



CRANFIELD UNIVERSITY

Computational Air Traffic Management

Marc Anthony Azzopardi

School of Aerospace, Transport and Manufacturing (SATM)
PhD Thesis

CRANFIELD UNIVERSITY

SCHOOL OF AEROSPACE, TRANSPORT AND MANUFACTURING (SATM)

PhD THESIS
Marc Anthony Azzopardi

Computational Air Traffic Management

Supervisor: Dr J.F. Whidborne

April 2015

© Cranfield University 2015. All rights reserved. No part of this publication may be reproduced without the written permission of the copyright owner.

ABSTRACT

World air transport has been on a steady exponential rise since the 1940's and the trend has shown remarkable resilience to external shocks. The level of air traffic has greatly exceeded the wildest expectations of the air traffic management pioneers that originally defined the basic precepts ATM that persist till today.

This has stretched ATM to a point where it is starting to show signs of ineffectiveness in the face of ever increasing congestion. Delays are on the rise, costs are ballooning, flights are being elongated unnecessarily, the system is becoming increasingly susceptible to disruption, and the high environmental impact of aviation is being compounded by the inability of air traffic controllers to optimise ATM operation in real-time. If these trends are not reversed, ATM could eventually face instability. The conservative, self-preserving outlook of the ATM community has confined progress to relatively minor tweaks of a tired human-centric paradigm. However, the diverging gap between ATM performance and fundamental requirements indicates the need for a step change.

In this work, the traditionally incremental approach to ATM research was broken to favour a more exploratory mindset. As a result, a new discipline called Computational Air Traffic Management has been defined to address the unique set of challenges presented by the ATM problem, by taking a more objective scientific approach.

A specific embodiment of a CATM system was designed, constructed, simulated and tested and shown to be a significant step towards demonstrating the feasibility of a fully autonomous multi-agent-based air transportation system based on optimisation principles. The system offers unique advantages in terms of resilience to disruption, efficiency and future scalability. The traffic density using such a system can be realistically increased many times higher than current levels while significantly improving on the current levels of safety, operating cost, environmental impact and flight delays. This work advances the field of ATM as well as the fields of Computational Intelligence and Dynamic Optimisation of High Dimensionality Non-Convex Search Spaces.

DEDICATIONS

To my brother Carl,
my fiancée Christine, and my parents

Acknowledgements

I would like to thank all the people who have supported me during this lengthy journey that has seen the highest highs and the lowest lows. I knew from the outset that it was going to be a technically challenging journey, but I did not know just how much until I was neck deep in a subject that has kept me fascinated for many years.

First and foremost, I would like to give a hearty thanks to my supervisor, Dr. James Whidborne who has offered his indispensable knowledge whenever I needed some insight. He has been a great motivator and has supported my unconventional approach to tackle a field that has dissuaded many.

I would also like to thank Dr John Betts, Dean of the Faculty of Engineering at the University of Malta for his unwavering support that kept me going when the obstacles looked too great. Similar gratitude goes to my previous head of department Prof. Carmel Pulé and Pro-Rector Prof. Richard Muscat.

Last but not least, I would like to thank my parents, siblings and fiancée for putting up with this roller coaster for far too long.

Thank you all!

TABLE OF CONTENTS

	Page
PREFACE.....	1
I. THE RESEARCH CONTEXT	2
II. THESIS SCOPE, OBJECTIVES AND LIMITATIONS	4
III. THESIS OUTLINE AND CONTRIBUTIONS	4
A. Research Questions	4
B. Methodology	5
C. Contributions.....	5
D. Organization of the Dissertation	7
IV. PUBLICATIONS ARISING FROM THIS WORK.....	8
A. Published.....	8
B. In Press	8
C. In preparation	8
 CHAPTER 1 - INTRODUCTION	 9
1.1 THE MOTIVATION	9
1.1.1 The Problem Background	10
1.1.2 The Cost of Human ATM	11
1.1.3 Future Trends in Demand	12
1.2 RESEARCH IN ATM	13
1.2.1 SESAR and NextGen	14
1.2.2 Full Autonomy?	15
1.3 THE NATURE OF THE ATM-ATC PROBLEM	15
1.4 CHARACTERISTICS OF AN IDEAL ATM SYSTEM	18
1.5 THE SCIENCE OF ATM	22
1.6 JOINING THE DOTS... THE EMERGENCE OF CATM	23
1.6.1 Free flight.....	25
1.6.2 The Role of the Human ATCo.....	26
1.6.3 ASAS and Workload Sharing	27
1.6.4 Distributed Control	27
1.7 THE EXPECTED EVOLUTION OF ATC AUTOMATION.....	28
1.8 THE IMPLEMENTATION OF COMPUTATIONAL ATM.....	33
1.9 THE TRANSITION TO CATM SYSTEMS	34
1.10 THE OUTLOOK.....	35
1.11 REFERENCES	36

CHAPTER 2 - COMPUTATIONAL AIR TRAFFIC MANAGEMENT	41
2.1 A NEW CONCEPT FOR ATM	41
2.1.1 Computational Complexity Considerations	43
2.1.2 Relaxing Optimality Requirement to Address Complexity	44
2.1.3 Nested Optimisation	44
2.1.4 The benefits of Dynamic Re-Convergence	45
2.1.5 Constraint modelling in CATM	46
2.1.6 Decentralised CATM	47
2.1.7 Offline vs. Online CATM	47
2.1.8 Airborne CATM	48
2.1.9 CATM and the Role of the Pilot in an Autonomous system	49
2.1.10 High Performance Grid-Avionics and CATM	49
2.1.11 Heterogeneous Computing and Fault Tolerance	51
2.1.12 Clustering and Information Flow in CATM	52
2.1.13 Communication Infrastructure	53
2.1.14 Sensor Networks	56
2.1.15 CATM & Flight Data Recorders	56
2.1.16 CATM Trajectory Descriptors	56
2.1.17 Security and Collaborative CATM	57
2.1.18 The Way Forward	58
2.2 REFERENCES	59
 CHAPTER 3 - MODELLING CATM	 62
3.1 CONTEMPORARY ATC	62
3.2 THE PROPOSED AIR TRAFFIC SYSTEM MODEL	64
3.2.1 The Agent Model of CATM	65
3.2.2 The Flight Agent Lifecycle	65
3.2.3 ATS Top Structure	66
3.3 FLIGHT GENESIS – THE 4D BUSINESS TRAJECTORY	72
3.3.1 The Strategic Offline Phase	73
3.3.2 Long-Term Sequential Coupling of Constraints	75
3.3.3 The Tactical Online Phase	78
3.4 MODELLING FOR CATM SIMULATION	80
3.5 ATMOSPHERIC MODELLING FOR ATM	81
3.5.1 The International Standard Atmosphere	81
3.6 AIRCRAFT MODELLING FOR ATM	84
3.6.1 Coordinate-System and Other Definitions	84
3.6.2 The EUROCONTROL Base of Aircraft Data, (BADA)	85
3.6.3 A Non-Holonomic, Kinematic Model for a Fixed-Wing Aircraft	86
3.6.4 3D Point Mass Model for a Fixed Wing Aircraft	87
3.6.5 4D Point Mass Model for a Fixed Wing Aircraft	96
3.6.6 The Virtual Time Parameter (τ)	97
3.6.7 A Discrete Time Model	99
3.6.8 Discrete Aspects of a Fixed-Wing Aircraft Model	101
3.6.9 Model Validation	101

3.7	AIRSPACE MODELLING FOR CATM	103
3.7.1	Classified Airspace	103
3.7.2	Special Zone Constraints	104
3.7.3	Obstacle Modelling	105
3.7.4	Aerodromes	106
3.8	TRAFFIC MODELLING FOR ATM	107
3.8.1	Intra-Continental 24-Hour Flight Data Samples	107
3.8.2	A Probabilistic Traffic Model	111
3.8.3	Traffic Density	112
3.9	WEATHER MODELLING FOR ATM.....	114
3.9.1	CFD Weather Simulation.....	114
3.9.2	Synthetic Wind Patterns.....	116
3.9.3	Generating Clouds and Weather cells.....	116
3.9.4	A Self-excited Circular Weather Domain.....	117
3.10	ENVIRONMENTAL IMPACT MODELLING FOR ATM.....	118
3.10.1	Carbon Footprint and CO ₂	118
3.10.2	Noise Mitigation.....	119
3.10.3	Other Pollutants and ATM	119
3.11	AIRPORT -OPERATIONS MODELLING FOR ATM	121
3.12	REFERENCES	122
CHAPTER 4 - COMPUTATIONAL INTELLIGENCE IN CATM.....		124
4.1	CATM PROBLEM FORMULATION	124
4.1.1	Single Flight Problem	124
4.1.2	Combined Multiple Flight Problem	126
4.1.3	Receding Horizon Problem.....	127
4.1.4	The CATM Architecture.....	129
4.1.5	Searching the Control Space vs Output Space.....	131
4.1.6	Dynamic Inversion and Differential flatness	131
4.2	CATM: THE MATHEMATICAL NATURE OF THE SEARCH SPACE	132
4.2.1	The Global Optimisation Problem (Non-Convexity)	132
4.2.2	The Local Optimisation Problem (Local-Convexity).....	133
4.2.3	The Disconnected Feasible Search Space Problem	134
4.2.4	The Dynamic Optimisation Problem	135
4.2.5	The Time-Linked Dynamic Optimisation Problem	136
4.2.6	The Constrained Optimisation Problem.....	136
4.2.7	The Multi-objective Optimisation Problem	138
4.2.8	The System Wide Objective Function	139
4.2.9	The Non-Separable High Dimensionality Optimisation Problem	140
4.2.10	Flight Interaction with Finite Resolution Trajectories	145
4.3	GLOBAL OPTIMISATION PHASE	148
4.3.1	Search Techniques	148
4.4	DYNAMIC PROGRAMMING.....	149
4.4.1	Bellman's Principle of Optimality	149
4.4.2	Structured Dynamic Programming	150
4.4.3	Iterative Dynamic Routing.....	151
4.4.4	Counteracting Collapse	152
4.4.5	Multi Aircraft Approaches with IDR.....	154

4.5	EVOLUTIONARY ALGORITHMS	156
4.5.1	Genetic Algorithms and Evolutionary Programming	156
4.5.2	The Fitness Function	156
4.5.3	Chromosome Encoding	156
4.5.4	Selection	158
4.5.5	Crossover.....	160
4.5.6	Mutation	161
4.5.7	Speciation	162
4.5.8	Coevolution	162
4.5.9	Clustering	163
4.5.10	Diversity and Dynamic Behaviour of GAs	165
4.5.11	Crossover Scheme Effects.....	166
4.5.12	Population Size Effects	167
4.5.13	Elitism and Idealism effects	168
4.6	SWARM INTELLIGENCE.....	169
4.6.1	Particle Swarm Optimization (PSO).....	171
4.6.2	Stability and PSO Constriction	172
4.6.3	Ideal Attractors (GPSO).....	173
4.6.4	Particle Initialisation	174
4.6.5	Upgrading PSO for Dynamic Behaviour (GPSO-D).....	174
4.6.6	Upgrading PSO for High dimensionality (CCGPSO-D)	175
4.6.7	Fine Grained Clustering (C-CCGPSO-D)	175
4.6.8	Multi-Swarms, Stagnation and Population Diversity Collapse	176
4.6.9	A Hybrid: Iterative Dynamic Evolutionary Programming	176
4.7	HIERARCHICAL OPTIMISATION FOR CATM	177
4.7.1	Meta-Heuristics: Exploration and Exploitation	177
4.7.2	Cascaded Optimisation	177
4.7.3	Nested Optimization	178
4.7.4	Cooperative Local and Global Optimization	178
4.8	LOCAL OPTIMISATION PHASE.....	179
4.8.1	A Simple Point Mass Model	179
4.8.2	Direct Collocation	180
4.9	IN SUMMARY	182
4.10	REFERENCES	183

CHAPTER 5 - SIMULATION, RESULTS AND DISCUSSION	188
5.1 THE SIMULATION PARADIGM	188
5.1.1 Scaling-up to the Continental CATM Problem	189
5.1.2 Multi -Threaded Approach	189
5.1.3 The Shared Almanac	190
5.1.4 Transmission Filtering	190
5.1.5 Software Architecture	190
5.2 CONSTRUCTION OF A CATM SIMULATION RIG.....	191
5.3 CATM: PSO C++ IMPLEMENTATION.....	192
5.3.1 Fast Random Number Generators.....	193
5.3.2 Single Instruction Multiple Data (SIMD) Acceleration.....	194
5.4 CATM: C-CCGPSO-D SIMULATION RESULTS.....	195
5.5 QUALITATIVE RESULTS	196
5.5.1 Cooperative Coevolution in Pathological Cases.....	196
5.5.2 Visualisation of Swarm Collapse.....	197
5.5.3 Visualisation of Convergence in large scale traffic scenarios	198
5.5.4 Visualisation of Convergence with obstacle constraints	199
5.5.5 Visualisation of Dynamic Optimisation.....	200
5.6 QUANTITATIVE RESULTS.....	203
5.6.1 Demonstrating Scalability.....	203
5.6.2 Communication Overheads.....	205
5.6.3 The Effect of Transmission filtering.....	206
5.6.4 Clustering Efficacy	206
5.6.5 Swarm Size effects.....	208
5.6.6 Constriction effects	210
5.6.7 Delay Distribution.....	211
5.7 REFERENCES	215
 CHAPTER 6 - CONCLUSIONS AND FUTURE WORK	 216
6.1 SUMMARY	216
6.2 CONCLUSIONS	217
6.2.1 Key Contributions	218
6.3 FUTURE WORK.....	219

LIST OF FIGURES

Fig 1.1.1: Structured Airspace over Central Mediterranean, and Central Europe	10
Fig 1.1.2: Eurocontrol: ATM accounts for a high proportion of Flight Delay.....	10
Fig 1.1.3: Current and Projected Delay Distribution due to Increased Congestion	11
Fig 1.1.4: Growth Trend Spanning 65 years in Aviation Industry (Source: ICAO)	12
Fig 1.4.1: Layering of Performance Concepts.....	18
Fig 1.5.1: High update rate leads to much better prediction	23
Fig 1.6.1: An Underlying Convergence?	25
Fig 1.7.1: The Staged Evolution of CATM.....	28
Fig 1.8.1: Top Level Architecture of CATM.....	33
Fig 2.1.1: Nested Optimization - Best of both worlds	44
Fig 2.1.2: The Continuous Nature of CATM.....	45
Fig 2.1.3: Distributed Processing = Grid-Avionics.....	48
Fig 2.1.4: The Historical Exponential Growth in Computing Performance	50
Fig 2.1.5: Triple Rank Heterogeneous Computing	51
Fig 2.1.6: Overlapping Clusters in CATM.....	52
Fig 2.1.7: The Optically and Radio Networked Airspace	54
Fig 2.1.8: NURBS and polytopes negotiating obstacles	57
Fig 3.1.1: Model of the Current Air Traffic System.....	63
Fig 3.2.1: The CATM System Conveyor Belt, The Flights Life Cycle	64
Fig 3.3.1: Concurrent and Sequential Flight Dependencies	76
Fig 3.3.2: Diurnal Cycle in Traffic Density and Constraint Propagation.....	77
Fig 3.5.1: The International Standard Atmosphere (ISA)	83
Fig 3.6.1: An aircraft shown flying northwards with Earth-referenced attitude	84
Fig 3.6.2: Aircraft forces during steady level flight	87
Fig 3.6.3: The balance of forces during a turn	88
Fig 3.6.4: The balance of forces during ascent	89
Fig 3.6.5: A virtual time parameter ' τ '	97
Fig 3.6.6: Typical Airliner State Transition Diagram	101
Fig 3.6.7: Take-off and Initial Climb	101
Fig 3.6.8: Take-off, Climb, and Cruise	102
Fig 3.6.9: With Holding Patterns	102

Fig 3.6.10: Take-off, Climb, Hold, Cruise, Hold, Descend, Approach, Land.....	102
Fig 3.7.1: Traffic Restrictions in European Airspace	104
Fig 3.7.2: Location of Europe’s Operational Airports	106
Fig 3.8.1: Europe – 24 hours of flights above 30,000 feet (Source: EuroControl)	107
Fig 3.8.2: Europe’s Busiest Airports: Circle size indicates aircraft movements	108
Fig 3.8.3: Flight Distribution by Aircraft Type	108
Fig 3.8.4: Diurnal Cycles in European Traffic, and Changes in Efficiency	110
Fig 3.8.5: Seasonal Variation in European IFR Traffic (Source: Eurocontrol)	111
Fig 3.8.6: Most heavily used interconnections from Europe’s busiest airports	111
Fig 3.8.7: Roulette Selection	112
Fig 3.8.8: European Traffic Density by location and time:	113
Fig 3.8.9: Density at 10:58am	113
Fig 3.9.1: 2D Wind vector field generated with random distributed forces	116
Fig 3.9.2: Cloud Advection in the Weather Simulator	116
Fig 3.9.3: Icelandic Volcanic Dust Cloud Advection in April 2010 (Eumetsat)	117
Fig 3.10.1: Bridging the aviation CO ₂ emissions Gap	118
Fig 3.10.2: Sample INM noise contours generated with the FAA application	119
Fig 3.10.3: Aircraft Emissions and their effect on climate change	120
Fig 4.1.1: The CATM Receding Horizon of $N_F = (N_{FL} + N_{FS} + N_{FA} + 1)$ Flights	127
Fig 4.1.2: Receding Horizon Control for a Single Flight, with Horizon = K_H	128
Fig 4.1.3: Receding Horizon Control for entire ATS, with Horizon = N_F	129
Fig 4.1.4: The CATM System Hierarchy within a Single Aircraft	130
Fig 4.2.1: Trajectories Trapped in Local Minima	132
Fig 4.2.2: Some locally optimal trajectories through a fixed obstacle field	133
Fig 4.2.3: Local Convexity assists convergence	134
Fig 4.2.4: Disconnected Trajectory Search Spaces	135
Fig 4.2.5: Dynamic Changes to Local Convexity	136
Fig 4.2.6: A Typical Bi-Objective Pareto Front	138
Fig 4.2.7: Rubidium Standard	143
Fig 4.2.8: Inter Sample-Point Conflict	146
Fig 4.2.9: Worst case scenario $\theta \rightarrow 180^\circ$	146
Fig 4.2.10: Octahedral and Spherical Buffer Zones Surrounding Aircraft.....	147
Fig 4.3.1: Algorithms considered for CATM global search	148
Fig 4.4.1: Forward Induction Dynamic Programming	150

Fig 4.4.2: Structured Output Search Space with Obstacles	150
Fig 4.4.3: First Pass of a structured ITR problem	151
Fig 4.4.4: Second pass: Search Space Contracts by a constant factor γ	151
Fig 4.4.5: IDR Path Planning progress in 2D with static obstacle	152
Fig 4.4.6: Countering Collapse in IDR	153
Fig 4.4.7: IDR Path Planning, in 2D with a dynamic obstacle	153
Fig 4.4.8: ROIDR Path Planning progress with 3 or 5 interacting flights in 2D	154
Fig 4.5.1: Cooperative Fitness Function	163
Fig 4.5.2: Demonstration of Clustering Effectiveness with Dynamic Behaviour	164
Fig 4.5.3: GA Optimisation of Pathological Non-Convex case with Obstacles	166
Fig 4.5.4: The effect of Genetic Pool Size on convergence Time and Outcome	167
Fig 4.5.5: The effect of idealism	168
Fig 4.6.1: Argentine Ant Experiment	169
Fig 4.6.2: Damped SHM in PSO	171
Fig 4.7.1: Cascaded Optimization	177
Fig 4.7.2: Embedded Local Optimization	178
Fig 4.8.1: Trajectory of a Newtonian Point Mass	179
Fig 4.8.2: Locally Optimized Trajectory of a Newtonian Point Mass	181
Fig 5.1.1: The CATM Simulation System Architecture	188
Fig 5.1.2: The CATM System Hierarchy within a Single Flight Cluster	189
Fig 5.2.1: CATM Simulation Rig Configuration	191
Fig 5.2.2: CATM Simulation Rig	191
Fig 5.3.1: SSE4 Code excerpt showing the MWC1616 Algorithm	194
Fig 5.5.1: Convergence of CCGPSO with a 14-way choke-point scenario	196
Fig 5.5.2: Convergence Charts of CCGPSO with a 14-way choke-point scenario	197
Fig 5.5.3: Visualization of Swarm collapse a 14-way choke-point scenario	197
Fig 5.5.4: After 2,000 iterations	198
Fig 5.5.4: After 40,000 iterations	198
Fig 5.5.4: After 1,000,000 Iterations	198
Fig 5.5.5: After 5,000,000 Iterations	200
Fig 5.5.6: Effects of an object collision with a dynamically converged solution	200
Fig 5.5.7: Transient caused by an object collision during convergence	201
Fig 5.5.8: Transient due to a sudden object collision showing dynamic behaviour	202
Fig 5.6.1: Performance of C-CCGPSO-D with Variable Problem Sizes	204

Fig 5.6.2: Reliable Convergence with Variable Problem Sizes	205
Fig 5.6.3: Performance of C-CCGPSO-D with Variable Cluster Sizes	206
Fig.5.6.4: Reliable Convergence with Variable Cluster Sizes	207
Fig.5.6.5: Performance of C-CCGPSO-D with Variable Swarm Sizes	209
Fig.5.6.6: Reliable Convergence with Variable Swarm Sizes	209
Fig.5.6.7: Performance of C-CCGPSO-D with Variable Constriction	210
Fig.5.6.8: Delay distribution of flights after 10,000 iterations	211
Fig 5.6.9: Delay distribution of flights after 100,000 iterations	212
Fig.5.6.10: Delay distribution of flights after 1,000,000 iterations	212
Fig.5.6.11: Delay distribution of flights after 5,000,000 iterations, (20 trials)	213
Fig.5.6.12: Delay distribution in the June 2009 flight sample analysed	213
Fig.5.6.13: Eurocontrol's Current and Projected Delay Distribution	214

LIST OF TABLES

Table 1.6.1: Tools developed to assist human pilots and controllers	24
Table 3.5.1: Atmospheric Layers as defined by temperature gradients	82
Table 3.6.1: BADA Drag Coefficient Definitions for Airbus A300-B4-622	90
Table 3.6.2: Interpretation of the BADA Energy share factor	95
Table 3.7.1: ICAO Classification of Airspace	103
Table 3.8.1: Flights Breakdown in 24h Sample	107
Table 3.8.2: Interconnection Matrix between Europe's 20 Busiest Airports	109
Table 3.8.3: Stratified Efficiency for All flights > 250km	110
Table 3.8.4: European traffic density over various areas	113
Table 4.5.1: Crossover Techniques Applicable to Real Coded Genetic Algorithms ...	160
Table 4.5.2: Mutation Techniques Applicable to Real Coded Genetic Algorithms.....	161
Table 4.5.3: Mutation Schedule	161
Table 4.5.4: Effect of Multipoint Crossover and multiple sexes	167
Table 4.5.5: Effects of Elitism and Idealism	168
Table 4.6.1: Constricted PSO Attraction Coefficients	173
Table 4.9.1: Comparison between some Global Search Techniques	182
Table 5.2.1: CATM Simulation Rig Configuration	192
Table 5.6.1: Scalability Tests over 8 octaves	203
Table 5.6.2: Communication Overheads	205
Table 5.6.3: The Effect of Swarm Size	208
Table 5.6.4: Swarm Constriction Effects	210
Table 5.6.5: Comparison and fit of Delay Distribution	213

LIST OF ABBREVIATIONS

2D	-	Bi-dimensional
3D	-	Tri-dimensional
4D	-	Four-dimensional
4D-BT	-	Four-dimensional Business Trajectories
ABC	-	Artificial Bee Colony
ACARS	-	Aircraft Communication Addressing and Reporting System
ACO	-	Ant Colony Optimisation
ADS-B	-	Automatic Dependant Surveillance-Broadcast
AFAP	-	As Far As Practicable
AI	-	Artificial Intelligence
AIS	-	Artificial Immune Systems
AMAN	-	Arrival Manager
ANSP	-	Air Navigation Service Provider
ARA	-	Air-to-Air Refuelling Areas
ARTCC	-	Air Route Traffic Control Centres
ASAS	-	Air Separation Assurance System
ASIC	-	Application Specific Integrated Circuit
ATA	-	Aerial Tactical Areas
ATC	-	Air Traffic Control
ATCo	-	Air Traffic Controller
ATCT	-	Aviation Training Centre
ATM	-	Air Traffic Management
ATFM	-	Air Traffic Flow Management
ATFCM	-	Air Traffic Flow and Capacity Management
ATS	-	Air Traffic System
ATSP	-	Air Traffic Service Provider
ATZ	-	Aerodrome Traffic Zone
AWDAP	-	Airborne Wind-shear Detection and Avoidance Program
BA	-	Bat Algorithm
BADA	-	Base of Aircraft Data
BCGA	-	Binary Coded Genetic Algorithms
CAMU	-	Central Airspace Management Unit
CARATS	-	Collaborative Actions for Renovation of Air Traffic Systems (Japan)
CATM	-	Computational Air Traffic Management
CC	-	Cooperative Co-evolution
C-CCPSO	-	Clustered Cooperative Coevolution Particle Swarm Optimisation
C-CCPSO-D	-	Clustered Cooperative Coevolution PSO with Dynamic behaviour
CCGA	-	Cooperative Co-evolutionary Genetic Algorithm
CCPSO	-	Cooperative Coevolution Particle Swarm Optimisation
CCGPSO-D	-	Cooperatively Co-evolving Guided PSO for Dynamic problems
C-CCGPSO-D	-	Clustered Cooperatively Co-evolving GPSO for Dynamic problems
CD	-	Conflict Detection
CDA	-	Continuous Descent Approach
CD&R	-	Conflict Detection & Resolution

CFD	-	Computational Fluid Dynamics
CFMU	-	Central Flow Management Unit
CM1	-	Connection Machine 1
CNS	-	Communication, Navigation and Surveillance
CPDLC	-	Controller-Pilot Data Link Communications
CPU	-	Central Processing Unit
CR	-	Conflict Resolution
CRDA	-	Converging Runway Display Aid
CSCC	-	CATM Super-Computing Centre
CUDA	-	Compute Unified Device Architecture
DMAN	-	Departure Manager
DoF	-	Degree of Freedom
DP	-	Dynamic Programming
DTM	-	Deterministic Turing computing Machine
EA	-	Evolutionary Algorithms
ECC	-	Error Control Code
EDA	-	Efficient / En-route Descent Advisor
EP	-	Evolutionary Programming
ETA	-	Estimated Time of Arrival
FA	-	Firefly Algorithm
FAA	-	Federal Aviation Administration
FADEC	-	Full Authority Digital Engine Control System
FANS	-	Future Air Navigation System
FDR	-	Flight Data Recorder
FIR	-	Flight Information Region
FIS	-	Flight Information Service
FPGA	-	Field Programmable Gate Array
FMS	-	Flight Management System
FSO	-	Free Space Optics
GA	-	Genetic Algorithm
GIS	-	Geographical Information System
GMT	-	Greenwich Mean Time
GPGPU	-	General Purpose Graphics Processing Unit
GPS	-	Global Positioning System
GPSO	-	Guided Particle Swarm Optimisation
GPSO-D	-	Guided Particle Swarm Optimisation for dynamic problems
GPU	-	Graphics Processing Unit
GSO	-	Glow-worm Swarm Optimization
GWPS	-	Ground Proximity Warning System
HLH	-	High Level relay Hybrids
HMR	-	Helicopters Main Routes
HPZ	-	Helicopters Protected Zones
HTH	-	High-level Teamwork Hybrid
IATA	-	International Air Transport Association
ICAO	-	International Civil Aviation Organization
ICAS	-	International Council of the Aeronautical Sciences Congress
IDP	-	Iterative Dynamic Programming
IDR	-	Iterative Dynamic Routing

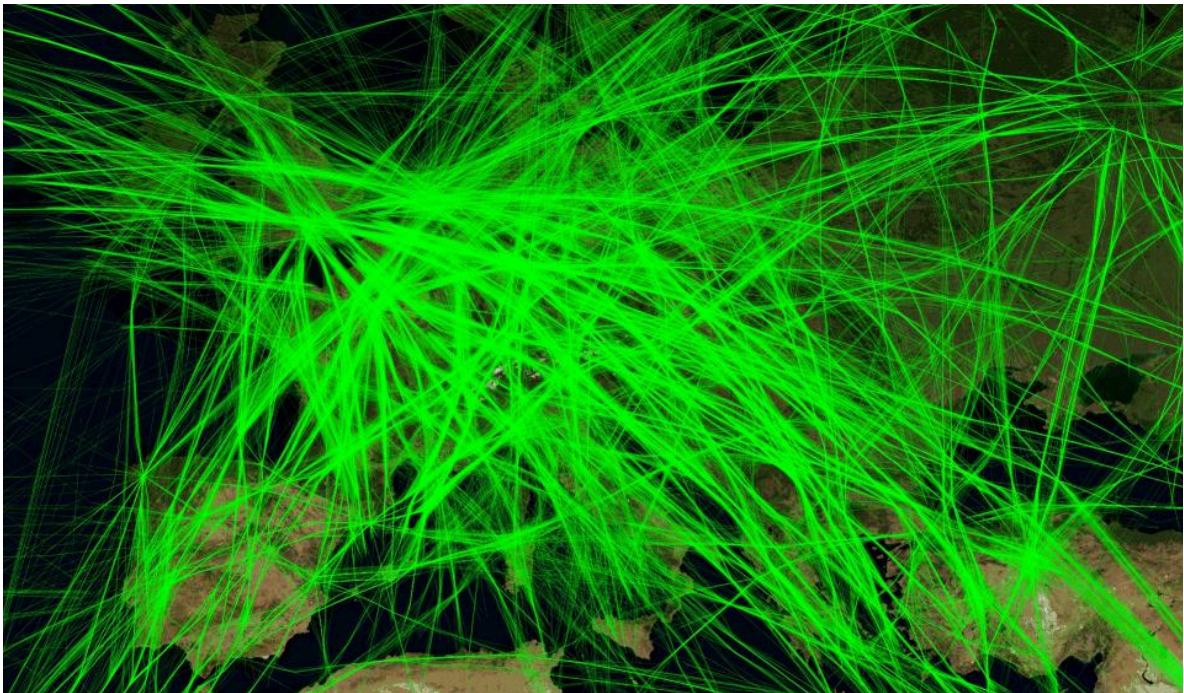
IFR	-	Instrument Flying Rules
IMA	-	Integrated Modular Architecture
INM	-	Integrated Noise Model
ISA	-	International Standard Atmosphere
ISBT	-	Initial Shared Business Trajectory
ISO	-	International Standards Organisation
KB2D	-	A 3D distributed CD&R algorithm
KB3D	-	A 2D distributed CD&R algorithm
KE	-	Kinetic Energy
KKT	-	Karush-Kuhn-Tucker
KPA	-	Key Performance Area
LCG	-	Linear Congruential Generators
LE	-	Lost Energy
LTH	-	Low-level Teamwork Hybrid
LLCD	-	Lunar Laser Communications Demonstrator
MAC	-	Media Access Control
MATZ	-	Military Aerodrome Traffic Zone
MIMO	-	Multiple-Input Multiple-Output
MONA	-	Monitoring Aids
MOOP	-	Multiple-Objective Optimisation Problem
MRP	-	Multi-Resolution Programming
MSBT	-	Mature Shared Business Trajectory
MSAW	-	Minimum Safe Altitude Warning
MTA	-	Military Training Areas
MTCD	-	Medium Term Collision Detection
MTOW	-	Maximum Takeoff Weight
MWC	-	Multiply With Carry
NAVAIDS	-	Navigation Aids
NE	-	New Energy
NEU	-	North East Up Convention
NextGen	-	Next Generation Air Transportation System (USA)
NLP	-	Non-Linear Programming
NOTAM	-	Notice to Airmen / Notification to Airmen
NP	-	Non-deterministic in Polynomial time
NSGA	-	Non-Dominated Search Genetic Algorithm (I/II)
NTM	-	Non-deterministic Turing Machine
NURB	-	Non-Uniform Rational B-Spline
OAM	-	Orbital Amplitude Modulation
ODE	-	Ordinary Differential Equation
OSI	-	Open Systems Interconnection
P2P	-	Peer-to-Peer
PAV	-	Personal Air Vehicles
PE	-	Potential Energy
PMM	-	Point Mass Model
PMP	-	Pontryagin's Minimum Principle
PPM	-	Pulse Position Modulation
PRAFSO	-	Parallel Relay Assisted Free Space Optics
PRNG	-	Pseudorandom Random Number Generator

PSO	-	Particle Swarm Optimisation
QAM	-	Quadrature Amplitude Modulation
R&D	-	Research & Development
RBT	-	Reference Business Trajectory
RCAF	-	Runway Collision Avoidance Function
RCGA	-	Real Coded Genetic Algorithms
RFD	-	River Formation Dynamics
RNAV	-	Area Navigation, (also Random Navigation)
ROIDR	-	Reduced Order Iterative Dynamic Routing
RPM	-	Revenue Passenger Miles
SES	-	Single European Sky
SESAR	-	Single European Sky ATM Research (EU)
SHM	-	Simple Harmonic Motion
SIMD	-	Single Instruction Multiple Data
SMAN	-	Surface Manager
SPO	-	Strategic Phase Optimisation
SPSO	-	Standard Particle Swarm Optimisation
SQP	-	Sequential Quadratic Programming
STCA	-	Short Term Collision Alert
SWIM	-	System Wide Information Management
SYSCO	-	System Supported Coordination
TAS	-	True Airspeed
TCAS	-	Traffic alerting tool and Collision Avoidance System
TMA	-	Terminal Area / Terminal Manoeuvring Area
TMI	-	Traffic Management Initiatives
TOPM	-	Take-Off Performance Monitoring
TPO	-	Tactical Phase Optimisation
TRACons	-	Terminal Radar Approach Control
TW	-	Target Window
UAV	-	Unmanned Air Vehicles
URET	-	User Request Evaluation Tool
UTC	-	Coordinated Universal Time
VDLM2	-	Very High Frequency Digital Link - Mode 2
VFR	-	Visual Flying Rules
VHF	-	Very High Frequency (Radio Communications)
VTOL	-	Vertical Take-off Aircraft
WPE	-	[SESAR] Work-Package-E program

Preface

Computational Air Traffic Management (CATM) traces its roots to an idea born out of a pressing need. This is a long term study in fully airborne, decentralised, autonomy for ATM and it all started with a focussed study we undertook on the optimisation of continuous descent approaches (CDA) in the presence of air traffic, to reduce fuel consumption, noise and emissions by gliding aircraft down to destination aerodromes with their engines on low thrust settings. However, it soon became apparent that efforts to optimise this phase of flight in isolation would be largely negated by constraints imposed, on or by, incoming traffic from the enroute phase. It was hoped at first that one could combine and collectively optimise the trailing end of enroute with CDAs. However, the problem arose again and again, until it was clear that any serious attempts at minimizing the system's total fuel consumption would have to look at all phases of flight... of *all* aircraft.

This is essentially the ATM problem, and it highly impinges on most aspects of aviation to the extent that it is no longer possible to sideline this major contributor to air transport inefficiency and cost, when the rest of the industry spends tens of billions of Euros to achieve low single digit improvements elsewhere. ATM is unfortunately a rather suboptimal, labour intensive affair, and yet there exists no concerted, in-built mechanism to collectively optimise air transport. A closer look at the nature of ATM reveals why: Complexity and Legacy.



Europe – 24 hours of flights above 30,000 feet (Source: EuroControl)

I. THE RESEARCH CONTEXT

The majority of research in ATM/ATC is carried out by the ATC supply chain, and national / supranational organisations of air traffic controllers (ATCo), (eg: Eurcontrol, FAA, NATS etc.) and is by its very nature quite close to the current market. On the other hand, it is the author's perception that a large proportion of unbridled *fundamental research* in the ATM community is conducted by "visiting" researchers (such as mathematicians, computer scientists and engineers) that happen to temporarily wander-off from adjacent fields to tackle some aspect of ATM... and then leave.

This distinction in approach is manifested in the trichotomy of research products that seems to exist. This influx of researchers is objectively beneficial to the community and cross-fertilizes ideas. They enter the field without many pre-conceived ideas and are able to look at the problem more broadly. For many years, the absence of a perennial critical mass of dedicated fundamental ATM researchers, meant that a steady reservoir of knowledge took long to be created and many worthwhile ideas grew old on the shelf, without ever being brought to fruition. This seems to be changing. The air traffic research effort that has taken place during the past couple of decades can therefore be broadly classified into three clear categories:

Type I Research: The first type works within the framework of the current ATM paradigm. It therefore feeds the relentless demand for new tools to assist air navigation service providers (ANSPs) with running their day-to-day task of ensuring the harmonious flow and separation of aircraft. These tools automate specific portions of the job. Some assist controllers with maximizing productivity, while others attempt to reduce operating costs which may coincide with lessening the environmental impact. Still other tools are intended as safety-nets and are developed in response to accidents or observed failure modes of the system. Minimizing the environmental burden is however further down on the ANSP priority list. Thus, in general, these tools aim to mitigate only some of the numerous shortcomings of ATC for just the short to medium term.

This research category includes a long list of technologies, some of which have actually found their way into many cockpits and control towers. Such tools play an important psychological role as they gradually wean-off ANSPs from tactical traffic micro-management while rendering them more acquiescent towards higher levels of automation. In fact, the operational philosophy of such tools has gradually shifted beyond the mere provision of enhanced usability and ergonomics (eg: electronic flight strips) and is now moving to supplant some of the core decision-making that was traditionally the prerogative of the human controller (eg: sequencing). The current deployments of the AMAN and DMAN are good examples of such a trend.

However, by their very nature, such technologies remain stop-gap solutions for extending the lifespan of a human-centric ATC system which traces its roots to the late-1920s. Their downside is that such a piece-wise addition of new layers of automation tends to introduce new complexity, unforeseen interactions and new system states. Some information ends up lost in translation with the result that the ultimate human decision maker is deprived from full situational awareness, which when coupled with a poor understanding of the underlying automation, can generate new failure modes and confuses issues of liability in the case of accidents (refer to EU project ALIAS).

Type II Research: It has been generally accepted that the ATM system is well past its prime and is long overdue for a major overhaul. To this effect, type II research has been established. It is best exemplified by Europe's SESAR, USA's NextGen and Japan's CARATS. Over two decades, these large public-private partnerships plan to inject the aging infrastructure with new life, add capacity and reduce costs.

The hallmark feature of such research is the process of unification of concepts through higher automation to reduce the level of fragmentation that currently exists. System Wide Information Management (SWIM) and 4D Trajectory-based operations are excellent examples. The adoption of better streamlined but far reaching, broadly applicable concepts multiplies ATM effectiveness while dividing its complexity.

Although, it was initially hoped that these projects would have fundamentally challenged the status quo, in hindsight they seem rather watered-down, and turn out to be still producing significant quantities of Type-I research. A considerable number of highly-specific crew-support tools are being developed around the incumbent human-centric paradigm. For instance SESAR proposes dynamic sectorisation, time based separation, and virtual control towers as an incremental means of adding capacity and reducing costs. These add complexity. The oft-declared a-priori objective of keeping the human ATCo in the loop is hampering progress. It partly explains why these projects stop short of any truly radical changes to the principles of ATM/ATC.

Type III Research: On the other hand, at the far end of the scale, there exists a third type of research activity which involves rethinking the *very foundations* of the ATM problem and some of this is taking place under SESAR's Work-Package E. This type of research looks beyond current technological, social and political constraints and considers instead various "what-if" scenarios that strategically overlook individual non-fundamental challenges to see past the technological horizon. It is often a matter of time for many of these challenges to be addressed, so it appears wise to study in advance the possibilities that could be unlocked in that event. It is also possible that what appears to be a stumbling block today, might become largely irrelevant in a different context. This approach not only broadens the research space, but provides a renewed impetus to addressing any extant challenges, once the rewards are so clearly described.

This kind of research will complete the unification process initiated by SESAR and NextGen. The objective here is to devise and verify new overarching operational paradigms that are intrinsically more robust, notwithstanding higher traffic densities, whilst guaranteeing equivalent or improved levels of safety, fewer delays, less fuel consumption and environmental impact – usually through a far higher degree of autonomy and near total reliance on technology. The intention is to develop and evaluate a cache of viable alternative paradigms that could eventually displace the current system, which is appearing increasingly inadequate to handle much higher traffic densities, as envisaged over the longer term.

However, radical as it may be, the outcome of any Type III research cannot overlook the realities of mixed equipage, operational constraints and the need for a smooth transition between old and new paradigms, but these matters will become research areas in their own right and are given somewhat subsidiary importance in relation to the initial development of idealised target concepts.

II. THESIS SCOPE, OBJECTIVES AND LIMITATIONS

This thesis aims to address Type III research questions. It appears that too little is being done to rethink ATM at its very roots. We will therefore attempt to reconsider the ATM problem in the light of all available modern engineering and mathematical tools in order to establish whether a coherent concept can be put together and be realistically implemented using current or expected forthcoming technology. We *will not* however, restrict ourselves to *any* of the limitations imposed by the current ATM paradigm.

The project is exploratory and will therefore not attempt to develop complete finished systems. That is a role that can be filled with diligent engineering spanning many years. We will rather focus on the feasibility aspects of the problem by looking at the major show-stoppers. The most important thing at this stage, is to establish a way forward for all the primary, critical subsystems. The proposed solution can serve as guide to steer future ATM research and development efforts.

III. THESIS OUTLINE AND CONTRIBUTIONS

This dissertation follows a precise format that first establishes the research questions, then the methodology for addressing them, followed by some experimental data to substantiate the chosen methodology. It finally provides a list of contributions and recommendations for future work.

A. Research Questions

The key research question is of course whether a fully autonomous ATM system will ever be feasible. The corollary of that question is the method with which it could be implemented. The final question is how to plan the changeover between the current ATM paradigm and the proposed new system. Throughout this dissertation, the first two questions will be addressed while the last question will be only briefly discussed.

Of course, attempting to answer these basic questions will raise many others of a more technical nature, and therefore, more specifically, the following open questions will be also addressed and/or discussed:

- Is there a science underpinning ATM?
- What structure does a practical autonomous air transport system take?
- ATM is a large non-convex optimisation problem, how can it be handled in real-time?
- Is the ATM problem computationally tractable?
- If tractable, what algorithms and conventional computing platforms can be used?
- Can sufficient computational resources be afforded in an airborne system?
- Is an airborne CATM system more vulnerable to failures?
- Is an airborne distributed trajectory optimisation algorithm feasible?
- If yes, how will the many aircraft converge to one solution for the airspace?
- How will the multitude of autonomous aircraft communicate?
- What language of interchange will be used throughout the system?

- How can the system be made to adapt to weather and other disturbances?
- How can optimisation speed be improved?
- How can the system be simulated offline?

B. Methodology

The problem being considered is way too large for implementation by any single person. Therefore, the methodology appropriate for this kind of research project is to first understand the current ATM system, and then define the fundamentals, and finally move on to suggest and test some partial solutions. We therefore embark on a project that will:

- 1) Explore the nature of the problem.
- 2) Identify the main requirements.
- 3) Study and establish its defining features.
- 4) Give the problem some formal structure.
- 5) Propose a conceptual solution.
- 6) Propose a plan for implementation and simulation of CATM systems.
- 7) Construct suitable place-holder models for the various sub-systems.
- 8) Identify the most challenging sub-systems and sub-problems.
- 9) Identify and study the most promising avenues for a breakthrough.
- 10) Implement a number of candidate solutions with improvements.
- 11) Test the most promising implementation with repeatable test scenarios.
- 12) Compare results with current ATM systems.
- 13) Draw conclusions on the feasibility of Computational Air Traffic Management.

C. Contributions

There are a number of novel contributions of various magnitudes that are described in this dissertation. The most important of these are highly conceptual. SESAR and NextGen began the process of merging various pieces of automation into more powerful tools to assist the human pilot and controller. CATM proposes to take this process to the limit and considers the total unification of all blocks of automation into a single powerful concept that elegantly deals with the entire problem holistically and autonomously, thereby obviating the need of humans in the loop. However, in addition to the primary contribution to the ATM domain (which is of a conceptual nature), a number of other secondary contributions are offered as supporting evidence in favour of the central concepts. The following specific contributions to the fields of distributed computational intelligence, optimisation science and ATM were made:

- 1) A fundamental rethink of the broad ATM problem from basic requirements.
- 2) Defined a novel *unifying* concept for the entire Air Traffic System (ATS).
 - a. Replaces fragmented automation with unified overarching autonomy.
 - b. Network-Centric Communications, Navigation and Surveillance (CNS).
 - c. Network-Centric real-time Air Space Management (ASM).

- 3) Proposed a new framework and architecture for making ATM Airborne.
 - a. Fully distributed, fully autonomous Air Traffic Management.
 - b. Grid Avionics was proposed as possible CATM implementation.
 - c. Grid avionics was shown to be a feasible CATM implementation.
- 4) Posed CATM as a collective optimisation problem and explored the implications.
- 5) Studied the relationship between the strategic and tactical elements of CATM.
- 6) Studied the suitability of various classic global optimisation techniques and variations to ATM problem.
- 7) Empirically evaluated a novel multi-resolution Iterative Dynamic Routing (IDR) technique, by extending ideas taken from iterative dynamic programming (IDP).
 - a. Proposed the use of a structured grid for reducing IDR dimensionality
 - b. Developed techniques against grid collapse for good dynamic behaviour
- 8) Studied suitability of various swarm intelligence and evolutionary approaches
- 9) Explored methods for distributing computational intelligence symmetrically among aircraft such that the computing power of the machines remains additive.
- 10) A Coevolutionary version of Particle Swarm Optimisation (PSO) was adapted to cater for dynamic environments using *GBest* and *PBest* online re-evaluation.
- 11) Ideal attractors were introduced as a technique for speeding up Genetic Algorithms and Particle Swarm Optimisation for problems with approximately-known globally optimum solutions. (such is the case in ATM).
- 12) Developed a methodology to address problems of scalability through clustering
- 13) Evaluated the suitability of various complexity reduction techniques, such as continuous optimisation, receding horizon control, B-Spline interpolation etc...
- 14) Empirically demonstrated the linear scalability and unconditional convergence of the proposed algorithm.
- 15) Studied the requirements for an airborne computational hardware platform.
- 16) Implemented and tested a distributed computing hardware and software simulation platform for testing CATM ideas.
- 17) Gave indications on the communications overhead and bandwidth required
- 18) Tested, validated and characterised algorithms on the developed platform.
- 19) Enumerated a number of open questions that merit further exploration.
- 20) Prepared a number of place holder models for atmospheric effects, aircraft, weather, wind, airspace obstacles, traffic, environmental impact and ground delays.

D. Organization of the Dissertation

The arguments in this dissertation are developed over six chapters. The following provides a brief summary on the purpose and objectives attained in each chapter in turn.

Chapter 1 - Introduction

This introduction discusses the problem background and presents a compelling argument for ATM reform. The principles of ATM are outlined together with the idealised behaviour expected from such a system. This chapter proceeds to discuss the historical development of the ideas that were developed into the CATM concept. The science applicable to of ATM is discussed followed by a three stage probable progression towards full CATM automation. The chapter concludes by providing an outline tentative plan for the changeover from conventional ATM to autonomous CATM-based paradigm.

Chapter 2 - Computational Air Traffic Management (CATM)

In this chapter specific detail is given about the CATM concept and its ancillary requirements. Complexity considerations about the ATM problem are discussed together with the benefits of approximate optimisation, decentralisation, nested optimisation and continuous re-convergence. Grid avionics concepts are presented together with essential aspects like fast optical communication, security, redundancy, sensor networks, flight monitoring, fault tolerance and computing platforms. The clustering concept based on the locality principle and sharing of trajectory descriptions are also introduced.

Chapter 3 - Modelling CATM

Chapter three focuses on the modelling aspects of CATM and the system it needs to interact with. These are needed for both operation and simulation. The completely revised Air Traffic System Model is presented in a formally structured manner that can be extended in future to introduce further detail. The process of flight genesis is discussed, starting from planning, inception, through to execution and completion. The chapter concludes by describing a number of system models that are suitable for CATM simulation including: atmospheric models, aircraft models, traffic models, weather models, environmental impact models and ground operations models.

Chapter 4 - Computational Intelligence in CATM

A detailed study of the various computational techniques available to optimise CATM trajectories is discussed, after providing a detailed topological analysis on the exact nature of the ATM problem. Three global optimisers are discussed; one based on dynamic programming and two other evolutionary and swarm intelligence metaheuristics. A simple collocation technique was outlined as a way of performing local search following global search. The chapter finishes by tabulating the salient point regarding some of the various global optimisation techniques available together with their overall suitability for CATM.

Chapter 5 - Simulated Test Scenarios, Results and Discussion

The testing methodology and the implementation aspects of the test setup are discussed in chapter five. Both qualitative and quantitative tests were conducted in order to show symmetric distributed particle swarm optimisation in difficult cases, large scale traffic handling, dynamic optimisation and the handling of obstacles constraints. Scalability was proved empirically as well as the effectiveness of clustering. Communications overheads were quantified with and without transmission filtering. The effects of particle swarm size and constriction were also quantified and the final delay distribution of the converged output was analysed and compared to typical traffic.

Chapter 6 - Conclusions and Further Work

The final Chapter concludes the dissertation by summarising the achievements and by providing answers to the research questions laid out in the preface. Avenues for further research are also outlined.

IV. PUBLICATIONS ARISING FROM THIS WORK

A number of publications arose from this work, one published at a leading IEEE conference, another one accepted for publication in the AESS magazine, another pending review, and two more papers are in preparation for submission to international journals.

A. Published

[1.1] M.A. Azzopardi and J.F.Whidborne, "Computational Air Traffic Management", *Proceedings of the 30th AIAA/IEEE Digital Avionics Systems Conference (DASC2011)*, IEEE/AIAA, Seattle, WA, USA, 2011, Oct 16-20, pp. 1.B.5-1. *Best Paper of session*.

B. In Press

[1.2] M.A. Azzopardi and J.F.Whidborne, "The case for Computational Air Traffic Management", *Aerospace and Electronic Systems Magazine*, IEEE, Seattle, WA, USA, Accepted, 2015

[1.3] M.A. Azzopardi and J.F.Whidborne, "Grid Avionics and CATM", *Aerospace and Electronic Systems Magazine*, IEEE, Seattle, WA, USA, Pending review, 2015.

C. In preparation

[1.4] M.A. Azzopardi and J.F.Whidborne, "Symmetric Distributed PSO for a CATM Grid Avionics System", In preparation for submission to an international journal.

[1.5] M.A. Azzopardi and J.F.Whidborne, "On the linear scalability of Computational Air Traffic Management", In preparation for submission to an international journal.

Chapter 1

Introduction

This chapter proposes the notion of Computational Air Traffic Management (CATM) as the discipline, that is focussed on establishing ways of mathematically posing the ATM problem, that subsequently leads to computationally tractable solutions that can form the basis for an ideal ATM system. The features of an ideal ATM system are first discussed and developed objectively in contrast to current systems. A high level top-down approach is then adopted to define the requirements of such a system, while also looking at the time-proven characteristics that have allowed present-day ATM to maintain its enviable safety record. However, the shortcomings of the current ATM paradigm are also enumerated and discussed and the case is made for a new autonomous traffic management methodology that relies far more on distributed computational intelligence, inter-aircraft communication and surveillance, than present, to holistically address the scalability problems faced by ATM today, while minimizing any negative environmental or societal impact of air transport.

Closer analysis reveals CATM as the natural convergence of the seemingly disparate, air traffic technologies that have been evolving over the past two decades. An inventory of interesting unanswered research questions in the overall CATM picture is gradually drawn-up along the discussion to indicate promising research directions leading towards CATM.

1.1 THE MOTIVATION

Air Traffic Management (ATM) is the all encompassing term for a worldwide distributed system tasked with keeping global air transport organised, interoperable, reliable, efficient, and safe. To this list we could add: secure, cost-effective and environmentally sustainable. ATM's remarkable history dates to the early 20th century [1.1], conceived out of necessity following the popularisation of powered flight during WW1 for military purposes, but it was not until after WW2 that the International Civil Aviation Organization (ICAO) was set up, in 1944 at the Chicago Convention, to organise the advent of commercial aviation. The ATM system was very successful at reaching its primary objectives and as of 2013 boasted of a sterling safety record of less than one hull loss accident in 5 million hours flown on western-built aircraft [1.2]. This should be taken in the context of 5.7 trillion annual revenue passenger miles (RPM) on a fleet of 26,600 airliners making around 100,000 flights *daily*. However, if air traffic continues to increase, as it has for the past 65 years, many things will have to change in order to maintain these levels of safety and reliability. Efficiency and reliability are already starting to falter [1.3] and the environmental impact is worrying [1.4].

1.1.1 The Problem Background

For many decades, structured airspace has been used [1.5]. This discretized the airspace into flight levels and airways between fixed navigation aids, (NAVAIDS) (**Fig. 1.1.1**) respecting the early limitations in navigation instrument accuracy [1.6]. However, it also transformed ATM into a combinatorial problem that makes it easier for human controllers to tackle conflict resolution at short notice. As traffic density grew, recursive sectorisation further divided the problem into manageable portions assigned to individual controllers. This also facilitated some VHF radio channel reuse [1.7].

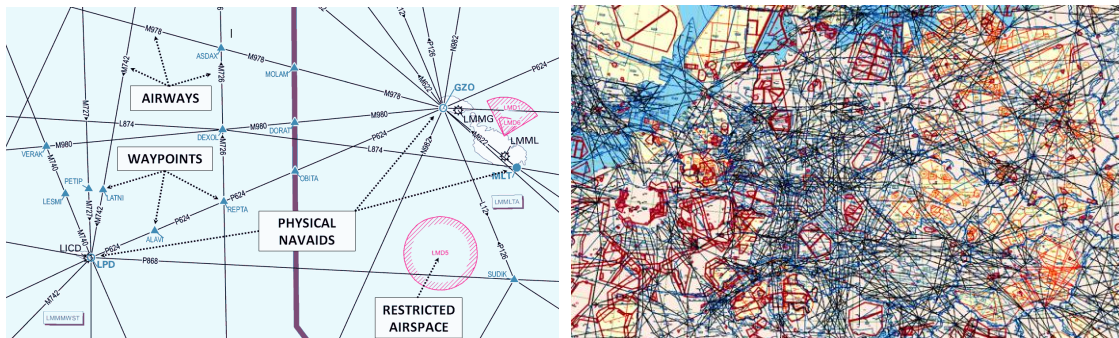


Figure 1.1.1: Structured Airspace over Central Mediterranean, and Central Europe [1.8]

However, subdivision of the airspace cannot go on indefinitely as it adds to the complexity, leads to instability and generates numerous handover requests between sectors, which is bounded by the limited number of non-interfering VHF radio channels that can be accommodated in a given area. When coupled with aircraft separation minima, structured airspace also brought with it a significant restriction in traffic capacity as well as significant elongation of routes. The shift to area navigation (RNAV) [1.9], and more recently, satellite-based RNAV [1.10], eased this problem by exploiting better accuracy of navigation equipment, allowing intermediate ephemeral waypoints to be defined as necessary (**Fig. 1.1.1**). This allowed more direct routes and airways to be established, but it was not long before congestion caught up with these advantages.

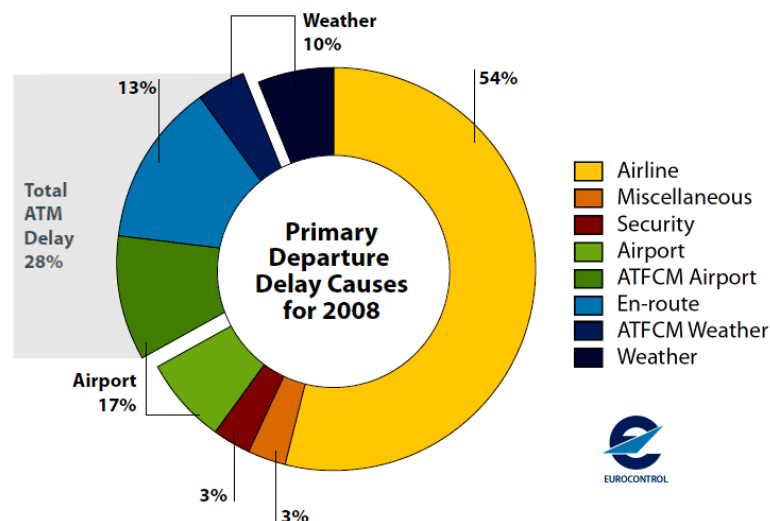


Figure 1.1.2: Measured by Eurocontrol, ATM accounts for a high proportion of Flight Delay [1.11]

The effects of congestion are well known (**Fig 1.1.2**), and typically translate in delays and higher costs for the travelling public [1.3], [1.11]. Efficient wind-optimised (eg: jet-stream) or minimum length great-circle routes have to be forfeited around zones of high traffic density, resulting in higher fuel burn, CO₂ emissions and delays. There is a point, in the not too distant future, where the current ATM paradigm will meet stiff challenges, and it may no longer be possible to incrementally adjust, as we have been doing so far, without significant impact on performance (**Fig 1.1.3**).

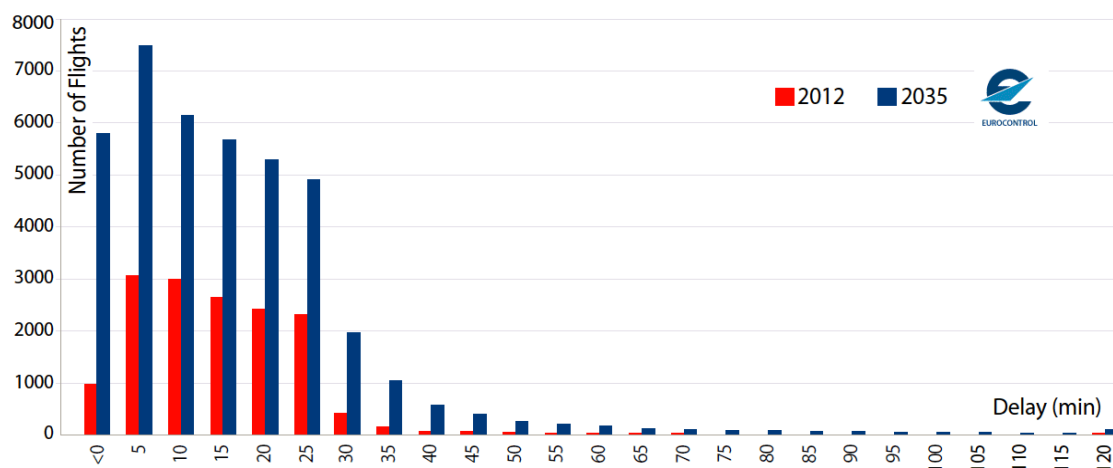


Figure 1.1.3: Current and Projected Delay Distribution due to Increased Congestion [1.3]

Flexibility, efficiency and capacity is ostensibly maximised with the 1995 concept of *free-flight* [1.12]-[1.15], that proposed to do away with centralised separation control. However, studies have shown that reactive conflict detection and resolution (CD&R) strategies, considered necessary for pilot-driven *free-flight*, result in a *domino effect* at high traffic densities, where resolving one traffic conflict creates new conflicts further down the line [1.16]-[1.18], which suggests that more sophisticated alternatives will be needed before the *free-flight* scenario can be taken further.

1.1.2 The Cost of Human ATM

Air Traffic Management is a very expensive enterprise weighing-in on a very cost sensitive industry, that is critically important to the world economy. With 195,000 Air Traffic Controllers (ATCo's) and executives, ATC costs the world around €25 billion annually to operate. It costs €579 in the EU, and €438 in the USA per IFR flight hour [1.19], [1.20]. Then there are the costs of ATM inefficiency. The industry estimates that in 2012, in Europe, delays cost €4.5 billion to airlines and €6.7 billion to passengers [1.20]. To that one must then add the ripple effect through the economy. Similar figures are found in the US. In 2012, air transport generated or induced €2 trillion (or 3.4%) of the world's economic activity, keeping 58.1 million people in employment [1.21]. This must be put in the perspective of the €600 billion in revenues for an airline industry, that in 2014 generated a meagre €16 billion in collective net profit for a 2.65% profit margin [1.22]. Airlines generated just €4.8 of net profit for every passenger carried.

