automation range from maintaining the current SM process, but with more capable tools, to fully automated conflict detection and resolution that does not require any controller involvement (Prevot, et

al., 2005). Variations exist within this ranking, but the general progression of automation levels is:

1. Assisted conflict detection, with limited resolution assistance (conflict probing);
2. Automated conflict detection, with limited resolution assistance (conflict probing);
3. Automated conflict detection, with automated resolutions upon the controller's request;
4. Automated conflict detection, with automated resolutions, ranked and displayed for the controller's selection;
5. Fully automated conflict detection and resolution.

EUROCONTROL has been developing a suite of tools to support level-4 automation (Kirwan & Flynn, 2002) under the Automated Support to Air Traffic Services (ASA) programme. Using the MTCD system and a conformance monitoring tool for conflict detection, the Conflict Resolution Assistant (CORA) tool generates a set of ranked resolutions. Kirwan and Flynn observe that "CORA is not intended to replace the controller's skill of conflict resolution, but rather is meant to support it and extend the controller's abilities and capacity for handling more traffic safely and expeditiously" (2002). Zemrowski cautions, though, that relegating the human to a monitoring role in this way could reduce alertness; rather, "attention must be made to ensure that the mundane monitoring can be done by the automation and that the role of the human is kept challenging but not frantic" (Zemrowski, 2008).

A series of controller-in-the-loop simulations were conducted by the NASA Ames Research Center to evaluate the effect of various levels of automation support on controller workload and separation violations (Prevôt, et al., 2008). Reassigning responsibility for conflict detection from controllers to automation (as with levels 2 to 5) was found to significantly reduce controller workload. But the study also found that even level 3 and 4 automation may not be adequate in future high density, high complexity traffic. Attempting to search manually for conflict free trajectories using an advanced graphical conflict probe at peak traffic densities (three times the current

levels) resulted in numerous separation violations. Selectively using the automation to generate a resolution (level-4) still resulted in very high workload in high density situations. The study concluded that higher levels of traffic density required higher levels of automation to maintain safe separation. Summarising this research and previous work, Prevôt, et al, said that “The HITL [Human-In-The-Loop] research so far has indicated that ground-based automated separation assurance is a generally sound concept for trajectory-based operations in high density en route airspace. Trajectory-based conflict detection and resolution automation integrated with data link should become a core NextGen technology and could possibly be operationally evaluated in the near future” (Prevôt, et al., 2009).

However, relying on automation raises the significant safety issue of handling failures. The NextGen concept of operation notes that the increased reliance on automation must be coupled with fail-safe modes that do not require full reliance on human cognition as a backup for automation failures (JPDO, 2009). Similarly, the SESAR consortium has called for “automation that is coupled with fail-safe modes that do not require full reliance on human situational awareness as a backup for automation failures” (SESAR, 2007). One solution is a layered approach to SM across different timelines. Erzberger and Paielli have proposed a concept that uses a strategic CD&R system (approximately 2 to 20 minute horizon) to generate optimised resolutions based on the aircraft’s intended 4D trajectory (2002). The strategic system would operate in parallel with a separate, fully-independent tactical CD&R system designed to identify and resolve short-term conflicts (0 to 2 minutes) when, for example, aircraft deviate from their flight plan. The legacy Traffic Collision Avoidance System (TCAS) would remain a final safety net to prevent collisions in the event that both the strategic and tactical resolvers fail to resolve the conflict.

1.3.2.3 Delegated Separation Responsibility

Trajectory based operations and improved communication and surveillance technology will also allow the delegation of some separation responsibility to flight crews (SESAR, 2007; JPDO, 2009). Aircraft properly equipped with on-board conflict

detection and resolution capability, called Airborne Separation Assurance Systems (ASAS), and CDTI displays will be permitted to self-separate from other traffic, thereby reducing controller workload while simultaneously enabling more user-preferred routing.

The feasibility of self-separation has been demonstrated by several large research programmes including the FREER (Free-Route Experimental Encounter Resolution) programme and the free-flight research by the National Aerospace Laboratory of the Netherlands (NLR) (Duong, et al., 1997; Ruigrok & De Gelder, 2006). However, operational concepts for implementing ASAS are still under consideration and must address a number of significant questions, including:

- Will both conflict resolution *and* detection be delegated? (Loscos, et al., 2007)
- What systems or strategies will be in place to mitigate failures? (Loscos, et al., 2007)
- Will self-separating aircraft be segregated into different airspace from conventionally managed aircraft? (Hoekstra, et al., 2000)
- How will fairness be managed? That is, in a conflict, which aircraft should manoeuvre? (Jonker, et al., 2005, Del Pozo de Poza, et al., 2009)

To address some of these questions and make progress towards initial ASAS operations in Europe, the Advanced Safe Separation Technologies and Algorithms (ASSTAR) project was conducted to establish a “common endorsement of the proposed ASAS applications” (Loscos, et al., 2007). Similarly, in the United States the NASA Distributed Air/Ground – Trajectory Management (DAG-TM) programme has been exploring the human-factors implications of ASAS equipment and operations (Lee, et al., 2003).

1.3.2.4 Mixed-Equipment Operations

Any new concept for SM must also consider methods to safely and efficiently handle traffic that is not fully equipped for trajectory-based operations.

There are two main reasons why air traffic may have mixed equipment or levels of capabilities. Firstly, with advances in technology, many features of TBO will require advanced avionics such as an RNP-capable FMS, ASAS equipment, and CDTI displays, as well as improved communications equipment such as an FMS-integrated datalink. Major avionics upgrades often take place over major maintenance periods – sometimes as infrequently as 7 years (Zemrowski, 2008). Thus, the transition from the current, clearance-based ATC system to a more highly automated, trajectory-based system will be gradual and ANSPs will need to provide different levels of service to some users. Secondly, ATM systems must account for the possibility of equipment failures. Strategies specific to TBO must be developed to mitigate the effects of faults due to human error or automation failure.

However, mixed-capability operations introduce a significant challenge to air traffic controllers. A study of mixed-RNP capability in oceanic operations by Forest and Hansman found that controllers reported a greater level of difficulty and a loss of situational awareness in scenarios where only 50% of the traffic was RNP equipped (2006). A subsequent simulation study by Pina and Hansman suggest that controllers have difficulty correctly identifying conflicts when equipage is less than 50% (2004).

In light of the difficulty of mixed-capability operations, one solution is to designate certain airspace for TBO traffic and exclude all aircraft that cannot be supported by either ground-based or airborne SM automation. ‘Segregated’ airspace could provide a homogeneous operating environment where all aircraft use similar procedures and automation tools to maintain safe separation, and could also encourage users to invest in advanced equipment (Forest & Hansman 2006; Kopardekar, et al., 2009). In a proposal for a more highly automated ATM system, Erzberger and Paielli concluded that, “The level of difficulty in handling encounters will strongly depend on the density of traffic and on the complexity of the traffic flow. As a rule, an unrestricted mix of equipped and unequipped aircraft will have to be avoided, since it would reduce capacity and efficiency” (2002). The segregated airspace solution is also supported by the NextGen Concept of Operations (JPDO, 2009).

Other research, however, has questioned the necessity of segregation (Hoekstra, et al., 2000; Kopardeker, et al., 2009). To highlight this, Doble, et al., conducted a simulation study of a mixed-ASAS concept where properly trained flight crews with ASAS-equipped aircraft assumed responsibility for separation from other ASAS aircraft and from Instrument Flight Rules (IFR) aircraft (2005). Ground-based air traffic controllers managed the separation of IFR traffic and issued flow management constraints to all aircraft. Under this operational concept, the number of ASAS aircraft in the sector was shown to have very little effect on controller workload in providing separation assurance for the IFR traffic, indicating that integrating mixed capability traffic in the same airspace may be feasible within certain constraints. Integration could even have a number of advantages over segregation. As has been stated, mixed-capability operations may be the normal state for an extended period. Restricting airspace access may come at the cost of underutilized airspace capacity and reduced flexibility, possibly offsetting the benefits of TBO (Kopardeker, et al., 2009).

1.4 Simulators for Separation Management Research

The new SM methods that are under development promise to help overcome the capacity barrier and ensure safe separation in future airspace, but also raise many new questions; systems and concepts are evolving and many uncertainties remain. Arguably, simulation can provide a key tool to answer these questions. Simulators can support the design, analysis, and verification of new systems by providing insights into safety, performance, and implementation issues (Galati *et al* 2003). As a result, simulation is used extensively to aid nearly every step of ATM research and development.

Unlike flight simulators, which are designed to model a single aircraft, air traffic simulators are used to model entire air traffic systems consisting of: multiple, simultaneous aircraft with a variety of performance profiles and flight plans; air traffic management services; weather and atmospheric effects; as well as the associated communication, navigation, and surveillance functions to provide interaction between

simulated entities. Air traffic simulators make it possible to research the relations and effects of complex, stochastic, highly-coupled systems over a range of conditions that would be impractical or unsafe to test with live trials (Sweet, *et al*, 2002).

The level of fidelity – that is, the degree of realism – of air traffic simulators varies according to the intended use. High fidelity simulators often contain detailed, complex airspace and aircraft models, and are used when accuracy and confidence in the results are important, such as pre-operational validation. Lower-fidelity simulators are used when the level of detail and realism is not as important, and when the complexity of higher-fidelity models would hinder efficient research. Lower-fidelity simulators are often used early in the development process, before designs and concepts of operation have been finalised, for example, to support ‘what-if’ analysis, to demonstrate feasibility, or to compare alternative designs and concepts.

Similarly, the speed of air traffic simulators varies according to the intended use. Real-time simulators compute one simulated second in one actual second, and are principally used when human interaction is required, such as human-in-the-loop research that involves real pilots and controllers. With fast-time simulators, one simulated second is computed in less than one actual second. Because fast-time simulators are able to compute scenarios in a fraction of the time required by real-time simulators, they are well suited for large-scale, stochastic simulations that would be unfeasible to conduct in real-time. As a result, fast-time simulators are valuable for initial research into new SM concepts and systems since they are able to provide coverage over a large set of possible scenarios. For example, some of the proposed CD&R algorithms are heuristic; requiring extensive simulations to prove their effectiveness, and ultimately, their safety. Higher fidelity, real-time simulators can then be used for a more focused set of human-in-the-loop experiments. A number of fast-time air traffic simulators have been developed to model future ATM environments that could be applied more specifically to SM research.

1.4.1 Commercial Simulation Platforms

Three commercial simulation packages are available and widely used by the ATM industry to examine the current air transportation system as well as new ATM concepts and systems: TAAM, RAMS Plus, and FACET. All three are fast-time, discrete-event simulators – that is, aircraft movement is updated at discrete time intervals.

1.4.1.1 TAAM

The Total Airspace and Airport Modeller (TAAM) was developed by The Preston Group in association with the Australian Civil Aviation Authority, and is now maintained by Jeppesen, a subsidiary of The Boeing Corporation (Jeppesen, 2008). TAAM is described as a “fast-time gate-to-gate simulation tool that enables operators to accurately predict and analyze the impact of present and future airspace and airport operations, whilst maintaining safety and efficiency.” (Jeppesen, 2008) TAAM is particularly valuable for high-fidelity simulations of terminal and airport operations, including ground movement and handling.

Comprehensive input data is needed to describe the airspace and ATC system in order to model a customised air traffic scenario; the level of fidelity of the simulation can then be managed by the level of detail provided in the input data (Feigh, 2003). TAAM can produce 3D visualisations of airport and traffic scenarios, as well as output data directly to databases and spreadsheets. The tool has been used to (Jeppesen, 2008):

- Plan airport improvements and extensions
- Study noise impact and the effect of severe weather
- Assess controller workload
- Design new terminal procedures

1.4.1.2 RAMS Plus

RAMS Plus is a high-fidelity, fast-time simulation tool that can be used to model gate-to-gate ATM and airport operations (Geisinger, 2003). Originally developed as the Reorganised ATC Mathematical Simulator by EUROCONTROL in 1991, the software is now licensed and maintained by ISA Software. RAMS Plus is scalable from the macro to

micro level, and can be used to as a planning and feasibility tool to investigate proposed changes to existing ATM systems, the impacts of new ATM elements, as well as classic controller workload analysis. As with TAAM, the simulator is fully data-driven requiring comprehensive input data (ISA Software, 2003). To model a basic airspace scenario, data is needed for navaid names and geographic locations; traffic schedules; traffic routes; sectorisation boundaries, and weather patterns. Default data is provided for aircraft models, aircraft performance, airport locations, ATC event rulebases, and country frontiers.

RAMS Plus executes on a single desktop PC and is capable of producing a range of outputs including 2D visualisations, ASCII text, and XML metadata. An interoperability framework makes it possible to integrate other tools with the simulator. The main strength of RAMS Plus is the ATC rulebase that can be customised into an artificial intelligence model of air traffic controller tasks such as conflict detection, resolution, and traffic flow management (Geisinger, 2003). The principle application areas of RAMS Plus have been (ISA Software, 2003):

- Evaluating alternative sectorisations
- Measuring airspace and conflict complexity and density
- Measuring airspace safety in relation to separation violations.
- Investigating free-routing and Reduced Vertical Separation Minimums (RVSM) concepts.
- Measuring the effect of ATM procedures on fuel burn.

1.4.1.3 FACET

The Future ATM Concepts Evaluation Tool (FACET) is a fast-time simulation platform designed at the NASA Ames Research Laboratory for the exploration, development, and evaluation of advanced ATM concepts (Geisinger, 2003). FACET is now commercially licensed and is promoted for visualization, off-line analysis, and real-time planning applications. FACET models the system-wide en-route airspace operations over the contiguous United States, using an airspace database, weather database, aircraft performance database, and traffic data (tracks, flight plans, and

schedules)(Bilimoria, et al., 2001). To facilitate fast execution on a single PC, aircraft trajectories are simulated with spherical earth kinematic equations, using lookup tables of aircraft performance.

FACET includes functionality for aircraft self-separation; prediction of aircraft demand and sector congestion; traffic flow management constraints, and wind-optimal routing. The simulator has principally been used to evaluate the feasibility of new concepts for distributed air/ground traffic management and advanced traffic flow management, as well as new decision support tools for air traffic controllers (Bilimoria, et al., 2001).

1.4.1.4 Limitations

All three of these simulators are very capable, high-fidelity tools and there are compelling reasons to use them for researching new SM concepts and systems. As commercial products, they have been thoroughly validated, and are trusted and used by NASA, EUROCONTROL, the FAA, as well as airlines. Both FACET and RAMS Plus have already been used to demonstrate aircraft self-separation concepts.

However, these tools also have some limitations that present challenges to researchers. The first is the inherent complexity of these simulators and the need for extensive input data. Commenting on TAAM and RAMS Plus, Donohue and Laska concluded that “These models provide a detailed analysis, but require significant amounts of data that are sometimes difficult to obtain. Learning to use these models takes considerable time and effort limiting their use to specialized individuals” (2001). FACET, on the other hand, is simpler but currently only models the high-altitude, U.S. airspace system (Bilimoria, et al., 2001). As a result, FACET is more appropriate for examining the system-wide effects of new procedures, rather than the local analysis of new conflict detection and resolution methods.

In addition, as commercial products, these simulators are generally closed-source (RAMS Plus has an open interface, but the source code is still inaccessible), limiting the ability of researchers to modify the simulators for specific applications – for example,

by adding customised models of navigation, communication, or surveillance uncertainties and errors; modelling datalink trajectory exchanges as with future TBO airspace; or integrating with novel conflict resolvers that do not use the standard rulebase. Furthermore, both TAAM and RAMS Plus are prohibitively expensive for individual researchers, although a free, limited university licence is available for RAMS Plus.

1.4.2 Non-Commercial Simulation Platforms

Non-commercial airspace simulators are also used to support ATM research and a number of these tools are discussed in the literature. However, these simulators were also found to be either too inaccessible or restrictive for exploratory SM research.

Large research organisations, such as NASA, NLR, and EUROCONTROL have developed their own high-fidelity simulation platforms. The NASA Airspace Concept Evaluation System (ACES) is a large-scale, distributed simulation framework to support system-wide evaluations and is highly integrated with other NASA simulation tools (Sweet, et al., 2002). NASA also uses the Center-TRACON Automation System (CTAS) which is used operationally as a trajectory prediction engine for decision support tools, but can also be used as a simulation platform (Murphy & Robinson, 2007). The NLR ATC Research Simulator (NARSIM) is a real-time ATM simulator that is distributed over multiple computers and controller workstations, and is primarily used to study human-machine-interfaces (Hoekstra, et al., 2000). Similarly, EUROCONTROL has developed the ESCAPE simulator (EUROCONTROL Simulation Capability and Platform for Experimentation) to study air traffic control in real-time (EUROCONTROL, 2007). These simulators provide accurate and detailed models of ATM systems, but consequently are very complex and require dedicated equipment, limiting their portability and extendibility. Furthermore, these simulators are generally unavailable for researchers not professionally associated with the research group.

Smaller, simpler simulators are also used by researchers, but unfortunately, these tend to be limited by a lack of public documentation, and are of unknown fidelity and quality (Feigh, 2003). For example, simulations are commonly used in studies of new

CD&R algorithms, but often are only referenced in the background and are designed for specific algorithms, as with Vela, et al., (2010), and Cetek (2009). Other simulators are developed for a specific applications, such as the Complete Air Traffic Simulator (CATS), introduced by Alliot, et al.,(1997) for evaluating reduced vertical separation minimums, or the Air Transportation System Simulator described by Carr, et al.,(2005) that is used for national-level simulations of U.S. airspace. As a result, they lack flexibility to easily add or modify components, a restriction that impedes their usefulness for exploratory research of novel CD&R algorithms and methods.

1.4.3 The Airspace Simulator by K. Feigh

To address the complexity and inaccessibility of existing simulators, Karen Feigh developed the Airspace Simulator at Cranfield University with the aim of providing an open source, low fidelity simulator that can be run on a single PC in fast-time (Feigh, 2003). The purpose was to provide ATM researchers with a simple yet flexible, non-proprietary airspace simulation platform for exploratory research into new ATM operating concepts and systems.

The Airspace Simulator was designed with a modular structure to make it portable and readily extendable. In the simulator, 4-dimensional aircraft trajectories are modelled using the kinematic BADA v3.3 performance database, which tabulates the cruise, climb, and descent performance at different flight levels for 186 different aircraft types. The use of the BADA database enables the simulator to rapidly calculate the position and velocities of a large number and variety of simulated aircraft at discrete time intervals, although this speed comes at the cost of reduced fidelity.

At the beginning of the program execution, the Airspace Simulator reads configuration files specifying the flight plans and initial conditions for every aircraft type. Aircraft are then guided along the flight plans using great circle, spherical earth navigation. Winds aloft are modelled with four dimensional wind fields, interpolated from observed wind data from the NOAA Profiler Network.

An indexed point-region quadtree is used to organise the simulated aircraft. This technique enables the simulator to locate all aircraft within a given geographic region very quickly, which is necessary to facilitate efficient conflict detection and range-limited radio transmissions. Air Traffic Control is provided through rule-based conflict resolution, defined by natural language scripts in order to model the behaviour of controllers. Information exchange between the ATC module and the pilot module takes place through a simple datalink model.

Test cases were run to simulate North Atlantic crossings, terminal manoeuvring at London Gatwick, and en-route operations between nine European airports. Feigh successfully demonstrated the simulator for up to 300 aircraft simultaneously and 4000 aircraft total over the simulated 16 hour period, representing twice the predicted 2020 traffic level of Europe's busiest airport (London Heathrow).

1.4.3.1 Case for Improving the Airspace Simulator for Separation Management Research

The Airspace Simulator shows potential to be a valuable fast-time, low-fidelity simulation platform for SM research. Because it is open-source with a modular architecture, the simulator could be adapted to specific applications. As a fast-time simulator with coverage of commercial aircraft performance, it could be used to conduct stochastic simulations of new CD&R algorithms and concepts over a range of conditions and traffic scenarios. However, upon closer evaluation it was apparent that there were four primary factors of the original design that limited the usefulness of the Airspace Simulator for SM research.

1) *Difficulty of integrating new CD&R systems;* Unfortunately, the navigation, communication, and air traffic control functions were not implemented distinctly, but were merged into one module, which made it difficult to integrate new CD&R systems, other than the included script. Although the natural language scripts are useful for modelling controller behaviour in current operations, it would be impractical to translate many of the conflict detection and resolution tools described in the literature into scripts of the form used by Feigh.

2) *Restricted navigation and flight control functionality*; The flight guidance produced by the navigation function was limited to great circle waypoint-to-waypoint navigation, and speed and altitude commands. Although waypoint-navigation is expected to be the predominate type of navigation in future TBO airspace, conflict resolution systems can issue a range of instructions, including:

- Speed assignments
- Altitude assignments
- Heading vectors
- Direct-to clearances
- Lateral offsets
- Sequences of waypoints

The simulator must be able to implement these instructions. There was also no provision for a navigation error model. Miquel, et al., has shown that navigation errors can significantly increase the percentage of unresolved conflicts in state-based ASAS systems. Such errors should be modelled when evaluating new CD&R algorithms (2007).

Furthermore, aspects of the navigation model were unnecessarily simplistic, such as instantaneous transitions between level flight and climbs/descents, and fixed-rate turns, reducing the fidelity of the model and introducing errors in experiment results. To highlight this, the Airspace Simulator models all heading changes as constant 3 degree per second turns. However, commercial transport aircraft at cruising airspeeds typically use constant bank turns (Mondoloni, 2006). The difference in the turn radius is significant; at 450 knots, a 3 degree per second turn will produce a turn radius of 2.4 nmi, while a constant bank turn with a 25 degrees bank angle will result in a 6.3 nmi turn radius.

3) *Restricted communications functionality*; Although the Airspace Simulator models basic datalink communication, the message set was limited and not easily extendable; for example, it does not support messages containing multiple waypoints (i.e. route modifications). As with the navigation model, there was no capability to model communication uncertainties such as transmission latencies and pilot response

delays. Additionally, the simulator did not contain a model of radio-telephone communications. An RT model requires a different message set than a datalink model, as well as different latencies and errors such as the verbal duration of the message and radio channel congestion. Although datalink is expected to be the primary mode of communications in future ATM systems, voice will continue to be used to control unequipped traffic and for off-nominal events. Without a RT model, the simulator cannot be used to investigate new voice-based automation tools, mixed-mode operations, or datalink failure effects.

4) *Impractical scenario generation method*; The scenario generation method simulates a steady traffic flow, with aircraft repeatedly flying a set of pre-defined flight plans. New aircraft enter the airspace at pseudo-random intervals, according to a Gaussian distribution. While this is useful for simulating a period of sustained traffic over known route structures, such as a stream of aircraft at an arrival fix, it is impractical for other common air traffic scenarios used to test CD&R algorithms and operational concepts, such as stochastic, replicated runs of crossing and passing traffic.

Reviewing the published literature, it is clear that research into new SM concepts and systems is active and ongoing. But it can also be seen that although a number of high-quality commercial and non-commercial ATM simulators exist, there is a lack of simple, flexible, and accessible fast-time simulators that are well suited for exploratory SM research. However, if the four limiting factors outlined above could be overcome, Feigh's Airspace Simulator could fulfil that role.

1.5 Research Aims

The purpose of the research presented in this thesis was to redevelop the Airspace Simulator into a more useful platform for evaluating the functionality, feasibility, performance, and robustness of new methods, algorithms, and strategies for separation management in future airspace.

1.6 Organisation of Remaining Chapters

The remaining chapters discuss the redesign and application of the Airspace Simulator. Chapter 2 derives the specific requirements needed to make the Airspace Simulator more useful for SM research, and discusses the overall design of the modified simulator. Chapter 3 focuses on the redesign of the navigation model. Similarly, Chapter 4 focuses on the redesign of the communications model. Chapter 5 then discusses the scenario generation method and the overall speed performance of the simulator.

The verification process ensures that the simulator performs as designed. Verification was completed for each module and is discussed in Chapters 3, 4 and 5. Final verification of the completed and integrated simulator is presented in Chapter 5. The validation process ensures the simulator adequately satisfies the intended purpose. To provide validation, the simulator was used as a platform for the preliminary analysis of automated SM support for mixed-equipage traffic using automated track angle vectoring. Chapter 6 develops the idea of automated vectoring, and discusses the results of the simulations.

Finally, Chapter 7 concludes with a summary of the work that was completed on the simulator, highlights the new insights into the flight guidance accuracy of vector navigation, and raises ideas for future work.

Chapter 2

Simulator Requirements and Design

The Airspace Simulator described in Section 1.4.3 was redeveloped into the Airspace Simulator–II to overcome the limitations of the original design and to meet the need for a simple, flexible, and accessible fast-time airspace simulation platform for separation management research. This chapter discusses the simulator design requirements and describes the overall software structure of the redeveloped simulator. The chapter concludes with a summary of changes made to the Airspace Simulator.

2.1 Simulation Requirements

The functional and performance requirements for the simulator were derived from the capabilities and constraints that were considered necessary for the software to be a useful tool to investigate separation management methods. These requirements are summarised in Table 2-1.

2.1.1 Capabilities

The functional requirements were established from the following five capabilities:

- *To simulate future air traffic systems.* The simulator must be capable of modelling the expected baseline ATM system in the 2015 to 2025 timeframe, including the communications, navigation, and surveillance domains. A basic requirement for any airspace simulator is the ability to model the performance and trajectories of multiple aircraft. But in addition to aircraft models, the transition to trajectory-based operations necessitates simulating trajectory exchanges over digital datalink, including downlink of trajectory intent and the uplink of trajectory modifications. To support TBO, both SESAR and NextGen call for area navigation with an accuracy of at least 1 nautical mile 95% of the time, so FMS-like flight guidance should be simulated within RNAV-1

(EUROCONTROL, 2008; JPDO, 2009). Modelling the future surveillance system requires ADS-B-type broadcasting of aircraft state and velocity data.

- *To integrate with CD&R systems.* Numerous CD&R systems have been proposed in the literature. For the simulator to be a useful tool for exploring the functionality, feasibility, performance, and robustness of these systems, it must be able to be integrated with centralized and decentralized CD&R tools. The simulator must be capable of outputting traffic surveillance and trajectory intent information to conflict detectors, and of receiving and implementing conflict resolutions from resolvers.
- *To simulate mixed-mode operations.* The study of separation management in mixed-mode operations is ongoing and is critical to determining the effects of system failures and the transition period from the current ATM system to a more highly automated system (Kopardekar, et al., 2009). The simulator should be capable of specifying and modelling different communication modes (datalink or voice) and navigational accuracies for different aircraft, and be able to designate either self-separation or centralised-separation responsibility for each aircraft.
- *To simulate errors and uncertainties.* The simulator should incorporate models of errors and uncertainties such as communication latencies, surveillance noise, and navigational wander. This capability is important because separation management is essential to the safety of flight, so any conflict detection and resolution process must be shown to be robust against errors and uncertainties in the communication, navigation, surveillance systems.
- *To support exploratory separation management experiments.* To simplify setup, the simulator should have the ability to automatically generate pseudo-random traffic scenarios, but it must also be possible to manually specify the scenario. Furthermore, it must have the ability to record experimental data for post-simulation analysis. It is also desirable to be able to visualise the traffic and to be able to control the speed of the simulation. Generally, the simulator must

be flexible and accessible to accommodate new functions, procedures, and applications.

2.1.2 Constraints

The simulator must operate within the constraints of speed and fidelity requirements. As discussed in Chapter 1, speed is defined as the number of discrete movements that can be calculated within a given period of time, and is influenced by the efficiency of the software design, the performance of the computer hardware, as well as the fidelity of the simulation. Fidelity describes the relative accuracy with which a simulation models a system. Higher fidelity models typically require more calculations per unit of simulated time than lower fidelity models, and as a result, can be slower.

Considering the simulators discussed in Section 1.4, there are several high fidelity commercial and non-commercial ATM simulators available, although at the cost of increased complexity and reduced accessibility. The need is for a simple, high speed simulator that can be used for exploratory research. So, it was decided that although aspects of the simulator fidelity would need to be improved, such as the navigation and uncertainty modelling, the general fidelity of the original Airspace Simulator was adequate for these purposes. In particular, it was decided to continue to use the tabular BADA aircraft performance database as the basis of the trajectory model, since the trajectory modelling method is known to be a significant factor of the total speed and complexity of an air traffic simulator (Suckhov, et al., 2003). The original speed requirement was maintained: to run faster than real-time for at least 300 simultaneous aircraft, and to simulate up to 4000 aircraft in total over a period of 12 to 16 hours, representing twice the projected 2020 peak traffic level at London Heathrow airport, the busiest airport in Europe (Feigh 2003).

Finally, to be accessible for research, the following constraints must also be met:

- Must be able to run on a single PC
- Must be non-proprietary

- Must be open-source

(Req. 1)	Capable of fast-time simulation of up to 300 aircraft simultaneously
(Req. 2)	Capable of simulating of up to 4000 aircraft in total
(Req. 3)	Simulates waypoint-to-waypoint flight guidance within 1 nmi
(Req. 4)	Capable of connecting with external CD&R software
(Req. 5)	Simulates broadcast of traffic state and trajectory intent
(Req. 6)	Allows input and execution of conflict resolutions
(Req. 7)	Simulates datalink and voice communication
(Req. 8)	Simulates navigation errors and uncertainties
(Req. 9)	Simulates surveillance errors and uncertainties
(Req. 10)	Simulates communications errors and latencies
(Req. 11)	Simulates mixed-mode traffic
(Req. 12)	Allows automatic generation of pseudo-random traffic scenarios
(Req. 13)	Allows manual setup of traffic scenarios
(Req. 14)	Capable of running on a single PC

Table 2-1: Summary of requirements

Significant modifications and improvements were made to the original Airspace Simulator in order to meet these requirements. The remaining sections of this chapter provide an overview of the redesigned simulator.

2.2 Programming Language and Operating System

The original Airspace Simulator was written in the Modula-2 programming language for the Linux operating system. Modula-2 has many features in common with C, C++, and FORTRAN, such as specialised data types and pointers, but has the advantage of requiring stricter syntax and data type definitions, thereby helping reduce programming errors. Unfortunately, Modula-2 is no longer commonly used and the original compiler, the Garden's Point Modula-2 compiler, is not supported on current Linux platforms. The first step in redeveloping the Airspace Simulator was to translate the source code into the C language. Standardised ANSI C was chosen because it is familiar to many engineers and researchers, and typically executes faster

than higher-level languages such as C++, C#, and Java (Kernighan & Ritchie, 1988). Good programming practices can help prevent syntax and type errors, and debuggers such as the open-source GDB can be set to enforce strict syntax and type requirements.

The Airspace Simulator-II compiles with the GCC (GNU Compiler Collection) 4.4.0 and executes in the Windows XP, Vista, and 7 operating systems on a single desktop or laptop PC. GCC is a free Linux-based compiler that is widely used and can be implemented in Windows through the MinGW software port. The use of ANSI C and GCC significantly improve the portability of the simulator.

The structure of the software directly affects the speed performance and accessibility of the simulator. The simulator organisation, key data structures, and, logical flow, described below, were all redesigned to meet the required capabilities described in Section 2.1.1, within the constraints of Section 2.1.2.

2.3 Simulator Organisation

The Airspace Simulator-II has been reorganised from the original design in order to better support the integration of new CD&R tools into the simulator with minimal modifications. The organisation, shown in Figure 2-1, groups the core functions into five modules, making it possible to enhance and customise the simulator by adding, modifying, or removing individual modules or sub-modules as needed.

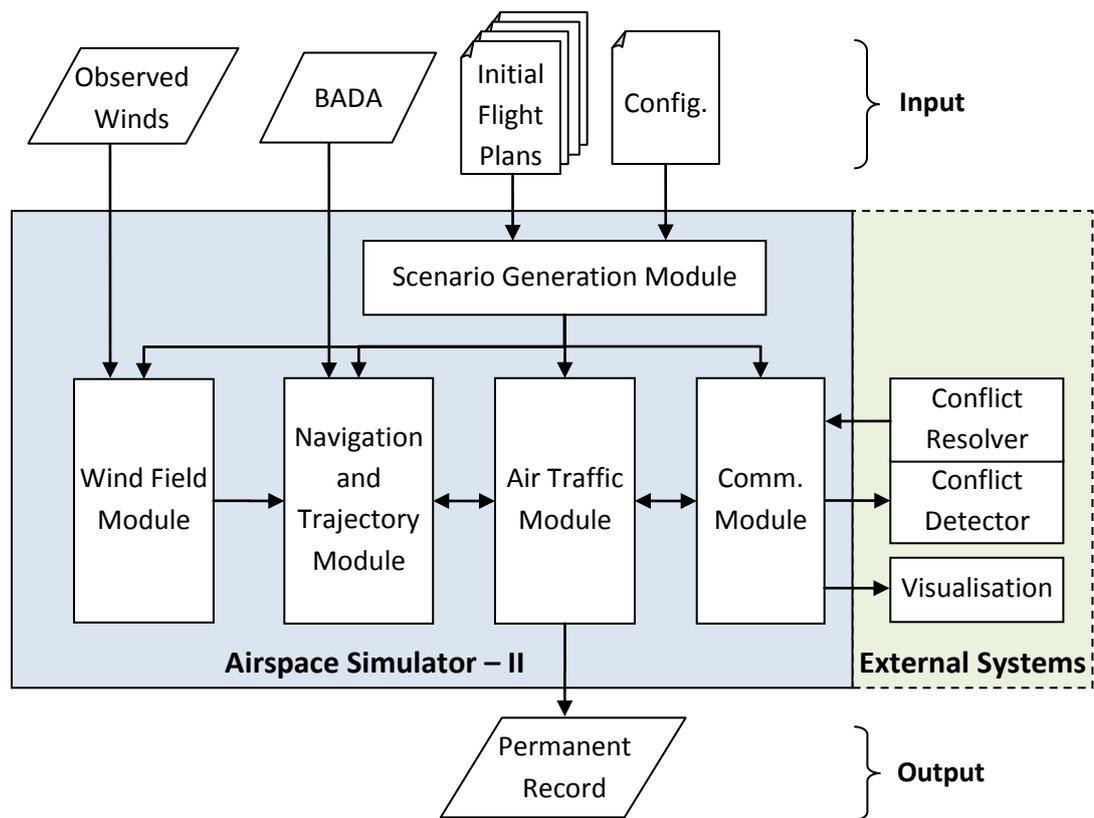


Figure 2-1: Simulator organisation

2.3.1 Scenario Generation Module

Simulated air traffic scenarios are arranged and controlled through the Scenario Generation Module. In keeping with the original design, scenarios can be manually set up through ASCII text files of aircraft flight plans and a configuration file listing aircraft assignments. These files are read by the Scenario Generation Module during the initialisation phase of the simulation. The flight plans describe the intended lateral and vertical flight trajectory from the initial aircraft position to the destination. They are discussed further in Section 3.2.1. The configuration file contains a list of every aircraft to be simulated over the course of the scenario. The list assigns each simulated aircraft to a performance type from the BADA database, a flight plan, a start time when the aircraft is to enter the traffic, a flight control mode (autopilot, or flight director), a communication type (ideal, datalink, or radio-telephony), and a separation mode (centralized, self-separation, or uncontrolled). The configuration file makes it possible to specify arbitrary traffic scenarios from 1 to 4000 aircraft, including mixed-mode

traffic. The file also contains basic simulation parameters such as time and aircraft limits, the data recording frequency, and the nominal surveillance broadcast rate. An example of the configuration file is presented in Appendix A.

It would be impractical to manually prepare flight plans and traffic assignments for a large number of aircraft – this was one limitation of the original Airspace Simulator. So, to better facilitate large, stochastic experiments, a sub-module of the Scenario Generation Module was designed to optionally generate pseudo-random flight plans and traffic assignments. This capability is described in Chapter 5.

2.3.2 Air Traffic Module

The core of the simulator is the Air Traffic Module. The purpose of this module is

- to maintain the information state of every actively simulated aircraft
- to generate new aircraft when prompted by the Scenario Generation Module
- to remove aircraft from the simulation when the aircraft arrives at the destination.

In order to eliminate the need to specify detailed initial conditions for each aircraft, as with the original Airspace Simulator, all aircraft are initialised at the first waypoint in the flight plan and are assumed to be in level, un-accelerated flight at cruise airspeed, heading along the course of the initial flight plan leg.

The Air Traffic Module controls the progression of the simulation. As a discrete event simulator, the simulation state is advanced iteratively by a fixed time step, Δt . This step size can be set in the configuration file, but is by default one simulated second per time step. The aircraft motion is modelled with the Euler forward integration method, and therefore a smaller Δt will result in smaller integration errors and more accurate trajectory modelling, but will increase the number of discrete events for a simulation of a given duration, thereby reducing the speed.

Finally, this module also organises the traffic spatially using an indexed-region quadtree to minimise the search time required to locate a specific aircraft or set of aircraft, as described by Samet (1989) and implemented by Feigh (2003).

2.3.3 Navigation and Trajectory Module

The Navigation and Trajectory Module interacts with the Air Traffic Module to guide aircraft along their flight plans, updating the air traffic trajectories at every time step. This module is decomposed into the Performance, FMS, Autopilot/Flight Director System, Dynamics, and Flight Technical Error sub-modules, and was completely redesigned in order to meet the project requirements. The new design and verification are discussed in detail in Chapter 3.

2.3.4 Wind Field Module

Wind disturbances have been shown to significantly affect the performance of the trajectory predictors that underlie conflict detection and resolution tools (Mondoloni, 2006; Cole, et al., 2000). To simulate variable wind fields, the Wind Field Module uses actual wind data extracted from 24 hours of archived readings of the NOAA Profiler Network (NPN) weather radars. The module interpolates the wind database according to the 4D aircraft location (latitude, longitude, altitude, and time of day). The magnitude and direction of the local winds can then be included in trajectory calculations by the Navigation and Trajectory module. The Wind Field Module is largely unchanged from the original; for a full description, refer to Feigh (2003). However, the ability to set a constant wind magnitude and direction in the configuration file was added to provide an additional wind option.

2.3.5 Communications Module

The Communications Module was developed to provide an interface between the Air Traffic Module and any external ‘third-party’ systems connected to the simulator. External systems, such as CD&R tools, traffic flow management tools, or visualisation tools, can either be implemented natively in ANSI C, or can be connected over a TCP/IP or UDP network.

The Communications Module simulates ADS-B-type surveillance by periodically broadcasting state and velocity data of all aircraft, and transmitting the trajectory intent when the aircraft is initialised or when the flight plan is modified – thereby allowing any connected systems to monitor the traffic. The module can also receive ATC instructions and flight plan modifications, enabling the connected systems to interactively manage the traffic. These communication exchanges can be modelled as datalink or VHF radio-telephone messages, using sub-models of latencies, noise, error rates, broadcast rates, and radio frequency occupation. The Communications Module is discussed further in Chapter 4.

2.4 Operational Data Structures

The simulator has three primary operational data structures:

- The Master Array to define and control the simulation;
- The Aircraft Array to store data between time steps;
- The Data Block to store the traffic state before being written to permanent memory.

2.4.1 The Master Array

The Master Array is necessary to store traffic assignments that are made by, or read by, the Scenario Generation Module during initialisation. The array is used by the Air Traffic Module to determine which new aircraft should enter the simulation, and under what mode of operation. The data fields of the Master Array are described in Table 2-2. The length of the array is set to the total number of aircraft to be simulated, as defined in the configuration file. At the completion of the simulation, the Master Array is written to the permanent record to serve as a log of the air traffic.

Field	Description
Aircraft ID	The unique aircraft identification number
BADA Index	The enumerated reference of the BADA aircraft performance model
Flight Plan Name	The name of the initial flight plan, limited to 20 characters
Start Time	The time the aircraft is to enter active simulation, defined in seconds from the beginning of the simulation where $t = 0$
Finish Time	The timestamp of when the aircraft was removed from active simulation
Control Mode	Specifies ideal, autopilot or flight director control modes, as discussed in Chapter 3
Communications Mode	Specifies ideal, datalink, or radio-telephony communications mode, as discussed in Chapter 4
Separation Mode	Specifies if the aircraft is self-separating, controlled by a centralised system, or uncontrolled
Aircraft Array Index	The location of aircraft data in the Aircraft Array, to allow quick lookup

Table 2-2: The Master Array

2.4.2 The Aircraft Array

At any given point of a simulation, a subset of the aircraft defined in the Master Array is being actively simulated. These aircraft are allocated an element of the Aircraft Array, a structure to store the dynamic flight data that is generated and used during the simulation. The relationship between the Master and Aircraft Arrays is illustrated in Figure 2-2.