Airline pilots are also a very expensive component of ATM, costing a conservative €120,000 to train and another €150,000 per annum in salaries and benefits,

per pilot. It is therefore estimated that over the 25 year life span (90,000 hours) of a large wide-bodied airliner such as the Airbus A380-800, an airline will incur some €40-€50 million in ATC costs, some €64 million in pilot salaries and about 2 million in training assuming an average of 17 employed crew per long haul aircraft [1.23]. To this, one must add the regular training in simulators and the significant cost of the cockpit and the crew-support avionics hardware that is found within. All this must finally be compared to the 2015 list price of the vehicle, at €350 (USD \$428) million.

Therefore, there is an opportunity cost of over €100 million per aircraft that could potentially be better spent on an alternative, possibly autonomous, ATM system.

1.1.3 Future Trends in Demand

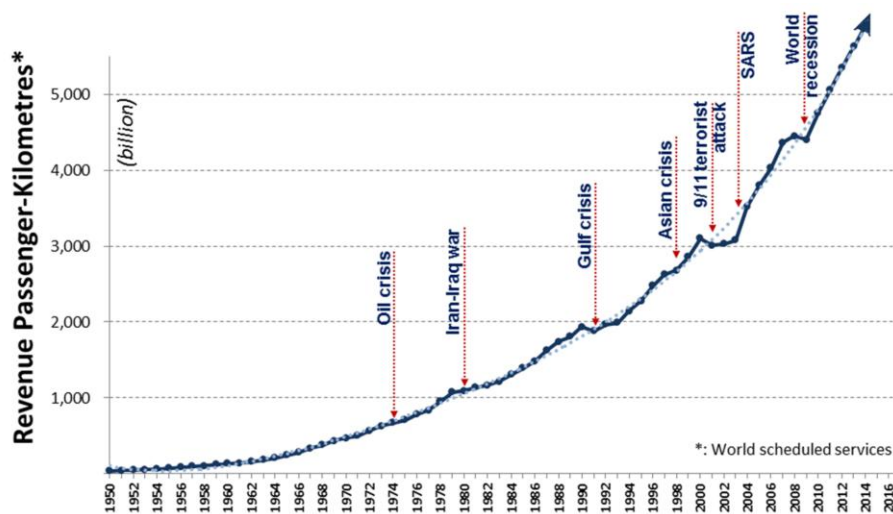


Figure 1.1.4: Growth Trend Spanning 65 years in Aviation Industry (Source: ICAO)[1.24]

If past trends are anything to go by, we can expect a continued exponential growth in air traffic for the foreseeable future. For the past 15 years, RPM has increased at a compounded 5% average annual rate, as the world economy increased by 2.8% per annum [1.24]. This growth trend is highly robust seems unfazed even by major security and economic world events. Just when growth appears to falter, it is soon followed by a renewed surge that resumes the steady exponential trajectory (**Fig 1.1.4**).

A rough, but conservative, back-of-the-envelope calculation based on currently available data, [1.25][1.26], quickly shows that even if the world population stays static, the potential for growth in air traffic is enormous. If China, India and Indonesia alone reach the same average level of RPM, per year, per capita to just match the average United States citizen, *today*, we can expect a *tenfold* increase in commercial air traffic volumes. This assumes (quite unrealistically) that the rest of the world stays put and that demographics do not change from 2014 levels. Eastern European countries are also on the path of rapid expansion in air travel rates [1.27]. In their case, the added effect compounds an already heavily congested region. The airframers predict 29,220 new airliners and 498,000 new pilots by 2032 [1.23]. Major growth spurts are also expected with the democratization of personal aviation and commercial unmanned flight.

1.2 RESEARCH IN ATM

Research in most aspects of air transport is vibrant and highly mature, particularly in respect of vehicle design, propulsion and avionics. These are fields with a solid scientific basis, which makes setting and reaching targets, a very structured affair. However, the same cannot yet be said of ATM, where the research community seems somewhat fragmented with numerous newcomers providing a plethora of disconnected, often contradictory, contributions from many adjacent fields. ATM still lacks a unifying theoretical framework, which researchers can use to make sense of the bigger picture [1.28]. This indicates that the ATM research community is still lagging behind current needs, with the bulk of the work surrounding an ATC paradigm that has not changed substantially since the 1920s [1.1]: That of human controllers on the ground issuing voice instructions to pilots in the air, or on the runway, to maintain orderly flow and separation of traffic. This is not meant to belittle the immense progress and increases in capacity over the decades, but the improvements were incremental, and for good reasons. However, alternative paradigms were never explored in earnest, and hence, all improvements remained largely peripheral to the entrenched main activity of the ATCo.

However, this leaves us with a system that has some fundamental shortcomings. As congestion increases, the ability to recover from disruption seems to decrease [1.3] and the rate of human ATCo error increases [1.29]. Given the world economy's dependence on air transport, this is disconcerting, particularly after witnessing the vulnerability of the sector to a single volcanic event, which shaved off up to 29% of global traffic for the best part of a week, leaving 10 million passengers stranded and €4 billion in economic losses [1.30].

Recent high profile aviation incidents and accidents have also highlighted the helplessness that the currently centralised ATM system faces. A computer disk failure at the NATS Swanwick ATC Centre in December 2013 caused severe disruption in the UK airspace with 300 cancelled flights and hundreds more delayed [1.31]. A rather similar incident at the same centre in December 2014 caused another global ripple effect after just 35 minutes of downtime [1.32].

When the relatively weak communication and surveillance links between ATC and aircraft are severed, bad things have happened. In both the 2009 Air France Flight 447 [1.33] and the 2014 Malaysian Airlines Flight 370 [1.34], loss of radar contact, ADS-B, ACARS, and pilot communication left ATC clueless for weeks on the fate and final trajectory of the aircraft. In such reasonably rare cases, any potential rescue response is sterilised, fatalities often reach 100%, and the authorities may spend years piecing together the puzzle of what could have happened – and only if the flight recorders are finally retrieved. It still surprises many that this continues to happen despite 21st century communications technology. The fact is, we can practically eliminate this information vacuum, if inter-aircraft networked communication is established and cooperation is enhanced.

A major overhaul of ATM will eventually be needed, but it would appear that the long term goal of ATM research rests undefined and there is still a general lack of

direction in the community's research efforts. This may be exemplified by research programmes on the Air Separation Assurance System (ASAS) for en-route self separation [1.35][1.36], which partly contradicts rigid 4D Contracts proposed in IFATS [1.37], or the more flexible 3½D Contract of Objectives (CoO) (also Target Windows) of Eurocontrol's CATS project [1.38] which is in turn incompatible with the tube-based traffic control considered by NASA [1.39] and many other variations, leading to a proliferation of inconsistent separation concepts. The same can be said about the *free-flight* proposals [1.14][1.15], which run counter to automated airspace concepts [1.40][1.41]. Integrating military aircraft, general aviation, unmanned (UAV) and personal (PAV) air vehicles, into the airspace is also problematic if each party abides by a different rule set, thus necessitating further segregation. Interoperability demands coherence in the air traffic system and yet there is wide incongruence on basic principles, and then there is the increasingly vestigial role of the pilot, in an age where an aircraft equipped with 4D-FMS is technically capable, or contractually required to follow a pre-planned 4D business trajectory with high accuracy.

1.2.1 SESAR and NextGen

The practising Air Navigation Service Provider (ANSP) community is a long-established and rather traditional community, which has nonetheless been living in the shadow of an overhaul for some time. The last decade has witnessed the emergence of two large multi-billion-dollar research and deployment initiatives, one on each side of the Atlantic: NextGen (USA) [1.42] and SESAR (EU) [1.43]. These programmes enjoy the political backing necessary to challenge the status-quo and aim to collaboratively bring the benefits of increased automation and modern information technology to the aging ATC infrastructure [1.44]. The stated goals of both SESAR and NextGen do not differ by much, mostly as a result of the strong political and practical effort to harmonize the developments on both sides of the Atlantic, [1.44]. In essence, both aim to reduce (but not remove) the need for human tactical intervention by emphasizing on strategic de-confliction.

Underlying these projects is the notion of aircraft explicitly sharing their intent with ANSPs in the form of detailed 4D business trajectories (4D-BT), as opposed to the current model of merely extrapolating future motion of aircraft from past radar trails. This necessitates a new breed of communication infrastructure that is built around net-centric principles using peer-to-peer data-links and a *System Wide Information Management* (SWIM) system [1.45]. The timely distribution of weather information also takes a very central role in these projects, because an accurate and coherent continental weather-picture goes a long way in reducing system uncertainty.

Despite the fairly long timescales of these programmes (10-15 years), their conservative nature is evident. They begin the research effort with the pre-condition of keeping the human in the inner control loop – with ultimate responsibility for separation assurance. The natural consequence of this decision is the introduction of a number of tools to facilitate the human's cumbersome role in the system and add to the ever expanding list of installed or proposed decision support tools, situational awareness aids, or safety net functions (see **Table 1.6.1**) Some re-partitioning of the tasks

traditionally associated with the controller and pilot is envisaged. Thus, neither NextGen nor SESAR are poised to undertake a complete overhaul.

However, the rate of change, was catalysed. Digital communications is leveraged with the SWIM system and will carry flight data, weather information and airspace restriction updates. Automatic dependant surveillance-broadcast (ADS-B) provides regular updates to and from surrounding traffic about GPS position, and other flight status. A number of different separation modes are introduced depending on context. One of these is the 4D-BT, that “contractually” binds each flight [1.38] to traverse specific way points at pre-determined times. Self separation through ASAS [1.36] is also contemplated in some circumstances.

If we join the dots, starting with the earliest emergence of fundamental concepts such as *free-flight* and distributed control, through to collision detection and resolution (CD&R) schemes, and following by the increased reliance on automation proposed by SESAR and NextGen such as fast-time air traffic simulation, flow/capacity management and trajectory based operations, it all seems to pave the way towards a consolidation of approach towards the ATM problem. This can be seen as the emergence of a new unifying ATC paradigm immersed in automation, and perhaps autonomy.

1.2.2 Full Autonomy?

This chapter attempts to rationalise the recent developments by putting them in the context of the overarching direction that the field has been taking. Many researchers and their contributions have been tacitly assuming that full automation of air transport is the way forward [1.46]-[1.49]. However, few in the current operational environment are willing to accept this kind of future, and are even less willing to present a clear roadmap. We therefore undertake the task of charting the basis of this new ATM paradigm. We call it "*Computational Air Traffic Management*" (CATM) to emphasize our expectation that numerical processing and computational emulation will grow to underpin all decision-making in these gradually emerging ATC systems [1.50]. Higher fidelity on-line modelling of traffic patterns, weather, aircraft and airports, provide the opportunity to generate quasi-deterministic, fuel-efficient and conflict-free trajectories for entire fleets. So a question that seems worth answering is: What will succeed SESAR and NextGen?

1.3 THE NATURE OF THE ATM-ATC PROBLEM

In order to analyse the complexities of ATM we begin by having a look at the role of two of its main subsystems, and the type of problem they seek to address. Oftentimes, ATM and ATC tend to be used interchangeably in the literature. However, as per ICAO definitions, ATM is a generalisation of the concept that includes both Air Traffic Control (ATC) and Air Traffic Flow and Capacity Management (ATFCM), among several other ancillary systems.

Air traffic is subject to a very wide variety of random influences and has to be able to deal with contingencies. Inclement weather, late boarding, industrial action,

malfunction, pilot error, accidents, or the occasional volcanic eruption, all tend to make the system highly non-deterministic. Some of these disruptions are limited in scope and impact. Others tend to *propagate* through schedules and *multiply* their effect, and if unchecked, may result in far more widespread disruption. The system is the result of several highly coupled and poorly modelled sub-systems and, from a complexity perspective, it means that the ensuing problem has all the markings of an intractable logistical quandary.

Yet, (so far), the established ATFCM/ATC mechanism still seems to be working remarkably well and the level of safety is admirable. So what is the underlying reason for its apparent success? How can this be possible, given the problematic nature of the task? There are indeed five underlying characteristics, which have rendered the problem of coordinating air traffic tractable in practice, and any future evolution or drive towards further automation, would be well advised to carry over some of these properties:

- Inbuilt Slack
- A Diurnal System-Resetting Cycle
- Pre-flight Planning (ATFCM)
- Slack Redistribution (also ATFCM)
- Hierarchical Authority Devolution

Since the system is so intrinsically chaotic, it depends on the presence of sufficient room to manoeuvre, or ‘slack’, in order for it to be able to reorder and readjust around contingencies. This effectively translates into a measure of flexibility and therefore, robustness [1.51]. Slack is present in every aspect of operation, such as vacant take-off/landing runway slots, generous air-separation minima, non-minimum-length trajectories and lax scheduling. Perhaps the greatest contribution comes in the form of the widespread traffic curfews that are imposed every night at many airports [1.52]. These have the effect of resetting the air traffic system daily, which in turn reduces the risk of carrying over disruption from one day to the next. However, slack also equates to inefficiency by reducing the utilisation factor of some very expensive infrastructure. As the environmental and noise footprint of modern aircraft decreases, economic forces will push to erase such curfews. So, while critical to the reliability of the system, slack cannot be inserted indiscriminately and the pressure is mounting to use it more judiciously.

Currently, pre-flight planning is performed by the Air Traffic *Flow* and Capacity *Management* function. This, as opposed to ATC, is a primarily strategic and pro-active activity that tries to pre-empt conflict and congestion through forecasting and planning [1.53]. Using a centralised ATFCM unit (such as the CFMU in Europe), it gathers information from a number of sources such as ANSP capacity, flight schedules, weather forecasts and other NOTAMs to optimise airline flight plans as much as possible right up to the moment of flight. So *the role of ATFCM is to facilitate the subsequent ATC problem by eliminating most of the uncertainty.*

On the other hand, ATC is the tactical arm of the ATM system and deals with the day to day variability, which inevitably remains within the system. ATFCM ensures that ATC will only have to deal with the occasional flight interaction that comes about because of the said variability. When inclement weather reduces the capacity in any part of the system, ATFCM takes Traffic Management Initiatives (TMI) to redistribute traffic and delay new departures. ATFCM attempts to maximise capacity by dynamically rebalancing the equilibrium. Most of the potential conflicts are removed by ATFCM at the planning phase in order to ensure that the ATC system, (which includes the pilot) is never overwhelmed. In fact, *if the system were perfectly deterministic, tactical traffic coordination could be rendered unnecessary*. Aircraft would simply follow their own pre-computed, conflict-free trajectories and ATC would therefore become redundant.

At present, the slack redistribution function is an accidental consequence of ATC. By adopting a reactive approach over a restricted space/time horizon, ATC deals with potential conflicts, a few minutes before they turn into dangerous losses of separation. When conflicts are detected, aircraft trajectories are reordered in a way that resolves the original problem. Slack is consumed or redistributed during this process. However, the process is rather crude and is wasteful of this valuable resource. Some pioneering work has been reported to better manage the slack in airline schedules [1.54], [1.55] but little has been done to optimally manage the slack present in the actual aircraft trajectories and across multiple airlines.

Finally, the ATC problem is considerably simplified by the devolved nature that ATC has embraced over the decades. Due to historical limitations in communication infrastructure, ATC had no choice but to evolve into a *hierarchically-flat* system, whereby decision-making occurs in a decentralised fashion with limited lateral interaction. It consists of a self-organising collection of *agents* operating independently across a large number of air traffic control centres. Some degree of centralisation commenced many years later, with ATFCM, but it still did not fundamentally change the way ATC operates.

ATC is a living example of Swarm Intelligence. No individual decision-maker needs to do very much, or even appreciate the full extents of the problem, yet collectively, they address the ATC problem in a remarkably effective way [1.56]. The rules of ATC are such that overall, each air traffic controller (ATCo) contributes to the global ATM problem by taking responsibility of a small sector of airspace. The collective, but independent, action of all ATCos solves the ATM problem. This behaviour is ultimately responsible for the relatively high capacity of the system.

1.4 CHARACTERISTICS OF AN IDEAL ATM SYSTEM

The requirements definition of an ideal ATM system is quite subjective. However, the ICAO defines 11 key performance areas (KPAs) for ATM together with their respective Key Performance Indicators (KPIs) and these have been formally adopted by SESAR (and more loosely in NextGen) as basic guiding principles [1.57]:

- | | |
|--------------------------------|--|
| • Capacity | ▶ Maximum Airspace Throughout per Hour |
| • Efficiency | ▶ Average Percentage Flight Elongation/Delay |
| • Safety | ▶ Incidents/Accidents per million Flights |
| • Flexibility | ▶ Time to Adapt/Respond to New Circumstances |
| • Predictability | ▶ Variations from Flight Plans / Schedules |
| • Security | ▶ Security Breaches per million Flights |
| • Access and Equity | ▶ Number of Effectively Excluded Operators |
| • Cost Effectiveness | ▶ ATCo Productivity (Flights per ATCo.hour) |
| • Global Interoperability | ▶ Number of Incompatibilities |
| • Participation | ▶ Level of participation by Airlines/States |
| • Environmental Sustainability | ▶ CO ₂ Emissions per passenger.km |

These KPAs cluster naturally into groups, which bind themselves to the layered systems approach (**Figure 1.4.1**) often described in the community [1.58]. These layers interact by setting performance demands in one direction and imposing constraints in reverse. Notwithstanding their importance, one finds that the ICAO KPAs are fairly generic and stack up too high in the top two layers. This makes these KPAs hard to translate into specific algorithms, making them less useful to the engineer tasked with designing bottom Layer-5 technologies.

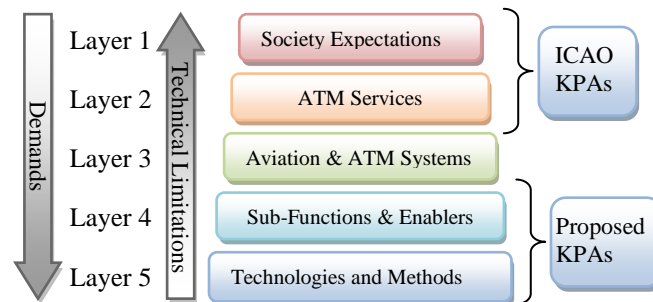


Figure 1.4.1: Layering of Performance Concepts [1.58]

For this reason, a number of additional KPAs are hereby proposed for the lower two layers, which effectively translate the ICAO KPAs into a language that is closer to what is useful for practical algorithm design and development. This section is organised such that each KPA is first described together with the respective KPI and then reference is made to how the current ATM paradigm performs in each case.

• **Scalability**, with the number of aircraft – This implies that ATC capacity can indefinitely increase in lockstep with traffic density and demand, without seizing up or generating delays. In practice this can be measured by the increase of ATC resources required with respect to an increase in traffic. A logarithmic ($\log n$), linear (n), or at worst, a polynomial (n^c) relationship, implies feasibility. ▶ Currently this is certainly

not a strong point of the today's ATC system. As mentioned earlier, as the traffic density increases, the airspace is being broken down into ever smaller sectors and more controllers have to be brought in [1.3]. This adds costs and creates an undue workload on the pilot who needs to hand-over too frequently between sectors. It also occupies and clutters the limited radio bandwidth available. Given the eventual integration of UAVs into the airspace, the complexity of mixed fleets operating at high multiples of today's traffic densities will ultimately make the human coordination of ATC unsustainable or prohibitively expensive for businesses and the travelling public.

- **Robustness**, with no single point of failure – The physical architecture of ATC should be such that no failure of any single piece of equipment or any damage to a finite region can bring the entire system down. Robustness of any system is readily measured by its failure rate. The solution can be partly achieved by using distributed multiple redundancy and a good way of achieving this is to make it airborne. ► Currently, ATC is only modestly distributed. Although the ATC problem is divided between tens of thousands of ATCos, most decision making is still centralized inside a handful of locations (ATCTs, TRACons and ARTCCs) sharing a limited amount of hardware. The recent NATS system failure of 12th December 2014 caused mayhem in the UK airspace with a global ripple effect after just 35 minutes of downtime [1.32]. Such central nodes become critical single points of failure, that in-turn make the system vulnerable to malfunction or terrorist attack. ATFCM is by its very nature even more centralised. The problem is that pilots become quite helpless in very dense traffic if regional ATC services shut down suddenly, jeopardising system safety and security.

- **Multi-Objective Optimality** – ATM should be subject to optimisation; however, there are numerous parameters and environmental interactions to take into account. Optimising with respect to one cost-function generally implies sub-optimal performance with respect to others. However, multi-objective ATM optimisation tackles this problem by optimising the system at various trade-off positions, thereby generating a Pareto-front [1.59]. By optimising across several variables at once it presents information that can be used to make an informed choice between the various conflicting requirements. A weighted metric can then be used to assess performance. ► ATC attempts to do this in a poorly structured and ad-hoc fashion. Much more automation would be required for it to be able to tackle this aspect rigorously and methodologically and reach its goal of efficiency and cost effectiveness.

- **Collective Optimality** in terms of cumulative environmental impact – From an environmental perspective it is pointless optimising one aircraft trajectory to reduce emissions at the expense of the efficiency of other trajectories. What matters is the collective environmental impact of all airspace users which is a readily quantifiable KPI. The goal of reaching a minimum carbon footprint can only be achieved by treating it as a worldwide (or continental) optimisation problem. ► Unfortunately, environment protection takes a low priority in conventional ATC. The focus on safety can easily swamp an ATCo's motivation to mitigate fuel burn or environmental impact. In addition, human ATCos lack the depth of visibility to predict 2nd, 3rd or nth order repercussions of each of their decisions.

• **Stateful Control** – During the execution of any flight, a good ATM system should keep track of the state of each flight in real-time. It should be aware of the optimal trajectory and should issue the fewest and smallest deviations from this prototype trajectory. This is a measurable KPI. As far as possible, any variation should take into account all previous and future interventions on that trajectory in order to treat it equitably. ► At the moment, ATC intervention and traffic deviations occur as necessary with little consideration for their cumulative effect on a flight. ATCo decisions are based on the instantaneous traffic conditions per sector, with no regard to a flight's history or future. There is no concerted effort between individual ATCos along any flight to holistically optimise its trajectory.

• **Load Balancing** across multiple resources – Ideally ATM should have an inbuilt load balancing mechanism that spreads demand for runway/airspace capacity over time to ensure that each resource is well utilised at all times of day, averaging out peaks with troughs in demand. The variance between resource utilisation factors is the relevant KPI in this context. This is another way of saying that slack needs to be evenly distributed in a controlled fashion. ► Contemporary ATC does a fairly good job at many of the major airports. For instance, London Heathrow's two runways are utilised at near full capacity, but only during daytime. Additionally, the load balancing process is still quite crude and only works in nominal situations.

• **Continuous Adaptability** in relation to new circumstances – Air traffic scenarios are highly dependent on uncontrollable factors such as the weather, accidents, boarding delays, or industrial action. In any of these scenarios, a rigid ATM system is highly suboptimal. Most contingencies develop gradually. An ideal ATM system should be able to utilise all available information to adapt to such contingencies, just as gradually as they evolve. Such a system would therefore avoid the creation of traffic conflicts at the outset, rather than reacting to them when discovered. The time it takes to re-achieve high efficiency is an appropriate KPI ► Today, *the update rate* in the ATC system is excessively low and bottlenecked by slow voice-based communications. Information flow is asymmetric and processing is hampered by human limitations. This limits the speed at which ATC can react to dynamic scenarios, which in turn limits the maximum number of aircraft that can be handled by any single ANSP at a given time. (This is reminiscent of the Icelandic ash cloud saga). Predictability is also negatively affected.

• **Resilience**, in relation to disruption, delay or perturbation – This ties with the previous point in that an ideal ATM system must be capable of rapidly re-organising itself in terms of structure and priorities. The system must quickly reach a new collective optimum after any change in operational conditions or constraints. The peak level of disruption in response to a given disturbance is a good figure of merit. ► ATC, as it stands, is unable to utilize real-time information effectively, which limits its flexibility. This is arguably one of the underlying causes behind the European meltdown of the air traffic system in the wake of the April 2010 Icelandic Eyjafjallajökull volcanic eruption. The ash cloud dispersion mechanism was quite inhomogeneous and many windows of opportunity existed for safe flight. However, even with detailed real-time

ash concentration information becoming available, the ATM system failed to reconverge fast enough to a new viable air traffic pattern that exploited the many windows of opportunity effectively. The crude solution adopted by the industry was to raise the maximum permissible volcanic ash concentration.

- **Prioritisation** depending on service agreements and emergencies – Some air traffic is more important than other traffic. This means that airlines should be able to request (possibly against payment) different priority levels over the use of limited airspace. Emergency traffic should have a means of being granted top priority at short notice. In this case the relevant KPI would have to be the degree of satisfaction of requests in relation to their priority. ► ATC already performs a certain degree of traffic ordering and regulations ensure that it will prioritise any traffic in distress. However, this is not done very gracefully and tends to disrupt excessively the harmonious flow of surrounding traffic.

- **Penalty Dispersion** for unforeseen events – Disruptions to the ATM operational equilibrium must be handled in an equitable manner such that no party is disproportionately penalised. The overheads introduced by unexpected scenarios should be shared by all members of the airspace such that the effect is diluted and everybody is given equal access to the airspace. This also improves network resilience to further disruption. The inefficiency suffered by the worst hit flight is a good KPI. ► ATCos lack a comprehensive picture of the complete traffic scenario. This makes it impossible for them to take coordinated decisions on large numbers of aircraft. At present the sharing of disruption overhead is more accidental than premeditated.

- **Interoperability** with numerous geographical information systems (GISs) such as meteorological data sources, noise-sensitivity of urban areas, population density, airspace restrictions, ecologically sensitive areas, volcanic ash distribution and other such dynamic databases – An ideal ATM system should lend itself to be easily interconnected with GISs such that the latest relevant information is available to the optimisation engine. This can be measured by the number and range of systems that can be integrated within a given ATM paradigm. ► Although ATC is fairly tightly coupled to meteorological data sources, too much detail risks serving the ATCo with an information overload. Hence, despite the availability of countless other sources of useful information, limited ATCo processing capacity precludes the possibility of digesting it in a timely fashion. In contrast, *Global Interoperability* in the ICAO sense, is a political matter rather than a technical one and only widespread adoption of the same guiding principles would guarantee any degree of international harmonisation.

- **Predictive Ability** – An ideal ATM system should be able to forecast the (short term) future air traffic status in order to avoid any build-up of traffic congestion. This also improves contingency planning. The ideal KPI in the case would measure the deviations of actual flights from the latest plans available to the ATM system ► ATFCM currently does a fairly good job of this, by extrapolating flight plans. However, there exists no technological measure to *guarantee* that pilots will execute such plans to the letter or that they will inform ATC of new plans before they depart from their agreed plans. ATC is overly-reliant on mutual pilot/ATCo trust.

1.5 THE SCIENCE OF ATM

Before the community can consider re-engineering the entire ATM system, the underlying *science of systems* needs to be clarified and well understood [1.28]. This is recognised in both NextGen and particularly SESAR through its Work-Package-E (WPE) programme. Among the most concerted efforts to define what could be called a theoretical basis of ATM, were a series of public funded projects conducted in Europe between 2001 and 2013. Starting with HYBRIDGE [1.60], iFly [1.61] and finally MoVes [1.62], these projects have undertaken the development of the basic science that governs complex systems such as ATM and power networks, and were influenced from seminal work conducted on the other side of the Atlantic, namely at UC Berkeley and Stanford University [1.63].

It started amidst the realization that the behaviour of many ordinary systems such as road vehicles, aircraft, washing machines or ATM cannot be suitably modelled, analysed, simulated, optimised or reliably controlled using either classical discrete time or continuous time techniques. These systems exhibit an interaction between a number of continuous processes which are closely interlinked through an overarching system with discrete states. This can in fact be modelled hierarchically as a finite state machine, whose state changes are partly governed by continuous variables, which may in turn dynamically change in accordance to analogue differential equations (describing the physical world) and the system's discrete state.

These cross-domain Hybrid Dynamical Systems or Hybrid Automata, as they are now called, have been successfully used to model various aspects of the ATM problem. Stochastic hybrid models of systems give new insight of how to characterise the behaviour of complex systems by making allowances for unmodelled, or poorly modelled dynamics, noise, random failures or external nondeterministic influences, (such as weather) while simultaneously considering the numerous interactions between continuous and discrete states in the system [1.64]-[1.66]. Initially, significant emphasis was placed on conflict resolution between multi-agent hybrid systems [1.67] with a clear reference to a reactive type of ATM. This kind of modelling allowed certain kinds of conflict resolution manoeuvres of aircraft to be formally verified with respect to safety by using the notion of reachable sets of hybrid systems. Important results related to *reachability* and *verification* were published by Lygeros, Tomlin, Prandini, Sastry and others [1.65]-[1.70]. However, the principles of hybrid automata are well suited to characterise other aspects of ATM. By studying the synergies with other fields such as game theory, interesting approaches for optimal controller design for multi-agent systems have been proposed [1.71][1.72].

After the creation of a scientific language that faithfully models ATM, optimisation science, operations research, and optimal control can form the theoretical pillars for solving the resulting numerical problem. It seems therefore, that the future lies in the ability to combine these fields either at the engineering level or through the discovery of even more fundamental science and mathematics that encompasses them all. Some may be tempted to point at Dynamic Programming (DP) for this purpose, and indeed DP is general enough to handle *all aspects* of ATM in one conceptually simple

framework. Neat as it may sound, the problem with this approach is of course the sheer size of the computational problem it generates, but it does change the perspective on ATM. So, one must never forget that ATM is not a problem without a known mathematical solution. ATM is only a problem with yet- unknown efficient computational solutions. Therefore we view Computational Air Traffic Management as a discipline, which focuses on establishing ways of mathematically posing the ATM problem, that subsequently leads to computationally tractable solutions that can form the basis for an ideal ATM system.

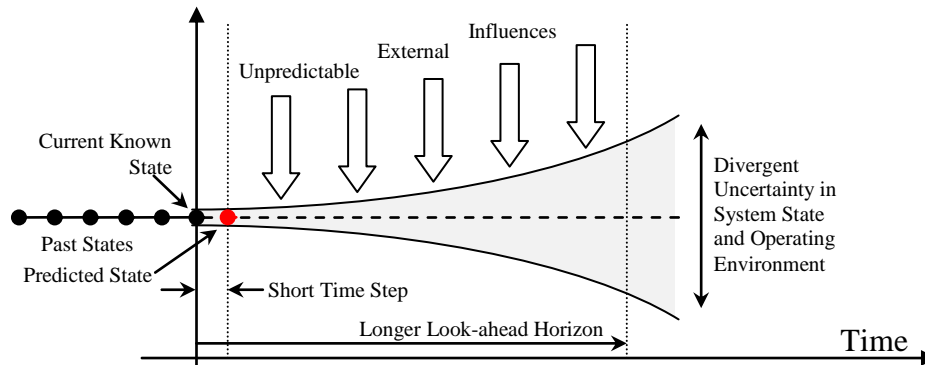


Figure 1.5.1: High update rate leads to much better prediction of the operational environment

In our opinion, one other key tenet of CATM is that *if sufficient computational, communication and surveillance resources are provisioned, the system update rate can be increased to the point, where over the short duration of each system time-step, the traffic scenario would advance by a negligibly small amount* – and this is a powerful enabler in dynamic systems like ATM with a diverging uncertainty envelope, because over the short term the uncertainty is very limited as is described in **Figure 1.5.1**.

At these time scales, the distinction between ATFCM and ATC blurs. The difference between the purely tactical and strategic phases degenerates. Given accurate knowledge of the present state, most phenomena that can affect the air traffic scenario, could be predicted with confidence over short horizons. In the near term, the uncertainty surrounding system states collapses in ways reminiscent of receding horizon control (**Fig 1.5.1**) [1.73]. So in a sense, the tactical is absorbed into a single pseudo-strategic planning process, which has a continuous span of influence that ranges from a short term horizon of a few seconds, to a much longer horizon lasting many months. Thus, by raising the system update rate, CATM encompasses the roles of both ATFCM and ATC.

1.6 JOINING THE DOTS... THE EMERGENCE OF CATM

In this project, we propose Computational Air Traffic Management to provide an efficient solution that merges both ATFCM and ATC into a common framework that no longer needs to distinguish between the two. The best features of the current abstractions are retained and combined with a variety of novel concepts that enhance capacity, performance and reliability.

The first major effort to shift the current ATC modus-operandi traces back to the mid 1980's when the ICAO set up the Future Air Navigation System (FANS)

committee [1.74] - but change in ATM is exceedingly slow and laborious, and despite profuse publication, implementation does not keep the same pace. This has got to do with the partly justified apprehension of “meddling with a working system.” After three decades of consideration, FANS is still far from universally adopted. The same fate seems to have met the Distributed Air/Ground Traffic Management (DAG-TM) system [1.75], that was the product of NASA’s Advanced Air Transportation Technologies (AATT) project and was supposed to be an embodiment of *free-flight* (described later).

TOOL		CONTROLLER OR PILOT SUPPORT FUNCTION
CONTROLLER	STCA	A short term collision alerting system that keeps controllers vigilant by visually highlighting pairs of flights on an imminent (2 min) collision course.
	MTCD	A medium term collision detection system that also takes into account flight plans to predict medium term conflicts, 30 minutes in advance.
	MONA	Monitoring aids to enhance situational awareness and provide reminders.
	SYSCO	System supported coordination between displays during sector handover.
	CRDA	A converging runway display aid for handling intersecting approaches.
	URET	A user request evaluation tool to replace paper strips to account for sector traffic
	pFAST	A passive final approach spacing tool for decision support near the terminal area
	EDA ¹	An en-route descent advisor for accurate plane delivery to arrival metering fixes
	EDA ²	An efficient descent advisor to minimise fuel burn during decent.
	TMA	A traffic management advisor to limit traffic into any TRACON.
	MSAW	Minimum safe altitude warning in case one aircraft gets too close to the ground.
	ASAS	A separation assurance workload reduction tool that reduces the need for frequent pilot-controller communication and intervention to maintain separation from preceding aircraft flying in the same direction, airway and altitude.
	CPDLC	A data link system to supplant existing bandwidth-inefficient voice communications between pilot and controller.
	SMAN	A surface manager to assist controllers with computing taxi routes and taxi times, taking into account platform constraints, surrounding traffic, and start-up delays.
AMAN	An arrival manager offloads the controller from the complex task of optimizing the sequence and buildup of the arrival queue, by taking into account wake vortex category spacing to maximize (but not overload) landing runway throughput.	
DMAN	A departure manager, that maximizes the use of the runway by assisting the controller to meet the takeoff schedule by optimizing the sequencing of taxi operations in preparation for departure.	
PILOT	GPWS	A ground proximity warning system designed to alert pilots if their aircraft is in the immediate danger of flying into the ground or another fixed obstacle.
	FMS	Flight management system that automates editing and execution of flight plans, stores navigation databases of SIDs, STARs and instrument approaches etc...
	AWDAP	An airborne wind-shear detection and avoidance program that detects and alerts the pilot, visually and aurally of a potentially hazardous wind shear condition.
	TOPM	A take-off performance monitoring tool to alert pilot of insufficient acceleration.
	RCAF	A runway collision avoidance function designed to alert or advise pilots in the case of an immediate danger of a runway incursion during take-off or landing.
	TCAS	A traffic alerting tool and collision avoidance system that advises the pilot about potential breaches of separation and gives resolution advisories in case of imminent collision with traffic that the pilot or controller might have missed.

Table 1.6.1. Just a few of the tools developed to assist human pilots and controllers

There have since been many other attempts to re-think the ATM problem even deeper, and with the benefit of 30 years of hindsight, one may attempt to extrapolate a trend in the general direction of development. The many Communication, Navigation and Surveillance (CNS) initiatives seem to be coalescing into ever larger projects. When NextGen and more particularly SESAR came along, they had a mandate to make sense of, and absorb, the results of the myriad smaller projects that preceded them. There were indeed *many* such projects, which had emerged reactively to produce tools, which deal with specific problems – such as those listed in **Table 1.6.1**.

Yet the proliferation of too many tools is probably not a good thing. It is the limitations of the human ATCo's (or pilot's) reaction time and situational awareness that set off the development of new tools to fill-in the gaps. **Figure 1.6.1** attempts to rationalise this observation graphically on flattened triple axis chart. ATM was progressively broken down into numerous sub problems that were addressed with an accumulation of specific tools. The cumulative number of interacting systems will peak sometime around the middle of this decade.

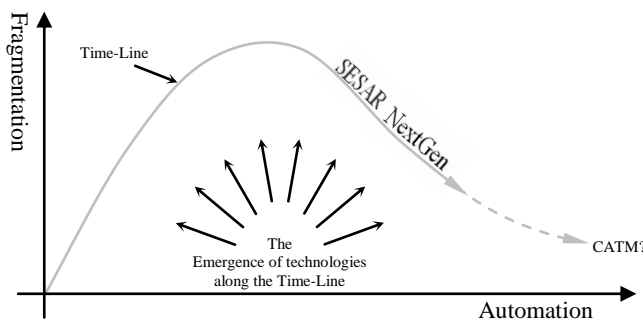


Figure 1.6.1: An Underlying Convergence?

This fragmentation of the ATM concept (measured by the number of tools in use) and the associated rise in complexity is detrimental to the system's performance and stability. It is probably amidst this realization that large programmes like SESAR and NextGen gained political traction. A new observable trend is the gradual reduction in the number of components that make up the system in favour of a simpler, over-arching, unifying paradigm that is applicable to a wider portion of the ATM problem. SWIM for instance, attempts to subsume efforts from: CDMnet, TMAnet, URETnet, VHF, VDLM2, ACARS, FANS, AFN, CPDLC, ADSB etc. with a single versatile communications network [1.76]. Broader, automation seems to be the coagulating agent, and as the human is gradually, but surely, driven out of the inner loop, ATM is likely to witness a drastic simplification. This convergence is likely to give rise to a CATM-driven system.

1.6.1 Free flight

The motivating idea for CATM is the devolution of control to the cockpits (but not necessarily the pilots) and this traces its roots to a concept championed by Capt. R. M. Baiada in the mid-1990's [1.12]. This early impetus for a paradigm shift in ATM became known as *free-flight* [1.77][1.78], and places the onus of traffic separation on the pilot as opposed to the ATCo. The concept is remarkably simple but it met some stiff scepticism because it predated much of the infrastructure and safety-net functions that would make it work [1.79][1.80]. Reliance on ATCos to passively monitor traffic for the occasional conflicts was not going to work either for reasons described in the

next section. *Free-flight* generally relies on enroute traffic sparsity to allow pilots to safely fly direct routes with minimal probability of interacting with surrounding traffic [1.81][1.13]. This was validated through human-in-the-loop simulation in sparse regions through projects such as “*Mediterranean Free-flight*” [1.35].

When compared to the capacity/safety ratio of current methodologies, the concept is initially hard to refute [1.79] and has seen the likes of NASA [1.83], the NLR [1.79] and IATA [1.80] taking it quite seriously, with the RTCA going as far as recommending a transition strategy [1.78]. Sparsity is further maximised by allowing traffic to break-free from the classic rigid airway structure [1.82]. Aircraft are instead allowed to spread out and fill the airspace, thereby diluting the traffic density and enhancing capacity. Low traffic density results in rather infrequent interactions, and for those rare occasions, pilots would be assisted with a wealth of cockpit-based, conflict detection (CD) technology to highlight potential conflicts. Such technology may even generate and suggest conflict resolution manoeuvres to the pilot (CD&R).

However, at high traffic densities, such as those encountered in terminal areas (TMAs) and other traffic hotspots, pilots are unlikely to cope with the full extent of *free-flight*. The sheer combinatorial explosion of possible manoeuvres and associated outcomes makes it impossible for humans to cope safely in completely unstructured airspace. The mere devolution of responsibility to the pilots does not help in such scenarios. Moreover, reactive CD&R tools caused more problems than they solved in high density traffic [1.17]-[1.18], resulting in the *domino effect* we mentioned earlier, where resolving one traffic conflict creates new conflicts further down the line. In this form, *free-flight* is precluded from anywhere close to busy terminal areas or in central Europe (for example), where ironically, the problem of congestion needs most attention. The requirement is for a traffic system that makes cockpit-based airborne separation, tractable, making the next step towards CATM inevitable.

1.6.2 The Role of the Human ATCo

For a while, *free-flight* was the source of much apprehension among the ATCo community [1.80]. For many it meant that the ATCo would be left with a vestigial role, or possibly, no role at all, pushing ANSPs into the process of redefining themselves [1.84]. If the cockpit assumes the main task of airborne self-separation, ANSPs could possibly take the role of system watchdogs. However, extensive studies by Parasuraman et al., have shown that a largely passive, monitoring role is considered detrimental from a human-factor’s perspective [1.85] because *humans are notoriously ineffective at monitoring systems with low event rates*. Despite best intentions, humans will gradually lose their state of vigilance as their perceived likelihood of an event diminishes [1.86]. Hence, because of this *vigilance decrement*, sparse traffic and very low workload may be ironically linked to various types of accidents.

One of the failure modes of ATC stems from the over-reliance on the human-in-the-loop, and pilots in *free-flight* are unlikely to fare any better. So the mere delegation of separation to human pilots does little to improve matters in this regard. In most forms of transport, human hypovigilance is a major source of accidents. Over the past 5

decades, human pilots and controllers collectively account for over half of all the fatal hull-loss accidents in aviation [1.2][1.88][1.87].

Even though ANSPs represent less than 4% of total air transport employment, they still exert a tremendous influence on the general development of the entire industry, SESAR and NextGen. For these reasons it is likely that for the interim, human ATCos will continue to hold a very active “inner-loop” role. Therefore, it is unsurprising that in both SESAR and NextGen, the human controller is given prominence and this will fundamentally influence what can or cannot be achieved in these projects.

1.6.3 ASAS and Workload Sharing

One of the touted solutions for sharing the ATC workload with the pilot, while avoiding the difficulty of operating *free-flight* within high density traffic, is called the Airborne Separation Assistance System (ASAS) [1.36]. In simple terms, ASAS reduces the level of ATC intervention needed to guide aircraft along certain routes. This is done by issuing instructions to an aircraft to select the aircraft ahead of it, as a target and to merge behind it while passing over a specified waypoint. A further instruction is then issued to delegate (to the cockpit) the task of maintaining adequate separation between the two aircraft. Repeating this process, creates a chain of aircraft that safely follow one another all the way down to the runway with much reduced ATCo involvement [1.89].

1.6.4 Distributed Control

Some say, that the lack of clarity on who should bear the ultimate responsibility for separation in the airspace raises doubts on whether delegating ATC to thousands of cockpits would be a sensible or safe thing to do [1.14]. The lack of a clear authority structure leaves the ANSP community unimpressed.

However, as per the *free-flight* school of thought, it is precisely the distributed nature of the current system that makes it relatively safe [1.90]. So in the interest of safety (in view of increased capacity) it would be best if ATC functions were distributed still further [1.15]. This removes centralised single points of potential failure and dilutes the workload by ensuring that surveillance capacity far exceeds demand. Additionally, by empowering the pilots to become their own air traffic controllers, the system’s ATC capacity would increase in proportion to the demand.

However, distributed control, per se, does not result in the desired outcome. The distributed system needs to be carefully crafted to ensure that the separate actions of each individual in the system uniformly contribute to the higher goal of assuring optimal separation at higher traffic densities. Instead of human decision makers, practical *free-flight* needs a well-designed, distributed optimisation and control *algorithm* based on established, well-understood science. CATM needs to be based on such solid theoretical foundations, if it is to offer any verifiable performance guarantees and win widespread confidence.

1.7 THE EXPECTED EVOLUTION OF ATC AUTOMATION

Air transport has reached a point where incremental improvements scarcely affect the overall level of service to the traveller. Many air transport sub-problems, impinge on the effectiveness of the air traffic management system and the high complexity of the cross-interaction between the different parts of the broader air transport problem, make it exceedingly difficult for humans to take advantage from any synergistic behaviour [1.91]. In a fast paced operational environment, humans struggle to fully appreciate the high degree of interconnectedness.

On the other hand, ATC automation frees the human ATCo from a very mundane, repetitive task to dedicate his valuable time and creativity to more strategic system design activities [1.40]. If we are to significantly improve air transport, progression towards autonomy seems inevitable and will evolve to reflect the ever-increasing degree of integration that is happening all around us. This is a slow process, which moves in lockstep with technological advances, the emergence of appropriate algorithms and the maturation of the relevant information systems. Serendipitously, there are many such techniques, which are converging to deal with such high-dimensionality, non-convex, dynamic optimisation problems, with influence from fields as diverse as Robotics, Systems Theory, Big-data, Telecommunications and Computational Intelligence.

The last two decades have seen a boom in the research and development of applicable algorithms and information systems and a number of these developments are being classified hereunder in relation to their degree of generality and ability to handle the problem from a holistic perspective. The evolution of CATM is best described in terms of these algorithms and the way they seem to stack into three distinct layers (Fig 1.7.1).

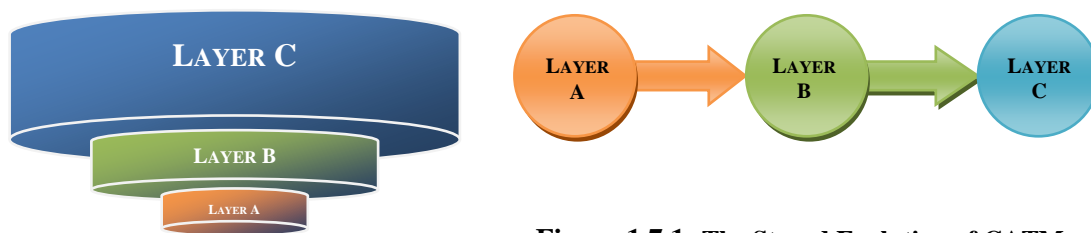


Figure 1.7.1: The Staged Evolution of CATM

The layers form an ordered set $\{A, B, C\}$ where each new layer advances the level of autonomy. Each set is also a superset of the one before, and they effectively depend on the existence of each other in a sequential fashion as shown.

1.7.1.1 LAYER A: Reactive Local Approaches

The baseline requirement for an autonomous and distributed CATM-A system is the ability for an individual aircraft to detect and subsequently avert potential conflicts with other aircraft. *Free-flight* fits neatly in this layer. In Morrel's words [1.92]:

There are three aspects of collision avoidance in the air, which require mathematical investigation:

- (a) discovery and evaluation of the risk;*
- (b) the remedial manoeuvre;*
- (c) the probabilities associated with (a) and (b).*

Conflict detection and resolution (CD&R) is the term given to a binary class of methods that attempt to guarantee conflict-free flight in a two-step process. They are by their very nature *reactive* and employ a Conflict Detection (CD) phase followed by a Conflict Resolution (CR) phase. They invariably employ some simplified model of the airspace combined with a wide variety of probabilistic, dynamic, logic-based (or combination) approaches to attempt to extrapolate the short-to-medium term evolution of the airspace situation. To pick one example, a CD technique based on Intent Inference [1.93] has been proposed. This attempts to estimate future pilot behaviour based on current position and motion, ADS-B intent broadcast, and detailed knowledge of the regional problem domain. However, in common with most CD techniques, the result is only a best effort attempt, which offers no guarantees of accuracy.

Similarly, CR strategies are also often unable to guarantee a solution to all possible types of conflict, even if empirical attempts to compare and demonstrate such best-effort approaches have been conducted using fast time simulation [1.94]. This approach may not inspire much confidence in a safety critical system such as ATM [1.95], but in reality neither does the present ATM system offer any guarantees, so it usually boils down to probability.

Some of the simpler algorithms (e.g. KB3D/KB2D) are amenable to absolute formal verification [1.96] under tightly controlled conditions, and stochastic hybrid systems theory goes a long way with approximate model checking [1.66] for more sophisticated ones, but when there are hundreds of interacting systems, and there are humans involved, even the construction of suitable models has proved elusive. An excellent and widely regarded taxonomy of both CD&R sub-phases may be found in [1.97].

In military language, one can say that due to their reactive, tactical nature, such CD&R methods are necessarily coupled with an underlying mission controller. The airline defines the mission, the pilot executes it, and CD&R and other ATC apparatus is there to give them peace of mind. The temptation is there to turn CD&R into a direct plug-in replacement of the current ATC paradigm, save for its human-centred ground-based nature. However, this is not entirely possible without jeopardising some safety. It was amply demonstrated that at higher traffic densities, any simplistic conflict resolution scheme is likely to cause a chain reaction, leading to an increased number of conflicts for all aircraft involved [1.18]. Thus, for full ATM automation, a higher level of sophistication is apparently required, rather than a merely reactive CD&R approach. Some form of forward looking system that can evaluate the long term consequences of any resolution advisory, seems indicated. Unfortunately, the mathematics of complex systems disagrees with the last statement. Complex chaotic systems, especially those

with such a high level of stochastic perturbation, are nigh impossible to predict too far out into the future [1.98]. The best that has been done in this context is to attempt to disperse localised complexity. After all, this is what ATFCM is about.

1.7.1.2 LAYER B: Pro-Active Global Approaches

A second, more advanced class of algorithms pre-empts conflicts by *generating* and *maintaining* a system certified set of 4D aircraft trajectories at the outset, rather than detecting near-term conflicts when they occur, to then resolve them on the fly.

A candidate CATM-B algorithm must be capable of handling the ATM of the entire body of flying objects from take off, to landing. Notable work has been performed for the enroute phase [1.16] but the scope of such algorithms must be widened further over *all* flight phases and to globally handle the entire airspace. Layer A is retained as an emergency safety net.

Safety is enhanced by further extending the reach of the algorithm into the pre-flight planning stage, so that when no feasible trajectory can be calculated for an aircraft, a flight can be aborted or rescheduled before it ever leaves the runway or before it enters a high density portion of the airspace. Contentious situations are thus avoided by design rather than by reacting to them when it could already be too late. This process shifts the greater part of separation responsibility to an off-line strategic phase of ATM and there are sound engineering reasons for doing this; Much greater computational (and/or human) resources can be dedicated to an off-line trajectory planner, and the choice of alternatives available to the trajectory synthesiser, is far greater than what can be devised in a few minutes before an imminent conflict.

SESAR seems to be moving in this direction with one of its more ambitious objectives being the introduction of the Business Trajectory. Coupled with SWIM, this will attempt to change the language of the ATM world, from sectors and segments to full 4D Trajectories. A universally accepted, efficient method of describing trajectories will also have to be devised before it can become the standard currency of exchange between aircraft, airlines and ANSPs.

At the core of the prototypical CATM-B system lies a generic algorithm that uses a robust global multi-objective dynamic optimiser to build a trajectory generator. There are few distinct optimisers that are completely appropriate and combinations of techniques are probably the best way forward.

Classical analytical techniques and most optimal control are based on variational calculus, which has been around for at least a century. They are built on Pontryagin's Minimum Principle (PMP). However, analytical methods are very limited in their utility to practical scenarios. Their numerical variant, Non-Linear Programming (NLP), has been shown to be far more generally applicable in both its *Indirect*, but more particularly in its *Direct* transcription forms [1.99]. A well known survey may be found in [1.100]. However, this class of solvers, though fast, is in any case susceptible to getting trapped in local optima and CATM is bound to be highly non-convex

However, a CATM-B solver needs to do more than this. Obviously, no amount of ATFCM-style pre-planning will ever completely remove the remnant uncertainty in the system and this is where the challenge lies. It needs to handle the fact that ATM is a dynamic optimisation problem. Besides providing air-traffic with a workable plan, it must maintain it in *dynamic equilibrium* with the constraints. It must seamlessly subsume an on-line branch to retain the means to dynamically adapt to an ever changing scenario, because the constraints (eg: weather) could realistically morph to the extent that the prepared plan is no longer optimal. In addition, besides tracking the optimum, it must determine whether the presumed optimum is indeed still *the global* optimum. A dynamic global optimiser is needed to *also* monitor the big picture and search for opportunities as they arise. It must gracefully react to external perturbation and this can only be done by employing a very long look-ahead horizon and then by continually making adjustments in order to satisfy some minimum measure of trajectory separation.

Although the last decade has brought with it some notable advances in optimisation techniques, achieving guaranteed global optimality in non-convex search spaces remains elusive to both theoreticians and practitioners. Yet a number of approximate techniques, based on AI or metaheuristics such as biomimetics have been applied in advanced robotics and look promising. Global optimisers are typically probabilistic and offer no guarantees of finding the best solution, but they are highly parallelizable and the greater the allocated resources, the higher the likelihood of getting good results. Given the computational capacity of current hardware, and modern computational intelligence, this is no longer a distant, intractable proposition, particularly if the combined effort of many aircraft are combined, and in a real-time environment, more than acceptable problem coverage can still be achieved. In the end, ATM was, and will always remain, a best effort attempt to find an efficient solution, even if not necessarily the best.

For instance, continental air traffic patterns are very reminiscent of insect swarms; the collective (including ANSPs) also behaves like a swarm of agents. In all likelihood, such a system should also be modelled and managed like the swarm it appears to be [1.101] to deal with its high dimensionality. This is based on the science of *Emergent Behaviour*, which has its roots in 19th century economic theory and 20th century game theory. It has received substantial attention over the past couple of decades for its role in cybernetics. In such systems, order is not achieved through explicit design. It emerges spontaneously as a result of the interaction between a plurality of agents. This is the basic tenet behind Swarm Intelligence [1.102] and various algorithms for ATM automation, based on these principles, are being devised and investigated as part of this work. These include Particle Swarm Optimisation (PSO) [1.103], Ant Colony Optimisation [1.104], and many other variants. These algorithms also fit nicely with evolutionary optimisation techniques and the results achieved so far are quite encouraging. It is expected that a combination of these ideas will form the basis for a practical multi-agent distributed solution to the ATM problem.

To pick another example, one approach in the quest for an over-arching ATM automation algorithm, is derived from Potential Field Theory. This concept traces its origins to a 1980 Ph.D. dissertation [1.105], and a 1985 seminal paper on obstacle

avoidance for mobile robotics [1.106] by Khatib and started gaining traction in the ATM domain after it was popularised by Martin Eby [1.46], and several others [1.47] in the mid 90's.

However, there are many other algorithms that deserve deeper investigation for the CATM problem. The incomplete list includes: memetic algorithms, swarm intelligence, genetic algorithms, simulated annealing, iterative dynamic programming, shooting methods, collocation, pseudo-spectral optimization methods and many more. But in summary, the most appropriate body of algorithms that are applicable to CATM-B are those amenable to high dimensionality, non-convex numerical *multi-trajectory optimisation* and the focus of research over the next decade or so needs to be geared on devising ways of combining techniques to efficiently and collaboratively generate trajectories for large numbers of interacting aircraft flying in a cluttered time-variant environment – in real-time and on standard hardware platforms.

1.7.1.3 LAYER C: Holistic Systemic Approaches

CATM-C takes the level of integration of the many subsystems to a whole new level, and depends on the existence of a satisfactory CATM-B algorithm. The higher level of integration required, results from the fact that ATM is a global conglomerate of various complex interacting systems. The high degree of coupling between systems raises the complexity but also provides avenues for improving efficiency.

Baggage handling systems, pricing of airline tickets, volcanic activity, airport capacity, regional politics and aircraft performance are all related in the end. The effectiveness and orderly interaction between these systems is what will ultimately filter through to the quality of service provided to the travelling customer. This is a rather scantily researched area and although the body of knowledge describing complex systems is rapidly growing, a comprehensive theoretical framework that is mature enough to offer useful engineering guidance to the CATM designer, is not even on the horizon.

For instance, airline scheduling has evolved into a refined art in the realm of operations research with many important results. So there is little point in addressing the trajectory synthesis problem in isolation. Any change in schedules will instantly invalidate the entire set of supposedly conflict-free trajectories. Hence, in the real world, an algorithm that encompasses the two aspects needs to be devised. In reality it needs to be broadened even further. So CATM-C must symbiotically combine trajectory synthesis with flight scheduling, ground operations, commuter demand, price structuring and other naturally-adjoined systems.

An excellent attempt to begin to make sense of the internal hierarchy inherent to all intelligent autonomous systems, can be found in the recent review paper by Veres *et al.* [1.107]. An autonomous system is viewed as a collection of interacting agents. Agent interaction and their method of interconnection is the key element for higher forms of CATM. The flow of information between agents, the establishment of egalitarian stable equilibria between conflicting requirements of separate agents, and the

dovetailing of the many interfaces, are other poorly addressed areas, worthy of much research.

Game theory classically provides the framework of choice when dealing with such multi-agent coordination problems. The PhD work by Waslander [1.108] explores the multi-vehicle collision avoidance sub-problem in the air traffic flow problem and addresses it using a decentralized, cooperative algorithm, which builds on results drawn from Economic/Game Theory. The work looks at the market mechanism as a radiant example of the emergent order and self organisation, which results from the competition between interacting agents (humans).

1.8 THE IMPLEMENTATION OF COMPUTATIONAL ATM

Drawing from the arguments made in previous sections, the shift from airspace-oriented to trajectory-oriented operations is necessarily built on trajectory optimisation, rather than airspace allocation. The major technical threshold will be the advent of *real-time, continuous, system-wide* and *grid connected* trajectory optimisation.

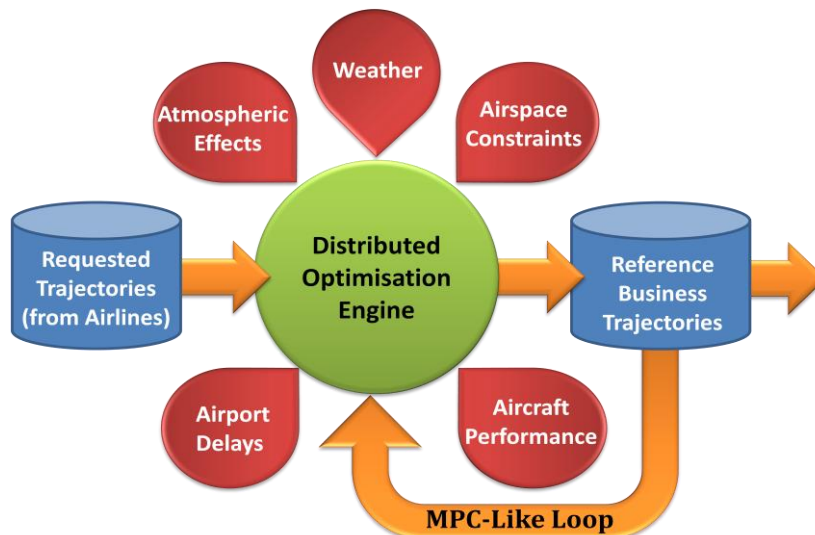


Figure 1.8.1: Top Level Architecture of CATM

One certainty is that the system will need to have an iterative nature in order to actively readapt to the constantly changing scenario (**Fig 1.8.1**). Since the input variables to the system are impossible to model exactly, due to their complex, apparently-random or chaotic nature, predictions of future inputs will only be valid for the short term – if at all. Stochastic models will have to be used to describe the rest. However, dynamic optimisation implies *continuous re-convergence* and is a hallmark feature of the underlying trajectory synthesis engine to keep track of the global optimum as the problem constraints drift.

Instead of synthesising and deploying a set of nominally conflict-free 4D trajectories, and then reacting to the unforeseen using (CD&R) tactical tools as proposed in SESAR or NEXTGEN, CATM uses an online optimiser to continually re-

adjust to the evolving scenario. This method keeps 4D trajectories well apart at all times and there will seldom be a conflict scenario to react-to tactically.

In line with the spirit of *free flight*, the computational burden of the online CATM system is borne by the cockpits on dedicated redundant hardware. This implies the creation of fast communication network between all aircraft such that they can keep each other updated with the latest events affecting the system as well as combine their computational effort additively in a way that solves the dynamic optimisation problem collaboratively.