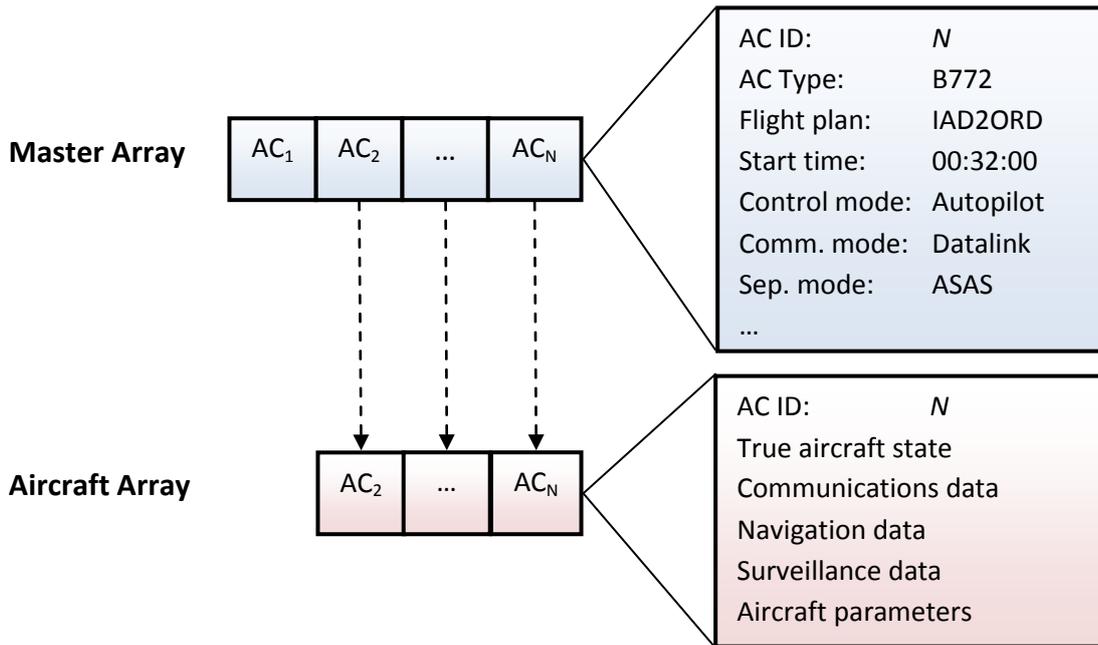


Figure 2-2: Relationship between the Master and Aircraft Arrays

The length of the Aircraft Array is set during initialization to the concurrent aircraft limit. The limit can be specified in the configuration file for up to 2500 aircraft. The Aircraft Array is organised into sub-structures that store aircraft-specific parameters and the instantaneous information-states of the active traffic, shown in Table 2-3. Pointer arguments were used for functions with *struct* arguments (i.e., the Aircraft Array and sub-structures) to ensure fast programme execution through the C pass-by-reference method.

It should be noted that the position and velocity of the aircraft, as well as the local winds, are stored in three locations: in the true aircraft state data, the surveillance data, and the navigation data. This repetition was necessary in order to model distinct, localised information-states. Currently, localized or compartmentalized information is a major source of uncertainty in the ATM system, prompting NextGen and SESAR to call for better system-wide information exchange (Carr, et al., 2005; JPDO, 2009; SESAR, 2007). For example, the estimated aircraft position used in an aircraft navigation computation, as derived by the FMS, may be different from the position estimate used by the CD&R routine, as derived by a multi-sensor surveillance system. The difference between the airborne and ground estimates could result in unpredicted

or undesirable CD&R performance. Maintaining separate copies of position, velocity, and the winds aloft in the navigation and surveillance structures enables the simulator to be used to gain a better understanding of the effects of localised, noisy data.

Field	Description
True Aircraft State	<ul style="list-style-type: none"> - The current position, velocity, and mass of the aircraft - The local winds aloft
Communications Data	<ul style="list-style-type: none"> - Message queue of resolutions waiting to be executed - Parameters such as latency statistics and probability of reception
Navigation Data	<ul style="list-style-type: none"> - Copy of the aircraft flight plan - Own-ship estimation of position, velocity, and winds aloft - FMS operational data such as the current waypoint and distance to go - Reference headings and altitudes for the autopilot system
Surveillance Data	<ul style="list-style-type: none"> - Estimation of the aircraft position and velocity - Estimation of the winds aloft
Aircraft Parameters	<ul style="list-style-type: none"> - Parameters such as aircraft ID, mode of operation, and active flag - Current quadtree index - Pointer to the BADA performance lookup table

Table 2-3: The Aircraft Array

2.4.3 The Data Block

In keeping with the original Airspace Simulator, the Data Block is used to store blocks of traffic data before being written to permanent memory. Feigh demonstrated that the speed of the simulation could be improved by writing large blocks of data at once, rather than separately writing data for every aircraft at every time step. The Data Block is a two-dimensional array, where each element stores the position, velocity, and mass of an aircraft for a given time step. The data structure is described in detail in Appendix B since it forms a part of the output file format. The columns and rows of the Data Block are used to index an aircraft in the Aircraft Array and a time step, respectively. Traffic data for 10 time steps are stored in the Data Block, before being flushed to the permanent record.

However, in order to increase flexibility and reduce memory requirements, a data resolution parameter can be set in the configuration file, defining the number of time

steps between traffic data storage operations. For example, if the data resolution is set to two, then the traffic data from every other time step would be stored in the Data Block; this data would be flushed every 20th time step, halving the memory requirement.

2.5 Logical Design

The program flow and the operational interaction of the core modules is evident from the logical design of the simulator, which has been organised as three distinct phases (Figure 2-3): the setup phase, simulation phase, and the shutdown phase.

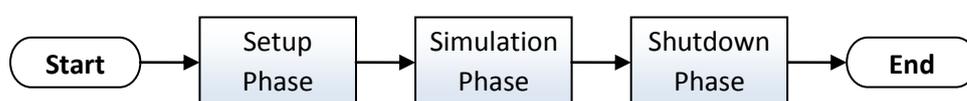


Figure 2-3: The simulator logical design

The setup phase is shown in Figure 2-4. The purpose of the initialisation routine is to read and store all input files (BADA, observed winds, configuration, and flight plans), to allocate and initialise data structures and variables, and to open a random access file as the permanent record. The scenario generation routine then populates the Master Array with pre-defined or pseudo-random traffic assignments.

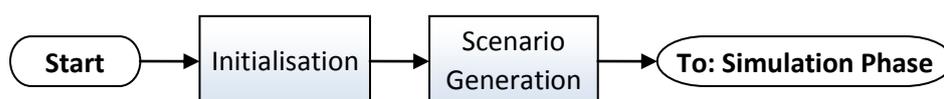


Figure 2-4: The setup phase

The simulation phase is composed of two loops. The outer simulation loop, shown in Figure 2-5, is executed once per time step and contains routines that apply airspace-wide. The inner simulation loop, shown in Figure 2-6, updates the data that is unique to individual aircraft, and is executed once per time step for every aircraft in the Active Array.

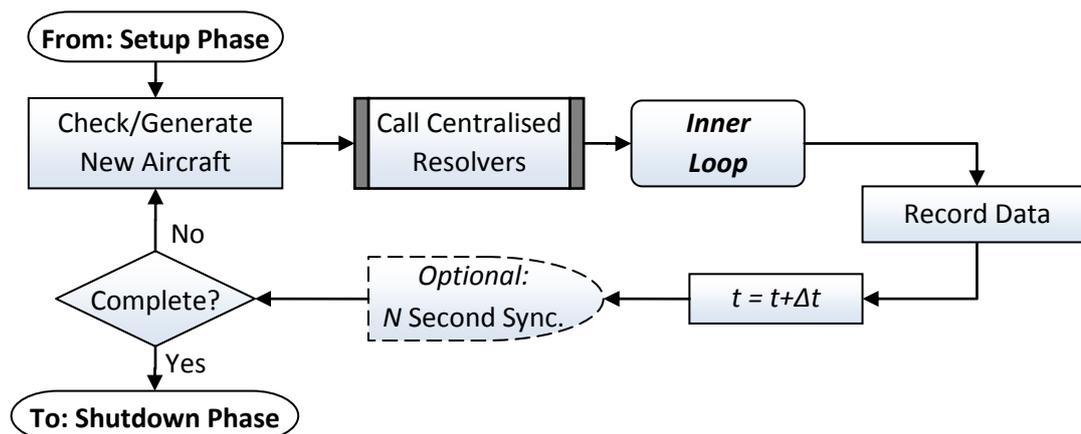


Figure 2-5: The outer simulation loop

The outer loop first checks the Master Array to determine if any new aircraft should be initialised in the airspace. The Master Array is sorted by aircraft start time. Therefore, given the index of the previously activated aircraft, the aircraft defined in the next Master Array element can be initialised if the start time is less than or equal to the current simulation time, t .

If any centralised CD&R tools are connected to the simulator, they are checked to determine if any conflict resolutions have been created for the traffic. Resolutions are sent to the message queues of the receiving aircraft. The inner loop then updates the state and broadcasts the surveillance data of every active aircraft. The updated traffic data is stored in the Data Block, and the Data Block is periodically written to the permanent record.

The simulated time, t , is then advanced by one time step. By default, the simulator executes as fast as possible. However, if required, the speed of the simulator can be slaved to a synchronising clock. For example, for real-time simulation, if Δt is set to 0.25 simulated seconds per time step and if one time step (one pass through the outer simulation loop) is calculated in less than 0.25 actual seconds, then the simulation is delayed until the interval is completed.

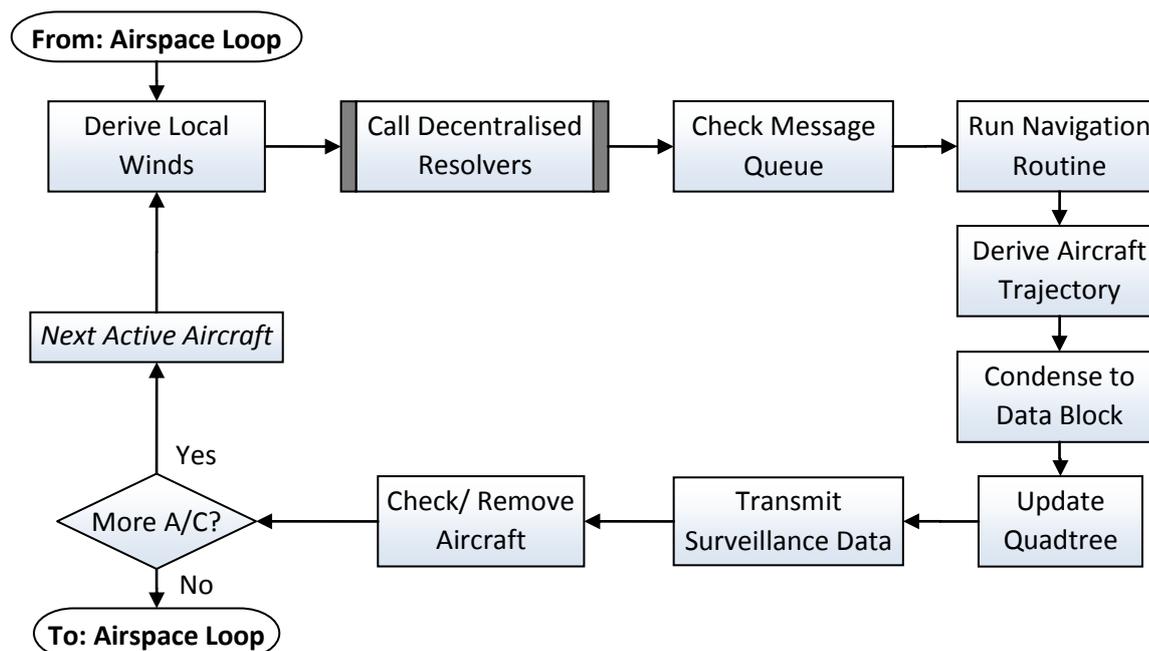


Figure 2-6: The inner simulation loop

The inner simulation loop is composed of the routines that apply to individual aircraft at each time step. First, the direction and magnitude of local winds are interpolated by the wind module. If the aircraft is set to self-separation mode, then the decentralised resolver is called to generate a resolution for any detected conflicts. Any resolutions are stored in the message queue.

The message queue is checked to determine if any ATC instructions or route modification messages in the queue are ready to be passed to the FMS. The navigation routine then produces flight guidance to control the aircraft along the current segment of the flight plan or according to the ATC instructions, and the new aircraft trajectory is computed and stored in the Data Block. The quadtree is checked to determine if the aircraft is within the indicated region. If not, the tree is rebuilt. Next, the surveillance data is broadcast to any external systems integrated with the simulator, such as CD&R, traffic flow management, or visualisation tools.

If the aircraft has completed the final leg of its flight plan, the finish time is recorded in the Master Array and the aircraft is removed from active simulation. The aircraft's element of the Aircraft Array is flagged as inactive, where inactive elements

are skipped by the inner loop. New aircraft can reuse inactive slots, and overwrite the data contained in that array element.

The outer simulation loop resumes upon reaching the end of the Aircraft Array, when all active aircraft have been updated.



Figure 2-7: The shutdown phase

The outer loop of the simulation phase exits when all aircraft have completed their assigned flight plans or when t reaches a time-out value defined in the configuration file. The shutdown phase (Figure 2-7) then writes any remaining traffic data to the random access file, along with the Master Array. The file is closed, completing the simulation.

2.6 Synchronisation with External Systems

A central requirement of the simulator was the capability of integrating both centralised and decentralised CD&R systems with the air traffic. However, the specific software implementation of the connected tools will determine the ease and extent of integration that is possible. CD&R systems that are connected over a network or deployed as a 'stand-alone' programme may be more difficult to integrate than systems implemented natively in C. It cannot be assumed that such external systems adhere to a standardised interface, and furthermore, the software may be in varying stages of development or have varying degrees of accessibility. So, in order to account for different software implementations, it is important that the integration method be as flexible as possible and minimise the modifications required of the connected systems.

One challenge of integrating CD&R systems to a fast-time simulator is maintaining time synchronisation. If time synchronisation is not maintained, the simulator may

advance at a faster rate than the CD&R systems are able to detect conflicts and calculate resolutions. Two connection modes were developed to accomplish time synchronisation while preserving flexibility: speed-control and handshaking.

With the speed-control mode, synchronisation is maintained by limiting the simulation speed to the fastest speed supported by the connected system. For example, if it is known that under certain conditions the CD&R software can run at least 2 times faster than real time, then the slaved simulator speed should be set to 0.5 seconds per simulated seconds. If 0.5 simulated seconds are computed in less than 0.5 actual seconds, then the simulation is delayed.

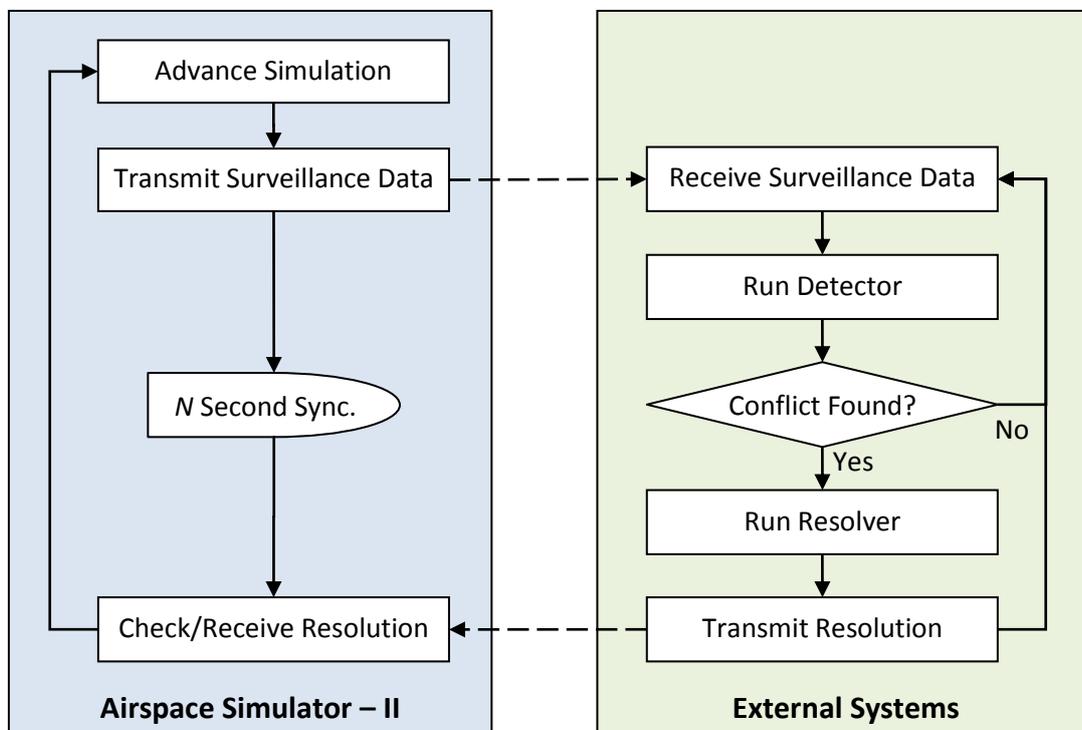


Figure 2-8: Speed-controlled synchronisation mode

The speed-control mode is simple because it does not require the connected system to incorporate any handshaking routines with the simulator. Rather, both the simulator and the connected system are allowed to execute independently, as shown in Figure 2-8. The simulator periodically transmits traffic surveillance data and periodically checks a message buffer to determine if any resolutions are waiting to be received. The conflict detection routine of the CD&R system runs continuously in

parallel. If at any point a conflict is detected, then a resolution is generated and transmitted to the simulator.

The drawback of the speed-control mode, however, is that the speed of the simulation is restricted to the 'worst-case' speed of the connected systems. The connected systems may actually be able to process faster than the constrained simulator speed. So, a handshaking mode was also developed in order to permit the simulator and any connected systems to run as fast as possible. In this mode, the simulator and connected systems run in series, using handshaking to maintain synchronisation, as shown in Figure 2-9. Both systems are allowed to execute at full speed when not waiting for handshake. However, the external systems must also be capable of implementing the handshaking procedure.

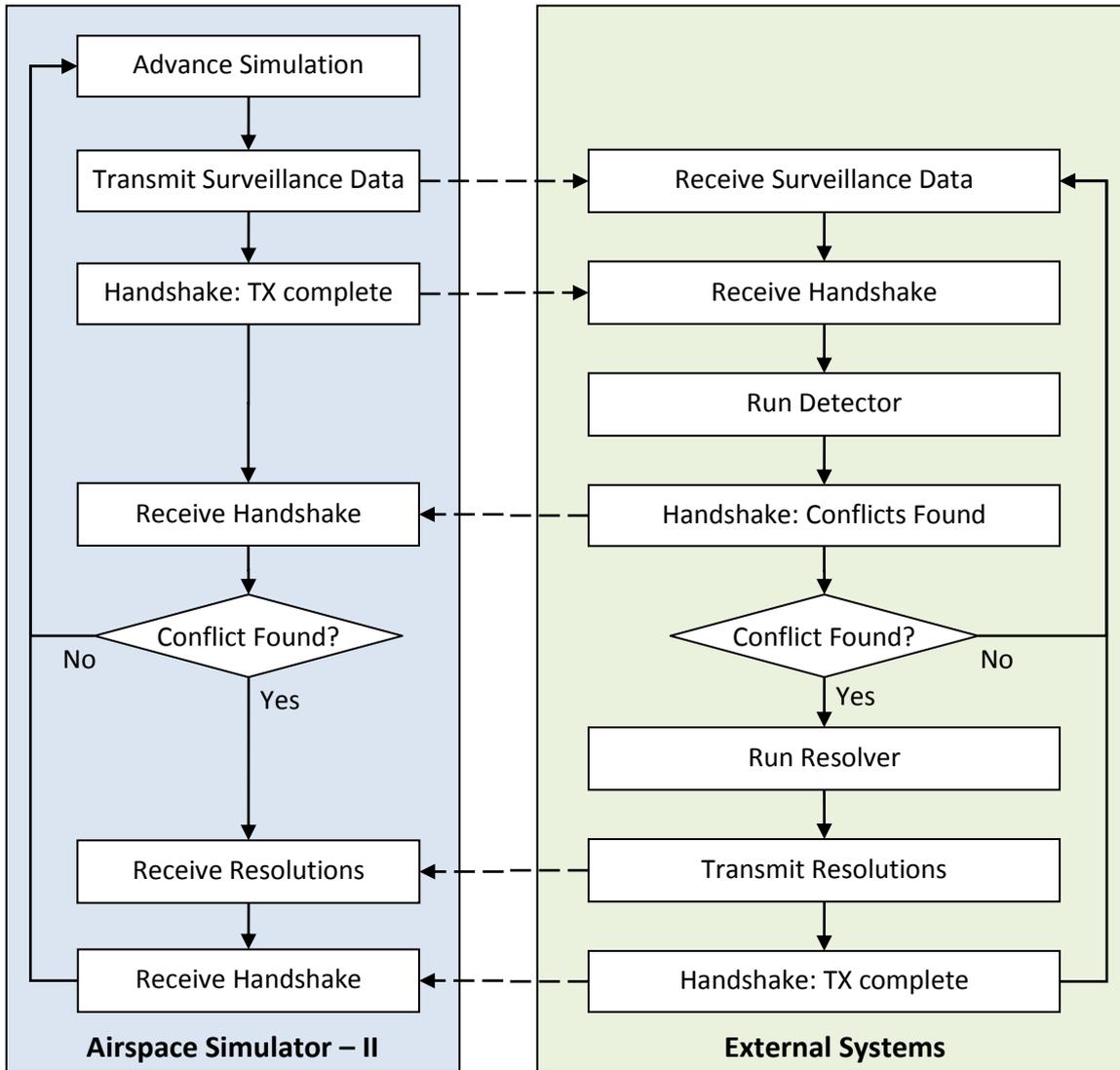


Figure 2-9: The handshake synchronisation mode

In both connection modes, the overall speed of the simulation will be directly affected by the performance of the connected systems. Thus, algorithm and software efficiency are important considerations when developing CD&R systems, in order to facilitate fast-time testing and analysis.

2.7 Simulator Outputs

2.7.1 Permanent Record

Simulation data is periodically recorded to a binary-format random access data file for post-simulation analysis. The file consists of a header, the master array, and the time series of traffic state data from the Data Blocks. The details of the file format are described in Appendix B.

At initialisation, the file pointer is offset to allocate space so the header and Master Array can be written at the beginning of the file during the simulator shutdown phase. The header records the basic simulator settings such as the time step size and wind source. In addition, the header stores global simulation results, including the total number of simulated aircraft, the duration of the simulation, and the programme execution time. The Master Array is then written to the file as a record of the traffic modes, flight plans, and aircraft models. The Data Blocks that are periodically written to the random access data file form a time series of traffic state data. Consequently, the Master Array also contains the traffic start and finish times and the index of the Aircraft Array, enabling the time series to be efficiently parsed.

The size of the recorded data is primarily a function of the number of events recorded and the number of concurrently simulated aircraft. The number of events recorded is the total number of time steps, divided by the data resolution. For example, simulation of a one hour traffic scenario with a one second time step and a two second data resolution will result in $3600/(2 \times 1) = 1800$ recorded events. A series of test simulations were conducted for a range of recorded events and concurrent aircraft counts. The resulting file sizes are plotted in Figure 2-10. It can be seen that the permanent record can become very large (>1 GB) for a large number of aircraft and recorded events. A warning is raised during initialisation if the file size is estimated to be greater than 700 MB, in order to ensure that the results of simulation can be written to one standard CD. However, the size limit can be adjusted as required.

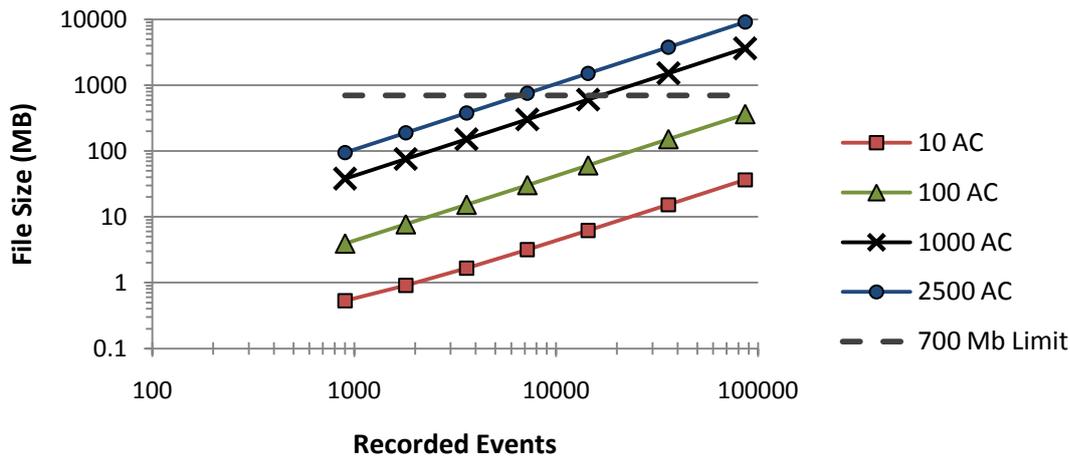


Figure 2-10: Log-log plot of permanent record file size

In order to evaluate the efficiency of the data recording method, the cumulative time elapsed in the data recording process over the course of a simulation (copying aircraft data to the Data Block, writing the Data Block to the file, as well as writing the header and Master Array) was measured for a range of recorded events and concurrent aircraft counts. These tests were conducted on a laptop computer with 2 GB RAM and a 2.2 GHz dual core CPU. Figure 2-11 shows the averages of 5 repeated trials. The tests did not include cases where the file size was greater than 700 MB.

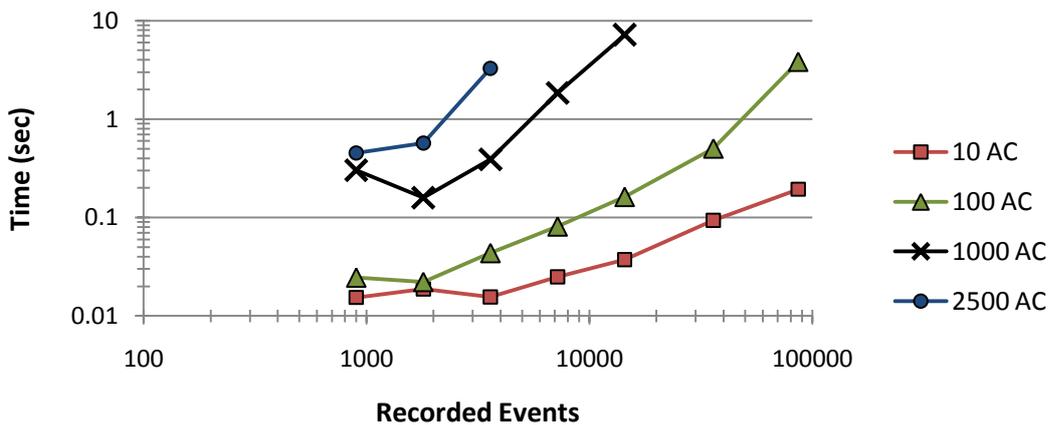


Figure 2-11: Log-log time profile of data recording

On average, the data was recorded in less than 0.7 seconds. As expected, the results indicate that the elapsed time generally increases with both the number of recorded events and the number of concurrent aircraft. However, even for 86400

recorded events (corresponding to 24 hours of simulated time with a 1 second time step) for 100 concurrent aircraft, the average elapsed time in the data recording process for the 5 trials was less than 5 seconds. The worst case standard deviation of the 5 trials was 1.07 seconds, but the mean of the standard deviations was only 0.12 seconds. These results indicate that the data recording method is efficient, and will typically consume less than 10 seconds.

2.7.2 Traffic Visualisation

Graphically displaying the air traffic as the simulation progresses can be a helpful tool for analysis. To provide visualisation, the Java-based *Tviz* mapping application, developed by Spence, was connected to the simulator using UDP network protocol (2009). The speed control connection mode was used because the visualisation becomes incomprehensible if the simulator is allowed to run at full speed. The synchronising speed is set to ten times faster than real-time if *Tviz* is used. Using the surveillance data output from the Communications Module, *Tviz* provides an ATC-like plan view display of aircraft, flight plans, as well as the locations of airports and navigation aids, as shown in Figure 2-1. Additionally, the *Tviz* user interface can show range rings, linear predictions of aircraft position, and distance/bearing lines. Integrating *TViz* highlights the advantage of the modular design and flexible connectivity of the simulator – the functionality of the simulator was expanded by connecting to a third-party, stand-alone module.

- the addition of navigation, communication, and separation mode specifications
- the addition of compartmentalized aircraft state information for the surveillance and navigation models
- the addition of the communications message queue

The logical design was also based on Feigh's work, but with the addition of the time-synchronisation capability, surveillance broadcasting, and monitoring for conflict resolution messages.

A further major change was the ability to network the simulator with external tools such as TViz and CD&R systems. Many other smaller modifications were made, including the addition of the data recording resolution parameter, provision for constant wind fields, and simplification of the configuration file. Components that remain largely unchanged include the BADA performance modelling, the quadtree to spatially-organise the traffic, and the wind modelling, representing 15% of the physical source lines of code.

Chapter 3

Navigation and Trajectory Module

The purpose of the Navigation and Trajectory Module is to simulate the performance and navigation of modern commercial aircraft and to compute their flight trajectories at every time step. Section 3.1 elaborates on the requirements that are specific to the Navigation and Trajectory Module. Section 3.2 then presents an overview of the method used, and Sections 3.3 through 3.7 discuss the design of the module in detail. The module is then evaluated in Section 3.8.

3.1 Requirements for Modelling Navigation and Trajectories

Future ATM systems will afford improved navigational accuracy in comparison with current operations. Given a series of Earth-referenced waypoints defining the centreline of the flight path, Precise-RNAV specifications for en route procedures under NextGen and SESAR require aircraft to remain within navigational bounds ± 1 nmi of the centreline for 95 percent of the flight time, as illustrated in Figure 3-1. The Navigation and Trajectory Module must be capable of guiding aircraft along the flight plan within these bounds. In addition to this basic navigation requirement, the module should be capable of modelling navigational errors and uncertainties in order to simulate realistic traffic behaviour when evaluating CD&R concepts and systems.

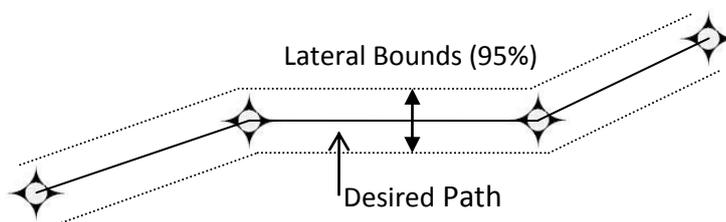


Figure 3-1: Illustration of lateral navigational accuracy bounds

Furthermore, the module must be capable of responding to conflict resolutions issued by CD&R systems. Common conflict resolution manoeuvres include:

- Speed assignments

- Altitude assignments
- Heading vectors
- Direct-to clearances
- Lateral offset assignments
- Sequences of waypoints

Speed and altitude changes are common conflict resolution methods used by air traffic controllers as well as automated CD&R systems (Kirk, et al., 2001; Paielli, 2008; Erzberger, 2006). Similarly, heading vectors are used by air traffic controllers for lateral conflict resolution. Heading vectors instruct aircraft to suspend lateral waypoint navigation and follow a constant specified heading. When the aircraft is clear of the conflict, the flight is permitted to resume its flight plan. Some automated CD&R systems, such as the TSAFE system, also generate heading vectors to resolve short range conflicts (Erzberger & Heere, 2010).

Direct-to clearances instruct the aircraft to proceed directly from its current position to a specified waypoint, as shown in Figure 3-2. This instruction is a useful resolution method in terminal environments because it allows aircraft to ‘cut the corner’ of a flight plan, altering the lateral flight profile and reducing flight delays (Karr, et al., 2009).

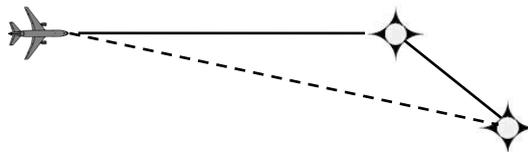


Figure 3-2: Direct-to manoeuvre

Lateral offset resolutions instruct the aircraft to fly a route parallel to a flight plan segment, offset by a given distance, as shown in Figure 3-3. Although many FMS have this capability, lateral offsets are rarely used in current ATM systems. However, Herndon, et al., have argued that lateral offsets could play a central role in en-route conflict resolution in future ATM systems, allowing trailing aircraft to overtake leading aircraft (Herndon, et al., 2004).

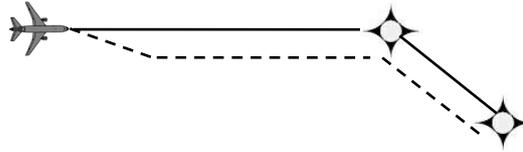


Figure 3-3: Lateral offset manoeuvre

Data link and increasing flight deck automation also make it possible for CD&R tools to precisely define a resolution manoeuvre using a sequence of waypoints (Karr, et al., 2009). For example, path-stretch manoeuvres (also known as turn-point manoeuvres) use two waypoints to create a ‘delay leg’ in the flight plan (Bach, et al., 2009). The aircraft is directed off the original path at the start point and then cleared direct to the active waypoint upon crossing the turn back point, as illustrated in Figure 3-4.

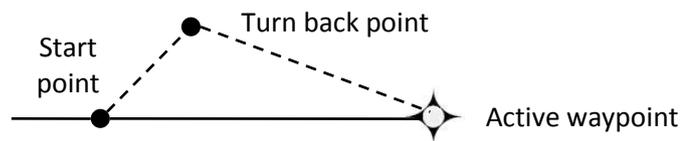


Figure 3-4: Path stretch manoeuvre

The simulator should be capable of modifying the original flight plan with any of the resolution methods described above.

3.2 Trajectory Modelling Overview

The precise navigation that is called for in future ATM systems is made possible through advanced avionics, which includes Flight Management Systems and autopilots, that can assist the flight crew in managing many aspects of the flight, from flight planning, to guidance and control. A modern FMS permits the crew to input objectives and constraints for the flight via the flight plan and direct entries; the FMS can then calculate a lateral and vertical/energy profile of an optimised trajectory. In flight, the FMS produces guidance to follow this trajectory, in order to achieve the objectives, given the constraints. When coupled to an Autopilot/Flight Director System (APFDS), the flight guidance produced by the FMS can be used by the APFDS to control

the aircraft. Autopilots provide automatic control of pitch and roll according to the selected mode of operation, such as tracking the FMS-defined trajectory. Alternatively, the flight director can present the pitch and roll cues directly to the crew, allowing the crew to manually control the flight along the desired trajectory. Longitudinal control is provided by an auto-throttle system or manual throttle inputs.

In order to simulate this behaviour, as well as ensuring that the navigation and trajectory modelling process is flexible, accessible, and easy to modify, it was decided to separate the flight guidance, control, and dynamics functionalities into sub-modules. The resulting process used to simulate aircraft trajectories is shown in Figure 3-5, and includes the aircraft performance model based on the EUROCONTROL Base of Aircraft Data (BADA), flight guidance by the Flight Management System, flight control by the Autopilot/Flight Director System and flight dynamics derived from the equations of motion. The desired trajectory is defined by flights plans and ATC instructions. Disturbances to the system are introduced by wind, navigation errors and state estimation errors.

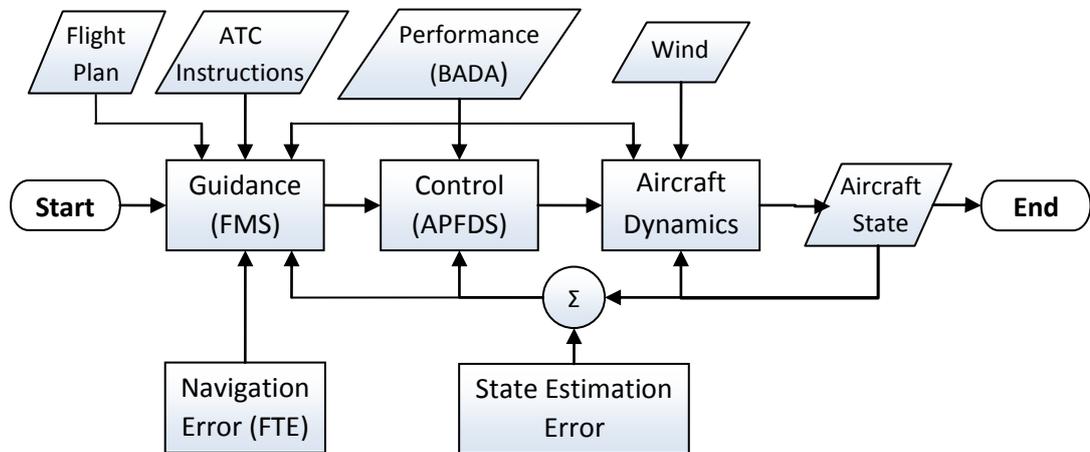


Figure 3-5: The navigation and trajectory modelling process

Spherical earth navigation was assumed to avoid the added computational complexity of ellipsoidal equations. All navigation calculations were performed using double-precision floating-point numbers to minimise rounding errors in trigonometric calculations.

3.2.1 Flight Plans

Every simulated aircraft is assigned a flight plan, which is read from an ASCII text file, consisting of a sequence of flight plan segments that describe the intended trajectory for the duration of the segment, where each segment in the flight plan contains the following data:

Segment name: An 8-character segment identifier.

Phase of flight: One of the following 3-character phase of flight identifiers: departure (DEP), en-route (ENR), manoeuvring (MNV), terminal (TRM) or missed approach (MAP). This value is used when determining manoeuvring limitations, as discussed in Section 3.4.1.1.

Segment Type: Three segment types have been defined: Initial Fix (IF), Track-to-Fix (TF) and Direct-to-Fix (DF) segments. The Initial Fix segment is always the first line of the flight plan, and defines the starting coordinates and altitude of the aircraft. The remaining lines consist of Track-to-fix and Direct-to-fix segments which define Great Circle arcs between two waypoints (fixes). For TF segments, the arc is measured from the previous waypoint in the flight plan to the segment waypoint. DF segments are used to implement direct-to instructions; the DF arc is measured from the aircraft position (at the point the DF segment is activated) to the segment waypoint.

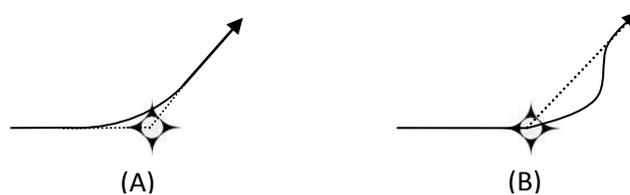


Figure 3-6: Fly-by waypoint (A), and fly-over waypoint (B)

Transition Type: Specifies how an aircraft transitions to the following segment. For fly-by transitions (FB), the FMS anticipates the turn to the next segment. For fly-over

transitions (FO), the aircraft overflies the waypoint before beginning the turn to the next segment, as shown in Figure 3-6.

Waypoint Coordinates: The latitude and longitude of the segment waypoint, in degrees and decimal minutes.

Segment Altitude: The target altitude in feet of the segment above mean seal level.

Segment Speed (optional): The target true airspeed of the segment, in knots. If not specified, the target airspeed is set from the BADA performance data.

Lateral Offset (optional): The target parallel offset distance in nautical miles, using the convention of positive for right of path, and negative for left of path.

Flight plans are read into the simulator during initialisation and stored in memory as a doubly-linked list, where the data structure for every segment contains pointers to the next and the previous segments, as illustrated in Figure 3-7. Linked lists enables the route to be easily modified in flight by inserting or removing segments and also minimises the memory used in comparison to allocating space for a fixed number of flight plan segments. The flight plan format is described in Appendix C.

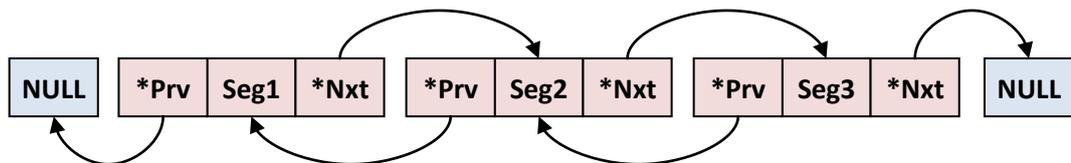


Figure 3-7: Linked list data structure

3.2.2 ATC Instructions

In addition to basic waypoint-to-waypoint navigation, the trajectory modelling process must implement conflict resolutions that are not referenced to underlying waypoints. Therefore, an input to the Navigation and Trajectory module is ATC Instructions, which temporarily override navigation along the lateral, vertical, and/or longitudinal modes of the flight plan. Lateral instructions include constant heading, constant track angles or a lateral offset. Vertical and longitudinal instructions include

altitude and true airspeed. The instructions are terminated either upon receipt of another instruction or automatically after the optionally-specified instruction duration.

By combining route modifications and instructions, it is possible to construct each of the required conflict resolution manoeuvres. For example, turn-point manoeuvres can be implemented as a route modification by merging two TF segments into the flight plan. Heading vectors can be simulated by a heading instruction followed by a DF segment when the aircraft is clear of the conflict; a demonstration of this is shown at the end of the chapter in Figure 3-22.

3.3 Performance Modelling

The BADA aircraft performance models, developed and maintained by the EUROCONTROL Experimental Centre (EEC), are widely used in fast time air traffic simulations and trajectory prediction tools (Suchkov, et al., 2003; Fairley & McGovern, 2009; Alam, et al., 2008; Mayer, 2002; Signor, et al., 2004). The simulator now uses version 3.8 of the database, containing the operational performance parameters and standard airline operating procedures for 111 of the most common aircraft types (Nuic, 2010). In addition, 207 types have been identified as having equivalent performance to one of the directly modelled aircraft; effectively allowing up to 318 aircraft types to be simulated, covering over 98% of the 2008-2009 European air traffic (Sheehan, 2009).

BADA uses the kinetic approach of trajectory modelling, where the equations of motion are simplified by independently modelling thrust and drag, rather than the full set of differential equations. This simplification can be justified for relatively small flight path angles typical of commercial transport aircraft (Suchkov, 2003). A kinetic approach allows aircraft to be modelled with a reduced point-mass equation called the Total Energy Model that equates the rate of work done by forces acting on the aircraft to the rate of increase in potential and kinetic energy

$$(T - D)V_{TAS} = mgh + mV_{TAS}\dot{V}_{TAS} \quad (\text{Eq. 3-1})$$

where T is thrust along the velocity vector, D is aerodynamic drag, V_{TAS} is true airspeed, m is aircraft mass, g is gravitational acceleration, h is geodetic altitude, and the dot accent mark indicates the time derivative (Nuic, 2010). The database contains the coefficients needed to derive the thrust and drag for various conditions, effectively allowing any two of the three variables of thrust, speed, or rate of climb/descent to be controlled.