1.9 THE TRANSITION TO CATM SYSTEMS

A final aspect that must not be overlooked is the transition between contemporary ATM to CATM. It is obvious that no such switch can happen overnight. The roll-out of CATM must be a planned affair spanning decades. There are many complexities that must be analysed and catered for but the following presents a general way how to go about it:

- Step 1:** Mandate CATM-compliant equipment in new aircraft while gradually retrofitting older aircraft. This computational equipment can be generic enough for it to be useful in the interim to house the software systems that manage SESAR and NextGen functions as well as CATM-A functionality that can be run synergistically.
- Step 2:** Initially CD&R would be introduced as a passive safety-net function, providing advisories (like TCAS II) [1.109] later as an ATC tool (like STCA/MTCD), and finally as a workload-reduction function with semi-automated conflict resolution.
- Step 3:** Extensive simulations, model checking, stochastic verification, followed by live flight trials would be conducted by the authorities in collaboration with the airlines to perfect the CATM-B software, before any commercial use.
- Step 4:** When significant aircraft is equipped, certain portions of the airspace can be segregated for exclusive use by CATM-B-compliant aircraft in the same way we classify airspace today: {VFR; IFR; CATM}. The CATM-B net-centric communications infrastructure and algorithms can be further tested and hardened during this period.
- Step 5:** The first unpowered CATM-B flights would take place using cargo aircraft between dedicated hubs over segregated flight levels. This gives time for the general public and regulatory bodies to build confidence in the system. It might take years and massive information campaigns to build such confidence.
- Step 6:** ATC is given access to all the detailed CATM-B flight plans and other aircraft will have to remain under ATC authority, which treats all CATM-B flights as fixed and uncooperative and keeps regular flights away from them.
- Step 7:** CATM-B Cockpits for public transport are initially reconfigured for single pilot operations and for a number of years one pilot, retrained as a flight engineer, is retained per cockpit. He will be responsible for executing emergency protocols to land the aircraft safely in the rare cases of unrecoverable system malfunction. Additional fall-back redundancy is achieved through the use of Remotely Piloted Aircraft Systems (RPAS). Teams of RPAS pilots are retained on the ground in the air traffic control centres, to remotely assist with non-nominal situations.
- Step 8:** After certification is rolled out for CATM-B public transport, the segregated airspace dedicated for CATM is gradually expanded over many years in relation to the percentage of equipped aircraft. ATCo numbers are reduced in proportion with the unequipped aircraft. Many will be retrained as CATM systems engineers.

- Step 9:** This period is also used to gradually phase out a proportion of the remaining pilot and ATCo workforce through early retirement schemes, and non-replacement of retirees. Eventually as the system matures, CATM-B aircraft will begin to fly pilotless. However, RPAS teams on the ground will be retained indefinitely.
- Step 10:** When the majority of the fleet is CATM-B compliant, the segregation is eliminated and the roles reverse whereby any remaining unequipped flights are now treated as uncooperative by the CATM-B system that uses all the surveillance tools at its disposal to track and avoid unequipped aircraft. Alternatively, the CATM network can provide an emulated Electronic ATC (e-ATC) service to unequipped aircraft, thereby taking control of all the flights in the system.
- Step 11:** Human piloted VFR flights will remain segregated indefinitely and a skeleton ATC service will be provided to this category only.
- Step 12:** The move to CATM-C can also be gradual, however this is not safety critical and can be rolled out in a piecemeal fashion as more ground and airline systems are interfaced with the CATM network, thereby increasing efficiency.
- Step 13:** Any modifications and updates to the CATM Software platform would be fully tested and re-certified in a simulated environment and can be rolled-out electronically over the CATM network. However, these would only be activated regionally, first to cargo aircraft, then the rest, but only when the aircraft is on the ground. This minimizes the risk associated with the unlikely event of a catastrophic malfunction.

1.10 THE OUTLOOK

The current SESAR and NextGen initiatives are undoubtedly necessary steps towards the modernisation of the ATC System. However, this is certainly not the end game. These are by no means going to provide us with the system we will need, to handle 2100 or even 2050 traffic. Small incremental changes and improvements to the current ATM system are justifiably rooted in good engineering practice, but they can only take us so far. In the meantime, a new autonomous ATM paradigm needs to be researched, devised and gradually phased in. But before any of this can happen, an in-depth understanding of the underlying science and engineering relevant to producing such systems is essential. This field is clearly still in its infancy and the more exciting times have yet to come.

Computational Air Traffic Management is envisaged to allow a large fleet of aircraft to self-organise for conflict-free flight, while taking into account external influences, disruption and internal variability as it occurs. In contrast to conventional ATC, CATM is a self-reliant synthetic *automaton*.

This unorthodox operational concept is part based on an extrapolation of the progress, which has occurred over the past three decades and builds on top of a number of fundamental requirements based on customer expectations rather than human controller limitations, or mere continuity with the past. A clean sheet approach has then been adopted in order to break free from the legacy that continues to confine the ATC community's imagination. This paves the way for a theoretical foundation for CATM to be developed. Further research in this direction needs to be pursued.

1.11 REFERENCES

- [1.1] B. Learmonth, J. Nash, D. Cluett (ed) "The First Croydon Airport (1915-1928)", London Borough of Sutton Libraries and Arts Services, Sutton Publishing; 1st ed. Edition, 1977, Nov, 4
- [1.2] IATA, "Safety Report 2013," 50th Edition, International Air Transport Association, Montreal—Geneva, ISBN 978-92-9252-349-7, 2014, Apr
- [1.3] Eurocontrol, "Challenges of Growth 2013 - Task 6: The Effect of Air Traffic Network Congestion in 2035," Report, European Organisation for the Safety of Air Navigation, Eurocontrol, 2013, Oct.
- [1.4] ICAO, "ICAO Environmental Report", Aviation Outlook, ICAO, 2010
- [1.5] FAA, "Instrument Procedures Handbook", Chapter 2: En Route Operations, U.S. Department of Transportation, Federal Aviation Administration (FAA), Flight Standards Service, Publication no. FAA-H-8083-16, 2014
- [1.6] FAA, "Pilot's Handbook of Aeronautical Knowledge", Chapter 14: Airspace, U.S. Department of Transportation, Federal Aviation Administration (FAA), Flight Standards Service, Publication no. FAA-H-8083-25A, 2008
- [1.7] M.A. Shamma, T.C. Nguyen, R.D. Apaza, "Frequency Reuse, Cell Separation, and Capacity Analysis of VHF Digital Link Mode 3 TDMA," NASA Glenn Research Centre, Cleveland, Ohio, 2003
- [1.8] Eurocontrol, "Eurocontrol Regional Charts (ERC)," <https://www.eurocontrol.int/articles/eurocontrol-regional-charts>, 2014, Oct, 16
- [1.9] Airbus, "Getting to Grips with Modern Navigation", Issue 4, Flight Operations Support & Line Assistance, Customer Services, Report: STL 945.0415/99 5, Blagnac, Cedex, France, 2002, Apr
- [1.10] FAA, "Advanced Avionics Handbook", Chapter 3: Navigation, U.S. Department of Transportation, Federal Aviation Administration (FAA), Flight Standards Service, Publication no. FAA-H-8083-6, 2009
- [1.11] D. Marsh, "Planning for Delay: influence of flight, scheduling on airline punctuality," EUROCONTROL Trends in Air Traffic, Volume 7, 2010
- [1.12] R.M. Baiada, "Free Flight: reinventing ATC-digital avionics a non-solution," In *Digital Avionics Systems Conference, DASC, The 14th AIAA/IEEE*, 1995, Nov. pp. 100-105
- [1.13] K. L. Bowers, "Determining the feasibility of a flight profile in a *Free-flight* environment," *Digital Avionics Systems Conference, DASC, The 15th AIAA/IEEE*, Atlanta, GA, 1996, Oct 27-31, vol. 1, pp. 81-86
- [1.14] B.K. Obbink, "Description of Advanced Operation: *Free-flight*," *WP9: Risk assessment for a distributed control system*, HYBRIDGE Project, EC Contract: IST-2001-32460, 2005, Mar 7
- [1.15] J. M. Hoekstra, R.C.J. Ruigrok and R.V. Gent, "*Free-flight* in a crowded airspace?," *Progress in Astronautics and Aeronautics Series*, vol. 193, pp. 533-546, 2001
- [1.16] M.R. Jardin, "Toward Real-Time En Route Air Traffic Control Optimization," *PhD, Dept. of Aeronautics & Astronautics*, Stanford University, Stanford, CA, USA, 2003
- [1.17] J. Krozel, M. Peters, K. D. Bilimoria, C. Lee, & J. Mitchell, "System performance characteristics of centralized and decentralized air traffic separation strategies". In Fourth USA/Europe Air Traffic Management Research and Development Seminar, Dec 2001
- [1.18] M.R. Jardin, "Analytical Relationships between Conflict Counts and Air Traffic Density," *AIAA Journal of Guidance, Control, and Dynamics*, vol. 28, no. 6, pp. 1150-1156, 2005
- [1.19] Eurocontrol & FAA, "Comparison of Air Traffic Management-Related 2012 Operational Performance: U.S./Europe," Eurocontrol Performance Review Commission, FAA Air Traffic Organization System Operations Services, 2013, Nov.
- [1.20] IATA, "IATA Economic Briefing: Inefficiency in European Airspace," IATA Economics, 2013
- [1.21] ATAG, "Aviation Benefits Beyond Borders," Air Transport Action Group, (ATAG), Geneva, Switzerland, 2014, Apr.
- [1.22] IATA, "Fact Sheet: Industry Statistics", Industry Financial Forecast Table, IATA Economics, International Air Transport Association, IATA, 2014, Dec.

- [1.23] Space Daily, "Airbus, Boeing project commercial aviation needs", http://www.spacedaily.com/reports/Airbus_Boeing_project_commercial_aviation_needs_999.html, Paris, 2013, Sep. 25
- [1.24] ICAO, "World Aviation and the World Economy," Facts and Figures, International Civil Aviation Organization, (ICAO), 2013
- [1.25] R.T. Carson, T. Cenesizoglu and R. Parker, "Forecasting (aggregate) demand for US commercial air travel," *International Journal of Forecasting*, vol. 27, no. 3, pp. 923-941, 2010
- [1.26] GlobeScan Incorporated, "Greendex - Consumer Choice and the Environment Market Basket Report", *Report for the National Geographic*, Toronto, Canada, 2009, May
- [1.27] D. Marsh, "A Matter of Time: Air Traffic Delay in Europe," *Trends in Air Traffic, Volume 2*, European Organisation for the Safety of Air Navigation (EUROCONTROL), Brussels, Belgium, 2008
- [1.28] J. Hoekstra and R. Curran, "What are the foundations of ATM?," *ASDA Seminar*, Association for the Scientific Development of ATM in Europe, Technical University of Delft, The Netherlands, 2009 Nov 5-6
- [1.29] W.C. Moon, K.E. Yoo and Y.C. Choi, "Air Traffic Volume and Air Traffic Control Human Errors," *Journal of Transportation Technologies*, vol 1, pp 47-53, 2011
- [1.30] G. Bisignani, "IATA - State of the Air Transport Industry," *66th IATA Annual General Meeting and World Air Transport Summit*, Berlin, Germany, 2010, Jun 7
- [1.31] NATS, "Air Traffic Control Disruption on 7th December 2013," A Report to the Civil Aviation Authority", Final Version Report , 2014, Jul. 3
- [1.32] D. Milliken & L. Tuhkanen, "Air traffic control failure disrupts flights over Britain," Reuters News Agency, <http://www.reuters.com/article/2014/12/12/us-britain-airlines-closure-idUSKBN0JQ1M620141212>, London, 2014, Dec. 12
- [1.33] BEA, "Final Report: on the accident on 1st June 2009, to the Airbus A330-203, registered F-GZCP, operated by Air France, flight AF 447 Rio de Janeiro - Paris", France, 2012, Jul., 27
- [1.34] S. S. Yeh, "The Mystery of Malaysia Airlines Flight 370", University of Minnesota-Twin Cities, http://works.bepress.com/stuart_yeh/11, 2014, Apr. 26, Accessed May 2014
- [1.35] F. Podiani, C. Fusai and P. Marti, "Mediterranean *Free-flight* - Un Programma Europeo di Ricerca Pre-operativa sulle Applicazioni ASAS," *MIMOS - 4° Convegno Tecnico Scientifico*, Torino, Italy, 2004, Oct
- [1.36] F. Casaux, "Airborne Separation Assistance System - The ASAS Concept," *SSR Improvements and Collision Avoidance Systems Panel (SICAS/WG2/489)*, Sydney, Australia, 1995, Mar. 11
- [1.37] IFATS, "Publishable Final Activity Report of the IFATS project", Report: IFATS/ONERA/07/D700.9-PSR, IFATS EU Project Consortium, 2007, Jul. 30
- [1.38] S. Guibert, L. Guichard, C. Rihacek, & J.Y. Grau, "Result from evaluation of 4d trajectory management with contract-of-objectives." CATS Project, In Digital Avionics Systems Conference, DASC'09. IEEE/AIAA 28th, IEEE, 2009, Oct., pp. 3-D.
- [1.39] J.T. Chen, D. Andrisani, J. Krozel, & J.S.Mitchell, "Flexible Tube-based Network Control," AIAA Guidance, Navigation, and Control Conference, AIAA 2009-5861, Chicago, Illinois, 2009, Aug, 10-13,
- [1.40] H. Erzberger, "The Automated Airspace Concept," *USA/Europe Air Traffic Management R&D Seminar, 4th*, Santa Fe, New Mexico, USA, 2001, Dec 3-7
- [1.41] R. Holdsworth, J. Lambert, & N. Harle, "Inflight path planning replacing pure collision avoidance, using ADS-B," *Aerospace and Electronic Systems Magazine*, IEEE, vol. 16, no. 2, 2001, pp. 27-32
- [1.42] NextGen, "Nextgen Implementation Plan," Washington, DC, USA, *Public Report*, Federal Aviation Administration (FAA) , 2010
- [1.43] SESAR, "European ATM Master Plan," Brussels, EU, *Public Report*, EuroControl, 2009
- [1.44] O. Pinon, D. Mavris and E. Garcia, "A Visual Analytics Approach to the Qualitative Comparison of the SESAR and NextGen Efforts," *9th AIAA Aviation Technology, Integration, and Operations Conference (ATIO)*, Hilton Head, SC, USA, 2009

- [1.45] J.S. Meserole and J. Moore, "What is System Wide Information Management (SWIM)?," *Aerospace and Electronic Systems Magazine*, IEEE, vol. 22, no. 5, pp. 13-19, 2007
- [1.46] M.S. Eby, "A self-organizational approach for resolving air traffic conflicts," *Lincoln Laboratory Journal*, vol. 7, no. 2, pp. 239-254, 1994
- [1.47] K. Zeghal, "Airborne Conflict Detection and Resolution Using Coupled Force Fields: Principles and Results," *AGARD/NATO Workshop on Air Traffic Management*, Neuilly-sur-Seine, France, 1997, May 27-29, vol. 1, pp. 15-1-15-11
- [1.48] G., Chaloulos, P. Hokayem, & J., Lygeros, "Distributed hierarchical MPC for conflict resolution in air traffic control," In American Control Conference (ACC), IEEE. Jun, 2010, pp. 3945-3950
- [1.49] X. Zhang, X. Guan, I. Hwang, & K. Cai, "A hybrid distributed-centralized conflict resolution approach for multi-aircraft based on cooperative co-evolutionary," *Science China, Information Sciences*, 2013, vol. 56, no. 12, pp. 1-16
- [1.50] M.A. Azzopardi and J.F. Whidborne, "Computational Air Traffic Management", *Proceedings of the 30th AIAA/IEEE Digital Avionics Systems Conference (DASC2011)*, IEEE/AIAA, Seattle, WA, USA, 2011, Oct 16-20, pp. 1.B.5-1.
- [1.51] E.K. Burke, P.D. Causmaecker, G.D. Maere, J. Mulder and M. Paelinck, "Forecasting flight schedule robustness," *Proceedings of the 2nd MISTA Conference*, New York, 2005, Jul 18-21, vol. 2, pp. 698-700
- [1.52] Boeing, "Airports with Noise and Emissions Restrictions", Seattle, Washington, USA, <http://www.boeing.com/commercial/noise/list.html>, 2014, Nov, Accessed: Feb. 2014
- [1.53] CAMU, "Enhanced ATFCM Operator's Manual", Version 2, Central Airspace Management Unit, Johannesburg, Rep of S. Africa, 2010, Apr, 1
- [1.54] E.K. Burke, P.D. Causmaecker, G.D. Maere, J. Mulder, M. Paelinck and G.V. Berghe, "A multi-objective approach for robust airline scheduling," *Computers & Operations Research*, vol. 37, no. 5, pp. 822-832, 2010
- [1.55] S. AhmadBeygi, A. Cohna and M. Lappa, "Decreasing airline delay propagation by re-allocating scheduled slack," *IIE Transactions*, vol. 42, no. 7, 2010
- [1.56] J. Kennedy, R.C. Eberhart and Y. Shi, "Swarm Intelligence," *The Morgan Kaufmann Series in Evolutionary Computation*, San Francisco, CA, USA, Morgan Kaufmann Publishers, 2001
- [1.57] ICAO, "Global Air Traffic Management Operational Concept (Doc 9854 AN/458)," International Civil Aviation Organization (ICAO), 2005
- [1.58] H.J. Hof, "Development of a Performance Framework in support of the Operational Concept," *ICAO Mid Region Global ATM Operational Concept Training Seminar*, Cairo, Egypt, 2005, Nov 28-Dec 1
- [1.59] A. Messac, A. Ismail-Yahaya, C.A. Mattson. "The normalized normal constraint method for generating the Pareto frontier." *Structural and Multidisciplinary Optimization*, 2003, vol. 25, no. 2
- [1.60] HYBRIDGE Project: "Distributed Control and Stochastic Analysis of Hybrid Systems Supporting Safety Critical Real-Time Systems Design: Final Project Report", Framework Protocol 5 (FP5) Project, IST-2001-32460, 2000 - 2005
- [1.61] iFly Project: "Safety, Complexity and Responsibility based design and validation of highly automated Air Traffic Management: Final Report and Recommendations", Framework Protocol 6 (FP6) Project, TREN/07/FP6AE/S07.71574/037180, 2007-2011
- [1.62] MoVeS Project: "Modeling, Verification and Control of Complex Systems: Final report on modeling, analysis, impact and potential exploitation", Framework Protocol 7 (FP7) Project, FP7-ICT-2009-257005
- [1.63] T.A. Henzinger, "The Theory of Hybrid Automata", 11th Annual Symposium on Logic in Computer Science (LICS), 1996, IEEE Computer Society Press, pp. 278-292
- [1.64] W. Glover, & J. Lygeros, "A stochastic hybrid model for air traffic control simulation." In *Hybrid Systems: Computation and Control*, Springer Berlin Heidelberg, 2004, pp. 372-386
- [1.65] A. Abate, M. Prandini, J. Lygeros, S. Sastry, "Probabilistic reachability and safety for controlled discrete time stochastic hybrid systems," *Automatica*, vol. 44, no. 11, pp. 2724-2734, 2008

- [1.66] A. Abate, J.P. Katoen, J. Lygeros, & M. Prandini, "Approximate model checking of stochastic hybrid systems." *European Journal of Control*, vol.16 no. 6, pp. 624-641, 2010
- [1.67] C.J. Tomlin, G.J. Pappas, S. Sastry, "Conflict resolution for air traffic management: A study in multiagent hybrid systems," *Automatic Control, IEEE Transactions on*, vol. 43, no. 4, pp. 509-521, 1998
- [1.68] J. Lygeros, D.N. Godbole, S. Sastry, "Verified hybrid controllers for automated vehicles," *Automatic Control, IEEE Transactions on*, vol. 43, no. 4, pp. 522-539, 1998
- [1.69] J. Lygeros, C.J. Tomlin, S. Sastry, "Controllers for reachability specifications for hybrid systems," *Automatica*, vol. 35, no. 3, pp. 349-370, 1999
- [1.70] G.AO Yan, J Lygeros, "On the reachability problem for uncertain hybrid systems," *Automatic Control, IEEE Transactions on*, vol. 52, no. 9, pp. 1572-1586, 2007
- [1.71] J. Lygeros, D.N. Godbole, and S. Sastry, "A game-theoretic approach to hybrid system design," *Hybrid Systems III, Lecture Notes in Computer Science*, Springer Berlin, vol. 1066, pp. 1-12, 1996
- [1.72] C.J. Tomlin, J Lygeros, S. Sastry, "A game theoretic approach to controller design for hybrid systems," *Proceedings of the IEEE*, vol. 88, no. 7, pp. 949-970, 2000
- [1.73] G. Chaloulos, P. Hokayem, J. Lygeros, "Distributed hierarchical MPC for conflict resolution in air traffic control." In *American Control Conference (ACC)*, pp. 3945-3950, IEEE., 2010, Jun
- [1.74] V.P. Galotti, "The Future Air Navigation System (FANS): Communications Navigation Surveillance Air Traffic Management," UK, Ashgate Publishing, 1997
- [1.75] M. Ballin, J. Hoekstra, D. Wing, and G. Lohr, "NASA Langley and NLR research of Distributed Air/Ground Traffic Management." *Proceedings of the AIAA Aircraft Technology, Integration, and Operations Conference*, AIAA-2002-5826. Los Angeles, CA., 2002
- [1.76] SESAR, "System Wide Information Management (SWIM)," Sesar Fact sheet, Document N° 01/2011
- [1.77] J. Krozel and M. Peters, "*Free-flight* research issues and literature search," *NASA Ames Research Center Report*, Moffett Field, CA, USA, 2000
- [1.78] RTCA, "Final report of the RTCA board of directors' select committee on *free-flight*." Report: CTF-3, RTCA Task Force 3, Radio Technical Commission for Aeronautics (RTCA), Washington, DC, 1995, Oct
- [1.79] R.C.J. Ruigrok and J.M. Hoekstra, "Human factors evaluations of *Free-flight*: Issues solved and issues remaining," *Applied Ergonomics*, vol. 38, no. 4, pp. 437-455, 2007
- [1.80] S. Zerkowitz, "The need for a new ATM operational concept: A strategic necessity?," *Digital Avionics Systems Conference, DASC, The 20th AIAA/IEEE*, 2001, Oct 14-18, vol. 2, pp. 7E6/1-6
- [1.81] R.A. Paielli and H. Erzberger, "Conflict probability estimation for *free-flight*," *Aerospace Sciences Meeting and Exhibit*, 35th, Reno, NV, USA, 1997, Jan. 6-9
- [1.82] R.V. Gent, J.M. Hoekstra and R.C.J. Ruigrok, "Conceptual design of *Free-flight* with airborne separation assurance," *NLR Report NLR-TP-98252, Proceedings of the 10th European Conference, Free-flight*, Amsterdam, 1997
- [1.83] S. Lozito, A. McGann, M. Mackintosh, & P. Cashion, "*Free-flight* and Self-Separation from the Flight Deck Perspective", NASA Ames Research Center, In *The First United States/European Air Traffic Management Research and Development Seminar*, Sacley, 1997, Jun
- [1.84] A. Beadle, "A Statement on the Future of Global Air Traffic Management," Version 1.0, IFATCA, 2007, Feb 27, www.ifatca.org, Accessed: Jan 2014,
- [1.85] U. Metzger and R. Parasuraman, 2001, "The Role of the Air Traffic Controller in Future Air Traffic Management: An Empirical Study of Active Control versus Passive Monitoring," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 43, pp. 519-528
- [1.86] R. Parasuraman, "Vigilance, Monitoring and Search," In: *Handbook of Perception and Human Performance - Cognitive processes and performance*. Edited by K. Boff LK, J. Thomas, vol. 2. New-York: John Wiley & Sons, pp43.1-43.39, 1986
- [1.87] R. Kebabjian, Aug, "Plane Crash Statistics", 2013, Dec, www.planecrashinfo.com, Acc: Mar 2014

-
- [1.88] Boeing, "Airplanes Statistical Summary of Commercial Jet Airplane Accidents Worldwide Operations (1959-2009, 2009 Statistical Summary, Aviation Safety, Boeing Commercial Airplanes)", Seattle, Washington, USA, July 2010
- [1.89] K. Zeghal and E. Hoffman, "Delegation of separation assurance to aircraft: towards a framework for analysing the different concepts and underlying principles," *International Council of the Aeronautical Sciences Congress (ICAS)*, Harrogate, UK, 2002
- [1.90] W.E. Kelly III, "Conflict detection and alerting for separation assurance systems," *Digital Avionics Systems Conference, DASC, The 18th AIAA/IEEE*, St Louis, MO, 1999, Oct, vol. 2, pp. 6.D.1.1-8
- [1.91] M. Prandini, L. Piroddi, S. Puechmorel and S.L.Brázdilová, "Toward air traffic complexity assessment in new generation air traffic management systems." *Intelligent Transportation Systems, IEEE Transactions on*, 12(3), 809-818. 2011
- [1.92] J. S. Morrel, "The mathematics of collision avoidance in the air," *Journal of Navigation*, The Royal Institute of Navigation, vol. 11, no. 01, pp. 18-28, 1958
- [1.93] J. Krozel, "Intent inference for *free-flight* aircraft," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Denver, CO, USA, 2000, Aug. 14-17, vol. 3, pp. 1715-1725
- [1.94] R. Irvine, "Comparison of pair-wise priority-based resolution schemes through fast-time simulation," *8th Innovative Research Workshop and Exhibition (INO2009)*, Brétigny-sur-Orge, France, 2009, Dec 1-3
- [1.95] C. Munoz, R.W. Butler, V.A. Carreño & G. Dowek, "On the Formal Verification of Conflict Detection Algorithms," *Report NASA/TM-2001-210864*, NASA Langley Research Ctr., Hampton, VA, 2001
- [1.96] A.L. Galdino, C. Munoz and M. Ayala-Rincon, "Formal verification of an optimal air traffic conflict resolution and recovery algorithm," *Logic, Language, Information and Computation*, in *Lecture Notes in Computer Science Volume 4576*, pp 177-188, 2007
- [1.97] J.K Kuchar and L.C. Yang, "A review conflict detection and resolution modeling methods," *IEEE Trans. on Intelligent Transportation Systems*, vol. 1, no. 4, pp. 179-189, 2000
- [1.98] E.N. Lorenz and R.C. Hilborn, "The Essence of Chaos," *Amer. Jour. of Phys.*, vol. 63, pp. 862, 1995
- [1.99] V. T. Taranenko, and V.G. Momdzhii, "Direct variational method in boundary tasks of flight dynamics," Mashinostroenie Press, Moscow, Russia, 1986
- [1.100] J.T. Betts, "Survey of numerical methods for trajectory optimization," *Journal of Guidance Control and Dynamics*, vol. 21, no. 2, pp. 193-207, 1998
- [1.101] S. Alam, M. H. Nguyen, H. A. Abbass, and M. Barlow, "Ants guide future pilots," *In Progress in Artificial Life*, Springer Berlin Heidelberg, pp. 36-48, 2007
- [1.102] A.P. Engelbrecht, "Fundamentals of Computational Swarm Intelligence," John Wiley and Sons Ltd, *University of Pretoria*, South Africa, 2005
- [1.103] J. Kennedy and R. Eberhart, Nov.27-Dec.1, "Particle Swarm Optimization," *IEEE International Conference on Neural Networks*, vol. 4, pp. 1942-1948, 1995
- [1.104] M. Dorigo and G. Di Caro, "Ant Colony Optimization: A New Meta-Heuristic," *1999 Congress on Evolutionary Computation [CEC'99]*, pp. 1470-1477, 1999
- [1.105] O. Khatib, "Commande dynamique dans l'espace opérationnel des robots manipulateurs en présence d'obstacles," *Ph.D., École Nat. Sup. de l'Aéronatique et de l'Espace (ENSAE)*, Toulouse, France, 1980
- [1.106] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *Robotics and Automation, Proceedings. IEEE International Conference on*, vol. 2, pp. 500-505, 1985, Mar.
- [1.107] S.M. Veres, L. Molnar, N.K. Lincoln and C.P. Morice, 2010, "Autonomous vehicle control systems – a review of decision making," *Proc. IMechE Part I: J. Systems and Control Engineering*, vol. 225, no. 2, pp. 155-195
- [1.108] S.L., Waslander, "Multi-Agent System Design for Aerospace Applications," PhD Dissertation, *Department of Aeronautics and Astronautics*, Stanford University, CA, USA, 2007
- [1.109] J.K. Kuchar and A.C. Drumm, "The traffic alert and collision avoidance system," *Lincoln Laboratory Journal*, vol. 16, no. 2, pp. 277-296, 2007
-

Chapter 2

Computational Air Traffic Management

This chapter provides the big picture. We will start by describing the conceptual components of an airborne, decentralized, fault-tolerant paradigm for a continental (and eventually global) ATM system that readily scales up to accommodate projected future aircraft densities, involving military, commercial, (autonomous or piloted), airborne systems. Parts of the discussion at this stage might seem speculative, however, it will be substantiated in later chapters in much more detail. The system is broadly consistent with the proposed objectives of both NextGen (Next Generation Air Transportation System) and SESAR (Single European Sky ATM Research). Several subsystems are classified, and discussed in the context of emerging computational and communications hardware. The case is then made for a new breed of grid-oriented avionics hardware that will enable aircraft to take CATM to the skies by leveraging on the many benefits brought by pervasive inter-aircraft ad-hoc communication networks and heterogeneous grid-computing.

The overarching concept described hereunder is in itself one of the main contributions of this work and this chapter lays down the context in which specific technical contributions are described in the subsequent chapters.

2.1 A NEW CONCEPT FOR ATM

Computational Air Traffic Management has been presented as the discipline that studies the emergent mathematical order which results from the collective behaviour of a number of interacting systems that make up a hypothetical future air transport system. It focuses on addressing several major problems related to the current aviation paradigm. Open problems such as rising congestion, delays, lost aircraft, pilot error, hijacking, volcanic ash susceptibility, high ATC cost, mid-air collisions, environmental impact and some kinds of hardware failure can be mitigated or solved with a neat overarching solution that leverages best-in-class technology.

The CATM name derives from the fact that computational intelligence can (and probably must) be used to take on the centre role in solving this well known,

- Large
- dynamic,
- variable size,
- infinite horizon,
- multi-parameter,
- highly constrained,
- nonlinear,
- non-causal,
- multi-modal,
- multi-objective,
- high-dimensionality,
- continuous-and-combinatorial, *optimization* problem.

The indications are that whatever the formulation, the CATM problem is non-deterministic in polynomial time (NP-Hard) on Deterministic Turing Machines (DTMs) [2.1], [2.2]. Although, one of the hardest problems known, we still believe that it can be made tractable using a multi-disciplinary approach that finds good approximate solutions based on certain simplifications and a careful use of meta-heuristics, so that fast convergence can be achieved using currently available computing technology.

The principles of swarm intelligence are the foundations of what we are about to discuss. The concept is the result of a clean sheet approach that began with a rethink of what really matters in an ideal air transport system [2.3]. This contrasts sharply with the methodology that the ATM R&D community usually adopts. However, laying down the principles of a coherent concept, helps define an ideal scenario that could well serve to guide future incremental development.

CATM is envisaged to spawn a practical, comprehensive and self-consistent set of ATM (and ancillary) algorithms that can be used to allow a large fleet of aircraft to *self-organise* for conflict-free flight, while taking into account external influences, disruption and internal variability as it occurs. In contrast to conventional ATC, a CATM-based system is a self-reliant synthetic *automaton*.

The shift from airspace-oriented to trajectory-oriented ATM operations paves the way for *computational* techniques that are based on trajectory optimisation, rather than airspace allocation. The major technical threshold will be the advent of *real-time*, *continuous*, and *system-wide* trajectory optimisation. CATM can provide an integrated solution that merges both strategic phases (ATFCM) and tactical phases (ATC) into a common framework that no longer needs to distinguish between the two. The best features of the current air traffic system are retained and combined with a variety of novel concepts that enhance capacity, performance and reliability.

A basic tenet of CATM is that *through the provision of sufficient computational, communication and surveillance resources, the system update rate can be realistically increased to the point, where over the short duration of each system time-step, the traffic scenario advances by a negligibly small amount* – and this changes many things.

At these time scales, the distinction between ATFCM and ATC blurs. The difference between the purely tactical and strategic phases degenerates. Most phenomena that can affect the air traffic scenario, can now be predicted with sufficiently high certainty over an appreciable number of time-steps. So in a sense, the tactical aspect is absorbed into a single pseudo-strategic planning process which has a continuous span of influence that ranges from a short term horizon of a few seconds, to a much longer term horizon lasting many months. Therefore by raising the system update rate, CATM encompasses the roles of both ATM and ATC.

The following sections explore the consequences of CATM adoption on the logical and physical architecture of a future grid connected avionics system.

2.1.1 Computational Complexity Considerations

Casting a highly non-convex problem such as CATM as an NLP problem faces us with another difficulty. Directed searches will typically get trapped in the first local optimum they encounter. Systematically subdividing the search space for subsequent local searches quickly runs into a wall. Complexity theory [2.4] tells us that as the number of managed aircraft increase, the exponential combinatorial expansion of the highly non-convex search space, makes systematic deterministic approaches to reach the global optimum, unfeasible on Deterministic Turing Machines.

In structured airspace, even the multi-airport, multi-sector, air traffic flow and capacity management (ATFCM) sub-problem, has been shown to be NP-hard and equivalent to job-shop scheduling in terms of complexity [2.5]. It is actually an NP-Complete combinatorial optimisation problem [2.6]. In a free-flight scenario, the problem formulation changes somewhat and several important observations can be made:

- With sectorisation removed, it shifts part of the problem to the continuous domain, vastly increasing the search space. However, this also increases the “good solution” space.
- Optimizing airport utilization remains firmly discrete and combinatorial in nature and remains NP-hard, and so is assigning airspace to trajectories.
- The combination of continuous and discrete optimisation makes this a hybrid system with all the associated complexities [2.7].
- Optimizing a single isolated trajectory in uniform airspace between two fixed points is convex, but nonlinear for virtually all aircraft types.
- Adding trajectories, increases the problem dimensionality linearly, but the search space increases exponentially.
- When taking into account realistic fixed-wing aircraft performance and airline objectives, it also becomes multi-objective and multi-parameter.
- The nonlinearity and problem size worsens dramatically if the search space is shifted to the aircraft control input side, but this can be avoided if dynamic inversion can be used.
- Aircraft performance constraints add another level of complexity with some applicable to the vehicle output space and some in the control input space.
- When obstacles such as restricted airspace are introduced, the search space becomes not only non-convex but also disconnected, with feasible islands surrounded with unfeasible search spaces.
- When the obstacles are moving (eg bad weather), or when new flights are added or removed, the problem becomes also dynamic. This means that the global optimum is likely to move or lose its status altogether over time and needs to be tracked continuously and reevaluated periodically with respect to other local optima.

2.1.2 Relaxing Optimality Requirement to Address Complexity

Non-deterministic Turing Machines (NTMs) are theoretically efficient at solving NP-Hard problems in polynomial time. Although such machines can be inefficiently emulated on DTMs (at great computational cost), no such machine has ever been constructed and nor is it likely to [2.8]. One feasible alternative is to consider the computation of non-deterministic optimisation algorithms directly on deterministic machines. Metaheuristics and memetic algorithms fall under this category [2.9]. These approaches use a stochastic process to broaden the search and rely on heuristics to negotiate the search space efficiently, but bear the limitation that no guarantees of reaching a global optimum can be given. With the addition of a stochastic element, the probability of reaching one of the lesser (but acceptable) solutions can be made arbitrarily high by dedicating the right amount, of the right kind, of computational resources. Any one of these solutions would still represent a major improvement over what is achievable today with human brain power alone.

2.1.3 Nested Optimisation

Gradient-based directed optimization is fast and reliable when dealing with smooth search spaces due to its good use of the information contained in the search landscape. It is however confined to searching local optima. Stochastic and metaheuristic optimisation are on the other hand much more exploratory and better suited at identifying regions containing global optima in non-convex problems. They are fairly quick at identifying these regions however they are notoriously slow to converge thereafter. Most time is spent refining local optima, where they become inefficient due to their poor use of local gradient information.

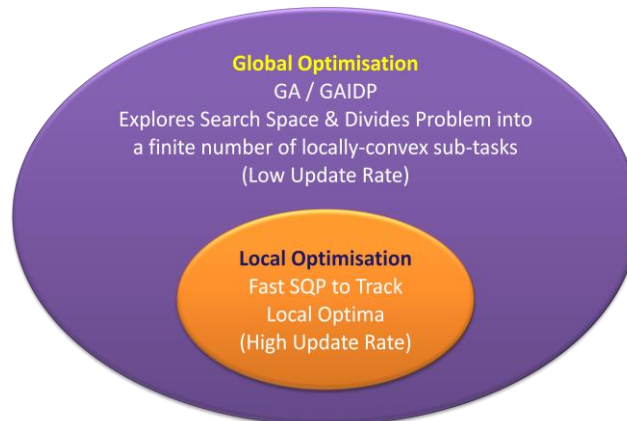


Figure 2.1.1: Nested Optimization - Best of both worlds

Local and global optimisation techniques have complementary strengths and therefore, one approach to this conundrum is to combine the techniques [2.10] (**Figure 2.1.1**), with an outer global optimizer robustly scouting the landscape, and nested inner local optimizers for accurately and rapidly evaluating the global potential of local optima as they are found. Such nested optimisation has been used to solve very large problems in aerospace engineering applications [2.11].

2.1.4 The benefits of Dynamic Re-Convergence

Most global optimisers assume a static search space and substantial computational effort is expended to reach a global optimum. However, when the problem changes, the search effort has to be restarted from the beginning, which is computationally inefficient. The ATM problem is constantly changing and therefore the converged state of the optimiser must be in a dynamic equilibrium with the problem constraints. What is needed is a *Continuous Dynamic Re-convergence*. This is the hallmark feature of CATM's underlying trajectory synthesis engine (**Figure 2.1.2**), which must keep track of the changing inputs and dynamic aspect of the problem.

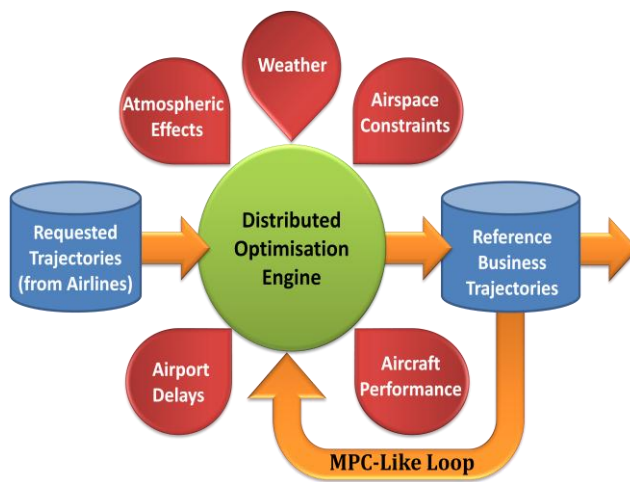


Figure 2.1.2: The Continuous Nature of CATM

and resolution (CD&R) tactical tools, CATM uses an online optimiser to continually re-adjust to the evolving scenario. This method keeps 4D trajectories well apart at all times and there will seldom be a conflict scenario which needs to be addressed tactically.

Given the nature of most numerical optimisers, there are also sound mathematical reasons for going about it this way. Most numerical optimisers are able to re-converge *much* faster and *much* more reliably, once provided with a good seed solution. The only sure way to guarantee that the next iteration re-converges rapidly to the same local optimum is to seed it with the previous solution. If the iteration time step is short enough, the problem domain would have evolved by a very limited amount in the interim. Hence, the solution to the next iterand is bounded to a region fairly adjacent to the solution of the previous iterand. *A high update rate keeps the inner algorithm of the trajectory synthesiser operating efficiently in a locally-convex domain thereby tracking local optima.* A sequential quadratic programming (SQP) local optimizer, for instance, would reconverge within one or two Newton steps.

However, in a dynamic optimization problem, the gradually evolving scenario might push a local optimiser to erroneously track a local optimum that may no longer coincide with *the* global optimum – assuming it was a global optimum to begin with.

Consider this scenario: A large weather cell may be causing most traffic to detour around its north side. As the cell gradually moves north, the trajectories will get

Since the input variables to the system are impossible to model exactly, due to their complex and apparently-random or chaotic nature, predictions of future inputs will only be valid for the short term – if at all. Stochastic techniques will have to be used to describe what can't be predicted with any certainty.

Instead of synthesizing and deploying a set of nominally conflict-free 4 Dimensional (4D) trajectories, and then reacting to the unforeseen using conflict detection

deflected further around it, and perhaps become unreasonably extended, until a situation develops where it becomes more economical to deviate around the cell, on its southern side. ATM Automation must be able to deal with such pathological cases. Actively tracking the motion of local convexity, allows optimizers to deal with real-time non-convex dynamic optimization problems [2.12], but this alone is insufficient.

To mitigate this risk, iteration is needed at both levels of the nested hybrid optimiser. The global optimiser also needs to periodically reassess the evolving scenario, to monitor the big picture and search for opportunities as they arise, albeit at lower update rates.

2.1.5 Constraint modelling in CATM

While simple geometry shows that the shortest trajectory for any aircraft travelling over a spherical earth would be to follow a simple great-arc route, real world effects means this is not possible in practice. Many effects, both external and internal, pose constraints that reduce the feasible solution space dramatically. The CATM optimiser makes use of a number of models to predict the constraints of the dynamic problem into the future. Some of these models are fixed while others change over time.

2.1.5.1 Aircraft models

A core component of the optimization process is the aircraft model itself. It guides the optimizer to generate flyable trajectories that conform to the dynamic constraints unique to each aircraft. The Base of Aircraft Data (BADA) is a database maintained by Eurocontrol [2.13]. In its latest form, it uses a parameterized total energy model to describe the performance and flight envelope of 405 aircraft or 90% of all aircraft in use in European airspace. A newer version with much better accuracy is currently in development.

2.1.5.2 Atmospheric models

Aircraft modelling goes hand in hand with atmospheric modelling as it bears on aircraft performance in a major way. The International Standard Atmosphere (ISA) [2.14] is an empirical model developed by the International Civil Aviation Organization and is considered appropriate for ATM purposes. This generally means that fast, fuel-efficient cruising must take place at high altitude where the air is rarefied.

2.1.5.3 Airspace models

Every national airspace is characterized by a number of imposed geographical, altitude and operational restrictions. (Instrument and visual flying rule zones, special notifications, military training ranges, no fly zones over nuclear reactors and penitentiaries etc...). These spatial constraints need to be assembled in a database in a format which is computationally efficient for the optimizer to access.

2.1.5.4 Weather models

Weather is one of the major sources of uncertainty in the system. The presence of wind, for instance, determines whether the shortest distance path will also be the most fuel efficient. An accurate weather model that can extrapolate current conditions by a few days is critically essential for the optimizer to generate trajectories with sufficient confidence. The accuracy of such predictions worsens as the prediction is extended further into the future, however this is compatible with the requirements of CATM.

2.1.5.5 Airport delay models

The greatest source of uncertainty in a flight is arguably its departure time. Such delays are hard to model in a deterministic way and can only be partially predicted in the immediate run up to a flight. If a delay has been caused by a late incoming aircraft, the system might be in a position to factor that in, otherwise stochastic models can be used.

2.1.6 Decentralised CATM

Everything in CATM revolves about the need for substantially higher system update rates. This can, in turn, only be accommodated by pooling the computational resources across many aircraft to form a high performance computing grid that collaborates to generate a decentralised communal solution. This provides an excellent opportunity to combine the situational, environmental and atmospheric data gathered by all the aircraft in the pool.

Much has been published about the merits and demerits of distributed or decentralised control and the reader is invited to view [2.17] - [2.27] for an in depth treatment. Although related, distributed control must not be confused with the notion of swarm theory. Distributed processing of an algorithm does not imply swarm intelligence, even though the converse is always true.

Decentralisation brings with it the resilience, capacity, performance and scalability it takes to get CATM off the ground, but it also introduces a new level of complexity, in that the optimiser algorithms must now be designed such that their throughput scales linearly with the addition of computational resources. Algorithms designed to run on a single threaded, von-Neuman processor are often inadequate or inefficient in the multiprocessor domain. This creates a new avenue for research into distributed but collaborative trajectory optimisation [2.20]. A number of research threads are being investigated as part of this work.

Luckily, one important feature of stochastic and heuristic optimisation approaches is that they are often easily parallelizable. A variety of non-Von Neumann parallel architectures are available to massively increase the computational capacity of avionics hardware today.

2.1.7 Offline vs. Online CATM

In CATM, airlines and all other airspace users are required to submit Initial Shared Business Trajectories (ISBTs) to the network well in advance of the actual flights. The network shoulders all responsibility for generating sets of conflict free trajectories that deviate as little as possible from the ISBTs.

It stands to reason that late ISBT requests will cost significantly more than those made much earlier. This incentivises early planning by airlines and lends itself for a better coordinated, more predictable, and efficient use of the Airspace. Most ISBTs would be expected months or years in advance as in the case of scheduled flights. A market mechanism can then be used to raise costs for submitting ISBTs for busy periods [2.21]. This effectively distributes traffic and minimised congestion. The CATM algorithm must then operate in two very distinct phases:

The Offline Phase: This is the strategic part of the system that deals with the long term pre-planning associated with processing ISBT requests in advance of the actual flights. It is involved in the issuance of business trajectories that have been scrutinised, and adjusted as necessary to take into account forecasted traffic, weather and runway availability using best effort statistical models for each. The offline phase is not a time critical system and can therefore operate as iterative batch jobs using as many resources as necessary on land based supercomputers. Flights approaching departure are progressively dedicated higher resources to take into account detailed weather forecasts and higher frequency traffic updates.

The Online Phase: This is the tactical part of the system and deals with the short term requirements and immediate constraints faced by each flight during execution. This part of the system must be lightweight and fast since it must be able to operate in real time and is therefore always online. The statistical models are replaced with real-time data and accurate predictive models.

Both phases adopt the same iterative approach however the distinction arises from the different levels of reliability of the constraint models they use to guide the optimiser

2.1.8 Airborne CATM

It should be clear by now that by “decentralised” we also mean taking CATM airborne (**Figure 2.1.3**). Aircraft will be essentially autonomous, and although traffic separation is delegated to the cockpit, this will not be the pilot’s role. The pilot’s role, (if still relevant), would be radically different from today. This is a big paradigm shift in air transport, but in reality there is no other practical way of achieving CATM. For reasons of efficiency and safety, it is essential that every aircraft is directly responsible of generating its own optimal conflict-free trajectory in real-time – in collaboration with all the rest. This ensures that no flight is left in the predicament of not being able to update its own flight plan in response to emergencies during a communications failure.

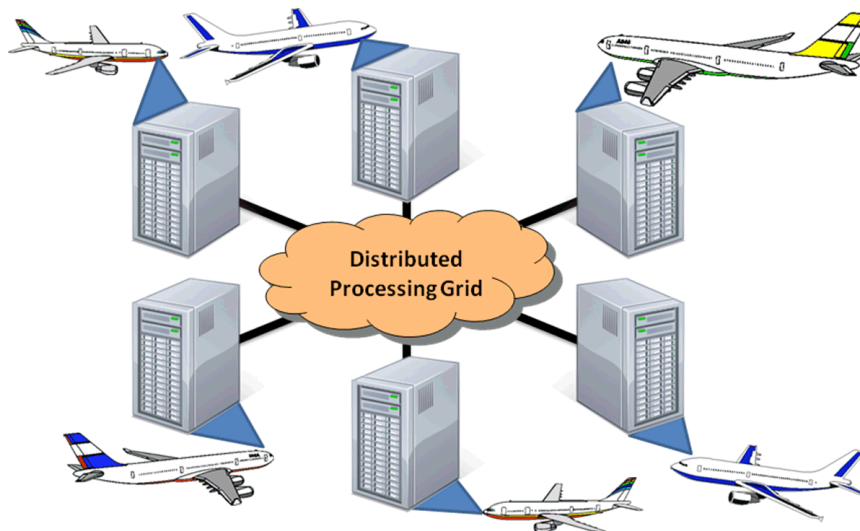


Figure 2.1.3: Distributed Processing = Grid-Avionics

This is also where the CATM concept starts departing from the early ideas of *free-flight*. Whereas in the latter case, ATM is largely dispensed-with and replaced with pilots' discretion and tactical CD&R, in CATM very tight strategic control of all aircraft behaviour needs to be maintained at all times.

2.1.9 CATM and the Role of the Pilot in an Autonomous system

In recent times, the task of flying aircraft has been largely delegated to the autopilot under the direction of the onboard flight management system (FMS). Pilots have been left with a predominantly managerial, communications, monitoring and coordination role. Pilots control the aircraft indirectly, through FMS reconfiguration.

Pilots, however, retain the key role of ensuring the safety of the aircraft and are hence tasked with the all important job of handling any off-nominal situations. Yet, with today's high reliability aircraft, 99.9% of all flights are uneventful [2.22]. Pilots are losing skills as they become increasingly complacent and reliant on the automation. The additional layers of automation introduce new system states, new failure modes, and new avenues for complicating and misreading unfamiliar situations. Thus, under these conditions, the ability of humans to react correctly under sudden intense stress, diminishes [2.23]. The 2009 Air France AF447 accident is a sobering example of how a simple temporary failure (iced pitot tubes) caused three pilots to misconstrue circumstances to the point of forcing their aircraft into a stubborn stall with tragic consequences [2.24].

Irrespective of the level of autonomy, a safety-net of last resort is required and it is generally accepted that it must involve some human decision making. Yet within the context of fully autonomous CATM, pilots would invariably be left with a highly vestigial role in the cockpit. Under these conditions it is questionable whether they would be able to retain their performance to deal with exceptional circumstances. Regular simulator training is meant to compensate for such *de-skilling*, however it is doubtful whether this alone can be sufficient. The role of the pilot will have to change.

One solution to this predicament is being envisioned as follows: A ground-based organisation of elite pilots would be specialised in handling the occasional off-nominal situations using a civilian version of RPAS (remotely piloted aircraft systems). RPAS centres would cover the national airspace of each country, thereby gathering exceptional experience with handling emergencies. This will of course rely on highly reliable bidirectional communication links, but this is discussed in detail further down.

2.1.10 High Performance Grid-Avionics and CATM

We envision that CATM will spur a new generation of grid-connected avionics hardware. For the first time in aviation history, the cockpits of passenger aircraft would be carrying numerical processors with performance which significantly exceeds those found in most desktop computers. For the cockpit, such processors would need to be formally verified, radiation hardened, multiplex redundant, and 100% tested, and they would of course be expensive, but their added cost would be quickly amortised, given the long term savings over what ATM costs the travelling public today. The world of

high performance computing has progressed to an impressive degree while ATM was looking the other way, and it is high time that it catches up to explore the possibilities.

A CATM node would reside alongside existing avionics hardware in the cockpit, sharing standard backplane busses. It would consist of a dedicated high capacity computing platform with a heterogeneous-computing architecture. The fault tolerant IMA design should have modest power consumption ($< 2\text{kW}$), with inbuilt fine grained redundancy and over-capacity with special provisions for graceful degradation or *soft failure*. The CATM system would appear as an additional layer on top of existing systems. It logically sits between the FMS and the communication/navigation subsystems. It therefore does not replace the FMS, but rather serves as its input, eliminating the need for manual FMS input. As flight plans are updated by the CATM network, these are downloaded into the FMS for immediate execution by the autopilot.

More aircraft in the air implies higher system complexity, but it also means more computational resources to dedicate to the problem. Whether the combinatorial increase in complexity would outstrip the linearly increasing computing capacity is a key research question that is being looked into. The global air traffic system consists of around 26,600 airliners, 100,000 military aircraft and over 300,000 active general-aviation aircraft. If we only consider airliners, one can safely assume that half of them are airborne at any point in time. Now, if each one holds a modern high performance avionics-grade CATM-compliant computer on board, this represents a massive amount of aggregate processing power which is deployed right where it is needed.

Given current levels of computing power density of CPUs [2.28] and GPUs [2.29], the average cockpit should be able to comfortably host between 10 and 50 Teraflops of radiation-hardened and reliable computing hardware. The aggregate across the airborne fleet can reach the Exaflop range, and that still does not take into account the additional capacity that can be made available at ground stations, bearing no constraints on space, power and weight. Moreover, CPU performance has historically grown exponentially. **Figure 2.1.4** shows a 21% annual trend that has been established in recent years. So if the first CATM-compliant aircraft take 20 years to roll out, the computational capacity of a cockpit, would conservatively top 1 Petaflop by then.

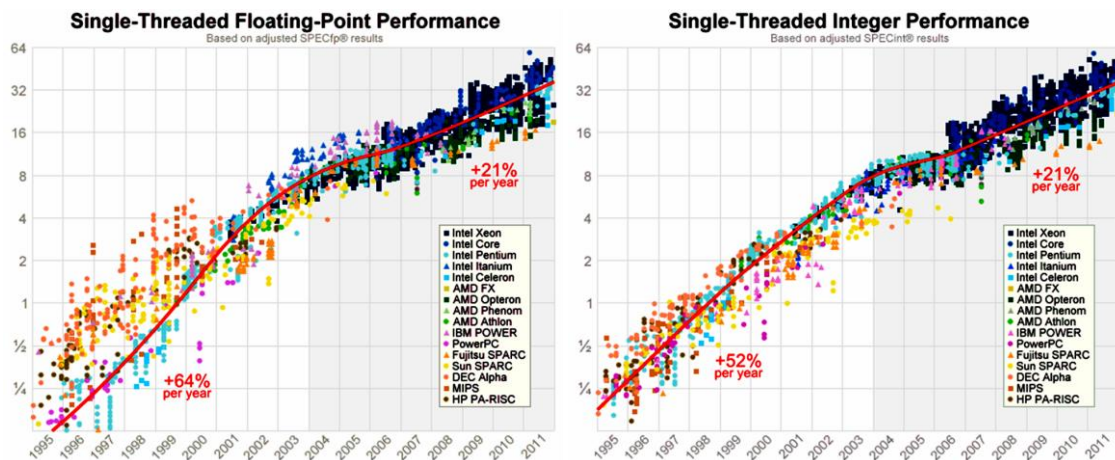


Figure 2.1.4: The Historical Exponential Growth in Computing Performance (source: PCLabs)

2.1.11 Heterogeneous Computing and Fault Tolerance

In recent years, the inability to extract higher performance from a single computer processor pushed the quest for higher processing power towards higher parallelization. This has resulted in a split in the industry, with one camp advocating in favor of small arrays of complex and powerful processors cores (Central Processing Units or CPUs) [2.28], and the other camp going for massive parallelization of relatively simple processors (Graphics Processing Units or GPUs) [2.29]. Both have unique advantages. The former is capable of executing software of impressive complexity and a practically unlimited number of variables. The latter is much better streamlined for high bandwidth computing, with 10x to 1000x advantage depending on optimizations [2.30], but can only efficiently deal with identical parallel instances of relatively simple routines.

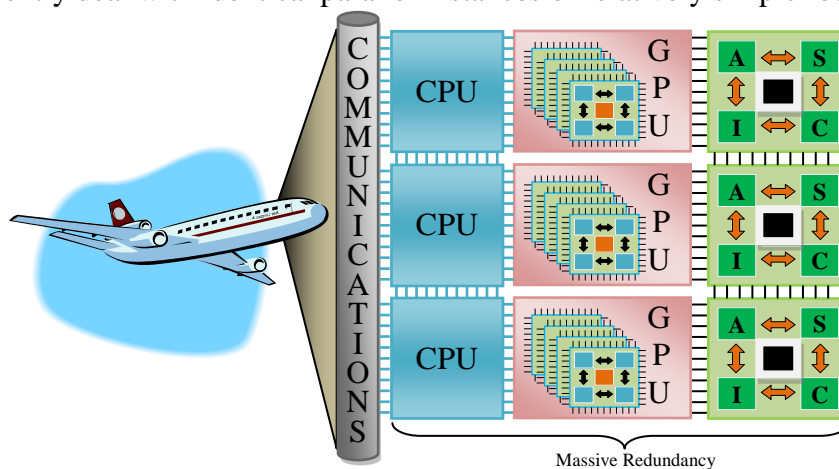


Figure 2.1.5: Triple Rank Heterogeneous Computing.

For a further 100x increase in performance, dedicated trajectory optimization can be achieved using embedded application specific integrated circuits (ASICs) [2.31]. Such hardware is designed to match the algorithm rather than the other way around. For a high value application such as avionics, this approach is very appropriate and can actually reduce overall costs. Another class of computing hardware called Field Programmable Gate Arrays (FPGAs) offers near ASIC speeds but with the possibility of on-the-fly reconfiguration.

Needless to say, the way forward seems like a combination of all three paradigms and triple-rank heterogeneous computing attempts to achieve this (**Figure 2.1.5**). In a typical optimisation algorithm involving numerous agents, there is an even mix of all kinds of tasks. Program flow often consists of highly branched routines that make use of many variables per thread, which can only be accommodated on CPU-like architectures. However, this is often interspersed with large batches of relatively small operations which need to be performed identically on large arrays of data elements. The CPU can hand over these portions to GPUs, which are much better suited for these kind of operations, particularly when they do not involve many transfers to main memory. For intensely demanding but highly uniform calculations the FPGA/ASIC model excels and can take over these parts. Therefore, there is a growing need for algorithm design to closely follow the strengths and weaknesses of the underlying hardware, while apportioning the relevant parts to the appropriate hardware [2.32].

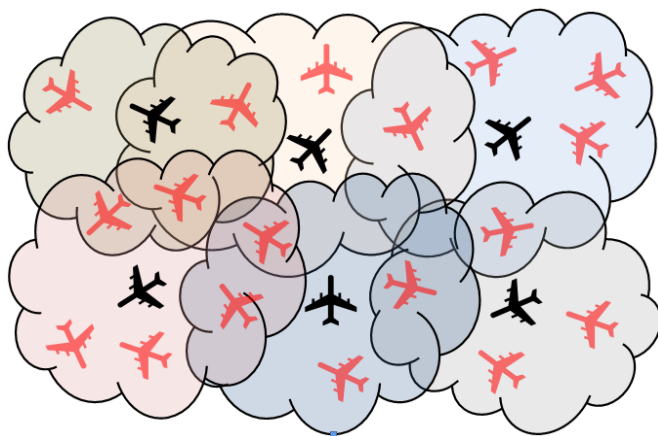
In a safety critical application such as ATC, fault tolerance is an imperative. This is traditionally achieved with triplex redundancy, built on a deep understanding of subsystem interactions. Total reliance on CATM makes the reliability of the computing platform ever more important. Luckily this is compatible with modern computing architecture. The massively parallel processing can be configured for graceful degradation, if any portion of the hardware fails. Failover techniques have been in development for decades in the server market and the technology is very well understood.

Massive redundancy is also a powerful tool to enhance radiation tolerance. Besides radiation hardening at the silicon level, data corruption can be detected and corrected on the fly by comparing the results from multiple threads. Moreover, probabilistic optimisers such as genetic algorithms and swarm intelligence are naturally immune to such disruption and take it in their stride as part of their mutation operators. Other optimisation algorithms can be designed to recover rapidly.

2.1.12 Clustering and Information Flow in CATM

An alternative way of looking at the N -Way CATM problem is to break it up into smaller overlapping clusters of M aircraft. This would reduce the computational demand in a major way by breaking up the combinatorial expansion into manageable portions. Communications is also set to benefit. It would be unreasonable to assume that an aircraft flying across eastern Canada needs to know any of the details about a flight between Rome and Moscow. At the same time, it is true that they may be indirectly related, and that they may affect each other through another flight that interacts with both. This is why the global air traffic system will have to form a network that contributes towards a single distributed processing grid on which CATM is hosted.

The only reasonable way of adhering to the *principle of (network) locality* [2.33], is to make CATM a network-centric solution, divided into clusters, and entirely airborne. Locality then implies that the influence that one side of the world's traffic exerts on the rest, need not be communicated directly. The effect is largely localised, and the further away one flies from flight XYZ123, the less it matters, but the effect still ripples through. Practical algorithms are required to take advantage of this fact.



$$\begin{aligned} \forall i \leq N \quad a_i \in C_i \\ M_i = |C_i| \quad \text{where } i, N \in \mathbb{N} \\ \forall j \leq M_i \quad a_j \in C_j, \quad j \neq i \end{aligned}$$

Figure 2.1.6: Overlapping Clusters in CATM

The relationship between flights may be defined in terms of their interactions. Two flights interact when the point of closest proximity between the pair of trajectories drops below some prescribed radius, δ . Closer proximity increases the likelihood that these trajectories would need to take into account each other in order to maintain adequate separation. Another kind of interaction arises when two flights are sequentially connected as a result of reusing the same aircraft.

Assume N total flights in the airspace, over a span of time. Over the course of a typical flight a_i , an aircraft might interact with $M_i - 1$ other flights (**Figure 2.1.6**), the set of which (including a_i) will be collectively referred to as the cluster C_i where flight a_i is the 'head node' of the cluster C_i . However, each of the M_i flights in cluster C_i , is also made the head node of its own cluster C_j , hence, there are as many clusters as there are flights.

All flights in a cluster share planning information and periodically collaborate to generate and maintain conflict free and efficient trajectories for the entire cluster. Each flight is potentially also a member of several other clusters, creating a network of highly overlapping clusters that include all N flights. These overlaps ensure that logically adjacent clusters generate compatible trajectories by virtue of those flights that straddle more than one cluster. Thus, most aircraft have interests in a number of adjacent clusters and ensure that their interests are adequately represented in each.