The BADA data is also published in kinematic form by solving Equation 3-1 *a priori* in the form of look-up tables of airspeeds, rates of climb/descent and fuel consumption at various flight levels and phases of flight. Using the tabulated form of the BADA data significantly reduces the computational complexity of generating aircraft trajectories and increases the speed of the simulation, although at the cost of reduced fidelity. This compromise was considered acceptable for the purposes of this research because the typical flight profiles of commercial transport aircraft are dependent on the use of flight management systems and automatic flight controllers resulting in relatively consistent and parameterised trajectories, reducing the errors of the model simplifications. Furthermore, the tabulated data approach simplifies the simulation by removing the need to input extensive flight data such as airline procedure and speed schedules.

3.4 The Flight Management System

Lateral, vertical, and longitudinal flight guidance along the flight plan or in response to ATC Instructions is provided by the Flight Management System model, which calculates commanded headings, altitudes and speeds for the flight control system.

3.4.1 Lateral Guidance

The FMS was designed to provide a heading reference for the flight controller, ψ_{ref} , to guide the aircraft along the heading or track of the active flight plan segment.

For heading instructions, ψ_{ref} is simply the heading specified in the instruction. For track angle instructions, the specified track angle, θ_{ref} , can be converted to heading guidance, ψ_{ref} , by correcting for wind drift as follows. The wind correction angle, WCA , can be found from the standard wind triangle

$$WCA = \sin^{-1} \left(\frac{v_{Wnd} \sin(\theta_{Gnd} - \theta_{Wnd})}{v_{TAS}} \right) \quad (\text{Eq. 3-2})$$

where (v_{Gnd}, θ_{Gnd}) is the aircraft's inertial velocity vector, (v_{Wnd}, θ_{Wnd}) is the wind velocity vector, and (v_{TAS}, ψ) is the aircraft's air-mass velocity vector. The reference heading can then be calculated by adding the wind correction angle to the desired track angle, normalising the resulting angle to $\pm\pi$ radians.

$$\psi_{ref} = \text{norm}(\theta_{ref} + WCA, \pm\pi) \quad (\text{Eq. 3-3})$$

Course guidance for flight plans is more complex, because, with the exception of flights along the equator and meridians, Great Circle paths do not follow a constant course. Rather, the actual course varies as the aircraft follows the path. The desired course, $\theta_{\text{Along Trk}}$, at any instantaneous point along the flight plan path can be found by the following method.

3.4.1.1 Great Circle Navigation

From the Spherical Law of Cosines, the Great Circle distance between any two points is given by

$$d_{A,B} = \cos^{-1}(\sin(\phi_A)\sin(\phi_B) + \cos(\phi_A)\cos(\phi_B)\cos(\lambda_B - \lambda_A))R_o \quad (\text{Eq. 3-4})$$

where ϕ_n is the latitude and λ_n is the longitude of point n , and R_o is the radius of the earth = 3440.655273 nmi (Paielli, 2005). Sinnott has argued that the Haversine formula is better suited for computing Great Circle distances because the inverse cosine function is not well conditioned for small distances due to rounding errors (Sinnott, 1984). However, the errors are less than 1 metre when using double precision floating-point numbers on computers with a 32bit word size. Additionally, the Haversine formula is more complex, requiring more than twice the computational time of Equation 3-4.

The initial bearing can be found as

$$\theta_{A,B} = \text{atan2} \left(\begin{array}{c} \sin(\lambda_B - \lambda_A) \cos(\phi_B), \\ \cos(\phi_A) \sin(\phi_B) - \sin(\phi_A) \cos(\phi_B) \cos(\lambda_B - \lambda_A) \end{array} \right) \quad (\text{Eq. 3-5})$$

where $\text{atan2}()$ is the four-quadrant arctangent(y/x) function (De Smith, et al., 2009).

Using Equations 3-4 and 3-5 and Napier's Rules for right spherical triangles, the cross track error, δ_{XTE} , along-track distance, d_{AT} , as shown in Figure 3-8, can be computed

$$\sin\left(\frac{\delta_{XTE}}{R_o}\right) = \sin(\theta_{A,Ac} - \theta_{A,B}) \sin\left(\frac{d_{A,Ac}}{R_o}\right) \quad (\text{Eq. 3-6})$$

$$\cos\left(\frac{d_{A,Ac}}{R_o}\right) = \cos\left(\frac{d_{AT}}{R_o}\right) \sin\left(\frac{\delta_{XTE}}{R_o}\right) \quad (\text{Eq. 3-7})$$

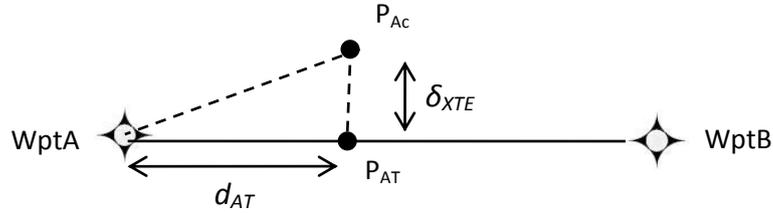


Figure 3-8: Great Circle navigation geometry

The location of the along track point, P_{AT} , can then be found by projecting a point from $WptA$ a distance d_{AT} with the initial bearing $\theta_{A,B}$ using

$$\phi_{AT} = \sin^{-1} \left(\sin(\phi_A) \sin\left(\frac{d_{AT}}{R_o}\right) + \cos(\phi_A) \sin\left(\frac{d_{AT}}{R_o}\right) \cos(\theta_{A,B}) \right) \quad (\text{Eq. 3-8})$$

$$\lambda_{AT} = \lambda_A + \text{atan2} \left(\begin{array}{c} \sin(\theta_{A,B}) \sin\left(\frac{d_{AT}}{R_o}\right) \cos(\phi_A), \\ \cos\left(\frac{d_{AT}}{R_o}\right) - \sin(\phi_A) \sin(\phi_{AT}) \end{array} \right) \quad (\text{Eq. 3-9})$$

Thus, the along track course, $\theta_{\text{Along Trk}}$, can be found from this point as $\theta_{AT,B}$.

A reference course, θ_{ref} , can then be computed using a linear control law that minimises the cross track error between the aircraft position and the flight plan segment

$$\theta_{ref} = \left\{ \begin{array}{ll} \theta_{Along Trk} - \frac{\pi}{2} * \frac{\delta_{XTE}}{r}, & \text{for } \left| \frac{\delta_{XTE}}{r} \right| < 1 \\ \theta_{Along Trk} - \frac{\pi}{4} * \text{sign}(\delta_{XTE}), & \text{for } \left| \frac{\delta_{XTE}}{r} \right| > 1 \end{array} \right\} \quad (\text{Eq. 3-10})$$

where $sign()$ returns +1 or -1 according to the sign of the number and r is the turn radius (Peters & Konyak, 2003).

The turn radius, r , of a coordinated, constant bank turn is

$$r = \frac{v_{TAS}^2}{g * \tan(\phi)} \quad (\text{Eq. 3-11})$$

where ϕ is the nominal bank angle, and g is gravitational acceleration (Mondoloni, 2006). The actual bank angle varies according to operator preferences and the phase of flight. The simulator uses the EUROCONTROL recommended schedule of 15 degrees for departure, en-route, missed approach flight and 25 degrees for terminal and manoeuvring flight, as specified in the segment data of the flight plan (2003).

The control law in Equation 3-10 limits the interception angle to ± 45 degrees for δ_{XTE} distances greater than half the turn radius, proportionally reducing the intercept angle until δ_{XTE} is zero. The reference course is converted to heading guidance, ψ_{ref} , for the flight controller by correcting for the local winds using Equations 3-2 and 3-3.

3.4.1.2 Turn Anticipation

Fly-by turns must be initiated before the waypoint in order to complete the turn on the next path. The turn anticipation distance can be derived from the manoeuvre geometry shown in Figure 3-9.

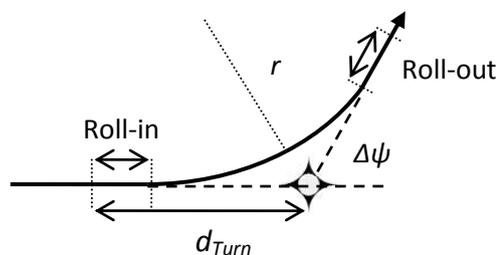


Figure 3-9: Constant bank turn

Turning manoeuvres of commercial air traffic at cruising altitudes and airspeeds are often assumed to be constant bank turns, allowing them to be constructed as a circular arc with radius given by Equation 3-11 (Mondoloni, 2006; Nuic, 2010). Assuming a linear bank rate, the distance required to establish the full bank angle is

$$d_{Roll-in} = v_{TAS} * \frac{\phi}{\dot{\phi}} \quad (\text{Eq. 3-12})$$

where $\dot{\phi}$ is the bank rate. One simplifying approximation is to assume the heading does not change until the nominal bank angle is achieved. Mondoloni has shown that omitting heading change during roll-in and roll-out is insignificant to the predicted position at the completion of the turn; the position error is typically less than 35 feet (2006). As with the bank angle, actual bank rates vary by operator preference and by the control mode. The simulator uses a bank rate of 3 degrees per second for aircraft assigned the *Ideal* or *Autopilot* control mode and 5 degrees per second for aircraft assigned the *Flight Director* control mode, as per EUROCONTROL recommendations (2003).

The turn anticipation distance, d_{Turn} , for a given heading change, $\Delta\psi$, is then

$$d_{Turn} = \tan\left(\frac{\Delta\psi}{2}\right) r + v_{TAS} \frac{\phi}{\dot{\phi}} \quad (\text{Eq. 3-13})$$

The turn is initiated and the current flight plan segment is sequenced once the distance from the aircraft to the active waypoint is less than d_{Turn} .

By definition, turn anticipation is not used for fly-over waypoints; rather the segment is sequenced when the aircraft crosses a line perpendicular to the active segment course at the terminating waypoint.

3.4.1.3 Lateral Offsets

Lateral offsets can be considered as ‘desired’ cross track error; so, to provide an offset capability, the desired offset value, d_{Offset} , is subtracted from the calculated cross track error value from Equation 3-6. Thus, in minimising the remaining cross track error, the lateral path control law (Equation 3-10) will guide the aircraft along the track, offset by a constant distance.

However, when a lateral offset is used in conjunction with fly-by turns, the turn anticipation distance must be corrected, as illustrated in Figure 3-10.

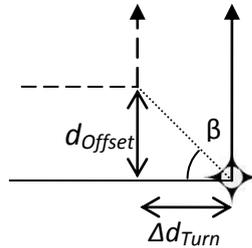


Figure 3-10: Turn correction for lateral offsets

The correction distance, Δd_{Turn} , can be found from Napier's Rules for right spherical triangles

$$\tan\left(\frac{d_{Offset}}{R_o}\right) = \sin\left(\frac{\Delta d_{Turn}}{R_o}\right) \tan(\beta) \quad (\text{Eq. 3-14})$$

where β is half the interior course change angle. The correct turn anticipation distance is the sum of d_{Turn} and Δd_{Turn} .

3.4.2 Longitudinal Guidance

The output reference airspeed, V_{TASref} is found by interpolating the BADA lookup tables for the current altitude and phase of flight. The data in the tables was compiled using speed and power profiles corresponding to common airline operational procedures, including:

- A 250 knot calibrated airspeed (CAS) limit below 10,000 ft;
- Constant CAS climb/descent between 10,000 ft and the Mach transition altitude (typically around 30,000 ft);
- Constant Mach climb/descent above the Mach transition altitude;
- Reduced power climb settings up to 80% of the aircraft ceiling.

It was also important for the simulator to be able to implement speed resolutions. For example, if an airspeed value is specified in the flight plan segment data or by an ATC instruction, then V_{TASref} should be set to that speed. However, setting abstract reference airspeeds highlights a limitation of using tabulated performance data. The total energy equation (Eq. 3-1) shows that thrust, airspeed, and vertical speed are

interdependent. Thus, if the airspeed is altered from the table values, the corresponding table values for either the fuel flow or rate of climb/descent will be inaccurate, because two of the three variables of thrust, speed, or rate of climb/descent can be controlled (fuel flow is a function of the thrust). Despite this known inaccuracy, it was decided to allow abstract reference airspeeds in the simulator because there are occasions where it is useful to control the speed independent of climb/descent or fuel flow inaccuracies, for example in evaluating speed-based resolvers for horizontal conflicts.

3.4.3 Vertical Guidance

As discussed above, the rate of climb and descent cannot be controlled independently from the airspeed and fuel flow. Consequently, the vertical guidance function of the FMS sets the altitude specified in the active flight plan segment or the ATC instruction as the reference altitude, h_{ref} , for the flight controller. In keeping with current operational procedures, the aircraft is then permitted to climb or descend at an optimal rate; in this case, the rate defined in the performance tables until h_{ref} is met (FAA, 2004). Because the flight controller includes a vertical speed control loop, it would be possible to further develop the FMS to provide vertical speed guidance.

3.5 The Autopilot/Flight Director System

The simulator APFDS models the lateral, vertical, and longitudinal flight control behaviour of pilots and autopilots using the proportional feedback controllers shown in Figures 3-11, 3-12, and 3-13. Proportional control was chosen because the system was observed to be stable and responsive over the operating ranges, thereby marginalising the advantages of derivative, integral, or other more advanced control methods given their additional complexity (Allerton, 2009).

3.5.1 Lateral Control

The lateral flight controller uses the two control loops shown in Figure 3-11 to hold the reference heading, ψ_{ref} , from the FMS module. The output is the new aircraft bank angle, used to derive the aircraft heading by the equations of motion.

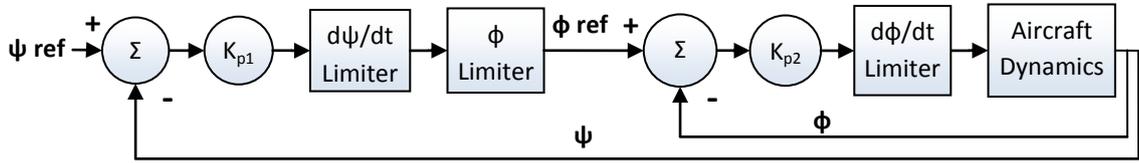


Figure 3-11: The lateral flight controller

The K_{p1} term is used to control the turn rate, $\dot{\psi}$, from the error between the reference and actual aircraft heading. The turn rate is then limited and converted to a bank angle, ϕ_{bank} . The turn rate of commercial air traffic is typically limited to ± 3 degrees (Mondoloni, 2006). The relation between the bank angle and turn rate is

$$\phi_{bank} = \text{atan} \left(\frac{\dot{\psi} \cdot V_{TAS} \cdot \cos(\gamma)}{g} \right) \quad (\text{Eq. 3-15})$$

where γ is the flight path angle. However, for small flight path angles and given that both the turn rate and bank angle are limited, Equation 3-15 can be approximated as (Allerton, 2009)

$$\phi_{bank} = \frac{\dot{\psi} \cdot V_{TAS}}{g} \quad (\text{Eq. 3-16})$$

For flight path angles between ± 10 degrees and turn rates between ± 3 degrees, the worst case approximation error is 3.7% at 250 knots and 3.3% at 500 knots true airspeed. As discussed in Section 3.4.1.1, the bank limit is set to 15 degrees for departure, en-route, missed approach flight and 25 degrees for terminal and manoeuvring flight.

The K_{p2} term is then used to control the bank angle rate from the error between the reference and current aircraft bank angle. The bank rate is limited to 3 degrees per second for aircraft assigned the *Ideal* or *Autopilot* control mode and 5 degrees per second for aircraft assigned the *Flight Director* control mode, and is then integrated with the current aircraft bank angle to derive the new bank angle. The controller was manually tuned to $K_{p1} = 0.07$ and $K_{p2} = 0.75$, which showed satisfactory response,

damping and stability for airspeeds between 50 and 500 kts and bank angles between 15 and 25 degrees with a time step of 1 second.

3.5.2 Longitudinal Control

The speed controller that was designed for the simulator is shown in Figure 3-12.

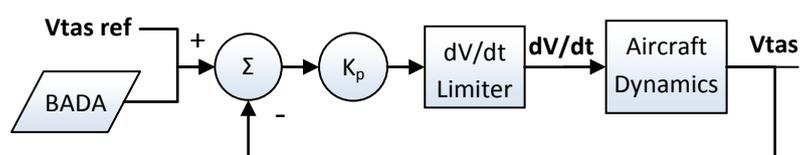


Figure 3-12: Longitudinal flight controller

As discussed in Section 3.4.2, the reference airspeed input is provided by either the flight plan segment data or the BADA database. A longitudinal acceleration, dV/dt , is commanded through the K_p term and the airspeed error, proportionally reducing the acceleration to zero as the aircraft approaches the reference airspeed.

The BADA user manual recommends that longitudinal acceleration be limited to 2 f/s^2 (1.1850 knots/s ; 0.6096 m/s^2) (Nuic, 2010). However, Mondoloni suggests that a more typical value for acceleration in commercial air traffic for small speed adjustments is 0.69 f/s^2 (0.4 knots/s , 0.2068 m/s^2) (2006). So, when the difference between the current speed and the reference speed is less than 8 knots, the limit is lowered to 0.69 f/s^2 , otherwise, the limit is 2 f/s^2 . The longitudinal acceleration is then integrated to derive the new airspeed. The speed controller was manually tuned to $K_p = 0.8$, which showed satisfactory response, damping and stability for airspeeds between 50 and 500 kts with a time step of 1 second.

3.5.3 Vertical Control

The vertical flight controller is equivalent to the autopilot height hold function. The two control loops shown in Figure 3-13 maintain the reference altitude, h_{ref} , provided by the FMS module. The output is the new flight path angle, γ , which is used to derive the aircraft altitude.

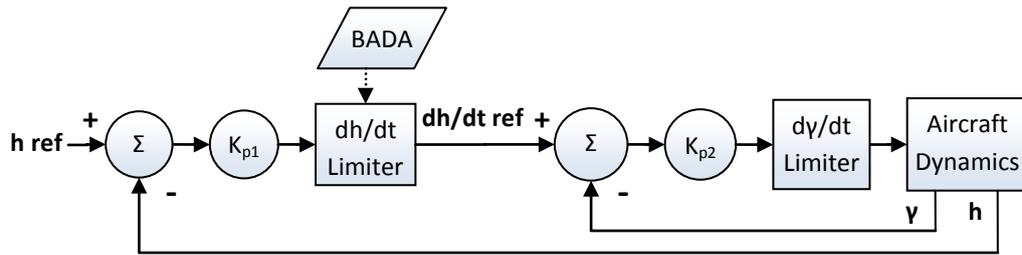


Figure 3-13: The vertical flight controller

First, a vertical speed command is set from the K_{p1} term and the altitude error. The vertical speed limits are interpolated from the BADA lookup tables for the phase of flight, altitude, and mass.

The commanded flight path angle can then be derived from the reference vertical speed as

$$\gamma = \tan\left(\frac{\dot{h}}{V_{gnd}}\right) \quad (\text{Eq. 3-17})$$

However a small angle approximation of

$$\gamma = \frac{\dot{h}}{V_{gnd}} \quad (\text{Eq. 3-18})$$

can be made with less than 1% error for flight path angles between ± 10 degrees and groundspeeds between 100 and 500 kts.

The K_{p2} term is used to control the flight path angle rate. In order to limit the aircraft's normal acceleration to 5 fps^2 , as per the guidance of the BADA user manual, the flight path angle rate is limited to

$$\dot{\gamma}_{\text{limit}} = \frac{5\Delta t}{V_{tas}} \quad (\text{Eq. 3-19})$$

(Nuic, 2010). The flight path angle rate command is integrated to derive the new flight path angle. The vertical flight controller was manually tuned to $K_{p1} = 0.15$ and $K_{p2} = 0.55$, which showed satisfactory response, damping and stability for airspeeds between 50 and 500 kts with a time step of 1 second.

3.6 Equations of Motion

Once the bank angle, flight path angle, and true airspeed are calculated from the flight control module described above, the following equations of motion are used to derive the remaining aircraft state and velocity vector terms.

Rearranging Equation 3-15, the turn rate, $\dot{\psi}$, can be computed from the bank angle and integrated to derive the new aircraft heading, ψ .

$$\dot{\psi} = \frac{g \cdot \tan(\phi_{bank})}{V_{TAS} \cdot \cos(\gamma)} \quad (\text{Eq. 3-20})$$

The aircraft inertial velocity is then converted to the North, East, Down vector components

$$V_N^i = V_{TAS} \cos(\gamma) \sin(\psi) \quad (\text{Eq. 3-21})$$

$$V_E^i = V_{TAS} \cos(\gamma) \cos(\psi) \quad (\text{Eq. 3-22})$$

$$V_D^i = V_{TAS} \sin(\gamma) \quad (\text{Eq. 3-23})$$

The Earth-Centred, Earth-Fixed velocity vector are found by adding the wind vector, V^{wnd} , from the Wind Field module

$$V_N = V_N^i + V_N^{wnd} \quad (\text{Eq. 3-24})$$

$$V_E = V_E^i + V_E^{wnd} \quad (\text{Eq. 3-25})$$

$$V_D = V_D^i + V_D^{wnd} \quad (\text{Eq. 3-26})$$

The latitude and longitude rates can be found from the North and East vector components by

$$\dot{\phi}_{Latitude} = \frac{V_N}{R_0 + h} \quad (\text{Eq. 3-27})$$

$$\dot{\lambda} = \frac{V_E \sec(\phi_{Latitude})}{R_0 + h} \quad (\text{Eq. 3-28})$$

The new position and altitude is obtained using Euler forward integration, where (Kayton & Fried, 1997)

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \dot{\mathbf{x}}_t \times \Delta t \quad (\text{Eq. 3-29})$$

The final step is to interpolate the fuel flow (in units of kg/s) from the BADA database for the respective aircraft type, which is integrated to obtain the new aircraft mass.

3.7 Navigation Error and Uncertainty Modelling

In addition to the basic performance, navigation, and trajectory calculations described above, an important functional requirement was the ability to model navigation errors and uncertainty in order to simulate the navigational noise of actual air traffic. This requirement has been addressed primarily through the Flight Technical Error model, with provision for modelling additional error through own-ship state estimation noise.

3.7.1 Flight Technical Error

The inability of actual flight control systems to steer aircraft perfectly along the desired course is known as Flight Technical Error (FTE). Equipment design and ambient environment variables, such as control dynamics or air turbulence, both influence FTE (ICAO, 1999). However, flight trials indicate that the predominate factor is the control mode: in one study the en-route FTE for manually-piloted flights using the flight director was 0.7 nmi (1296.4 m), while for autopilot coupled flights the error was reduced to 0.13 nmi (240.8 m) (Hunter, 1996; cited from Peters & Konyak, 2003). In both cases the period varied between 4 and 8 minutes.

FTE is significantly auto-correlated due to the feedback control loop of most flight control systems (Levy, et al., 2003). Given that, a reasonable stochastic model is a second order Gauss-Markov process:

$$\begin{pmatrix} \delta \dot{r}_{FTE} \\ \delta \dot{v}_{FTE} \end{pmatrix} = \begin{bmatrix} \mathbf{0}, & \mathbf{1}, \\ -\omega_0^2, & -2\beta\omega_0 \end{bmatrix} \begin{pmatrix} \delta r_{FTE} \\ \delta v_{FTE} \end{pmatrix} + c \begin{pmatrix} \mathbf{0} \\ \mathbf{u}_{FTE} \end{pmatrix} \quad (\text{Eq. 3-30})$$

where δr_{FTE} is the lateral position error, δv_{FTE} is the lateral position error velocity, c is the scale factor of the forcing function, ω_0 is the natural frequency of the system, β is

the damping of the system, and u_{FTE} is zero mean unity variance Gaussian white noise. The simulator implements the discretized form of this process with the parameters of Table 3-1 to generate lateral position wander at each time step, eliminating the need to separately model FTE factors such as air turbulence and pilot control imprecision (Peters & Konyak, 2003).

	B	ω_0 (mHz)	FTE, 1σ (m)
Manual	0.50	2.78	1296.4
Autopilot	0.50	2.78	240.8

Table 3-1: Flight technical error parameters

The FTE function is called before the FMS computations. The resulting δr_{FTE} is then passed to the FMS and added to the lateral offset from the flight plan data or ATC instruction. This causes the FMS to produce guidance commands that direct the aircraft to ‘follow’ the δr_{FTE} in relation to the underlying flight plan track.

If no flight technical error is applied to a given aircraft, then the control mode in the Master Array can be set to *Ideal*. Otherwise, the control mode can be set to either *Autopilot* or *Flight Director* to specify the FTE category to be incorporated with a particular aircraft. These modes enable the simulator to model traffic with varying but appropriate levels of navigational accuracy.

3.7.2 State Estimation Noise

In addition to FTE, another source of navigational error is own-ship state estimation noise. Own-ship estimation noise is any error between the estimated aircraft state used by the navigation routines, and the true aircraft state. Provision has been made for this type of navigational error by including a data field for the estimated aircraft states in the FMS data structure, separate from the true state data. The estimated (noisy) state data is used in the guidance and control functions, while the true data is used to update the actual aircraft state from the flight control inputs.

Because the appropriate noise model is highly dependent on the application and assumptions about sensors, and due to project time restrictions, a specific state estimation noise model was not implemented. Rather, a placeholder function has been included to add user-defined noise to the true aircraft state data when it is copied to the FMS data structure. The function can be further developed as necessary.

3.8 Verification and Evaluation of the Module

A series of simulation tests were conducted in order to ensure correct implementation of the described models, and to evaluate the performance of the trajectory modelling process in terms of navigation accuracy, functionality, and computational complexity.

The first step of the verification process was to confirm the BADA database was correctly applied in the simulator, in order to ensure that accurate aircraft performance data was being supplied to the navigation and trajectory functions. Next the flight controller was tested to ensure the lateral, vertical, and longitudinal controllers functioned as designed. The FMS functionality was then tested to establish the navigational accuracy of the system and to verify correct flight plan and ATC instruction following. Once the FMS, flight controllers, and performance models were verified, the flight technical error model was tested to demonstrate the ability to simulate navigational uncertainty. Finally, the execution speed of the module was tested in order to assess the impact of the computational complexity on the performance of the simulator.

3.8.1 Verification of the BADA v3.8 Database Implementation

Verification of the aircraft performance data was accomplished by comparing the durations of climb, cruise, and descent manoeuvres in the simulator to the expected duration, as derived directly from the aircraft's BADA Performance Table File. The expected flight time for a level cruise of a given distance can be found from the true airspeed values in the BADA database by

$$Duration = \frac{Distance}{True\ Airspeed} \quad (Eq. 3-31)$$

Similarly, the expected duration of a climb or descent over a given altitude can be found from the average of the Rate of Climb/Descent (ROCD) speeds between the lower and upper altitudes as

$$Duration = \frac{Altitude\ Difference}{ROCD\ Average} \quad (Eq. 3-32)$$

A Boeing 747-200 and Airbus A320 were tested in the simulator and the flight times of the manoeuvres were recorded. Table 3-2 compares the simulated and BADA-derived cruise, climb, and descent flight times to the nearest second.

	CRUISE			CLIMB			DESCENT		
	1000 nmi at FL330			From FL290 to FL330			From FL330 to FL290		
	Sim.	BADA	Error	Sim.	BADA	Error	Sim.	BADA	Error
B747-200	7211	7200	0.15%	135	136	0.71%	85	85	0.00%
A320	7942	7930	0.16%	229	231	0.91%	71	71	0.00%

Table 3-2: Manoeuvre durations (seconds)

The results show close correspondence between the simulated and BADA-derived flight times, indicating that the database was correctly implemented. The small percentage error can be attributed to numerical and navigational error, and is well within the navigational tolerances of modern commercial aircraft (Kayton & Fried, 1997).

3.8.2 Evaluation of the Flight Control System

The next step in the verification process was to evaluate the response of the lateral, vertical, and longitudinal controllers to step inputs. For this test, a Boeing 777-200 was used in the terminal phase of flight with the *Ideal* control mode.

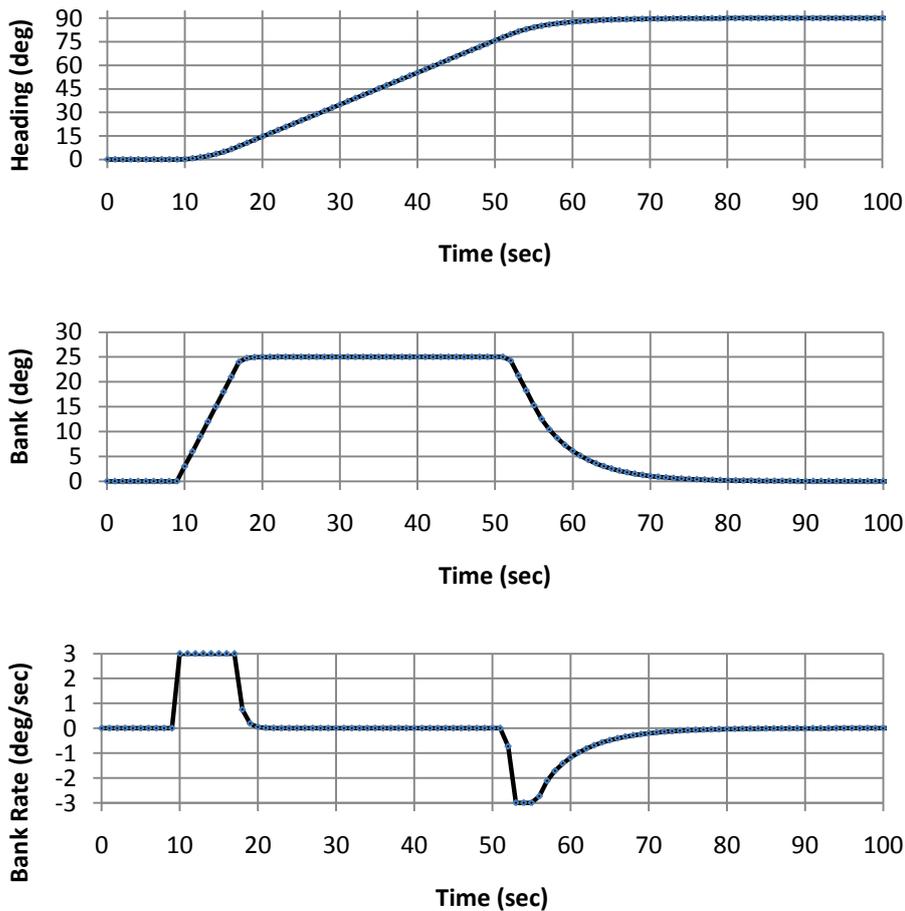


Figure 3-14: Lateral controller response to 90 degree heading change

In the first test, a 90 degree heading change command was sent to the lateral flight controller. The instantaneous heading, bank angle, and bank rate were recorded, and are plotted in Figure 3-14. The plots show that the controller applied bank incrementally until the 25 degree limit was reached. The bank was then proportionally reduced to zero as the aircraft completed the turn. The bank rate was limited to ± 3 deg/sec as desired with the autopilot control mode.

In the second test, the vertical flight controller responded to a 2000 ft altitude change, from FL350 to FL370. As before, the altitude, vertical speed, and flight path angle were recorded and are shown in Figure 3-15. The flight path angle was increased to 3.1 degrees, until the ROCD reference value from the BADA database was met. The linear reduction of the flight path angle and vertical speed is due to the change in

aircraft performance with altitude. Finally, flight path angle was proportionally reduced to zero as the aircraft completed the climb.

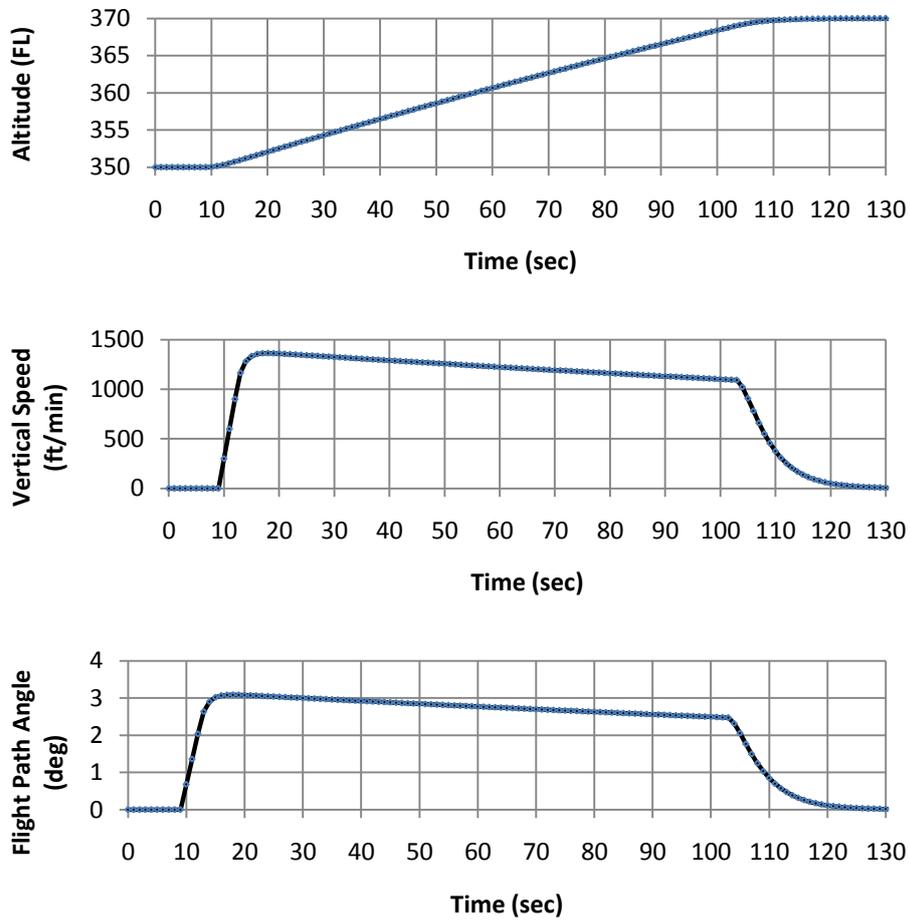


Figure 3-15: Vertical controller response to a 2000 ft climb

The longitudinal controller was tested by commanding the aircraft to accelerate from 250 kts to 280 kts. Figure 3-16 shows the acceleration was correctly limited to 2 fps^2 until within 8 knots of the commanded airspeed, at which point the acceleration was reduced to 0.69 fps^2 . The acceleration was proportionally reduced to zero as the aircraft approached the target speed.

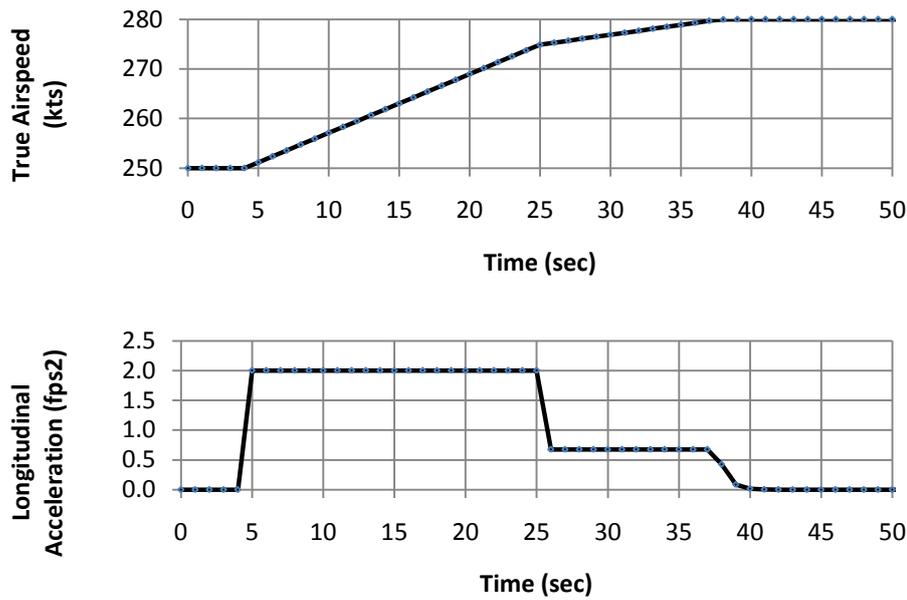


Figure 3-16: Longitudinal controller response to a 30 knot speed change

All three controllers demonstrated the desired response, with the aircraft completing the manoeuvres within an interval that is reasonable for commercial transport aircraft (Pratt, 2000).

The tests described above were repeated in a test matrix that covered a variety of airspeeds (50 to 500 kts), airspeed changes (5 to 50 kts), altitudes (10000 to 40000 ft), altitude changes (500 to 5000 ft), bank angle limits (15 and 25 degrees), and heading changes (15 to 180 degrees). Due to space constraints, only one set of test results could be presented (shown above). All the flight controller tests produced similarly satisfactory results.

3.8.3 Evaluation of the Flight Management System

The FMS was tested for inherent navigational accuracy (that is, without FTE) and correct path/terminator implementation.

3.8.3.1 Evaluation of Navigation Accuracy

The purpose of this test was to evaluate the long distance navigational capability of the FMS and to verify correct Great Circle navigation. A Boeing 777-200 was flown at FL330 on 1000 nmi track-to-fix routes in the cardinal and intercardinal directions. In order to isolate inherent navigational accuracy from FTE, the flight control mode was set to *Ideal*. With all eight routes, the initial point was Heathrow Airport, since it was expected that any Great Circle navigation error would be accentuated at Northerly latitudes. Two tests were flown for each route – one with wind fields from the NPN wind module, and one without any winds. In all sixteen flights, the position of the aircraft was recorded every 111 nmi, as indicated in Figure 3-17. The MATLAB Mapping Toolbox was then used to independently calculate the intermediate waypoints with the same spacing along the Great Circle course between the start and destination points. The distance between the simulation points were compared to the MATLAB waypoints and have been plotted in Figure 3-18.

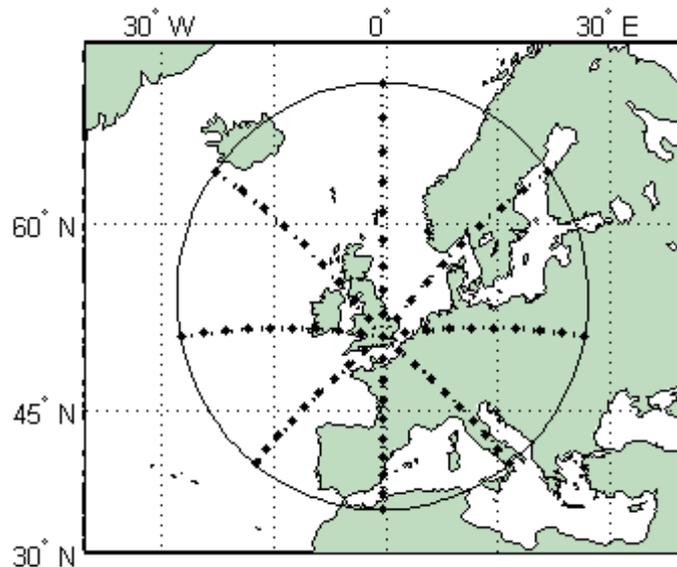


Figure 3-17: Sample points in navigation test

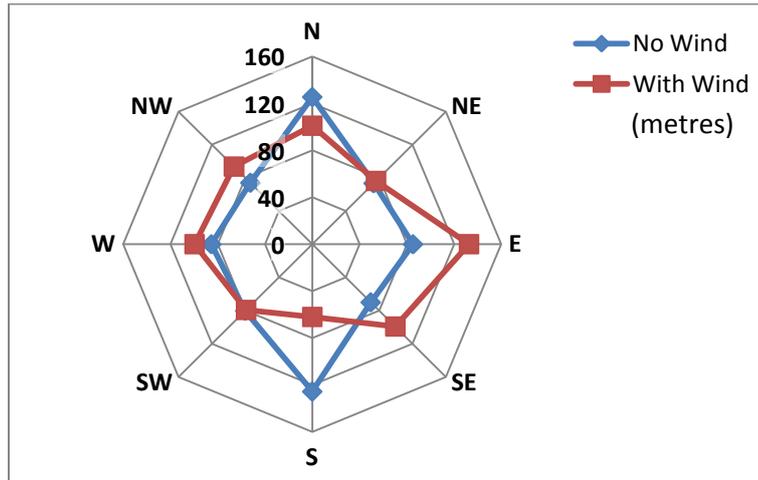


Figure 3-18: Radar plot of navigational accuracy

The results show that the FMS produces very accurate flight guidance for long-distance Great Circle courses. The average error without wind was 90 m, and with wind was 93 m. Both cases are well below the 1852 m (1 nmi) threshold requirements for RNAV-1 and are a 90% improvement from the 926 m (0.5 nmi) navigational accuracy demonstrated by the original Airspace Simulator. The small differences between the simulator and MATLAB can be attributed to numerical errors and the limitations of proportional controllers with small feedback errors. The results also demonstrate that the FMS control is effective in the presence of winds. The asymmetrical error in the wind case is because the aircraft encountered different wind fields for each route due to the geospatially-referenced wind model.

3.8.3.2 Verification of Flight Plan and ATC Instruction Following

Next, a series of test flights for a Boeing 777-200 at FL330 were completed to verify the capability of the simulator to implement conflict resolutions by route modifications and ATC instructions. In all test flights, the original flight plan was the route shown in Figure 3-19 from the initial fix to *Wpt2*.

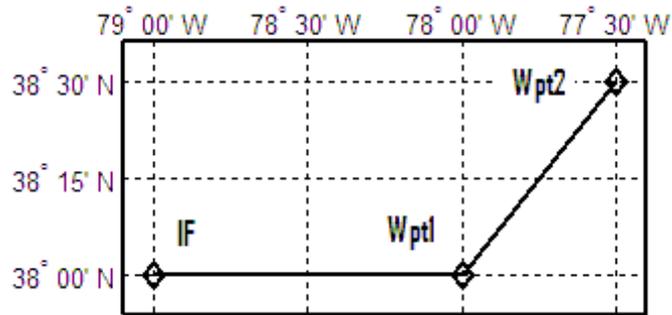


Figure 3-19: Test route

This route was flown with both fly-by and fly-over transitions at *Wpt1* to verify correct segment transitions. Figure 3-20 plots the fly-by tracks in solid red and the fly-over in the blue dash-dot line. As can be seen, the fly-by case anticipates the turn, and the fly-over case flies past the waypoint and then turns to intercept the course.

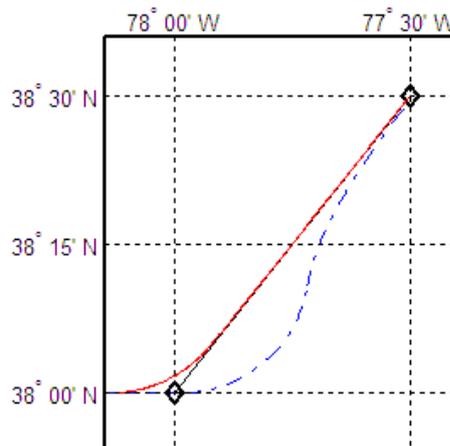


Figure 3-20: Segment transition test

In order to demonstrate the ability of the FMS to automatically incorporate route modifications, a sequence of two waypoints was passed to the FMS after 60 seconds of simulation, representing a turning point manoeuvre with a 45 degree delay segment of 15 nmi. Figure 3-21 shows the turn-away and turn-back points (red diamonds) were successfully inserted, changing the resulting trajectory (solid red line).

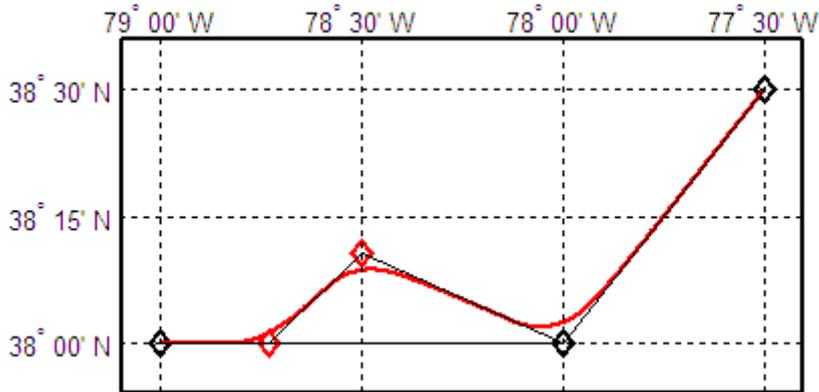


Figure 3-21: Route modification test

The purpose of the next test was to verify heading instructions, track angle instructions, and direct-to-fix segments. A 30 degree heading instruction was give to the aircraft after 60 seconds of simulation. After an additional 180 seconds, a DF route modification was given to *Wpt2*. The flight was repeated with a 30 degree track angle instruction. In both cases, a steady wind-field was set to 100 knots from the west to make the difference between heading and track angle instructions more apparent. The results are plotted in Figure 3-22 for the heading case in solid red and track angle in blue dash-dot. The aircraft location when the DF instruction was given is shown by the red and blue diamonds; the difference in locations is due to the wind displacement on the heading segment.

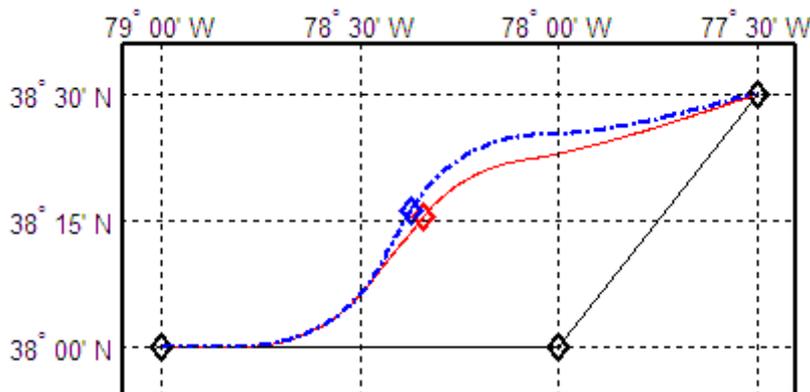


Figure 3-22: Heading, track angle, and direct-to test

Lateral offsets were tested by commanding a 2 nmi offset to the right and left of the course after 60 seconds of simulation. The instruction was given a duration of 240 seconds. As can be seen from Figure 3-23, the turn point was correctly translated for

both right of course (solid red) and left of course (blue dash-dot), to maintain the correct offset distance following the turn.

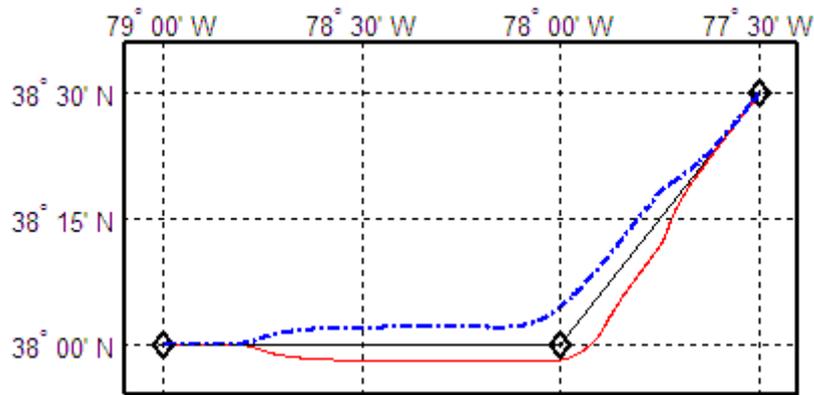


Figure 3-23: Lateral offset test

Finally, two test flights were conducted to verify the ability to give timed altitude and speed instructions. In the first, a 2000 ft climb instruction was passed to the FMS after 60 seconds of simulation, with an instruction duration of 6.5 minutes. Similarly, in the second flight the aircraft was instructed to accelerate by 20 knots. The resulting altitude and speed profiles are shown in Figure 3-24A and 3-24B, respectively. It can be seen that the aircraft correctly follows the instructions, and then returns to the flight plan altitude and BADA reference speed when the instructions timed-out.

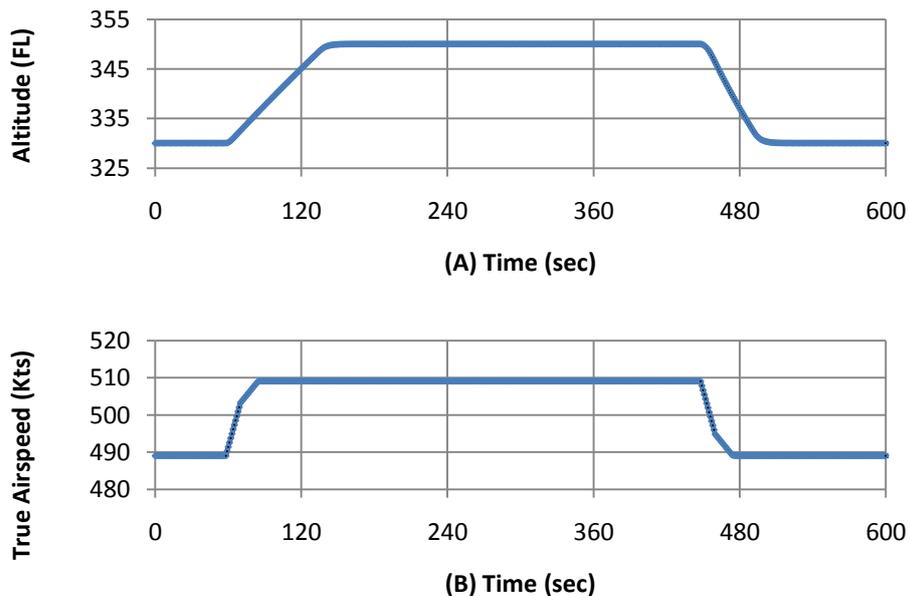


Figure 3-24: Test of timed altitude (A) and speed (B) instructions

These test flights have demonstrated the ability of the simulator to construct the required conflict resolution manoeuvres described in Section 3.1 using route modifications and timed- or manually-terminated instructions, as well as the ability of the FMS to correctly execute the manoeuvres.

3.8.4 Verification of the Flight Technical Error Model

In order to test the FTE model, two simulations of 30000 seconds each were completed with a B737-700 at FL290 and 431 kts. In the first simulation the aircraft was controlled by the autopilot, and in the second the aircraft was controlled by the flight director. The instantaneous flight technical error produced by the FTE module was recorded at each time step along with the actual cross track error, in order to determine if the cross track error (XTE) corresponded to the FTE as would be expected.

Portions of recorded data from both simulations are show in Figures 3-25A and 3-25B. As can be seen, the cross track error appears to closely follow the FTE, indicating that the FTE model causes the cross track error.

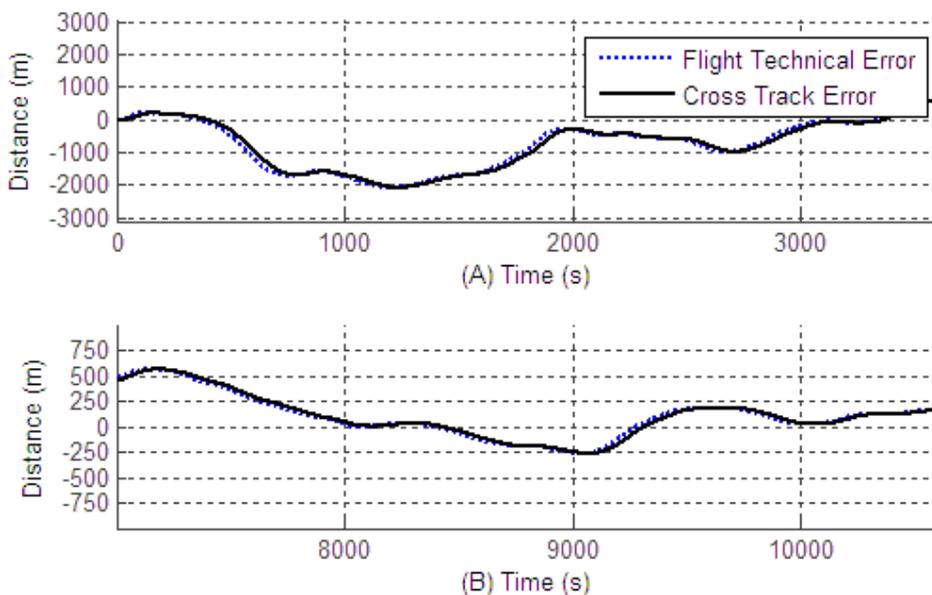


Figure 3-25: Time series of flight technical error

The mean and standard deviation of the sampled FTE position, δr_{FTE} , and FTE velocity, δv_{FTE} , were computed, and are compared to the desired values in Table 3-3.

The small errors indicate FTE model parameters have been correctly applied in the simulator.

		δr_{FTE} (m)			δv_{FTE} (m/s)		
		Desired	Sample	Error	Desired	Sample	Error
Flight Director	Mean	0	-136	136	0	-0.0079	0.0079
	Std. Dev.	1296	1210	86	3.6011	3.4971	0.1040
Autopilot	Mean	0	-27	27	0	-0.0068	0.0068
	Std. Dev.	241	225	16	0.6688	0.6929	0.0241

Table 3-3: Comparison of flight technical error parameters

3.8.5 Evaluation of Module Execution Speed

The new Navigation and Trajectory Module is a significant improvement to the original simulator in terms of functionality, flexibility and fidelity. However, this additional realism comes at the cost of additional computational complexity. In order to determine the speed performance of the module, a representative flight was simulated from London to New York, and replicated 29 times. The simulator execution time was categorised and recorded.

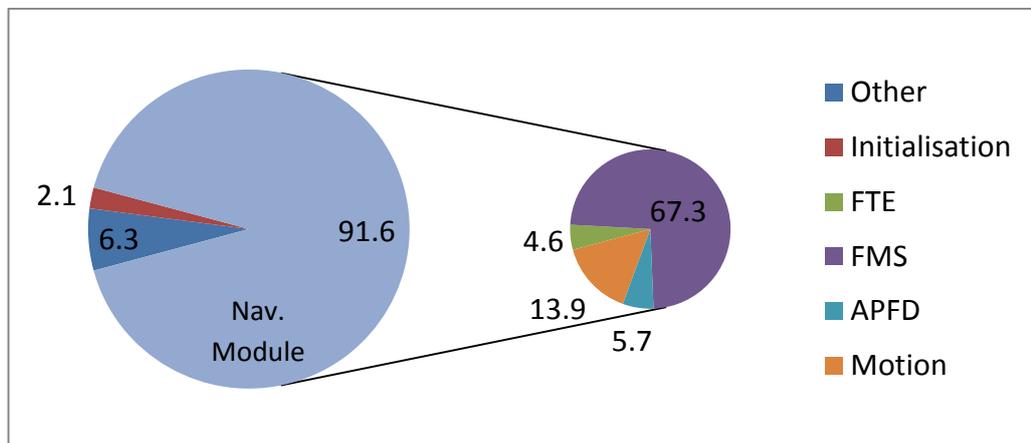


Figure 3-26: Breakdown of simulator execution time

The larger graph in Figure 3-26 shows the relative percentages of the execution time of the initialisation process, the Navigation and Trajectory Module, and all other simulator functions. The graph shows that the Navigation and Trajectory Module represents over 91% of the entire computation time, while initialization accounts for

only 2.1% and all other simulator functions represent only 6.3%. These percentages are not surprising, since updating the navigation and trajectory must be completed every time step for every airplane, while most other simulator functions, such as the communications modelling or writing the data block, apply less frequently.

The smaller graph shows the breakdown of execution time within the module; the FMS model accounts for the majority (67.3%) of the module's execution time. FTE represented only 4.6% of the module's execution time, which indicates that the additional realism provided by the FTE model has a relatively small impact on the simulator's overall speed performance.

Despite the *relative* costliness of the Navigation and Trajectory Module compared to the other modules, the actual speed of the simulator remains very fast. In this case, more than 186 flight hours were computed in less than 3 seconds, which shows that the software was efficiently designed and implemented. Thus, it was decided that the functionality, flexibility and fidelity provided by the new module was worth the computational complexity.

3.9 Summary

The requirements of the Navigation and Trajectory Module have been discussed in terms of navigation accuracy, the ability to model navigation errors, and the ability to implement conflict resolution manoeuvres. The trajectory modelling process that was designed to meet these requirements was described, focusing on the performance model, FMS, APFDS, equations of motion, and the flight technical error.

The evaluation of the module has shown that the BADA database has been correctly applied in the simulator. The Autopilot/Flight Director model was shown to produce the desired response to heading, altitude and speed commands generated by the FMS module. The FMS showed precise Great Circle waypoint-to-waypoint navigation, and demonstrated the ability to accept and execute flight plan modifications and ATC instructions in-flight. Finally, the flight technical error model

was shown to produce random lateral wander consistent with the desired model parameters.

The next chapter will describe the Communications Module that enables connected CD&R tools to control the traffic using the route modifications and ATC instructions described in this chapter.

Chapter 4

Communications Module

The Communications Module has two purposes: to handle the exchange of messages between the simulator and any connected external systems such as CD&R tools, and to model those exchanges as either ADS-B, datalink, radio-telephony, or ASAS messages. The module enables external systems to monitor the traffic through ADS-B-like surveillance, and to control the traffic through instructions and route modifications. Section 4.1 identifies the requirements that are specific to the Communications Module. Section 4.2 then describes the communications process designed to meet these requirements. Sections 4.3 and 4.4 discuss the latency and surveillance sub-modules in detail, and finally, an evaluation of the module is presented in Section 4.5.

4.1 Requirements for Modelling Communications

A design goal for the project was the ability to integrate new CD&R tools with the simulator. By implication, the Communications Module must be able to output traffic surveillance information to conflict detection and traffic monitoring systems, and receive conflict resolutions as input from conflict resolvers. The module should simulate ADS-B-type surveillance – expected to be the baseline surveillance method of future ATM systems – by broadcasting the aircraft state, velocity vector, and intent data to any connected external systems. The input message set must include the conflict resolution types outlined in Section 3.1.

Additionally, to enable the testing of separation management systems and concepts under non-ideal conditions, the module is required to simulate the errors and uncertainties associated with datalink and radio-telephony communications, and ADS-B surveillance.

4.2 The Communications Process

The communications process used to pass conflict resolution messages to aircraft (input messages) is shown in Figure 4-1. Messages can be generated from two sources:

- internal messages from other modules within the simulator
- external messages from modules and systems connected to the simulator

It is important that the method of integrating the simulator with external systems be flexible and minimise any modifications required of the external systems. To accomplish this, customised *interface functions* are used to map input from the external systems (i.e. conflict resolutions) to data structures that are compatible with the simulator, as well as to manage the relevant networking protocol and procedures. The interface functions must be adapted according to the specific system that is attached.

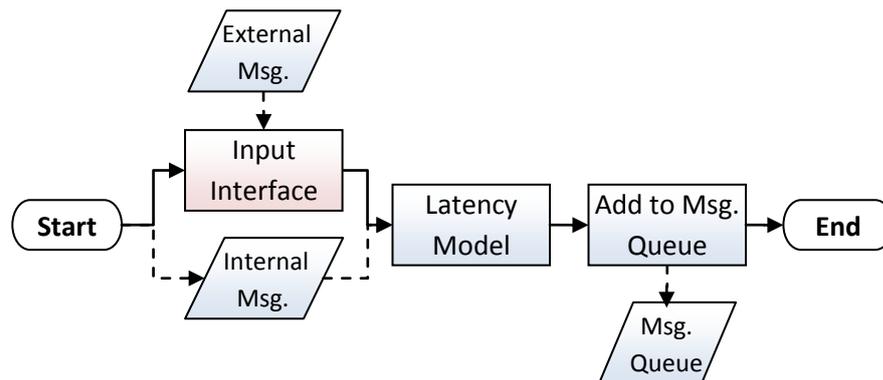


Figure 4-1: The communications input process

The message is then sent to a latency model to simulate the stochastic delays and timing-uncertainties of either datalink or radio-telephony communications, or the delays in flight crew response to ASAS-generated resolutions. Once given a delivery time from the latency model, the message is stored in the receiving aircraft's message queue until the delivery time has been met.

The communications output process (communication from the simulator to any connected modules) is shown in Figure 4-2. A surveillance model is used to simulate noise and failure of the surveillance data before it is broadcast. As with the input

process, customised output interface functions are used to connect the simulator to external tools. For example, because *Tviz* uses the messaging system developed for the *Smart Skies Project*, an output interface function was developed to convert the simulation surveillance data to a Smart Skies-compatible message and send it over a UDP network (Baumeister, et al., 2009). As a result, no modifications were required of the *Tviz* software to network it to the simulator.

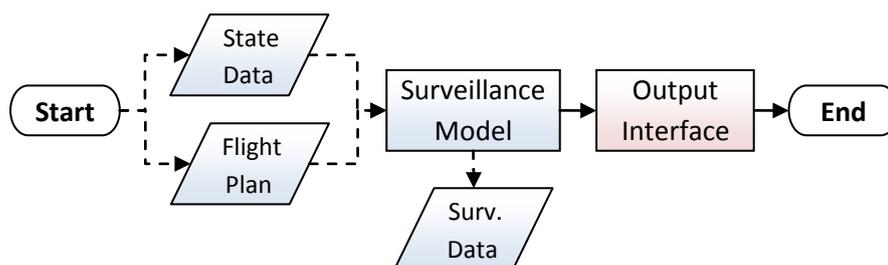


Figure 4-2: The communications output process

4.2.1 The Message Set

The input message set includes *route modifications* and *instructions*. Route modifications consist of a series of new flight plan segments of track-to-fix or direct-to-fix waypoints, with fly-by or fly-over transitions, as well as the segment altitude, speeds and lateral offset. Instructions consist of a specified heading, track angle, lateral offset, altitude, or true airspeed for the aircraft to maintain. A flag is set to indicate if the instruction is to be manually terminated (upon receipt of another message) or automatically terminated after a given time. If time-termination is specified, then the message also stores the specified the instruction duration.

When using the Datalink and ASAS communication modes, any of these message types may be used. However, in order to simulate the message complexity limitations of voice communications, radio-telephony messages are restricted to instructions or a single direct-to-fix segment.

The output message set consists of the periodically-broadcasted surveillance data elements listed in Table 4-1. In addition, to ensure the conflict detection systems have accurate trajectory intent information, the flight plan is transmitting when an aircraft

is first generated and when any route modifications are made within the aircraft's FMS.

System Information:	State:	Air-Mass Velocity:	Ground Velocity:
<ul style="list-style-type: none"> • Aircraft identification • Active waypoint 	<ul style="list-style-type: none"> • Latitude • Longitude • Altitude • Bank angle 	<ul style="list-style-type: none"> • True airspeed • Heading 	<ul style="list-style-type: none"> • Ground speed • Track angle • Vertical speed

Table 4-1: The surveillance output data set

Although ADS-B is expected to be the primary surveillance source in future ATM systems and is assumed here, the surveillance data can be limited the aircraft position and altitude, in order to simulate the reduced information content of current Primary and Secondary Surveillance Radar (PSR and SSR) systems.

4.2.2 The Message Queue

The message queue is a linked list that stores messages until the latency period is completed; that is, the message is held by the message queue until the current simulation time is greater than the message's delivery-time. Using a linked list allows multiple messages to be stored for each aircraft and also reduces the memory requirement in comparison to allocating a fixed-size message array. The data structure for the message queue nodes and the type definition of the ATC Instructions are shown in Figure 4-3. Simple interface functions can be added to the communication module to translate messages to and from this structure for CD&R systems that use different data structures.

```

struct ac_types_MSG_QUEUE_NODE
{
    comm_MSGTYPE      MsgType;    // 1 = Instruction, 0 = Route Mod
    float             DeliverTime; // Time at msg receipt + latency
    ac_types_INSTRUCTION Instruction;
    struct ac_types_MSG_QUEUE_NODE *PrevMsg; // pointer to prev msg
    struct ac_types_MSG_QUEUE_NODE *NextMsg; // pointer to next msg
    struct nav_PATH_NODE *RouteMod; // pointer to route modification
};

typedef struct
{
    char TimeManFlag; // 1 = timer terminated, 0 = manually terminated
    char HdgTrkOffFlag; // 0 = no lat cmd, 1 = trk ang, 2 = hdg, 3 = lat offset
    boolean FPActive; // False = suspend flight plan, true = resume
    float LatCmd; // Lateral command (rads or nmi)
    float AltCmd; // Altitude command; 0 = no alt command (m)
    float SpdCmd; // Speed command; 0 = no speed command (m/s)
    float Timer; // Timeout value, from receipt of command (s)
} ac_types_INSTRUCTION;

```

**Figure 4-3: Data structure message queue nodes
and type definition of ATC Instructions**

The message queue for each aircraft is checked at every time step. If the queue contains any messages (e.g. not NULL), the delivery-time is checked. Once the delivery-time is met, the message is passed to the FMS and the message node of the queue is freed from memory. If the message is an instruction, the FMS suspends the flight plan and produces flight guidance according to the instruction. The flight plan is reactivated when the message duration timer expires or upon receiving either a direct-to-fix flight plan segment or another instruction with the *FPActive* field equal to 'true'. However, if the message is a route modification, the new flight plan segments must be merged into the active flight plan using the following method.

4.2.3 Incorporating Flight Plan Modifications

An algorithm was needed to automatically merge route modifications into the flight plan. The problem is simple if the final waypoint of the route modification corresponds to an original waypoint. In that case, the route modification waypoints

simply replace any intermediate waypoints in the original flight plan between the aircraft and the final route modification waypoint.

However, requiring the final waypoint of the route modification to correspond with an original waypoint forms a constraint on the generation of conflict resolution solutions, and not all conflict resolvers are designed to incorporate this principle. In current operations, route discontinuities are manually corrected by the flight crew (Palmer, et al., 2000), but an automated solution was needed for the simulator. Simplistic solutions, such as merging at the nearest segment/waypoint or inserting the sequence of new segments before active waypoint, can lead to unexpected and unrealistic behaviour. An algorithm was needed to merge route modifications in an intelligent, predictable way; however there is little published literature and no generally accepted solution.

For example, consider the case of an automated tactical separation management system that has generated a flight plan modification consisting of three waypoints (squares in Figure 4-4) to correct a previous resolution that failed to resolve the conflict. These new segments must be merged with the active flight plan (diamonds).

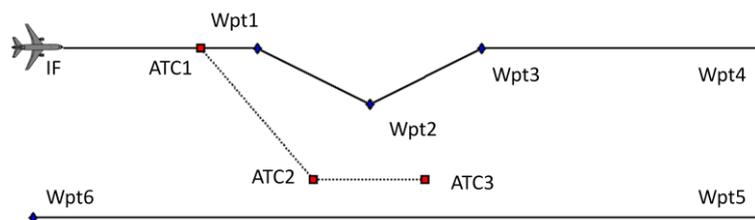


Figure 4-4: Scenario requiring merging of resolution waypoints ATC 1...3 into a flight plan defined by waypoints WPT 1...6

Figure 4-5 traces the ground tracks of two incorrect outcomes produced by the simulator, illustrating the challenge of automatically modifying flight plans. The first case (dashed grey track) demonstrates the need for an intelligent method of merging flight plans. The flight plan segment was inserted directly into the active flight plan before the original active waypoint, resulting in a flight path that backtracks to the original active waypoint following the completion of the resolution manoeuvre. This

behaviour is undesirable and could be prevented using a more intelligent merging routine. The second flight (solid blue track) illustrates the limitations of overly-simplistic merging routines. In this case, the final new waypoint was merged with the closest leg in the original flight plan. This again results in an undesirable flight path, by bypassing a significant portion of the original flight plan.

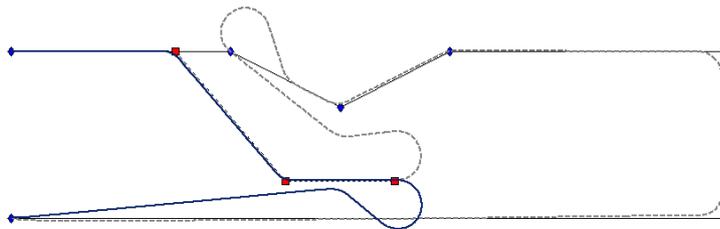
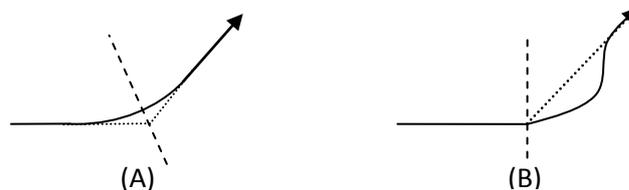


Figure 4-5: Resulting ground tracks of two incorrect merges

To avoid these errors, an algorithm was developed to merge the lateral components of two flight plans based on the principle of wayline leg sequencing. A flight plan segment with a fly-by transition can be sequenced when the aircraft crosses an infinite-length wayline at the bisector of current segment and next segment (Sptizer, 2001), illustrated in Figure 4-6A. For fly-over transitions the wayline is perpendicular to the final segment course, illustrated in Figure 4-6B.



**Figure 4-6: (A) Bisecting wayline (dashed line) for fly-by transitions;
(B) Perpendicular wayline for fly-over transitions**

The algorithm, shown in Figure 4-7, operates by treating each new waypoint in the route modification list as a *virtual aircraft*. The virtual aircraft can then be tested in the original flight plan against the terminating wayline of the active segment. If the virtual aircraft has crossed the wayline, the next segment in the original flight plan is flagged and evaluated. The wayline test is repeated until the virtual aircraft fails to cross the wayline of the flagged leg. Then, the next new waypoint in the route modification list is considered the virtual aircraft, and evaluated against the flagged leg.

After all new waypoints have been treated as the virtual aircraft, the flagged waypoint indicates the merge point between the original flight plan and the route modification. The original waypoints up to the flagged waypoint are removed and replaced with the new waypoints, and the first of the new waypoints is then set as the active segment, completing the merge.

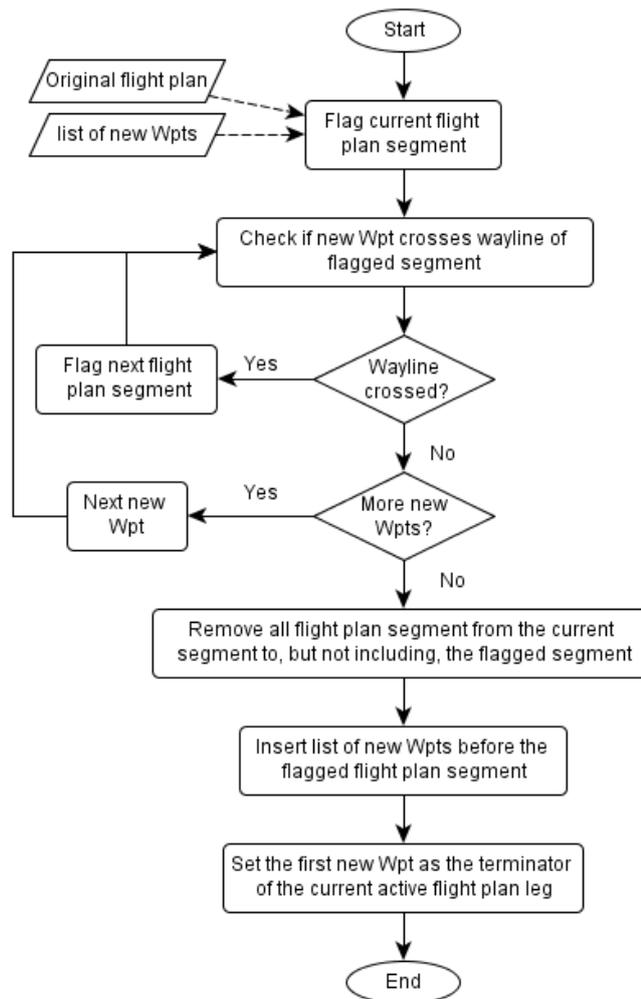


Figure 4-7: An automated algorithm for merging route modifications into a flight plan

For example, using the scenario described previously, the algorithm processed as follows:

- Point *ATC1* was the virtual aircraft. Point *ATC1* did not cross the *WPT1* wayline, so the active waypoint remained *WPT1*

- Point *ATC2* became the virtual aircraft. Point *ATC2* crossed the *WPT1* wayline but not *WPT2*, so the active waypoint was updated to *WPT2*
- Point *ATC3* became the virtual aircraft. Point *ATC3* crossed the *WPT2* wayline but not *WPT3*, so the active waypoint was updated to *WPT3*

Because *ATC3* was the final new waypoint, the merger was completed by replacing *WPT1* and *WPT2* with *ATC 1...3*. The resulting flight plan and ground track are shown in Figure 4-8, demonstrating a predictable and desirable merge between the two flight plans.

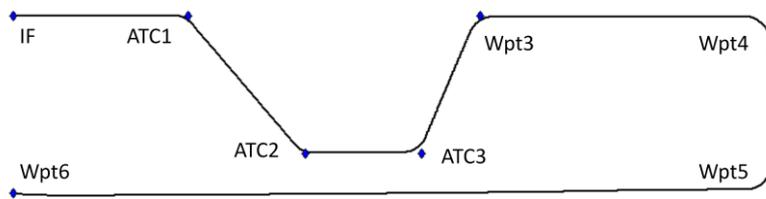


Figure 4-8: Resulting ground tracks after using the automatic merging algorithm

The purpose of the algorithm is to provide an automated merging capability that models the way a human pilot would incorporate a conflict resolution described as a series of waypoints. Clearly, to fully model human behaviour and decision making, a much more complex algorithm would be necessary to take into account difficult situations and flight goals, such as maximising economy or meeting time-of-arrival constraints. Additionally, there are currently no regulations or protocols governing the merging of conflict resolutions into flight plans; this is an issue that will need to be further addressed before these types of conflict resolutions can be used operationally. For example, the problem would be resolved if CD&R systems issued route modifications complete to the destination, but this may be an inefficient use of datalink bandwidth.

4.3 Latency Modelling

Communications latency is any delay between the generation of a message (i.e. a conflict resolution), and when the message is executed by the flight crew. Latency can introduce stochastic variation to aircraft trajectories when completing conflict

resolution manoeuvres, and as a result, should be taken into account when designing and evaluating CD&R systems and concepts. Sections 4.3.1 through 4.3.3 discuss the default latency values used in the simulator; however, the latency can be redefined as desired in the configuration file. If the aircraft's communications mode is set to *Ideal*, then latencies are set to zero.

4.3.1 Datalink Latencies

Uplink latencies (messages from ATM ground-systems to the aircraft) in datalink communications can be attributed to the link technical delay and pilot response delay. Link technical delay is defined as the time between when a message is sent by the air traffic controller to when it is ready to be displayed to the flight crew, including any data processing delays and network traversal times. Technical delay varies from system to system, but is typically less than 2 seconds (Grogan, 2007; Bolczak, et al., 2004; Delhaise & Esposito, 2007). In order to keep the latency model technology independent, and because little information could be found on the distribution of delays, the link technical delay was modelled as uniformly distributed between 0.5 and 3.5 seconds.

The second source of latency is the pilot response delay. It is assumed that even in more highly automated future ATM systems, pilots will remain responsible for the safety of the flight and will examine any trajectory modifications before executing the manoeuvre – as is currently the case (FAA, 2004). As a result, the flight crew will need to recognise, review, and respond to any trajectory modifications, introducing a significant source of latency. There have been a numerous studies on human response time to datalink messages (Mackintosh, et al., 1999; Lozito, et al., 2003; FAA, 1996; Gonda, et al., 2005; Knox & Scanlon, 1990). The wide range of latency values reported suggest that the pilot delay is highly dependent on factors such as operational procedures, the human-machine interface design, cockpit distractions, message complexity, and crew training.

Recognising these variations, the pilot delay model used in the simulator is based on the latency values published by Mackintosh, et al (1999). In this study, five flight

crews flew the NASA Advanced Concepts Flight Simulator with a B757/767 FMS, using datalink as the primary ATC communication medium. A mean pilot delay of 28.6 seconds with a large standard deviation of 38.8 seconds was reported between the message alert (visual and chime) and when the crew was finished handling all the message elements. These values are supported by statistics of initial datalink operations in both the United States and Europe, which also report mean pilot delays of approximately 30 seconds with large variance (EUROCONTROL, 2007; Gonda, et al., 2005).

Statistics of pilot and controller message durations and response times tend to be skewed with a peak at a short time interval and a long tail for both radio-telephony and datalink communications (Gonda, *et al.*, 2005; Hung, 2005; Knox & Scanlon, 1990; Graglia, 2002; FAA, 1996). The lognormal distribution was selected as a non-negative, left-skewed distribution to model these random variables. The lognormal distribution has been shown to fit empirical distributions of many communication parameters, including call durations both in mobile and fixed telephony systems, human reaction and response times, and speech segment durations (Ulrich & Miller, 1993; Ratcliff & Murdock, 1976; Hockley, 1984; Rosen, 2005; Guo, *et al.*, 2007).

For each datalink message, the link technical delay and pilot response delay are randomly drawn from uniform and lognormal distributions and summed with the current simulator time to form the message delivery time. Messages are stored in the message queue until the simulator time is greater than this delivery time.

4.3.2 Radio-Telephone Latencies

Aircraft unequipped with data links or with inoperative data links must rely on voice radios to communicate. Latencies in radio-telephone communications can be attributed to communication transaction times and frequency occupation.

Cardosi analysed 46 hours of voice recordings of en-route airspace communications in the United States in a study of communication transaction times (1993). Traffic avoidance instructions were found to have mean controller speech

duration of 4.85 seconds and the subsequent pilot reaction time was observed to have a mean of 3.31 seconds. Assuming the pilot-flying acknowledges the clearance to the pilot-not-flying before implementing the manoeuvre, the expected delay before an instruction is executed is 8.16 seconds. However, 16% of the messages required the controller to repeat or clarify the instruction at least once, primarily because the pilot failed to respond to the initial call. In the worst observed case, this led to a delay of 31 seconds. Including these instances, the total time required to correctly communicate traffic avoidance manoeuvres had a mean of 10.85 seconds with a standard deviation of 5.91 and a 99th percentile of 40 seconds.

A second source of uncertainty in radio-telephony communications is channel occupation. Communications over the standard amplitude-modulated VHF radio are half-duplex; if the controller is ready to transmit a conflict resolution instruction but another transmission (by either the pilot or controller) is underway, then the controller must wait.

Graglia (2002) presented the results of an extensive analysis by the Centre d'Études de la Navigation Aérienne (CENA) of 60 hours of recorded pilot-controller en-route communications from twelve French sectors. Controllers transmitted 45% of the 19000 transmissions with mean speech duration of 3.7 seconds and standard deviation of 2.0 seconds, while pilot messages had mean duration of 2.9 seconds and standard deviation of 1.5 seconds. The physical occupancy of the channel – that is, the cumulative duration of transmissions over a given period – averaged 30%. However, over short periods of time, the physical occupancy was occasionally higher, and peaked to 75% over an 8 minute period. Using the CENA information, and by conservatively assuming independence between message durations and the channel occupancy, a model of the transmission delay as a function of the channel occupation percentage can be estimated, as shown in the pseudo-code of Figure 4-9.

```

rv1, rv2, rv3 = random numbers on unit interval
IF rv1 < channel occupancy percentage THEN
  IF rv2 < controller transmissions percentage THEN
    rv4 = random controller transmission duration
    tx_delay = rv4*rv3
  ELSE
    rv4 = random pilot transmission duration
    tx_delay = rv4*rv3
  ENDIF
ELSE
  tx_delay = 0
ENDIF

```

**Figure 4-9: Pseudo-code model of transmission delay
given the channel occupation percentage**

Figure 4-10 shows the mean and standard deviation of delays due to channel occupation of simulations of 30000 samples each, for ten physical occupancy percentages. As can be seen, the range of transmission delays is relatively small. By default, the simulator conservatively uses the 60% channel occupation values, for a mean of 1.0 and standard deviation of 1.4 seconds.

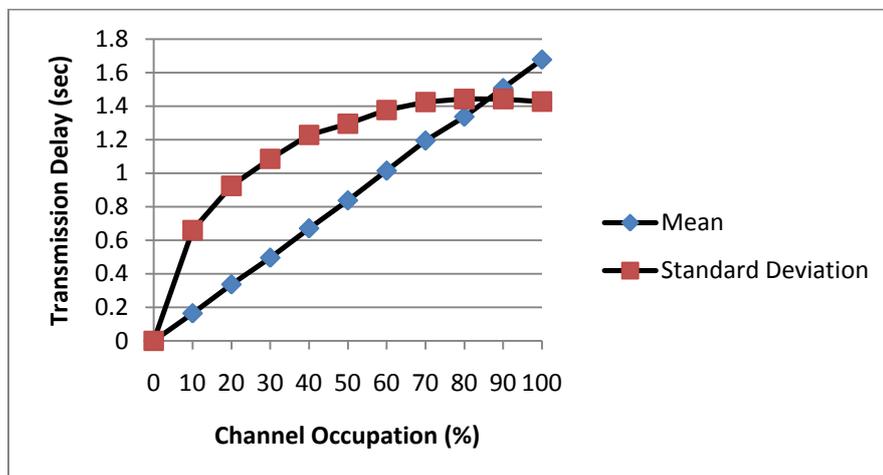


Figure 4-10: Transmission delays due to channel occupation

For each radio-telephony message, the communication transaction time and frequency occupation delay are randomly drawn from lognormal distributions and summed with the current simulator time to form the message delivery-time.

4.3.3 ASAS Resolutions

For conflict resolution generated by ASAS equipment, there is no delay caused by the datalink protocol and are no failed messages due to distance or interference, because the equipment is on-board the aircraft. However, there is still a delay between when a resolution is generated and when it is implemented due to pilot reaction and response times. It is assumed that delay in response to a conflict resolution generated by ASAS equipment will be similar to the reaction and response delay for data linked messages. Thus, the latency for an ASAS resolution was also modelled with a mean of 28.6 seconds and a standard deviation of 38.8 seconds. For each ASAS resolution, the latency is drawn from a lognormal distribution and summed with the current simulator time to form the message delivery-time.

The latency values discussed above can be redefined in the configuration file according to the specific experiment requirements.

4.4 Surveillance Broadcasting

The simulator periodically broadcasts the state and velocity of every active aircraft to provide traffic surveillance data to research and visualisation tools connected to the simulator. The broadcast rate can be specified in the configuration file, but is by default once per second. Thus, under ideal conditions, the surveillance update period for simulated ADS-B messages is 1 second.

However, with the 1090 MHz ADS-B communication structure, ADS-B messages can fail to be received by its recipient due to the Mode-S broadcast range and interference effects (FAA, 2002), resulting in stochastic variation of the surveillance update period. Blom, et al, have shown that collision risk increases linearly with decreasing ADS-B availability and reliability (2007). Thus, it was important to model imperfect ADS-B message reception.