However, at any single point in time, only a small subset of the aircraft in cluster C_i is within a radius δ of each other. These aircraft are within range for direct communication. The rest of the members of the cluster could be thousands of nautical miles away, and have to rely on the network of aircraft to communicate with their peers via proxy.

2.1.13 Communication Infrastructure

A resilient, inter-aircraft, high-bandwidth communication network infrastructure that interconnects all aircraft to each other and the various ground stations is required. Both inter-aircraft and air-ground data bandwidths need to increase by many orders of magnitude (100MB/s~1000MB/s) with respect to what is planned for Controller-Pilot Data Link Communications (CPDLC), Automatic Dependent Surveillance-Broadcast (ADS-B) or even System Wide Information Management (SWIM).

The broadcast-oriented philosophy that underlies all popular airborne communication links is a major bottleneck because it hinders frequency-band reuse. Before CATM can become possible, the industry must depart from this model to favour directional, beam-steered, high power, point-to-point, tracking data-links that are able to pre-compensate or accommodate the significant Doppler shifts that are inherently present when there is fast relative motion between the end points. Certain microwave bands (in the Ku band) are suitable, but tracking free space optical (FSO) links are also attracting much attention due to their vastly increased bandwidth and the very low-divergence afforded by lasers, which implies high power efficiency, better security, immunity to jamming, and greatly improved electromagnetic interference characteristics. In 2002, Lawrence Livermore Laboratory demonstrated a 2.5Gbit link over a distance of 28km [2.34] and a 1.37 terabit link was reported in 2012 over four

polarisation-multiplexed, 16-QAM (quadrature amplitude modulated) beams using orbital angular momentum (OAM) multiplexing [2.36].

Multi-gigabit free-space optical laser links [2.35] between mobile platforms have also been investigated with some success for inter satellite, space-ground and high altitude inter-aircraft applications and these can complement additional microwave links. NASA has also been developing its own FSO, in the form of the Lunar Laser Communications Demonstrator (LLCD) to communicate between lunar orbit and a ground station over 380,000 kilometres including the terrestrial atmosphere [2.37]. The asymmetric link was successfully demonstrated in October 2013 with an error-free uplink rate of 20 Mbps and a 622Mbps downlink using a 500mW optical source, 16-ary pulse position modulation (PPM) and a 1/2-rate serially-concatenated turbo code [2.38].

Rising warm air and clear-air turbulence still present major challenges in the form of optical refraction, causing deep fades particularly in cases involving high relative velocity. Fog and cloud are also very problematic. However, these are not considered insurmountable [2.39]. High diversity multi-aperture transceivers, interleavers, forward error control and carefully designed protocol stacks, are a promising way forward.

In recent years, substantial work has been undertaken to ruggedize FSO links using Multiple-Input-Multiple-Output (MIMO) techniques [2.40]-[2.45]. These systems enhance diversity by combining numerous parallel channels with carefully designed error control codes (ECC) and modulation schemes to reduce the susceptibility of fading and temporary outages. Parallel Relay Assisted FSO (PRAFSO) is another powerful technique that is particularly suited to CATM (**Figure 2.1.7a**). Since fading and atmospheric attenuation is distance and direction dependant, PRAFSO makes use of multiple hops along intermediate nodes to circumvent the problem [2.46]. Relaying Protocols are being developed to maximise throughput and link availability [2.47].

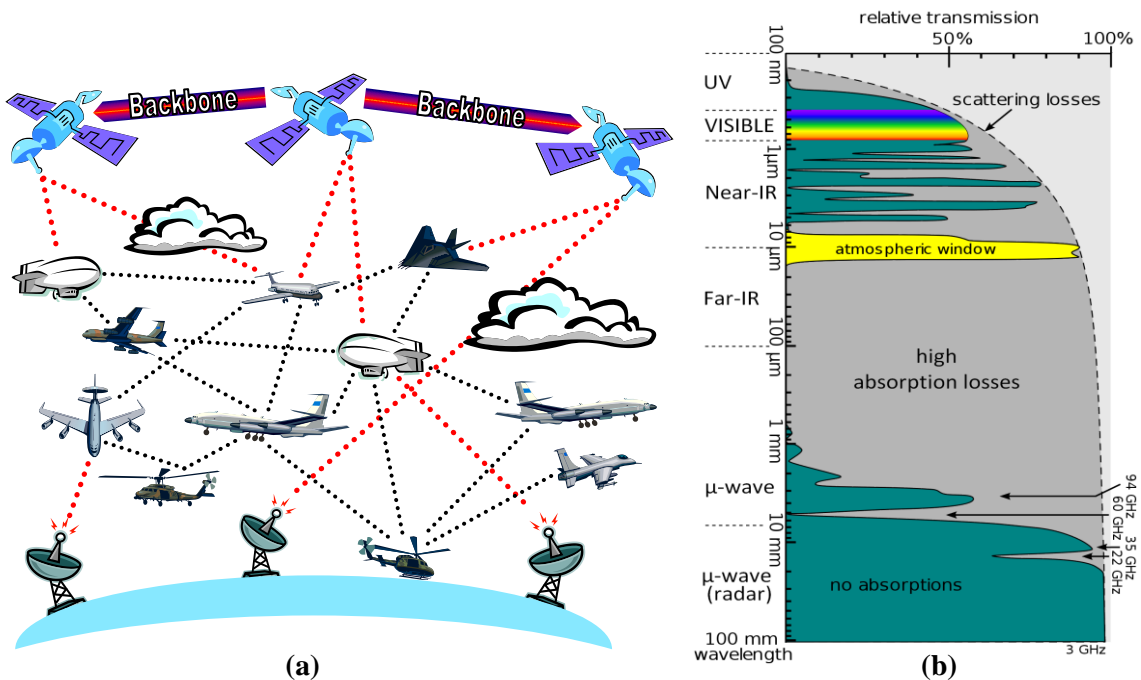


Figure 2.1.7: The Optically and Radio Networked Airspace, and absorption spectrum

In dense traffic, clear-air FSO links of a few km would be short enough such that link power should not be an issue. In sparser traffic, additional solar powered drones or blimps can be deployed to serve as communication hubs (**Figure 2.1.7a**). FSO linked satellites can feasibly be used to create a fast network backbone, while in cloud and fog, other windows in the optical/radio spectrum (**Figure 2.1.7b**) would have to temporarily take over [2.48]. Communications technology has come a long way, overcoming remarkable challenges in the past. By combining multiple communication modalities and leveraging the net-centric communications model being suggested for CATM - using satellite backbones, ground stations and other intermediate floating nodes - it should be possible to create a fast and high availability communications infrastructure.

Comprehensive information gathering and fast and reliable data links are the corner stone of good situational awareness. This cannot be over-stated, because with them, little can happen that can be truly unexpected. Most phenomena involving aircraft, evolve slowly and the system is given ample time to adapt gradually. In addition, with the ability of staying vigilant for all external influences on the system, the limited amount of slack in the system can be carefully rationed in a way that introduces extra room for recovery, should off-nominal situations arise.

Besides the operational advantages there is an argument for safety. Very few accidents are both sudden and catastrophic. With the debatable exception of malicious attacks, most incidents involve a detectable, quantifiable, gradually-evolving run-up scenario. By design, it takes an improbable confluence of events to convert incidents into accidents and timely distribution of information is the pre-condition which makes them preventable. The recent German Wings co-pilot sabotage event (flight 4U9525) that has shocked the industry could have been prevented had there been the communications infrastructure to assess, and then take control of the aircraft remotely.

The evolution of the *Internet* amply demonstrates the kind of network topologies that scale up efficiently [2.48]. Multiply-interconnected, decentralised, ad-hoc peer-to-peer (P2P) networks can offer some of the highest levels of network scalability, resilience, capacity, and availability known. Classic client-server architectures are not suited for the most part. A net-centric approach is better indicated, given the way data needs to flow in the proposed CATM system. CATM can be accommodated onto an ad-hoc network infrastructure conformant to standard layered OSI principles. Long range communications can then rely on multiple hops along the net-centric architecture rather than direct long distance links. Standardisation of the CATM language of interchange is part of the game. A universally accepted descriptor for the Business Trajectory is required before a CATM data packet structure can be defined.

It is not expected that CATM will require voluminous data transfers between network peers, but exchanges will occur at high frequency. They are largely limited to intermediate optimisation results, essentially consisting of compressed trajectory bundles, for consideration by the rest of the cluster. A multicast forwarding model is therefore suitable. However, network latency is likely to pose problems and therefore CATM data packet transfers need to be completed rapidly in order to allow quick system re-convergence to the prevailing problem domain.

Finally it must be said that although CATM depends entirely on efficient communications, there is much room for optimisation of the CATM communications protocol. Using clever data reduction and compression techniques, CATM would possibly be able to function with rather modest bandwidth requirements which would reduce the strain on radio communications. This will become a hot area of research.

2.1.14 Sensor Networks

A key element that ensures that air transport functions safely with as few surprises as possible is information gathering and distribution. Whether it involves knowing the locations of other aircraft and obstacles, evolving weather conditions or the mechanical/electronic health of each airframe, the timely and exhaustive information gathering is what separates recoverable situations from inexplicable disasters. The recent Malaysian Airlines MH370 case is one such example where lack of information allowed a relatively minor incident to grow into one of the most expensive aviation disasters in history. Current information technology greatly exceeds the data capacity requirements associated with ATM, the basic science is well understood and it is now a matter of bringing it all together.

2.1.15 CATM & Flight Data Recorders

The CATM communications infrastructure also presents an excellent opportunity to revamp the half-century-old flight data recorder (FDR) concept. The continuous, high update rate, status transmissions effected by each and every aircraft can easily be recorded in encrypted format by all peers in a cluster as well as ground stations. Commercially sensitive information would only be decryptable by the respective airlines. This multiple redundancy reduces the dependency on retrieving these FDRs from remote locations, hence insuring against the loss of such data in the event of an accident. Additionally, from a communications perspective, such distributed recording is more technically feasible than direct data streaming to distant ground stations, as was recently proposed [2.49], [2.50] following the southern Atlantic Air France Flight 447 accident where the flight recorder was lost in the deep ocean, and was only recovered two years at great cost [2.24]. The Malaysian MH370 lost in the Indian Ocean served to further embarrass the industry in this regard, but will hopefully spur some change.

2.1.16 CATM Trajectory Descriptors

In CATM, trajectories are defined as 4D curves that aircraft are contractually obliged to follow closely. They specify the exact location (and hence velocity) of each aircraft at any point in time, from take-off to landing. Such trajectories are pre-calculated before take-off, but are continually updated to take into account evolving conditions, with mutual agreement of all members in a cluster. Since trajectory synthesis and optimisation ultimately needs to be computed in real-time, some form of trajectory parameterisation is required to reduce the search space to reasonable levels. CATM also requires a versatile, compact and deterministic method for describing 4D trajectories. Thus, ensuring a single standard method of parameterisation allows for

efficient storage, processing and exchange of large numbers of trajectory contracts and intermediate partial solutions across the distributed CATM processing grid.

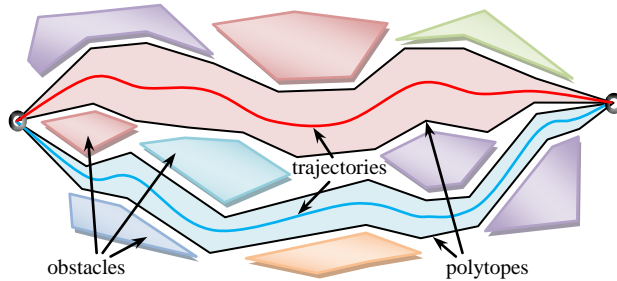


Figure 2.1.8: NURBS and polytopes negotiating obstacles

shapes (**Figure 2.1.8**). Several other parameterisation techniques are possible and a few are being listed hereunder:

- *Splines*
- *Bezier Curves*
- *B-Splines*
- *NURBS*
- *Spectral Methods*

Some curves, such as NURBs also exhibit the very useful property that guarantees that the interpolated trajectory is fully enclosed within the convex hull produced by the knots defining the curve. If the hulls can be shown not to intersect, then the resulting trajectories are mathematically guaranteed to be conflict free. This can greatly simplify formal verification at a later stage.

2.1.17 Security and Collaborative CATM

The collaborative element of CATM goes hand in hand with its decentralised nature and ensures that the trajectory solutions found for every aircraft are equitably balanced among the interests of every airline and aircraft. The CATM numerical computational tasks are also collaboratively shared among all aircraft. Information is requested and mutually shared as necessary by every aircraft running the CATM algorithm. Since certification guarantees that all aircraft operating in a particular airspace must be conformant to the same set of specifications, the risk of having one aircraft departing from a prescribed set of egalitarian CATM rules to take advantage of its peers, should be negligible.

One technique that can be adopted to further guarantee acceptable behaviour, involves *online-certification*. Once a common set of formally verified CATM rules of engagement is established and deployed, adherence is mandated and can be digitally enforced. The aircraft's active firmware can be digitally signed using internationally recognised certificates and electronic watchdogs can be put in place to ensure that the correct version is being executed before being granted access to the CATM network. Standardisation is key here and it has to be centrally managed, and adherence to the

A promising approach was explored by Flores and Milam at Northrop Grumman for UAV applications [2.51]. The concept is to employ well behaved interpolators to describe trajectories using relatively few 4D knots or nodes along the curves, while retaining a very rich repertoire of possible trajectory

standard can be verified by the network. The degree of variation allowed between different aircraft types and airlines would also need to be defined and standardised.

Public-key cryptography should be used in a CATM system. Once a set of conflict-free trajectories has been generated by a cluster, it is broadcast as a block to the entire system for verification and approval by consensus. In order to protect against data corruption or malicious manipulation by intermediate nodes in the network, each transmitted block is signed by the cluster's head node using its private key. This makes it computationally infeasible to modify the content of a block or forge the signature without discovery and rejection by other nodes.

New blocks are issued periodically with trajectory updates. One interesting possibility for achieving consensus over the CATM network is by using Nakamoto block chains [2.52] (not unlike the ones used for crypto-currency like bitcoin), which link each block to previous versions using hash chains and converge on a single version using a proof of work system. Each node in the network effectively casts a vote to select a particular evolutionary path of a block of trajectories. The longest surviving chain is the one that the network effectively adopts to build upon.

In the case of equipment failure where one aircraft may become unresponsive to the requests sent by the others, CATM defaults to a surveillance mode, where all aircraft in the vicinity of the damaged peer use their on-board radar to locate, track and take into account all its motion. Its intent is inferred in real-time and all the information gathered is shared on the CATM network to generate a set of trajectories avoiding the errant peer. The same applies for aircraft that for any reason is regarded non-conformant or non-collaborative by the rest of the community.

2.1.18 The Way Forward

In this chapter the main conceptual elements behind Computational Air Traffic Management have been outlined. Several of the practical aspects of the CATM paradigm were considered. Although CATM creates a demand for specific technologies to be developed or adapted, there appears to be no major show stopper in terms of infrastructure. After having previously described the back drop which underpins CATM [2.3] as a possible successor for SESAR and NextGen, we have now shed more light on how it can be brought to fruition using some well understood ideas.

However, the current description leaves many open research questions related to the optimisation algorithms that will be addressed in the next chapters. Developments related to distributed computational intelligence and decentralised robust control from homologous fields must be taken on board. Subsequent chapters will provide further detail about the system's overall structure as well as exploring a few promising algorithms, and organisational paradigms in some detail. We then shift our focus on the description, simulation and testing of some candidate decentralized CATM algorithms.

2.2 REFERENCES

- [2.1] Gilmore, J. F. 1991, "Autonomous vehicle planning analysis methodology," in Proc. Association for Unmanned Vehicle Systems Conference, Washington, DC, pp. 503–509.
- [2.2] Durand, N., Alliot, J.-M., et al, 1995, "Optimal Resolution of En Route Conflicts," Air Traffic Control Quarterly, Vol. 3, No. 3, pp. 139-161.
- [2.3] Azzopardi, M.A. & Whidborne, J.F., 2011, Oct 16-20, "Computational Air Traffic Management", Proceedings of the 30th AIAA/IEEE Digital Avionics Systems Conference (DASC2011), IEEE/AIAA, Seattle, WA, pp. 1.B.5-1.
- [2.4] Garey, M. R, Johnson, D. S., 1979. "Computers and Intractability: A Guide to the Theory of NP-Completeness," A Series of Books in the Mathematical Sciences. San Francisco, Calif., W. H. Freeman and Co.
- [2.5] Bertsimas, D., and Stock, S., 1988, "The Air Traffic Flow Management Problem with Enroute Capacities," Operations Research, Vol. 46, No. 3, May–June, pp. 406–422.
- [2.6] Garey, M.R., Johnson, D.S., Sethi, R, 1976, "The Complexity of Flowshop and Jobshop Scheduling," Mathematics of Operations Research, Vol 1, no. 2: pp. 117–129
- [2.7] Majumdar, R., Tabuada, P., 2009, "Hybrid Systems: Computation and Control", Proceedings of the 12th International Conference, Lecture Notes in Computer Science (LNCS), HSCC 2009, San Francisco, CA, April 13-15, 2009, Volume 5469 2009
- [2.8] Sanjeev, A., Boaz, B., 2009, "Computational Complexity: A Modern Approach," Cambridge University Press, New York, 1st ed.
- [2.9] Moscato, P., 1989, "On evolution, search, optimization, genetic algorithms and martial arts: Towards Memetic Algorithms." Caltech Concurrent Computation Program, C3P Report 826
- [2.10] Renders, J. M., & Flasse, S. P., 1996, "Hybrid methods using genetic algorithms for global optimization", Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions, 26(2), 243-258.
- [2.11] Castellini, F., et al., 2011, Apr. 4-7, "Global and Local Multidisciplinary Design Optimization of Expendable Launch Vehicles", 52nd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, Denver, Colorado, AIAA 2011-1901
- [2.12] Alba, E., Nakib, A., and Siarry, P., (Eds.), 2013, "Metaheuristics for Dynamic Optimization", Studies in Computational Intelligence, Springer-Verlag, Berlin, Germany
- [2.13] Eurocontrol Experimental Centre (EEC), 2013, May, "User Manual For The Base Of Aircraft Data (BADA)," Revision 3.11, Technical/Scientific Report No. 13/04/16-01, Project BADA, European Organisation for the Safety of Air Navigation, Eurocontrol
- [2.14] ICAO, 1964, "Manual of the ICAO Standard Atmosphere," ICAO Document No. 7488, 2nd Edition.
- [2.15] Mezura-Montes, E., Coello Coello, C.A., 2011, "Constraint-handling in nature-inspired numerical optimization: Past, present and future," in Swarm and Evolutionary Computation, Vol 1, Issue 4, December 2011, pp 173-194
- [2.16] Ray, T., Singh, H.K., Isaacs, A. Smith, W., 2009, "Infeasibility driven evolutionary algorithm for constrained optimization," in: E. Mezura-Montes (Ed.), Constraint-Handling in Evolutionary Optimization, in: Studies in Computational Intelligence Series, vol. 198, Springer-Verlag, pp. 145–165.
- [2.17] Roussos, G. P., D.V. Dimarogonas and K.J. Kyriakopoulos, 2009, Aug 23-26 "Distributed 3D Navigation and Collision Avoidance for Nonholonomic Aircraft-like Vehicles," Proceedings of the 2009 European Control Conference (ECC'09), Budapest, Hungary
- [2.18] Diekmann, R., R. Luling and J. Simon, 1992, "A General Purpose Distributed Implementation of Simulated Annealing," 4th IEEE Symposium on Parallel and Distributed Processing, SPDP

- [2.19] Tomlin, C.J., G. Pappas, J. Kosecka, J. Lygeros and S. Sastry, 1998, "Advanced air traffic automation: a case study in distributed decentralized control," *Ctrl. Problems in Robotics and Automation*, pp. 261-5
- [2.20] Giulietti, F., L. Pollini and M. Innocenti, 2000, "Autonomous formation flight," *Control Systems Magazine*, IEEE, vol. 20, no. 6, pp. 34-44
- [2.21] Waslander, S.L., 2007, "Multi-Agent System Design for Aerospace Applications," PhD Dissertation, Department of Aeronautics and Astronautics, Stanford University, CA, USA
- [2.22] ATSB, 2013, Oct. 30, "Aviation Occurrence Statistics 2003 to 2012", Aviation Research Report, Australian Transport Safety Bureau (ATSB), Commonwealth of Australia
- [2.23] Sheridan, T. B., & Parasuraman, R., 2005, "Human-automation interaction" in *Reviews of human factors and ergonomics*, vol 1, no.1, pp.89-129.
- [2.24] BEA, 2012, Jul. 27, "Final Report: on the accident on 1st June 2009, to the Airbus A330-203, registered F-GZCP, operated by Air France, flight AF 447 Rio de Janeiro - Paris", France
- [2.25] Stipanovic, D., G. Inalhan, R. Teo and C.J. Tomlin, 2002, Dec., "Decentralized Overlapping Control of a Formation of Unmanned Aerial Vehicles," 41st IEEE Conference on Decision and Control, vol. 3, pp. 2829-2835
- [2.26] Bui, Lam T., H.A. Abbass and D. Essam, 2006, January, "Local Models: An Approach to Distributed Multiobjective Optimization," ALAR Technical Report Series
- [2.27] Defoort, M., 2010, Sep. 8-10, "Distributed receding horizon planning for multi-robot systems," Control Applications (CCA), IEEE International Conference on, Yokohama, Japan, pp. 1263-1268
- [2.28] Intel, 2013, "Intel® Xeon Phi™ Product Family Performance Brief", Intel Corporation, Santa Clara, CA, Rev 1.4, 30th Dec.
- [2.29] Nvidia, 2013, "Tesla Kepler Family Product Overview", NVIDIA Corporation, Santa Clara, CA, Oct.
- [2.30] Lee, V.W., 2010, "Debunking the 100X GPU vs. CPU myth: an evaluation of throughput computing on CPU and GPU", Proceedings of the 37th annual international symposium on Computer architecture (ISCA '10), pp 451-460, ACM New York, NY
- [2.31] Ross, I. M., & Karpenko, M. 2012. "A review of pseudospectral optimal control: From theory to flight." *Annual Reviews in Control*, vol.36, no.2, pp. 182-197.
- [2.32] Yang, C., et al., 2010, "Adaptive optimization for petascale heterogeneous CPU/GPU computing" In *Cluster Computing (CLUSTER)*, 2010 IEEE International Conference on, pp. 19-28.
- [2.33] Denning, P. J., 2005, "The Locality Principle," *Communications of the ACM*, vol. 48, no.7, pp. 19-24
- [2.34] Johnston, D., 2002, "Livermore gets longest laser communication link", Lawrence Livermore National Laboratory, Report 925/423-4902.
- [2.35] Chan, V.W.S., "Free-Space Optical Communications", 2006, *Journal of Lightwave Technology*, Dec. 2006, Vol 24, no 12, pp. 4750-4762
- [2.36] Wang, J., Yang, J. Y., Fazal, I. M., Ahmed, N., Yan, Y., Huang, H., ... & Willner, A. E. 2012, "Terabit free-space data transmission employing orbital angular momentum multiplexing." *Nature Photonics*, vol. 6, no.7, pp. 488-496.
- [2.37] Boroson, D. M., Scozzafava, J. J., Murphy, D. V., Robinson, B. S., & Shaw, H., 2009, Jul., "The lunar laser communications demonstration (LLCD)." In *Space Mission Challenges for Information Technology*, SMC-IT 2009. Third IEEE International Conference on, pp. 23-28
- [2.38] Boroson, D. M., Robinson, B. S., Murphy, D. V., Burianek, D. A., Khatri, F., Kovalik, J. M., ... & Cornwell, D. M., 2014, Mar., "Overview and results of the Lunar laser Communication Demonstration." In *SPIE LASE*, International Society for Optics and Photonics, pp. 89710S-89710S
- [2.39] Epstein, B., Mehta, V., 2004, "Free space optical communications routing performance in highly dynamic airspace environments", Aerospace Conference, 2004. Proceedings. IEEE, vol. 2, pp. 1398-1406

-
- [2.40] Wilson, S. G., Brandt-Pearce, M., Cao, Q., & Leveque III, J. H., 2005, "Free-Space Optical MIMO Transmission With Q-ary PPM," *Communications, IEEE Transactions on*, vol.53, no.8, pp.1402-12
- [2.41] Chakraborty, K. 2005, Sep., "Capacity of the MIMO optical fading channel." In *Information Theory, 2005. ISIT 2005. Proceedings. International Symposium on*, IEEE, pp. 530-534
- [2.42] Letzepis, N., & Guillen i Fabregas, A. 2009, "Outage probability of the Gaussian MIMO free-space optical channel with PPM." *Communications, IEEE Transactions on*, vol. 57, no. 12, pp.3682-3690
- [2.43] Bayaki, E., Schober, R., & Mallik, R. K. 2009, "Performance analysis of MIMO free-space optical systems in gamma-gamma fading." *Communications, IEEE Transactions on*, vol.57, no.11, pp.3415-3424
- [2.44] Zeng, L., O'Brien, D., Minh, H., Faulkner, G., Lee, K., Jung, D., ... & Won, E. T. 2009, "High data rate multiple input multiple output (MIMO) optical wireless communications using white LED lighting." *Selected Areas in Communications, IEEE Journal on*, vol. 27, no. 9, pp. 1654-1662.
- [2.45] Hajjarian, Z., Fadlullah, J., & Kavehrad, M. 2009, "MIMO free space optical communications in turbid and turbulent atmosphere." *Journal of Communications*, vol. 4, no. 8, pp.524-532.
- [2.46] Safari, M., & Uysal, M. 2008, Relay-assisted free-space optical communication. *Wireless Communications, IEEE Transactions on*, vol.7, no.12, pp. 5441-5449
- [2.47] Chatzidiamantis, N. D., Michalopoulos, D. S., Kriezis, E. E., Karagiannidis, G. K., & Schober, R. 2013, "Relay selection protocols for relay-assisted free-space optical systems." *Journal of Optical Communications and Networking*, vol. 5, no.1, pp. 92-103.
- [2.48] Harris,A., Sluss,J.J., Refai,H.H., LoPresti,P.G., 2006, Aug., "Free-space optical wavelength diversity scheme for fog mitigation in a ground-to-unmanned-aerial-vehicle communications link", *Journal of Optical Engineering, Atmospheric and Ocean Optics* vol.45, no.8
- [2.49] Benderli, Gökçe and P. Smith, 2006, "Could Air Traffic Management learn from other network industries?," *European Transport Conference 2006, Seminars : European Transport Policy and Research*
- [2.50] Kavi, Krishna M., 2009, Aug, "Beyond the Black Box: Instead of storing flight data on board, aircraft could easily send the information in real time to the ground", *Magazine Article, IEEE Spectrum*.
- [2.51] Chester, M. 2010, Jun., "Streaming Flight Data...Creating An Aerial Central Nervous System To Save Lives", *Article, SharedEmergency Article, Wordpress.com*
- [2.52] Flores, M. E. and M.B. Milam, 2006, Jun. 14-16, "Trajectory generation for differentially flat systems via NURBS basis functions with obstacle avoidance," *American Control Conference, Minneapolis, MN*, pp. 5769-5775
- [2.53] Nakamoto, S., 2008. "Bitcoin: A peer-to-peer electronic cash system," <http://s.kwma.kr/pdf/Bitcoin/bitcoin.pdf>, Electronically published paper, Accessed, 1, 2012.
-

Chapter 3

Modelling CATM

This chapter begins to detail a coherent conceptual framework that makes up an idealised Autonomous ATM system. It starts off with a very general agent-based formal mathematical description of the ATM system, and goes on to demonstrate how high fidelity fast-time simulation can help online optimisation to solve the Autonomous ATM problem. An advanced communications infrastructure is assumed to interconnect all aircraft in a peer to peer network. Each aircraft is assumed to be uniquely addressable and able to reliably transfer substantial quantities of information with very low latency to any other aircraft.

The CATM concept depends on the ability to model the behaviour of all influencing systems in order to predict or describe their behaviour some time into the future. The accuracy of these models varies depending on the nature of the underlying process. Aircraft models can be made quite accurate while weather and airport delays are significantly more challenging due to their stochastic chaotic nature. The final sections will provide some insight into the kind of models that can be used to simulate the CATM problem for experimental evaluation, as well as running CATM itself in a real world scenario.

3.1 CONTEMPORARY ATC

Before delving further into how the proposed ATS should work, it is worth having a systems-theoretic look at how the current ATC system is put together, and how the various variables in the system are related to one another, before we proceed to redefine several ICAO ATM and ATC functions. Like most vehicles, we can model aircraft using n^{th} order non-linear ordinary differential equations (ODEs), where some terms, such as pilot controls are interpreted as an m -sized vector of inputs $\mathbf{u}(t)$ and some other terms such as the motion of the aircraft are interpreted as an r -sized vector of outputs $\mathbf{y}(t)$. An additional input, $\mathbf{w}(t)$ represents the additive external wind disturbances. A state space model is then defined by two non-linear, vector-valued functions $\mathbf{f}(\bullet, \bullet, \bullet)$ and $\mathbf{g}(\bullet, \bullet, \bullet)$ (Eq 3.1.1), where $\mathbf{x}(t)$ and $\dot{\mathbf{x}}(t)$ respectively represent the current state and state derivative of the system.

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) + \mathbf{w}(t) \\ \mathbf{y}(t) &= \mathbf{g}(\mathbf{x}(t), \mathbf{u}(t), t)\end{aligned}\tag{3.1.1}$$

The pilot together with his aircraft form a closed-loop, multiple-input multiple-output (MIMO) feedback control system. The human pilot acts as flight controller and regularly receives (often verbal) instructions $\mathbf{r}(t \rightarrow t_f)$ from ATC in the form of small localized modifications to the reference flight plan, which affect the remaining portion of the journey, $(t \rightarrow t_f)$. The pilot then adjusts the controls to the aircraft in order to ensure that its motion is conducive to executing the given/agreed flight plan.

The aircraft's motion is affected by other factors, besides the pilot controls in the form of external disturbances, such as wind, and internal errors, due to unmodelled dynamics, tracking error and system noise. The aircraft's motion results in a cumulative (integral) effect on the aircraft's position. Navigation equipment tracks the exact real-time location of the aircraft in 3D space and provides feedback to both the pilot in the cockpit and ATC via ADS-B.

ATC maintains a much broader view of the situation by tracking the motion $1 \rightarrow M$ $[y(t)]$ and location $1 \rightarrow M$ $[p(t)]$ of all M other airborne aircraft. Radar is the primary feedback tool and provides a bird's eye picture of all aircraft within radar range using a variety of techniques. Active radar systems request a response from interrogated aircraft, while passive radar systems rely on the electromagnetic reflection. This is complemented by regular ADS-B position and status broadcasts by most aircraft. Besides issuing verbal instructions, ATC also collects information by having conversations with the pilots over VHF radio. In recent years, some of this communication is increasingly being taken over by data links such as CPDLC.

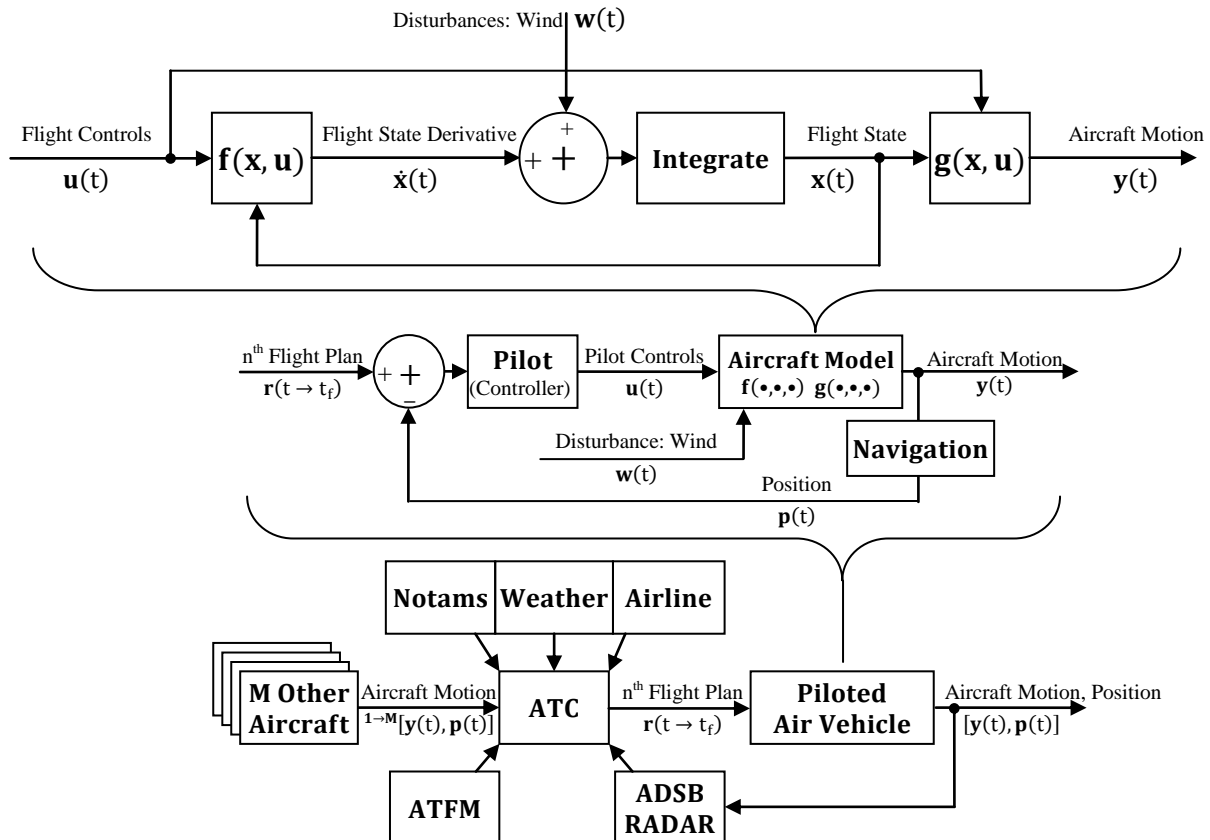


Figure 3.1.1: Model of the Current Air Traffic System

ATC is primarily responsible for aircraft separation from other aircraft and airspace restrictions (eg NOTAMS). It draws together information from many sources to instructions consistent with its role. So strictly speaking **Figure 3.1.1** does not reflect the division of workload as it is handled in practice. This is because some top-level ATC-like functions are in effect handled by the pilot himself. So in essence,

Figure 3.1.1 is intended to depict the conceptual relationship between logical functions rather than the actual workload distribution. The pilot for instance is free to modify flight plans in the interest of safety. Weather cell avoidance is also largely a pilot responsibility due to the much better situational awareness available in the cockpit. The aircraft may also fly through large swaths of airspace (eg over oceans and deserts) with no ATC services within communications range and must be able to act independently during these periods. Pilots are also able to track nearby traffic using both ADS-B-in and Radar. This provides an additional layer of surveillance and in fact pilots are ultimately responsible for taking emergency manoeuvres if they reach the assessment that the traffic or weather situation is becoming dangerous.

Finally, ATFM collates all flight plans in advance and ensures that at no point is airspace or runway capacity exceeded. ATFM liaises with the airlines, and the pilots by vetting nominal flight plans before take-off. It also delays flights from taking-off if congestion is detected downstream. So ATFM is a system-wide monitoring and management function.

3.2 THE PROPOSED AIR TRAFFIC SYSTEM MODEL

In the proposed Air Traffic System (ATS), the human-centred ATC is essentially eliminated. Instead, the airspace is populated by a number of distributed interacting entities. The goal is to ensure that this interaction is stable and reliably results in the emergent behaviour that one would expect from a well-functioning, optimal air transport system.

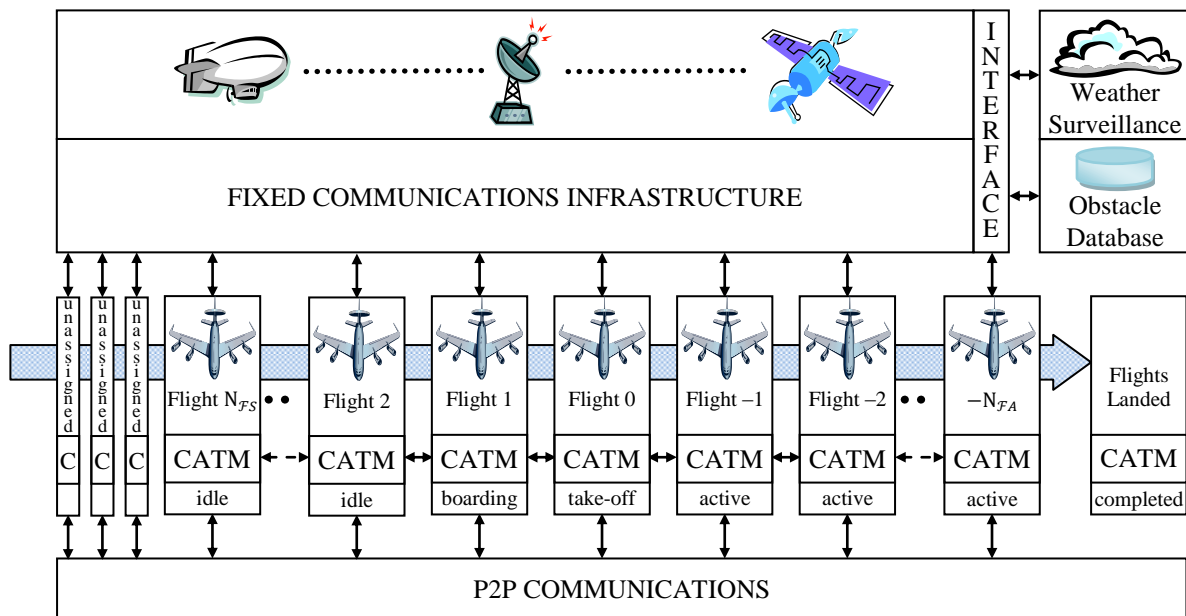


Figure 3.2.1: The CATM System Conveyor Belt Executing Flights Over Their Life Cycle

3.2.1 The Agent Model of CATM

Agents can be viewed as a generalisation of the well known *state machine*. They collect information about their surroundings through a variety of sensors and communication interfaces. They possess numerous states: some discrete some continuous. This information may then be stored, processed, and possibly combined with other stored information in order to define a future set of actions. This plan of action is then executed by means of a variety of actuators that the agent may be equipped with. This definition is totally independent of the mode of implementation and may involve either physical or virtual sensors/actuators. Human beings, animals, aircraft, flights or logic state machines clearly fit as subsets of this very general definition of automata. CATM represents each flight as an independent autonomous *agent*, or *automaton*. These agents form the elemental building blocks of the proposed CATM system. They are logical ephemeral entities to which hardware resources are assigned as needed.

3.2.2 The Flight Agent Lifecycle

Flight agents undergo a life cycle from inception, to planning, to flying and finally retirement. They are usually spawned as dormant entities that are devoid of any resources other than their basic mission definitions and data structures. Over the course of their lifecycle flight agents graduate along a sequence of 4 sets \mathcal{F}_L , \mathcal{F}_S , \mathcal{F}_A and \mathcal{F}_C . Scheduled flights are usually first placed in \mathcal{F}_L and are assigned ground-based computational resources for offline long-term flight planning. During this time, the flight agent will generate tentatively optimal trajectories based on all the information it can get hold of.

As the time for a flight's departure nears, each flight agent is uniquely assigned to a physical aircraft which endows the flight with dedicated computational resources as well as the means to execute the flight. The agent is now in set \mathcal{F}_S and remains there until it reaches time of departure. During this period the flight agent actively collaborates with other queued flights as well as other airborne flights in order to update its optimal trajectory.

After take-off, the flight agent enters set \mathcal{F}_A , and remains in it until landing at its final destination. During this time the agent actively communicates with all other flights, gathering information about the weather and traffic and sharing it with all the agents that lie within communications range. Information is relayed between agents such that all flights that lie outside the range are also furnished with updates. Each flight agent holds a database (the *Almanac*) of all other flights, which it updates regularly as new information is received. A small group of flights is shortlisted based on their expected proximity at any point during the journey. These flight agents constitute a 'cluster' which is associated with each flight. The members of a cluster collaborate to generate optimal conflict free trajectories.

When the flight reaches its destination and lands, it joins the set of completed flights \mathcal{F}_C . It thereby leaves the network and all flight information is archived in a central CATM repository for subsequent analysis and research purposes. The aircraft is released and is prepared for its next assignment. **Figure 3.2.1** shows all flights on an endless conveyor belt as they move between the four time variant sets \mathcal{F}_L , \mathcal{F}_S , \mathcal{F}_A , and \mathcal{F}_C .

3.2.3 ATS Top Structure

The proposed CATM system requires a combination of airborne and land based infrastructure. The airborne infrastructure consists of aircraft and their onboard computing platforms. The ground based infrastructure includes, aerodromes and all their ancillary hardware, idle aircraft that is being queued for imminent flights and a number of strategically located CATM Super-Computing Centres (CSCCs) that assist the CATM grid with the computational load of longer term upcoming flights. The CSCC is also responsible for collecting all the weather data from the airborne grid and to then generate detailed long terms weather forecasts. The scheduling arm of the CATM system assigns new flights to recently landed aircraft as they become available. The reuse of aircraft makes flights interdependent as a result of the availability of previously landed aircraft. This gives rise to a combinatorial optimisation problem that deserves separate attention.

By describing any arbitrary system using a rigorous language, in terms of agents, it may be possible to utilise some of the various formal verification tools that have been developed, as an extension of mathematics. Although outside the scope of this dissertation, an agent model would allow the system to be analysed, verified and hence certified in terms of performance, robustness, correctness or utility of any subsystem with respect to a given specification. This is especially relevant in the context of critical systems such as ATM, where increased automation further accentuates the need for formal verification.

We shall begin by laying down some notational definitions to give the system some structure before proceeding to describe the modelling in the next sections and algorithms in the next chapter. A number of simplifications are being made for the sake of clarity. However, this does not prejudice the addition of further detail to the model.

Definition 1: [ATS] Let S represent the global air traffic system as a quad consisting of a set of flights (F), set of airspace equipment (E), the set of airspace obstacles (O) and an associated relationship between the members of the said sets (R), that is:

$$S := (F, E, O, R) \quad (3.2.1)$$

where: $F := \{\mathcal{F}_L, \mathcal{F}_S, \mathcal{F}_A, \mathcal{F}_C\}$ (3.2.2)

$$E := \{\mathcal{A}, \mathcal{D}, \mathcal{P}\} \quad (3.2.3)$$

$$O := \{\mathcal{W}, \mathcal{R}\} \quad (3.2.4)$$

$$R := \{R_L, R_S, R_A\} \quad (3.2.5)$$

and:

\mathcal{F}_L represents the set of unassigned long term scheduled flights

\mathcal{F}_S represents the set of assigned short term scheduled flights

\mathcal{F}_A represents the set of active flights

\mathcal{F}_C represents the set of completed flights

\mathcal{A} represents the set of operational passenger aircraft

\mathcal{D} represents the set of decision makers in the system

\mathcal{P} represents the set of all recognised aerodromes

\mathcal{W} represents the set of non-traversable weather objects

\mathcal{R} represents the set of non-traversable airspace restrictions

R_L represents the ground based offline CATM Problem (Strategic Aspect)

R_S represents the ground part of the online CATM Problem (Tactical Aspect)

R_A represents the airborne part of online CATM Problem (Operational Aspect)

Definition 2: [FLIGHTS] Let $\mathcal{F}_L, \mathcal{F}_S, \mathcal{F}_A, \mathcal{F}_C$ represent the sets of long-term scheduled, short-term scheduled, active and completed flights respectively. Over time a unique flight f_i graduates from one set to the next in this order with the help of transitioning processes: R_L, R_S and R_A

$$\begin{aligned}\mathcal{F}_C &:= \{f_i : 1, \dots, N_{CC}\} \\ \mathcal{F}_A &:= \{f_i : (N_{CC} + 1), \dots, N_{AA}\} \\ \mathcal{F}_S &:= \{f_i : (N_{AA} + 1), \dots, N_{SS}\} \\ \mathcal{F}_L &:= \{f_i : (N_{SS} + 1), \dots, N_{LL}\}\end{aligned}\quad (3.2.6)$$

such that where $\mathcal{F}_L \rightarrow \mathcal{F}_S \rightarrow \mathcal{F}_A \rightarrow \mathcal{F}_C$ (3.2.7)

$$R_L: \mathcal{F}_L \mapsto \mathcal{F}_S \quad (3.2.8)$$

$$R_S: \mathcal{F}_S \mapsto \mathcal{F}_A \quad (3.2.9)$$

$$R_A: \mathcal{F}_A \mapsto \mathcal{F}_C \quad (3.2.10)$$

and let $N_{FL} = (N_{LL} - N_{SS}) \in \mathbb{N}^+$ be the number of unassigned long term scheduled flights

$N_{FS} = (N_{SS} - N_{AA}) \in \mathbb{N}^+$ be the number of short term scheduled flights

$N_{FA} = (N_{AA} - N_{CC}) \in \mathbb{N}^+$ be the number of active flights

$N_{FC} = N_{CC} \in \mathbb{N}^+$ be the number of completed flights

$N_{\mathcal{F}} \in \mathbb{N}^+$ be the number of incomplete flights

and thus $N_{\mathcal{F}} = N_{FL} + N_{FS} + N_{FA} + (1 \text{ Transitioning flight})$ (3.2.11)

Definition 3: [PLANS] Let f_i denote the i^{th} flight, which consists of the 8-tuple:

$$f_i := (a_p, t_q, p_d, p_c, p_a, t_d, t_a, b_i^*) \quad (3.2.11)$$

where $a_p \in \mathcal{A}$ is a specific aircraft,

$t_q \in \mathcal{T}$ is an aircraft type,

$p_c \in \mathbb{P}$ is the current 4D location of the aircraft

$p_d \in \mathcal{P}$ is the departure airport, p_c overlaps p_d for a flight queued for departure

$p_a \in \mathcal{P}$ is the arrival airport, p_c overlaps p_a for a recently completed flight

$t_d \in \mathbb{R}^+$ is the scheduled UTC departure time/date, expressed in seconds,

$t_a \in \mathbb{R}^+$ is the scheduled UTC arrival time/date, expressed in seconds,

$b_i^* \in \mathbb{B}_i$: is the optimal 4D Business Trajectory assigned to i^{th} flight

Definition 4: [AIRCRAFT] Let \mathcal{A} represent the set of all uniquely-identifiable operational passenger aircraft, of any type, which may be scheduled to fly at any given moment.

$$\mathcal{A} := \{a_p : 1, \dots, N_{\mathcal{A}}\} \quad (3.2.12)$$

Definition 5: [AIRCRAFT TYPES] Let \mathcal{T} represent the set of all certified passenger aircraft types, each with their own performance models, passenger capacity and operational envelopes,

$$\mathcal{T} := \{t_q : 1, \dots, N_{\mathcal{T}}\} \quad (3.2.13)$$

where $N_{\mathcal{T}}$ is the number of certified passenger aircraft types

Definition 6: [AIRPORTS] Let \mathcal{P} represent the set of all recognised aerodromes possessing an ICAO 4-letter designation:

$$\mathcal{P} := \{p_k : 1, \dots, N_{\mathcal{P}}\} \quad (3.2.14)$$

where $N_{\mathcal{P}}$ is the number of recognised aerodromes

Definition 7: [RUNWAYS] Also let every $p_k \in \mathcal{P}$ denote the pair:

$$p_k := (L, R) \quad (3.2.15)$$

where: $L \in \mathbb{R}^3$ is the airport 3D location,
 $R \in \mathbb{N}^+$ is the assigned runway (and hence take-off/landing direction)

Definition 8: [AIRSPACE] Let \mathbb{P} represent the 4 dimensional (4D) infinite set of all flyable spatio-temporal points on the 20,000 meter airspace manifold surrounding planet Earth, such that every element $p \in \mathbb{P}$ consists of the 4-tuple:

$$\forall p \in \mathbb{P}, \quad p := (\varphi, \lambda, h, t) \quad (3.2.16)$$

$$\text{thus:} \quad \mathbb{P} \subset \mathbb{R}^4 \quad (3.2.17)$$

where: $\varphi \in (-\frac{\pi}{2}, +\frac{\pi}{2})$, is the Latitude in radians
 $\lambda \in (-\pi, +\pi)$, is the Longitude in radians
 $h \in \mathbb{R}^+$, is the Altitude above means sea level in meters
 $t \in \mathbb{R}^+$, is the UTC time in seconds

Definition 9: [TRAJECTORIES] Let \mathbb{B}_i represent the infinite set of feasible 4D Business Trajectories for the i^{th} flight in question, f_i , between its current location p_c to the destination airport p_a , taking into account the set of all airspace obstacles \mathbf{O} and all other aircraft.

$$\mathbb{B}_i := \{b_i^j : 1, \dots, \infty\}, b_i^* \in \mathbb{B}_i \quad (3.2.18)$$

where $b_i^j \in \mathbb{B}_i$ is the j^{th} feasible Business Trajectory for flight, f_i
 $b_i^* \in \mathbb{B}_i$ is the optimal Business Trajectory for flight, f_i

Definition 10: [TRAJECTORY NODES] Let every $b_i^j \in \mathbb{B}_i$ represent a time ordered vector of trajectory nodes (ie: timed waypoints) that uniquely define the 4D Business Trajectory taken by a flight f_i :

$$b_i^j := [n_k : 1, \dots, N_p], n_k \in \mathbb{P} \quad (3.2.19)$$

$$\therefore \mathbb{B}_i \subseteq \mathbb{P}^{N_p} \quad (3.2.20)$$

where: $N_p \in \mathbb{N}^+$ denotes the number of nodes (ie: waypoints) in trajectory b_i^j

Definition 11: [TRAJECTORY INTERPOLATION] Also let \mathfrak{R} denote the vector-valued deterministic mapping that uniquely generates a high resolution trajectory vector from the smaller vector of trajectory nodes in b_i^j :

$$\mathfrak{R} : \mathbb{P}^{N_p} \mapsto \mathbb{P}^{uN_p} \quad (3.1.21)$$

where: $u \in \mathbb{N}^+$ is the upsampling rate for a standardised interpolating function such as a Non-Uniform Rational B-Spline (NURB).

Definition 12: [TRAJECTORIES] Let $\mathbb{B}_{\mathcal{F}_A}^*$ represent the finite optimal set of 4D Business Trajectories individually assigned to all the active flights in set $\mathcal{F}_A := \{f_i : 1, \dots, N_{\mathcal{F}_A}\}$

$$\mathbb{B}_{\mathcal{F}_A}^* := \{b_i^* : 1, \dots, N_{\mathcal{F}_A}\} \quad (3.2.22)$$

$$\text{similarly:} \quad \mathbb{B}_{\mathcal{F}_C}^* := \{b_i^* : 1, \dots, N_{\mathcal{F}_C}\} \text{ for completed flights} \quad (3.2.23)$$

$$\text{similarly:} \quad \mathbb{B}_{\mathcal{F}_L}^* := \{b_i^* : 1, \dots, N_{\mathcal{F}_L}\} \text{ for long term scheduled flights} \quad (3.2.24)$$

$$\text{similarly:} \quad \mathbb{B}_{\mathcal{F}_S}^* := \{b_i^* : 1, \dots, N_{\mathcal{F}_S}\} \text{ for short term scheduled flights} \quad (3.2.25)$$

Definition 13: [TRAJECTORY COST] Let Ξ denote the deterministic scalar-valued mapping that uniquely generates an arbitrary positive cost metric as a function of any interpolated trajectory vector. Also let ${}^i_j J \in \mathbb{R}^+$ denote this cost metric for the trajectory j of flight f_i .

$$\Xi: \mathbb{P}^{uN_P} \mapsto \mathbb{R}^+ \quad (3.2.26)$$

thus ${}^i_j J = \Xi(\mathfrak{R}(b_i^j))$ (3.2.27)

and ${}^i_* J = \Xi(\mathfrak{R}(b_i^*))$ (3.2.28)

Definition 14: [SYSTEM COST] Let $J_{\mathcal{F}}$ denote the systemic (overall) excess % cost incurred by the optimised global air traffic system \mathcal{S} to accommodate all the flights in the union of set $\mathcal{F}_S \cup \mathcal{F}_A$

then $J_{\mathcal{F}} = J_{\mathcal{F}_S} + J_{\mathcal{F}_A}$ (3.2.29)

thus
$$J_{\mathcal{F}} = \frac{100}{N_{\mathcal{F}_S}} \sum_{i=1}^{N_{\mathcal{F}_S}} \left(\frac{{}^i_j J}{{}^i_* J} - 1 \right) + \frac{100}{N_{\mathcal{F}_A}} \sum_{i=1}^{N_{\mathcal{F}_A}} \left(\frac{{}^i_j J}{{}^i_* J} - 1 \right)$$
 (3.2.30)

also
$$J_{\mathcal{F}_L} = \frac{100}{N_{\mathcal{F}_L}} \sum_{i=1}^{N_{\mathcal{F}_L}} \left(\frac{{}^i_j J}{{}^i_* J} - 1 \right)$$
 (3.2.31)

where $J_{\mathcal{F}_L}$ is the future excess % cost that system \mathcal{S} will incur to accommodate all the flights b_i^j in \mathcal{F}_L
 $J_{\mathcal{F}_S}$ is the future excess % cost that system \mathcal{S} will incur to accommodate all the flights b_i^j in \mathcal{F}_S
 $J_{\mathcal{F}_A}$ is the remaining excess % cost incurred by system \mathcal{S} to accommodate all the flights b_i^j in \mathcal{F}_A
 ${}^i_* J$ is the cost of flying b_i^* which is the optimal trajectory for any flight f_i taken in isolation

Definition 15: [AIRCRAFT IN SERVICE] Also let \mathcal{A}^F be the set of all actively flying passenger aircraft, and \mathcal{A}^Q be the set of all idle passenger aircraft on the ground while they are queued for flights.

$$\mathcal{A}^F := \{a_p : 1, \dots, N_{\mathcal{A}^F}\} \quad (3.2.32)$$

$$\mathcal{A}^Q := \{a_p : 1, \dots, N_{\mathcal{A}^Q}\} \quad (3.2.33)$$

where $N_{\mathcal{A}^F} = |\mathcal{A}^F| = |\mathcal{F}_A| = N_{\mathcal{F}_A}$ (3.2.34)

and $N_{\mathcal{A}^Q} = |\mathcal{A}^Q| = |\mathcal{F}_S| = N_{\mathcal{F}_S}$ (3.2.35)

therefore $N_{\mathcal{A}^F} \in \mathbb{N}^+$ is the number of airborne aircraft

therefore $N_{\mathcal{A}^Q} \in \mathbb{N}^+$ is the number of queued aircraft

Definition 16: [CATM PROCESSING NODES] Let \mathcal{G} represent the set of ground CATM processing nodes and let all aircraft in set \mathcal{A} also be CATM processing nodes.

$$\mathcal{G} := \{g_i : 1, \dots, N_{\mathcal{G}}\} \quad (3.2.36)$$

such $\mathcal{A} = \mathcal{A}^F \cup \mathcal{A}^Q$ (3.2.37)

that

where $N_{\mathcal{G}} \in \mathbb{N}^+$ is the number of ground based CATM nodes

$N_{\mathcal{A}} = (N_{\mathcal{A}^F} + N_{\mathcal{A}^Q})$ is the number of aircraft based CATM nodes

Definition 17: [CATM Network] Let \mathcal{D} represent the set of all CATM processing nodes in the system:

$$\mathcal{D} := \mathcal{A} \cup \mathcal{G} \quad (3.2.38)$$

Definition 18: [PROXIMITY] Let d_{pq} denote the distance between points of closest proximity of the latest-available optimal trajectories of a flight f_p and a flight f_q .

such that
$$d_{pq} = \min \{ |\mathfrak{R}(b_p^*) - \mathfrak{R}(b_q^*)| \}$$
 (3.2.39)

Definition 19: [CLUSTERS] Let \mathcal{C} denote the set of clusters of all flights in \mathcal{F}_S and \mathcal{F}_A . Each flight is the head-node of a unique cluster, which it then consults to generate its trajectories. There are as many clusters as there are flights.

$$\mathcal{C} := \{c_i : 1, \dots, N_{\mathcal{C}}\} \quad (3.2.40)$$

such that: $N_{\mathcal{C}} = |\mathcal{C}| = |\mathcal{F}_S \cup \mathcal{F}_A| = N_{\mathcal{F}_S} + N_{\mathcal{F}_A}$ (3.2.41)

therefore: $c_i \subseteq \{\mathcal{F}_S \cup \mathcal{F}_A\}$ (3.2.42)

and let: $N_{c_i} = |c_i|$ (3.2.43)

where $c_i \in \mathcal{C}$ is the cluster corresponding to flight f_i which acts as its head node.
 $N_{\mathcal{C}} \in \mathbb{N}^+$ is the number of clusters
 $N_{c_i} \in \mathbb{N}^+$ is the number of flights joining flight f_i in a cluster c_i (including f_i)

Definition 20: [MEMBERSHIP IN CLUSTER] Let d_{LIM} denote the distance threshold to flight f_p that defines which flight f_q can join a cluster c_p .

$$\forall f_q \in \{\mathcal{F}_S \cup \mathcal{F}_A\}: \quad f_q \in c_p \quad \text{iff} \quad d_{pq} \leq d_{LIM} \quad (3.2.44)$$

Definition 21: [WEATHER] Let \mathcal{W} represent the weather system by the pair:

$$\mathcal{W} := (W, C) \quad (3.2.45)$$

where: $W: \mathbb{P} \rightarrow \mathbb{R}^3$ is a velocity vector field describing wind flow patterns over time
 C is the time-variant set of non traversable weather objects.

Definition 22: [WEATHER CELLS] Let C be the time variant set of non-traversable weather objects:

$$C := \{c_k : 1, \dots, N_C\} \quad (3.2.46)$$

where N_C is the number of non traversable weather objects

Definition 23: Let the k^{th} weather object, c_k , denote the 4-tuple:

$$c_k := (\varphi_k, \lambda_k, h_k, r_k) \quad (3.2.47)$$

where: $\varphi_k \in (-\frac{\pi}{2}, +\frac{\pi}{2})$, is the time-variant Latitude in radians
 $\lambda_k \in (-\pi, +\pi)$, is the time-variant Longitude in radians
 $h_k \in \mathbb{R}^+$, is the time-variant Altitude above mean sea level in meters
 $r_k \in \mathbb{R}^+$, is the time-variant Radius of the weather object in meters

Definition 24: [RESTRICTIONS] Let \mathcal{R} represent the set of non-traversable airspace restrictions:

$$\mathcal{R} := \{r_k : 1, \dots, N_{\mathcal{R}}\} \quad (3.2.48)$$

Definition 25: Let k^{th} restricted zone, r_k , denote the 4-tuple:

$$r_k := (VRT_k, FLT_k, FLB_k, SCD_k) \quad (3.2.49)$$

where $VRT_k \subset \mathbb{R}^3$ is the set of vertices defining the restricted zone
 $FLT_k \in \mathbb{N}^+$ is the top of restricted zone expressed as a flight level
 $FLB_k \in \mathbb{N}^+$ is the bottom of restricted zone expressed as a flight level
 $SCD_k \subset \mathbb{N}^+$ is the time schedule during which the restriction is in force.