In order to minimise extraneous variables in simulation experiments, the causal factors of message failure were not independently modelled; rather, the stochastic failure behaviour is modelled by applying constant, uniform probability of success for surveillance broadcasts. The Effective Update Period (EUP) performance requirements for ADS-B surveillance in the terminal and en-route airspaces with 95% confidence are shown in Table 4-2 (RTCA, 2006).

	TMA	En-Route
EUP (seconds)	5	10
At (nmi)	80	150

Table 4-2: EUROCAE Effective Update Period performance requirements

The probability that the state data is updated within k seconds with a given confidence can be derived from the binomial distribution

$$b(x; n, p) = \binom{n}{x} p^x (1 - p)^{n-x} \quad (\text{Eq. 4-1})$$

for: $n = k \times m$

where x is the number of received messages in n transmissions with the single message probability of reception p , and m is the broadcast rate. Solving for p where $x = 0$ and $b = 0.05$ results in

$$p = 1 - e^{\frac{\ln(0.05)}{k \times m}} \quad (\text{Eq. 4-2})$$

Thus, the minimum p meeting the EUROCAE requirements can be derived, as shown in Table 4-3. p for a given experiment can be set in the configuration file.

	TMA	En-Route
k	5	10
M	1	1
P	0.4507	0.2589

Table 4-3: Minimum allowable single message probabilities of reception

For every surveillance broadcast, a uniform random variable, rv , is drawn from the unit interval. If rv is less than or equal to p , the message is considered successful and is sent to the output interface; otherwise, the message is considered failed and the function exits without broadcasting the message.

Another source of surveillance error is estimation noise – any error between the estimated aircraft state made by the surveillance system, and the true aircraft state. As with own-ship estimation noise in the Navigation and Trajectory Module, provision has been made for this type of error through a placeholder function to add user-defined noise to the true aircraft state data when it is copied to the communications data structure. The function can be further developed as necessary; however, it can generally be assumed that surveillance data use by CD&R systems will have first passed through a tracker that filters and smoothes the data, thereby minimizing the effect of estimation noise.

4.5 Evaluation of the Module

Four tests were conducted on the Communications Module to verify correct operation, including: verifying the model of stochastic surveillance broadcast failures, verifying the model of stochastic communication latencies, and demonstrating trajectory exchange with an externally connected CD&R system.

4.5.1 Verification of Surveillance Broadcast Failure Model

Correct implementation of the ADS-B message failure model was tested with a flight lasting 86400 seconds using a 1 Hz surveillance broadcast rate, resulting in 86400 samples. The minimum en-route case was applied, where each broadcast was given a probability of success of 0.2589.

During the simulation, 22361 of the surveillance messages were received by *Tviz*; or 25.88% of the broadcasts. The mean period between updates was 3.9036 seconds and the 95th percentile was 10 seconds, correctly corresponding to the 95% confidence bound of the ADS-B EUP performance requirements.

4.5.2 Verification of Latency Modelling

Next, the latency model was tested in order to verify that it generates random variables according to the desired distributions. 10000 samples were taken of each of

the latency components: the datalink technical delay (uniform), datalink/ASAS response delay (lognormal), radio-telephony transaction duration (lognormal), and radio-telephony channel occupation (lognormal). A chi-squared goodness-of-fit test was conducted for each latency component for the null hypothesis that the observed latencies from the simulator came from the distributions described in Section 4.3. In each case the null hypothesis could not be rejected at the 5% significance level, indicating a good fit of the simulator’s latency distributions with the desired distributions. The p-value – the probability of observing the given statistic or one more extreme, assuming the null hypothesis – is shown in Table 4-4, and in all cases is greater than 0.05 (Walpole, et al., 2002).

For the lognormal distributions, the maximum likelihood estimates of the lognormal parameters, μ and σ , were computed with the 95% confidence intervals for the parameter estimates, in Table 4-4 (Walpole, et al., 2002). In each case, the confidence interval for the parameter estimates contained the desired parameters. For the uniformly distributed technical delay, the observed mean and standard deviation (*stdv*) are less than 0.0087 and 0.004 seconds, respectively, from the desired values.

Variate	p-value	Parameter	Desired	MLE	95% Interval
Datalink/ ASAS response delay	0.4240	μ	2.8314	2.8246	2.8047; 2.8444
		σ	1.0218	1.0131	0.9993; 1.0274
RT transaction duration	0.3459	μ	2.2543	2.2622	2.2522; 2.2722
		σ	0.5097	0.5121	0.5051; 0.5193
RT channel occupation delay	0.1225	μ	-0.5426	-0.5532	-0.5737; -0.5328
		σ	1.0417	1.0433	1.0291; 1.0580
Link technical delay	0.82	mean	2.0000	2.0087	N/A
		stdv	0.8660	0.8656	N/A

Table 4-4: Sampled latency distributions

The effect of latencies can be seen by the dispersal of aircraft trajectories in Figure 4-11. Sixteen identical aircraft were simulated on a heading of 90 degrees. A 45 degree heading change instruction was sent to the aircraft after 20 seconds of flight. One aircraft was simulated without any communications delays, whose flight track is shown

by the thick black line. The radio-telephony model was used for the remaining 15 aircraft, delaying the start of the turn and resulting in a cross track error from the zero-latency case. Conflict resolution algorithms must take this uncertainty into account when generating resolutions, and should be evaluated for robustness against variable communication delays.

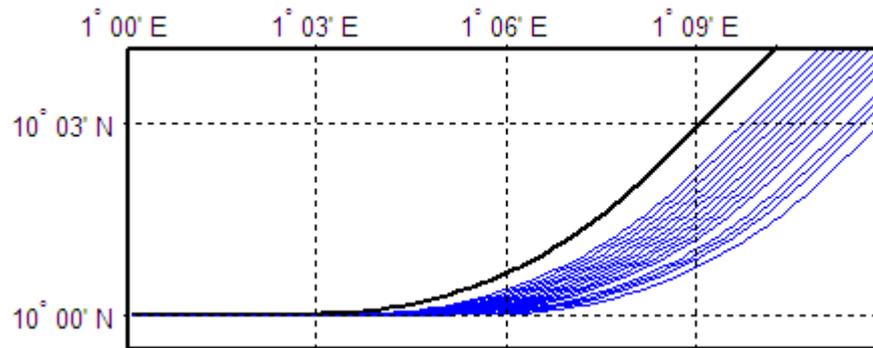


Figure 4-11: The effect of radio-telephony latency on a heading change instruction

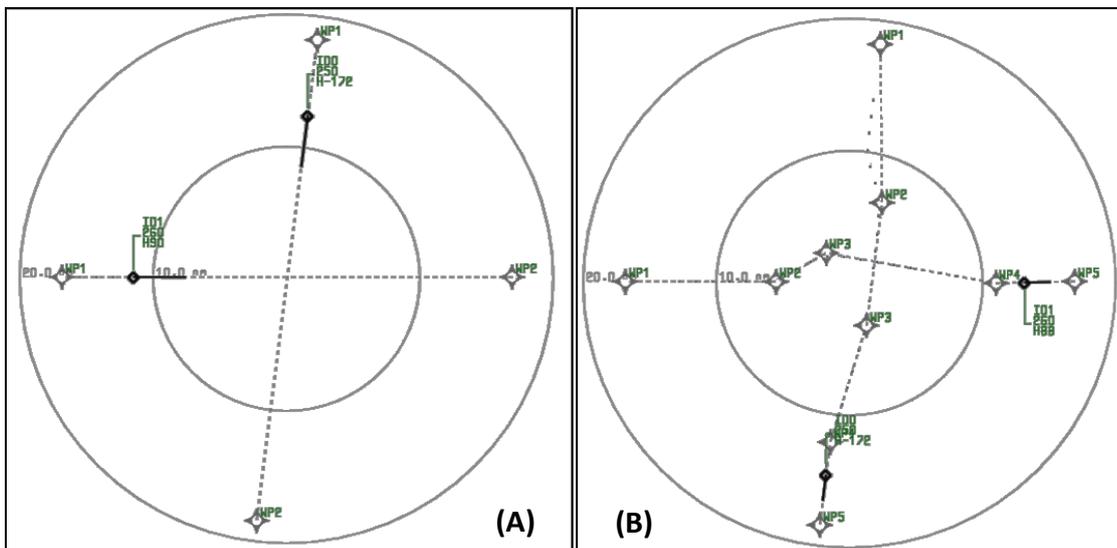
4.5.3 Verification of Trajectory Exchange

The final verifying test of the Communications Module was to connect the simulator to an automatic conflict detection and resolution system, in order to confirm the ability of the simulator to downlink aircraft surveillance and trajectory intent information to externally connected systems, and to uplink and respond to conflict resolutions from CD&R systems.

The simulator was connected to a centralised CD&R system under development at the University of Sheffield, designed to provide safe aircraft separation for up to 5 minutes into the future (Spence & Allerton, 2009). Conflict detection is provided through a linear prediction, state-based (does not account for aircraft intent) detection routine. The conflict resolver uses the genetic algorithm approach. The algorithm tests and costs a precompiled database containing a variety of horizontal manoeuvre sequences, in order to find suitable manoeuvre sequences for each involved aircraft while attempting to minimise off-track manoeuvring and aircraft separation incursions. The resolver uses a 5 minute look-ahead time when computing resolutions, but the

optimum time for the detector to flag a conflict is 150 seconds before the predicted conflict; at this point the genetic algorithm has the most choice and variety in the solutions. Input and output interface functions were developed to connect the CD&R system to the simulator using a TCP/IP connection, and the surveillance, flight plan, and trajectory modification commands data structures developed for the Smart-Skies project (Clothier & Walker, 2009). The speed-control method was used to maintain time synchronisation.

A simple two-aircraft crossing conflict scenario was developed, as shown in Figure 4-12A. The ability to downlink flight plans from aircraft to external systems was confirmed visually using TViz. Figure 4-12A shows a TViz screenshot before the conflict was detected. After the conflict was detected, resolutions were generated and automatically uplinked to the aircraft. The resolution flight plans can be seen in Figure 4-12B, confirming the ability of the simulator to accept and execute resolutions from externally connected CD&R systems.



**Figure 4-12: TViz screenshot before conflict resolution (A)
and after conflict resolution (B)**

4.6 Summary

The requirements for the Communications Module have been identified, and the communication process was described that enables external systems to monitor the traffic through ADS-B-like surveillance, and to control the traffic through instructions and route modifications. The evaluation of the module has shown correct implementation of the surveillance error model and the communications latency distributions. The flight plan merging algorithm was successfully demonstrated for a scenario that fails under simpler merging routines. The ability to integrate the simulator with a third-party CD&R tool, including the ability to output surveillance data and input conflict resolutions, was shown by connecting to a centralised, tactical CD&R system. The ability to generate internal messages and to respond to all message types in the message set (Section 4.2.1) was shown previously in Chapter 3.

Next, Chapter 5 will discuss the new functionality added to the Scenario Generation Module and evaluate the overall speed performance of the simulator.

Chapter 5

Scenario Generation

In this chapter, an approach is described to automatically generate simulation scenarios. The rationale for adding this capability to the Airspace Simulator-II is discussed in Section 5.1. Section 5.2 reviews the design of the pair-wise encounter scenario generator, followed by an evaluation in Section 5.3. The performance of the simulator when computing large traffic scenarios is then presented in Section 5.4.

5.1 Motivation

As discussed in Chapter 2, air traffic scenarios can be manually defined in a configuration file by listing each aircraft to be simulated, assigning each simulated aircraft a BADA performance type, a flight plan, a start time, a flight control mode, a communication type, and a separation mode. The aircraft assignment list, stored in the Master Array, allows arbitrary traffic scenarios to be specified in detail, including the scenarios used as case-studies by Hoekstra, et al. (2000), Pallottino, et al. (2002), Spence, et al. (2008), and Chaloulos, et al. (2008) to evaluate CD&R system performance. Although such case-studies can be useful for initial evaluations, due to the number of possible aircraft interaction geometries and scenario variables, CD&R concepts and algorithms must be tested over a large range of cases to analyze and quantify system behaviour and performance. Farley, Kupfer and Erzberger argue that “candidate algorithms must be stressed by traffic volumes, densities, and complexities that are commensurate with today’s busiest airspace as well as that of the envisioned future. Further, it is necessary to expose the algorithm to the full breadth and variety of conflict situations that occur in real-world operations now or in the foreseeable future” (2007). It is impractical to manually prepare flight plans and traffic assignments for a large number of aircraft, so developing an automatic scenario generation capability was an essential component of this thesis.

A review of the literature reveals three scenario types are predominately used to evaluate CD&R concepts and systems:

- *Scenarios based on historical data:* Real-world traffic scenarios can be extracted from recorded radar and flight plan data from the FAA or the European Central Flow Management Unit (CFMU), e.g. Farley, et al. (2007) and Vela, et al. (2010). In order to test systems at higher traffic densities and complexities than observed in current airspace, Paglione, et al., have developed a method of extrapolating historic traffic data to 2x and 3x the nominal level (2003). Unfortunately, such historic radar and flight plan data is generally not publically available to researchers, and furthermore, requires significant processing to extract and format the data for simulator use.
- *Pseudo-random pair-wise encounters:* A second common scenario type requires pairs of aircraft to be placed on flight paths that, if not resolved, will lead to a loss of separation. The majority of actual traffic conflicts involve only a single conflict pair; Bilimoria & Lee have reported that over 80% of conflicts in U.S. high altitude airspace (FL180 and above) involve only two aircraft and have no interaction with other aircraft (2001). As a result, this method is often used as part of a Monte Carlo-style experiment series to test resolvers over a range of conflict geometries, as with Blom, et al., (2007), Cetek (2009) and Chen and Zhao (2009).
- *Random traffic patterns:* Pair-wise encounters, however, do not fully ‘stress’ CD&R systems. Detecting potential conflicts and searching for conflict free routes is more complex when there are other aircraft in proximity. Consequently, scenarios of pseudo-random traffic patterns are also used to evaluate CD&R concepts and systems, for example, Cetek (2009), Spence, et al. (2008), Archibald, et al. (2008), and NASA’s Safety Performance of Airborne Separation experiment series (Consiglio, et al., 2007, 2008; Karr, et al., 2009). There are many different implementations of this method, but most use pseudo-randomly generated flight plans and aircraft to ensure a desired traffic level or density in a given airspace region.

Out of these three scenario types, it was decided to develop an automatic scenario generator for pair-wise encounters. Historical data scenarios were precluded due to the restricted source data and the implementation complexity. The pair-wise encounter method was included because it permits greater control over certain experimental variables (such as the encounter geometry, local winds, etc) than using historical data or using pseudo-random traffic patterns. Additionally, neither Blom (2007, Cetek (2009) nor Chen and Zhao (2009) detail their method of generating conflict pairs, thus, developing a rigorous pair-wise scenario generator for this thesis would also contribute to the subject.

Due to project time constraints the scenario generator currently only produce lateral scenarios – that is, all aircraft are at the same flight level. Bilimoria & Lee found that for approximately 75% of all encounters, both aircraft are in level flight, based on simulations using recorded radar data of US airspace operations (2001). It is possible, however, to extend the basic methods discussed in the next section to include vertical traffic scenarios so that the remaining 25% of encounters can be generated accurately.

5.2 Pair-Wise Conflict Scenario Generator

A Loss of Separation (LOS) event occurs when the distance between two aircraft is less than the required separation minimum. The pair-wise conflict scenario generator was designed to automatically prepare two-aircraft LOS encounters by pseudo-randomly:

- selecting BADA performance types, flight control modes, communication modes, and separation modes;
- creating flight plans such that a LOS event will occur unless the conflict is resolved;
- controlling start times, so only one aircraft pair is simulated at a time.

A configuration text file was developed to contain the key scenario design parameters that can be controlled by the user (an example is provided in Appendix D). An overview of the scenario generation process is shown in the pseudo-code of Figure 5-1

and is described in more detail in the following sections. Section 5.3 then presents a verification and discussion of the implementation.

```
READ configuration file
i = 0
FOR i < (length of Master Array * 2)
  DO Ac1 = random aircraft type
  WHILE (Ac1 ceiling < scenario altitude)
  DO Ac2 = random aircraft type
  WHILE (Ac2 ceiling < scenario altitude)

  Modes1 = random comm., flt. control, separation modes
  Modes2 = random comm., flt. control, separation modes

  Fp1 = compute first flight plan
  WRITE Fp1 to file
  EncAng = random encounter angle
  Radial = random radial from scenario centre
  Fp2 = compute second flight plan
  WRITE Fp2 to file

  Master Array[i] = [Ac1, Modes1, Fp1]
  Master Array[i+1] = [Ac2, Modes2, Fp2]
  i = i+2
END
```

Figure 5-1: The pair-wise scenario generation process

5.2.1 Aircraft Type Selection

The first step of the scenario generation process is to assign aircraft types from the BADA database; which is needed to determine the performance of the aircraft involved in the conflict. The scenario generator pseudo-randomly selects aircraft types, saving them into the Master Array. Two selection modes were developed, where the choice between the two modes can be set in the configuration file.

The first mode uses a uniform distribution – that is, every aircraft type has an equal probability of selection. However, if the scenario altitude (the initial altitude of all aircraft in the scenario, as specified in the configuration file) is above the ceiling of the selected aircraft type, then a different type is selected.

The second mode models the frequency distribution of aircraft types in European airspace. The CFMU maintains daily records of aircraft types accessing European airspace, which have been correlated to aircraft in the BADA database by Sheehan (2009). Currently, ten types dominate European air traffic, accounting for nearly 50% percent of the total traffic as shown in Table 5-1. The observed frequencies of the remaining 308 BADA-modelled types are listed in Appendix-A of Sheehan (2009). The CFMU statistics were implemented in the simulator such that the probability of selecting a given BADA type matches the frequency distribution of European air traffic. Again, if the scenario altitude is above the ceiling of the selected aircraft type, then a different type is selected.

Rank	BADA Type	Full Name	% Total Traffic	Cumulative %
1	A320	Airbus A-320	11.37%	11.37%
2	B738	Boeing 737-800	9.55%	20.92%
3	A319	Airbus A-319	8.58%	29.50%
4	B733	Boeing 737-300	3.96%	33.46%
5	A321	Airbus A-321	3.92%	37.38%
6	AT72	ATR-72	2.75%	40.12%
7	B737	Boeing 737-700	2.58%	42.70%
8	B734	Boeing 737-400	2.44%	45.15%
9	CRJ2	RJ-200 Regional Jet	2.37%	47.51%
10	B735	Boeing 737-500	2.37%	49.88%

Table 5-1: The ten most frequent aircraft types in European airspace

5.2.2 Mode Selection

The communications mode, flight control mode, and separation responsibility are chosen for each aircraft, and are saved in the Master Array. The mode selections are made randomly according to probabilities defined in the configuration file, giving the user control over the resulting distributions. For example, 75% of the aircraft could be set to datalink communications and the remaining 25% set to radio-telephony communications.

5.2.3 Flight Plan Generation

Flight plans are created for each conflict pair such that a LOS event will occur unless the conflict is resolved, as shown in Figure 5-2.

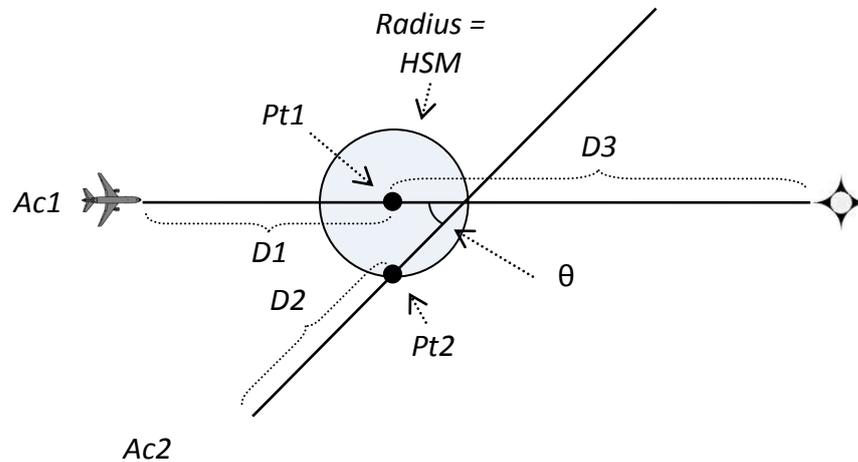


Figure 5-2: A pair-wise conflict scenario at initialisation

The flight plan generation method was designed to provide a large variety of encounter geometries, while giving the user control over four parameters:

- *Distribution of encounter angles, ϑ* : The encounter angle is a commonly-used parameter to characterise conflict geometry. For example, Rantanen, et al., (2006) have demonstrated that controllers prefer different resolution strategies based on the encounter angle (such as the use of vectoring for acute angle conflicts), while Bilmoria uses a geometric optimisation approach for conflict resolution that produces ‘families’ of solutions based on the encounter angles and speeds (2000). Clearly, it is important to allow the user to specify the distribution of encounter angles in the configuration file.
- *Horizontal separation minimum, HSM*: Required separation minimums vary according to airspace (Porras and Parra, 2007), so it is also important to allow the user to specify the HSM in the configuration file.
- *Time of first loss of separation, t_{LOS}* : This parameter is used to control the amount of time available for the CD&R system to detect and resolve the conflict. A small t_{LOS} can be used to test the CD&R system for ‘pop-up’ conflicts that are detected only at very short notice, while larger t_{LOS} can be used to

simulate conflicts detected at a long distance. Specifying t_{LOS} in the configuration file allows the user to ‘stress’ the CD&R system appropriately.

- *Distance to the final waypoint, D3*: This parameter is used to control the amount of ‘manoeuvring space’ beyond the conflict. In some cases, a small $D3$ will force the resolver to implement more drastic manoeuvres than a large $D3$ in order to clear the conflict while also satisfying the waypoint constraint. As a result, it was decided to make this a user-defined parameter in the configuration file.

The flight plan generation method is based on the relationships shown in Figure 5-2. Point 1 ($Pt1$) and point 2 ($Pt2$) are the locations of aircraft 1 ($Ac1$) and aircraft 2 ($Ac2$) when loss of separation occurs, t_{LOS} . $Pt1$ and $Pt2$ are not waypoints in the flight plan, but are the planned aircraft locations at the start of the conflict, around which the flight plans are created. The coordinates of $Pt1$ and the course of $Ac1$ at that point, $Crs1$, are specified in the configuration file, fixing the geographic location of the scenario. For each conflict pair, the slower aircraft is always considered $Ac1$ in order to ensure that for small encounter angles the faster aircraft is properly configured to overtake the slower aircraft.

The flight plan for $Ac1$ is written first. Given $Pt1$ and $Crs1$, the initial waypoint is found by projecting a point a distance of $D1$ in the initial bearing $Crs1+\pi$ from $Pt1$ using Equations 3-8 and 3-9. Knowing the true airspeed of $Ac1$ from the BADA data and the wind velocity vector at $Pt1$, the ground speed of $Ac1$ can be found from Equations 3-2 and 3-3. Assuming a constant wind field, $D1$ can be found so $Ac1$ will be located at $Pt1$ at time t_{LOS} , as

$$D1 = V_{Gnd} * t_{LOS} \quad \text{(Eq. 5-1)}$$

where t_{LOS} is the desired time between aircraft initialisation and the first loss of separation. Similarly, $Ac1$ ’s second waypoint is projected a distance of $D3$ in the initial bearing $Crs1$ from $Pt1$.

Once both waypoints are found for $Ac1$, they are written to a text file in the flight plan format discussed in Section 3.2.1, and the flight plan name is saved in the Master

Array. By default, the flight plans legs are written as track-to-fix (*TF*) with en-route (*ENR*) manoeuvring, at an altitude given in the configuration file. Lateral offset and speed control are set to zero.

Next, the flight plan for *Ac2* is generated, adding stochastic variation to the conflict geometry through the pseudo-random selections of the encounter angle and the location of *Pt2*.

The encounter angle is selected according to the distributions specified in the three encounter angle bins in the configuration file: in-trail (0-60 deg), crossing (60-120 deg), and opposing (120-180 deg). These are the bins typically used to categorize conflict geometries. For example, to evenly sample all encounter angles, the bins can each be set to 33.33%. Alternatively, Bilimoria & Lee reported the distribution of encounter angles in high altitude U.S. airspace (2001), shown in Table 5-2. Using the values in column 3 will produce a more realistic distribution of encounter angles.

Bin Range (deg)	Encounter Type	% Total Encounters	Cumulative %
0-60	In-trail (Passing)	50%	50%
60-120	Crossing	20%	70%
120-180	Opposing	30%	100%

Table 5-2: Distribution of encounter angles (Bilimoria & Lee, 2001)

The scenario generator pseudo-randomly selects a bin with an associated probability, and samples uniformly from the range of the selected bin. The result is then multiplied by either ± 1 (with equal probability) so that both hemispheres are included. Thus, *Crs2*, the course of *Ac2* at *Pt2*, can be found from the normalised angular difference between *Crs1* and the encounter angle.

Next, the location of *Pt2* must be found such that the initial LOS occurs at time t_{LOS} . This requires *Pt2* to be located somewhere on the perimeter of a circle centred on *Pt1* with radius equal to the horizontal separation minimum.

However, the relative motion of $Ac2$ in respect of $Ac1$ must be taken into account in order to restrict $Pt2$ to the relevant hemisphere, as shown in Figure 5-3.

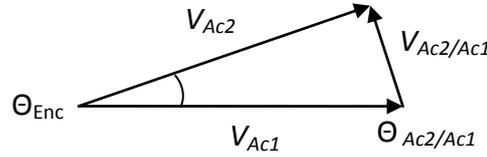


Figure 5-3: Velocity triangle geometry

First, the vector component of $Ac2$'s velocity in the direction of $Ac1$ can be found from the Law of Cosines:

$$V_{Ac2/Ac1} = \sqrt{V_{Ac1}^2 + V_{Ac2}^2 - 2V_{Ac1}V_{Ac2} \cos \theta_{Enc}} \quad (\text{Eq. 5-2})$$

Next, the Law of Sines can be used to determine the relative angle:

$$\theta_{Ac2/Ac1} = \text{asin} \left(\frac{V_{Ac2} \sin \theta_{Enc}}{V_{Ac2/Ac1}} \right) \quad (\text{Eq. 5-3})$$

To test for and correct the ambiguous case of Arcsine when determining $\theta_{Ac2/Ac1}$ (for obtuse $\theta_{Ac2/Ac1}$), the Law of Cosines is re-applied to check for consistency:

$$\theta_3 = \pi - \theta_{Enc} - \theta_{Ac2/Ac1} \quad (\text{Eq. 5-4})$$

$$V_{Ac1 \text{ Derived}} = \sqrt{V_{Ac2/Ac1}^2 + V_{Ac2}^2 - 2V_{Ac2/Ac1}V_{Ac2} \cos \theta_3} \quad (\text{Eq. 5-5})$$

$$\begin{aligned} & \text{if } (V_{Ac1 \text{ Derived}} \neq V_{Ac1}) \\ & \text{then } \theta_{Ac2/Ac1} = \pi - \theta_{Ac2/Ac1} \end{aligned} \quad (\text{Eq. 5-6})$$

Finally, the location of $Pt2$ can be found by projecting a point by the distance of HSM from $Pt1$ in an initial bearing chosen from a uniform distribution between a lower and upper limit, as shown in Figure 5-4:

$$\text{Lower Limit} = \text{Crs1} + \theta_{Ac2/Ac1} - \frac{\pi}{2} \quad (\text{Eq. 5-7})$$

$$\text{Upper Limit} = \text{Lower Limit} + \pi \quad (\text{Eq. 5-8})$$

These steps of restricting the hemisphere of $Pt2$ were necessary to ensure that the initial LOS will occur at time t_{LOS} .

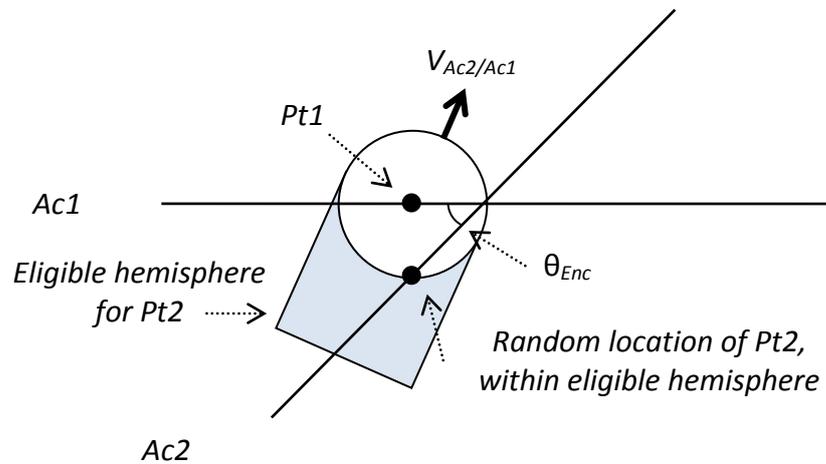


Figure 5-4: A pair-wise conflict scenario at initialisation

Once the encounter angle and $Pt2$ have been selected, the flight plan for $Ac2$ can be created in a similar way to the flight plan for $Ac1$. The initial waypoint is found by projecting a point a distance of $D2$ in the initial bearing $Crs2+\pi$ from $Pt2$, where $D2$ is found from Equation 5-1. The second waypoint is projected a distance of $D3$ in the initial bearing $Crs2$ from $Pt2$. The waypoints are then written to a text file and the flight plan name is saved in the Master Array.

This approach to generating flight plans produces a large variety of encounter geometries, while also preventing aircraft from being initialised with an immediate loss of separation – a simulation artefact that could cause erroneous results in experiments. For example, starting aircraft in conflict, which tends to occur with small encounter angles, could wrongly indicate that a given CD&R system is ineffective for certain geometries. Attempting to avoid the problem by simply eliminating small encounter angles would reduce variety in the solutions and would introduce a systematic gap to the scenario generator's coverage. However, taking into account the aircraft relative motion and controlling the point of first loss of separation, as described above, eliminates the artefact without restricting coverage.

5.2.4 Start Time Control

At this point in the scenario generation process, the final data field in the Master Array that has not been assigned a value is the start time. Rather than specifying the start time for each conflict pair during the setup phase of the simulation – as is the case when the scenario generator is not used – the Master Array is left sparse. The next conflict pair is only initialised once both aircraft have completed their flight plans (at which point their start times are recorded in the Master Array). This restricts the simulation to a single conflict pair at a time, minimising the computational load on the CD&R system.

5.3 Verification and Discussion of Scenario Generator

This section presents a summary and discussion of the evaluation and analysis effort undertaken to:

- Verify correct implementation of parameters described in the configuration file;
- Verify that the methodology used to generate flight plans satisfies the user parameters and produces variety in the encounter geometries;
- Analyse the fast-time performance when using the scenario generator.

In order to test the scenario generator, a simulation was run for 2000 conflict pairs. The scenario generator was configured as follows. Time to first loss of separation was set to 4 minutes (240 s). The horizontal separation minimum was set to 5 nmi, and the distance from first loss of separation to the final ways point was 20 nmi. The scenario altitude was FL320, and the traffic types were randomly selected according to Sheehan's European traffic model. Encounter angles were randomly selected according to Bilimoria & Lee's upper-airspace model values. All aircraft used ideal navigation, and were un-separated (that is, no CD&R services were used). The communications mode was set to 80% datalink and 20% radio-telephone. The simulator was configured to a 1 second time step and 4000 total aircraft. A *Tviz* screen

second, an analysis of the Closest Point of Approach (CPA) in order to verify the variety in the resulting geometries.

The 2000 recorded t_{LOS} samples had a mean of 237 seconds and standard deviation of 12.2 seconds, compared to the desired value of 240s. However, the scatter plots in Figure 5-6 show that the spread of the sample points was larger for small encounter angles and speed differences.

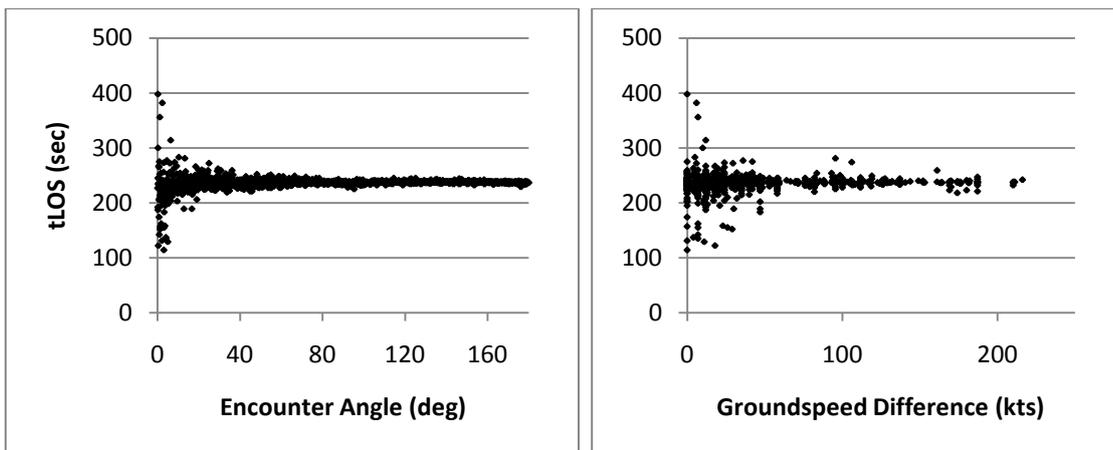


Figure 5-6: Scatter plots of t_{LOS} verses encounter angle and speed differences

To further analyse the data, the samples were sorted into a 2D matrix by encounter angle, X_{EA} , and speed difference, $X_{\Delta V}$. A factorial regression was conducted to fit the data to:

$$Y = \beta_0 + \beta_1 * X_{EA} + \beta_2 * X_{\Delta V} + \beta_3 * X_{EA} * X_{\Delta V} \quad (\text{Eq. 5-9})$$

The regression showed no significant trends for the *mean* of the binned data – that is, the average t_{LOS} value did not vary with encounter angle or speed difference. However, analysis of the t_{LOS} *standard deviation* showed significant main effects and interaction, as shown in Table 5-4 – t_{LOS} varied the most when *both* the encounter angles and speed difference were small. This interaction can be explained: for slowly overtaking flights, the relative velocity vector of $Ac2$ in respect to $Ac1$ is very small. As a result, the *initial* aircraft separation is close to the separation minimum, from Equation 5-1. Thus, any numerical errors or rounding errors (such as those due to writing the waypoints to text files and then reading them during initialization) significantly affects t_{LOS} .

	β_0	B_1	B_2	B_3
Coefficient	3.26	-3.04	-1.46	1.45
p-Value	7.07E-17	5.15E-15	7.74E-05	0.000128

Table 5-4: Regression results of t_{LOS} standard deviation

An additional result of numerical and rounding errors is non-conflict scenarios – that is, scenarios where an LOS event did not occur. However, this accounted for only 5% of all cases, and the mean miss distance was less than 200 metres from 5 nmi. These events typically occurred when $Pt2$ was placed near the points where the relative velocity vector was tangential to the separation minimum circle around $Pt1$. Overall, the results of the t_{LOS} analyses indicate the methodology of generating flights plans as described in Section 5.2.3 is generally satisfactory to control t_{LOS} .

The closest point of approach was also recorded for each conflict pair; the histogram is shown in Figure 5-7. The mean CPA was 3.24 nmi with a standard deviation of 1.5 nmi. Regression analysis did not reveal any significant trends; the mean and variance of the binned CPA data did not change significantly with either encounter angle or the speed difference, indicating independence and that the scenario generation method is not biased toward certain conflict geometries.

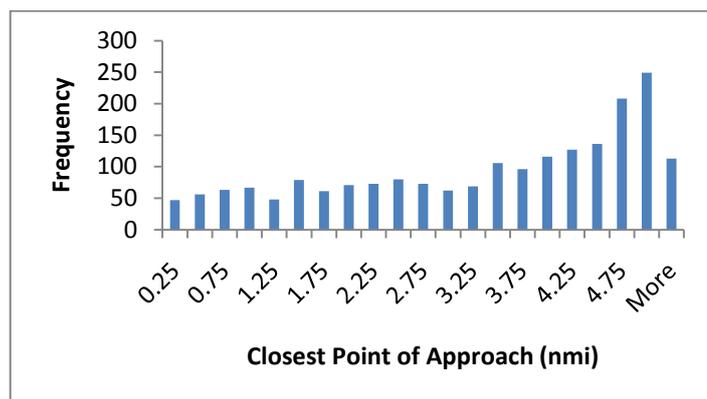


Figure 5-7: CPA histogram

During these tests, t_{LOS} was relatively small, 240 seconds, representing last-minute conflicts not previously identified or corrected by long-distance CD&R systems. To

determine if the conflict miss rates and t_{LOS} standard deviation are influenced by the size of t_{LOS} , the test was repeated for t_{LOS} of 15 minutes (900 s), with all other simulator and scenario parameters remaining the same. The resulting t_{LOS} standard deviation did not significantly differ, increasing by only 1.3 seconds. The non-conflict scenario rates were also similar, increasing by only 0.1%. These results indicate that the selection of t_{LOS} does not significantly affect the ‘correctness’ of the solutions.

5.3.3 Fast-Time Performance

Next, the fast-time performance of the simulator was evaluated in order to determine the component of execution time that can be attributed to the scenario generator verse the aircraft trajectory generation (i.e. the actual simulation). A scenario generator that requires an excessive amount of time to compute could marginalize the usefulness of a fast-time simulator.

The scenario generator and the resulting 4000 simulated flights were executed three times. The total program execution time averaged 37.3 seconds with a standard deviation of 2.5 seconds on a laptop computer using a 2.2GHz Intel Core2 Duo processor with 2 GB of RAM. On average, approximately 40% of the execution time (14.8 s) was attributed to the scenario generator, and 60% (22.5 s) was attributed to all other simulation tasks such as initialization, simulation, and shutdown. This speed and percentage breakdown was considered acceptable given the number of flight plans generated and the number file writing operations.

This evaluation also demonstrates that the simulator satisfies the design requirement of computing 4000 *total* aircraft faster than real time. Under this configuration, a total of 813,460 simulated seconds (225 hours, and 1,626,920 discrete movements) were computed in 37.3 seconds; a performance of more than 20,000 times faster than real-time. The second component of the requirement – computing at least 300 *concurrent* aircraft in fast-time – will be discussed in Section 5.4.

5.3.4 Discussion

The primary advantages of this scenario generator are that:

- 1) Conflicts are ‘forced’ to occur for over 95% of scenarios, and as a result, resources are not wasted simulating ‘non-events’;
- 2) Scenario parameters can be easily specified by the user, permitting many experimental variables to be controlled;
- 3) A wide variety of scenario geometries are produced.

The limitation of this approach, however, is that the complexity of the conflict detection and resolution problem is reduced by restricting the simulation to two aircraft. In order to provide a means of testing CD&R systems under more complex situations, a random route-type scenario generator should be used.

5.4 Verification of Fast-Time Performance

At this point in the thesis, all the simulator requirements have been demonstrated except the ability to run faster than real-time for at least 300 simultaneous aircraft. To accomplish this and to establish a performance baseline, two tests were conducted: short-distance flights, and long distance flights.

The flights are summarized in Table 5-5. The Madrid/Barcelona airport pair was chosen to represent short distance flights because it was the most popular airport pair within the EU-27 region (De La Fuente Layos, 2009). Similarly, the Heathrow/JFK pair was selected to represent long-distance flights, since it was the most popular extra-EU airport pair. The number of waypoints in a flight plan does not significantly influence computation time, thus, simplified ‘direct-to’ flight plans were used, initiated at ground level.

Departure	Destination	Aircraft Type	Cruise Alt (FL)	Distance (nmi)	Flight Duration (sec)
Madrid Barajas	Barcelona	CRJ2	350	261	2115
London Heathrow	New York JFK	B773	380	2993	22356

Table 5-5: Flight summaries for concurrent aircraft test

The concurrent number of aircraft flying these routes was increased from 10 to 2500. Each simulation (consisting of n concurrent aircraft) was repeated 3 times. The results of the test series is shown in Table 5-6, and are plotted in Figures 5-8 and 5-9.