Definition 26: Let relationship \mathbf{R} be defined as the union of three sub relationships R_L , R_S and R_A

$$R := \{R_L \cup R_S \cup R_A\} \quad (3.2.50)$$

such that R_L : $\forall f_i \in \mathcal{F}_L$, ground based decision makers, \mathcal{G} collaborate and take into account long term future predictions about the set of obstacles \mathbf{O} to assemble the optimal set of long term future trajectories, $\mathbb{B}_{\mathcal{F}_L}^*$ by choosing b_i^* from each set \mathbb{B}_i s.t. $J_{\mathcal{F}_L}$ is minimised when calculated over $\mathbb{B}_{\mathcal{F}_L}^*$. This process is iterative and takes into account changing conditions. It ends when flight f_i is assigned a mature trajectory b_i^* and thus f_i transitions from \mathcal{F}_L to \mathcal{F}_S (3.2.51)

and R_S : $\forall f_i \in \mathcal{F}_S$, the set of both idle and airborne decision makers \mathcal{A} collaborate and take into account current conditions and near term future predictions about the set of obstacles \mathbf{O} to assemble the optimal set of near term trajectories, $\mathbb{B}_{\mathcal{F}_S}^*$ by choosing b_i^* from each set \mathbb{B}_i s.t. $J_{\mathcal{F}_S}$ is minimised when calculated over $\mathbb{B}_{\mathcal{F}_S}^* \cup \mathbb{B}_{\mathcal{F}_A}^*$. This process is iterative and takes into account changing conditions. It ends when flight f_i reaches time for departure and thus f_i transitions from \mathcal{F}_S to \mathcal{F}_A (3.2.52)

and R_A : $\forall f_i \in \mathcal{F}_A$, the set of both idle and airborne decision makers \mathcal{A} collaborate and take into account current conditions and near term future predictions about the set of obstacles \mathbf{O} to assemble the optimal set of active trajectories, $\mathbb{B}_{\mathcal{F}_A}^*$ by choosing b_i^* from each set \mathbb{B}_i s.t. $J_{\mathcal{F}_A}$ is minimised when calculated over $\mathbb{B}_{\mathcal{F}_S}^* \cup \mathbb{B}_{\mathcal{F}_A}^*$. This process is iterative and takes into account changing conditions. It ends when flight f_i reaches time for landing at its destination and thus f_i transitions from \mathcal{F}_A to \mathcal{F}_C (3.2.53)

then $\mathbb{B}_{\mathcal{F}_L}^*$ is the solution of the offline CATM problem.
 $\mathbb{B}_{\mathcal{F}_S}^*$ is the solution of the offline CATM problem as it transitions to become online. (3.2.54)
 $\mathbb{B}_{\mathcal{F}_A}^*$ is the solution of the online CATM problem.

Having described the air traffic system in symbolic terms we shall now describe in some detail the connection between the three sub-relationships of \mathbf{R} . The details of the algorithms that can be used to generate optimised conflict-free flight trajectories will be the topic of chapter four.

3.3 FLIGHT GENESIS – THE 4D BUSINESS TRAJECTORY

Flights are defined by airlines, often many months in advance. The CATM system proposed is subsequently tasked with shouldering all responsibility for generating sets of conflict free trajectories that deviate as little as possible from the ideal Initial Shared Business Trajectories (ISBTs) as requested by the airlines. This is a continuous process and encompasses all phases of flight and pre-flight planning. The entire concept revolves about the fundamental notion of a 4D *Business Trajectory* (4DBT) [3.1]. The 4DBT is one of the hallmarks of the Single European Sky (SES) initiative [3.2] that eventually led to the introduction of the SESAR Research Programme [3.3]. A business trajectory is the least common denominator in the system and is essentially the bargaining unit in SES. This departs from the previous emphasis that was placed on sectors and airways. Aircraft will no longer be required to vie for a slot in a structured airspace. They will rather be allowed to negotiate for their own ideal trajectory in free space, while respecting airspace regulations, minimum flight separation, airspace restrictions and danger zones. They might also have to purchase their right for a business trajectory at market (auction) prices.

The business trajectory is therefore no more than a 4D trajectory which has been balanced against the requirements of all other concurrent users of the airspace, while trying to minimise flying-time, fuel consumption and environmental impact. It can be assumed that airlines will strive for trajectories that minimise their own cost. CATM therefore rebalances an airline's requirements for profit against social and environmental obligations. Each airspace user places a request for an Initial Shared Business Trajectory (ISBT) from the ATM system and this is met as closely as the constraints allow. CATM makes a distinction between scheduled flights and non scheduled flights as follows:

- **Scheduled flight:** “A flight for which an Initial Shared Business Trajectory exists at least 1 week before the day of operation”
- **Non-scheduled flight:** “A flight for which the Initial Shared Business Trajectory is made available on or shortly before the day of operation itself”

It stands to reason that late ISBT requests will have to cost significantly more than those made much earlier. This incentivises early planning by airlines and lends itself for a better coordinated, more predictable and more efficient use of the airspace. Most ISBTs would be expected to be filed by the airlines, months or years in advance. The CATM algorithm must then operate in two very distinct phases:

- **Strategic Phase Optimisation (SPO):** This is the proactive part of the system and deals with the long term pre-planning associated with processing ISBT requests in advance of the actual flights. It is involved in the issuance of business trajectories that have been scrutinised, adjusted as necessary and accepted. The strategic phase is not a time critical system and can therefore operate offline, using as many resources as necessary. Supercomputers are ideal for such a task. In our model it is represented by the relationship R_L .

- **Tactical Phase Optimisation (TPO):** This is the reactive part of the system and deals with the short term requirements and constraints faced by each flight during (and just before) execution. This part of the system must be lightweight and fast since it must be able to operate in real-time and is therefore always online. This phase is again split into two parts to distinguish between imminently queued flights and currently airborne flights. These two sub phases must be coordinated to allow real-time airspace capacity control, and a smooth transition between phases after take-off. The TPO corresponds to the role shouldered today by human air-traffic controllers. In our model it is represented by the seamless combination of R_S and R_A .

3.3.1 The Strategic Offline Phase

In CATM, the strategic phase of the algorithm solves the offline optimisation problem to minimise a cost function, which favours, among other things, the lowest fuel consumption, lowest environmental impact and the shortest final time for each journey of each aircraft, subject to simulated and modelled expected future constraints.

3.3.1.1 Some Definitions

For the sake of consistency, the following definitions pertaining to the temporal boundaries employed in all flights, will apply. Several of the time quantities are absolute and are ordinarily expressed in terms of the Coordinated Universal Time (UTC), which is the reference taken with respect to the 0° Prime Meridian at the Royal Observatory in Greenwich, United Kingdom.

- $[T_0^i]$ = This is the UTC instant at which flight f_i commences its take-off run.
= This is equivalent to the *Expected Time of Departure* (ETD)
- $[T_F^i]$ = This is the UTC instant at which flight f_i touches down at its destination.
= This is equivalent to the *Expected Time of Arrival* (ETA)
- $[\vec{T}_0^i]$ = This is the take-off instant of flight f_i relative to itself. It is always = 0.
- $[\vec{T}_F^i]$ = This is the touch-down instant of flight f_i relative to itself.
= This is equivalent to the duration of flight f_i .
- [T] = This is simply the current UTC time.

3.3.1.2 Long Term Scheduling (Flow Management)

Flying schedules are established based on the load-factor at each airport and traffic density at each point in space. This part of the algorithm takes over contemporary ATFM functions and is run offline on a daily basis using a centralised Strategic-Phase Optimiser (SPO). This would be executed on a large, fault-tolerant, distributed cluster of ground-based supercomputers designed specifically for the purpose of running this algorithm.

The SPO attempts to strike a balance between all entities which have expressed an interest in using some part of the airspace on a particular day and time. Each airline submits an *Initial Shared Business Trajectory* (ISBT). Each of these business trajectories is subjectively optimal and is used as a seed for the SPO algorithm. These ISBTs would have already been optimised for fuel consumption, environmental impact and other aspects by each aircraft operator, in isolation. However, the operator lacks the

visibility to take into account any traffic or weather constraints. This role is relegated to the centralised SPO.

Initially, the SPO executes without ETA boundary constraints, to allow the best possible trajectories to be generated, while alternating with a combinatorial optimiser to solve sequencing and scheduling issues. As the process continues, new business trajectories are generated for each aircraft and are communicated back to the owners. Airlines may in-turn choose to accept the changes or submit modified requests. If the outcome is deemed unsatisfactory to the airline, they may also elect to cancel the flight altogether. In order to deal with these changes in a timely manner, the process must start early, and up to a year in advance [$T_0^i + 365\text{days}$] in order to make sure that airspace and airport schedules are reasonably mature by the time these need to be advertised to customers. After about six months into this process, the trajectories are deemed to be reasonably converged. At this point, each ISBT^{*i*} which reaches [$T_0^i + 180\text{days}$] is upgraded to a new status called a *Mature Shared Business Trajectory* (MSBTs) and is published. This permits airlines to issue provisional time-tables and hence start accepting ticket reservations from their clients.

3.3.1.3 Near Term Queuing

The optimisation process needs to continue running regularly right up to [T_0^i]. However, it changes in objective to solve a fixed-time boundary problem to guarantee the now-agreed ETA schedules. The frequency of these optimisation-cycles gradually increases as the time of execution approaches. On the last day before execution, starting at [$T_0^i + 24\text{h}$], substantial processing power is dedicated to refine all those MSBTs that are about to commence within the next 24 hours, taking into account weather forecasts. Optimisation-cycles for these MSBTs, can run every few minutes. During the last few hours, say starting at [$T_0^i + 180\text{m}$], MSBTs are assigned to specific aircraft. The problem is handed over to the Tactical Phase Optimiser (TPO), where on-board computation takes over MSBT optimisation. The optimisation cycles are now run in real-time in conjunction with all queued and airborne aircraft. This increase in cycle frequency is crucially important to allow the system to rapidly take into account the latest updates in weather and traffic conditions. This information is continuously collected by all airborne aircraft as they execute the tactical phase of the algorithm. By generating the most up-to-date trajectories, the system ensures that no aircraft departs before there is a sufficiently high confidence that its MSBT can be accurately completed without modification. This helps to ensure that each aircraft has a sporting chance to accurately meet all its 4D *Target Window* (TW) contractual obligations, the ETA, and ultimately, its landing slot at the destination airport. Target windows are a low number of fixed 4D points in space-time that the flight is expected to traverse. They give the system some measure of determinism.

As each business trajectory f_i reaches its respective [T_0^i], and just before the aircraft commences its take-off run, the business trajectory together with all TWs and the ETA are frozen. It is now called a *Reference Business Trajectory* (RBT). This information is permanently recorded, uploaded to the aircraft's 4D-FMS and broadcast to all other aircraft as well as the destination airport. The RBT^{*i*} represents the intention

of the aircraft for the duration of that flight f_i . If any further modifications to the RBT are required in response to unforeseen changes in circumstances during the flight, this is now the sole responsibility of the onboard TPO. However, unless major disruption occurs, the degrees of freedom for changes to the RBT are limited to minor modification in the inter-TW portions of the RBT.

A measure of flexibility must be retained in the system in the form of scheduled slack. In the event of an Aircraft missing its take-off slot, the flight is automatically deferred. Ground operations will advise the SPO whether the flight is likely to be delayed or cancelled. If the flight is cancelled (such as due to aircraft malfunction) the MSBT is withdrawn and all resources are returned to the CATM pool and will be shared by the other MSBTs in the queue. As a result, take-off slots might be shifted forward to relocate the newly freed slack. If the flight is merely delayed (such as due to late boarding of passengers), a new estimate for the desired $[T_0^i]$, is issued by the airline and new landing slots assigned. This estimate is compared with all remaining MSBTs in the queue on the basis of a priority factor which would be purchased by all airlines. It stands to reason that higher priority factors cost more money. If the RBT had already been issued, as in the case of an aborted take-off, an RBT recall request is issued to all other aircraft and the destination airport. The flight is removed from the pool and resources re-assigned as slack. A new near term request is then submitted just like any other unscheduled flight.

3.3.1.4 *Unscheduled flights*

When a flight request is issued at short notice it cannot not undergo the long term scheduling by the SPO and is therefore open to substantial variation depending on available traffic and runway slack. Similarly, aborted take-offs and delayed flights have to be treated like other unscheduled flights. Such flights are taken into account by CATM as they become available. However, precedence and “right of way” must be given to the long term prescheduled flights as these are contractually required to abide by their agreed tight schedules. Any changes here would have an undesirable ripple effect on any ongoing connections. In such cases CATM functions in a unilateral fashion. The optimisation algorithms still follow the same logic. However, the incumbent traffic is assumed to be immutable and the newly added flights are tasked with finding optimal trajectories among the existing traffic. These flights are only guaranteed a best effort approach.

3.3.2 Long-Term Sequential Coupling of Constraints

The SPO process is complicated by the fact that, in-reality, ATM is a continuous perpetual-time problem, that temporally, spans forward to infinity. In principle, one could say that the repercussions of today’s inclement weather or an airport workers’ strike would continue to bear some influence on the future global ATM problem for a very long time, albeit gradually decaying to an infinitesimally small degree. This is to say that all flights are (to some extent) constrained by the all previous conditions and actions taken by the preceding flights. The present (and past) is fixed and immutable, while the future must accommodate present constraints as they unfold. This

conceptually creates a receding time-horizon problem where constraints can only be imposed forward in time. This principle has to be employed throughout the SPO process such that freshly-optimised future MSBTs pertaining to $[T_0^i + \alpha]$ serve as constraints to MSBTs at time $[T_0^i + \alpha + \delta T]$ further into the future, all the way to $[T_0^i + 365 \text{ days}]$.

Two kinds of dependencies arise. The first is quite obvious and relates to the fact that aircraft need to be reused to service return or interconnecting flights. This may cause delays to propagate through flight schedules, which disrupts flight sequencing and upsets any combinatorial optimality of the schedule. The second kind of dependency is due to the interaction between concurrent trajectories of airborne flights. A flight might need to be lengthened to avoid conflict with another and this can generate a delay, which in turn affects other flights interacting with other parts of the trajectory, and ultimately the very same schedule. **Figure 3.3.1** illustrates graphically these two kinds of dependency.

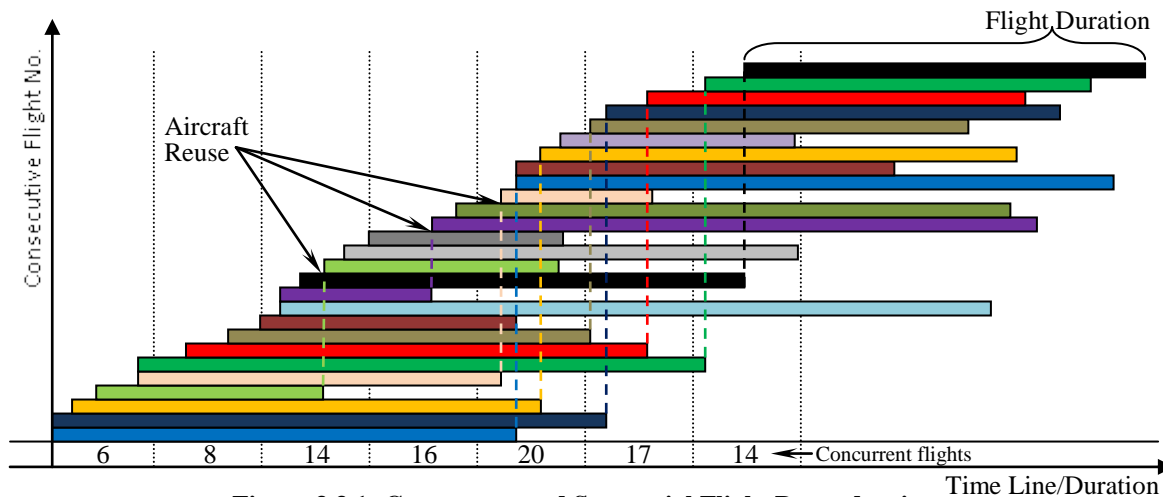


Figure 3.3.1: Concurrent and Sequential Flight Dependencies

Currently, the ATM system avoids propagating these long-term effects by effectively “resetting” itself every 24 hours. This decouples one day’s traffic problem from the next. However, this is only possible because most airport activity shuts down for several hours every night. This lull in airport activity provides enough “slack” in the diurnal traffic cycle to allow absorbing the cumulative delays generated over each day, without them spilling over into the next day’s schedule. Even the effects of large disruptions, such as a major accident at an airport, are practically completely recovered over a few days.

However, this method of operation is exceedingly wasteful of resources by always operating significantly below capacity. As airport day-time congestion increases, this practice cannot be expected to continue indefinitely. Traffic will continue to spill further into the night and market forces will eventually ensure that airport activity becomes a perpetual continuum with an ever decreasing variation between the rise and ebb in traffic. Newer, quieter, cleaner aircraft will gradually soften the current public opposition of continuous airport operations. As this new equilibrium settles in, the effects of long term constraint coupling will begin to bite.

That said, diurnal cycles of air traffic density will probably remain to some extent and this needs to be interpreted in the context of a global problem, which spans all time zones of the planet. The peak and trough in traffic spins like a wave with respect to the planet and in a way, one can view constraint propagation in this manner. A single SPO optimisation iteration sweeps 365 times round the planet and is a colossal computational task (**Fig 3.3.2**).

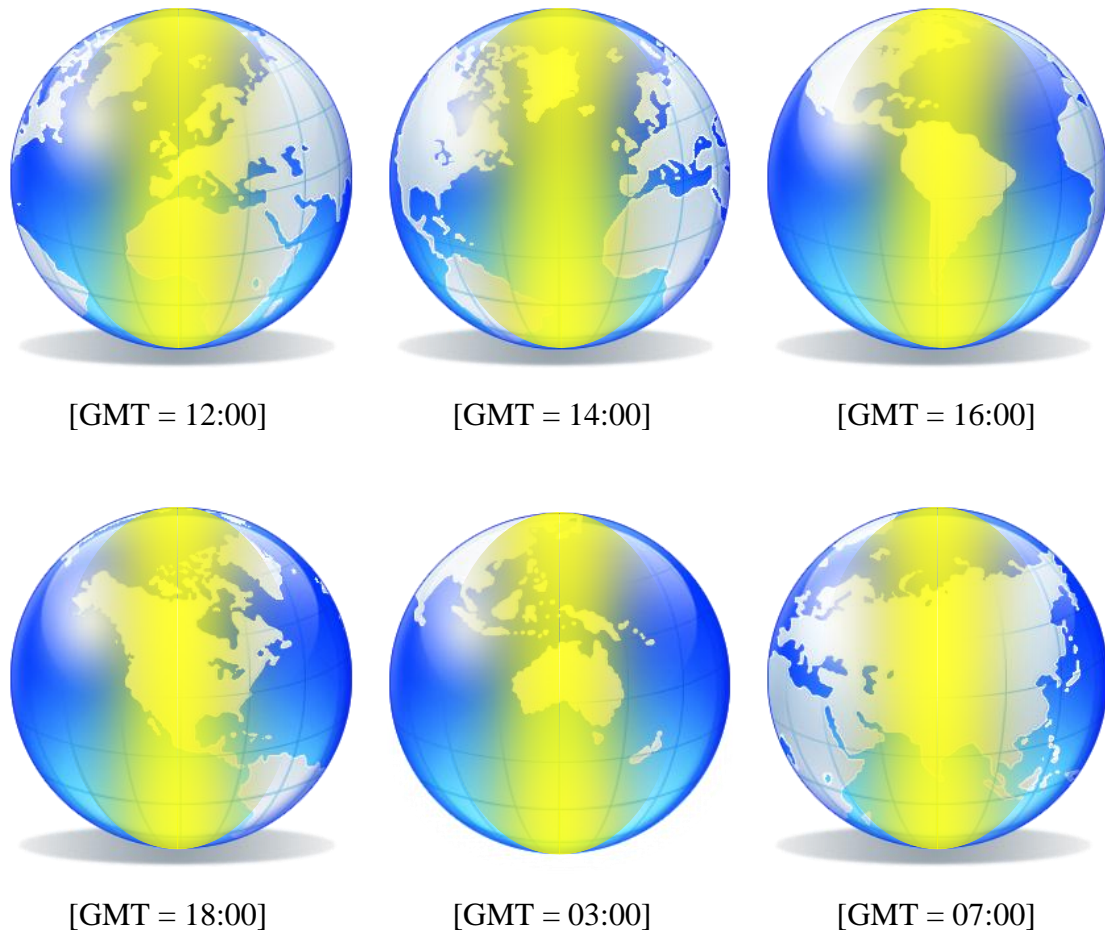


Figure 3.3.2: Diurnal Cycle in Traffic Density and Constraint Propagation

In the offline portion of CATM the phenomenon of propagation is accommodated by ensuring that all RBTs of flights commencing at any UTC time $[T]$ are optimised against all prior traffic commencing at $[T - x]$, where x is any interval shorter than the duration of each flight. This forms a temporal chain of dependencies, which must be solved by the SPO in the form of a receding time-horizon, which sweeps through the problem over successive iterations until it converges. In practice this takes place by using the latest system state information as the initial state for future projections. Flights are optimised in chronological order, and the many overlaps and sequence constraints ripple through in time. It stands to reason that the flights furthest into the future are subject to the greatest uncertainty.

3.3.3 The Tactical Online Phase

The Tactical Phase Optimiser (TPO) must deal with the unforeseen and unexpected. One would imagine weather to be one of the principle randomisers in the ATM problem. Think again! Humans prove to be even less predictable. Late passenger boarding, passengers falling ill, airport closures, runway accidents, volcanic eruptions, traffic diversions, security alerts, terrorist activity, industrial strike-action... are much harder to model and forecast. Some may share some interdependency or causal link to an underlying process, but others are purely zero-knowledge stochastic phenomena. The list goes on and on, and this might appear to unravel all that careful, planning performed over an entire year by the SPO.

3.3.3.1 *Response to Gradual Environmental deviations*

Practically all deviations in the expected weather conditions – however large they may be – evolve gradually over time. The same applies to the trajectories of all surrounding traffic. Surrounding aircraft may – for whatever reason – be uncooperative, unequipped or choose to violate its published RBT and take on an unforeseen twist in its trajectory. Yet, even in this case, the offending aircraft is still limited by relatively slow dynamics. Positions and velocities can only vary gradually and high frequency surveillance provides the CATM system with a real time picture of events.

The TPO is ideally suited to gradually re-optimize traffic for these kinds of gradual changes. By enforcing frequent optimisation sweeps, the last known optimal solution can only be slightly suboptimal with respect to the current theoretical best. The previous solution is therefore always used to seed the next solution. Now by always starting off from a good approximation of the optimal solution, very rapid re-convergence is essentially guaranteed. This makes the TPO very efficient to track (local) optima.

However, ensuring that the local optimum being tracked by the TPO is also globally optimal is not as trivial. (Global optima are special cases of local optima.) The CATM problem is fundamentally dynamically-non-convex and global optima might, over time, morph into globally suboptimal local optima. Thus a new global optimum might emerge in some other part of the search space that is not being tracked. The TPO addresses this by running global and local optimisation algorithms in tandem to ensure good problem-space coverage at all times.

3.3.3.2 *Response to Sudden Catastrophic Events*

Even with sudden major disruption, very often all is not lost. The work done by the SPO pays-off *most* of the time, since the vast majority of flights are usually unaffected by these occasional regional hiccups. Even when problems do occur, very often, they are highly localised in space and time (such as late passenger boarding, runway accidents, and security alerts) and the system only needs to perform some slight adjustments in order to accommodate the emerging contingencies. The near-optimal solution provided by the SPO must contain just enough slack to be able to accommodate a few new constraints. The task of the TPO is precisely this. It performs these fine adjustments and bends the global traffic solution ever so slightly to take into account the

new conditions. The pain is distributed such that no part of the system bears the full brunt of the additional cost. This scheme must take care of the overwhelming majority of contingencies and ensures that there is insignificant future propagation of the effects.

There will of-course remain the possibility of having some large sudden disruption that causes a wide systemic ripple effect. For this reason, the TPO must be fully equipped to re-optimize the trajectories from scratch. It is worth noting that the size of problem that the TPO needs to solve, albeit large, is limited to the airborne and queued flights at any moment, and is therefore much smaller than the formidable 365day problem that the ground based SPO faces. The combined processing power on board airborne aircraft must be sized to ensure sufficiently fast reconvergence of active trajectories, from any random seed, to generate new optimal trajectories between the current position and the final destination.

3.3.3.3 Feedback To The Strategic Phase Optimizer

All RBT changes that result from the TPO real-time adjustments are continuously relayed back to the SPO. This feedback ensures that all the MSBTs in current incubation at the SPO are kept up to date with the latest traffic and weather constraints.

Airborne aircraft is uniquely well positioned to gather detailed real-time information about atmospheric conditions and weather phenomena. The distributed network created by flying aircraft with their onboard weather radars and sensors, creates a formidable telemetry capability that if tapped would greatly assist weather simulators on the ground to keep track of unfolding conditions. Weather is a chaotic system, which makes it highly sensitive to initial conditions and external stimuli. This imposes a fundamental limitation to how far out into the future forecasts can remain reliable. Thus accurate real time knowledge of the current weather situation, collected at numerous points round the globe has the potential for significantly enhancing the accuracy of forecasts, and as a result, SPO solutions.

3.4 MODELLING FOR CATM SIMULATION

In CATM, simulation of the air traffic system takes on three distinct roles with highly divergent requirements. Thus there is a clear case for substantial research and development effort in all three directions. A number of separately validated models of varying precision must be brought together to mimic the behaviour of the key sub-systems in the ATS.

CONCEPTUAL VALIDATION: During development, the basic ideas behind CATM must be empirically tested for stability, feasibility and algorithm convergence. Such basic research can only take place in a small-scale simulated environment. For this reason, computationally-lightweight models are highly desirable to allow adequate exploration of numerous ideas in finite time. Hence, models used at this stage need not be very accurate. However, they must be sufficiently *representative* of the typical behaviour of the sub-systems making-up the simulated ATS.

OPERATIONAL VALIDATION: Due to the life critical nature of air transport, the validation of the entire CATM system from a reliability perspective, cannot take place on a live system. Detailed empirical evaluation must be performed in a high fidelity simulated environment, requiring substantial modelling and computational effort. A cost-benefit trade-off must be established between the accuracy of simulation and the computational complexity incurred. This is also the stage where operational-grade algorithms are developed, verified and finally certified by air transport authorities.

CATM OPERATION; The forward-looking and predictive nature of CATM requires a significant degree of fast time simulation in order to extrapolate (and take into account) future traffic patterns, weather and aircraft behaviour. Indeed, the SPO (and to a lesser extent, the TPO) is highly dependent on this. Accurate models designed for efficient computation on avionics-grade CATM hardware must be developed for this purpose. The key to high performance is to design dedicated hardware in tandem with the algorithms.

In the foregoing sections we shall describe a number of place-holder sub-system models that are suitable for conceptual CATM simulation. A number of these models were implemented and tested to gain better insight into their utility as place-holder models in an eventual CATM simulator. Where there was a conflict, emphasis was placed on simplicity rather than high accuracy. Matlab was used for the most part and the models were qualitatively or quantitatively validated. We will discuss the following models:

- | | |
|------------------------------|---|
| ATMOSPHERIC MODELS: | • The International Standard Atmosphere (ISA) |
| AIRCRAFT MODELS: | • A BADA-based 3-DOF Point Mass Model |
| TRAFFIC MODELS: | • A Probabilistic Traffic Model |
| WEATHER MODELS: | • A Wind Pattern Model |
| | • A Cloud and Weather cells Model |
| ENVIRONMENTAL IMPACT MODELS: | • Aircraft Emission Models (CO ₂ , NO _x , Soot, HC) |
| | • A Noise impact Model |
| | • A Water Vapour and Contrails Model |
| AIRPORT MODELS: | • A Probabilistic delay and disruption model |
| | • An Aircraft Scheduling Model |

3.5 ATMOSPHERIC MODELLING FOR ATM

Meaningful ATM simulation requires a representative atmospheric model in order to generate trajectories, which are aerodynamically feasible, let alone optimal. The atmospheric models need to be simple enough for fast inner-loop calculations, yet complete enough to provide a realistic macroscopic picture of the atmosphere.

There are various popular standards that define the atmosphere. The International Standard Atmosphere (1975), divides the atmosphere into a number of layers up to a geopotential height of 50km. The ICAO Standard Atmosphere (1993) overlaps ISA for the first 50km but extends the model to 80 km. The US Standard Atmosphere (1976) overlaps both, but extends the model further to 1000 km. Above about 86 km, atmospheric conditions vary widely depending on solar activity and its interaction with the geomagnetic field. This results in large variations in the temperature and density of the atmosphere, making any model unreliable at these altitudes. Luckily, aircraft is typically confined to the lowest two layers and the models are fairly representative at these altitudes. Because of the overlap only the International Standard Atmosphere will be discussed and this forms the basis for the rest of the project [3.4].

3.5.1 The International Standard Atmosphere

The International Standard Atmosphere (ISA) is the most widely used model and is based on an empirical model built around a dataset, which was standardised in 1975. It is available at a charge from the International Standards Organisation (ISO) as document ISO 2533:1975 [3.5]. The model is available as a large set of look-up tables. However, it may be condensed to a relatively simple, closed-form, algebraic abstraction. Better models take into account the significant variations with latitude as well as geographic variations, but for ATM simulation confined to the northern hemisphere European airspace, such detail is not required. The ISA model is considered valid for most mid-latitudes of the northern hemisphere. For the sake of standardisation, the ISA model fixes a number of atmospheric and geophysical variables, which will be treated as constants in this thesis. These variables constitute the standard ISA conditions:

T_0	is the mean-sea-level temperature;	288.15 K (15°C)
T_{trop}	is the tropopause transition temperature;	216.65 K (-56.5°C)
H_{trop}	is the tropopause transition height	@ $T_0 = 15^\circ\text{C}$; 11,000 m
ρ_{trop}	is the tropopause transition density	@ $T_0 = 15^\circ\text{C}$; 0.36392 Kg/m ³
ρ_0	is the mean-sea-level air density	@ $T_0 = 15^\circ\text{C}$; 1.225 Kg/m ³
a_0	is the mean-sea-level speed of sound	@ $T_0 = 15^\circ\text{C}$; 340.294 m/s
p_0	is the mean-sea-level air pressure ;	101325 N/m ²
K_T	is the temperature gradient with altitude;	-0.0065 K/m
R	is the real gas constant for air;	287.05287 m ² /Ks ²
g	is the mean-sea-level acceleration due to gravity;	9.80665 m/s ²

The Earth’s atmosphere self-stratifies into distinct bands, which are in constant thermodynamic equilibrium. They exhibit clearly distinct temperature/altitude gradients and in fact the ISA model defines the layers in these terms. These are affected by surface temperatures, which vary widely between the poles and the equator. In order to partly accommodate some of these variations, the ISA model provides a means for extending itself in relation to the mean terrestrial temperature. This also allows the model to take into account seasonal changes in mean temperature. **Table 3.5.1** describes these layers and their boundaries at standard ISA conditions taken at the 45th north parallel. However, the information pertaining to the Thermosphere, Thermopause and Exosphere is taken from the US Standard Atmosphere (1976) and represents a very approximate mean.

#	Layer Name	GeoPotential Height (m)	Geometric Height (m)	Temp. Grad (°C)	Temp. Base (°C)	Gravity (m/s ²)	Pressure Base (pa)	AirDensity (Kg/m ³)
0	Troposphere	0	0	- 6.5	+15.0	9.8066	101,325	1.2250E-0
1	Tropopause	11,000	11,019	0.0	-56.5	9.7727	22,632	3.6392E-1
2	Stratosphere 1	20,000	20,063	+1.0	-56.5	9.7450	5,474.90	8.8035E-2
3	Stratosphere 2	32,000	32,162	+2.8	-44.5	9.7082	868.02	1.3225E-2
4	Stratopause	47,000	47,350	0.0	-2.5	9.6622	110.91	1.4275E-3
5	Mesosphere 1	51,000	51,413	-2.8	-2.5	9.6499	66.939	8.6160E-4
6	Mesosphere 2	71,000	71,802	-2.0	-58.5	9.5888	3.9564	6.4211E-5
7	Mesopause	84,852	86,000	0.0	-86.2	9.5466	0.3734	6.9580E-6
8	Thermosphere	89,716	91,000	+4.0	-86.2	9.5318	0.1538	2.8600E-6
9	Thermopause	286,480	300,000	+0.04	+702	8.9427	8.770E-6	1.792E-11
10	Exosphere	864,071	1,000,000	- -	+726	7.3218	7.514E-9	3.561E-15

Table 3.5.1: Atmospheric Layers as defined by temperature gradients [3.5]

3.5.1.1 The Troposphere

The troposphere is the lowest atmospheric layer to which weather is largely confined. Temperature drops steadily with altitude at a rate of -6.5°C per km until it reaches an iso-thermal layer at the base of the stratosphere. The tropopause is defined as a boundary between the troposphere and the stratosphere and, at standard ISA conditions, is taken to lie at an altitude of 11,000m. However, this is only true at mid-latitudes and the exact location of the tropopause is latitude and temperature dependant and may lie anywhere between 20km, at the equator, down to 8km, at the poles.

To date, most commercial aviation, cruises within the lower two layers, namely the upper-troposphere and the lower-stratosphere. This allows jet-aircraft to take advantage of the very low air temperatures, which significantly improves engine thermodynamic performance. Moreover, the low air-density reduces drag, which in turn allows higher velocities to be achieved. **Figure 3.5.1** shows these layers graphically. For these reasons just mentioned and also to reduce the noise footprint on urban communities below, typically airliner jet-aircraft attempts to minimise the time spent in the troposphere [3.8].

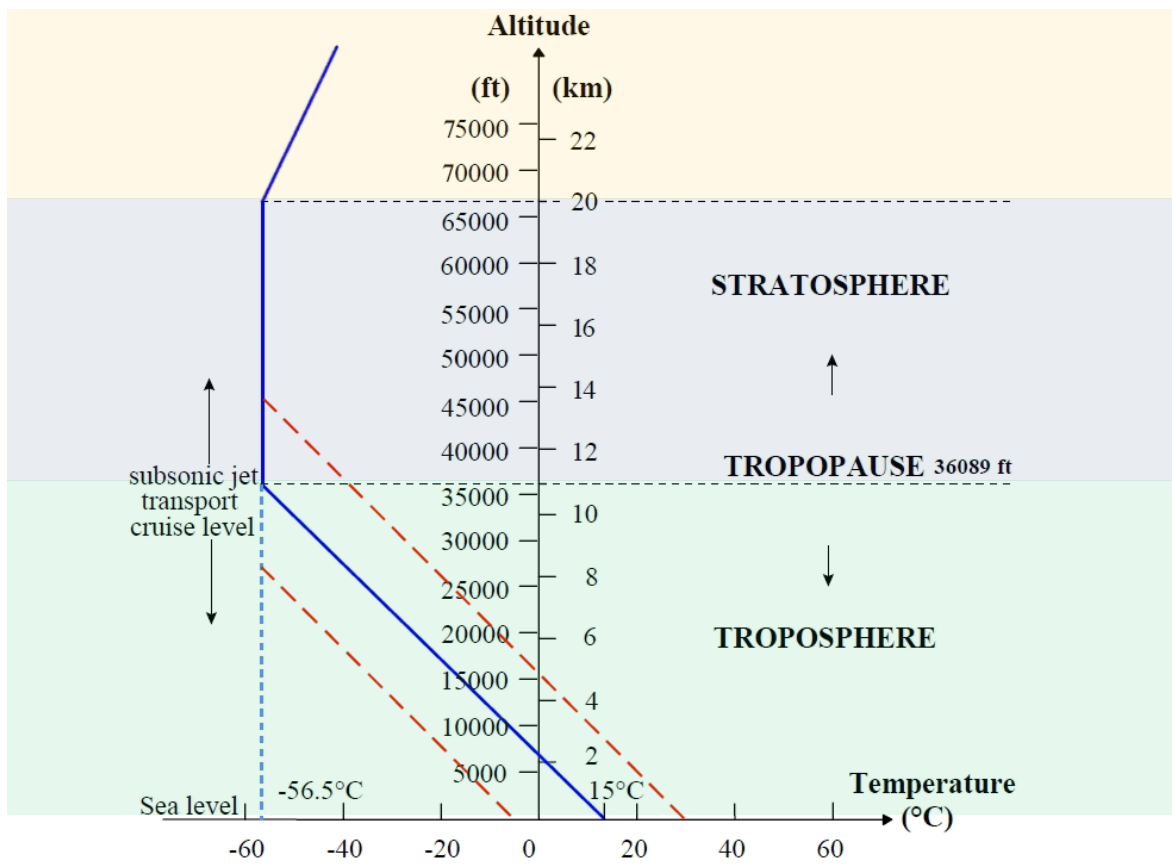


Fig 3.5.1: The International Standard Atmosphere (ISA) [3.8]

3.5.1.2 The Stratosphere

The stratosphere begins above the tropopause and extends all the way up to the stratopause, at about 47km above mean sea level. The stratosphere is itself stratified into a further three sub layers. The lowest of these, presents very little change in temperature with altitude and is therefore termed iso-thermal. The other two layers exhibit a modest but positive temperature gradient and are hence called inversion layers.

Due to the very low temperature of the lower stratosphere, water vapour is nearly absent and this precludes most cloud formation. This essentially eliminates most weather effects from the stratosphere and is another good reason why aircraft tend to fly at these altitudes. Although, the iso-thermal layer extends to 20km, the higher altitudes are not used by commercial air transport. One reason is that cosmic radiation increases dramatically with altitude due to lessened atmospheric shielding. Routine exposure to such high radiation levels poses a significant health risk for frequent travellers and particularly for the crew. Heavy radiation shielding is not economically feasible.

3.6 AIRCRAFT MODELLING FOR ATM

In ATM, mathematical models for aircraft are necessary for extrapolating future motion of aircraft and hence predict the short-term evolution of traffic scenarios. This forward-looking aspect of ATM is essential to identify future potential conflicts as early as possible. Moreover, the use of aircraft models (together with atmospheric models) allows the ATM system to establish an appropriate flight envelope of all aircraft and ensures that realistic radar-vectoring instructions are issued in relation to the aircraft type. Aircraft modelling can be taken to very high levels of fidelity, however, this comes at a substantial computational cost. In ATM, where simulations may concurrently involve hundreds or thousands of aircraft, a compromise favouring simplicity is struck.

3.6.1 Coordinate-System and Other Definitions

The Aerospace sciences abound with different coordinate systems, however, for the purposes of ATM, an appropriate coordinate system should be referred to the Earth frame of reference. This simplifies representation of aircraft in real geographical space. Hence, for all subsequent models an absolute, right-handed, North-East-UP (NEU), Earth-referred coordinate system is employed, where:

- x -axis points north, parallel to the Earth's surface, along a longitude curve. [m]
- y -axis points east parallel to the Earth's surface, along a latitude curve. [m]
- h -axis points upward, towards the sky, orthogonal to the surface. [m]
- ϕ is the bank angle that the wing-axis makes with Earth's surface [°]
- γ is the flight path angle the body axis makes with the Earth's surface [°]
- σ is the heading angle the body axis makes with the x -axis (North) [°]

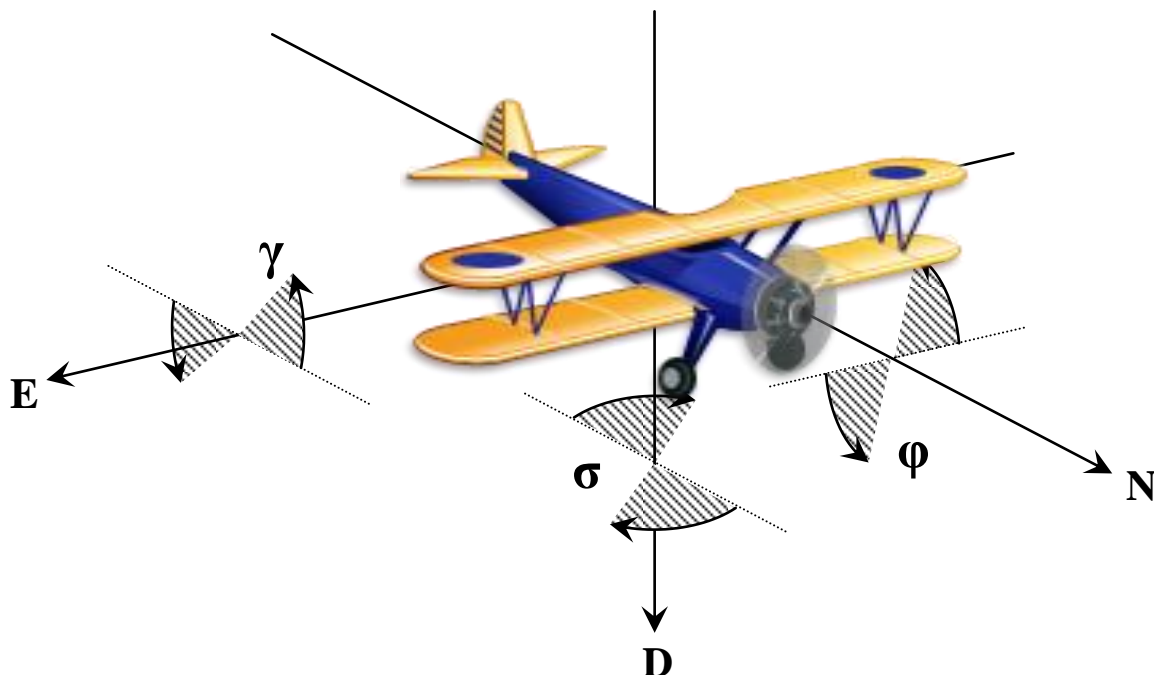


Fig 3.6.1: An aircraft shown flying northwards with [$\phi = 0$; $\gamma = 0$; $\sigma = 0$] Earth-referenced attitude

and: $\phi = 0$ for level flight, +ve banks an aircraft to the right
 $\gamma = 0$ for level flight, +ve flight path angle points an aircraft's nose skywards
 $\sigma = 0$ for north facing flight, +ve heading turns an aircraft eastwards

The illustration of **Figure 3.6.1** defines the NEU spatial and angular coordinates with respect to a fixed-wing aircraft that happens to be travelling north along a level trajectory. Other important parameters worth defining are listed below:

α	is the angle of attack that the airflow makes with the wings	[°]
m	is the total mass of the aircraft (including fuel)	[Kg]
η	is the thrust specific rate of fuel-burn	[Kg/kNs]
ρ	is the air density as a function of temperature	[Kg/m ³]
T	is the air temperature as a function of altitude	[Kelvin]
Γ	is the total instantaneous thrust from all engines	[kN]
g	is the acceleration due to gravity	[m/s ²]
w	is the [w_x, w_y, w_h] wind velocity vector in NEU coordinates	[m/s]
t	is the elapsed time of flight since take-off	[s]
C_L	is the coefficient of lift	[unit-less]
C_D	is the coefficient of drag	[unit-less]
PE	is the Potential Energy of an aircraft	[J]
KE	is the Kinetic Energy of an aircraft	[J]
NE	is the New Energy gained by an aircraft	[J]
LE	is the Lost Energy by an aircraft	[J]
R_{CP}	is the Reduced Climb Power engine derating coefficient	[unit-less]
μ	is an energy share factor to partition energy into PE or KE	[unit-less]
L	is the force of lift	[kN]
D	is the force of drag	[kN]
W	is the force of weight	[kN]
S	is the total surface area of the wings	[m ²]
V	is the true airspeed (TAS) of the aircraft along the body axis	[m/s]

3.6.2 The EUROCONTROL Base of Aircraft Data, (BADA)

The aerodynamic coefficients C_L and C_D and other detailed performance data for each aircraft type can be derived from specialised databases such as EuroControl's BADA (Base of Aircraft Data) [3.6]. BADA was developed to be a simplified semi-empirical model of aircraft performance that is sufficiently accurate for ATC applications [3.7] but omits detailed aerodynamic models, which have very little bearing from an ATM perspective. BADA includes information about a number of popular aircraft types, but also includes convenient "synonym" tables, which map the many different aircraft variants in use, to a core subset of aircraft types.

3.6.3 A Non-Holonomic, Kinematic Model for a Fixed-Wing Aircraft

For the purposes of ATM, all aircraft can be conveniently described by a 3-degree of freedom (3-DoF), 3-dimensional (3D), kinematic model for a rigid body. The large differences in scales between the large distances covered by an aircraft and the small effects caused by higher-order dynamics of the said aircraft, imply that this is an acceptable simplification for most scenarios. Aspects such as mass, moments-of-inertia and the various aerodynamic forces are of little consequence at the ATM scale, which often spans thousands of kilometres. Indeed, many commercial ATM simulators (such as Micronav BEST) may even avoid the 3-DoF model altogether, relying instead on purely rule-based kinematic models based on constant-radius arcs and straight lines. However, as the demand for accuracy increases, and given the capacity of today's computational hardware, such crude modelling may soon become outmoded.

Fixed wing aircraft (which account for the vast majority of air traffic) have several restrictions on the permissible manoeuvres they can perform. Such vehicles are termed *non-holonomic* since they have a number of degrees of freedom, which are not directly controllable. Non-holonomic motion is governed by strict relationships between the various degrees of freedom, meaning that final system state is path-dependent. The purpose of the kinematic model is to capture these relationships and pose them in terms of set of control variables. Moreover, further magnitude constraints have to be imposed on the controllable degrees of freedom. Parameters such as maximum bank-angle, velocity and service ceiling have to be taken into account in relation to the particular type of aircraft, passenger comfort considerations and the relevant regulations. The latter constraints define the flight envelope of an aircraft. If an ATM simulator is to produce flyable aircraft trajectories, it must employ an aircraft kinematic model, which embodies this flight envelope and non-holonomy.

The set of first-order, non-linear, time-invariant, differential equations (3.6.1) define the basic kinematic rules for defining the motion of a fixed wing aircraft in an NEU coordinate system. The wind vector, $\mathbf{w} = [w_x, w_y, w_h, 0, 0, 0]^T$, as calculated from a weather model, is additively combined to the aircraft component velocities [3.9].

$$\begin{aligned} \dot{x} &= V \cdot \cos(\sigma)\cos(\gamma) + w_x \\ \dot{y} &= V \cdot \sin(\sigma)\cos(\gamma) + w_y \\ \dot{h} &= V \cdot \sin(\gamma) + w_h \end{aligned} \quad (3.6.1)$$

Since for pure kinematics the aircraft is deemed mass-less, this model is unable to capture the relationship between attitude and airspeed. The airspeed and attitude derivatives $[\dot{\phi}, \dot{\gamma}, \dot{\sigma}]$ are therefore considered as independent inputs. The final model is therefore:

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{h} \\ \dot{\phi} \\ \dot{\gamma} \\ \dot{\sigma} \end{bmatrix} = \begin{bmatrix} \cos(\psi)\cos(\gamma) & 0 & 0 & 0 \\ \sin(\psi)\cos(\gamma) & 0 & 0 & 0 \\ \sin(\gamma) & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} V \\ \dot{\phi} \\ \dot{\gamma} \\ \dot{\sigma} \end{bmatrix} + \begin{bmatrix} w_x \\ w_y \\ w_h \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (3.6.2)$$

3.6.4 3D Point Mass Model for a Fixed Wing Aircraft

For accurate path planning, a purely kinematic model is often considered insufficient. A 3-DoF, point mass model (PMM) is usually preferred. This augments the kinematic model to include the mass of the aircraft, which is itself a variable due to the significant rate of change in fuel-mass associated with most aircraft as per equation (3.6.3)

$$\dot{m} = -\eta\Gamma \quad (3.6.3)$$

The basic forces acting on the point-mass aircraft are thrust, drag, lift and weight and during steady level flight these forces are in equilibrium. **Figure 3.6.2** depicts this scenario.

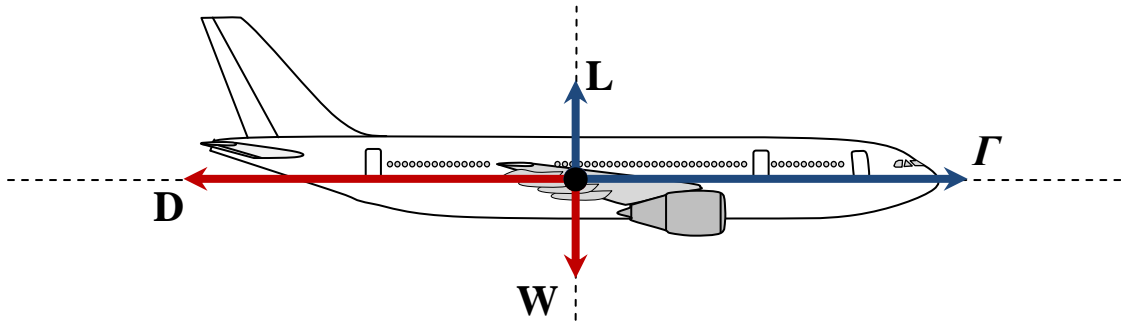


Fig 3.6.2: Aircraft forces during steady level flight

Since the engines are not being modelled, the thrust Γ is taken as an independent input. This assumes that ideal engines are able to deliver as much thrust as required, whenever required. The task for ensuring this, is usually relegated to the Full Authority Digital Engine Control System (FADEC), which micro-manages the engine for peak performance. However, in this model we ignore realities such as spool-up delay time and assume that the engine control system is doing its job perfectly.

Weight is, of course, mass dependant as per equation (3.6.4):

$$W = mg \quad (3.6.4)$$

3.6.4.1 AERODYNAMIC FORCES

Lift and drag are highly correlated forces, which depend heavily on the aircraft's aerodynamics and configuration such as the extension of flaps or slats. The equations for lift and drag are given below in eq. (3.6.5) and (3.6.6) respectively.

$$L = \frac{\rho C_L S}{2} V^2 \quad (3.6.5)$$

$$D = \frac{\rho C_D S}{2} V^2 \quad (3.6.6)$$

The coefficients of lift C_L and drag C_D are functions of the angle of attack α and are defined in terms of their nominal values by the empirical equations (3.6.7) and (3.6.8)

$$C_L = \hat{C}_L(1 + c\alpha) \tag{3.6.7}$$

$$C_D = \hat{C}_D(1 + b_1\alpha + b_2\alpha^2) \tag{3.6.8}$$

The parameters c , b_1 and b_2 are empirically determined curve-fitting coefficients. However, in the usual case of airliners flying level at trimmed conditions the angle of attack is either small or nearly zero ($\alpha \approx 0$) and will be in any case nearly constant. This means that the aerodynamic coefficients C_L and C_D reduce to their nominal values \hat{C}_L \hat{C}_D most of the time. This simplification will be assumed throughout the model.

3.6.4.2 AERODYNAMIC FORCES – ALTERNATIVE EVALUATION

An alternative (BADA compatible) way for determining the lift L of an aircraft is by considering the balance of forces during steady-state flight, where an aircraft necessarily needs to balance its weight due to gravity with the force of lift so that:

$$\begin{aligned} L &= W \\ L &= mg \end{aligned} \tag{3.6.9}$$

However, when an aircraft banks, this balance must be maintained throughout the turn for the aircraft not to change its vertical velocity. The lift must therefore increase, such that its vertical component remains sufficient to match the weight of the aircraft as indicated in **Figure 3.6.3**. Accordingly, equation (3.6.10) shows the expression for lift including a correction cosine term for the non-zero bank angle φ .

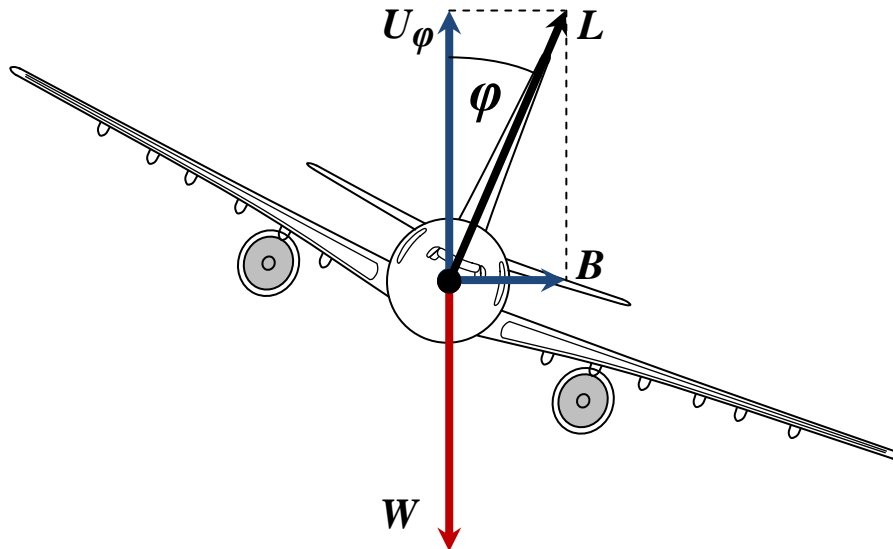


Fig 3.6.3: The balance of forces during a turn

$$\begin{aligned} U_\varphi &= W \\ L \cdot \cos(\varphi) &= mg \\ L &= mg/\cos(\varphi) \end{aligned} \tag{3.6.10}$$

A similar situation arises when the aircraft pitches upwards or downwards such that the flight path angle γ is no longer zero. In this scenario, as depicted in **Figure 3.6.4**, the lift must also increase to compensate. The equation (3.6.11) for lift in this case includes a correction cosine term for the non-zero flight path angle.

$$\begin{aligned} U_\gamma &= W \\ L \cdot \cos(\gamma) &= mg \\ L &= mg/\cos(\gamma) \end{aligned} \quad (3.6.11)$$

Naturally, the two effects can be combined such that both flight path angle and bank angle are non-zero. In this case, the net effect can be adequately modelled by equation (3.6.12)

$$L = mg/(\cos(\gamma) \cdot \cos(\varphi)) \quad (3.6.12)$$

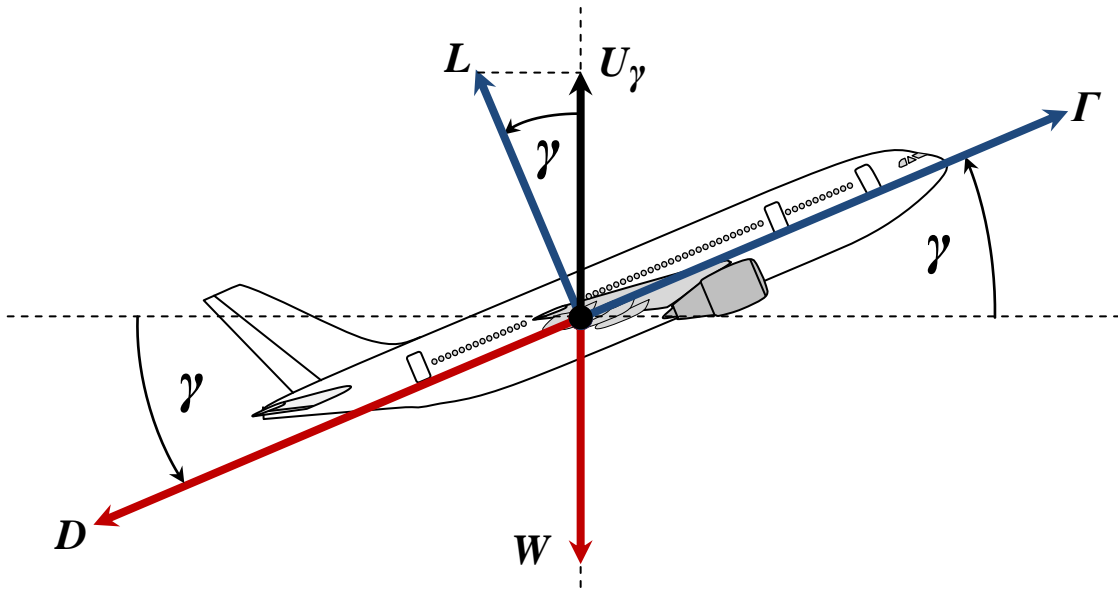


Fig 3.6.4: The balance of forces during ascent

By treating the flight path angle as an independent input to the system, all aircraft dynamics in the vertical plane are essentially bypassed. Vertical motion becomes independent of the inertia of aircraft and can therefore be modelled through the purely kinematic relationships of equation (3.6.2). This is a fairly valid assumption since it turns out that an aircraft performs very little, if any, acceleration or deceleration in the vertical direction whose outcome does not comply with these simple kinematics. This means that for all intents and purposes, the steady-state flight model can be applied across all flight phases. Therefore, equation (3.6.12) can be safely used throughout the entire flight.

Once that an accurate estimate for the lift is available, the lift coefficient C_L can be calculated using a re-arranged version of equation (3.6.5) as follows:

$$C_L = \frac{2L}{\rho S V^2} = \frac{2mg}{\rho S V^2 \cos(\gamma) \cos(\varphi)} \quad (3.6.13)$$

The drag C_D can then be calculated from the coefficient C_L by the following relation:

$$C_D = d_1 + d_2 + d_3 \cdot C_L^2 \quad (3.6.14)$$

Where d_1 , d_2 and d_3 are aircraft-specific BADA coefficients that additionally depend on the aircraft configuration in accordance to the definition given in **Table 3.6.1**

#	Flight Phase	Slats	Flaps	Gear	Spoiler s	d_1	d_2	d_3
0	Take-off	15°	0°	Down	None	$C_{D0,TO}$	$C_{D0,\Delta LDG}$	$C_{D2,TO}$
1	Initial Climb	15°	0°	Up	None	$C_{D0,IC}$	0	$C_{D2,IC}$
2	Cruise	0°	0°	Up	None	$C_{D0,CR}$	0	$C_{D2,CR}$
3	Approach	15°	15°	Up	None	$C_{D0,AP}$	0	$C_{D2,AP}$
4	Landing	30°	40°	Down	None	$C_{D0,LD}$	$C_{D0,SP} + C_{D0,\Delta LDG}$	$C_{D2,LD}$

Table 3.6.1: BADA Drag Coefficient Definitions for Airbus A300-B4-622

The model is further augmented to include first order atmospheric effects in accordance to the International Standard Atmosphere (ISA). In the equations (3.6.5), (3.6.6) and (3.6.13) the air density ρ is a function of temperature, which is in turn a function of altitude by the following equations (3.6.15), (3.6.16) when below the tropopause.

$$\rho = \rho_0 \left[\frac{T}{T_0} \right]^{-\frac{g}{K_T R} - 1}, \quad \left(-\frac{g}{K_T R} - 1 \right) = 4.25583 \quad (3.6.15)$$

$$T = T_0 - K_T h, \quad K_T < 0; \quad h < 0 \quad (3.6.16)$$

When flying above the tropopause, the density is calculated as in equation (3.6.17). For standard ISA conditions, ρ_{trop} , is a constant. However for non ISA conditions, ρ_{trop} , should be calculated using equation (3.6.15) in order to maintain model continuity.

$$\rho = \rho_{trop} e^{-\left(\frac{g}{R \cdot T_{trop}}\right) \cdot (h - H_{trop})} \quad (3.6.17)$$

Finally, we assume that only coordinated turns can be performed. This is an important consideration, which is generally observed by all airliners, in the interest of passenger comfort. This means that during the turn, any change in bank angle will only result in a gradual change in yaw, with no side-slip and no change in vertical velocity. No other quantities are affected and passengers only experience a nearly imperceptible increase in weight. **Figure 3.6.3** shows the balance of forces during a turn. When an aircraft banks, the lift vector L may be resolved into two components:

- A deflection force B , which acts on the Aircraft's mass to change its heading, and
- An upward force U , which should oppose the aircraft's weight W due to gravity.

When the aircraft operates the ailerons to achieve the desired angle of bank, it creates a relative difference in lift between the wings. With no further action, this would result in the aircraft slipping towards the lower wing. Moreover, during the bank only part of the lift is available to counteract gravity. To ensure that the aircraft does not change its altitude, the net lift, L must increase commensurately by a slight increase in the angle of attack. Hence, the elevator needs to be adjusted slightly upwards.

When the aircraft operates the ailerons to achieve the desired angle of bank, it also creates a relative difference in drag between the wings. More precisely: the rising wing ends up slowing down with respect to the dropping wing. This causes undesirable adverse-yaw which may, in-turn, be counteracted by adjusting the rudder angle slightly to force the fuselage to “turn into” the turn and hence realign with the air flow vector.

It is therefore clear that aileron, rudder and elevator must be operated in a coordinated fashion for the aircraft to achieve a coordinated turn. Luckily, modern avionic automation or good pilots ensure that this is always the case. Therefore, coordinated turns can be safely assumed and this considerably simplifies the de-facto kinematics of an airliner.

For zero vertical-change turns, the relationship for the course-deflection force B may be derived simply as shown in equation block (3.6.18):

$$\begin{aligned} B &= L \cdot \sin(\varphi) \\ B &= mg \cdot \sin(\varphi) / (\cos(\gamma) \cdot \cos(\varphi)) \\ B &= mg \cdot \tan(\varphi) / \cos(\gamma) \end{aligned} \quad (3.6.18)$$

3.6.4.3 NEWTONIAN DYNAMICS

The PMM can now take into account basic Newtonian dynamics and from the second law of motion we get the acceleration of the aircraft along the body axis as per equation (3.6.19). The forces acting along this axis are thrust Γ , drag D , and a component of the weight mg , which appears whenever the flight path angle γ is non-zero.

$$\begin{aligned} \dot{V} &= F/m \\ \dot{V} &= \{ \Gamma - D - mg \cdot \sin(\gamma) \} / m \\ \dot{V} &= \left\{ \Gamma - \frac{\rho C_D S}{2} V^2 - mg \cdot \sin(\gamma) \right\} / m \\ \dot{V} &= \Gamma/m - \frac{\rho C_D S}{2m} V^2 - g \cdot \sin(\gamma) \end{aligned} \quad (3.6.19)$$

If an aircraft flies with a velocity, V and experiences a constant course deflection force, B , which is always orthogonal to the direction of motion, the aircraft transcribes a circular path with radius r at an angular velocity ω .

From elementary Newtonian mechanics the following relationship holds:

$$\begin{aligned} B &= m \cdot \omega^2 r = m \cdot V^2 / r \\ \therefore r &= m \cdot V^2 / B \end{aligned} \quad (3.6.20)$$

In addition, using equation (3.6.20) twice, the angular velocity ω may be expressed in terms of the linear velocity, V and the course deflection force, B as follows:

$$\begin{aligned} m \cdot \omega^2 r &= m \cdot V^2 / r \\ \omega^2 &= V^2 / r^2 \\ \omega &= V / r \\ \omega &= V / (m \cdot V^2 / B) \\ \omega &= B / (m \cdot V) \end{aligned} \quad (3.6.21)$$

Finally, after recognizing that the rate of change of heading, $\dot{\sigma} \equiv \omega$ the final relations may be derived using eq.(3.6.21) and (3.6.18) and (3.6.5):

$$\begin{aligned}\omega &= B/(m \cdot V) \\ \dot{\sigma} &= [mg \cdot \tan(\varphi)/\cos(\gamma)]/(m \cdot V) \\ \dot{\sigma} &= \frac{g \cdot \tan(\varphi)}{V \cdot \cos(\gamma)}\end{aligned}\quad (3.6.22)$$

3.6.4.4 COMPLETE 3D DYNAMIC MODEL

The equations (3.6.3), (3.6.19) and (3.6.22) can now be used to extend kinematic model of equation (3.6.2) to produce a complete, but simplified, dynamic model for a fixed-wing aircraft. The inputs to this model are the bank angle φ , the flight path angle, γ and the thrust, Γ in addition to the wind vector, \mathbf{w} The full dynamic model is therefore captured in (3.6.23) and may be solved by iteratively integrating the state vector $\dot{\mathbf{x}}$ over each time step.

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{h} \\ \dot{V} \\ \dot{\sigma} \\ \dot{m} \end{bmatrix} = \begin{bmatrix} V \cdot \cos(\sigma)\cos(\gamma) \\ V \cdot \sin(\sigma)\cos(\gamma) \\ V \cdot \sin(\gamma) \\ \Gamma/m - V^2 \rho S C_D / 2m - g \sin(\gamma) \\ g \cdot \tan(\varphi)/V \cdot \cos(\gamma) \\ -\eta \Gamma \end{bmatrix} + \begin{bmatrix} w_x \\ w_y \\ w_h \\ 0 \\ 0 \\ 0 \end{bmatrix}\quad (3.6.23)$$

This point mass model (PMM) approximation essentially implies that the aircraft's moments of inertia about any axis are ignored and therefore the input attitude $[\varphi, \gamma]$ of the aircraft can be changed instantaneously. In reality there are dedicated nested control systems, whose job is to hide these higher dynamics by rotating the aircraft as quickly as possible in response to the desired reference bank or flight-path angle, as requested by the FMS. This makes the PMM assumptions very acceptable in the ATM world since the time it takes an aircraft to adjust its attitude is negligible compared to the duration of the journey. Of course, this approximation results in a small positional (track) error after each manoeuvre, but this is minimal and can be easily dismissed as an external disturbance to be corrected-for by the FMS outer control loop [3.10].

3.6.4.5 ENERGY SHARE FACTOR (ESF)

It turns out that the flight path angle is an unfeasible way for controlling an aircraft. This is because there is no apparent link between the flight path angle and the actual engine power that is required to achieve that flight path angle. This allows the model to be forced to unrealistic flight path angles that are unattainable in practice. In order to calculate realistic flight path angles, the total power that the engines impart to an aircraft needs to be appropriately partitioned between the gains in potential energy (PE) (gains in altitude) and the gains in kinetic energy (KE) (gains in velocity).

Thus an energy partitioning factor, μ , is defined as the ratio between potential energy (PE) gains and total energy ($PE+KE$) gains [3.11].

$$\mu = \frac{dPE}{dKE + dPE} = \frac{\dot{PE}}{\dot{KE} + \dot{PE}} \quad (3.6.24)$$

For the following, we will ignore that the mass m is also varying with time. This is a reasonable assumption because any changes in m are insignificant compared to changes in h over the time frame of a typical aircraft manoeuvre. Hence, the rate of change in potential energy is determined as follows:

$$\dot{PE} = \frac{d(PE)}{dt} = \frac{d(mgh)}{dt} = mg \frac{dh}{dt} = mg\dot{h} \quad (3.6.25)$$

However, from basic kinematics: $\dot{h} = V \cdot \sin(\gamma)$, and therefore:

$$\dot{PE} = mgV \cdot \sin(\gamma) \quad (3.6.26)$$

The rate of change in kinetic energy is determined as follows:

$$\dot{KE} = \frac{d(KE)}{dt} = \frac{d(\frac{1}{2}mV^2)}{dt} = \frac{1}{2}m \frac{d(V^2)}{dt} = \frac{1}{2}m(2V)\dot{V} = mV\dot{V} \quad (3.6.27)$$

However, from Newton's second law $\dot{V} = F/m$ and the total force acting along the body axis is $F = (\Gamma - D) - mg \cdot \sin(\gamma)$. Combining these with equation (3.6.27) yields:

$$\dot{KE} = mV\dot{V} = mV \frac{F}{m} = V(\Gamma - D) - Vmg \cdot \sin(\gamma) \quad (3.6.28)$$

From the definition (3.6.24), the energy share factor μ , can now be determined as follows:

$$\mu = \frac{\dot{PE}}{\dot{KE} + \dot{PE}} = \frac{mgV \cdot \sin(\gamma)}{V(\Gamma - D) - Vmg \cdot \sin(\gamma) + mgV \cdot \sin(\gamma)} = \frac{mg \cdot \sin(\gamma)}{(\Gamma - D)} \quad (3.6.29)$$

Equation (3.6.29) can be rearranged to obtain an expression for $\sin(\gamma)$, as shown in (3.6.30)

$$\sin(\gamma) = \frac{(\Gamma - D)}{mg} \mu \quad (3.6.30)$$

and using the well known trigonometric identity $\cos(\gamma) = \pm\sqrt{1 - \sin^2(\gamma)}$ we also obtain

$$\cos(\gamma) = \sqrt{1 - \left(\frac{(\Gamma - D)}{mg} \mu\right)^2} = \sqrt{1 - \left(\frac{(\Gamma - 0.5V^2\rho SC_D)}{mg} \mu\right)^2} \quad (3.6.31)$$

where only positive values of the square-root need be considered.

Finally, the flight path angle γ can be eliminated from the state equation (3.6.23) which can now be re-expressed in terms of the energy share factor μ as follows:

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{h} \\ \dot{V} \\ \dot{\sigma} \\ \dot{m} \end{bmatrix} = \begin{bmatrix} V \cdot \cos(\sigma) \cdot \cos(\gamma) \\ V \cdot \sin(\sigma) \cdot \cos(\gamma) \\ V \cdot (\Gamma - 0.5V^2 \rho S C_D) \mu / mg \\ (\Gamma - 0.5V^2 \rho S C_D)(1 - \mu) / m \\ g \cdot \tan(\varphi) / \{V \cdot \cos(\gamma)\} \\ -\eta\Gamma \end{bmatrix} + \begin{bmatrix} w_x \\ w_y \\ w_h \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (3.6.32)$$

3.6.4.6 INTERPRETATION OF THE ENERGY SHARE FACTOR

The surplus engine power or “New Energy” (NE) is the excess power that is available to the aircraft after deducting the effects of drag. Similarly, the power deficit that results when the drag power exceeds the thrust power, is termed “Lost Energy” (LE). The ESF indicates the proportion of NE that goes into either increases in velocity or altitude. It can take any value, however, sensible values range from -200% to 200%. The BADA standard limits the ESF to between 30% and 170%.

When the ESF is precisely 100%, the KE does not change. All the engine power output is used up as work against drag and gravity. All the NE is transferred to an increase in PE, and therefore altitude h . Hence, the vertical velocity \dot{h} increases in proportion to the μ -term in the third row of equation (3.6.32) and the acceleration \dot{V} drops to zero because of the $(1 - \mu)$ term in the fourth row of equation (3.6.32).

When the ESF is precisely 0%, the PE does not change. All the engine power output is now used up as work against drag and the aircraft’s inertia. All the NE is transferred to KE, and therefore velocity V . Hence, the vertical velocity \dot{h} drops to zero in relation to the μ -term in the third row of equation (3.6.32) and the acceleration \dot{V} increases in tandem with the $(1 - \mu)$ term in the fourth row of equation (3.6.32).

When the ESF lies between 0% and 100%, the NE is shared accordingly between PE and KE. When the ESF exceeds 100%, it implies that there is a net conversion from KE to PE. This occurs at high positive flight path angles which result in a gain in altitude at the expense of velocity. Likewise, a negative ESF (less than 0%), implies a net conversion from PE to KE. This is expected when the flight path angle points sharply downwards, resulting in an increase in velocity at the expense of altitude.

The ESF is not a perfect representation of truth however. In this model, all exchanges between PE and KE are modulated by the power surplus (NE) or deficit (LE). This is not entirely true in practice, since an aircraft can still exchange PE for KE or vice versa when the thrust exactly matches the drag. However, catering for this eventuality poses the other difficulty of deciding on the rate of energy transfer. This should clearly be in the same ball park figure as the aircraft’s power-train rating. Hence, this is the standard adopted by BADA and a summary of all the possible combinations of thrust and drag and ESF can be found in **Table 3.6.2** below.

	$\mu < 0$	$\mu = 0$	$0 < \mu < 1$	$\mu = 1$	$\mu > 1$
$\Gamma > D$	$\dot{h} = -ve$ $\dot{V} = +ve$ $NE + PE \Rightarrow KE$	$\dot{h} = 0$ $\dot{V} = +ve$ $NE \Rightarrow KE$	$\dot{h} = +ve$ $\dot{V} = +ve$ $NE \Rightarrow KE + PE$	$\dot{h} = +ve$ $\dot{V} = 0$ $NE \Rightarrow PE$	$\dot{h} = +ve$ $\dot{V} = -ve$ $KE + NE \Rightarrow PE$
$\Gamma = D$	$\dot{h} = 0$ $\dot{V} = 0$	$\dot{h} = 0$ $\dot{V} = 0$	$\dot{h} = 0$ $\dot{V} = 0$	$\dot{h} = 0$ $\dot{V} = 0$	$\dot{h} = 0$ $\dot{V} = 0$
$\Gamma < D$	$\dot{h} = +ve$ $\dot{V} = -ve$ $KE \Rightarrow LE + PE$	$\dot{h} = 0$ $\dot{V} = -ve$ $KE \Rightarrow LE$	$\dot{h} = -ve$ $\dot{V} = -ve$ $KE + PE \Rightarrow LE$	$\dot{h} = -ve$ $\dot{V} = 0$ $PE \Rightarrow LE$	$\dot{h} = -ve$ $\dot{V} = +ve$ $PE \Rightarrow KE + LE$

Tab 3.6.2: Interpretation of the BADA Energy share factor

3.6.4.7 POWER DERATING COEFFICIENT

Aircraft seldom use all the available power during take-off or climb, unless it is absolutely necessary. This is done to reduce the wear-and-tear on the engines and hence prolong their lifespan and reduce maintenance costs. This is usually done by assuming an ambient temperature that is higher than it actually is. This engine derating needs to be taken into account in the model by introducing a power derating coefficient R_{CP} , which models this reduced climb power. The state equation (3.6.32) is again modified to include this coefficient as follows:

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{h} \\ \dot{V} \\ \dot{\sigma} \\ \dot{m} \end{bmatrix} = \begin{bmatrix} V \cdot \cos(\sigma) \cdot \cos(\gamma) \\ V \cdot \sin(\sigma) \cdot \cos(\gamma) \\ V \cdot (\Gamma - 0.5V^2 \rho S C_D) (R_{CP}) \mu / mg \\ (\Gamma - 0.5V^2 \rho S C_D) (1 - \mu) / m \\ g \cdot \tan(\varphi) / \{V \cdot \cos(\gamma)\} \\ -\eta \Gamma \end{bmatrix} + \begin{bmatrix} w_x \\ w_y \\ w_h \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (3.6.33)$$

3.6.4.8 AIRCRAFT MASS, PAYLOAD AND FUEL

An aircraft's maximum mass M_T consists of the empty mass M_E , the payload mass M_P and the fuel mass M_F . It is also fair to assume that the fuel mass required aboard a flight is proportional to the total mass of the aircraft and journey range R : $M_F = k_F \cdot M_T \cdot R$

$$\begin{aligned} M_T &= M_E + M_P + M_F \\ \text{Thus: } M_T &= M_E + M_P + k_F \cdot M_T \cdot R \end{aligned} \quad (3.6.34)$$

Using the catalogue range R_G , the constant k_F for any aircraft may be estimated as follows:

$$k_F = (M_T - M_E - M_P) / (R_G \cdot M_T) \quad (3.6.35)$$

The minimum fuel mass required to complete a flight of length R_L may then be calculated:

$$\begin{aligned} M_F / [k_F \cdot R_L] &= M_E + M_P + M_F \\ M_F ([k_F \cdot R_L]^{-1} - 1) &= M_E + M_P \\ M_F &= (M_E + M_P) / ([k_F \cdot R_L]^{-1} - 1) \end{aligned} \quad (3.6.36)$$

3.6.5 4D Point Mass Model for a Fixed Wing Aircraft

Path Planning is the process of finding an obstacle-free, feasible path through N dimensional space between any two points. Optimal path planning performs the same task but with the added objective of minimising some predefined objective function. Path planning defines no restrictions on the velocity or time taken for an agent to traverse the path. The non-linear state space model described by equation (3.6.23) is useful for 3 dimensional (3D) path planning in 3D space. However, the state vector \mathbf{x} is an incomplete representation and cannot be used alone to determine whether two aircraft with states \mathbf{x}_1 and \mathbf{x}_2 will collide in an ATM problem. That two aircraft paths clearly intersect in 3D space is not a good-enough reason to assume that the aircraft will collide.

Trajectory Synthesis, on the other hand, introduces the time element and hence augments the problem to an $(N+1)$ dimensional space. Hence, for 3D space we get 4D trajectories. Optimal trajectory synthesis again performs the same task but with the added objective of minimising some predefined value function, which may now involve time. A 4D trajectory defines the location of an agent in 3D space at any time instant t . The state vector is likewise augmented to include the time element and we may now rewrite equation (3.6.23) as shown below in equation (3.6.37). It now becomes a simple matter to determine whether two aircraft with states \mathbf{x}_3 and \mathbf{x}_4 are occupying the same point in space at the same instant in time. Since the derivative of time with respect to itself is always unity, the state equation is simply augmented with a constant “1”.

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \\ \dot{x}_7 \end{bmatrix} \equiv \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{h} \\ \dot{t} \\ \dot{V} \\ \dot{\sigma} \\ \dot{m} \end{bmatrix} = \begin{bmatrix} V \cdot \cos(\sigma) \cdot \cos(\gamma) \\ V \cdot \sin(\sigma) \cdot \cos(\gamma) \\ V \cdot (\Gamma - 0.5V^2\rho SC_D)(R_{CP})\mu/mg \\ 1 \\ (\Gamma - 0.5V^2\rho SC_D)(1 - \mu)/m \\ g \cdot \tan(\varphi)/\{V \cdot \cos(\gamma)\} \\ -\eta\Gamma \end{bmatrix} + \begin{bmatrix} w_x \\ w_y \\ w_h \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (3.6.37)$$

The element \dot{x}_4 is integrated with all the other state variables in $\dot{\mathbf{x}}$ and as the integration proceeds the Element x_4 in \mathbf{x} becomes the running time parameter, which essentially time-stamps each spatial coordinate of the agent. Now by employing the vector mapping shown in equation (3.6.38), it may now be convenient converting equation (3.6.37) to the mathematical shorthand shown in equation (3.6.39)

$$\text{Let: } \mathbf{u} = \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \end{Bmatrix} \equiv \begin{Bmatrix} \varphi \\ \mu \\ \Gamma \end{Bmatrix}, \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} \equiv \begin{bmatrix} x \\ y \\ h \\ t \\ V \\ \sigma \\ m \end{bmatrix}, \quad \mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \\ w_5 \\ w_6 \\ w_7 \end{bmatrix} \equiv \begin{bmatrix} w_x \\ w_y \\ w_h \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (3.6.38)$$

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) + \mathbf{w}(t) \quad (3.6.39)$$

The input “vector” \mathbf{u} is not a vector in the strict sense of the word. It is rather a collection of input variables. This is because the non-linear form of the system state vector function $\mathbf{f}(\bullet)$ precludes the possibility of isolating the input variables into a separate vector. Equation (3.6.39) should be interpreted as follows: The vector function $\mathbf{f}(\bullet)$ depends on the current state vector \mathbf{x} ; the current input vector \mathbf{u} and the current wind (disturbance) vector \mathbf{w} to produce the derivative state vector $\dot{\mathbf{x}}$ which may in turn be integrated to generate a future state vector \mathbf{x} . Note that all vectors are functions of time, but for the sake of brevity, this is often taken for granted and hence equation (3.6.39) is usually expressed in the form shown in equation (3.6.40):

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) + \mathbf{w} \quad (3.6.40)$$

3.6.6 The Virtual Time Parameter (τ)

For reasons that will become clearer later ([3.12],[3.13] and [3.14]), it is desirable to recast the 4D non-linear state space model described by equation (3.6.40), in terms of a normalised virtual time parameter ‘ τ ’ which can only take values from the interval $[0, 1]$. [3.12], [3.13], [3.14].

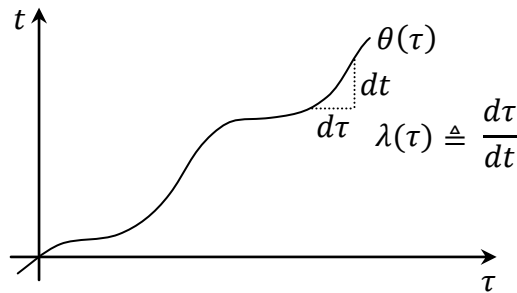


Fig 3.6.5: A virtual time parameter ‘ τ ’

This parameter is defined by the following relation (3.6.41), where $\theta(\bullet)$ is a monotonic increasing function as shown above in **Figure 3.6.5**.

$$t = \theta(\tau) \text{ where } \frac{d\theta(\tau)}{d\tau} > 0, \quad \forall \tau \in \{\mathbb{R}_+ \cap [0, 1]\} \quad (3.6.41)$$

Now
let:

$$\lambda \equiv \lambda(\tau) \triangleq \frac{d\tau}{dt} \quad (3.6.42)$$

For the sake of clarity, the following shall adopt the convention that:

$$\dot{\bullet} \equiv \frac{d\bullet}{dt} \quad \text{and} \quad \bullet' \equiv \frac{d\bullet}{d\tau}$$

Hence:

$$\begin{aligned} \mathbf{x}' &\equiv \frac{d\mathbf{x}}{d\tau} = \frac{dt}{d\tau} \cdot \frac{d\mathbf{x}}{dt} = \frac{1}{\lambda} \cdot \frac{d\mathbf{x}}{dt} = \frac{\dot{\mathbf{x}}}{\lambda} \\ &\Rightarrow \dot{\mathbf{x}} = \lambda \mathbf{x}' \end{aligned} \quad (3.6.43)$$

We may now reformulate the differential equation (3.6.40) in terms of the new virtual time parameter ‘ τ ’ as follows in equation (3.6.44). This operation scales the time parameter from t to τ but preserves the numerical consistency of the differential equation by scaling it as necessary using the differential coefficient λ .

$$\begin{aligned} \mathbf{x}'(\tau) &= \frac{\dot{\mathbf{x}}(t)}{\lambda} = \frac{1}{\lambda} \mathbf{f}(\mathbf{x}(\tau), \mathbf{u}(\tau), \mathbf{w}(\tau)) \\ \mathbf{x}' &= \frac{1}{\lambda} \mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{w}) \end{aligned} \tag{3.6.44}$$

ie:

By using equation (3.6.43), the chain rule and the multiplication-rule iteratively, it can be shown that the relationship between \mathbf{x}' and $\dot{\mathbf{x}}$ extends to higher derivatives as will be shown next. The derivation (3.6.45) shows the method for producing the 2nd derivative.

$$\begin{aligned} \ddot{\mathbf{x}} &= \frac{d\dot{\mathbf{x}}}{dt} = \frac{d(\lambda\mathbf{x}')}{dt} = \lambda \frac{d(\lambda\mathbf{x}')}{d\tau} \\ \ddot{\mathbf{x}} &= \lambda \frac{d\lambda}{d\tau} \cdot \mathbf{x}' + \lambda^2 \frac{d\mathbf{x}'}{d\tau} \end{aligned} \tag{3.6.45}$$

ie:

$$\boxed{\ddot{\mathbf{x}} = \lambda\lambda'\mathbf{x}' + \lambda^2\mathbf{x}''}$$

The derivation (3.6.46) shows the method for producing the 3rd derivative. This process may be continued indefinitely, each time resulting in a polynomial relationship between the derivatives of $\mathbf{x}(t)$ and $\mathbf{x}(\tau)$.

$$\begin{aligned} \ddot{\mathbf{x}} &= \frac{d\dot{\mathbf{x}}}{dt} = \frac{d(\lambda\lambda'\mathbf{x}' + \lambda^2\mathbf{x}'')}{dt} = \lambda \frac{d(\lambda\lambda'\mathbf{x}' + \lambda^2\mathbf{x}'')}{d\tau} \\ \ddot{\mathbf{x}} &= \lambda \left(\frac{d(\lambda\lambda'\mathbf{x}')}{d\tau} \right) + \lambda \left(\frac{d(\lambda^2\mathbf{x}'')}{d\tau} \right) \\ \ddot{\mathbf{x}} &= \lambda \left(\frac{d(\lambda\lambda'\mathbf{x}')}{d\tau} \right) + \lambda \left(\frac{d(\lambda^2)}{d\tau} \mathbf{x}'' + \lambda^2 \frac{d(\mathbf{x}'')}{d\tau} \right) \\ \ddot{\mathbf{x}} &= \lambda \left(\frac{d(\lambda\lambda')}{d\tau} \mathbf{x}' + \lambda\lambda' \frac{d(\mathbf{x}')}{d\tau} \right) + \lambda(\lambda'^2\mathbf{x}'' + \lambda^2\mathbf{x}''') \\ \ddot{\mathbf{x}} &= \lambda \left(\left(\frac{d\lambda}{d\tau} \lambda' + \lambda \frac{d\lambda'}{d\tau} \right) \mathbf{x}' + \lambda\lambda'\mathbf{x}'' \right) + (\lambda\lambda'^2\mathbf{x}'' + \lambda^3\mathbf{x}''') \\ \ddot{\mathbf{x}} &= \lambda \left((\lambda'^2 + \lambda\lambda'')\mathbf{x}' + \lambda\lambda'\mathbf{x}'' \right) + (\lambda\lambda'^2\mathbf{x}'' + \lambda^3\mathbf{x}''') \\ \ddot{\mathbf{x}} &= (\lambda\lambda'^2\mathbf{x}' + \lambda^2\lambda''\mathbf{x}' + \lambda^2\lambda'\mathbf{x}'') + (\lambda\lambda'^2\mathbf{x}'' + \lambda^3\mathbf{x}''') \end{aligned} \tag{3.6.46}$$

ie:

$$\boxed{\ddot{\mathbf{x}} = (\lambda\lambda'^2 + \lambda^2\lambda'')\mathbf{x}' + (\lambda^2\lambda' + \lambda\lambda'^2)\mathbf{x}'' + \lambda^3\mathbf{x}'''}$$

3.6.7 A Discrete Time Model

The problem of continuous time trajectory optimisation can be tackled using optimal control techniques. The Pontryagin's Minimum Principle (PMP) is invoked, which states that a necessary (but insufficient) condition for optimality can be achieved by forming and minimising the Hamiltonian Function in a higher dimensional space. Analytical formation of the Hamiltonian is unfortunately only possible in relatively few special cases. A more general optimisation method is therefore preferred.

A better solution is to employ discrete-time numerical techniques. These lend themselves to the use of a computer to rapidly scan the problem search space in an organised fashion. By discretizing the original problem, the non-linear optimal control problem is converted into a structured Non-Linear mathematical Programming (NLP) problem. This can then be tackled using one of several well-established NLP techniques. If a sufficiently fine discretization is used, the outcome of the NLP optimisation should then converge to the discretized version of the solution to the original continuous-time problem. To this end, the aircraft model needs to be discretized and this will be performed as follows.

First the virtual time parameter τ will be normalised and be defined in the closed interval $[0, 1]$. The real time parameter is also restricted to the closed interval $[t_i, t_f]$.

$$\begin{aligned} \tau \in \{\mathbb{R}_+ \cap [0, 1]\} \quad \text{and} \quad t \in \{\mathbb{R}_+ \cap [t_i, t_f]\} \\ \text{where } t_i \in \mathbb{R}_+; t_f \in \mathbb{R}_+; t_f > t_i \end{aligned} \quad (3.6.47)$$

Furthermore, τ will be discretized into P equidistant virtual time stages of duration L

$$\text{where:} \quad L = \Delta\tau \equiv (\tau_{k+1} - \tau_k) \quad \text{for } k \in \mathbb{Z}_+; k \leq P$$

$$\text{such that:} \quad \sum_{k=1}^P L = 1 \quad \Rightarrow \quad P \cdot L = 1 \quad \text{where } k, P \in \mathbb{Z}_+; L \in \mathbb{R}_+ \quad (3.6.48)$$

Additionally, the real time parameter t will also be discretized into P time stages of a variable duration $\Lambda(k)$. The sequence $\Lambda(k) \in \mathbb{R}_+$ and is defined for $k \in \{1, 2, 3 \dots P\}$ and is subject to these further conditions:

$$\begin{aligned} \sum_{k=1}^P \Lambda(k) &= (t_f - t_i) \\ \text{where:} \quad \Rightarrow (t_{k+1} - t_k) &= \Lambda(k) \cdot P(\tau_{k+1} - \tau_k) \\ &\Rightarrow \Delta t_k = \Lambda(k) \cdot P\Delta\tau_k \\ &\Rightarrow \frac{\Delta t_k}{\Delta\tau_k} = P \cdot \Lambda(k) \end{aligned} \quad (3.6.49)$$

It should therefore be clear that the (continuous) time ratio parameter $\lambda(\tau)$ is related to its discrete version $\Lambda(k)$ by the following relation:

$$P \cdot \Lambda(k) = \frac{\Delta t_k}{\Delta \tau_k} \approx \frac{dt}{d\tau} = \frac{1}{\lambda(\tau)} \Big|_{\tau = kL} \quad (3.6.50)$$

Finally, the discrete version of differential equation (3.6.44) can now be expressed in terms of $\Lambda(k)$, P , and kL .

$$\mathbf{x}'(kL) = P \cdot \Lambda(k) \cdot \mathbf{f}(\mathbf{x}(kL), \mathbf{u}(kL), \mathbf{w}(kL)) \quad (3.6.51)$$

For the sake of brevity the virtual time sampling interval L is omitted so that the discrete time equation becomes:

$$\mathbf{x}'(k) = P \cdot \Lambda(k) \cdot \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k), \mathbf{w}(k)) \quad (3.6.52)$$

The discrete-time, non-linear, time-invariant, differential equation now needs to be directly integrated over each virtual time sampling interval L in order to evaluate the next time-interval state-vector $\mathbf{x}'(k+1)$ from the current input $\mathbf{u}(k)$, the current disturbance $\mathbf{w}(k)$, and the current state vector $\mathbf{x}(k)$. This may be performed as follows by drawing some parallels with quasi-linearization theory [3.15]:

$$\begin{aligned} \mathbf{x}([k+1]L) &= \mathbf{x}(kL) + \int_{kL}^{(k+1)L} \mathbf{x}'(\tau) d\tau \\ \mathbf{x}([k+1]L) &= \mathbf{x}(kL) + \int_{kL}^{(k+1)L} P \cdot \Lambda(k) \cdot \mathbf{f}(\mathbf{x}(\tau), \mathbf{u}(\tau), \mathbf{w}(\tau)) d\tau \\ \mathbf{x}([k+1]L) &= \mathbf{x}(kL) + P \cdot \Lambda(k) \cdot \int_{kL}^{(k+1)L} \mathbf{f}(\mathbf{x}(\tau), \mathbf{u}(\tau), \mathbf{w}(\tau)) d\tau \end{aligned} \quad (3.6.53)$$

For small L equation (3.6.53) reduces to the difference equation (3.6.54), which can now be solved iteratively for all k , where $k = 1, 2, 3 \dots P$, which essentially covers the entire normalized time interval $[0 \leq \tau \leq 1]$ and hence also $[0 \leq t_i \leq t \leq t_f \leq \infty]$:

$$\begin{aligned} \mathbf{x}([k+1]L) &\approx \mathbf{x}(kL) + P \cdot \Lambda(k) \cdot L \cdot [\mathbf{f}(\mathbf{x}(kL), \mathbf{u}(kL), \mathbf{w}(kL))] \\ \mathbf{x}(k+1) &\approx \mathbf{x}(k) + \Lambda(k) \cdot [\mathbf{f}(\mathbf{x}(k), \mathbf{u}(k), \mathbf{w}(k))] \end{aligned} \quad (3.6.54)$$

The time sequence $\Lambda(k)$ can now be treated just like any another control input variable in equation (3.6.54). This control input needs to be determined when solving the optimal control problem, which results in the optimal state trajectory $\mathbf{x}^*(k)$. This provision introduces a new degree of freedom, which allows the trajectory optimizer to change the (real) time-step resolution along the state trajectory in order utilize all sample points in an efficient manner.

3.6.8 Discrete Aspects of a Fixed-Wing Aircraft Model

The aircraft model is further enriched by the fact that the model parameters are not constant. They are rather functions of various modes that the aircraft can take during normal operation. An aircraft in take-off mode has significantly different aerodynamic and performance characteristics than the same aircraft in cruise mode or in descent. These changes can be captured by a state machine, which defines the permissible mode transitions and the associated model parameters. **Figure 3.6.6** shows all the allowable state changes for a typical airliner.

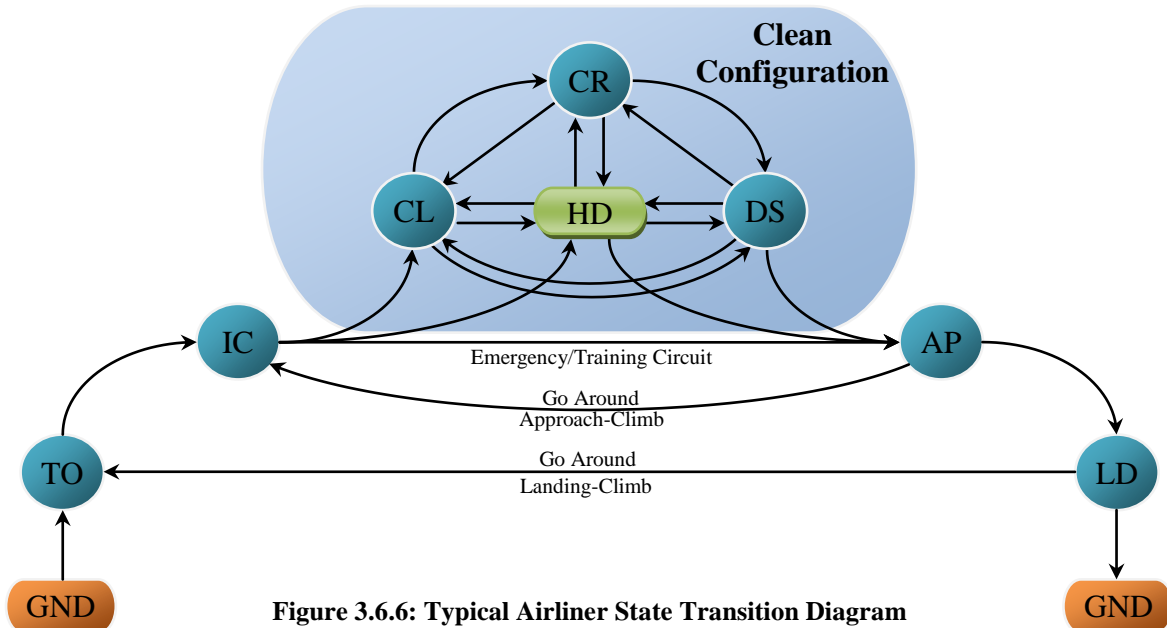


Figure 3.6.6: Typical Airliner State Transition Diagram

3.6.9 Model Validation

The full model described in the forgoing was coded using Matlab. Tables of aircraft parameters, were extracted from the BADA manual in [3.6], and accompanying data files. For this case two aircraft were modelled: An Airbus A300-B4-622 with Pratt & Whitney PW4158 engines, and a Boeing 767-300ER with Pratt & Whitney PW4060 engines. These include aircraft specification and performance parameters like mass, maximum payload, service ceiling, C_L , C_D , fuel consumption coefficients etc... This data need not be entirely accurate, so long as it is in representative of typical aircraft performance. This was then qualitatively validated with a few test flights that demonstrate typical behaviour. **Figure 3.6.7** shows the model's behaviour during take-off, initial climb and climb. The trajectory and ground track are shown. Other aircraft types can be easily modelled by swapping between BADA definition tables.

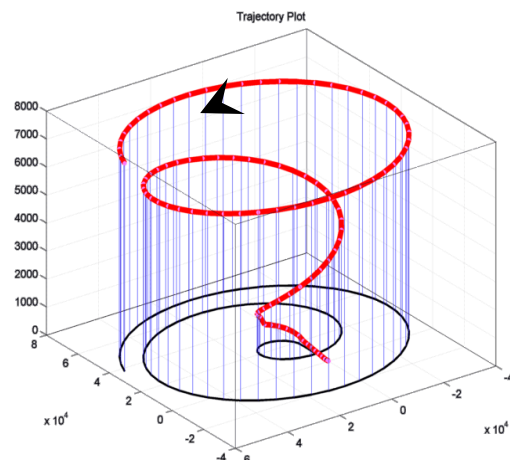


Figure 3.6.7: Take-off and Initial Climb

A further scenario is shown in **Figure 3.6.8**, where the aircraft takes off and climbs linearly and after reaching cruising altitude, proceeds linearly to the destination airport over, which it enters a holding pattern prior to descent. All such data was obtained in open loop, by simply providing the model with appropriate input control parameters.

Figure 3.6.9 shows a similar scenario from another perspective for a 4 hour flight. The Airbus A300-B4-622 is being used here. This time, holding patterns were engaged both during ascent and just before descent. The corresponding flight data is shown in **Figure 3.6.10** below. The horizontal axes of each chart are labelled in seconds while all other parameters are expressed in SI units. The data is clearly qualitatively valid and although it is hard to compare with real Airbus 300 data, as this is hard to obtain due to commercial sensitivity, the data appears to be quantitatively valid. The aircraft also correctly transitions between states shown earlier in the state diagram of **Figure 3.6.6**, and the state changes are plotted in the AeroConfig chart (below).

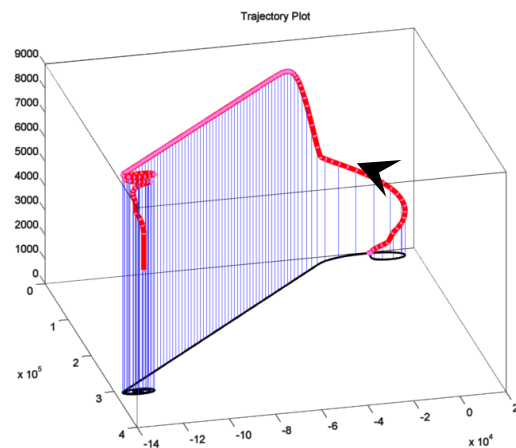


Figure 3.6.8: Take-off, Climb, and Cruise

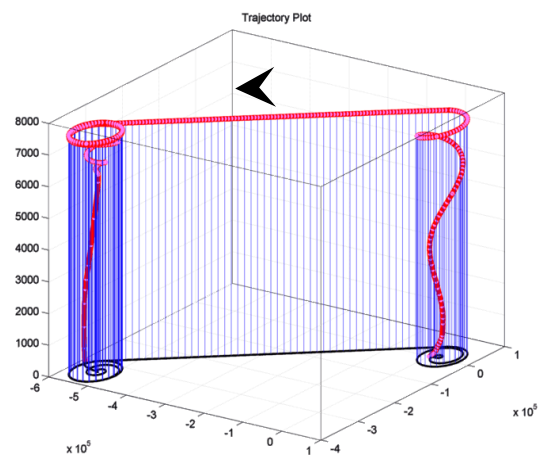


Figure 3.6.9: With Holding Patterns

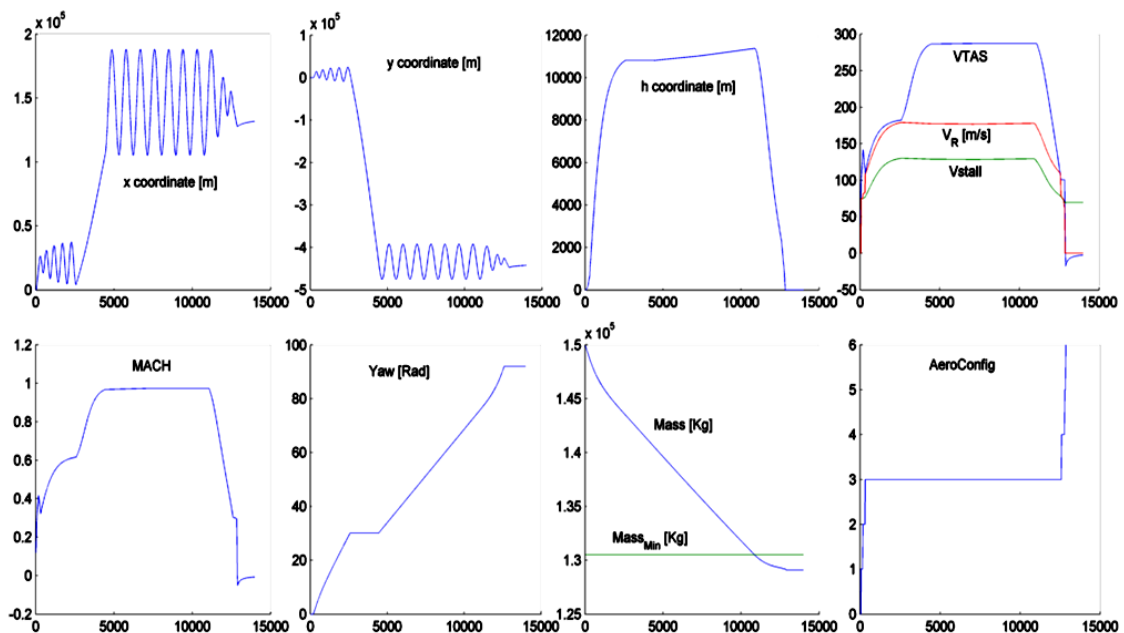


Figure 3.6.10: Flight Data:- Take-off, Initial Climb, Hold, Cruise, Hold, Descend, Approach, Land

3.7 AIRSPACE MODELLING FOR CATM

In the hands of man, airspace is not one continuum. It is in fact divided into a number of zones, restrictions and classifications. One of the major limitations for travelling direct shortest-path routes relates to the presence of such classified and restricted airspaces. These need to be accurately modelled to be taken into account by trajectory optimisers. Some of these classifications may well be rendered obsolete as modern avionics and ATM paradigms begin to take hold of the system. Others will persist due to safety concerns.

3.7.1 Classified Airspace

The ICAO classifies airspace into seven categories: Class-A through to Class-G [3.16]. These define the flying rules (VFR or IFR) and the minimum level of service that will be provided by the relevant Air Navigation Service Providers (ANSPs) and Air Traffic Service Providers (ATSPs). **Table 3.7.1** provides a summary of these definitions. Very often these airspaces are bracketed between specific flight levels, however, the precise definitions tend to vary from one country to the next.

For instance, in the UK [3.17], save for some exceptions, Class A extends between FL55 through to FL245 and is considered controlled airspace. It is limited to IFR only and ATC takes on the full responsibility of separating aircraft flying in this volume. At the other end of the scale, Class G is considered uncontrolled airspace. In the UK it extends from FL660, upwards and flights in this space are entirely unregulated. In between, there are varying degrees of restriction and service which vary geographically, particularly in the vicinity of the Terminal Areas (TMA) surrounding major airports.

Class	Flight Rules	Separation	Service Provided	Speed Limit	Comms. Required	ATC Clearance
A	IFR only	All Aircraft	ATC	Not applicable	2-way	Yes
B	IFR	All Aircraft	ATC	Not applicable	2-way	Yes
	VFR	All Aircraft	ATC	Not applicable	2-way	Yes
C	IFR	IFR from IFR IFR from VFR	ATC	250 kts < 10kft	2-way	Yes
	VFR	VFR from IFR	ATC, Traffic Info.	250 kts < 10kft	2-way	Yes
D	IFR	IFR from IFR	ATC, VFR Traffic Info.	250 kts < 10kft	2-way	Yes
	VFR	nil	IFR/VFR and VFR/VFR Traffic Information	250 kts < 10kft	2-way	Yes
E	IFR	IFR from IFR	ATC, VFR Traffic Info. AFAP	250 kts < 10kft	2-way	Yes
	VFR	nil	Traffic Information AFAP	250 kts < 10kft	No	No
F	IFR	IFR from IFR AFAP	Air Traffic Advisory, FIS	250 kts < 10kft	2-way	No
	VFR	nil	FIS	250 kts < 10kft	None	No
G	IFR	nil	FIS	250 kts < 10kft	2-way	No
	VFR	nil	FIS	250 kts < 10kft	None	No

FIS: *Flight Information Service*; AFAP: *As Far As Practicable*; IFR/VFR: *Instrument/Visual Flying Rules*

Table 3.7.1: ICAO Classification of Airspace [3.17], [3.23]

3.7.2 Special Zone Constraints

In addition to classification, the airspace is peppered with special zones and areas, which have been earmarked for specific uses. For example, in Europe there are thousands of restricted zones and they are independently defined by each country to safeguard sensitive military installations, or even state prisons or to protect aircraft from natural or military hazards (Fig. 3.7.1).

Aerodrome traffic zones whether civilian (ATZ) or military (MATZ), are cylindrical exclusion zones designed protect any aircraft climbing and descending into these airfields from other crossing traffic. These are often surrounded by an “inverted wedding cake” TMA protection zone. Helicopters Main Routes (HMR) and Helicopters Protected Zones (HPZ) warn fixed-wing aircraft of the potential presence of helicopters performing erratic manoeuvres and are therefore no-go zones. Military training areas (MTA) are used only temporarily for intensive military training. Such areas may follow weekly time tables or may be reserved ad-hoc by issuing appropriate NOTAMs (Notification to Airmen). Air-to-Air Refuelling Areas (ARA) and other military flight combat training in relegated to Aerial Tactical Areas (ATA). These may also be activated by NOTAMs.

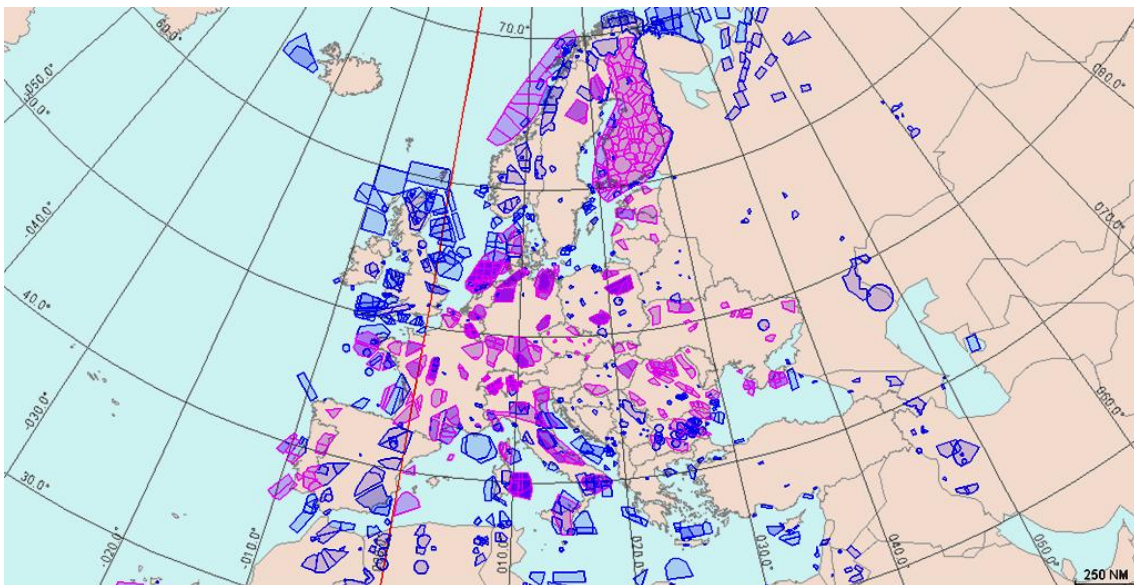


Figure 3.7.1: Traffic Restrictions in European Airspace [Generated with SkyView v.2.5.1]

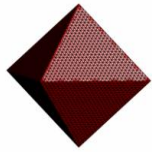
Some areas may be reserved for airborne surveillance aircraft, which tend to follow specific holding patterns for extended periods. Danger Areas are not always active and they are often due to the presence of military firing ranges. These often have a characteristic wedge shape with the tip lying over the gunnery equipment. Active conflict zones were historically assumed to be safe for civil aviation travelling at cruising altitude. However, following the recent flight disaster over, East Ukraine, where a fully laden Malaysian Airlines flight M17 was shot down, these have also been added to the list of danger zones. Disaster relief and emergency operations may also warrant the creation of temporary danger areas. Still more areas may be permanently prohibited due to specific industrial installations, nuclear power stations, high security prisons, missile silo ranges, sky scrapers and anything that warrants protection from

potential accidental or malicious collisions. Unmanned Aerial Vehicles (UAVs) also warrant their own patch of segregated airspace to which they are confined until they can convince the regulators that they can guarantee a high level of safety and reliability.

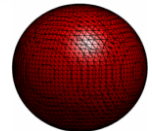
In the end, as far as the SPO is concerned these special airspaces are easily handled once their presence is included in a database. In the case of temporarily restricted airspaces that may be activated by NOTAMs, time-tables may be used to predict their pattern of availability. It is assumed that any changes in status cannot be enforced suddenly without an acceptable lead-time. However, in any case, the SPO will need to re-adapt to the new status if any chunk of airspace becomes unavailable. Given the large number of no-fly zones defined in any airspace, certain simplifications have to be applied in order to mitigate the performance impact imposed by such path constraints.

3.7.3 Obstacle Modelling

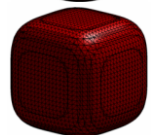
In order to produce meaningful CATM simulation results and flyable trajectories, these no-fly zone classifications need to be identified and modelled appropriately. For a simulator to function, every candidate trajectory waypoint needs to be evaluated by the optimiser in respect of these spatial restrictions, several of which may have very irregular shapes. For this to be possible a well behaved search space needs to be constructed taking into account these zones, and without creating sharp discontinuities that can lead to numerical problems later.



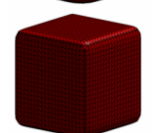
$$\wp \geq H \left(3 - \left| \frac{x - p_x}{s_x} \right| - \left| \frac{y - p_y}{s_y} \right| - \left| \frac{z - p_z}{s_z} \right| \right) \quad (3.7.1)$$



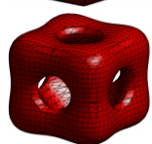
$$\wp \geq H \left(\left| \frac{x - p_x}{s_x} \right|^2 + \left| \frac{y - p_y}{s_y} \right|^2 + \left| \frac{z - p_z}{s_z} \right|^2 \right) \quad (3.7.2)$$



$$\wp \geq H \left(\left| \frac{x - p_x}{s_x} \right|^4 + \left| \frac{y - p_y}{s_y} \right|^4 + \left| \frac{z - p_z}{s_z} \right|^4 \right) \quad (3.7.3)$$



$$\wp \geq H \left(\left| \frac{x - p_x}{s_x} \right|^{20} + \left| \frac{y - p_y}{s_y} \right|^{20} + \left| \frac{z - p_z}{s_z} \right|^{20} \right) \quad (3.7.4)$$



$$\wp \geq H \left[\left| \frac{x}{s_x} \right|^2 + \left| \frac{y}{s_y} \right|^2 + \left| \frac{z}{s_z} \right|^2 \right] \cdot \exp \left(- \left| \frac{x}{s_x} \right|^2 - \left| \frac{y}{s_y} \right|^2 - \left| \frac{z}{s_z} \right|^2 \right) \quad (3.7.5)$$

where:

$$\begin{aligned} x, y, z & \text{ are the coordinates of the spatial trajectory point being evaluated} \\ s_x, s_y, s_z & \text{ are the obstacle scaling coefficients in each dimension} \\ p_x, p_y, p_z & \text{ are the obstacle positioning coefficients in each dimension} \\ \wp & \text{ is the chosen iso-potential boundary of the object (used for} \\ & \text{drawing)} \end{aligned} \quad (3.7.6)$$

The methodology adopted in this project was to model each obstacle as a smooth “*electrostatic potential*” function, with iso-potential contours that conservatively mimic the shape of the obstacle. A number of obstacle *primitives* were created by extending a simple and elegant method described by Cowling et al. [3.19]. The primitives allow substantial scaling and are quite adaptable to regular shapes such as spheres, ellipsoids, cubes, cuboids and tetrahedrons. A few examples are shown next to equations 3.7.1 to 3.7.5. However, in the case of irregular shapes these may be modelled using several partly overlapping primitives. The beauty of this method is that the potential field generated is additive and decays rapidly with distance from the object centre, and the net effect of all the obstacles is easily evaluated through simple superposition of the fields.

This aggregate potential field serves three purposes: (1) It provides the optimiser with information regarding the degree of obstacle encroachment; (2) it provides directional information that indicates to the optimiser how trajectories must change in order to avoid encroachment; (3) it finally provides a measure of how close a trajectory is to breaching obstacle boundaries. When numerous objects are involved, the computational effort to evaluate each trajectory node becomes quite high, especially when transcendental functions are involved. The most likely solution would require a hardware-accelerated (eg: FPGA) implementation of the database, to provide rapid access, and a fast encroachment evaluator for the optimizer. During simulation, a parallelized GPGPU CUDA implementation worked very well to accelerate this step, often by factors in excess of a thousand.

3.7.4 Aerodromes

Airspace modelling would not be complete without including the location of FIRs and airports (Fig. 3.7.2), including orientation, capacity, altitude and number of runways. This data was painstakingly compiled from a number of databases to produce a reasonably accurate picture of the European airspace, shown in **Figure 3.7.2** below.

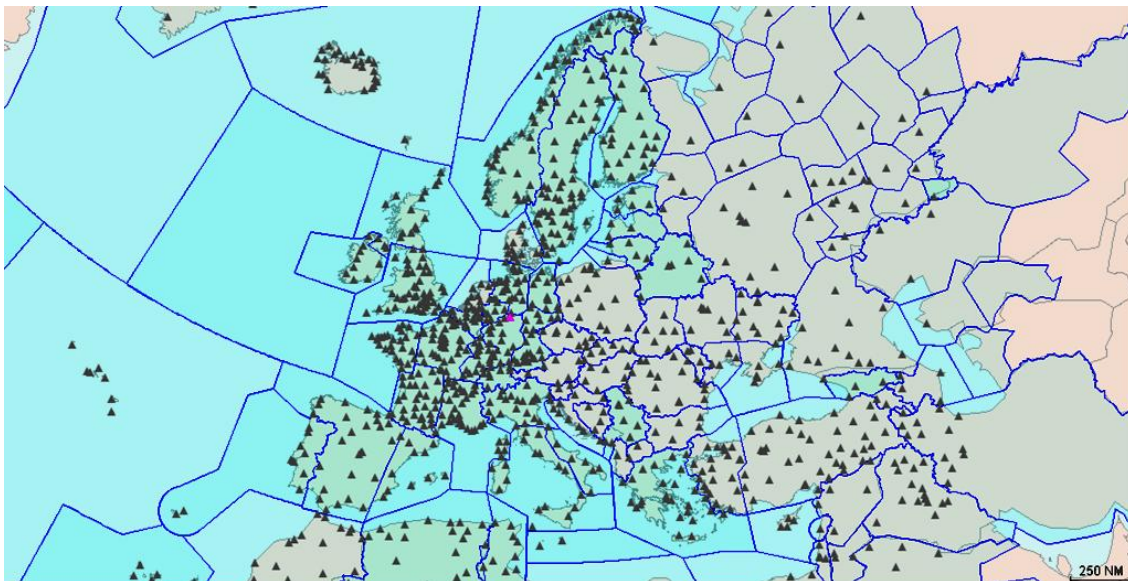


Figure 3.7.2: Location of Europe’s Operational Airports [Generated with SkyView v.2.5.1]

3.8 TRAFFIC MODELLING FOR ATM

A realistic ATM simulation must start by recreating typical continental air-traffic patterns. This may then be extended to explore various scenarios of traffic growth, either by extrapolating current growth trends or by analysing the underlying demographic realities of developing nations. We focussed our analysis to the European continent.

3.8.1 Intra-Continental 24-Hour Flight Data Samples

A practical way of modelling European traffic is by studying current traffic behaviour using 24hour traffic samples to or from the European continent as collected by EuroControl. The dataset used in this study corresponded to 26th June 2009 and represented a total of 31,444 flights of which 44 were aborted before take-off and 281 were go-arounds. A full breakdown is given in **Table 3.8.1**. A further 462 flights were to-or-from unregistered locations and might have been classified military flights. After

Flight Category	Qty
Total Flights Recorded in Sample	31444
Less the Aborted Takeoffs	- 44
Less the Indeterminate Flights*	- 462
Less the Go Around Traffic	- 281
All Regular Flights Recorded	30657
Less Light Aircraft Traffic (<50T)	- 10644
Regular Airliner Flights Recorded	20013
*Flights Starting or ending at Unknown Locations	

Table 3.8.1: Flights Breakdown in 24h Sample

deducting these special cases, 30,657 regular flights remain. 10,644 of these are light traffic less than 50 Tons, leaving 20,013 regular airliner flights which were further analysed. This traffic is beautifully visualised in the traffic superposition shown in **Figure 3.8.1**. Various high density zones are immediately visible over North Western Europe.

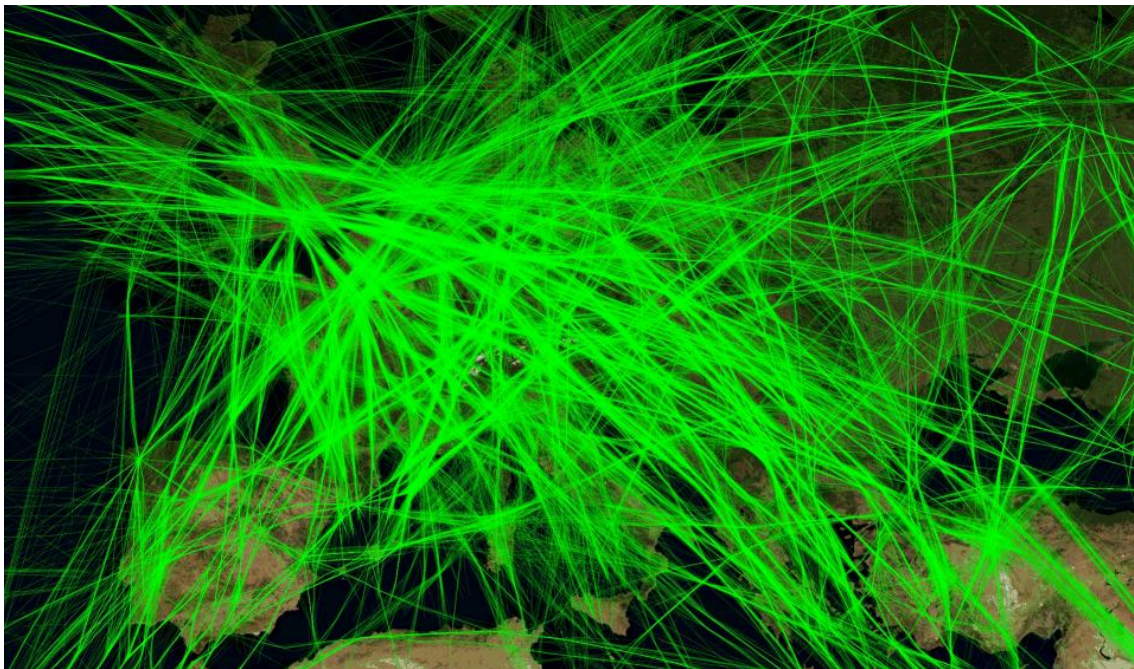


Figure 3.8.1: Europe – 24 hours of flights above 30,000 feet [Source: EuroControl]

The data describes flights as a series of waypoints traversed at given times, which means that fairly detailed trajectories can be constructed from this data. Software tools were written in Matlab to cross-section the data in several ways, thereby allowing important statistics to be extracted such as the activity by airport, by aircraft type and by route.

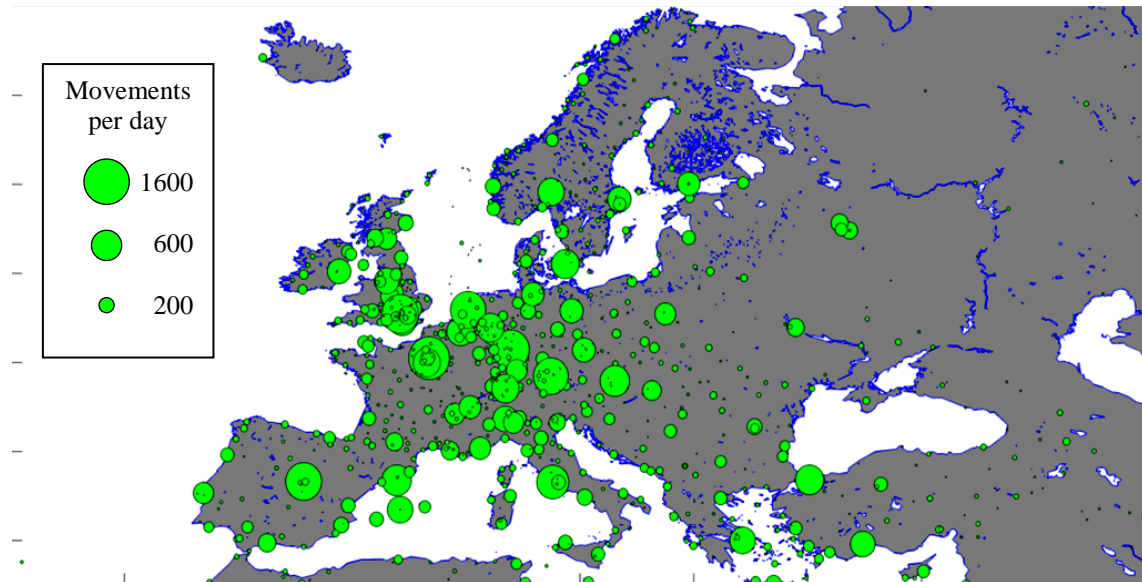


Figure 3.8.2: Europe’s Busiest Airports: Circle areas indicate daily aircraft movements

The graphic in **Figure 3.8.2** shows the busiest from about 1,137 European airports analysed. Every green circle denotes one airport and the area of which indicates the number of aircraft movements recorded over 24 hours. As is expected, airports associated with the major capital cities carry the majority of European traffic. Heathrow (London), Paris (Charles de Gaulle), Amsterdam (Schipol), Rome (Fiumicino), Madrid (Adolfo Suárez), Istanbul (Atatürk), Frankfurt and the rest of the top 20 busiest airports carry more airliner traffic than all the remaining European airports combined. This is highly suggestive of the high air traffic congestion that is bound to arise in the vicinity of these important hubs. **Table 3.8.2** overleaf, shows the interconnection magnitude between these 20 European hubs in matrix form and accounts for 61.2% of all European traffic or 18,776 individual flights.