	Concurrent Aircraft, n	Execution Time (sec)		Total Discrete Events	Fast-Time Gain
		Mean	Stdv.		
Short Distance	10	0.3	0.0	21155	7052
	50	1.3	0.0	105773	1627
	100	2.5	0.0	211546	846
	500	12.2	0.2	1057732	173
	1000	24.2	0.2	2115463	88
	2500	61.1	0.4	5288658	35
Long Distance	10	2.6	0.0	223558	8598
	50	12.3	0.2	1117791	1813
	100	24.2	0.3	2235582	923
	500	120.1	0.9	11177912	186
	1000	254.1	4.4	22355825	88
	2500	658.3	10.2	55889562	34

Table 5-6: Concurrent aircraft test results

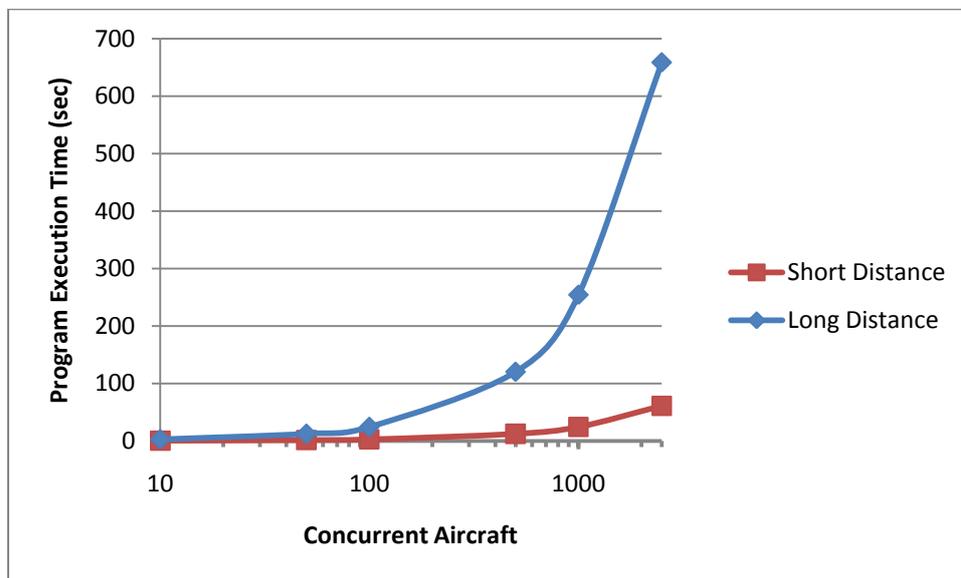


Figure 5-8: Execution time per concurrent aircraft

Figure 5-8 shows the long distance simulations took significantly longer to compute than the short distance flights. This is expected, because the longer distance (and thus,

longer duration) trans-Atlantic flights resulted in more discrete events and computations (as can be seen in the Total Discrete Events column of Table 5-6). The fast-time gain, plotted in Figure 5-9, is the number of simulated seconds divided by program execution time. It can be seen that the number of concurrent aircraft is inversely proportional to the fast-time gain, and the duration of the flight is insignificant. Again, this relationship was expected due to the nested loop in the simulator architecture (the outer airspace loop and the inner aircraft loop), resulting in the quadratic performance behaviour. The test shows the simulator ran over 900 times faster than real-time for 300 simultaneous aircraft and over 30 times faster than real-time for 2500 simultaneous aircraft, satisfying the design requirement. With these results, it is believed that the limiting factor in terms of execution time during CD&R simulation experiments will be the performance of the attached CD&R system, rather than the airspace simulator.

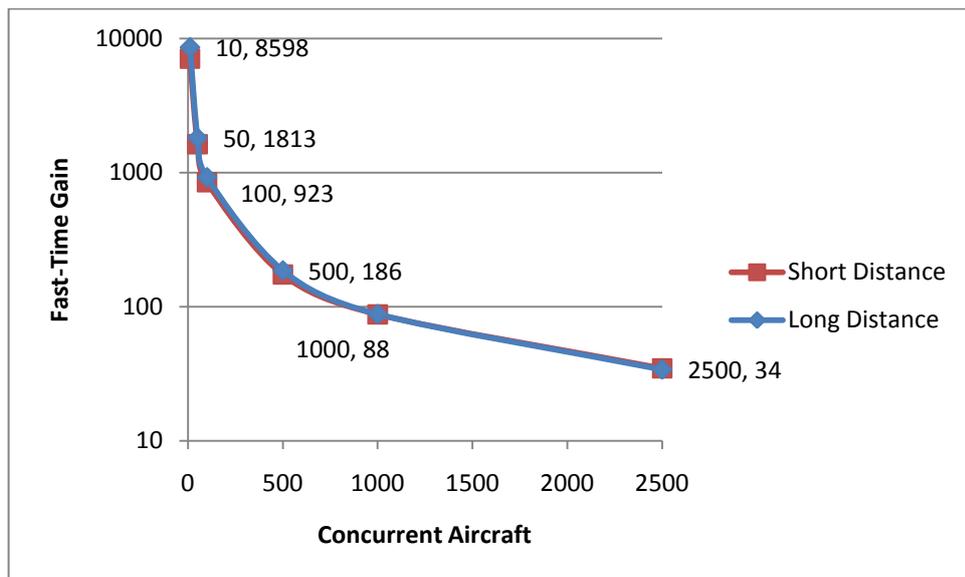


Figure 5-9: Fast-time gain per concurrent aircraft

5.5 Summary

This chapter has argued the need for an automatic scenario generation capability to prepare flight plans and traffic assignments for a large number of aircraft. The design of the pair-wise scenario generator was outlined, which can be used to test

CD&R methods and systems over a large variety of two-aircraft encounters while allowing control over many experimental variables. The test described in Section 5.3 demonstrated correct operation of a pair-wise scenario generator and confirmed the ability of the simulator to compute up to 4000 aircraft total faster than real time for 225 flight hours.

The overall speed performance was assessed for up to 2500 simultaneous aircraft, and the relationship was established between the program execution time, the number of concurrently simulated aircraft, and the duration of the flights. The simulator was able to compute the trajectories of 2500 concurrent aircraft flying nearly 3000 nmi in less than 11 minutes – that is, nearly 34 times faster than real-time.

Next, chapter 6 will describe the application of the simulator to evaluate the potential for vector navigation to provide a means to control aircraft lacking datalink capability in a more highly automated ATM system.

Chapter 6

Research Application - Vector Navigation

The application of the simulator to an active ATM research question was considered an integral part of this thesis in order to:

- Confirm the simulator meets the goal of a simple, flexible, and accessible fast-time simulator suited for exploratory separation management research;
- Demonstrate the utility of the simulator with a novel application;
- Identify strengths and weaknesses of the simulator design.

Thus, this chapter does not directly discuss the development of the Airspace Simulator- II, but uses the simulator as a platform to provide initial insight into controlling air traffic containing aircraft that are not equipped with datalink in highly automated ATM systems. Section 6.1 presents the need for automated support for separating mixed-equipage traffic. Section 6.2 then introduces the design and implementation of a simple method of automatically providing track angle navigation through verbal vectors. Finally, an evaluation of the prototype system is presented in Section 6.3, addressing the experiment design, the simulation results and a discussion of the findings.

6.1 Automation Support for Mixed-Equipage Traffic

The NASA NextGen-Airspace Project has studied the effect of integrating mixed-equipage traffic in the same airspace (Doble, et al., 2005; Prevôt, et al., 2008; Kopardeker, et al., 2009). As discussed in Section 1.3.2 of this thesis, Doble, et al., conducted a controller-in-the-loop simulation study of a mixed ASAS and IFR operations, and found that integrating self-separating aircraft and centrally-controlled traffic in the same airspace may be feasible within certain constraints (2005). However, the study did not attempt to establish the limits of feasibility or assess the implications for airspace structuring.

To address these issues, Prevôt, et al., conducted a follow-on study of an operational concept where ground-based CD&R automation was used to control traffic that was equipped with an FMS-integrated datalink, while air traffic controllers managed the remaining aircraft using current IFR methods and the limited aid of CD&R automation (2008). The ground automation could be used to detect conflicts for both the datalink-equipped and voice-only traffic, as long as the voice-only aircraft maintained their original flight plan. Additionally, controllers could request the automated system to produce a turn-point resolution for unequipped aircraft.

Prevôt used four controller-in-the-loop simulations with 0, 15, 30, and 45 datalink-equipped aircraft in a sector. Over the course of these simulations, the unequipped traffic count was increased linearly from 5 to 20 aircraft. The results of the study suggest that “a limited number of IFR aircraft may be manually controlled in the same airspace as a potentially large number of aircraft that are controlled by a different entity – the ground automation in this case.” However, Kopardeker, et al., further examined the data and found that the complexity of the IFR (voice-only) traffic was a significant limiting factor (2009). Specifically, workload was very high when IFR traffic was being manoeuvred off the original trajectory, because the controller was required to closely monitor these aircraft in order to issue a turn back clearance at the appropriate time and to detect possible conflicts. Controllers in the study indicated that they could only safely manage a maximum of three such aircraft concurrently before the airspace became too complex, however, if IFR aircraft maintained their course, then up to twelve aircraft could be managed effectively.

6.1.1 The Need for Improved Automation Support

Kopardeker highlighted the difficulty of monitoring aircraft in different states in a mixed airspace environment, suggesting that an improved automation tool with “an ability to monitor the turn back point in the voice-initiated lateral route change could lessen the overall monitoring workload and increase safety.” Rantanen, et al., drew a similar conclusion when analysing controller resolution manoeuvre preferences (2006). Vectoring was considered the least favoured resolution strategy by controllers

because it adds to their workload – due to the additional duty of monitoring and managing navigation.

Thus, in order to support traffic not equipped with an FMS-coupled datalink, there is need for a Resolution Monitoring and Advising Tool (RMAT) that could:

- Accurately implement lateral resolution manoeuvres generated by CD&R systems through voice communications;
- Reduce the task of monitoring aircraft not on the original trajectory.

RMAT could facilitate mixed-equipage operations and could also form the basis of a backup system in the event of datalink failure by reducing reliance on human cognition to manage separation. Additionally, with automated support, vectoring could become a more favourable resolution option for controllers, which is important as the airspace becomes more crowded.

Several ground-based decision support tools have been previously developed that include provision for verbal lateral route change advisories, including the Problem Analysis, Resolution and Ranking (PARR) tool, the TSAFE CD&R system, the suite of PHARE Advanced Tools (PAT), and the En Route Descent Advisor (EDA). These systems were examined to determine if they could meet the need for improved automation support of unequipped traffic.

PARR is a decision support tool developed by MITRE CAASD as an enhancement to the URET tool, providing conflict resolution advisories for conflicts detected up to 20 minutes ahead (Kirk, et al., 2001). For short-term CD&R, TSAFE is a tool developed by NASA to provide a ‘safety net’ for situations where the loss of separation is predicted to occur in less than 3 minutes (Erzberger & Paielli, 2002). PAT is a suite of DSTs designed by EUROCONTROL to demonstrate the merits of air-ground integration, including conflict detection and resolution tools and traffic flow management functions (van Gool & Schröter, 1999). One design feature that PARR, TSAFE, and PAT have in common is that for aircraft not equipped with datalink, lateral trajectory modifications are displayed to the controller as heading vectors. Heading vectors

specify the magnetic heading to be followed by the aircraft. If the wind direction and speed are known, a heading can be derived that will provide the ground track angle of a given leg of a lateral manoeuvre.

A limitation of heading guidance, however, is that it is susceptible to wind prediction errors, resulting in trajectory prediction uncertainty. Additionally, PARR, TSAFE, and PAT do not provide automatic monitoring of aircraft – that is, the controller must keep track of when to issue both the turn-away vector and the turn-back instruction. These limitations support concerns that the CD&R tool used in the Kopardeker study was not acceptable for aircraft vectored off their original trajectory.

The CTAS EDA attempts to overcome these limitations by taking advantage of the RNAV capability of modern FMS. The EDA tool, developed by NASA, provides terminal controllers with advised speeds, turns and descents to achieve timed arrivals in the terminal area (Green & Vivona, 2001). If speed control is insufficient to achieve spacing or separation, the EDA calculates a lateral path stretch manoeuvre. The position of the manoeuvre points are verbalised in EDA clearances relative to existing waypoints by a bearing relative to magnetic north and a distance on the bearing. This is the Place, Bearing, Distance (PBD) method. The advantage of PBD is that the manoeuvre can be accurately flown using FMS lateral navigation. The disadvantage of PBD, however, is that defining multiple waypoints in this way can result in highly complex clearances, especially if combined with other message elements such as speed instructions to meet a time of arrival constraint (Schoemig, et al., 2006). Multiple studies have demonstrated that message complexity is strongly correlated with miscommunication and operating errors (Loftus, et al., 1979; Grayson & Billings, 1981; Morrow & Rodvold, 1993; Bürki-Cohen, 1996). Air traffic controllers have been warned that even for named waypoints from predefined databases, errors in manual waypoint entry are the single most common cause of pilot 'blunder' errors in RNAV operations (EUROCONTROL, 2010c).

To summarize, heading vectors are familiar to pilots and controllers and use simple phraseology, but can result in inaccurate flight guidance. PBD waypoints can produce

accurate flight guidance through RNAV, but at the cost of increased message complexity. As a result, neither of these methods are ideally suited for automation support of mixed-equipment operations in high density, high complexity airspace.

However, a third possible method is available that may be able to address these limitations. Many modern automatic flight control systems have sufficient functionality to control the direction of the inertial velocity vector (track angle) in addition to the air-mass velocity vector (heading), by correcting for wind drift (Vakil & Hansman, 2002; Lambregts, 1998; Roskam, 2003), as measured by the inertial navigation system, GNSS, or Doppler radar. Thus, the desired ground track of any leg of a lateral conflict resolution could be specified directly as a track angle vector, making the instruction more robust against the effects of wind estimation error in the ground automation. If track angle vectors were automatically generated and executed at the appropriate time, aircraft using this method could closely mimic the trajectory of datalink-enabled aircraft, enabling the same trajectory-based separation management system to be used for both datalink equipped and voice-only traffic, as shown in Figure 6-1. Controllers could be automatically signalled to issue these vectors at the required time, significantly reducing the monitoring task, improving turn-timing precision, and providing a means of avoiding full reliance on human situational awareness. This control approach may provide a way to take advantage of the simplicity and familiarity of vector navigation, but with reduced trajectory error.

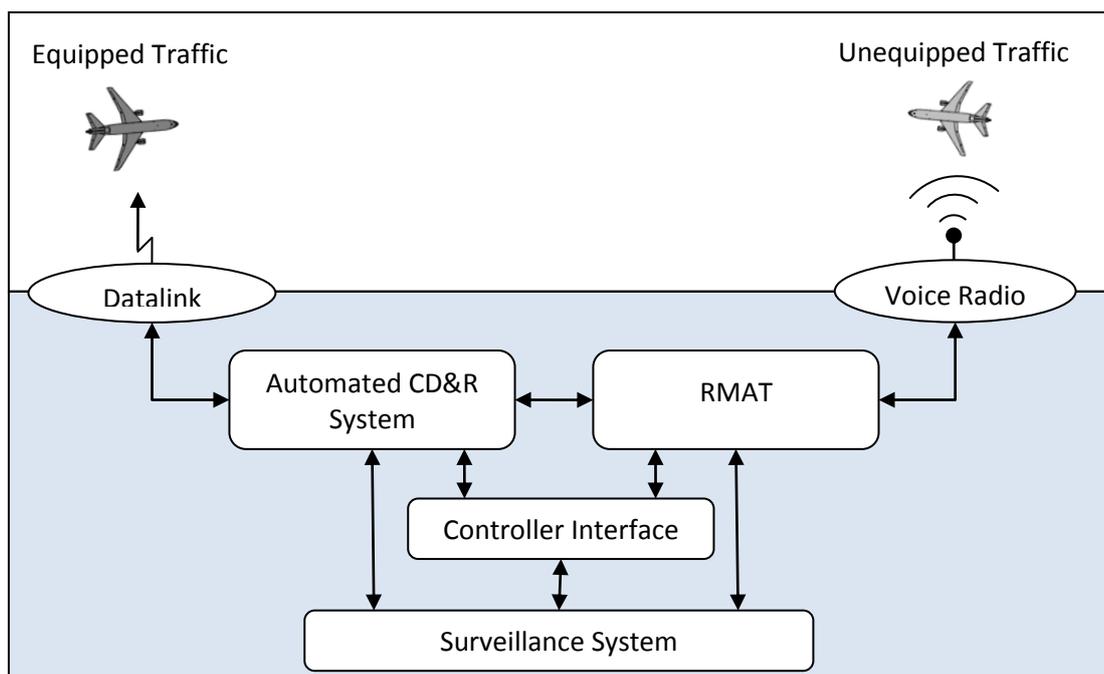


Figure 6-1: Mixed-equipage operations using an RMAT system to interface voice-only aircraft with the CD&R system

It was decided to develop a prototype RMAT system that would use track angle vectors to provide semi-automatic lateral flight guidance to non-datalink traffic. The prototype was implemented in the Airspace Simulator-II, and evaluated using stochastic simulations to assess the design and to evaluate effect of communications timing uncertainty in verbally issuing the vectors.

6.2 Design of an RMAT System

The prototype RMAT system was designed to issue the vectors progressively as the aircraft completes the manoeuvre, in contrast to providing one message that contains information for every leg of the resolution trajectory (i.e. vectors and execution times). Progressive vectoring has two advantages. First, it avoids the complexity of issuing multiple commands in one transmission. Morrow and Rodvold have shown that messages with more than two commands increase pilot requests for clarification and the number of incorrect readbacks (1993). Secondly, progressive vectoring allows the system to control any trajectory errors at subsequent waypoints. However, progressive

vectoring comes at the cost of additional radio transmissions, requiring at least one transmission per leg of the resolution manoeuvre. Despite the additional transmissions, the controller monitoring task can still be reduced by automatically issuing visual or aural alerts of upcoming transmissions.

Timing the execution of a track angle change instruction is also an important design consideration to ensure the aircraft completes the turn on the desired path; turning early or late will lead to a cross track error. Standard air traffic control procedures for radar vectoring require the flight crew to promptly comply with vector instructions and initiate the turn (Nolan, 2004). On this basis, the RMAAT system was designed to control the execution time of the turn by issuing the instruction 'just-in-time' before the desired execution point, taking into account the time required for the controller to communicate the vector and for the flight crew to respond and implement the instruction. However, the actual time required to successfully transmit the message cannot be known *a priori*. For example, if the pilot does not correctly read back the instruction, it must be retransmitted. Similarly, because the standard DSB-AM voice radios are half-duplex, the controller must wait to transmit the instruction if the channel is already occupied by another transmission. The effect of this timing uncertainty on the resulting trajectory accuracy will be examined in Section 6.3.

The system architecture shown in Figure 6-2 was derived around the concept of progressive, just-in-time track angle vectoring. The architecture consists of two primary modules that interact with a list of active resolutions: the Initial Processing Module and the Resolution List Manager.

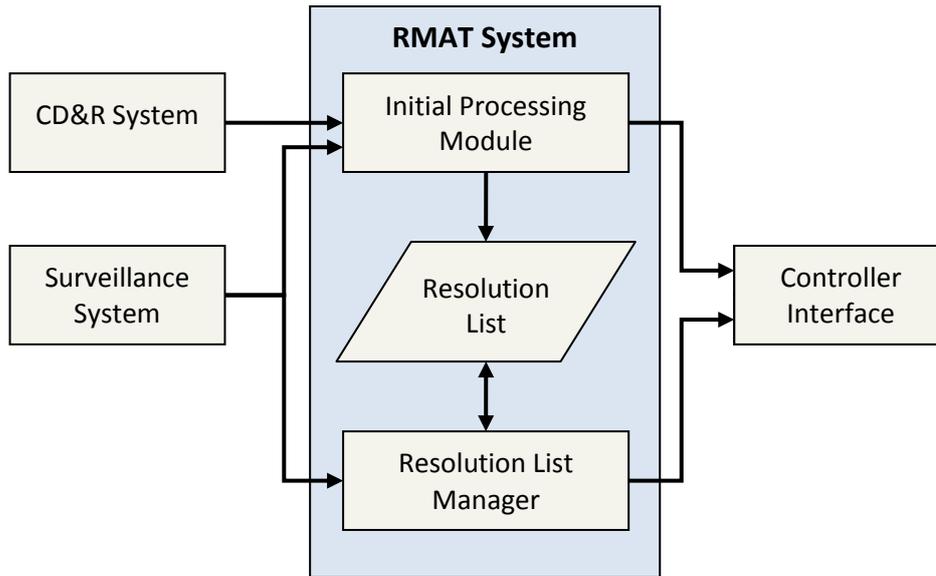


Figure 6-2: RMAT system architecture

The purpose of the Initial Processing Module is to accept resolution manoeuvres from the CD&R system, validate the manoeuvre, and store the resolution in the resolution list. The resolution list contains information on the state of every active resolution, enabling the RMAT system to manage conflict resolutions for multiple aircraft. Each resolution is allocated to a separate node of the resolution list. The Resolution List Manager periodically checks and updates each node in the list, issues alerts and instructions to the controller and removes nodes from the list when aircraft are returned to normal navigation.

The input to the RMAT system comes from the conflict detection and resolution system and the surveillance system. Conflict resolution trajectories generated by the CD&R system are sent to the Initial Processing Module, and the surveillance system provides the information necessary to calculate the vector instructions and to automatically monitor the traffic under RMAT control. For these purposes, the near-term weather forecast and the trajectory intent information (i.e. the original contract trajectory) are considered as part of the surveillance system.

The output of the RMAT system, sent to the controller interface, can contain alerts of upcoming instructions, the text of the instructions, a signal to transmit the instruction, as well as warnings of possible errors.

6.2.1 Initial Processing Module

This module is called every time a new conflict resolution is received from the CD&R system, and performs the five following steps, as indicated by the diagram in Figure 6-3.

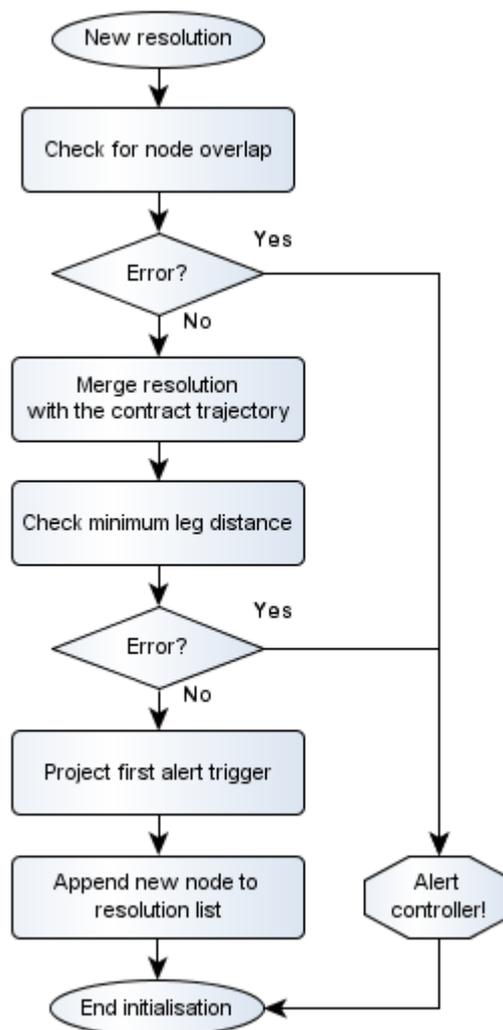


Figure 6-3: Initial Processing Module control flow

Step 1) Check for node overlap: The first check ensures there are no pre-existing nodes for the target aircraft in the resolution list. This might be the case if a revised

conflict resolution is issued by the CD&R system, before the initial resolution has been completed. If an identical aircraft identifier is found in any node of the resolution list, the module sends an alert the controller interface since this event would require the controller's intervention.

Step 2) Merge resolution with the contract trajectory: If no overlap is found, the sequence of waypoints defining the resolution is then merged into the original contract trajectory, using the algorithm outlined in Section 4.2.3. This step is necessary to correctly calculate the turn parameters of the initial and final vectors.

Step 3) Check minimum leg distances: The module must ensure minimum leg distances are met. If the legs of the resolution trajectory are too close together, the aircraft may not have completed the turn when the alert trigger is crossed. This could lead to erroneous calculation of the transmission trigger placement (these triggers are discussed in the following step). The controller is warned if this criterion is not met.

Step 4) Sets first alert trigger: Using the method of progressive, just-in-time track angle vectors, the RMAT system must signal the controller to issue the instruction at the appropriate time before each waypoint. This event can be automatically triggered by computing the ideal *transmission point* along the current flight path such that the aircraft completes the turn on the desired path, taking into account the manoeuvre geometry and communication delays. Practically, however, the controller should be alerted of an upcoming vector *before* the transmission point in order to prepare and plan for the transmission, thus, an *alert point* is set 15 seconds before the transmission point, as shown in Figure 6-4.

Step 5) Append a new node to the resolution list: Finally, a new resolution node is created and appended to the list, initialised to the alert trigger of the first resolution waypoint. Every node contains the aircraft identifier, the merged

trajectory, a flag indicating the active waypoint, the location of the current trigger point and a flag indicating the type of trigger (*alert* or *transmission*).

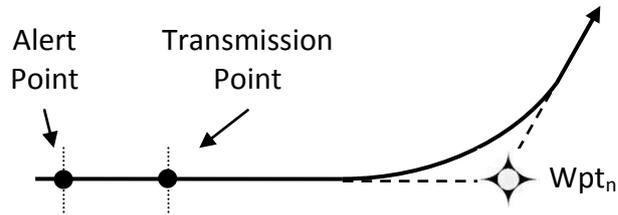


Figure 6-4: Location of alert and transmission trigger points

6.2.2 Resolution List Manager

The purpose of the Resolution List Manager is to automatically monitor the progress and generate flight guidance for all aircraft identified in the resolution list, minimizing the need to manually monitor the progress of vectored aircraft. Every node in the resolution list is periodically checked to determine if the aircraft has crossed the current trigger point. If a trigger is crossed, the module outputs the appropriate cues to the controller. This process is summarised in Figure 6-5.

The first step is to test if the aircraft has crossed the current trigger point, using the wayline method discussed previously in Section 4.2.3. If the trigger point has not been crossed, then the node check ends and the List Manager cycles to the next node, which is tested in the same way.

When the alert trigger is crossed, the transmission point is recalculated. This is because crossing the alert trigger with any cross-track error or track angle error (TAE) will change the geometry of the manoeuvre, and as a result, the location of the ideal transmission point. Track errors may occur for various reasons such as variable wind fields or flight technical error.

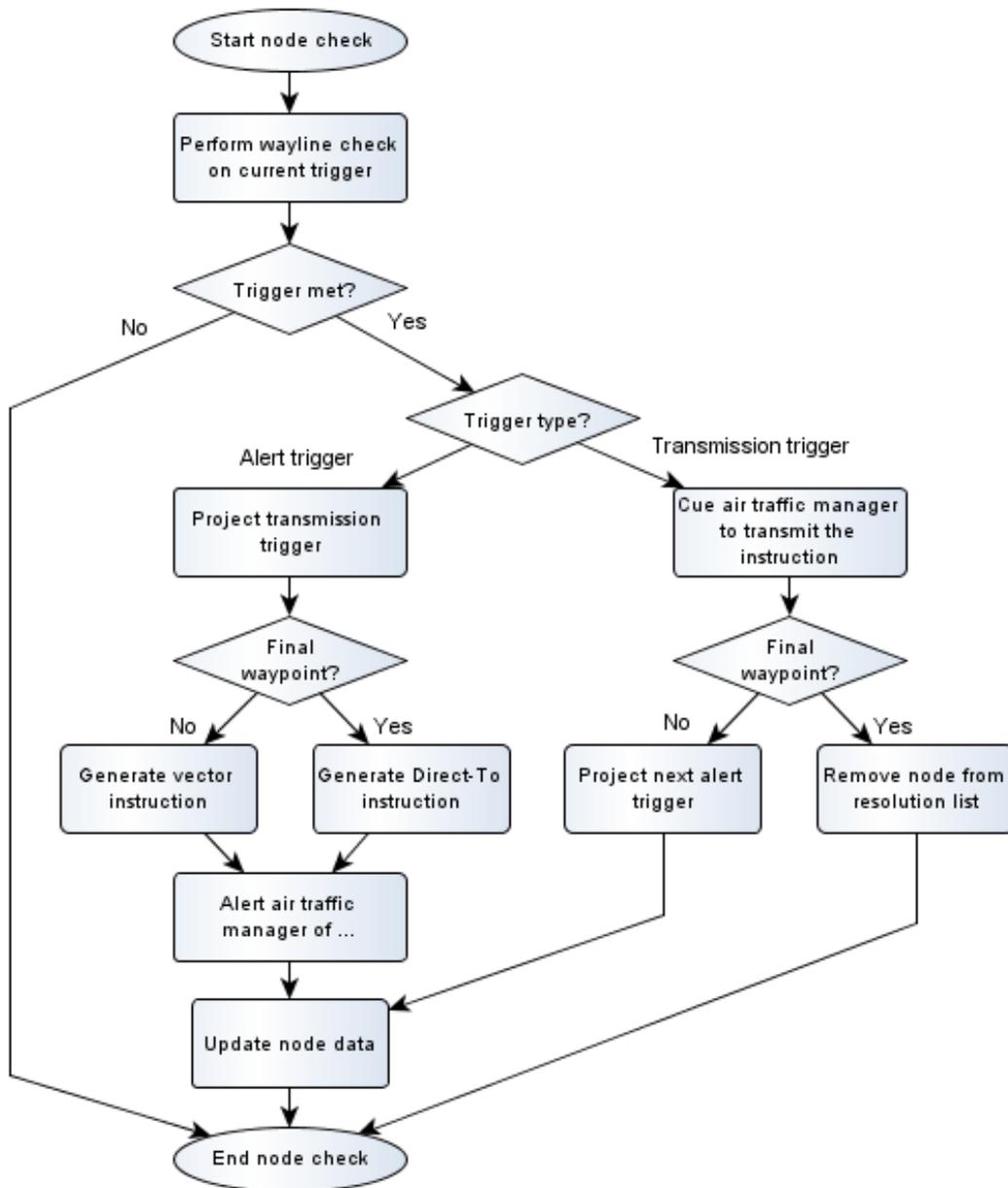


Figure 6-5: Resolution List Manager control flow

After the transmission point is recalculated, the controller can then be alerted of the upcoming vector and provided with the text of the instruction. If the aircraft is on the final waypoint of the resolution route, the text specifies a direct-to instruction and included the phrase “resume own navigation.” Otherwise, a constant track angle is commanded between the current waypoint (φ_1, λ_1) and the next waypoint (φ_2, λ_2) of the resolution route, calculated using the Rhumb Line equation

$$\theta_{RL} = \text{atan2} \left(\Delta\lambda, \text{Ln} \left(\tan \left(\frac{\varphi_2}{2} + \frac{\pi}{4} \right) \right) - \text{Ln} \left(\tan \left(\frac{\varphi_1}{2} + \frac{\pi}{4} \right) \right) \right) \quad (\text{Eq. 6-1})$$

where $\text{atan2}()$ is the four-quadrant arctangent(y/x) function, Ln is the natural log, and $\Delta\lambda$ is the smaller angular difference between λ_1 and λ_2 , less than 180° (USAF, 2001).

The next time the trigger is crossed (approximately 15 seconds after the alert), the controller is signalled to transmit the instruction and the aircraft executes the manoeuvre. If the resolution manoeuvre contains additional waypoints, the active waypoint in the node data is incremented by one, and the next alert trigger is calculated. When there are no remaining waypoints, the node is removed from the resolution list.

6.2.3 Manoeuvre Modelling

This section will describe how the locations of the trigger points are set by modelling the turn manoeuvre, the communications delay, and the planning delay.

6.2.3.1 Placement of the Alert Trigger

The alert trigger point is projected a distance of $d_{Alert Trig}$ from the current resolution waypoint on the reverse course of the flight plan leg. $d_{Alert Trig}$ is composed of the turn anticipation distance (d_{Turn}) as well as the distance needed to account for system delays (d_{Delay}), as shown in Figure 6-6.

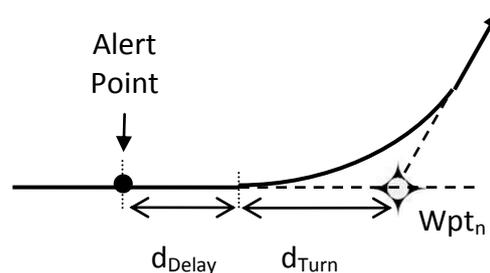


Figure 6-6: Placement of the alert trigger

The RMAT system was based on the same turn modelling assumptions made for the simulator FMS in order to calculate the turn anticipation distance – that is, the assumption of constant bank turns, and that the heading does not change until the

bank is fully established. A nominal bank angle of 15 degrees is assumed, with a bank rate of 3 degrees per second.

In addition to the turn anticipation distance, the alert trigger point must account for system delay, including the review delay and the predicted communication delay. Given the groundspeed and delay estimates, the distance flown during the delay, d_{Delay} , is

$$d_{Delay} = v_{Gnd} * (t_{Alert} + t_{Comm}), \quad (\text{Eq. 6-2})$$

where t_{Alert} is the 15 second buffer between the alert point and the transmission point, and t_{Comm} is the predicted communication delay, including the predicted message duration, predicted pilot response and reaction delay, and any delays caused by readback errors and delayed transmissions. The RMAT system has been designed with a t_{Comm} of 10.85 seconds, which Cardosi found to be the mean transaction time required to correctly communicate a heading vector, including cases that required the controller to repeat or clarify the instruction at least once (1993).

Once d_{Delay} and d_{Turn} have been computed, the distance of the alert trigger from the current waypoint is given by

$$d_{Alert Trig} = d_{Turn} + d_{Delay} \quad (\text{Eq. 6-3})$$

Finally, the coordinates of the alert trigger can be calculated by projecting the alert point a distance of $d_{Alert Trig}$ from the current waypoint along the reverse course of the current route, using Equations 3-8 and 3-9. As discussed previously, when the aircraft crosses the wayline perpendicular to the leg course at this point, the Resolution List Manager calculates the ideal transmission point, and alerts the air traffic manager of an upcoming instruction.

6.2.3.2 Placement of the Transmission Trigger

If the aircraft crosses the alert trigger with any cross-track error (XTE) or track angle error (TAE), the geometry of the manoeuvre is changed. In order for the aircraft to correctly intercept the next path, the turn is recalculated using an *effective*

waypoint rather than the true waypoint when computing the coordinates of the transmission trigger, as illustrated in Figure 6-7.

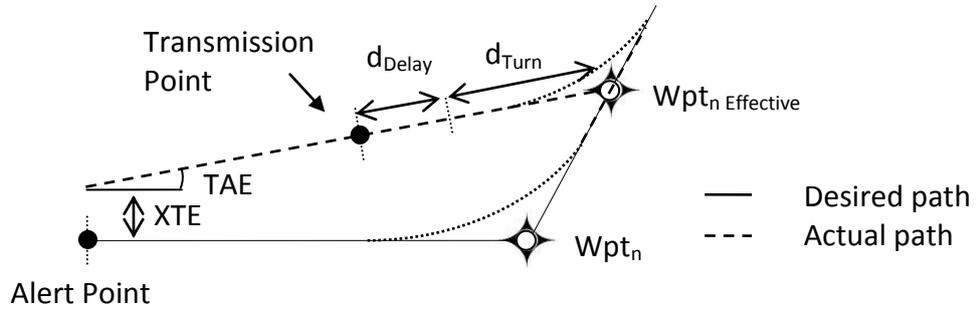


Figure 6-7: Effective waypoint and transmission trigger point geometry

Given the estimated aircraft position and track angle, and the coordinates of waypoint n and waypoint $n+1$, the navigational method of finding the intersection of two radials can then be used to derive the coordinates of waypoint $n_effective$ (Hofmann-Wellenhof, et al., 2003). To find point C of the ABC spherical triangle, the interior angles are given by

$$\Delta\theta_A = \mathit{norm}(\theta_{AC} - \theta_{AB}, \pm\pi) \quad (\text{Eq. 6-4})$$

$$\Delta\theta_B = \mathit{norm}(\theta_{BA} - \theta_{BC}, \pm\pi) \quad (\text{Eq. 6-5})$$

where θ is the relative bearing between two points from Equation 3-5. The Spherical Law of Cosines is used to derive the remaining angle

$$\Delta\theta_C = \cos^{-1} \left(-\cos(\Delta\theta_A) \cos(\Delta\theta_B) + \sin(\Delta\theta_A) \sin(\Delta\theta_B) \cos\left(\frac{d_{AB}}{R_o}\right) \right) \quad (\text{Eq. 6-6})$$

Finally, the distance between points A and C is given by

$$d_{AC} = \mathit{atan2} \left(\sin\left(\frac{d_{AB}}{R_o}\right) \sin(\Delta\theta_A) \sin(\Delta\theta_B), \cos(\Delta\theta_A) + \cos(\Delta\theta_B) \cos(\Delta\theta_C) \right) R_o \quad (\text{Eq. 6-7})$$

Using this method and letting point A be the true waypoint, B be the aircraft position and C be the effective waypoint, the coordinates of the effective waypoint point can then be projected from the true waypoint a distance of d_{AC} at bearing θ_{AC} .

When the effective waypoint has been calculated, the ideal transmission trigger can then be computed in the same way as the alert trigger described in Section 6.2.3.1,

albeit without the alert delay, t_{Alert} , and substituting the true waypoint with the effective waypoint. When the aircraft crosses the wayline at the transmission point, the Resolution List Manager cues the controller to transmit the instruction and calculates the next alert point, if applicable.

6.2.4 Implementation in the Airspace Simulator II

The modular design of the simulator made it relatively straightforward to implement an RMAT prototype. Many of the manoeuvre modelling functions used functions developed for the Navigation and Trajectory Module, simplifying the software design and coding process.

The RMAT system was added between the CD&R system and the Communications Module as shown in Figure 6-8. The Initial Processing Module allows resolutions intended for datalink-equipped aircraft to pass through, but separates and processes resolutions intended for voice-only aircraft using the process described in Section 6.2.1.

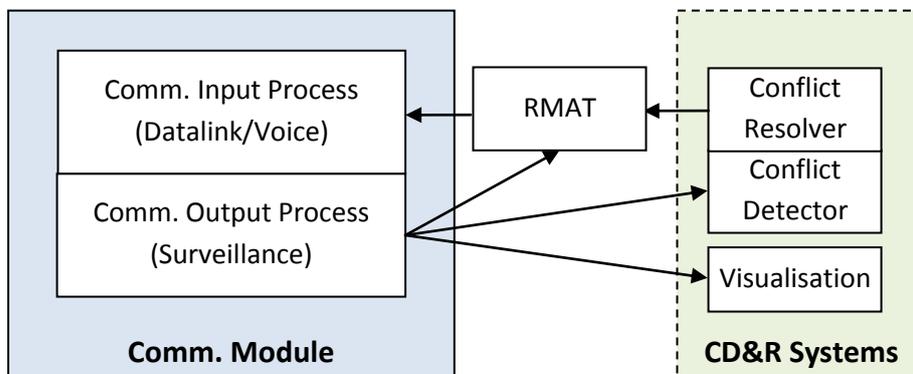


Figure 6-8: RMAT Implementation

The resolution list was implemented as a singly-linked list, where each node in the list contains a pointer to the next node. The Resolution List Manager was called in the outer simulation loop to ensure that the message trigger for every node was checked once per time step. When the transmission triggers were crossed, the RMAT instructions were formatted as verbal track angle instructions with manual termination, according to the message set described in Section 4.2.1, and sent to the

Communications Module where they were given a random latency value and were added to the aircraft message list, as described in Section 4.2. However, the final RMAT instruction for each node was formatted as a direct-to-fix clearance, representing the command to resume normal navigation.

It should be noted that the RMAT system could also have been implemented independent of the simulator using the same network interface as the CD&R system. This would have been useful if RMAT had been developed as a stand-alone program.

6.3 Initial Evaluation of RMAT

The remainder of this chapter provides an evaluation of the RMAT prototype described above. First, the ability of RMAT-guided aircraft to follow conflict resolution manoeuvres is compared to datalink-equipped aircraft in order to verify correct operation and to establish a performance baseline. The RMAT method necessarily relies on trajectory predictions. Errors in the trajectory prediction process can reduce the RMAT performance from the baseline. Therefore, the second aspect of the evaluation focuses on the effect of communication timing uncertainty on conflict resolutions. This evaluation is not intended to fully analyse every aspect of the concept or the prototype system, but to offer initial insights into the capabilities and limitations of this approach to supporting mixed-equipment traffic in the same airspace.

6.3.1 Comparison of RMAT to Datalink

If the logical design and implementation of the RMAT system is correct, the mode of communicating a conflict resolution instruction to an aircraft should not change the outcome of that resolution; the performance of an automated CD&R system should be similar regardless of whether the resolution instruction is sent directly to the aircraft via datalink, or whether the resolution instruction is issued as progressive verbal vectors via the RMAT system.

To test this conjecture, a simulation study was conducted for 1000 pairwise encounter scenarios created using the scenario generator described in Chapter 5. The

scenarios were applied to three simulation runs. In the first run, the CD&R services were not used and the aircraft were uncontrolled. In the second run, the centralised CD&R system described in Section 4.5.4 was used, and both aircraft were datalink-equipped. In the final run, the centralised CD&R system was used with the RMAT system. The closest point of approach (CPA) was recorded for each scenario to determine if RMAT significantly affected the result of the CD&R system on the conflicts.

The Sheehan traffic model described in Section 5.2.1 was used and encounter angles were distributed uniformly so that conflicts would occur with a variety of airspeeds and geometries. The scenario altitude was FL320. Time to first loss of separation was set to 5 minutes, and the post-conflict distance, D_3 , was 15 nmi. A 5 nmi conflict separation minimum was set for the CD&R system and for the scenario generator. Winds and flight technical error were omitted.

The results of the simulations are shown in the following two figures. Figure 6-9 shows a histogram of the measured CPA for each of the three runs, and Figure 6-10 compares the means and standard deviations of the CPA.