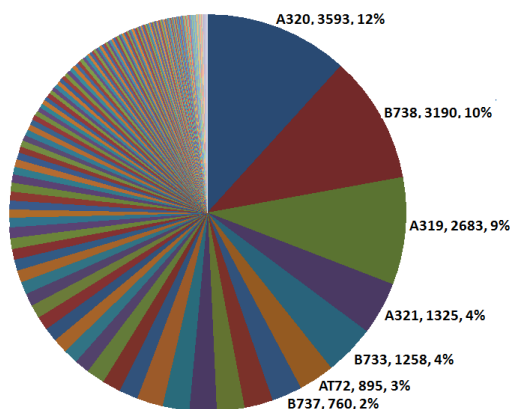


Figure 3.8.3: Flight Distribution by Aircraft Type

The traffic sample also contains information on the kind of aircraft used for each flight as well as the Maximum takeoff weight (MTOW) and flight level. This allows further analysis and estimation of the fuel consumed by each flight category and is especially useful in calculating efficiencies environment impact. Eurocontrol’s BADA model data includes

substantial information on the rate of fuel burn for each aircraft type at different altitudes and configurations. **Figure 3.8.3** shows the distribution of European flights by aircraft type.

ICAO		LAT		ARRIVALS																				SUB-TOTALS		
DEPARTURES		LOW	TOT	LFPG	EGLL	EDDF	LEMD	EHAM	EDDM	LIRF	LEBL	LOWW	EGKK	LTBA	EKCH	EDDL	LFPO	LSZH	EBBR	LEPA	ENGM	LGAV	LTAI			
		49	783	2.5	0	16	20	15	16	16	19	15	10	2	9	10	12	1	12	1	0	5	8	1	188	
LFPG	PARIS CH DE GAULLE	49	783	2.5	0	16	20	15	16	16	19	15	10	2	9	10	12	1	12	1	0	5	8	1	188	
EGLL	LONDON/HEATHROW	51.5	678	-0.5	16	0	16	13	19	15	11	8	7	0	6	12	10	0	13	13	1	9	7	0	176	
EDDF	FRANKFURT MAIN	50	676	8.6	18	16	0	13	14	13	8	6	15	0	9	8	8	0	11	11	6	7	5	5	173	
LEMD	MADRID BARAJAS	40.5	650	-3.6	14	13	13	0	9	8	11	49	2	7	3	3	5	13	6	9	28	0	4	0	197	
EHAM	SCHIPHOL AMSTERDAM	52.3	622	4.8	14	21	13	10	0	14	8	12	11	8	6	11	6	0	11	6	3	9	4	5	172	
EDDM	MUENCHEN 2	48.4	612	11.8	16	15	15	8	14	0	8	7	13	3	6	8	24	0	9	11	8	4	7	3	179	
LIRF	ROME FIUMICINO	41.8	509	12.2	16	12	8	11	8	8	0	9	4	6	4	3	2	0	4	7	0	2	9	0	113	
LEBL	BARCELONA	41.3	435	2.1	14	8	6	52	12	7	9	0	4	6	2	5	5	8	4	6	31	2	2	0	183	
LOWW	WIEN SCHWECHAT	48.1	428	16.6	9	8	16	2	11	12	4	4	0	2	7	4	13	1	12	9	2	3	3	14	136	
LTBA	ISTANBUL-ATATURK	41	410	28.8	10	6	10	3	6	6	4	2	5	0	0	3	5	0	4	3	0	1	4	20	92	
EGKK	LONDON/GATWICK	51.1	407	-0.2	1	1	0	7	8	3	6	6	2	0	0	6	3	0	2	2	6	3	2	3	61	
EKCH	COPENHAGEN KASTRUP	55.6	383	12.7	9	12	8	3	11	8	3	5	4	6	2	0	8	1	6	8	6	16	1	1	118	
EBBR	BRUSSELS NATIONAL	50.9	378	4.5	2	13	11	9	7	11	7	6	9	2	4	9	0	0	7	0	2	5	4	3	111	
LFPO	PARIS ORLY	48.7	370	2.4	2	0	0	14	0	0	0	8	1	0	0	1	3	0	0	0	1	1	1	0	32	
LSZH	ZURICH	47.5	371	8.5	12	12	12	6	10	9	4	4	13	3	5	6	13	1	0	9	4	3	3	4	133	
EDDL	DUESSELDORF	51.3	369	6.8	11	10	7	5	6	24	2	5	12	3	5	8	0	3	13	0	10	3	1	6	134	
LEPA	PALMA DE MALLORCA	39.6	326	2.7	0	1	6	26	2	7	0	26	2	6	0	5	10	1	3	2	0	4	0	0	101	
ENGM	OSLO/GARDERMOEN	60.2	338	11.1	5	9	7	0	9	4	2	2	2	3	1	18	3	1	3	5	3	0	0	2	79	
LGAV	ATHINAI E. VENIZELOS	37.9	334	23.9	9	7	5	4	4	7	9	1	4	2	5	1	1	1	3	5	0	0	0	0	68	
LTAI	ANTALYA	36.9	304	30.8	0	1	5	0	3	7	0	0	10	3	21	1	5	0	4	4	0	3	0	0	67	
SUB-TOTALS			178	181	178	201	169	179	115	175	130	62	95	122	136	31	127	111	111	80	65	67				

Table 3.8.2: Interconnection Matrix between Europe’s 20 Busiest Airports

Figure 3.8.4 (a) shows the variation in traffic over the course of a single day. The traffic sample starts with a clean sheet, ignoring all airborne flights active at midnight, and begins to log all new flights entering European airspace during the subsequent 24hours. The traffic log extends through a further 24 hours until all the flights that started late during the first day have landed. This means that to generate the periodic graphic shown, one must overlap and wrap around the tail of the traffic log with the leading traffic and this then gives the full picture of the cyclical traffic volumes at any hour of the day.

European domestic flights increase rapidly at the early hours of the morning, initially attributed to central European traffic, to be joined by the UK within an hour. Incoming and outgoing extra-European traffic is multimodal, peaking at 3am and 3pm, reflecting the different time zones of the main departure sites, across the Atlantic. The net effect of all three contributors is that traffic peaks to substantial levels around the middle of the day, with an average of 3000 airborne aircraft at any given hour between 6am and 6pm. Central ATFM services try their best to spread this traffic load evenly over the day, and this would explain the relatively flat top of the graph.

The optimality of the trajectories were also analysed, leading to an interesting finding that flights over the European continent average 6.5% longer (1.63 Million NM) than they strictly need to be, based on direct great circle routes totalling 25 million NM. This figure worsens significantly if one had to restrict the statistic to European Domestic Flights, which are highly exposed to European capacity limitations. **Table 3.8.3** ignores all traffic shorter than 250km to restrict our view to the main transport market. As expected, transatlantic flights are the least affected in percentage terms due to the long distance direct routes travelled over the ocean. Thus, although these account for close to half the total km. Tons, in the system, the excess fuel burn pales in comparison to what is consumed in domestic European flights. The latter account for over half the fuel wastage and excess CO₂.

Flight Category (> 250km)	Qty (#)	MTOW (Tons avg.)	Total Transport (km.Tons)	Elongation			Excess Fuel (Tons)
				Mean (%)	Max (%)	Min (%)	
Domestic European	22000	66	1,619,072,748	9.15%	239%	0.00%	5,357
Overpassing Europe	493	114	295,183,590	5.51%	95%	0.00%	569
North American	1258	189	1,628,599,577	2.37%	14%	0.04%	1,484
South American	120	197	214,370,510	1.71%	7%	0.25%	140
African	1078	107	433,320,087	4.34%	62%	0.14%	634
Middle Eastern	1156	121	506,272,383	5.61%	45%	0.45%	1,048
Far Eastern	463	191	752,707,583	3.50%	23%	0.53%	1,009
Russian Federation	992	86	182,226,438	5.30%	37%	0.52%	351
All Regular Flights	30657	77	5,661,438,121	6.48%	3900%	0.00%	10,827
All Flights > 250km	27559	80	5,631,745,914	6.30%	239%	0.00%	10,592
All Airliners > 50T	19183	102	5,425,373,873	5.66%	239%	0.00%	9,815

Table 3.8.3: Stratified Efficiency for All flights > 250km

Moreover this inefficiency changes over the course of a day and seems related to traffic density. In fact there is a high diurnal correlation between flight elongation and total airborne traffic at any hour. This can be observed in **Figure 3.8.4 (b)**. This is understandable because at high load, air traffic controllers will naturally require some flights to detour around areas of high traffic density, resulting in an associated increase in flying time.

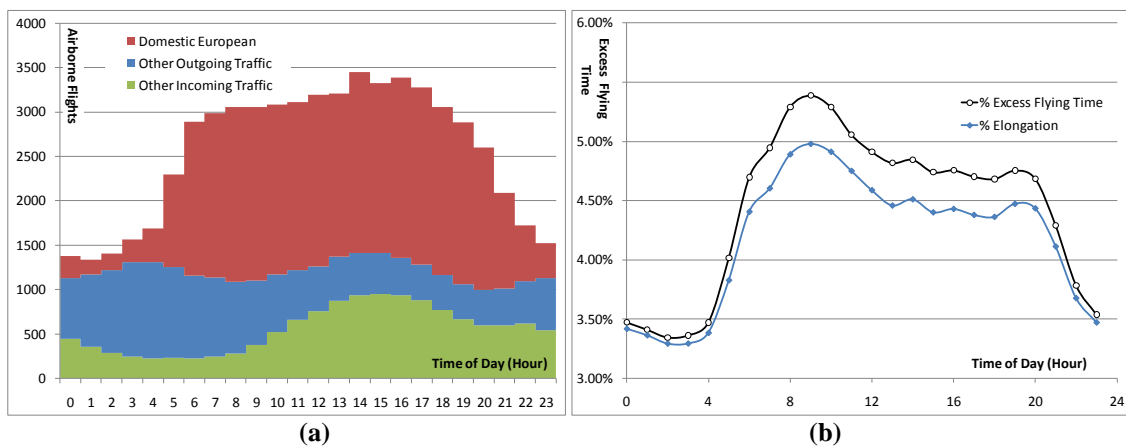


Figure 3.8.4: Diurnal Cycles in European Traffic, and associated Changes in Efficiency

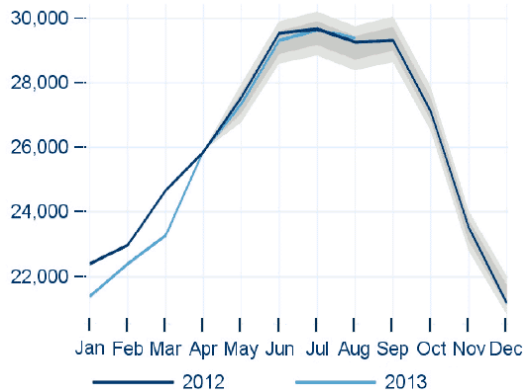


Figure 3.8.5: Seasonal Variation in European IFR Traffic (Source: Eurocontrol)

In addition, the fast rise in new traffic in the morning between 4am and 7am, as shown in **Figure 3.8.4 (a)**, leads to a very marked increase in inefficiency that seems to peak around an hour later at around 8am, as shown in **Figure 3.8.4 (b)**. This could be attributed to the inability of the air traffic system to reactively adjust rapidly enough to the sudden onset of new load, leading to additional inefficiency. Although, this sample represents a single day, this effect is entirely expected and would be interesting to analyse its recurrence over several other

traffic samples, and will be looked at in future studies.

Furthermore, this has to be seen in the context of a whole year, where the June sample represents the worst case scenario from a seasonal perspective as shown in **Figure 3.8.5**. Although 6.5% is not an alarming loss, this daily inefficiency translates into about 18.2 million tonnes of excess CO₂ per year and costs the industry more than €2.9billion in excess fuel costs at current prices. This further strengthens the case for the CATM project because there is a significant scope for fuel reduction through trajectory optimisation.

3.8.2 A Probabilistic Traffic Model

Aside from measuring the shortcomings of the current system, traffic analysis is important in order to establish the typical patterns of activity between the nodes of the air traffic system, which can then be used as an empirical basis for a stochastic *traffic generator* model for use in a verisimilar CATM simulator. A graph was generated that interlinks all airports in the system where each edge in the graph represents the probability of a flight arising between the two connected airports, as a function of time. **Figure 3.8.6** encodes this graph by the thickness and colour of each edge.

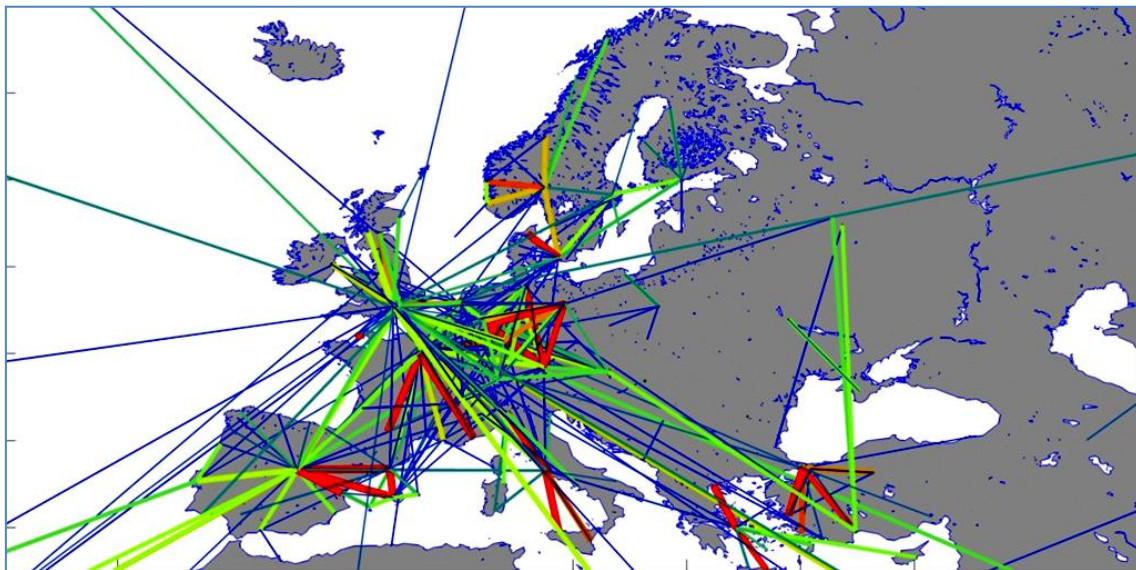


Figure 3.8.6: Most heavily used interconnections from Europe's busiest airports (>7 flights per day)

To mimic real traffic, a flight schedule needs to be constructed. The resolution of the schedule determines the peak traffic. For traffic that matches typical volumes, a schedule with 1 second intervals can be used to cover the duration of a day. This amounts to 86,400 possible time slots. A random number generator selects real numbers uniformly for every slot in the schedule, in the interval between 0.0 and 1.0. If uniform traffic of around 30,000 flights is required over the day, a simple threshold is set to 0.65277 (= 30,000/86,400). If a random number α_i exceeds the threshold, a flight is synthesised, if not, a new random number is selected for the next slot in the schedule. After 86,400 random trials are attempted we should be left with approximately 30,000 flights at random intervals aligned to a 1 second grid. If time-varying traffic is required the threshold can be adjusted accordingly over the span of the schedule, hour by hour or as necessary.

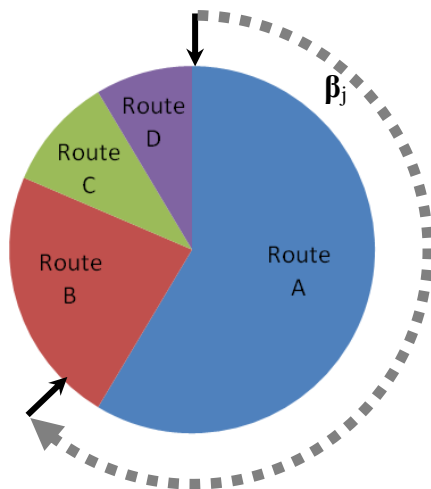


Figure 3.8.7: Roulette Selection

However, in order to generate traffic realistically, we also require an algorithm that generates flights between pairs of airports with probabilities that are proportional to the actual utilisation of each specific route. This can be achieved neatly using roulette selection (see: **Figure 3.8.7**). Whenever a flight needs to be synthesised to fill any time slot in the schedule, another random number $0 < \beta_j < 2\pi$ needs to be generated from another uniform source.

This number is used to rotate a roulette by β_j radians as shown in the figure. The roulette pointer determines the route selected. If the roulette consists of a pie chart depicting the percentage of measured traffic attributable to each route, then the probability of selecting any route by this procedure will be exactly proportional to the measured popularity of that route. Similarly, aircraft types can be selected using the pie chart of **Figure 3.8.3**. Traffic can thus be generated at any multiple or fraction of typical traffic volumes using the technique.

3.8.3 Traffic Density

In order to ground simulations to realistic scenarios, the regional worst case density of typical traffic had to be worked out. The data was split into one minute long time intervals and all the waypoints traversed at every specific time were grouped and counted according to their location on a 1x1 degree reticle on the WGS84 spheroid. This kind of grid is non linear, but for Greenwich, one latitudinal degree works out to 110.6 kilometres, while a longitudinal degree is 69.19 kilometres. Therefore, if we take Greenwich to represent the approximate latitudinal average of the European continent between the Scandinavian Laplands and Sicily, each square in the reticle can be assumed to represent an average area of around 7700 km². There are of course better but significantly more involving ways to map out the density. One improvement is to assume a trapezoidal grid and work-out the sides of each block, based on the exact latitude and longitude of the edges. The aircraft trajectories can also be interpolated with splines to yield more accurate estimates of aircraft location at any instant in time.

However, the forgoing gives a good indication of average density given the coarse sample data available. **Figure 3.8.8** shows the colour coded density of traffic across the reticle at six different times along the day.

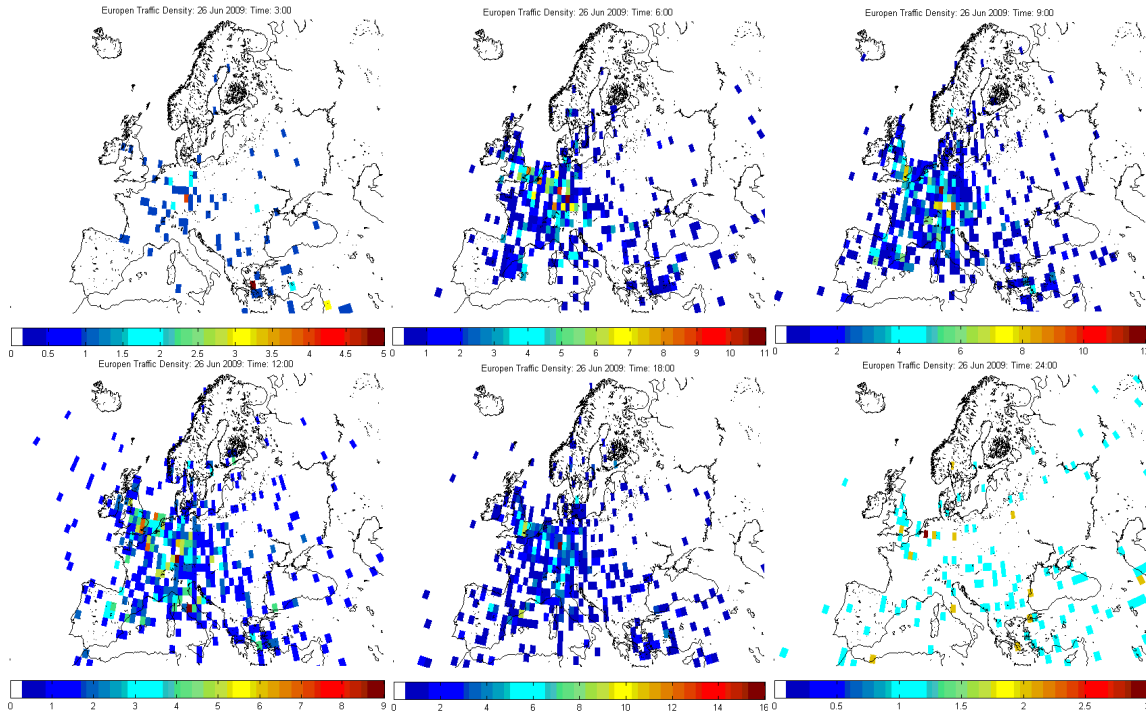


Figure 3.8.8: European Traffic Density by location and time: {3:00, 6:00, 9:00, 12:00, 18:00, 24:00}

Therefore, the highest traffic density reached on the hour, on the busiest day of 2009, was over Frankfurt, where 16 aircraft were flying in the same 7700 km² cell at 18:00. This translates into a traffic density of around 2000 aircraft per million km² (or 54.9 per 10,000 NM²). Slightly higher traffic density is observed over London (Heathrow) where it peaks at 65.2 aircraft per 10,000 NM² at 9:19am. On the other hand, when averaged of a larger area, by conducting the same exercise directly over a coarser reticle of 13 degrees (**Figure 3.8.9**) (corresponding to 1 million km²), the peak density averages out at 255 aircraft per million km² (or 8.7 per 10,000 NM²) over Germany and that takes place at 10:58am. **Table 3.8.4** summarises these findings. When generating synthetic traffic schedules, or interpreting simulation results, it is important to ensure that the traffic density is contrasted with these figures in order to be able to draw meaningful comparisons.

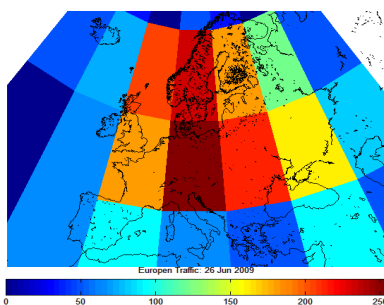


Figure 3.8.9: Density at 10:58am

Flight Category (> 250km)	Peak Qty (#)	Peak Time	Sampling Area (km ²)	Density in (10,000NM ²)
Domestic Europe	1966	14:00	11,500,000	5.7
Over Germany	255	10:58	1,000,000	8.7
Over London	19	9:19	7,700	65.2
Over Frankfurt	16	18:00	7,700	54.9

Table 3.8.4: European traffic density over various areas

3.9 WEATHER MODELLING FOR ATM

There is significant risk associated when planning ATM trajectories using an optimiser. This is because it can easily produce very tightly optimal solutions that are practically unable to re-adapt to the less-than-ideal real circumstances. An accurate statistical evaluation of the likelihood and distribution of weather cells is extremely important because it allows the SPO to insert just the right amount of slack to allow the system to re-adapt to unforeseen weather conditions, without major repercussions on future traffic schedules.

This can be implemented by simulating a weather system over the entire optimisation horizon of 1 year. The weather model need not evolve accurately like a weather forecasting model would be required to do. However, it needs to behave in a stochastically equivalent manner. In addition, the weather simulator need not iterate within the SPO. It is simply seeded once with real weather data and is allowed to simulate a fictitious weather system forward into the future, in fast time. The ensuing weather data is stored in a database and is used as a reference weather constraint model throughout the entire optimisation horizon of the SPO.

Allowances for inclement weather are made by SPO by judging on the average effects that this has on daily operations. This stochastic data is compiled by the Tactical Phase Optimiser (TPO) and serves to guide the weather simulator in building the reference weather database of probable weather conditions. Over time, this becomes increasingly representative of the average weather conditions expected day-by-day, hour-by-hour, over all continents, over an entire year. Weather is a stochastic chaotic process, making it hard to predict too much in advance with any accuracy. So an average weather expectation is the best that can be assumed by the SPO in the months and weeks leading to the moment of departure $[T_0^i]$ of each flight i .

However, as $[T_0^i]$ comes close, say $[T_0^i - 7\text{days}]$, fairly accurate weather predictions, as provided by the meteorological offices worldwide, become available. This data can of course be integrated into the SPO's solution. This way the SPO has 7 days time to gradually converge to the real meteorological picture. This process nearly eliminates the unpredictability of the weather constraints. By the time of departure $[T_0^i]$, real and measured weather data can be incorporated into the solution and the depth of weather forecast required spans just a few hours – the duration of the flight.

3.9.1 CFD Weather Simulation

Weather simulation is a complex fluid mechanics problem involving the world's atmosphere with an unevenly distributed energy input, resulting in localised convection currents, evaporation, advection of water vapour, cloud formation and precipitation. It is a system seething with activity and in constant evolution, where the future very much depends on the previous state and current stimuli. Classical physics describes the idealised flow of viscous fluids with the well known Navier-Stokes differential equation, named after the 18th and 19th century physicists, Claude-Louis Navier and George Gabriel Stokes. This vector algebra equation assumes conservation of momentum, mass and energy in a Newtonian continuous fluid and can be stated as in equation 3.9.1:

$$\rho \left(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} \right) = -\nabla p + \nabla \mathbf{T} + \mathbf{f} \quad (3.9.1)$$

where: ρ represents the fluid's density
 \mathbf{v} represents the fluid flow velocity vector
 p represents the fluid pressure
 \mathbf{T} represents the (deviatoric) component of the total stress tensor
 \mathbf{f} represents the forces acting on the fluid (per unit volume)
 ∇ represents the vector differential operator

The problem with applying the Navier-Stokes equation, is that with the exception of very simple and small scale examples, analytical solution of this equation is essentially impossible for large systems such as weather and had rendered progress in the area exceeding difficult until the advent of Computational Fluid Dynamics (CFD) which uses computers to evaluate these equations numerically.

However, detailed CFD simulation can be highly computationally demanding and is inadequate for real-time simulation of fluid flow. This motivated a computer graphics researcher, in 1999, to develop a simplified stable numerical technique to simulate fluid like behaviour for application in computer games and such. The well regarded paper by Jos Stam, called "Stable Fluids" [3.20] lays down a 4 step method for generating life-like fluid flow patterns. It simulates a viscous fluid entraining a secondary substance with variable density, such as smoke, dust or cloud. The underlying Navier-Stokes equation can be somewhat simplified, assuming unity density and incompressibility, and rearranged to solve for the fluid-flow velocity vector field – in our case: The wind-flow field. (Eq. 3.9.3)

$$\text{Wind flow:} \quad \frac{\partial \mathbf{v}}{\partial t} = -(\mathbf{v} \cdot \nabla) \mathbf{v} + \nu \nabla^2 \mathbf{v} + \mathbf{f} \quad (3.9.3)$$

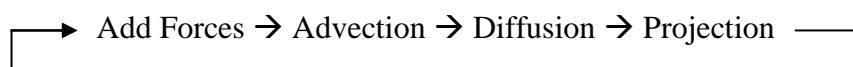
where: \mathbf{v} represents the wind flow velocity vector at any point in the field
 ν represents the air's kinematic viscosity
 \mathbf{f} represents the forces acting on the air mass
 ∇ represents the vector differential operator

For wind-entrained cloud we can use the equation 3.9.5 describing advection:

$$\text{Advection:} \quad \frac{\partial \rho}{\partial t} = -(\mathbf{v} \cdot \nabla) \rho + \kappa \nabla^2 \rho + S \quad (3.9.5)$$

where: ρ represents the cloud density
 κ represents a diffusion constant
 S represents the source of cloud

By decomposing the vector field into the sum of a mass conserving field (with zero divergence) and a gradient field (of an underlying scalar density field), in his paper, Stam suggests a 4 step procedure to simulate the evolution of fluid flow and advection over time.



3.9.2 Synthetic Wind Patterns

The model simulates a synthetic weather system and can generate spatio-temporal weather data in real-time. Through iterative application of these four steps, the progression over time of the vector field can be simulated for a series of discrete time steps. This vector field is highly reminiscent of global wind patterns and can be used to provide a CATM simulator with the gradually evolving time-variant wind patterns typical of real weather systems.

A wind simulator based on Stam's paper was written in C++ and interfaced to Matlab using Mex for analysis. The resulting wind vector field is exemplified in **Figure 3.9.1**. A full treatise of the mathematical procedure is far beyond the scope of this work and interested readers are referred to Stam's excellent papers on the subject: [3.20], [3.21] and [3.22]. It will suffice to say that the simulation using Stam's procedure can be conducted in any number of dimensions greater or equal to 2 and periodic boundaries allow a closed weather system to be simulated by stitching opposing boundaries to form an N-dimensional torus, where the boundaries wrap around indefinitely in all directions. This is not quite spherical like planet earth, but is still good enough in the sense that it is spatially periodic while temporally a-periodic.

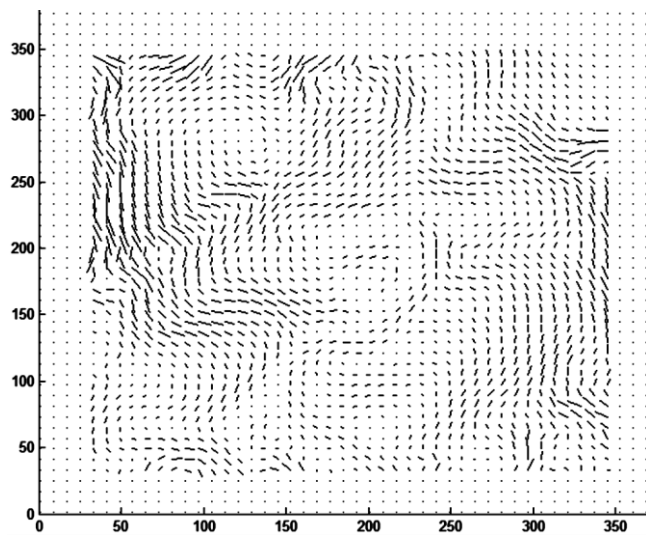


Figure 3.9.1: 2D Wind vector field generated with random distributed forces hidden in the margins.

3.9.3 Generating Clouds and Weather cells

Through the advection step, a scalar density field is also generated in the process and this can be used to mimic the formation and dissipation of cloud. Localised sources of cloud are carried along by the wind vector field and this results in a very compelling rendition of weather cell formation as can be seen in evolving in **Figure 3.9.2**.

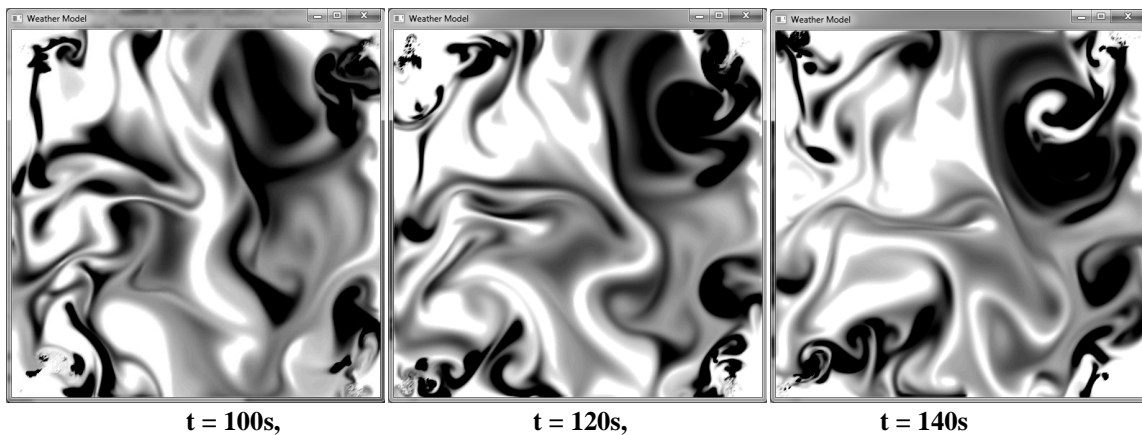


Figure 3.9.2: Evolving Cloud Advection in the Weather Simulator

The cloud formation must be balanced with a degree of cloud dissipation to avoid saturating the airspace with cloud. Weather cells are merely assumed to be zones of high cloud density. All regions of the airspace exceeding some arbitrary degree of cloud density are deemed non-traversable and are therefore treated as moving obstacles in the airspace. In a CATM optimiser, these obstacles can be parametrically modelled using penalty functions or potential fields just like stationary restricted airspaces. Thus the regions of high density may be modelled using overlapping spheroids, which can in turn be used to generate a potential field. Alternatively, if detailed geographic data is available, the degree of penalty (or potential) may be determined directly by sampling the cloud density field.

This methodology can be extended to volcanic ash cloud dispersion, and would have resulted in far less disruption if it were adopted during the Icelandic Eyjafjallajökull eruption of 2010. As can be seen in the infrared satellite photo of **Figure 3.9.3**, ash cloud dispersion also follows very gradual and predictable patterns that could have easily been modelled as a moving obstacle with changing shape. Thus rather than blanket grounding all western European flights, for days, this kind of online adaptation would have allowed the flights to navigate the ash cloud safely, much like is done today with weather cells.

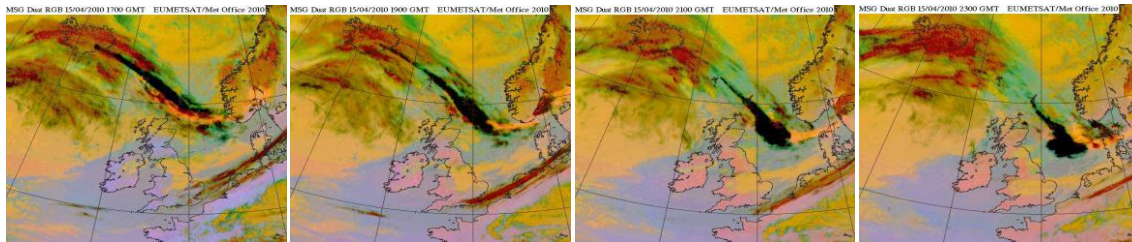


Figure 3.9.3: Icelandic Volcanic Dust Cloud Advection in April 2010 (Eumetsat)

3.9.4 A Self-excited Circular Weather Domain

The simulator requires an external stimulus or disturbance in order to trigger fluid flow. It also requires sources of cloud. This can be achieved by exerting localized forces on the fluid, which are taken into account in the first stage of Stam's procedure. Momentary forces result in chaotic fluid motion that persists for a while, until the viscosity of the fluid eventually dissipates all shearing motion and the swirls and vortices eventually die away.

The creation of convincing weather patterns requires continuous application of such forces at various points to keep the system energised and dynamic. Cloud must also be added from time to time. The location, direction and intensity of such forces and associated cloud density must follow some stochastic process. However, these forces may create very high fluid velocities in the immediate vicinity of their application. This would be problematic if this were to occur in the middle of the airspace, as it would result in undesirable discontinuous behaviour in the vector (wind) field. Same applies to cloud. It would be unrealistic if cloud were added sporadically throughout the airspace. However, this problem can be circumvented by restricting these stimuli to the margins of the airspace, which are not traversed by any aircraft. The effects of the stimuli gradually diffuse into the active portion of the airspace.

3.10 ENVIRONMENTAL IMPACT MODELLING FOR ATM

The environment is one of the principle areas that air transport impinges on. The volume of global air traffic has increased exponentially following World War 2 with no signs of abatement, and has reached levels that are now comparable with most other major industries. It is therefore one of the primary objectives when speaking of optimisation of air traffic management. However, environmental impact is an intensely studied technical subject and only a brief overview will be covered in the following sections. Numerous impact models are available in the literature and are subject to frequent updates as the underlying science is better understood. The interested reader is referred to the bibliographic references for detailed information.

3.10.1 Carbon Footprint and CO₂

The carbon footprint of aviation is one of the most vigorously debated subjects in the area of environmental impact. This is because of the imminent danger posed by climate change. Carbon dioxide (CO₂) is considered to be the primary greenhouse gas emitted, and accounts for over 80% of total anthropogenic greenhouse gas emissions. In 1992, it was estimated that air transport accounted for about 2% of the world’s total CO₂ emissions, or 13% of transport [3.24]. However, the seemingly unstoppable exponential rise in aviation is cause for concern. This CO₂ contribution has since increased to about 2.5% (650 megatons per year), meaning that aviation is exceeding the pace of growth of other sectors [3.25].

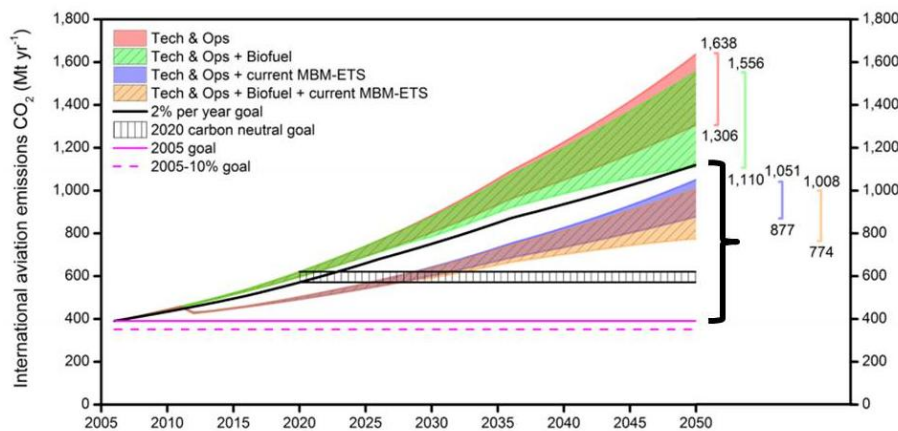


Figure 3.10.1: Bridging the aviation CO₂ emissions Gap [3.26]

In the most optimistic of scenarios (Figure 3.10.1), it still appears that the increase in CO₂ emissions will greatly exceed the 2005-10% limits proposed by the European Union at the 37th ICAO General Assembly [3.26]. Carbon neutrality is an even more distant goal.

Modelling the CO₂ footprint of aviation is not difficult. This is obviously directly related to fuel consumption since practically all fuels in use today are carbon based. Therefore, when modelling for the CO₂ footprint, one may simply link this effect to the fuel consumption model that is already accurately modelled by BADA [3.7]. Obviously, it is the cumulative effect that matters, and therefore optimising one flight the expense another is rather futile. Minimising *gross* emissions (fuel-burn) must be the objective of CATM.

3.10.2 Noise Mitigation

When operating close to dense urban areas, noise abatement is an important consideration. This is an often politically sensitive matter that is taken very seriously by legislators and operators alike. The Integrated Noise Model (INM), currently at version 7.0d, has been developed by the FAA [3.27], as per the SAE AIR 1845 standard [3.28], and is provided as a tool for the community. It allows a noise profile to be created for each aircraft type, and with over 100 models to choose from, it can realistically estimate noise levels as perceived at several points, at ground level, given particular trajectories. Cumulative surface noise contours (Fig. 3.10.2) exceeding a certain sound pressure level (in dBA) are not allowed near urban areas exceeding a certain density and may have to follow alternate maths to the TMA. This can be realistically expected to affect rates of descent and the number of aircraft flying per unit time over certain areas.

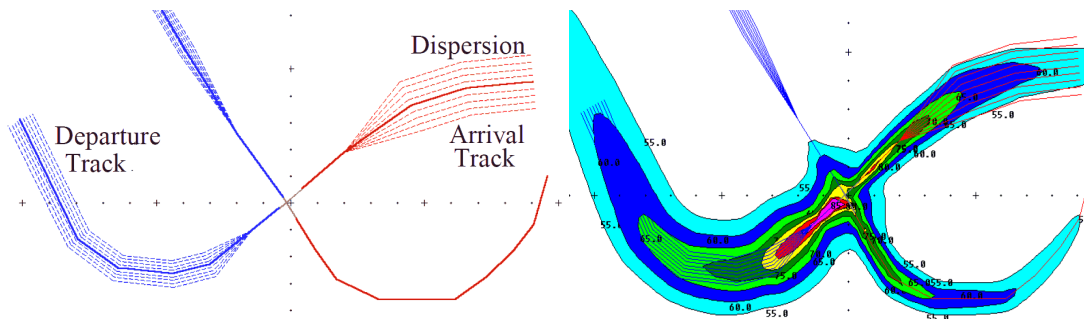


Figure 3.10.2: Sample INM noise contours generated with the FAA application [3.29]

There has been much progress to reduce the noise profile of aircraft engines over the years. However, there is a limit and the noise impact remains clearly ATM related. A noise impact model is easily integrated into a trajectory optimiser by assigning a cost in relation to the integrated noise level at every urbanised area. No difficulties are foreseen to combine a fast, but simple INM-like model into the optimiser. This can be based on the same SAE AIR 1845 calculation procedure and is easily written in C++. Noise minimisation can either be included as an additional objective or posed as a region-dependant constraint.

3.10.3 Other Pollutants and ATM

Besides CO₂ and noise, aircraft engines produce a whole array of other insidious pollutants that are either directly toxic to humans or cause adverse reactions or chain reactions in the wider ecosystem. The impact is, either way, still borne by humanity and wildlife. Most of these pollutants are functions of engine performance and as such depend on the careful design and maintenance of the said engines by their manufactures and operators. They include the following:

- Sulphur Oxides (SO_x), acids: SO₂, SO₃, H₂SO₃, H₂SO₄
- Nitrogen Oxides (NO_x), acids: NO, NO₂, HNO₂, HNO₃
- Particulates and Soot: C
- Unburnt Hydrocarbons: C_nH_m
- Water Vapour: H₂O

These pollutants follow complicated chemical and ecological pathways that ultimately result in environmental damage, as well as economic and social losses. This is far too complicated to discuss in any detail here. However, a pictorial representation of how aviation emissions unleash their effects is given in **Figure 3.10.3** with a detailed explanation given in [3.25].

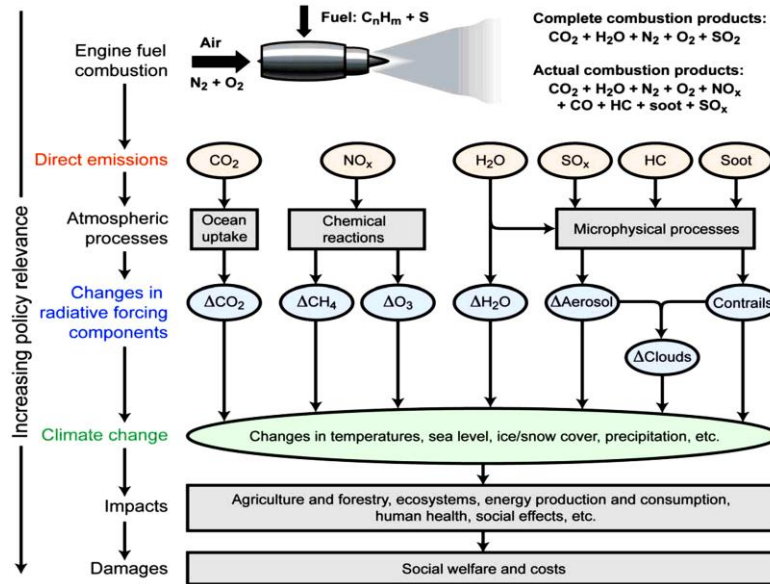


Figure 3.10.3: Aircraft Emissions and their effect on climate change [3.25]

Some pollutants are due to contaminants present in the fuel (like sulphur) and are only mitigated by minimising the fuel consumed by the engines. However, a number of other pollutants are emitted during certain operating regimes of the engine. This means that it is possible to carefully plan thrust requirements (and hence trajectories) such that the generation of some of these pollutants is minimised. For example, nitrogen oxides are generated when nitrogen and oxygen in the air combine when they are exposed to the high temperatures inside the turbine. The temperature is obviously dependant on the level of thrust which is a parameter that CATM can control. CDAs are another ATM related method to reduce NOx [3.30].

And finally another subset of pollutants has most effect at certain altitudes, and in certain atmospheric conditions. Methane, for example, has a powerful ozone depleting potential that is maximised when it is emitted at high altitude. Ordinary harmless water vapour in aircraft exhaust, when emitted at high altitudes, may have a greenhouse effect. This happens when the temperatures are such that the water forms small ice crystals which are commonly visible as *contrails*. However, at certain flight levels the atmospheric conditions may be such that these ice crystals trigger further cloud formation by nucleating water vapour in the air. This is commonly visible as cirrus cloud and has a powerful heat retention and temperature forcing effect on the planet. Soot has a similar cloud generation effect at certain altitudes. On some days, the Atlantic is significantly covered with aviation generated cloud. This can obviously be mitigated by adjusting the aircraft trajectories such that these flight levels are avoided during cruise. CATM can exert a powerful role in the regard. With careful modelling by environmental scientists, these processes may be characterised such that the conditions

under which they prevail can be predicted. This information can then be used by an optimiser. Simplified linearised models for these pollutants are given in equations 3.10.1 [3.30], however these ignore altitude effects.

$$\begin{aligned}
 pCO_2 &= N_E \cdot (EICO_2) \cdot F_f \\
 pCO &= N_E \cdot (EICO) \cdot F_f \\
 pHC &= N_E \cdot (EIHC) \cdot F_f \\
 pNO_X &= N_E \cdot (EINO_X) \cdot F_f
 \end{aligned}
 \tag{3.10.1}$$

where: N_E represents the number of engines in an aircraft
 F_f represents the fuel flow to the engines in g/sec
 pCO_2 represents the number of grams of CO_2 emitted per unit time
 pCO represents the number of grams of CO emitted per unit time
 pHC represents the number of grams of HC emitted per unit time
 pNO_X represents the number of grams of NO_X emitted per unit time (3.10.2)

$EICO_2$ is an emission index as given in the ICAO engine emission database
 $EICO$ is an emission index as given in the ICAO engine emission database
 $EIHC$ is an emission index as given in the ICAO engine emission database
 $EINO_X$ is an emission index as given in the ICAO engine emission database

3.11 AIRPORT -OPERATIONS MODELLING FOR ATM

The effect of ground airport operations on ATM is very significant indeed. Unfortunately, the transition of any scheduled flight (from N_{FS}) to the airborne state (into N_{FA}) is subject to a high degree of uncertainty. Airline operations, security incidents and other miscellaneous random occurrences, represent a substantial majority of causes of delay that are encountered in the ATS as a whole, as was seen earlier in **Figure 1.1.2** in chapter 1.

The causes, magnitude, and frequency of these delays vary from airport to airport and from airline to airline and it is therefore hard to model properly. However, its effect is highly deleterious on the air traffic system performance and some form of random perturbation must be included in the overall system for realistic simulation.

This perturbation model is mostly intended to measure the effectiveness in redistributing slack and hence test the resilience of the autonomous scheduling and trajectory optimisation engine. Unless there are individual statistics available, the best effort method to model these effects is to base the perturbation model on the overall statistics published by Eurocontrol on a regular basis.

3.12 REFERENCES

- [3.1] Zaki, G., Bousmanne C., 2012, Mar 7, "Trajectory Management Concept," ATC Global, Technical workshop on 4D Business trajectory, Amsterdam, The Netherlands.
- [3.2] IATA, 2010, "A Blueprint for the Single European Sky - Delivering on safety, environment, capacity and cost-effectiveness," International Air Transport Association (IATA), pp. 1-22
- [3.3] "What is the SESAR project?", Single European Sky ATM Research (SESAR), European Commission Project, http://ec.europa.eu/transport/modes/air/sesar/index_en.htm, accessed Jun 2014
- [3.4] AIAA, 2004, Nov 17th, "Guide to Reference and Standard Atmosphere Models," ANSI/AIAA, G-003B-2004, AIAA Standards, American Institute of Aeronautics, 1801 Alexander Bell Drive, Reston, VA
- [3.5] ICAO, 1975, "Standard Atmosphere", ISO 2533:1975, International Organization for Standardization, Geneva, Switzerland, pp. 1-108
- [3.6] Nuic A., 2011, Apr, "User Manual for the Base of Aircraft Data (BADA)", Revision 3.9, EEC Technical Report No. 11/03/08-08, EUROCONTROL.
- [3.7] Nuic, A., Poles, D., & Mouillet, V. 2010., "BADA: An advanced aircraft performance model for present and future ATM systems." International Journal of Adaptive Control and Signal Processing, Vol. 24 No. 10, pp. 850-866.
- [3.8] AIRBUS, 2002, "Getting to Grips with Aircraft Performance," AIRBUS France - Flight Operations Support & Line Assistance, Blagnac, Cedex, France
- [3.9] Roussos, G. P., D.V. Dimarogonas & K.J. Kyriakopoulos, 2009, Aug 23-26, "Distributed 3D Navigation and Collision Avoidance for Nonholonomic Aircraft-like Vehicles," Proceedings of the 2009 European Control Conference (ECC'09), Budapest, Hungary
- [3.10] Glover, W. & J. Lygeros, 2004, "A Multi-Aircraft Model for Conflict Detection and Resolution Algorithm Evaluation," WP1: Identification and modelling of uncertain hybrid systems, HYBRIDGE, IST-2001-32460, Contract of European Commission
- [3.11] Renteux, J-L, 1987, "Aircraft Modelling Standards for Future ATC Systems," EUROCONTROL Division E1, Brétigny-sur-Orge, France
- [3.12] Taranenko, V. T. & V.G. Momdzhhi, 1986, "Direct variational method in boundary tasks of flight dynamics," Mashinostroenie Press, Moscow, Russia
- [3.13] Yakimenko, O. A., & Dobrokhodov, V. N., 1998, Jul., "Airplane trajectory control at the stage of rendezvous with maneuvering object algorithms synthesis." In Aerospace and Electronics Conference, NAECON 1998. Proceedings of the IEEE National pp. 375-382
- [3.14] Cowling, Ian, 2008, "Towards Autonomy of a quadrotor UAV," Dept. of Aerospace Sciences, Cranfield University, Cranfield, Bedfordshire, UK
- [3.15] Kirk, Donald E., 1970, "Optimal Control Theory: An Introduction," Prentice-Hall Inc., Englewood Cliffs, New Jersey
- [3.16] International Civil Aviation Organization (ICAO) Chicago Convention 1944, ICAO Annex 11: Air Traffic Services, Chapter 2, Section 2.6 and Appendix 4
- [3.17] CAA, 2014 Mar, "CAP 493: Manual of Air Traffic Services Part 1," Edition 5, Civil Aviation Authority, Aviation House, Gatwick Airport South, West Sussex, UK
- [3.18] World Airport Codes, Online Database, Fubra Limited, <http://www.world-airport-codes.com/>, Accessed: Nov 2013
- [3.19] Cowling, I., 2008, "Towards autonomy of a Quadrotor UAV," Ph.D. Dissertation, Cranfield University, Bedfordshire, UK.
- [3.20] Stam, J., 1999, Jul., "Stable Fluids" In Proceedings of the 26th annual Conference on Computer Graphics and Interactive Techniques, ACM Press/Addison-Wesley Publishing Co. pp. 121-128
- [3.21] Stam, J., 2001, "A Simple Fluid Solver Based on the FFT", Journal of Graphics Tools, Vol. 6, No. 2, pp. 43-52

-
- [3.22] Stam, J., 2003, Mar., "Real-time fluid dynamics for games." In Proceedings of the Game Developer Conference, Vol. 18, pp. 25.
- [3.23] Nolan, M., 2010, "Fundamentals of Air Traffic Control," Fifth Edition, Cengage Learning, Boston, MA
- [3.24] IPCC, 1999, "IPCC Special Report: Aviation And The Global Atmosphere," A Special Report of IPCC Working Groups I and III, Intergovernmental Panel on Climate Change (IPCC),
- [3.25] Lee, D. S., Fahey, D. W., Forster, P. M., Newton, P. J., Wit, R. C., Lim, L. L., & Sausen, R., 2009, "Aviation and global climate change in the 21st century." Atmospheric Environment, vol. 43, no. 22, pp. 3520-3537.
- [3.26] Lee, D. S., Lim, L. L., & Owen, B., 2013, "Bridging the aviation CO2 emissions gap: why emissions trading is needed." Report, Manchester Metropolitan University.
- [3.27] FAA, "The Integrated Noise Model (INM)" , https://www.faa.gov/about/office_org/headquarters_offices/apl/research/models/inm_model/, Accessed: Jun 2013
- [3.28] SAE, "Procedure for the Calculation of Airplane Noise in the Vicinity of Airports", Society of Automotive Engineers (SAE) International, AIR 1845 Standard, AIR1845A, 2012, Aug, 16
- [3.29] Trani, A. A., "INM - Integrated Noise Model Basics," CEE 4674 - Airport Planning and Design, Department of Civil Engineering. Virginia Tech
- [3.30] Alam, S., Nguyen, M. H., Abbass, H. A., Lokan, C., Ellejmi, M., & Kirby, S., 2010, Oct., "A dynamic continuous descent approach methodology for low noise and emission." In Digital Avionics Systems Conference (DASC), 2010 IEEE/AIAA 29th, pp. 1-E
-

Chapter 4

Computational Intelligence in CATM

It was suggested in previous chapters that both phases of the CATM problem involve searching a vast disconnected non-convex search space of feasible trajectories to single out the optimal solution. The solution consists of a set of optimal control policies for all aircraft that results in them flying minimum cost trajectories. Therefore this may be formulated as an optimal control problem. In this chapter we shall see how this can be accomplished. One may initially attempt to apply some of the very generic and powerful tools in the optimisation toolkit (like dynamic programming) that can theoretically handle CATM directly, elegantly and reliably to give globally optimal results. However, in-keeping with Bellman’s “curse of dimensionality,” [4.1] we will soon discover that all the computing time and memory in the universe would not suffice to produce any noticeable progress in solving the problem.

A different approach is clearly required. Since a generic methodology clearly exists, in the form of dynamic programming (DP) to solve all optimal control problems, the issue is not whether a method exists, but whether an *efficient* method exists. This is because it appears that the principal difficulty is a computational one. Thus, the research focus seems best directed towards complexity reduction to render the problem tractable. This may take several forms and will be the recurring theme of what will be discussed in this chapter. It is important that when dealing with such large problems one does not lose the perspective of what is relevant. With this mindset several practical solutions for CATM can be devised.

4.1 CATM PROBLEM FORMULATION

Without much loss of generality, we can initially ignore the distinction between the phases of the CATM problem. We will begin by casting the entire CATM problem using classical notation [4.1] as a large collection of optimal control problems, one per flight, each of which undertakes to minimise (or maximise) some arbitrary cost functional nJ .

4.1.1 Single Flight Problem

For any flight n , this cost functional nJ may be expressed in terms of the end points (Mayer), the running cost (Lagrange) or the sum of both (Bolza). Thus in general, for any one flight this cost may be formulated as the Bolza problem shown in eq 4.1.1.

$$\begin{array}{l}
 \text{Bolza} \\
 \text{Cost:}
 \end{array}
 \quad
 {}^nJ = \underbrace{\Phi[{}^n\mathbf{x}_0, {}^n\mathbf{x}_f, {}^nt_0, {}^nt_f]}_{\text{Mayer Cost}} + \underbrace{\int_{{}^nt_0}^{{}^nt_f} \mathcal{L}[{}^n\mathbf{x}, {}^n\mathbf{u}, t] dt}_{\text{Lagrange Cost}} \quad (4.1.1)$$

Thus if we consider just one aircraft at a time, the problem now entails finding the vector of controls ${}^n\mathbf{u}$ and trajectories ${}^n\mathbf{x}$ such that nJ is minimised to ${}^nJ^*$, subject to some trajectory constraints, boundary conditions and dynamic constraints relating ${}^n\mathbf{u}$ to ${}^n\mathbf{x}$. Thus for any flight $n \leq N$, the single flight problem can be stated as in eq. 4.1.2:

$$\begin{aligned} \text{Bolza} \quad & {}^nJ^* = \min_{{}^n\mathbf{x}, {}^n\mathbf{u}} {}^nJ = \\ \text{Cost:} \quad & \min_{{}^n\mathbf{x}, {}^n\mathbf{u}} \left\{ \Phi[{}^n\mathbf{x}_0, {}^n\mathbf{x}_f, {}^nt_0, {}^nt_f] + \int_{{}^nt_0}^{{}^nt_f} \mathcal{L}[{}^n\mathbf{x}, {}^n\mathbf{u}, t] dt \right\} \end{aligned} \quad (4.1.2)$$

Subject to:

$$- \text{First order dynamic constraints:} \quad {}^n\dot{\mathbf{x}} = \mathbf{a}[{}^n\mathbf{x}, {}^n\mathbf{u}, t] \quad (4.1.3)$$

$$- \text{Trajectory constraints:} \quad b[\mathbb{M}, \mathbb{O}, {}^n\mathbf{u}, t] \leq 0 \quad (4.1.4)$$

$$- \text{Boundary conditions:} \quad \psi[{}^n\mathbf{x}_0, {}^n\mathbf{x}_f, {}^nt_0, {}^nt_f] = 0 \quad (4.1.5)$$

where:

N is the number of flights

N_o is the number of airspace objects that must be avoided

$D_x, D_u, D_o \in \mathbb{N}$ are dimensionalities of state, control input and object vectors respectively

$t, {}^nt_0, {}^nt_f \in \mathbb{R}$ are the current, initial, final times of flight n respectively

${}^n\mathbf{x} \equiv {}^n\mathbf{x}(t) \equiv \{{}^nx_1, {}^nx_2, \dots, {}^nx_{D_x}\} \in \mathbb{R}^{D_x}$ is the current state vector of each flight n

${}^n\mathbf{u} \equiv {}^n\mathbf{u}(t) \equiv \{{}^nu_1, {}^nu_2, \dots, {}^nu_{D_u}\} \in \mathbb{R}^{D_u}$ is the current control input vector of each flight n

${}^n\mathbf{x}_0 \equiv {}^n\mathbf{x}({}^nt_0) \in \mathbb{R}^{D_x}$ is the initial state vector of the flight n at take-off

${}^n\mathbf{x}_f \equiv {}^n\mathbf{x}({}^nt_f) \in \mathbb{R}^{D_x}$ is the final state vector of the flight n at landing

$\Phi: \mathbb{R}^{(2D_x+2)} \rightarrow \mathbb{R}^+$ is a positive scalar terminal-state weighting function

$\mathcal{L}: \mathbb{R}^{(D_x+D_u+1)} \rightarrow \mathbb{R}^+$ is a positive scalar intermediate state and input weighting function

$\mathbf{a}: \mathbb{R}^{(D_x+D_u+1)} \rightarrow \mathbb{R}^{D_x}$ maps the control input and current state vectors to the state derivative

$b: \mathbb{R}^{(ND_x+N_oD_o+D_u+1)} \rightarrow \mathbb{R}$ is a scalar intermediate state and input weighting function

$\mathbb{O} = \{{}^0\mathbf{o}, {}^1\mathbf{o}, \dots, {}^{N_o}\mathbf{o}\}$ is the set of all airspace object current locations, shapes and sizes

$\mathbb{M} = \{{}^0\mathbf{x}, {}^1\mathbf{x}, \dots, {}^N\mathbf{x}\}$ is the set of all other flight states

$\psi: \mathbb{R}^{2D_x+2} \rightarrow \mathbb{R}$ is a scalar terminal-state weighting function

Pontryagin's minimum principle (PMP) provides the necessary (but insufficient) conditions to achieve optimality [4.2]. However, solving such optimal control problems analytically is an exceedingly difficult task that has eluded mathematicians for all but the simplest problems, for hundreds of years.