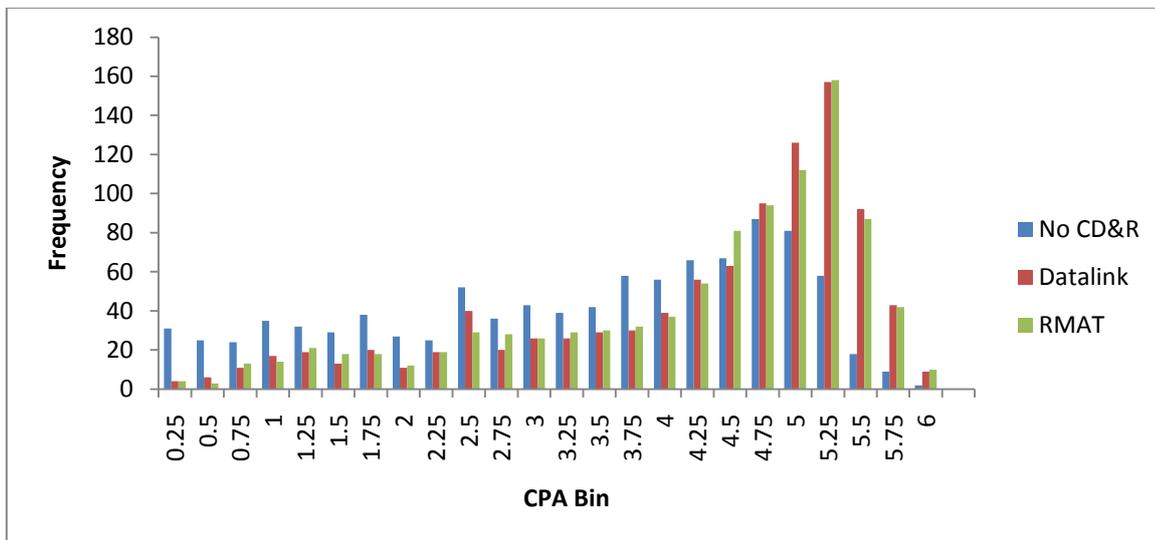


Figure 6-9: Histogram of the closest points of approach

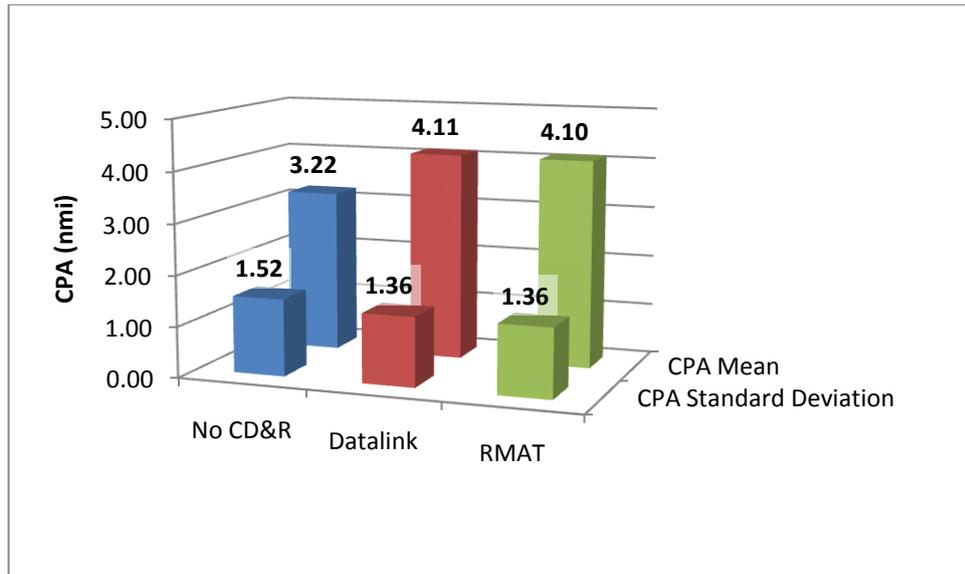


Figure 6-10: Summary of CPA statistics by run

First, it should be noted that although the use of the CD&R system increased the average aircraft separation by approximately 1 nmi, the resolver did not consistently satisfy the 5 nmi separation minimum. The reason for this anomaly was that the resolver was under development; in particular, the weights of the genetic algorithm’s cost factors had not been optimized.

Despite the limitation of the CD&R system, it can be seen that the RMAT did not significantly alter the CPA compared to datalink-equipped aircraft. On average, the use of RMAT reduced the separation by only 0.01 nmi. Furthermore, a linear regression was conducted for the scenario encounter angle and airspeed difference, shown in Table 6-1, and did not reveal any significant trend differences between the datalink and the RMAT simulation runs; that is, the error between RMAT and datalink was not a function of encounter angle or speed difference. Thus, under ideal conditions RMAT-guided aircraft closely match the trajectories of datalink-equipped aircraft, confirming the logical design and implementation of the RMAT prototype.

		x	
		Enc. Angle (deg)	Speed Diff. (kts)
y (nmi)	Datalink CPA	$y = 0.0038x + 3.7645$	$y = 0.0034x + 4.0198$
	RMAT CPA	$y = 0.0037x + 3.7699$	$y = 0.0034x + 4.0092$

Table 6-1: Linear regression results on datalink and RMAT CPA

6.3.2 Effects of Communication Timing Uncertainty on CPA

However, under less than ideal conditions there are several potential sources of error that could reduce the system’s performance, including:

- Communication timing uncertainty,
- Wind direction or magnitude forecast error,
- Cross track or along track position estimation error,
- Groundspeed or track angle estimation error,
- Bank angle or bank rate prediction error.

For example, any difference between the predicted and the actual bank angle and bank rate will alter the turn radius, affecting the location of the trigger points. This section focuses on one specific error source – communications timing uncertainty – and examines the effect of the uncertainty on the resulting closest point of approach.

As discussed in Section 6.2.3.1., the RMAT system was designed with an expected verbal transaction time of 10.85 seconds, based on a study by Cardosi into the time required to correctly communicate a heading vector (1993). Thus, if the actual transaction time is greater than 10.85 seconds when using the RMAT system, then the aircraft will turn after the desired point as illustrated in Figure 6-11. Similarly, if the actual transaction time is less than 10.85 seconds, then the aircraft will turn before the desired point.

To test the effect of the flight path variation on the CD&R system, a study was conducted for 250 pairwise encounter scenarios created using the scenario generator. The scenarios were applied to seven simulations runs. In the first run, the CD&R services were not used and the aircraft were uncontrolled. In the second run, the

centralised CD&R system was used and both aircraft were datalink-equipped. In the remaining five runs, the centralised CD&R system was used with the RMAT system. Every RMAT message was assigned a random radio-telephony latency drawn from the Lognormal distribution specified in the simulator configuration file, as discussed in Section 4.3.2. The means and standard deviations of the specified latency distributions are listed in Table 6-2; the mean was incremented by 10 seconds for each run. The final column shows the resulting average error between the RMAT-predicted communication delay and the actual communication delay.

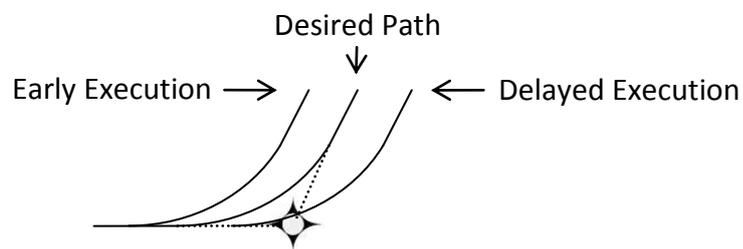


Figure 6-11: Effect of communication timing errors on the RMAT flight path

Run Name	Communication Delay (sec)		Mean Error (sec)
	Mean	Standard Deviation	
No CD&R	N/A: CD&R Not Used		
DL	N/A: Datalink Run		
10	10.85	5.91	0
20	20.85	5.91	10
30	30.85	5.91	20
40	40.85	5.91	30
50	50.85	5.91	40

Table 6-2: Radio-telephony latency settings and the resulting average error

The CPA was recorded for each scenario to determine if RMAT communication timing error significantly affected the outcome of the conflict resolutions. The resulting mean and standard deviation of the CPA for each run is shown in Figure 6-12. Surprisingly, it appears that increasing the average timing error did not have an appreciable effect.

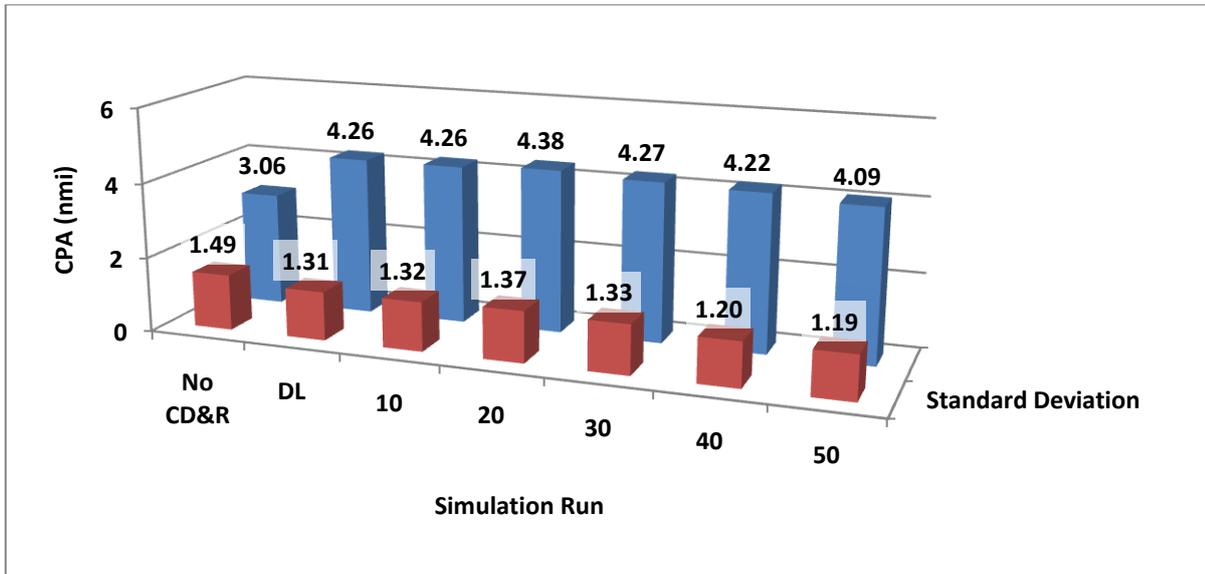


Figure 6-12: CPA statistics by simulation run

However, by taking the difference of the RMAT CPAs against the datalink CPAs for each scenario, it can be seen that the CPA differences are symmetric around zero. This is plotted in Figure 6-13 and explains the unexpected statistics of Figure 6-12. Figure 6-14 shows the resulting mean and standard deviation of the CPA for each run of the absolute value of the CPA differences. It can be seen that both the mean and standard deviation increase with the mean RMAT communication timing error. When the mean communications delay error was 40 seconds, the CPA varied on average by 0.52 nmi from datalink-equipped aircraft for identical resolution manoeuvres.

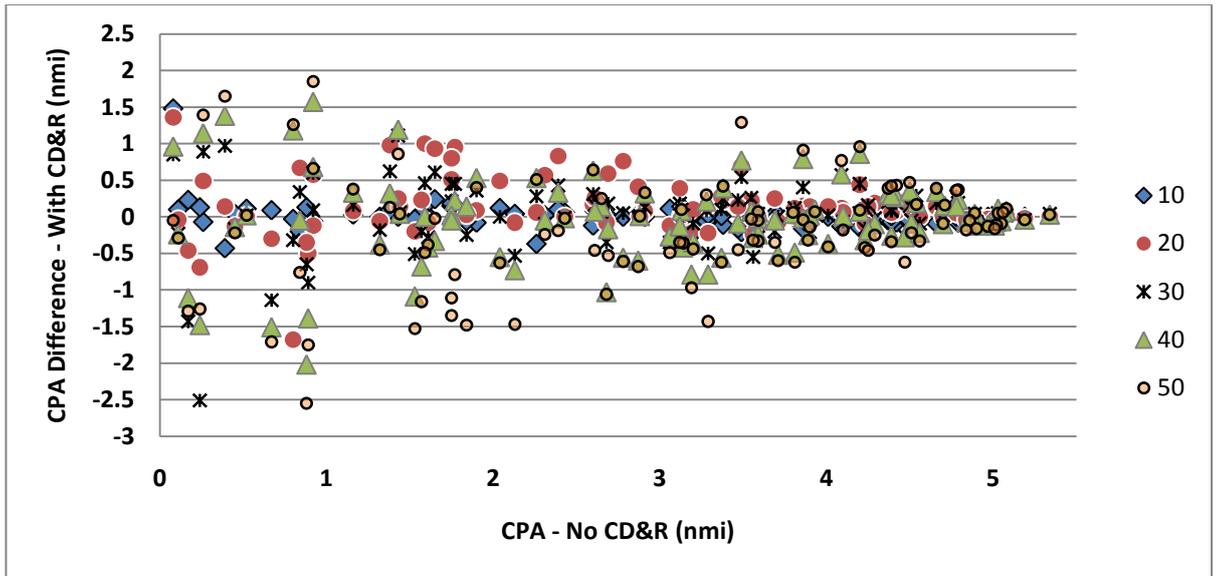


Figure 6-13: Difference of RMAT and Datalink CPAs, plotted by scenario CPA

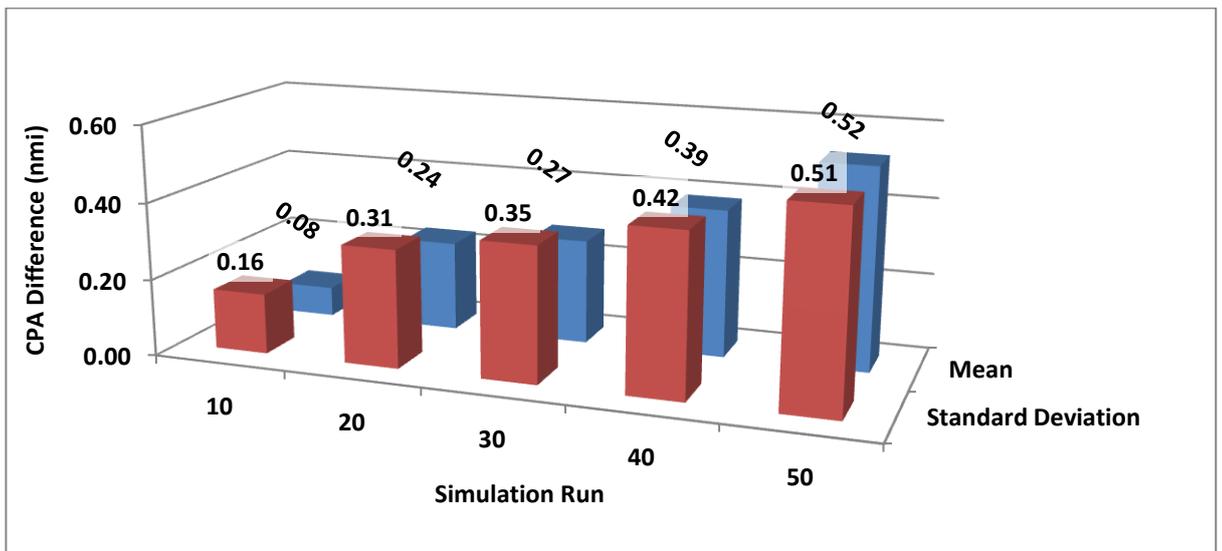


Figure 6-14: The absolute value of CPA difference statistics by simulation run

The results of the test reveal several significant points. First, communications timing uncertainty had only a small effect on CPA when the mean of the communication delay distribution correctly matched the RMAT expected value; during that run, the CPA varied by only 0.08 nmi from the datalink-equipped aircraft. As the mean of the communication delay distribution increased from the RMAT expected value, the CPA difference grew. To minimise this effect, RMAT may need to be ‘tuned’ to the communication delay distribution of the local airspace.

Secondly, for the CD&R system used in this test, delaying the turn did not necessarily increase the severity of the conflict. In nearly half the cases, communication delays actually increased the closest point of approach and therefore reduced the severity of the conflict. This may be due to the relative cost weightings used by the CD&R genetic algorithm to select resolution manoeuvres, such as distance added to the flight plan verse CPA.

Finally, it must be noted that the trajectory error due to communication delays is a function of the turn angle. Figure 6-15 illustrates a scenario where the aircraft has over flown the turn point by a distance of $D_{Trn ATE}$. The cross track error (XTE) resulting from $D_{Trn ATE}$ can then be solved by Napier's rules for right spherical triangles:

$$XTE = \sin^{-1}(\sin(\alpha) * \sin(D_{Trn ATE})) \quad (\text{Eq. 6-8})$$

The cross track error can range between 0 and $D_{Trn ATE}$, for α between 0 and 90 degrees. Thus, the CPA difference would probably be more pronounced for conflict resolutions that required larger turn angles.

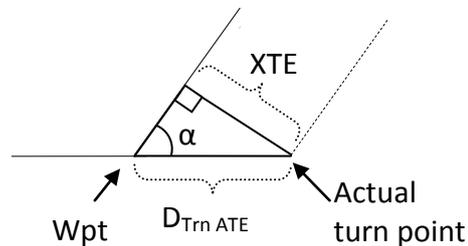


Figure 6-15: Illustration of cross track error resulting from along track error in the turn point location

6.4 Chapter Summary

The study described in this chapter has confirmed that the simulator meets the goal of a simple, flexible, and accessible fast-time simulator suited for exploratory separation management research. The argument was made of the need for improved automation support to support traffic not equipped with an FMS-coupled datalink in a

more highly automated ATM system. The Resolution Monitoring and Advising Tool was presented as a novel concept of providing semi-automatic lateral flight guidance to non-datalink traffic using progressive track angle vectors.

The design of an RMAT system was described, focusing on the Initial Processing Module to accept conflict resolutions from the CD&R system and the Resolution List Manager to periodically check the resolution list and signal the controller to transmit the vector at the ideal time. The maneuver modeling section then discussed the calculation of the alert and transmission trigger points, followed by a description of how RMAT was implemented in the simulator software.

The first set of simulation experiments showed that the closest point of approach did not significantly vary when using RMAT as opposed to datalink under ideal conditions, indicating that RMAT was correctly designed and implemented. The mean CPA difference was less than 0.01 nmi. In the second set of simulation experiments, communication timing uncertainty was added with an increasing mean delay. The addition of the timing variance changed the mean CPA difference to only 0.08 nmi. However, increasing the mean delay from the RMAT expected value caused both the mean and standard deviation of the CPA difference to increase.

Chapter 7

Discussion and Conclusion

The main aim of the research presented in this thesis is to redevelop the Airspace Simulator to provide a more useful platform to design and verify new methods, algorithms, and strategies for separation management in future airspace. In this chapter, Section 7.1 will further discuss the major achievements and engineering trade-offs made over the course of the research, and will highlight the benefits and limitations of the software design. Section 7.2 then concludes with a review of the success in fulfilling the requirements and presents ideas for future work.

7.1 Discussion

There were three overarching considerations during the research and development of the simulator: execution speed, fidelity, and functionality.

7.1.1 Execution Speed

As discussed previously, execution speed is a fundamental performance parameter for fast-time simulators, and was an important design consideration for the algorithms and data structures developed for the simulator.

Section 5.4 of this thesis demonstrated that the speed of the simulator was primarily a function of the flight durations and the number of simulated aircraft (when the scenario generator is used, the flight plan file writing operations also significantly contribute to the execution time). The relationship between execution speed, flight durations, and the number of aircraft is due to the nesting of the outer and inner simulation loops of the basic logical design; each active aircraft is updated each time step as described in Section 2.5. One technique of minimising the effect of the nested loop is dynamically resizing the time-steps, which enables faster execution by reducing the total number of discrete steps (Sokolowski & Banks, 2009). However, this option

was not used for two reasons. First, maintaining a steady clock was necessary for consistent surveillance so that conflict detectors and traffic visualization tools could correctly monitor the traffic. Secondly, conflict resolution inputs are asynchronous and unexpected – a variable step size could introduce undesirable latencies to the aircraft response.

An additional challenge during the project was balancing speed optimization with the clarity and simplicity of the source code. For example, the APFDS was implemented by the functional decomposition shown in Figure 7-1. Thus, seven separate procedure calls are required during execution. Alternatively, a more optimised approach would be to write the entire APFDS functionality into a single C function, requiring only one procedure call. Profiling the APFDS code (as implemented) showed that procedure call overhead accounted for 6.5% of the function execution time. The disadvantage of the more optimised approach is that a large block of inline code can be hard to understand – modifying, replacing, or testing code becomes more complex, and learning the operation of the APFDS becomes more difficult.

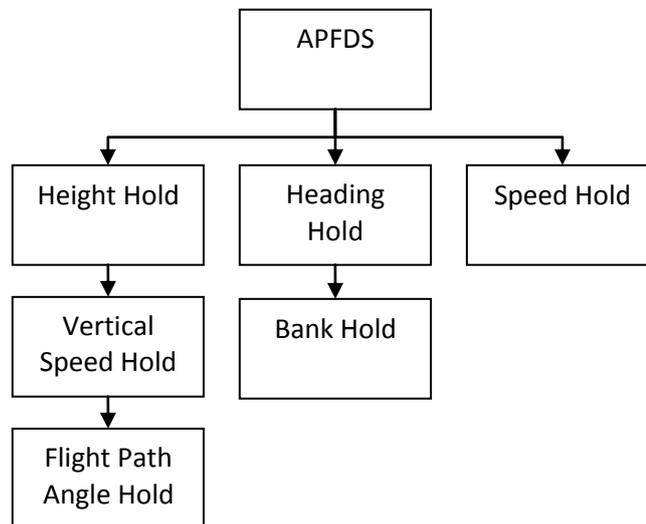


Figure 7-1: Functional decomposition of the APFDS

Given the performance of modern computer processors and C compilers, the time-savings from a more understandable implementation will likely be greater over the life of a research project than the time-savings due to greater execution speeds from extensive code optimisation. That is, using a simulator that is easy to learn,

understand, and modify may result in a shorter project timeline than using a simulator that can compute experiments very fast but requires a steep learning curve. The exception is if the simulator is dedicated to a single, repeated purpose. In that case, over the course of multiple simulations the time-savings from speed may be greater than the savings from the learning curve.

Despite the limitations of the nested loop, and the balance of simplicity and optimisation, the simulator was demonstrated to run over 900 times faster than real-time for 300 simultaneous aircraft. This raises the point that the speed of the CD&R system is an equally important design issue. The computational complexity of identifying and resolving conflicts can be greater than that of simulating traffic trajectories – in fact, many CD&R systems incorporate internal trajectory predictions and are effectively small-scale simulators. Thus, if run in parallel, the limiting factor to the speed of an experiment may be the CD&R system, not the simulator. This was the case when the genetic resolver was connected to the simulator, necessitating the synchronisation methods discussed in Section 2.6.

7.1.2 Fidelity

The level of fidelity was another critical design consideration for the simulator. As discussed in Chapter 2, increasing the realism typically reduces speed and increases complexity.

The challenge of balancing speed and fidelity was most evident during the development of the Trajectory and Navigation Module. The module needed to model the behaviour of modern commercial aircraft with sufficient realism to be useful for separation management research, while maintaining fast execution. As discussed in Section 3.3, the tabulated BADA database was selected as the basis for the performance model because of its wide coverage of commercial aircraft and also because it significantly increases the speed of the simulation by reducing the computational complexity of generating aircraft trajectories. Additionally, the tabulated data simplifies the simulation by removing the need to model detailed flight data such as airline procedures and speed schedules. However, this implementation

comes at the cost of reduced fidelity in comparison to more complex performance databases, for example point-mass or six degree-of-freedom models. Furthermore, the use of tabulated data limits accurate speed control and precise vertical navigation due to the interdependence of thrust, airspeed, and vertical speed, as described in Section 3.4.2.

The balance between fidelity and complexity can be seen with the radio-telephony communication model. Unlike the air-ground communications model by Monticone, et al, the Airspace Simulator-II model does not implement the full set of pilot and ATC voice messages, such as transponder settings and communications handoffs (2005). Rather, it models ATC instructions, and uses the latency model to take into account errors and frequency congestion. Thus, the effects of non-directive messages are captured in the model, even though the actual messages are not directly simulated. The simplification eliminates the need for extensive input data such as airspace sector boundaries and additional modules, such as intelligent message generators. Furthermore, the approach eliminates the need to simulate large amounts of traffic merely to produce a desired level of frequency congestion. For example, when testing the RMAT system in Section 6.3, the random latency values included the effect of high frequency congestion, even though only two aircraft were simulated.

7.1.3 Functionality

The simulator was also made more versatile. New functionality was added including navigation, surveillance, and communications error modelling, enabling the simulator to be used to investigate new separation management systems and concepts. The simulator was specifically designed so that many parameters and experimental variables could be easily controlled, such as latency parameters, FTE parameters and the time-step size. The software was organised into functional modules in order simplify the expansion or customisation of simulator functionality.

The synchronisation and network code was added to allow the simulator to be connected to other systems, enabling CD&R tools to be interfaced to the simulator without the need for extensive modification or translation to C. Additionally,

networking also allows the simulator to be used for other ATM applications; for example, the use of the simulator to augment the traffic of other simulation systems.

Many aspects of the research presented in this thesis contribute to the wider ATM research community, apart from the simulator. The scenario generator designed for this thesis provides an efficient way of creating a large variety of pair-wise conflicts while also giving the user control over many scenario variables. The approach reduces artefacts such as scenarios with non-conflicting aircraft and helps ensure efficient use of simulation resources. The same method can be applied by researchers using other airspace simulators to provide more rigour and confidence in the scenario coverage of Monte Carlo experiments.

Similarly, the novel flight plan merging algorithm described in Section 4.2.3 also applies to Unmanned Aerial Systems (UAS). Flight plans for UAS missions may involve many waypoints, and as a result it would be an inefficient use of bandwidth to transmit an entire modified flight plan when only several waypoints are altered. The flight plan merging algorithm could be a solution to minimise data loads and consistently construct the desired flight plan.

7.2 Conclusion

As a result of the limitations of the original design, the Airspace Simulator was significantly redesigned to enable the simulator to be used for the specific application of CD&R research.

The simulator remains accessible to most researchers, in keeping with the purpose of the original Airspace Simulator. The BADA database is freely available to academic researchers and the GCC compiler is also open source and can run on several platforms. Similarly, the modular design makes it relatively simple to tailor the simulator to specific research applications, and the translation to the more familiar ANSI C language makes it easier to work with. Configuration and setup was simplified from the original simulator, making the Airspace Simulator-II easier to learn and use.

The fast trajectory modelling allows many cases to be studied quickly, and makes it feasible to use the simulator for large-scale, stochastic experiments.

7.2.1 Satisfaction of Requirements

The 14 functional and performance requirements set for this project were considered necessary for the software to be a useful tool to investigate new separation management methods. Over the course of the thesis, each requirement was shown to be satisfied, as follows:

Req. 1, Capable of fast-time simulation of up to 300 aircraft simultaneously: In order to be capable of conducting stochastic simulations of new CD&R algorithms and concepts over a range of conditions and traffic scenarios, the simulator needed to be able to compute multiple, simultaneous aircraft in fast-time. Requirements 1 and 2 represent twice the projected 2020 peak traffic level at the busiest airport in Europe. Achievement of Requirement 1 was demonstrated in Section 5.4; simulations of short distance and long distance flights show the Airspace Simulator-II ran over 900 times faster than real-time for 300 simultaneous aircraft and over 30 times faster than real-time for 2500 simultaneous aircraft.

Req. 2, Capable of simulating of up to 4000 aircraft in total: As with Requirement 1, this requirement was achieved with a large margin. Section 5.3.3 showed that 4000 aircraft (representing a total of 225 flight hours) could be computed in 37.3 seconds on a laptop computer. Aircraft can be simulated with the flight performance of any of the 318 aircraft types in the BADA database.

Req. 3, Simulates waypoint-to-waypoint flight guidance within 1 nmi: Developing a navigation and flight control system that was modular, flexible, and accurate formed a significant portion of this project, as described in Chapter 3. In order to accurately model Trajectory Based Operations, as called for by SESAR and NextGen, FMS flight guidance was needed within 1 nmi. Section 3.8.3.1 demonstrates that the FMS and Autopilot/Flight Director were able to maintain an average cross track error of less than 0.051 nmi for 1000 nmi waypoint-defined routes, including the presence of

winds. This average error represents a 90% improvement over the average cross track error reported of the original Airspace Simulator.

Req. 4, Capable of connecting with external CD&R software: A central requirement of the simulator was the capability of integrating both centralised and decentralised CD&R systems with air traffic models. The core simulator functions were reorganised to better support the integration of new CD&R tools into the simulator and minimise modifications. By incorporating both UDP and TCP/IP network interfaces and by maintaining time synchronization, the simulator was successfully integrated with a third-party, closed-source CD&R tool, as discussed in Section 4.5.4. The network capability enabled the CD&R tool to be used to evaluate the RMAT system described in Chapter 6.

Req. 5, Simulates broadcast of traffic state and trajectory intent: This requirement was necessary to provide the information required by conflict detection routines; many conflict detectors use a combination of traffic state and traffic intent information. The simulator models ADS-B-type surveillance systems by broadcasting aircraft state and velocity data and periodically down-linking the flight plan. This requirement was shown to be met by connecting the simulator to the TViz visualisation tool described in Section 2.7.2 and confirms the traffic progression. In addition, the simulator was integrated with the CD&R tool in Section 4.5.4, which used the traffic data provided by the simulator to predict conflicts and compute resolutions.

Req. 6, Allows input and execution of conflict resolutions: The simulator must be able to receive and implement the most common conflict resolution manoeuvres issued by resolvers. The ability to *monitor* for conflict resolution messages was added to the logical design, as discussed in Chapter 2. The ability to *accept* resolutions was confirmed by the successful integration with the CD&R tool in Section 4.5.4. The ability to correctly *implement* resolution manoeuvres was illustrated in Section 3.8.3.2.

Req. 7, Simulates datalink and voice communication: Datalink is expected to be the primary mode of communications for future ATM systems, so simulation of datalink

exchanges was considered a basic requirement. However, voice will continue to be used to control unequipped traffic and for off-nominal events. A voice communication model was necessary to investigate the effects of datalink failures, mixed-mode operations and new voice-based automation tools. The communications module was rewritten to include both communication modes, expanded message sets and communication latencies. A communications message queue was added to store conflict resolution messages and ATC instructions until they could be delivered to the recipient aircraft. The ability to simulate communications between the CD&R system and aircraft was demonstrated in Section 4.5.4, and during the RMAT testing in Section 6.3 over 1250 datalink and voice messages were exchanged between the CD&R system, the RMAT system and the aircraft.

Req. 8, Simulates navigation errors and uncertainties: Separation management is essential to the safety of flight, so any conflict detection and resolution process must be shown to be robust against errors and uncertainties in communication, navigation and surveillance systems. This requirement was met by adding navigational wander to the aircraft flight paths through the FTE model, as demonstrated in Section 3.8.4. Additionally, the Aircraft Array data structure was redesigned to support localised information states, allowing the navigation and surveillance functions to use distinct estimations of position, velocity and the winds aloft. Information states enable the simulator to be used to gain a better understanding of the effects of localised, noisy data.

Req. 9, Simulates surveillance errors and uncertainties: A simple probability of reception model was described in Section 4.4 and evaluated in Section 4.5.1. The stochastic failure behaviour of ADS-B broadcasts was modelled by applying a constant, uniform probability of success for surveillance broadcasts. In addition, a placeholder function was included to enable user-defined noise to be added to the true aircraft state data when it is copied to the communications data structure, in order to simulate the errors due to localised information states.

Req. 10, Simulates communications errors and latencies: Latency models were designed for both the datalink and voice communication modes, representing the communications delay between a conflict resolution or ATC instruction being transmitted and executed by the flight crew. The voice communication transaction times also include delays due to repeated messages and blocked transmissions. However, in order to provide flexibility in the model and because latency statistics are known to vary by airspace, the latency parameters can be defined in the simulator configuration file. The latency models were verified in Section 4.5.2.

Req. 11, Simulates mixed-mode traffic: Because the study of separation management in mixed-mode operations is ongoing, it was important for the simulator to be capable of specifying and modelling different communication modes (datalink or voice) and navigational accuracies for different aircraft, and be able to designate either self-separation or centralised-separation responsibility for each aircraft. Simple mode flags were added to the Master Array, allowing the simulator to make distinctions between the flight control mode (autopilot, or flight director), communication mode (ideal, datalink, or radio-telephony), and separation mode (centralized, self-separation, or uncontrolled). This capability was discussed in Section 2.3.1 and confirmed in the flight tests of Section 5.3. The modes proved particularly useful during the RMAT simulation experiments described in Section 6.3.

Req. 12, Allows automatic generation of pseudo-random traffic scenarios: To better facilitate large, stochastic experiments, the simulator needed the ability to automatically generate pseudo-random traffic scenarios. Chapter 5 addressed the design and validation of a pair-wise scenario generator that can automatically produce flight plans and traffic assignments resulting in a large variety of horizontal two-aircraft encounters.

Req. 13, Allows manual setup of traffic scenarios: During the development of the simulator, it became apparent that in addition to automatically generating scenarios, it is also necessary to prepare arbitrary traffic scenarios manually. This feature is

described in Section 2.3.1 and was used for all of the simulation tests of Chapters 3 and 4.

Req. 14, Capable of running on a single PC: The final requirement was that the simulator must be capable of running on a single desktop computer, which was a constraint on the solution but helped ensure the simulator would be easily accessible to researchers. All the simulations conducted for this thesis used either a single desktop or laptop PC.

7.2.2 Suggestions for Further Work on the Airspace Simulator – II

The simulator presented in this thesis is complete, and as discussed above, it meets all of the requirements that were set for this research project; however, there are three main areas where further work could expand and improve the usefulness of the simulator as a research platform.

7.2.2.1 Expanding the FMS Functionality

The FMS could be improved with the addition of waypoint time-of-arrival control and precise vertical navigation (such as Top-of-Descent calculation). As discussed in Section 3.4, the FMS is capable of these functions, but is limited by the tabulated BADA performance data. Converting to the full BADA model instead of the tabulated data would provide these functions as well as improve the performance fidelity. However, the additional computational load would likely have an impact on the runtime performance, it would be necessary to allow the user to optionally select the performance modelling mode appropriate to their need.

7.2.2.2 Expanding the Scenario Generation Method

Due to project time constraints, the scenario generator only produces lateral scenarios, which accounts for approximately 75% of encounters. However, the same basic method discussed in Chapter 5 could be expanded to include vertical traffic scenarios. The challenge is that this would require the aircraft climb performance to be calculated from the BADA data during scenario generation, effectively requiring a small-scale simulation before the actual simulation. Several approaches could be

useful in order to minimise the additional computational load caused by doubling the simulation, such as using a larger time step to reduce the number of discrete movements, or by assuming small change over the climb/descent and simply averaging several points. These options would need to be investigated to evaluate the time-savings in comparison with correctly generating vertical encounters.

A further improvement to the scenario generation would be complementing the pair-wise generator presented in this thesis with the addition of a random traffic pattern scenario generator or a scenario generator based on common airport pairs. As discussed in Section 5.1, detecting potential conflicts and searching for conflict free routes is more complex when other aircraft are in proximity, and as a result, pair-wise encounters do not fully stress CD&R systems. A number of approaches are available to maintain a desired traffic level in a given airspace region with pseudo-random flight plans, which could be adapted for use in further developments of the Airspace Simulator-II (Feigh, 2003; Singor, et al, 2004).

7.2.2.3 Incorporating a Native CD&R System

A final suggestion is to implement a CD&R system within the simulator. Although this was beyond the scope of the thesis, a native CD&R system would be a useful addition to the functionality provided in the simulator. Some users may elect to use their own CD&R systems, but even a relatively simple native CD&R system could provide a ready-to-go option for users, or could be used as a baseline for comparison. The simulator architecture was designed to readily support extensions such as one or more native CD&R systems.

7.2.3 Suggestions for Further Work on Automated Vector Navigation

The RMAT discussion presented in Chapter 6 only covers an initial analysis of automated track angle vector navigation. Additional work remains to determine the full potential and limitations of the RMAT approach. For example, RMAT is effectively an open loop control system; that is, instructions are issued to aircraft only once per vector. The aircraft do not know the true resolution trajectory and so cannot correct

errors, unlike normal trajectory based operations where the full resolution can be communicated via datalink and implemented in a closed-loop FMS. As a result, there would need to be more analysis of the navigational accuracy that could be expected from an RMAT system in the presence of errors, examining both the main effects and interaction effects of error variables, particularly weather forecast errors and surveillance errors. The simulator could be used as a platform for these tests.

Similarly, the effect of the additional communication load needs to be examined in more detail. For two aircraft in conflict, both messages may need to be delivered in short succession, possibly causing an additional delay to one of the messages. As a result, the message latencies would no longer be independent, identical random variables, as was assumed in the initial analysis.

Glossary

ACES	Airspace Concept Evaluation System
ADS-B	Automatic Dependent Surveillance – Broadcast
ANSP	Air Navigation Service Provider
APFDS	Auto Pilot/ Flight Director System
ASA	Automated Support to ATS (Air Traffic Services)
ASAS	Airborne Separation Assurance Systems
ASSTAR	Advanced Safe Separation Technologies and Algorithms
ATC	Air Traffic Control
ATE	Along Track Error
ATM	Air Traffic Management
BADA	Base of Aircraft Data
CATS	Complete Air Traffic Simulator
CD&R	Conflict Detection and Resolution
CDTI	Cockpit Display of Traffic Information
CFMU	Central Flow Management Unit
CORA	Conflict Resolution Assistant
CPDLC	Controller-Pilot Data Link Communication
CTAS	Center-TRACON Automation System
DAG-TM	Distributed Air/Ground – Trajectory Management
DL	Data Link
DSB-AM	Double Side Band – Amplitude Modulation
DST	Decision Support Tool
ECC	EUROCONTROL Experimental Centre
EDA	En Route Descent Advisor
ESCAPE	EUROCONTROL Simulation Capability and Platform for Experimentation)
FA	Area Forecast
FACET	Future ATM Concepts Evaluation Tool
FCS	Flight Control System
FMS	Flight Management System
FREER	Free-Route Experimental Encounter Resolution
FTE	Flight Technical Error
HITL	Human-In-The-Loop (or Hardware-In-The-Loop)
IFR	Instrument Flight Rules
LNAV	Lateral Navigation
MTCD	Medium-Term Conflict Detection
NARSIM	NLR ATC Research Simulator
NLR	National Aerospace Laboratory of the Netherlands
NPN	NOAA Profiler Network
PARR	Problem Analysis, Resolution and Ranking
PAT	PHARE Advanced Tools
PBD	Place, Bearing, Distance (or Point, Bearing, Distance)
RMAT	Resolution Monitoring and Advising Tool
RNAV	Area Navigation
RNP	Required Navigation Performance

Glossary

RSP	Required Surveillance Performance
RT	Radio-Telephone
SM	Separation Management
SSR	Secondary Surveillance Radar
TAAM	Total Airspace and Airport Modeller
TBO	Trajectory Based Operations
TCAS	Traffic Collision Avoidance System
TCP	Trajectory Change Point
TIS-B	Traffic Information System – Broadcast
URET	User Request Evaluation Tool
XTE	Cross Track Error

Appendix A

Simulator Configuration File

The simulator configuration file is a plain text file named '*ATMSim_Config.txt*,' read during simulator initiation in order to configure simulation parameters and to define the traffic scenario. The file is separated into the following sections: time, system, global scenario parameters, communications, and air traffic. Section titles are written in all capitals and used to parse the file. The order of data within a section is significant. Data is separated from plain text by an equal sign and is followed by a semicolon. The exception is the list of aircraft assignments, which follows the format: BADA type, flight plan name, aircraft start time, control mode (ideal – 'ID', autopilot – 'AP', or piloted – 'FD'), communication mode (ideal – 'ID', datalink – 'DL', or radio-telephone – 'RT'), and separation mode (centralised – 'CENT', ASAS – 'ASAS', or uncontrolled – 'UNCO'), ended with a new line. Figure A-1 shows an example configuration file.

```

Airspace Simulator-II Initial Conditions
Updated: 15/Sept/2010
-----
TIME

Fast time      = Y;          (Yes/No)
Time step      = 1.00;       (sec per step)
Time limit     = 500000;    (sec)
-----

SYSTEM

Data resolution:      = 1;          (time steps between recording data)

Enable TCP/IP network? = N;          (Yes/No)
  Port                = 52000;
  IP                  = 127.0.0.1;

Use traffic visualisation? = N;          (Yes/No)
  Port                = 52000;
  IP                  = 127.0.0.1;

Use scripted ATC?     = N;          (Yes/No)
  Script address      = folder\filename.atc;
Random number seed    = 2;          (if 0, fully random)
-----

GLOBAL SCENARIO PARAMETERS

Number of Aircraft
  Total              = 30;
  Concurrent max     = 30;

Wind                 = 150015;    (NPN, or dirmag in deg/knots)
Time of day          = 8;        (hh, 24h)
Surveillance period = 2.0;      (sec/cycle)
Prob. of Reception   = 1.0;      (0...1)

```

Figure A-1: Example configuration file. Continues on following page...

```

-----
COMMUNICATIONS

Datalink (CPDLC)

Link technical delay      (uniform, sec)
  Lower limit            = 0.50;
  Upper limit            = 3.50;

Pilot response delay     (Lognormal, sec)
  Mean                   = 28.60;
  S.D.                   = 38.80;

Voice (VHF DSB-AM)

Transaction time         (Lognormal, sec)
  Mean                   = 50.04;
  S.D.                   = 5.90;

Channel occupation delay (Lognormal, sec)
  Mean                   = 0.0;
  S.D.                   = 0.0;

ASAS

Pilot response delay     (Lognormal, sec)
  Mean                   = 28.60;
  S.D.                   = 38.80;

-----
AIR TRAFFIC

Flight plan directory = drive\folder;
Random Traffic       = No;          (No, Pairwise)
Scenario config file = drive\folder\PairwiseEncounters_Config.txt;

Assigned Traffic      (BADA FP TIME CONT COMM SEP)
B772 LHR_2_JFK 0 AP DL CENT
A320 LHR_2_JFK 0 AP DL CENT
B772 LHR_2_JFK 50 FD RT ASAS
B772 LHR_2_JFK 100 AP RT ASAS

```

Figure A-1: Example configuration file. Continued from previous page.

Appendix B Output File Format

Simulation data is periodically recorded to a binary-format random access data file for post-simulation analysis. The file consists of a header, the master array, and the time series of traffic state data, as illustrated in Figure B-1.

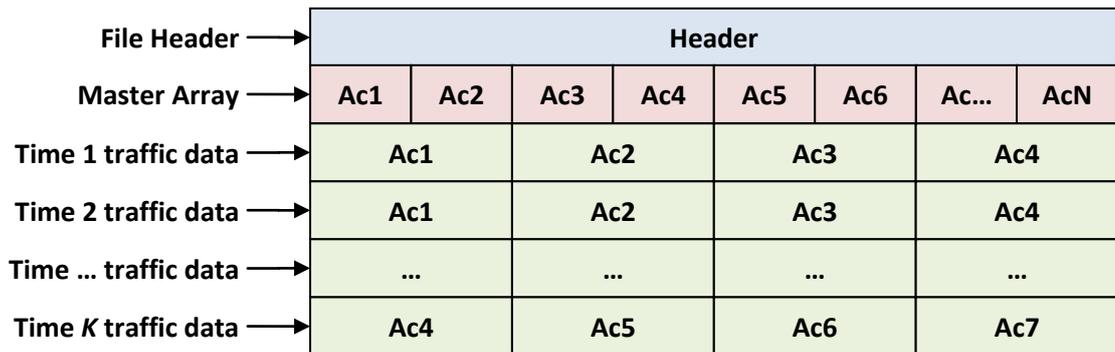


Figure B-1: File organisation

The header data consists of the data fields in Table B-1. Every element of the Master Array consists of the data fields in Table B-2. The length of the array is *MaxTotalAc*, from the header. The traffic state data consists of the data fields in Table AX-3. Every time block, *m*, represents the traffic at $m \times (TimeStepSize \times DataRes)$. There are a total of $(TimeStepSize \times DataRes) / TotalSimTime$ time blocks, with data for *MaxConcurAc* aircraft per time block.

Variable	C data type	Size (B)	Notes
NumAcFinished	short unsigned int	2	Total number of aircraft simulated
MaxTotalAc	short unsigned int	2	Length of the Master Array. May be different than numAcFinished
MaxConcurAc	short unsigned int	2	Length of the Data Block
DataRes	short unsigned int	2	Time steps between recording data
WindType	short unsigned	2	If a number: the wind dir/mag as format xxxyyy deg,knots. Else, the wind source:

Appendix B: Output File Format

	int		NPN = NOAA Profiler Network, FD = Area Forecast, or N = None
RndSeed	short unsigned int	2	Random number seed
TimeStepSize	float	4	Time step for integration (s/step)
TotalSimTime	float	4	Duration of simulation in simulated seconds
ExecutionTime	float	4	Duration of simulation in actual seconds
ChanOccDelayVoice_Mu	float	4	Logn mu: Channel occupation delay
ChanOccDelayVoice_Sigma	float	4	Logn sigma: Channel occupation delay
LinkTechDelay_Lower_lim	float	4	Uniform lower bound: Lnk technical delay
LinkTechDelay_Upper_lim	float	4	Uniform upper bound: Link technical delay
RespDelayASAS_Mu	float	4	Logn mu: ASAS crew response delay
RespDelayASAS_Sigma	float	4	Logn sigma: ASAS crew response delay
RespDelayCPDLC_Mu	float	4	Logn mu: Datalink crew response delay
RespDelayCPDLC_Sigma	float	4	Logn sigma: Datalink crew response delay
TransTimeVoice_Mu	float	4	Logn mu: Radio-telephony transaction times
TransTimeVoice_Sigma	float	4	Logn sigma: Radio-telephony transaction times

Table B-1: Header data fields

Appendix B: Output File Format

Variable	C data type	Size (B)	Notes
ContMode	unsigned char	1	1 = autopilot, 0 = flight director
CommMode	unsigned char	1	1 = datalink, 0 = radio-telephony
SepMode	unsigned char	1	2 = centralised, 1 = self-separation, 0 = uncontrolled
FpName[20]	char	21	Flight plan name
CurAcInd	short unsigned int	2	Aircraft index in the current aircraft array
BadaInd	short unsigned int	2	Index of aircraft performance model
AcID	unsigned int	4	Unique aircraft ID
Start	float	4	Start time in seconds from simulation beginning
Finish	float	4	End time in seconds from simulation beginning

Table B-2: Master Array data fields

Variable	Units	C data type	Size (B)	Notes
AcID		unsigned int	4	Unique aircraft ID
Lat	rads	Float	4	Geodetic latitude
Lon	rads	Float	4	Geodetic longitude
Alt	m	Float	4	Altitude above mean seal level (MSL)
Mass	kg	Float	4	Mass
Vtas	m/s	Float	4	True Airspeed
Vgnd	m/s	Float	4	Groundspeed
Trk	rads	Float	4	Ground track angle
Hdg	rads	Float	4	Heading (air-mass track angle)
ROCD	m/s	Float	4	Rate of Climb/Decent
Xte	m	Float	4	Cross track error from flight plan leg

Table B-3: Traffic state data fields

Appendix C

Flight Plan Format

Flight plans are ASCII text files with *.fpl* file extensions, and consist of the initial fix followed by a sequence of flight plan segments. An example flight plan is shown in Figure C-1. A description of the segment data fields is presented in Table C-1. Data fields are separated by white-space. Segments are separated by a new line.

```
INIT ENR IF FB S25:00.00 E0131:00.00 30000 0 0.0
WP01 ENR TF FB S25:00.00 E0131:30.00 30000 0 0.0
WP02 ENR TF FO S25:30.00 E0131:30.00 30000 0 0.0
WP03 ENR TF FO S25:30.00 E0132:00.00 30000 0 0.0
WP04 ENR TF FB S25:00.00 E0132:00.00 30000 0 0.0
WP05 ENR TF FB S25:00.00 E0132:30.00 30000 0 0.0
```

Figure C-1: Example flight plan

Appendix C: Flight Plan Format

Leg data	Format	Units	Notes:
Leg name	8 character string		The leg identifier. Up to 8 characters
Phase of flight	3 character string		The phase of flight ENR: en-route TMA: terminal
Leg Type	2 character string		IF: Initial Fix TF: Track-to-Fix DF: Direct-to-Fix
Transition Type	2 character string		FB: fly-by FO: fly-over
Latitude	1 character Integer:Integer. Integer	Degrees, minutes, decimal minutes	Latitude of terminating waypoint, using single character North/South indication
Longitude	1 character Integer:Integer. Integer	Degrees, minutes, decimal minutes	Longitude of terminating waypoint, using single character East/West indication
Leg Altitude	Integer	Feet	The target altitude of the leg above mean seal level.
Leg Speed	Integer	Knots	(optional) The target true airspeed of the leg. If set to 0, the target airspeed is set from the BADA performance data.
Lateral Offset	Floating point	Nautical Miles	(optional) The target parallel offset distance, using the convention of positive for right of path, and negative for left of path.

Table C-1: Leg segment data field

Appendix D

Scenario Generator Configuration File

The scenario generator configuration file is a plain text file named 'PairwiseEncounters_Config.txt,' read during simulator initiation if the simulator configuration file indicates the scenario generator is to be used. The order of data within the file is significant. Data is separated from plain text by an equal sign and is followed by a semicolon. Figure D-1 shows an example configuration file.

```
Pair-wise Encounters Random Scenario Generator configuration

Prefix
    = RMAtest1_;

Centre Point (N/S deg:min.dec E/W deg:min.dec)
    = N00:00.00 E000:00.00;

Time to first loss of separation (sec)
    = 200;

Horizontal Separation Minimums (nmi)
    = 5;

Distance beyond conflict to final waypoint (nmi)
    = 40;

Aircraft type model (UNI, Uniform from BADA list, EUR, Sheehan European traffic
model)
    = EUR;

Encounter angle distribution model (0-100, must sum to 100)
    In-trail (0-60 deg)           = 33.33;
    Crossing (60-120 deg)        = 33.33;
    Opposing (120-180 deg)       = 33.33;
```

Figure D-1: Example configuration file. Continues on following page...

```
Probability distribution of communications modes (0-100, must sum to 100)
  Ideal                = 100;
  Radio-telephone     = 0;
  Datalink             = 0;

Probability distribution of flight control modes (0-100, must sum to 100)
  Ideal                = 100;
  Autopilot           = 0;
  Flight Director     = 0;

Probability distribution of separation modes (0-100, must sum to 100)
  Centralised         = 100;
  Self-separation     = 0;
  Uncontrolled        = 0;

Altitude (ft)
  = 32000;

Terminator Type (TF, DF)
  = TF;

Overfly (FO)/ Fly-by (FB)
  = FB;

Phase of Flight (DEP, ENR, MNV, TRM, MAP)
  = ENR;
```

Figure D-1: Example configuration file. Continued from previous page.

Bibliography

- Alam, S., Abbass, H.A., and Barlow, M., 2008. ATOMS: Air traffic operations and management simulator. In: *IEEE Transactions on Intelligent Transportation Systems* 9 (2), art. no. 4526195, pp. 209-225.
- Allerton, D., 2009. *Principles of flight simulation*. New York (NY): John Wiley & Sons Inc.
- Alliot, J. M., Bosc, J. F., Durand, N., and Maugis, L., 1997. CATS: A Complete Air Traffic Simulator. In: *Proceedings of the 2nd AIAA/IEEE Digital Avionics Systems Conference*, pp. 8.2-30-8.2-37.
- Andrews, J., 2001. Capacity benefits of the Automated Airspace Concept (AAC): A preliminary investigation. MIT Lincoln Laboratory Report No. 42PM-AATT-0014, Cambridge (MA).
- Andrews, J., Erzberger, H., and Welch, J., 2006. Safety Analysis for Advanced Separation Concepts. *Air Traffic Control Quarterly* 14 (1), pp. 5-24.
- Archibald, J.K., Hill, J.C., Jepsen, N.A., Stirling, W.C., and Frost, R.L., 2008. A satisficing approach to aircraft conflict resolution. *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, 38 (4), pp. 510-521.
- Bach, R., Chu, Y., and Erzberger, H., 2009. A path-stretch algorithm for conflict resolution. NASA Ames Research Center. NASA/CR-2009-214574.
- Barhydt, R., Palmer, M.T., Chung, W.W., and Loveness, G.W., 2004. ADS-B within a multi-aircraft simulation for distributed air-ground traffic management. *The 23rd Digital Avionics Systems Conference*, Vol 1, 24-28 Oct. 2004.
- Baumeister, R., Estkowski, R., Spence, G., and Clothier, R., 2009. Test Architecture for Prototyping Automated Dynamic Airspace Control. *Proceedings of the CEAS 2009 European Air & Space Conference*, UK. 26th-29th Oct, 2009.
- Bilimoria, K.D., Sridhar, B., Chatterji, G.B., Sheth, K.S., and Grabbe, S.R., 2001. FACET: Future ATM concepts evaluation tool. *Air Traffic Control Quarterly* 9 (1), pp. 1-20.
- Bilimoria, K.D., 2000. A geometric approach to aircraft conflict resolution. AIAA-2000-4265, *AIAA Guidance, Navigation and Control Conference*, 14-17 August, Denver, Colorado.
- Blom, H., Obbink, B., and Bakker, G., 2007. Safety risk simulation of an airborne self separation concept of operation, *Proceedings 7th AIAA-ATIO Conference*, September 18-20, Belfast, Northern Ireland.

Bibliography

- Bolczak, R., Gonda III, J.C., Saumsiegle, W.J., and Tornese, R.A., 2004. Controller-pilot data link communications (CPDLC) Build 1 value-added services, *The 23rd Digital Avionics Systems Conference*.
- Brudnicki, D., Lindsay, K., and McFarland, A., 2007. Assessment of field trials, algorithmic performance, and benefits of the user request evaluation tool (URET) conflict probe. *Proceedings of the 16th Digital Avionics Systems Conference*, Irvine(CA), pp. 35-44.
- Bürki-Cohen, J. 1996. How to say it and how much: The effect of format and complexity on pilot recall of air traffic control clearances. In B. Kanki & V. Prinzo (Eds), *Methods and metrics of voice communications* (Report DOT/FAA/AM-96/10). Washington, DC: Federal Aviation Administration.
- Callantine T., 2007. Tradeoffs in high density trajectory-based operations, *AIAA/IEEE Digital Avionics Systems Conference - Proceedings*, art. no. 4391881, pp. 3C11-3C112.
- Cardosi, K., 1993. Time required for transmission of critical Air Traffic Control messages in an en route environment. *The International Journal of Aviation Psychology*, 3(4), 303-313.
- Carr, F., Kolitz, S., Lepanto, J., Scheidler, P., Wilde, J., and Smith, J.C., 2005. An air traffic system simulation of the NAS with CNS and CD&R models. *Collection of Technical Papers - AIAA Modeling and Simulation Technologies Conference 2005*, 2, pp. 1077-1085.
- Cetek, C., 2009. Realistic Speed change maneuvers for air traffic conflict avoidance and their impact on aircraft economics. *International Journal of Civil Aviation*, Vol. 1, No. 1.
- Chaloulos, G., Roussosy, G.P., Lygerosz, J., and Kyriakopoulosx, K.J., 2008. Ground assisted conflict resolution in self-separation airspace. *Proceedings of the AIAA Guidance, Navigation and Control Conference and Exhibit*, Honolulu, Hawaii, Aug. 18-21, 2008. AIAA-2008-6967.
- Chen, H., and Zhao, Y., 2009. Required action time and control effectiveness in resolving pairwise conflicts. *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Chicago, Illinois, Aug. 10-13, 2009. AIAA-2009-5749.
- Clothier, R., and Walker, R., 2009. The smart skies project, *2009 AUVTI North America Conference*, Washington, DC.
- Cole, R., Green, S., Jardin, M., Schwartz, B., and Benjamin, S., 2000. Wind prediction accuracy for air traffic management decision support tools. *3rd USA/Europe Air Traffic Management R&D Seminar*, Napoli, Italy.
- Consiglio, M., Hoadley, S., Wing, D., and Baxley, B., 2007. Safety performance of airborne separation: preliminary baseline testing. *Seventh AIAA Aviation Technology, Integration and Operations (ATIO) Conference*, AIAA, Hilton Head, SC, USA.

Bibliography

- Consiglio, M., Hoadley, S., Wing, D., Baxley, B., and Allen, D., 2008. Impact of pilot delay and non-responsiveness on the safety performance of airborne separation. *Eighth AIAA Aviation Technology, Integration and Operations (ATIO) Conference*, AIAA, Honolulu, HI, USA.
- De La Fuente Layos, L., 2009. Air passenger transport in Europe in 2007. EUROSTAT, *Statistics in Focus*, 1/2009.
- De Smith, M., Goodchild, M.F., and Von Longley Troubador, P.A., 2009. *Geospatial analysis: A comprehensive guide to principles, techniques and software tools*. Matador, Market Harborough Press.
- Del Pozo de Poza, I., Ruiz, M.V., and Goodchild, C., 2009. Assessing fairness and equity in trajectory-based operations. *9th AIAA Aviation Technology, Integration, and Operations Conference (ATIO)*, Hilton Head (SC), Sep. 21-23. AIAA-2009-6973.
- Delhaise, P., and Esposito, M., 2006. VDL Mode 2 capacity analysis through simulations. *WP5 - Simulation Results*, Edition 2.
- DLBST (Data Link Benefits Study Team), 1996. Benefits of controller-pilot data link atc communications in terminal airspace: final report. FAA Report number DOT/FAA/CT-96/3.
- Doble, N. Barhydt, R., and Hitt, J.M., 2005. Distributed conflict management in en route airspace: human-in-the-loop results. *Proceedings of the 24th AIAA/IEEE Digital Avionics Systems Conference (DASC)*, Wash., DC.
- Donohue, G. L., and Laska, W. D., 2001. United States and European airport capacity assessment using the GMU macroscopic capacity model. In *Air Transportation Systems Engineering*, Donohue, G. L., and Zellweger, A. G., editors.
- Duong, V. N., and Hoffman, E. G., 1997. Conflict resolution advisory service in autonomous aircraft operations. *AIAA/IEEE Digital Avionics Systems Conference - Proceedings 2*, pp. 9.3-10-9.3-17.
- Dwyer, J.P., and Landry, S., 2009. Separation assurance and collision avoidance concepts for the next generation air transportation system. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 5618 LNCS (PART 2), pp. 748-757.
- ECC (EUROCONTROL Experimental Centre), 2009. Data analysis report. ADAPT2 (Aircraft Data Aiming at Predicting the Trajectory 2) Project.
- Erzberger H., and Paielli R., 2002. Concept for next generation air traffic control system. *Air Traffic Control Quarterly*, vol. 10, no. 4, Winter 2002, pp. 355-378.

Bibliography

- Erzberger, H., 2006. Automated conflict resolution for air traffic control. *Proceedings of the 25th International Congress of the Aeronautical Sciences (ICAS)*, Hamburg, Germany, 3-8 Sep. 2006.
- Erzberger, H., Davis, T. J., and Green, S., 1993. Design of center-TRACON automation system. *AGARD Meeting on Machine Intelligence in Air Traffic Management*, Berlin, Germany, 11-14 May 1993.
- Erzberger, H., and Heere, K., 2010. Algorithm and operational concept for resolving short-range conflicts. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering* 224 (2), pp. 225-243.
- EUROCONTROL (European Organisation for the Safety of Air Navigation), 2003. Guidance material for the flight inspection of RNAV procedures. v3.0.
- EUROCONTROL (European Organisation for the Safety of Air Navigation), 2005. The Surveillance Strategy for ECAC, Ed. 2.0.
- EUROCONTROL (European Organisation for the Safety of Air Navigation), 2007. *Uplink 9*, Link 2000+ Programme Newsletter, June 2007.
- EUROCONTROL (European Organisation for the Safety of Air Navigation), 2007. Real-time simulator platform. <http://www.eurocontrol.int/crds/public/standard_page/what_rts.html> Accessed 17 Jan. 2009.
- EUROCONTROL (European Organisation for the Safety of Air Navigation), 2008. Navigation application & NAVAID infrastructure strategy for the ECAC area up to 2020, v2. Navigation Sub-Group (NSG) of the ANT.
- EUROCONTROL (European Organisation for the Safety of Air Navigation), 2010a. Medium term forecast: IFR flight movements 2010-2016. STATFOR Doc378.
- EUROCONTROL (European Organisation for the Safety of Air Navigation), 2010b. ATC data link operational guidance for LINK 2000+ services, v5.1. Prepared by LINK 2000+ Operational Focus Group.
- EUROCONTROL (European Organisation for the Safety of Air Navigation), 2010c. Area navigation in European terminal control: P-RNAV module. EUROCONTROL Training Zone. <<https://trainingzone.eurocontrol.int>>. Accessed 12 Jan 2010.
- FAA (Federal Aviation Administration), 1996. Benefits of controller-pilot data link ATC communications in terminal airspace, FAA Report FAA/CT-96-3, September 1996.

Bibliography

- FAA (Federal Aviation Administration), 2002. 1090 MHz extended squitter assessment report. Federal Aviation Administration and EUROCONTROL Experimental Centre. 1090-WP-12-05. June 2002.
- FAA (Federal Aviation Administration), 2004. *FAR/AIM: Federal Aviation Regulations/Aeronautical Information Manual*.
- FAA (Federal Aviation Administration), 2006. Roadmap for performance based navigation, v. 2.0. US Department of Transportation, Wash., DC.
- FAA (Federal Aviation Administration), 2007. Terminal area forecast summary. FAA-APO-07-1, US Department of Transportation, Wash., DC.
- FAA (Federal Aviation Administration), 2010. Air traffic control. FAA order 7110.65. Department of Transportation, Wash., DC.
- FAA (Federal Aviation Administration), 2010. FAA aerospace forecast fiscal years 2010–2030. US Department of Transportation, Wash., DC.
- Fairley, G.T., and McGovern, S., 2009. A kinematic/kinetic hybrid airplane simulator model. *ASME International Mechanical Engineering Congress and Exposition, Proceedings 1*, pp. 11-20.
- Fan, T. P., and Kuchar, J. K., 2000. Evaluation of Interfaces for Pilot-Air Traffic Control Data Link Communications. *Proceedings of the 19th IEEE Digital Avionics Systems Conference*.
- Farley, T., Kupfer, M., and Erzberger, H., 2007. Automated conflict resolution: A simulation evaluation under high demand including merging arrivals. *Proceedings of the 7th AIAA Aviation Technology, Integration and Operations Conference*, Belfast, Northern Ireland, Sep. 18-20, 2007. AIAA-2007-7736.
- Feigh, K., 2003. *An airspace simulator for air traffic management research*. Master of Philosophy Thesis, Cranfield University, UK.
- Flanagan, Paul D., Currier, Judith B., Willis, and Kenneth E., 1973. Simulation in the design of automated air traffic control functions. *Proceedings of the 6th conference on Winter Simulation Conference*, San Francisco (CA), pp 449 - 462.
- Forest, L., and Hansman, R. J., 2006. The future oceanic ATC environment: analysis of mixed communication, navigation, and surveillance equipage. *ATC Quarterly, V14 (2)*, pp.117-138.
- Funabiki, K., Iijima, T., and Nojima, T., 2003. Evaluation of a trajectory-based operations concept for small aircraft: Airborne aspect, *The 22nd AIAA/IEEE Digital Avionics Systems Conference – Proceedings*, Volume 2. pp.12.C.5 - 121-11.

Bibliography

- Galati, G. Naldi, M. and Pavan, G., 2003. Stochastic simulation techniques as related to innovation in communications-navigation-surveillance and air traffic management (CNS/ATM). *Simulation Modelling Practice and Theory*, 11(3-4), pp.197-209.
- Gawinowski, G., Drogoul, F., Guerreau, R., Weber, R., and Garcia, J., 2008. Erasmus contribution to the 2020 SESAR scenario. *AIAA/IEEE Digital Avionics Systems Conference - Proceedings*, art. no. 4702793, pp. 3A21-3A210.
- Geisinger, K., 2003. Guide to methods and tools for safety analysis in air traffic management. First edition, Global Aviation Information Network, Working Group B, Analytical Methods and Tools. <http://flightsafety.org/files/methods_tools_safety_analysis.pdf>.
- Gonda, S., Saumsiegle, W., Blackwell, B., and Longo, F., 2005. Miami controller-pilot data link communications summary and assessment. *The Sixth International ATM R&D Seminar ATM-2005*.
- Graglia, L., 2002. Etude vocalise: Analyse générale trafic CRNA / France 2000, version 1.0. CENA/ICS/R02-002.
- Grayson, R., and Billings, C., 1981. Information transfer between air traffic control and aircraft: Communication problems in flight operations. In C. Billings and E. Cheaney (Eds.), *Information transfer problems in the aviation system* (NASA Technical Paper 1875, pp. 47-62). Moffett Field, CA: NASA Ames Research Center.
- Green, S. M., and Vivona, R. A., 2001. En route descent advisor concept for arrival metering. AIAA-2001-4114, *Guidance, Navigation, and Control Conference*, Montreal, Canada.
- Grogan, P., 2007. ATN/VDL Mode 2 Capabilities. *7th ICNS Conference*, Herndon, Virginia May 1-3.
- Guo, J., Liu, F., and Zhu, Z., 2007. Estimate the call duration distribution parameters in GSM system based on K-L divergence method. *International Conference on Wireless Communications, Networking and Mobile Computing, WiCOM 2007*, art. no. 4340517, pp. 2988-2991.
- Herndon, A.A., Dearmon, J.S., and Spelman, J., 2004. Use of lateral/parallel FMS procedures and implementation issues. *Proceedings of the AIAA/IEEE Digital Avionics Systems Conference*, pp. 2.D.3-1-2.D.3-8.
- Hockley, W.E., 1984. Analysis of Response Time Distributions in the Study of Cognitive Processes, *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 10, 598-615.

Bibliography

- Hoekstra, J., Ruigrok, R., van Gent, R., Visser, J., Gijsbers, B., Valenti Clari, M., Heesbeen, W., Hilburn, B., Groeneweg, J., and Bussink, F., 2000. Overview of NLR free flight project 1997-1999. NLR-TP-2000-227.
- Hoekstra, J.M., Ruigrok, R.C.J., and van Gent, R.N.H.W., 2000. Free flight in a crowded airspace. *Proceedings of the 3rd USA/Europe Air Traffic Management R&D Seminar*, Napoli, 13-16, June 2000.
- Hung, Brian T., 2005. *An Analysis of En Route Domain Air Traffic Control Voice Transcripts*, MITRE Technical Report, MTR 05W0000071, The MITRE Corporation, McLean, VA.
- Hunter, G., 1996. CTAS error sensitivity, fuel efficiency, and throughput benefits analysis, Tech. Report 96150-02, Seagull Technology, Los Gatos, CA.
- ICAO (International Civil Aviation Organization), 1999. *Manual on required navigation performance (RNP)*. Second edition. Doc 9613-AN/937.
- ISA Software, 2003. RAMS Plus Simulation Solutions. <<http://www.ramsplus.com/files/What%20Is%20RAMS%20Plus.pdf>> Accessed 21 Jan. 2009.
- Jeppesen, 2008. TAAM Solutions: Total Airspace and Airport Modeller. <<http://www.jeppesen.com/documents/aviation/government/TAAM-ebrochure.pdf>> Accessed 17 Jan. 2009.
- Jha, P.D., Crook, I., 2009. Benefit assessment of advanced navigation (RNAV/RNP) procedures at Dallas Fort Worth (DFW). *Air Traffic Control Association - 54th Air Traffic Control Association Annual Conference 2009*, pp. 29-35.
- Jones, E., and Schleicher, D.R., 2001. En-Route Data Exchange (EDX): Phase 2 field evaluation final report. Prepared for FAA Aeronautical Data Link (ADL) Product Team.
- Jonker, G., Meyer, J.-J., and Dignum, F., 2005. Efficiency and fairness in air traffic control. *Proceedings of the 17th Belgian-Dutch Conference on Artificial Intelligence*.
- JPDO (Joint Planning and Development Office), 2004. Next generation air transportation system integrated plan. Wash., DC.
- JPDO (Joint Planning and Development Office), 2009. NextGen CONOPS: Concept of operations for the next generation air transportation system, V3.0. Wash., DC.
- Karr, D. A., Vivona, R. A., Roscoe, D. A., DePascale, S. M., and Consiglio, M., 2009. Experimental performance of a genetic algorithm for airborne strategic conflict resolution. *AIAA Guidance, Navigation, and Control Conference (GNC) 2009*, 10-13 Aug. 2009, Chicago, IL.

Bibliography

- Kauppinen, S., Brain, C., and Moore, M., 2002. European medium term conflict detection field trials. *Proceedings of the AIAA/IEEE Digital Avionics Systems Conference*, pp. 2C11-2C112.
- Kayton, M., and Fried, W. R. 1997. *Avionics navigation systems*, 2nd edition. John Wiley & Sons Inc., New York, NY.
- Kernighan, B.W., and Ritchie, D.M., 1988. *The C programming language, second edition*. Prentice Hall, Englewood Cliffs (NJ).
- Kirk, D.B., Bowen, K.C., Heagy, W.S., Rozen, N.E., and Viets, K.J., 2001. Problem analysis, resolution and ranking (parr) development and assessment. *4th USA/Europe Air Traffic Management R&D Seminar*, Santa Fe(CA).
- Kirk, D.B., Heagy, W.S., and Yablonski, M.J., 2001. Problem Resolution Support for Free Flight Operations. *IEEE Transactions on Intelligent Transportation Systems 2 (2)*, pp. 72-79.
- Kirwan, B. and Flynn, M., 2002. Towards a controller-based conflict resolution tool – a literature review. ASA.01.CORA.2.DEL04-A.LIT. <http://www.eurocontrol.int/eatm/gallery/content/public/library/litreview.pdf>.
- Knox, C. E., and Scanlon, C. H., 1990. Flight tests using data link for air traffic control and weather information exchange. *SAE (Society of Automotive Engineers) Transactions 99 (Sect 1)*, pp. 1683-1699.
- Kopardekar, P., Smith, N., Lee, K., Aweiss, A., Lee, P., Prevot, T., Mercer, J., and Mainini, M., 2009. Feasibility of mixed equipage operations in the same airspace. *Eighth USA/Europe Air Traffic Management Research and Development Seminar*.
- Kuchar, J.K., and Yang, L.C., 2000. A review of conflict detection and resolution modelling methods. *IEEE Transactions on Intelligent Transportation Systems 1 (4)*, pp. 169-179.
- Lambregts, A.A., 1998. Automatic flight controls concepts and methods. FAA National Resource Specialist, Advanced Controls.
- Lee, P.U., Smith, N., Mercer, J.S., Battiste, V., Johnson, W., Martin, L., Mogford, R., and Verma, S., 2003. Free maneuvering, trajectory negotiation, and self-spacing concepts in distributed air-ground traffic management. *Proceedings of the 5th USA/Europe Air Traffic Management Research and Development Seminar*, Budapest, Hungary.
- Lei, Z., Jun, Z., Yanbo, Z., and Peng, W., 2008. Game-theoretical method for conflict resolution. *AIAA/IEEE Digital Avionics Systems Conference - Proceedings*, pp. 3D61-3D68.

Bibliography

- Lester, E., and Hansman, R.J., 2007. Benefits and incentives for ads-b equipage in the national airspace system. MIT International Center for Air Transportation. Report No. ICAT-2007-2.
- Levy, B.S., Som, P., and Greenhaw, R., 2003. Analysis of flight technical error on straight, final approach segments. *Proceedings of the 59th Annual Meeting of the Institute of Navigation, Albuquerque(NM)*.
- Liang, D., Marnane, W., and Bradford, S., 2001. Comparison of U.S. and European airports and airspace to support concept validation. *Progress in Astronautics and Aeronautics*, v.93 pages 27-47. AIAA, Reston, VA.
- Loftus, G., Dark, V., and Williams, D., 1979. Short-term memory factors in ground controller/pilot communication. *Human Factors*, 21, 169-181.
- Loscos, J.-M., Drévilion, H., Graniero, G., Foster, C., Miquel, T., and Degelder, N., 2007. D7.3 - Final report and recommendations, *Sixth Framework Programme, Priority 1.4, Aeronautics and Space, Advanced Safe Separation Technologies and Algorithms*.
- Lozito, S., Verma, S., Martin, L., Dunbar, M., and McGann, A., 2003. The Impact of voice, data link, and mixed air traffic control environments on flight deck procedures, *Air Traffic Control Quarterly* 11 (4), pp. 293-310.
- Mackintosh, M., Lozito S., McGann A., and Logsdon, E., 1999. Designing procedures for controller-pilot data link communication: effects of textual data link on information transfer, *1999 World Aviation Conference*, October 19-21, San Francisco, CA.
- Mayer, R.H., 2002. A flight trajectory model for a pc-based airspace analysis tool. *AIAA Modeling and Simulation Technologies Conference and Exhibit*.
- Miquel, T., Chamayou, C., Louyot, P., Loscos, J.-M., and Anderson, J., 2007. Assessment of navigation errors on airborne state-based conflict resolution. *Proceedings of the AIAA/IEEE Digital Avionics Systems Conference*, art. no. 4391901, pp. 4A61-4A610.
- Mondoloni, S., 2006. Aircraft trajectory prediction errors: Including a summary of error sources and data, v. 0.2. Prepared for the FAA/Eurocontrol Action Plan 16: Common Trajectory Prediction Capabilities.
- Monticone, L., Snow, R., and Wang, P., 2005. Air/Ground communications traffic modeling capability for the Mid-Level Model (MLM). Mitre Technical Report 05W0000080.
- Morrow, D., and Rodvold, M., 1993. The influence of ATC message length and timing on pilot communications. (NASA contractor report 177621). Moffett Field, CA: NASA Ames Research Center.

Bibliography

- Mueller E., 2007. Experimental evaluation of an integrated datalink and automation-based strategic trajectory concept. *Collection of Technical Papers - 7th AIAA Aviation Technology, Integration, and Operations Conference 1*, pp. 831-845.
- Murphy, J.R., and Robinson, J.E., 2007. Design of a research platform for en route conflict detection and resolution. *Collection of Technical Papers - 7th AIAA Aviation Technology, Integration, and Operations Conference 2*, pp. 1092-1106
- Nolan, M.S., 2004. *Fundamentals of Air Traffic Control*. 4th ed. Belmont(CA): Brooks/Cole.
- Nuic, A., 2010. User manual for the base of aircraft data (BADA) revision 3.8. EUROCONTROL Experimental Centre Technical/Scientific Report No. 2010-003, Bretigny-sur-Orge, France.
- Paglione, M. M., Oaks, R. D., and Bilimoria, K. D., 2003. Methodology for generating conflict scenarios by time shifting recorded traffic data. AIAA-2003-6724, American Institute of Aeronautics & Astronautics, Reston, Virginia, 2003.
- Paielli, R., 2005. Trajectory specification for high-capacity air traffic control. *Journal of Aerospace Computing, Information and Communication* 2 (9), pp. 361-385.
- Paielli, R.A., 2008. Tactical conflict resolution using vertical maneuvers in en route airspace. *Journal of Aircraft* 45 (6), pp. 2111-2119.
- Pallottino, L., Feron, E.M., and Bicchi, A., 2002. Conflict resolution problems for air traffic management systems solved with mixed integer programming. *IEEE Transactions on Intelligent Transportation Systems, Vol. 3, No. 1*.
- Palmer, E., Prevot, T., Romahn, S., Smith, N., Callantine, T., Williams, D., and Osegura-Lohr, R., 2000. Linking cockpit and air traffic control automation: CTAS/FMS Integration. DAG-TM Industry Workshop, May 22 to 25, 2000.
- Peters, M., and Konyak, M.A., 2003. The engineering analysis and design of the aircraft dynamics model for the FAA target generation facility. Tech. Report 99162-01, Seagull Technology, Los Gatos, CA.
- Pina P., and Hansman, R. J., 2004. Cognitive and operational implications of non-homogeneous aircraft equipage for aviation system transformation. ICAT-2007-4.
- Porras, J.F., and Parra, M. 2007. ATM initiatives on reduced separation minima. IEEE, 26th Digital Avionics Systems Conference, 21 October 2007.
- Pratt, R., 2000. *Flight control systems: Practical issues in design and implementation*. In IEE Control Engineering Series 57. The Institution of Electrical Engineers, Herts, UK.

Bibliography

- Prevôt T., Homola J., and Mercer J., 2008. Human-in-the-Loop evaluation of ground-based automated separation assurance for NextGen. *26th International Congress of the Aeronautical Sciences (ICAS) and AIAA-ATIO, Anchorage(AK)*.
- Prevôt, T., Homola J., Mercer J., Mainini M., and Cabrall C., 2009. Initial Evaluation of NextGen Air/Ground Operations with Ground-Based Automated Separation, *Eighth USA/Europe Air Traffic Management Research and Development Seminar*.
- Prevôt, T., Lee, P., Callantine, T., Smith, N., and Palmer, E., 2003. Trajectory-oriented time-based arrival operations: Results and recommendations, *Proceedings of the 5th USA/Europe Air Traffic Management Research and Development Seminar*, Budapest, Hungary.
- Prevot, T., Lee, P., Smith, N., and Palmer, E., 2005. ATC technologies for controller-managed and autonomous flight operations. *Collection of Technical Papers - AIAA Guidance, Navigation, and Control Conference 3*, pp. 1806-1848.
- Rantanen, E.M, Yang, J., and Yin, S., 2006. Comparison of pilots' and controllers' conflict resolution maneuver preferences. *Proceedings of the Human Factors and Ergonomics Society 50th Annual Meeting*.
- Ratcliff, R. and Murdock, B.B., Jr., 1976. Retrieval processes in recognition memory. *Psychological Review*, 83, 190-214.
- Rosen, K.M., 2005. Analysis of speech segment duration with the lognormal distribution: A basis for unification and comparison, *Journal of Phonetics*, 33 (4), pp. 411-426.
- Roskam, J., 2003. *Airplane flight dynamics and automatic flight controls, part II*. DAR Corporation, Lawrence(KS).
- RTCA (Radio Technical Commission for Aeronautics), 2006. Safety, performance and interoperability requirements document for the ADS-B non-radar-airspace (ADS-B NRA) application. RTCA DO-303 and EUROCAE ED-126.
- Ruigrok, R.C.J., and De Gelder, N., 2006. Flight testing the airborne separation assistance system. *Collection of Technical Papers - AIAA Guidance, Navigation, and Control Conference 2006 1*, pp. 339-362.
- Samet, H. 1990. *The Design and Analysis of Spatial Data Structures*. Addison-Wesley Longman Publishing Co., Inc., Boston(MA).
- Schoemig, E.G., Armbruster, J., Boyle, D., Haraldsdottir, A., and Scharl, J., 2006. 3D Path Concept and Flight Management System (FMS) Trades. *25th IEEE/AIAA Digital Avionics Systems Conference*. pp. 1-12.

Bibliography

- SESAR (Single European Sky ATM Research), 2006. Air transport framework: The current situation. *SESAR Definition Phase -Milestone Deliverable 1*. The SESAR Consortium.
- SESAR (Single European Sky ATM Research), 2007. The ATM target concept. *SESAR Definition Phase, Deliverable 3*, DLM-0612-001-02-00.
- Sheehan, C., 2009. Coverage of 2008-09 European air traffic for the base of aircraft data (BADA) Revision 3.7. EUROCONTROL Experimental Centre Technical/Scientific Report No. 2009-011, Bretigny-sur-Orge, France.
- Signor, D., Davis, P., Lozito, S., Andre, A., Sweet, D., and Wallace, E., 2004. Efficient air traffic scenario generation. *AIAA 4th Aviation Technology, Integration and Operations (ATIO) Forum*.
- Sinnott, R. W. 1984. Virtues of the Haversine. *Sky and Telescope* 68 (2), pp159-161.
- Sokolowski, J., and Banks, C., 2009. *Principles of Modeling and Simulation: A Multidisciplinary Approach*. John Wiley & Sons: New Jersey.
- Spence, G., and Allerton, D., 2009. A genetic approach to automated aircraft separation. CEAS 2009 European Air & Space Conference, UK. 26th-29th Oct, 2009.
- Spence, G., and Allerton, D., 2009. Simulation of an automated separation management communication architecture for uncontrolled airspace. 10th AIAA Modelling & Simulation Technologies Conference, Chicago.
- Spence, G.T., 2009. *Traffic Visualisation Software Instructions*. The University of Sheffield, UK.
- Spence, G.T., Allerton, D.J., Baumeister, R., and Estkowski, R., 2008. Real-time simulation of a distributed conflict resolution algorithm. *Proceedings of the 26th International Congress of the Aeronautical Sciences*.
- Spitzer, C., 2001. *The avionics handbook*. The Electrical Engineering Handbook Series, CRC Press, USA.
- Sprong, K.R., and Mayer, R.H., 2007. Analysis of RNAV arrival operations with descend via clearances at phoenix airport. *AIAA/IEEE Digital Avionics Systems Conference - Proceedings*, art. no. 4391874, pp. 3A51-3A512.
- Suchkov, A., Swierstra, S., and Nuic, A., 2003. Aircraft performance modeling for air traffic management applications. *5th USA/Europe Seminar on ATM R&D*, Budapest, Hungary, 2003.
- Sweet, D.N. Manikonda, V. Aronson, J.S. Roth, K. and Blake, M., 2002. Fast-Time simulation system for analysis of advanced air transportation concepts. *AIAA Modeling and*

Bibliography

Simulation Technology Conference, Monterey, California, 5-8 August 2002. AIAA-2002-4593.

Thompson, S.D., Andrews, J.W., Harris, G.S., and Sinclair, K.A., 2006. Required surveillance performance accuracy to support 3-mile and 5-mile separation in the national airspace system. MIT Lincoln Laboratory Report No. ATC-323, Cambridge(MA).

Ulrich, R., and Miller, J., 1993. Information processing models generating lognormally distributed reaction times. *Journal of Mathematical Psychology* 37 (4), pp. 513-525.

USAF (United States Air Force), 2001. *Air Navigation*. Air Force Pamphlet 11-216, 1 March 2001.

Vakil S.S., and Hansman R.J. Jr., 2002. Approaches to mitigating complexity-driven issues in commercial autoflight systems. *Reliability Engineering and System Safety*, 75 (2), pp. 133-145.

van Gool, M., and Schröter, H., 1999. PHARE final report, v 1.0. Prepared for: The Programme for Harmonised ATM Research in EUROCONTROL (PHARE). Doc. PHARE/EHQ/MAN/FR.

Vela, A.E., Salaün, E., Solak, S., Feron, E., Singhose, W., and Clarke, J.-P., 2009. A two-stage stochastic optimization model for air traffic conflict resolution under wind uncertainty. *AIAA/IEEE Digital Avionics Systems Conference - Proceedings*, art. no. 5347531, pp. 2.E.51-2.E.513.

Vela, A.E., Solak, S., Clarke, J.-P.B., Singhose, W.E., Barnes, E.R., and Johnson, E.L., 2010. Near real-time fuel-optimal en route conflict resolution. *IEEE Transactions on Intelligent Transportation Systems*, PP(99), pp 1 – 12.

Walpole, R., Myers, R., and Myers, S., 2002. *Probability and Statistics for Engineers and Scientists*, 7th ed. Prentice Hall (NJ).

Wichman, K.D., Klooster, J.K., Bleeker, O.F., and Rademaker, R.M., 2007. Flight validation of downlinked flight management system 4D trajectory, *AIAA/IEEE Digital Avionics Systems Conference - Proceedings*, art. no. 4391833, pp. 1D11-1D110.

Zemrowski, K.M., 2008. Impacts of increased reliance on automation in ATM. *Proceedings of the 2nd IEEE Systems Conference*, Montreal, Que. pp 1-6.

SDG