So it is generally preferable to transcribe them as generic Non-Linear Programming (NLP) problems that can be approximated numerically on a digital computer. This is done by discretizing the state ${}^n\mathbf{x}(t)$ and control inputs ${}^n\mathbf{u}(t)$, and representing them as vectors of points taken at distinct intervals along the time axis [4.3], with a uniform time step h such that $t = kh$ where the index $k = 0, \dots, K \in \mathbb{N}$.

$$\begin{aligned} {}^n\mathbf{x}(t) & \xrightarrow{\text{discretised}} {}^n\mathbf{x}(kh) \xrightarrow{\text{represented as}} {}^n\mathbf{x}_k \\ {}^n\mathbf{u}(t) & \xrightarrow{\text{discretised}} {}^n\mathbf{u}(kh) \xrightarrow{\text{represented as}} {}^n\mathbf{u}_k \end{aligned} \quad (4.1.6)$$

where: K is the number of distinct time intervals making up the discretized state trajectory

Therefore, following the same logic, the single flight problem can be restated as an NLP (in Bolza form) as the minimization of the approximate cost ${}^n J$:

$$\begin{array}{l} \text{Bolza} \\ \text{Cost:} \end{array} \quad {}^n J^* = \min_{{}^n \mathbf{x}_k, {}^n \mathbf{u}_k} {}^n J \approx \min_{{}^n \mathbf{x}_k, {}^n \mathbf{u}_k} \left\{ \Phi[{}^n \mathbf{x}_0, {}^n \mathbf{x}_K] + \sum_{k=0}^{K-1} L[{}^n \mathbf{x}_k, {}^n \mathbf{u}_k, k] \right\} \quad (4.1.7)$$

Subject to:

$$- \text{ State transition constraints:} \quad {}^n \mathbf{x}_{k+1} = A[{}^n \mathbf{x}_k, {}^n \mathbf{u}_k, k] \quad (4.1.8)$$

$$- \text{ Trajectory constraints:} \quad B[\mathbb{M}, \mathbb{O}, {}^n \mathbf{u}_k, k] \leq 0 \quad (4.1.9)$$

$$- \text{ Boundary conditions:} \quad \Psi[{}^n \mathbf{x}_0, {}^n \mathbf{x}_K] = 0 \quad (4.1.10)$$

where: ${}^n \mathbf{x}_0, {}^n \mathbf{x}_K \in \mathbb{R}^{D_x}$ are the initial, final state vectors of the flight n respectively
 ${}^n \mathbf{x}_k \in \mathbb{R}^{D_x}$ is the state vectors, at step k , of each flight n respectively
 ${}^n \mathbf{u}_k \in \mathbb{R}^{D_u}$ is the control input vectors, at step k , of each flight n respectively
 $\Phi: \mathbb{R}^{2D_x} \rightarrow \mathbb{R}^+$ is a positive scalar terminal-state weighting function
 $L: \mathbb{R}^{(D_x + D_u + 1)} \rightarrow \mathbb{R}^+$ is a positive scalar intermediate state and input weighting function
 $A: \mathbb{R}^{(D_x + D_u + 1)} \rightarrow \mathbb{R}^{(D_x + D_u + 1)}$ maps the control input and state vectors to the next state
 $B: \mathbb{R}^{(ND_x + NO_{D_u} + D_u + 1)} \rightarrow \mathbb{R}$ is a scalar intermediate state and input weighting function
 $\Psi: \mathbb{R}^{2D_x} \rightarrow \mathbb{R}$ is a scalar terminal-state weighting function

In the discrete case, the Karush-Kuhn-Tucker (KKT) conditions provide the necessary (but insufficient) conditions to achieve optimality [4.4], and will in fact converge to the PMP, when the time step h is taken to the limit of 0 [4.5].

4.1.2 Combined Multiple Flight Problem

We have already discussed in previous chapters that it would be of little value if we were to optimise some trajectories at the expense of others. Flight trajectories can and do interact. However, an optimal solution in CATM is the set of feasible trajectories that results in the minimisation of some arbitrary cost functional that takes into account delays, fuel burn, emissions and any number of other objectives – *across the whole collection of flights*. Balanced collective optimality is the goal. One way of doing this, is to reorient the problem to express the overall system cost J_N as the sum of costs incurred by all flights, as shown below in eq 4.1.11:

$$\text{Thus:} \quad J_N^* = \sum_{n=1}^N {}^n J^* = \min J_N = \min \sum_{n=1}^N {}^n J \quad (4.1.11)$$

$$\text{and} \quad J_N^* = \min_{{}^n \mathbf{x}_k, {}^n \mathbf{u}_k} \left\{ \sum_{n=1}^N \Phi[{}^n \mathbf{x}_0, {}^n \mathbf{x}_{nK}] + \sum_{n=1}^N \sum_{k=0}^{nK-1} L[{}^n \mathbf{x}_k, {}^n \mathbf{u}_k, k] \right\} \quad (4.1.12)$$

where: ${}^n K$ is the number of distinct time intervals making up flight n relative to itself.

However, once we augment the problem to include each flight into one cost measure, another difficulty arises with separate flights having different terminal constraints. CATM in its entirety is an ongoing, never ending process. New flights are added and removed from the problem domain as time progresses with no specific start or end to the continuum. This turns it into a variable-size, infinite horizon problem, even though a small subset of the potential flights are actually flying at any one time.

$$\text{CATM: } J_{\infty}^* = \min_{n_{\mathbf{x}_k}, n_{\mathbf{u}_k}} \left\{ \sum_{n=1}^{\infty} \Phi[n_{\mathbf{x}_0}, n_{\mathbf{x}_{K_F}}] + \sum_{n=1}^{\infty} \sum_{k=n_{K_I}}^{n_{K_F}-1} L[n_{\mathbf{x}_k}, n_{\mathbf{u}_k}, k] \right\} \quad (4.1.13)$$

where: n becomes a unique unbounded incremental index of every flight ever flown
 n_{K_I} is the absolute time index at which flight n commences
 n_{K_F} is the absolute time index at which flight n ends

This problem, as expressed in eq 4.1.13, does not have a solution in causal time. It is in fact a *non-causal problem*, which is to say that current decisions and/or evaluations of the cost functional depend on future events and information which has yet to become available when new flight plans are registered by the airlines.

4.1.3 Receding Horizon Problem

In order to render the problem causal, a limitation must be imposed on the information required to solve the problem. A limited look-ahead time horizon is established. Thus a limited number of flights are considered at any one time starting with the one nearest to completion N_0 , moving onto all N_{FA} active flights, one transition flight, N_{FS} queued flights and finally ending with N_{FL} long term scheduled flights. Thus, the total horizon N_F represents the sum of flights that the system takes into account into the future. Note that N_{FA} , N_{FS} , N_{FL} and N_F are all variable quantities which depend on many factors such a travel demand and aircraft availability. This makes the horizon depth variable.

$$\text{CATM: } J_{N_F}^* = \min_{n_{\mathbf{x}_k}, n_{\mathbf{u}_k}} \left\{ \sum_{n=N_0}^{N_0+N_F} \Phi[n_{\mathbf{x}_0}, n_{\mathbf{x}_{K_F}}] + \sum_{n=N_0}^{N_0+N_F} \sum_{k=n_{K_I}}^{n_{K_F}-1} L[n_{\mathbf{x}_k}, n_{\mathbf{u}_k}, k] \right\} \quad (4.1.13)$$

where: N_0 is index of the flight nearest to completion
 N_F is the total size of the horizon being considered by the solver.
 $N_F = (N_{FL} + N_{FS} + N_{FA} + 1)$

As flights are progressively completed, N_0 is incremented and the horizon recedes by one flight at a time, to ignore completed flights and to encompass new flights further into the future (**Figure 4.1.1**). This is reminiscent of Model Predictive Control and should result in a good approximation of the desired CATM solution if the chosen horizon is deep enough.

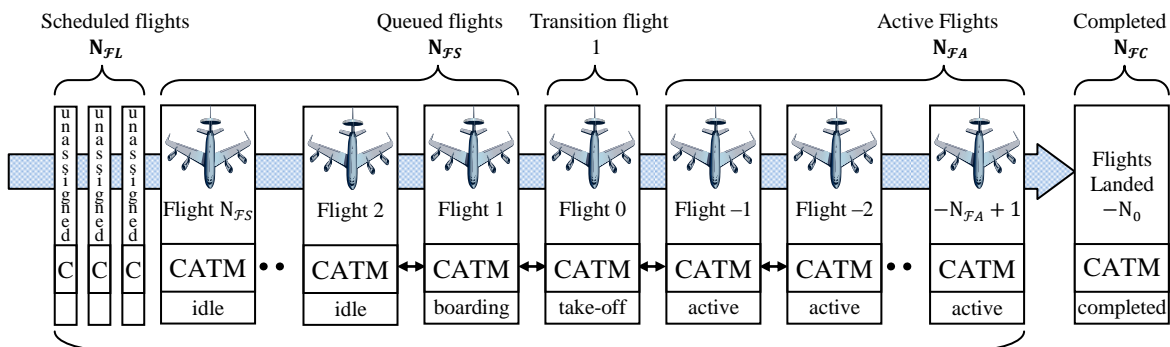


Figure 4.1.1: The CATM Receding Horizon of $N_F = (N_{FL} + N_{FS} + N_{FA} + 1)$ Flights

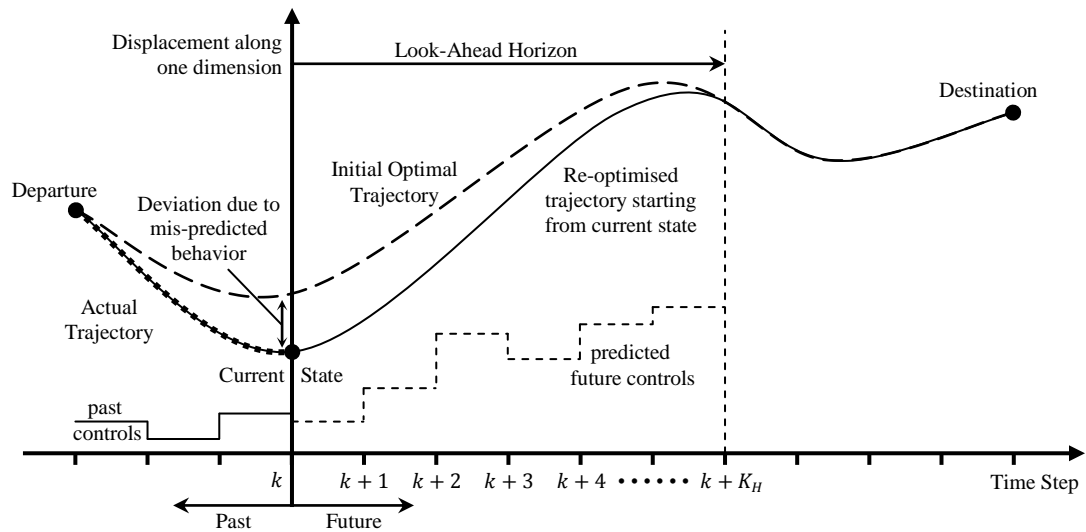


Figure 4.1.2: Receding Horizon Control for a Single Flight, with Horizon = K_H

In **Figure 4.1.2**, the concept of receding horizon control is depicted for a single aircraft. With an aircraft model and an airspace model, we can use the known departure and destination points to generate an optimal trajectory and a set of aircraft controls that minimises some cost metric. The aircraft is then tasked with following this reference trajectory by executing the given controls. However, two things happen that cause some deviation over time. The aircraft is not perfectly represented by the model. Thus the actual motion of the vehicle will drift from the expected trajectory. Similarly, the airspace is also not perfectly represented by its model and airspace conditions will change to depart from predicted behaviour. Thus the optimal reference trajectory will be invalidated and will require renewal over time. Unless corrected, the net result is that the originally estimated aircraft controls will take the aircraft away from the true optimal trajectory.

For the single aircraft scenario (**Figure 4.1.2**), these deviations from optimality are addressed at every time-step, one step at a time, by recalculating a new optimal trajectory starting from the current location while taking into account a limited look-ahead horizon of time steps (in this case K_H) over which the aircraft will be brought back on track with the most recently estimated optimal path.

Thus at every time step, the receding horizon controller takes note of the current state, and using the given model and best available knowledge of current airspace conditions, drafts a new optimised trajectory between the *current* state and the *desired state* at K_H . The aircraft is then allowed to execute this updated plan over the next time step *only*, and the process is repeated thereafter for every subsequent time step.

A similar situation arises in the multi-aircraft scenario shown for comparison in **Figure 4.1.3**. However, in this case the Air Traffic System state has no known departure or destination, because it consists of the superposition of many concurrent flights, which are being regenerated continuously. However, an optimal system state trajectory can still be generated between the known present state and some future desired state. The look-ahead horizon (in this case N_F) is expressed in flights rather than in time steps. Every step in the horizontal axis of **Figure 4.1.3** represents a flight transition, when one flight lands and another takes off reusing the same aircraft.

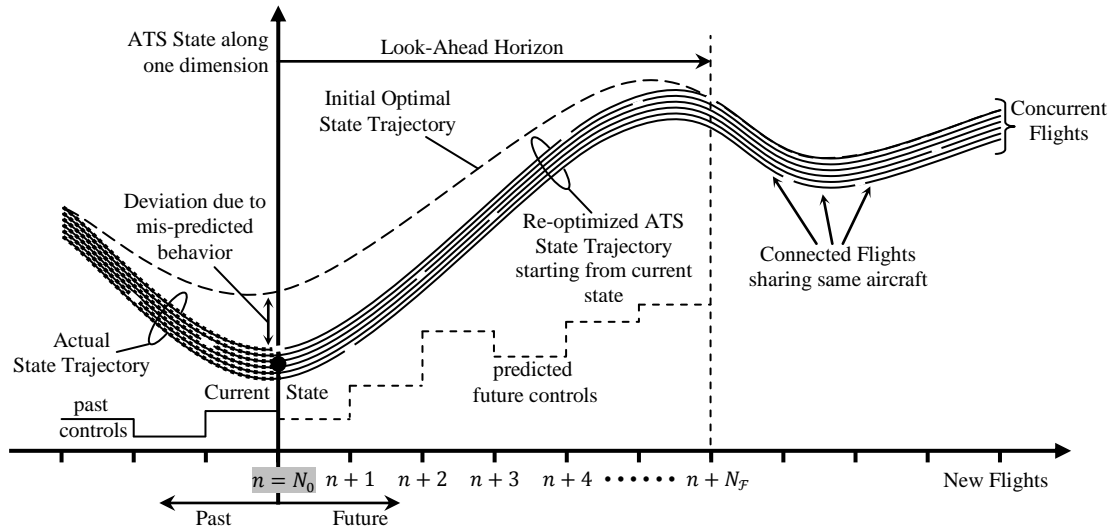


Figure 4.1.3: Receding Horizon Control for entire ATS, with Horizon = N_F

In this simplified scenario we are assuming that there are 6 aircraft in the system executing a maximum of 6 concurrent flights at any point in time, and thus the collective ATS system state is being represented by the multiline curve shown above. The same ideas are readily extended to larger numbers and variable traffic.

4.1.4 The CATM Architecture

In Chapter 3 we discussed a sketch model architecture of the contemporary ATM system. It is now time to describe the corresponding architecture proposed for CATM. In **Figure 4.1.4**, we show how the CATM ATS can be modelled using system theoretic principles. The ATS is depicted as a hierarchical set of nested feedback control systems.

We recall that equations 4.1.14 represent the general nonlinear time variant differential equation description of an aircraft, with flight controls $\mathbf{u}(t)$ as inputs, $\mathbf{w}(t)$ as additive external wind disturbances and aircraft motion $\mathbf{y}(t)$ as output.

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) + \mathbf{w}(t) \\ \mathbf{y}(t) &= \mathbf{g}(\mathbf{x}(t), \mathbf{u}(t), t)\end{aligned}\quad (4.1.14)$$

Here we assume that an auto pilot is given the responsibility of maintaining the aircraft in level flight along a course set by the flight guidance computer. The autopilot performs accurate coordinated manoeuvres as necessary to minimise the error between the set and current course. The flight guidance computer forms an overarching control loop that ensures that the course taken matches the trajectory assigned to the aircraft by the CATM computer. On board navigation instruments provide the necessary feedback.

The receding horizon optimiser shown sitting in every aircraft in **Figure 4.1.4** is the local representative of a much larger receding horizon optimisation system, which collectively has full situational awareness of the entire air traffic system. It collects information from the current aircraft and shares it with the rest. In the meantime it collects information for the rest of the aircraft, and on combining it with local data, is able to generate a new optimised trajectory that serves as the updated flight reference. This occurs at every time step as discussed earlier.

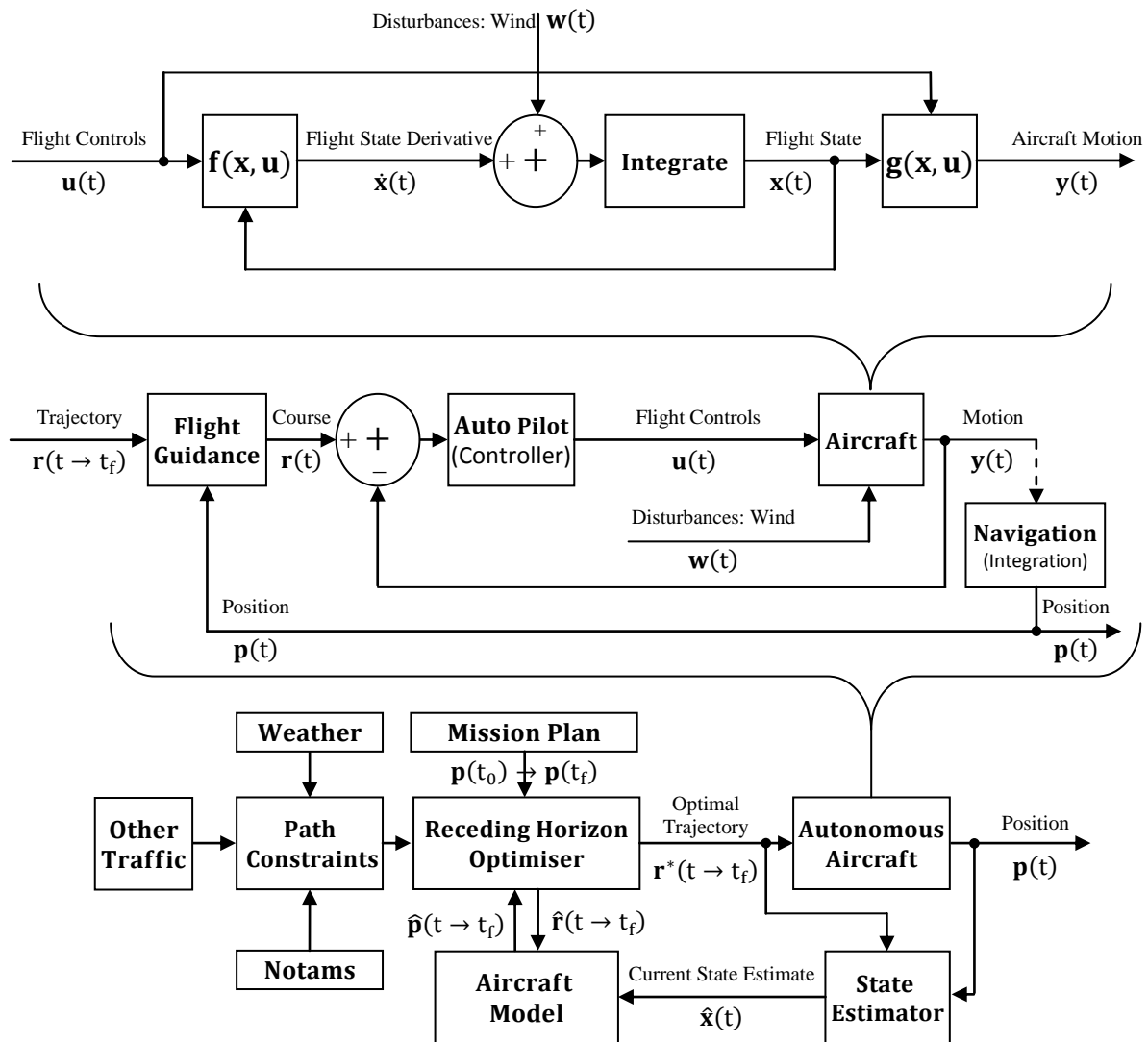


Figure 4.1.4: The CATM System Hierarchy within a Single Aircraft.

4.1.5 Searching the Control Space vs Output Space

Optimal control, is driven by the desire to generate the sequence of input controls that will take the system through the required optimal trajectory. So it is not surprising that the search space includes all the input variables as well as the output variables, as is the case with equation 4.1.13. The dynamic constraints (Equations 4.1.3 and 4.1.8) ensure that the optimised inputs are consistent with the required optimal output. It is essential that this is enforced to guarantee that realistic controls are generated while observing all constraints.

However, one might wonder whether this added complexity is at all necessary in cases of simple systems, where there is a known one to one mapping between input and output. In fact it transpires that for a certain class of systems this indeed results in unnecessary duplication of work, and only serves to inflate the search space to no real benefit. This would reduce the performance of the optimisation algorithms.

So what if the optimiser were to search just the input space for an optimum set of controls? Given known initial conditions, this could then be applied to the system's dynamic model and the corresponding output would be generated trivially. However a problem immediately arises if there are any constraints that must be observed at the output. These would presumably have to be transformed into corresponding constraints at the input, and this might prove to be difficult in some cases involving complicated constraints. The cost might also depend on output behaviour.

Such a scheme would also cause difficulties in ill-behaved non-linear systems. It is quite possible that non-linear dynamics would connect a relatively small input space to a significantly larger output space, making the output highly sensitive to small optimisation errors present at the input. So what if the reverse were attempted; that of namely searching the output space and then calculating the corresponding input using some form of inverse dynamics?

4.1.6 Dynamic Inversion and Differential flatness

Roughly speaking, dynamic inversion is when a system's inputs and state are expressed in terms of its outputs. This is only possible in so called differentially flat systems which were first described by Fliess at al. [4.6]. This property has been demonstrated in a number of systems such as Vertical Take-off Aircraft (VTOL) [4.7], certain rotary wing aircraft [4.8] and quadrotors [4.9]. However, except for single input systems, so far there seems to be no unifying theory that allows systems to be classified as differentially flat [4.10]. Each case has to be analyzed on its own merits.

Unfortunately, typical civil aviation, fixed wing aircraft models are a notable exception. However, over a narrow flight-regime corresponding to coordinated flight conditions, simplified models suitable for ATM purposes have been demonstrated to be differentially flat [4.11]. Once this has been established, we may continue all our optimisation in the output domain with the peace of mind of being able to impose the dynamic constraints down the line using the appropriate models.

4.2 CATM: THE MATHEMATICAL NATURE OF THE SEARCH SPACE

We will next be considering the topological nature of the CATM optimisation problem. This will serve as a guide to selecting the most appropriate algorithms down the line. From this point onwards we will also assume that all the optimisation is taking place at the output. Optimisation will therefore be conducted on the trajectories themselves and appropriate aircraft models will have to be used to allow this to be translated into the corresponding control inputs.

4.2.1 The Global Optimisation Problem (Non-Convexity)

In a non-convex optimisation problem, there are an unknown number of regional *local optima*, only one of which is the overall *global optimum*. The search space is said to be *multimodal* and by demonstrating the existence of more than one regional minimum, it is enough to establish this fact. *Efficient* optimisation with such a search space is in general an unresolved mathematical problem, for which non-deterministic stochastic searches apply. Frequently, these take the form of metaheuristics such as evolutionary algorithms or swarm intelligence [4.12]. Such nature-inspired algorithms

have been successful in many domains [4.13].

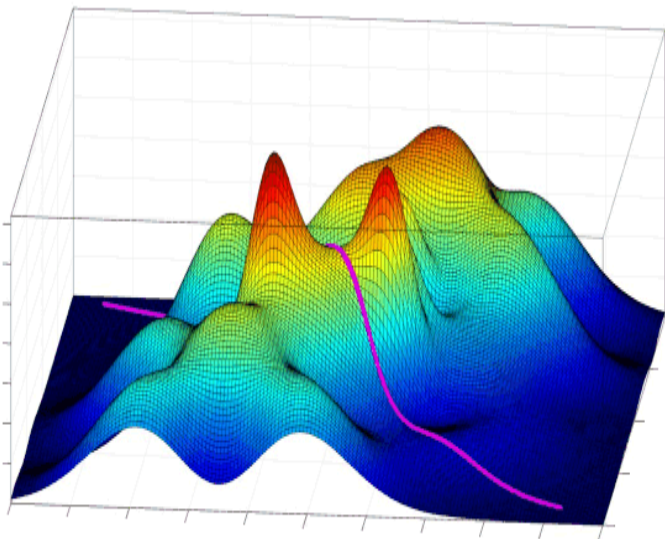


Figure 4.2.1: Trajectories Trapped in Local Minima

Classical gradient optimisation techniques would get trapped in the first local optimum they encounter as they move from their initial seed solution. In the context of trajectory optimisation, trajectory search can be viewed as getting trapped between regions which evaluate to higher cost functionals. **Figure 4.2.1** depicts this scenario in the simplified case of finding the shortest route through a 2D cost map.

However, by Bellman's principle of optimality, any segment of a globally optimal sequence of events, is necessarily optimal too, and the overall cost is the summation of the sub-costs of each problem segment.

Most local optima arise when parts of the optimal sequence are replaced with suboptimal segments, such as the various suboptimal ways of routing through an obstacle field. **Figure 4.2.2 (a)** highlights one particular scenario showing a few of the many possible 2D trajectories of reaching B from A, for a single trajectory through such a fixed obstacle field. In this simple context, optimality is defined as the shortest path that clears all obstacles with adequate separation, and proximity to an obstacle can be interpreted as raising the cost of a trajectory. Obstacles could in general also include other aircraft. However, this is ignored in this example for better clarity.

Each binary decision to route above or below any obstacle creates an opportunity to lengthen or shorten the path. Here, T_6 is the optimal path. So with reference to this global optimum, each deviation from this path results in a local minimum bounded by obstacles.

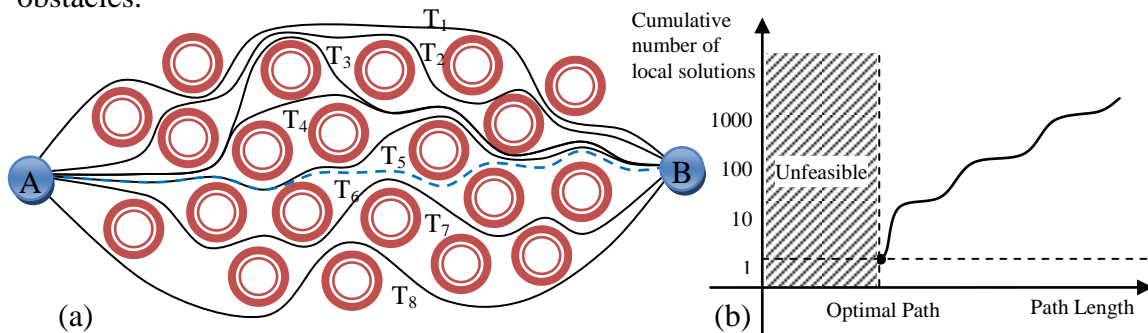


Figure 4.2.2: Some possible locally optimal trajectories through a fixed obstacle field

Each possible trajectory finds itself in a well of local optimality (and indeed local feasibility). T_5 deviates by one obstacle and is slightly longer than T_6 . However, there are many combinations of single obstacle deviations that will lead to a slight elongation. The set of solutions with two obstacle deviations is even larger and so forth. These are in general even less optimal because the effects of deviations from the optimum are always additive.

Now in a large highly interconnected problem such as ATM, there are very many permutations in which deviations from the optimum can occur over the whole airspace, resulting in equally many local optima. However, from the point of view of optimality, by the law of large numbers, one would expect these local optima to follow some stochastic distribution that is fairly smooth as shown in **Figure 4.2.2 (b)**. This means that it is likely that there exist a large number of local optima that approach the global optimum to a sufficient degree to make reaching *the* global optimum rather irrelevant. This is an advantageous property that can be exploited to increase the utility of a would-be stochastic-search-based global optimiser. While there are no guarantees of actually striking the quasi-convex well containing the global optimum, the probability of reaching solutions that are fairly close in quality, can be made arbitrarily high.

4.2.2 The Local Optimisation Problem (Local-Convexity)

It was established in the previous section that in CATM, global convexity is generally not possible due to the numerous path constraints. It is however desirable to exploit any quasi-convexity at a local level. This would greatly improve convergence rates if we could utilise gradient information to point at the stationery points in the cost functional. With this in mind, cost functionals are best designed to be locally differentiable and well behaved. This is to say that in the immediate neighbourhood of any local (or global) optimal solution, evaluations of the cost functional must be continuous and smooth. Therefore small changes to the trajectories, result in proportionally small changes in cost metric. This is readily achievable in practice, as we will see, and with the right formulation, the local problem can lend itself for efficient processing using many free (eg: PSOPT [4.14]) or commercial (eg: SNOPT [4.15]) Non-Linear Programming (NLP) fast numerical solvers.

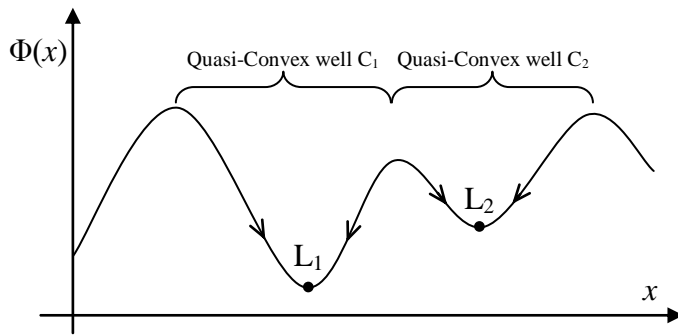


Figure 4.2.3: Local Convexity assists convergence

In **Figure 4.2.3** we depict the scenario of a multimodal cost function, $\Phi(x)$, evaluated along one dimension, x . The function has two local minima L_1 and L_2 that occur within quasi-convex wells C_1 and C_2 respectively. Within the confines of each quasi-convex well, standard gradient-based numerical optimisation techniques, like

Newton methods, can be applied to identify the minima very efficiently, by fully utilizing the substantial quantity of directional information contained in the shape of each well, which effectively points towards the local solutions.

However, in order to guarantee local convergence, these algorithms require an approximate seed or starting point that lies within the confines of the quasi-convex well. Such information must be obtained using some other means. This suggests the requirement of an overarching search algorithm that scans the wider search space, to identify the approximate locations of quasi-convex wells. This could be used to generate approximate seed solutions that are then fed into a subsidiary gradient search algorithm, which in turn speeds up the “last mile” of convergence, allowing the optimiser to accurately and rapidly compare the true depth of quasi-convex wells for their global optimality potential. Such a hierarchical exploratory approach to optimisation strikes an ideal combination of diversification with intensification.

4.2.3 The Disconnected Feasible Search Space Problem

Somewhat related to non-convexity is the mathematical notion of topological connectedness [4.16] of the search space. Connectedness is an important attribute when searching spaces. The presence of non-feasible areas in the middle of the search space creates a situation where regions of feasible solutions are separated by areas which are not. Discontinuities in feasibility create insurmountable barriers for algorithms that normally rely on gradual iterative improvement, such a gradient descent. It also causes problems with metaheuristic algorithms that rely on population-based candidate solution diversity to scan the search space. An exclusion of population members from these areas would result in a reduction of population diversity.

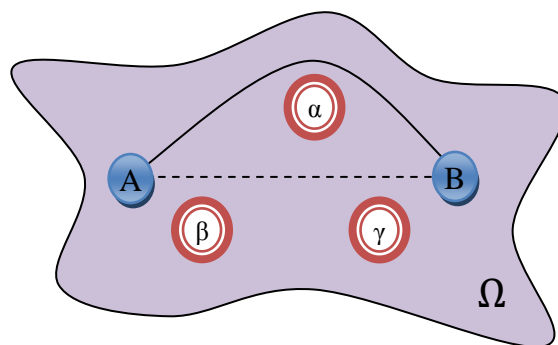


Figure 4.2.4: Disconnected Trajectory Search Spaces

Figure 4.2.4, shows a region Ω in 2D Euclidean space \mathbb{R}^2 . The region Ω is non-simply connected due to the presence of three non traversable objects α , β and γ . This is to say that it is possible to draw a loop within region Ω , that would be unable to be gradually contracted to a point, without trapping non-feasible space.

However, from a trajectory synthesis perspective, Ω is actually a disconnected search space in \mathbb{R}^2 . This is because there exists no way in which the trajectory shown connecting A to B can ever be gradually transformed into the straight line (shown dashed) connecting A to B. In \mathbb{R}^3 and above it would again become non-simply connected, as long as α , β and γ are of a finite size smaller than Ω .

Topologically, the air traffic system is confined to a thin spherical shell surrounding planet earth, physically constrained by the spherical ground below and performance constrained by the rarefied atmosphere at the top. Thus, it is entirely conceivable that large weather systems (like hurricanes), significant volcanic eruptions, conflict zones and other airspace restrictions might affect all the flight levels in at least some areas. This again re-creates a similar scenario to that shown in **Figure 4.2.4**. It is also possible that local optima might be surrounded by unfeasible space.

4.2.4 The Dynamic Optimisation Problem

If the shape of the feasible search space depends on the presence and location of moving weather cells, sporadic volcanic eruptions and the location of other flying aircraft (some of which might be CATM-unequipped and even unresponsive) then the topology of the said feasible airspace is time varying. This takes CATM to join a new class of Problems called Dynamic Optimisation that have become a hotly researched topic in recent years [4.17]. With reference to **Figure 4.2.5**, this can be characterised by a number of different behaviours including:

- 1) Changes in Relative Regional Optimality (C_1 vs C_2)
- 2) Spatial Shifting of Local Convexity (C_2)
- 3) Resizing, Obliteration and Creation of Local Convexity (C_3)
- 4) Changes in the Number of Variables, Constraints and Objectives
- 5) Spatial Shifting of Feasibility
- 6) Resizing, Obliteration and Creation of New Feasible Spaces

Figure 4.2.5 shows three common scenarios in a dynamically changing search space as a function of time. In a minimisation problem, quasi-convex well C_1 is getting increasingly less optimal and in fact cedes the status global optimality to C_2 which is meanwhile shifting as well as deepening. Quasi-convex well C_3 , was initially non-existent but developed over time.

A trivial but clearly inefficient methodology to deal with such problems involves periodic restarts of the optimisation process every time a change is detected [4.18]. However, a far better approach would be to carry-over as much information as possible from previous iterations or generations to track the evolution of local and global optima, which are typically gradual in the ATM environment. A locally focussed algorithm would be well suited to efficiently track and reevaluate changes in local optima, by using previous results to seed subsequent iterations. However, a global perspective must also be maintained to compare the changing relative optimality of local wells and in order to discover any newly formed quasi-convex wells. Changes in feasibility can be handled in similar ways.

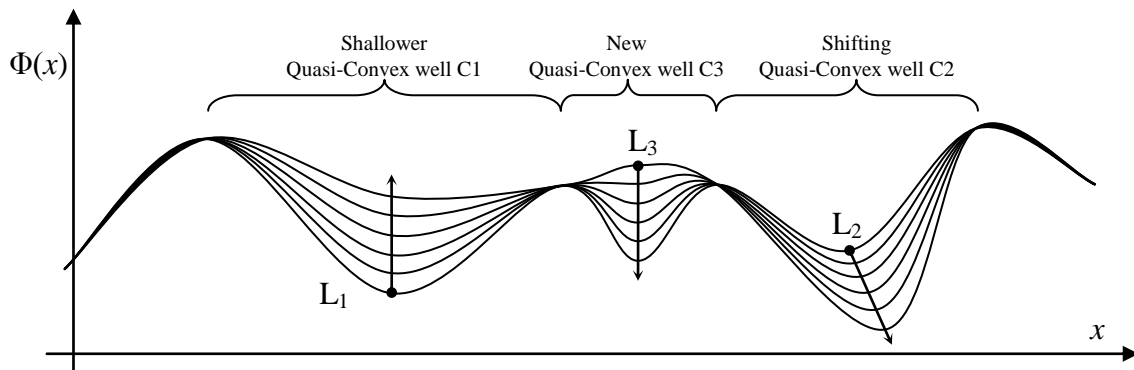


Figure 4.2.5: Dynamic Changes to Local Quasi-Convexity

Changes in problem size are also characteristic of ATM and the efficient optimiser must also be flexible enough to carry-over information across problem instances with increased or decreased dimensionality. Most global optimisers are not designed to handle this.

4.2.5 The Time-Linked Dynamic Optimisation Problem

Dynamic optimisation deals with systems whose optimality profile changes over time. *Time-linked* dynamic optimisation deals with systems where current decisions taken following previously found solutions, affect the system's future evolution of its optimality profile [4.17]. If weather were the only source of dynamic behaviour, then arguably CATM would not exhibit any time linkage. However, this is not so, because clearly any trajectories generated by the optimiser, and executed by the aircraft, are bound to alter the very constraints that characterise the future progression of the problem.

This is an important consideration because it affects the choice of suitable algorithms. For offline optimisation to be possible, the system must simulate the effects of the optimisation-based decisions and sweep forward across the problem. In fact this is related to the receding horizon discussion that was made earlier.

4.2.6 The Constrained Optimisation Problem

In the case of local optimisation, constraints are usually expressed algebraically and reduce the size of the problem by limiting the feasible search space. There are efficient implementations of NLP solvers [4.15] that will perform very well, and indeed better, in the presence of numerous active constraints [4.19].

However, in the case of metaheuristic techniques, as often utilised for global search, the method of handling constraints is not very straight forward [4.20], especially in the case of population based methods such as swarm intelligence and genetic algorithms, which utilise pools or swarms of evolving candidate solutions to better explore the search space.

Efficiency becomes a problem in these algorithms, which were originally devised for unconstrained optimisation [4.21]. The partly stochastic processes for generating new members for every subsequent population generation does not automatically

guarantee constraint-compliance. Therefore simple trial and error approaches (eg: Monte Carlo methods), would be computationally expensive, as they would imply testing every new member against the constraints. It is evident that if there are many constraints (as in CATM), performance would be adversely affected as it could take numerous attempts in order to generate a single feasible candidate, while discarding the rest. This simple technique is aptly called the *death penalty* method [4.22] and brings with it significant repercussions on population diversity. The convergence characteristics of the algorithm will suffer as a result of failing to capitalise on the fact that unfeasible candidate solutions can often contribute valuable information to the genetic pool or swarm.

A number of alternative paradigms have been proposed in the literature over the years and a recent survey on the subject [4.23] classifies them in the following manner:

1. *Penalty Functions* – These attempt to attribute a high cost to the unfeasible areas of the search space in a bid to dissuade searches in these areas. The penalty functions are often designed to provide a measure of *transgression* that gradually increases as the search moves further into unfeasible spaces. This ensures that the cost functional remains smooth and differentiable, while providing directional information away from the constraint boundaries. Many variants are possible such as static, dynamic and adaptive penalties, and the suitability of each is problem dependant. In practice, penalty functions are applicable to a wide variety of problem types and their results are either used to modulate the fitness function [4.24] or are simply added to the list of optimisation objectives [4.25][4.24].
2. *Decoders* – These attempt to remap an irregularly shaped feasible space to a regularly (typically rectangular) shaped version, which would then lend itself to being expressed as a set of simple inequalities [4.26]. Unfortunately, this is not always straight forward or even possible in many cases.
3. *Special Operators* – These include a broad range of problem-specific operations conducted on unfeasible candidate solutions in order to recover their feasibility or to tease information from unfeasible members back into the realm of feasibility [4.27]. This is a very interesting approach however it seems limited to very specific cases.
4. *Separation of Objective Function and Constraints* – These approaches split the optimisation procedure into consecutive phases. The first attempts to map out the feasible space by generating candidate solutions that merely meet the constraints. The second phase focuses on the optimality aspect. Alternatively a lexicographic approach may be taken whereby: the better of two feasible solution is always chosen, a feasible solution is always chosen over an unfeasible one, and between two unfeasible solutions, the one with the lowest level of transgression is chosen [4.28]. However, although still popular, this technique has been shown to be detrimental on population diversity in evolutionary and swarm based techniques.

In the context of CATM, the following objective functions (eq: 4.2.1 – 4.2.3) show how constraints on obstacle and aircraft interaction or trajectory length can be expressed as soft-penalty cost functions. The associated caveat is that the reduction in constraint complexity is hereby transferred to multiple objective complexity. So the suitability of this approach depends on how effectively multiple objectives can be handled.

Trajectory Length/Cost:
$${}^n J_L \approx \sum_{k=0}^{K-2} \|{}^n \mathbf{x}_{k+1} - {}^n \mathbf{x}_k\|^2 \tag{4.2.1}$$

Obstacle Interaction:
$${}^n J_O \approx \sum_{i=1}^{N_O} \sum_{k=0}^{K-1} \|{}^n \mathbf{x}_k - {}^i \mathbf{o}\|^2 \tag{4.2.2}$$

Other Aircraft Interaction:
$${}^n J_A \approx \sum_{m=1}^M \sum_{k=0}^{K-1} \|{}^n \mathbf{x}_k - {}^m \mathbf{x}_k\|^2 \tag{4.2.3}$$

where: ${}^n J_L, {}^n J_O, {}^n J_A$ are the respective costs from the point of view of the n^{th} aircraft.
 M, N_O, K are the numbers of other aircraft, obstacles and trajectory segments respectively.
 ${}^n \mathbf{x}_k, {}^i \mathbf{o}$ are the locations of n^{th} flight trajectory knots, and location of obstacles respectively.

4.2.7 The Multi-objective Optimisation Problem

The classic method used to deal with multiple-objective optimisation problems (MOOP) is to construct a Pareto front of *non-dominated* solutions, each of which is optimal in its own right [4.29]. Population based methods are particularly suited for MOOPs. In 2002 Deb et al. developed one of the best known multi-objective adaptations of GAs called NSGA-II [4.30]. This algorithm uses a non dominated raking scheme to assign fitness to population members and in doing so maps out the Pareto front. It has become the de facto benchmark for the field and continues to provide competitive results [4.31].

A convex Pareto front of a bi-objective minimisation problem is shown in **Figure 4.2.6**. Each solution on the front represents a unique, but optimal trade-off between the competing objectives. Solutions below this front would be desirable but unfeasible, given the constraints imposed on the system. In a continuously-variable problem there are infinitely many Pareto solutions and hence, human judgment is usually invoked to decide which trade-off suits this purpose or design criteria. Although convex fronts are common, it is worth noting that real world problems can have various other shapes, including disconnected ones and multimodal Pareto fronts.

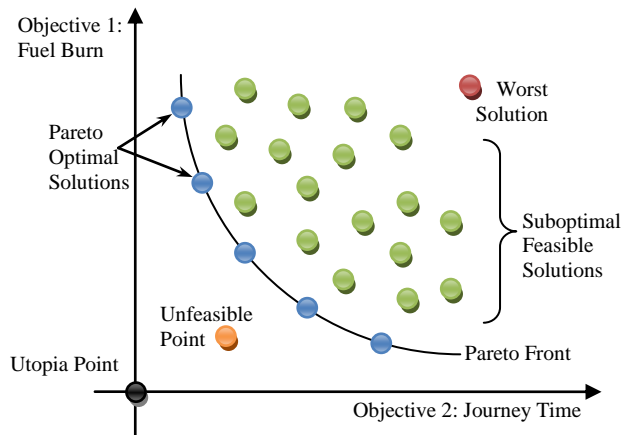
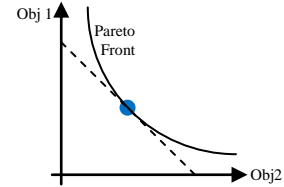


Figure 4.2.6: A Typical Bi-objective Pareto Front

Although ATM is a multi-objective problem, constructing Pareto fronts along the many potential objectives is not very helpful in CATM. The real time nature of the application means there is no time for human intervention to consider trade-offs between objectives in an operational environment. For this reason, the relative importance of each objective is pre-determined at the design stage and is then weighted relative to the rest with a vector, w_i using one of various norms such as Equations 4.2.4 to 4.2.6.

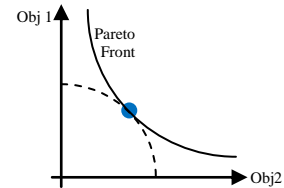
Linear Norm:
(Manhattan Norm)

$$\|x\|_1 = \sum_{i=1}^n |x_i| \quad (4.2.4)$$



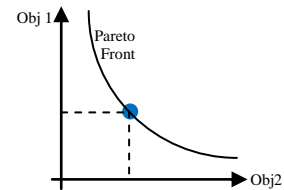
Quadratic Norm:

$$\|x\|_2 = \sqrt{\sum_{i=1}^n |x_i|^2} \quad (4.2.5)$$



Infinity Norm:
(Maximum Norm)

$$\|x\|_\infty = \sqrt[p]{\sum_{i=1}^n |x_i|^p} \Big|_{p \rightarrow \infty} = \max(|x_1|, |x_2| \dots |x_n|) \quad (4.2.6)$$



where: $x_i \in \mathbb{R}$ is the i^{th} objective of the problem

This effectively allows the multi-objective problem to be scalarised and converted back into a single objective problem [4.29]. Geometrically, this is equivalent to the intersection of the Pareto front with a hypercurve, the shape of which depends on the scalarisation function used. The difficulty here lies in weighting these norms to assert relative importance as this seems to be a highly subjective exercise requiring expert input and much tweaking [4.32].

4.2.8 The System Wide Objective Function

The CATM problem involves many agents, the behaviour of which can be individually optimised. However, performance must be measured systemically if holistic performance benefits are to be obtained. This can be achieved by constructing a system-wide objective function. Excess distance travelled is by far the most indicative metric of inefficiency, with which both flying time and fuel consumption are directly related. With this in mind we adapt a well known ATM systemic efficiency metric suggested by Krozel at al [4.33] to give percentage systemic inefficiency (SI) as in equation 4.2.7:

$$SI = \frac{100}{N_{\mathcal{F}}} \sum_{n=1}^{N_{\mathcal{F}}} \left(\frac{n_{J_L}}{n_{*J_L}} - 1 \right) \quad (4.2.7)$$

where: $SI = 0\%$ for perfect direct trajectories for all flights.

4.2.9 The Non-Separable High Dimensionality Optimisation Problem

A particularly challenging aspect of CATM is the sheer size of the search space, and this is the result of high dimensionality coupled with the large dynamic range of each variable. The size of the state vector of CATM is equivalent to the sum total of the number of states of all the agents in the system, and that includes all current and future flights in the optimisation horizon as given in equation 4.2.8.

$$|\mathbf{X}_{CATM}| = |{}^n\mathbf{x}_0| \cdot |\mathcal{F}_L \cup \mathcal{F}_S \cup \mathcal{F}_A| \quad (4.2.8)$$

Given the laborious methods required to scan such non-convex search spaces, the high dimensionality results in excessively slow convergence rates for the optimisation process. Ordinary dynamic programming is an impractical proposition as already discussed, while stochastic algorithms (if applied naively) would inevitably sample the search space too thinly to reach any meaningful conclusions in reasonable time.

4.2.9.1 Divide and Conquer Techniques

One classic way of addressing high dimensionality problems is through a *divide and conquer* methodology [4.34]. When the problem allows, the simplest method involves using a number of separate optimisers to handle a few variables at a time, thereby avoiding the combinatorial explosion that results from the multiplication across all dimensions. Consider the N dimensional reference dynamic programming problem, whereby each dimension is discretized into a K -point grid. The aggregate size of this discrete search problem S_1 is related exponentially to N as given by equation 4.2.9. However, if the problem can be divided into M sub-problems of N/M dimensions each, so the new total size of all M sub-problems is given by S_2 in equation 4.2.10, the resulting effort is very substantially smaller than attempting to solve S_1 directly.

$$S_1 = K^N \quad (4.2.9)$$

$$S_2 = M \cdot \left[K^{\frac{N}{M}} \right] \quad (4.2.10)$$

where:

$$S_2 \ll S_1$$

However, such a technique is only possible when the dimensions do not interact and can thus be considered *separable*. Unfortunately, this is seldom the case and CATM is almost certainly not among this class of problems. However, not all is lost because there are varying degrees of interaction between the variables defining the airspace. On closer inspection, it can be observed that some of the dimensions in the CATM search space are less likely to interact than others. Interaction seems to naturally cluster the search variables into limited groups. It should be obvious that consecutive waypoints of one aircraft are far more intertwined than those between different aircraft, and the interaction becomes far less important when considering flights over distant continents or flights that are months apart. Flight clustering was already discussed in the context of locality and network communications in previous chapters. However, it must also be seen as a means of partitioning the problem into manageable, limited dimensional portions involving a small number of highly interacting flights. Receding horizon optimisation can also be viewed as a means of limiting dimensionality.

In the context of metaheuristic algorithms, multi-population techniques have been proposed as one way of partitioning large problems [4.35]. One of the earliest attempts was described by Potter and De Jong, using a cooperative co-evolutionary genetic algorithm (CCGA-1) [4.36]. Here they use a genetic algorithm (GA) having a separate sub-population for each variable resulting in marked speciation and far less processing overhead than a standard GA. Cooperation between species was enforced by evaluating members of each population in conjunction with the best members from every other. Thus they concatenated their genetic material before undergoing fitness evaluation, giving equal credit to all components. As expected, the algorithm (CCGA-1) performed very well on problems with separable variables such as Rastrigin and Schwefel functions. However, it performed much worse than a standard GA on the well known Rosenbrock function that happens to exhibit high interaction between variables. This was improved somewhat in (CCGA-2) by altering the credit assignment aspect, by including vectors with randomised concatenation.

The principles of cooperation and co-evolution have also been extended to Particle Swarm Optimisation (PSO) [4.37], [4.38], which is a popular swarm intelligence optimisation technique [4.39]. It was observed that blind decomposition of interacting variables could result in the spurious creation of pseudo-minima [4.37]. Alternative means of high dimensionality decomposition were thus proposed, using random groupings [4.35] and adaptive weights [4.38] and both combined, in cases where a-priori knowledge of interaction was unavailable. However, when such knowledge is available, conventional wisdom suggests grouping correlated variables into the same swarm [4.37].

All this seems to suggest that clustering in CATM should be the method of choice for decomposing the CATM problem into manageable portions. By assigning all members of one cluster to one swarm, and different swarms to different clusters, variable dependencies can be tackled effectively, while allowing for very significant complexity reduction. Demonstrating the efficacy of clustering in CATM will be one of the central themes of this thesis.

4.2.9.2 Multi-Resolution Programming

Another technique for tackling non-separable high dimensionality is through what is best termed as *multi-resolution programming* (MRP). This involves searching the problem space very sparsely with just a few “sample grid points” per dimension. When the most promising region is established, a denser grid is cast in just this neighbourhood along all dimensions, in the hope of finding a better result. Thus the second iteration involves no more variables than the first but a higher spatial resolution is achieved by instead concentrating the search on where the global optimum seems most likely. This process is repeated iteratively, progressively reducing the volume of the N dimensional space searched at each stage. The algorithm terminates when the degree of improvement per iteration levels off. Such an intensification strategy can be applied to all global search techniques, including genetic algorithms and swarm intelligence but is best exemplified by iterative dynamic programming (IDP) [4.41].

IDP was proposed in 1990 by Rein Luus [4.42] for application in the industrial chemistry sector, motivated by the fact that DP is so broadly applicable, but equally disappointingly impractical. In IDP, each of the N state dimensions of an aircraft is sampled with a uniform, but finite, discrete grid of resolution G , resulting in an array of $N \cdot G$ possible states at each time step p . This takes place for all P time stages that make up one trajectory, with the exception of the initial and final states, which are represented by just one immutable grid point as dictated by the required take-off and landing boundary conditions. Standard DP is used to select the optimum grid point at every stage which minimises the cost of the whole P stage trajectory. During subsequent iterations, the range of grid points is contracted by a constant factor γ , centred about the previously found trajectory. In [4.42], sampling the input control space rather than the output space, was suggested as a means to ensure feasibility of the output grid. However, in other cases problem reformulation was suggested as a way of avoiding having constraints at both input and output [4.43]. This could be interpreted as a form of dynamic inversion.

The reduction of computational cost with IDP is substantial and has allowed problems with as many as 300 dimensions to be solved on 2001 computing hardware [4.44]. It is easy to see why. Equation 4.2.11 enumerates the total number of states S that a typical ATM system can take at any time stage, if it consists of A aircraft, N states per aircraft and R is the desired resolution of grid points per state dimension.

$$S = R^{A \cdot N} \quad (4.2.11)$$

Then a naive brute force tree-search of the resulting ATM problem spanning P time stages, would result in C_{TREE} objective function evaluations, C_{TREE} comparisons between each result, and C_{TREE} additions between stages as shown in Equation 4.2.12

$$C_{TREE} = S^1 + S^2 + \dots + S^{P-1} + S^P \quad (4.2.12)$$

thus

$$C_{TREE} = S^{P!} = R^{(A \cdot N) \cdot P!} \quad (4.2.13)$$

For the typical DP formulation, the decision-tree structure is stopped from expanding exponentially with respect to P by exploiting the overlapping substructure of the problem along the time axis. The number of mathematical operations is reduced as per Eq. 4.2.14.

$$C_{DP} = P \cdot S^2 = P \cdot R^{2A \cdot N} \quad (4.2.14)$$

Also let the resolution R be expressed as a product ($G \cdot \gamma^I$) of a coarser grid G , and a contraction factor γ exponentiated by an integer constant I as per Equation 4.2.15.

$$C_{DP} = P \cdot (G \cdot \gamma^I)^{2A \cdot N} \quad (4.2.15)$$

For the IDP algorithm the number of operations is reduced further as per Eq. 4.2.16, where I can be interpreted as the number of iterations required to reach the equivalent resolution of the original DP formulation, $R = G \cdot \gamma^I$.

$$C_{IDP} = P \cdot I \cdot G^{2A \cdot N} \quad (4.2.16)$$

An additional reduction in complexity can be had if the interaction between aircraft can be ignored and somehow modelled in the objective function. See equation 4.2.17.

$$C_{RIDP} = P \cdot I \cdot A \cdot G^{2 \cdot N} \quad (4.2.17)$$

Thus finally:

$$C_{RIDP} \ll C_{IDP} \ll C_{DP} \ll C_{TREE} \quad (4.2.18)$$

However, the mechanism of IDP and other MRPs also imply important caveats: The coarse initial grid, as well as subsequent iterations, risk missing many local optima altogether, in highly multimodal problems. Pathological cases involving narrow quasi-convex wells might escape initial sampling, and cause the optimizer to converge incorrectly if one of these wells happens to contain the global optimum. IDP, in its original form, seems incapable of dynamic optimisation, since convergence is unidirectional due to grid collapse. The ability to reconverge when changes are detected needs to be included somehow.

4.2.9.3 Time Dimension Exclusion

The way to handle the time dimension is a recurring issue in trajectory optimisation problems. While evaluating the objective function, one might be initially tempted to search directly for collisions by measuring 4D Euclidian distance between all knot combinations in 4D space, but this gives rise to a quadratic problem at the most critical point. This approach would entail $4P^2N(N - 1)$ comparisons for N aircraft and P trajectory knots during each evaluation of the objective function, which would greatly hamper scalability of any multi-trajectory algorithm.

However, time is a special dimension, in that temporal location is common to all and motion along it can only be forward and linear, which is to say that different aircraft cannot be at different points along the time axis. All aircraft traverse the time dimension in synchrony. While this might appear obvious at face value, the implications are quite significant. It means that the search space along the time axis collapses to a single point, and that the remaining dimensions form an ordered set. Time can be used as a free variable. Collisions in the remaining three dimensions can therefore be considered one step at a time resulting in $3PN(N - 1)$ comparisons. This means that computational complexity increases linearly (rather than quadratically) with the number of trajectory knots.

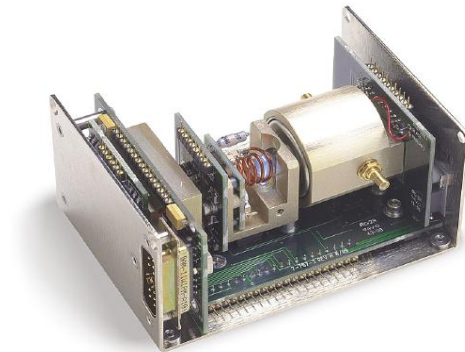


Figure 4.2.7: Rubidium Standard [4.45]

In an operational environment, this also implies that all trajectories must be synchronised to a predetermined universal time grid. Each take-off, landing and knot in every trajectory must align to one global time standard of discrete intervals. However, this is not technologically hard to implement by equipping all aircraft with high reliability, low jitter atomic clocks. Low cost and compact Rubidium standard atomic time references are available off the shelf and have a design life of over 20 years [4.45] (Figure 4.2.7).

These offer excellent short term time stability of the order of 1 part in 10^{12} , over durations of typical flights. This translates into a 10ns uncertainty (equivalent to a spatial uncertainty of about $3\mu\text{m}$) over a typical 3 hour flight travelling at 800km/h, which is insignificant by aerospace standards. Longer term stability can also be ensured by locking the onboard atomic clock to GPS reference signals that are anyway received by most modern aircraft.

4.2.9.4 Parameterisation

We have already discussed the topic of parameterisation in the context of reducing communication overhead between aircraft in chapters 1 and 2. However, it also serves as another important method for reducing dimensionality. The underlying concepts are simple.

In CATM, output flight trajectories are defined as 4D curves that aircraft are contractually obliged to follow closely. They specify the exact location (and hence velocity) of each aircraft at any point in time, from take-off to landing. This implies the need for a high resolution representation for the trajectories. However, rather than numerically solving for a dense grid of sample points, it would be advantageous to solve for a much reduced number of variables defining a smooth parametric curve that can in turn closely interpolate the desired state trajectory. There are a number of ways how this can be accomplished and all use some form of parametric curves [4.46].

Splines can be used to define trajectories as a sequence of (usually) cubic polynomials attached end-to-end with additional constraints placed on the continuity of derivatives at the junctions.

Bezier Curves, $B(t)$ can define trajectories in terms of a free parameter, t , a set of control points, P_i , and a linear combination of Bernstein polynomials, $b_{i,n}(t)$:

$$B(t) = \sum_{i=0}^n P_i \cdot b_{i,n}(t) \quad \text{where } t \in [0, 1] \quad (4.2.19)$$

$$b_{i,n}(t) = \sum_{i=0}^n \binom{n}{i} (1-t)^{n-i} t^i \quad \text{and where } i = 0, \dots, n \quad (4.2.20)$$

B-Splines (not to be confused with polybeziers) are an extension of Bezier curves, which allow greater flexibility and control of the resultant trajectories. Their order is independent of the number of control points and they offer compact support (control points only affect the curve locally). They are defined as a linear combination of basis splines, which are in turn defined recursively as per the Cox-deBoor algorithm [4.47] where t is the free parameter, p is the order of the B-Spline:

$$N_{j,1}(t) = \begin{cases} 1 & \text{if } t_j \leq t \leq t_{j+1} \\ 0 & \text{otherwise} \end{cases} \quad \text{and where } i, j = 0, \dots, n \quad (4.2.21)$$

$$N_{i,p}(t) = \frac{t-t_i}{t_{i+p-1}-t_i} N_{i,p-1}(t) + \frac{t_{i+p}-t}{t_{i+p}-t_{i+1}} N_{i+1,p-1}(t) \quad (4.2.22)$$

NURBS (or non-uniform rational B-splines) are a further enhancement of B-splines, offering the additional ability to model conical curves. They also define trajectories in terms of a free parameter, t , a set of $n+1$ control points, P_i , and weights W_i , and a combination of B-Spline Basis functions, $N_{i,p}(t)$:

$$B(t) = \frac{\sum_{i=0}^n W_i P_i \cdot N_{i,p}(t)}{\sum_{i=0}^n W_i \cdot N_{i,p}(t)} \quad \text{and where } i = 0, \dots, n \quad (4.2.23)$$

Spectral Methods approximate trajectories as a linear combination of continuous functions $\Gamma_k(t)$ that span the entire time interval $t_f - t_i$ of the trajectory. These methods include the Fourier series, the Laguerre series and the Chebyshev series, depending on the choice of kernel function $\Gamma_k(t)$. For some curves, accurate representation requires $N = \infty$, but more importantly, good approximations can often be achieved with much smaller N .

$$f(t) = \sum_{k=0}^N a_k \Gamma_k(t) \quad \text{where } t \in [t_i, t_f], \quad N \in \mathbb{N} \quad (4.2.24)$$

4.2.9.5 Convex Hull Property of Beziers, B-Splines and NURBS

Some curves, such as NURBs also exhibit the very useful property that guarantees that the interpolated trajectory is fully enclosed within the convex hull produced by the knots defining the curve. If the hulls can be shown not to intersect, then the resulting trajectories are mathematically guaranteed to be conflict free. This can greatly simplify formal verification at a later stage. The significance of this in the ATM context is quite remarkable.

The hulls are derived from a weighted combination of the vertices forming a tubular polytope that defines the feasible region joining the departing point to the destination. The finite number of vertices of the polytope defines the 4D NURBS trajectory contained within it, which can be uniquely generated at an arbitrarily high resolution, as necessary. By extension of the convex hull property, it can also be easily shown that non-intersecting polytopes also imply non-intersecting trajectories.

4.2.10 Flight Interaction with Finite Resolution Trajectories

The limited discretization resolution of trajectories can lead to certain pathological cases that must be avoided with the appropriate provisions. It is for example conceivable that two trajectories intersect in 4D space such that none of the sample points breaches minimum-separation criteria, while still colliding in each other's inter sample interval as shown in **Figure 4.2.8**. This problem does not arise if the sampling rate is increased, but this comes at the cost of higher dimensionality and the associated difficulties. Thus, the sampling rate must be increased no more than absolutely necessary.

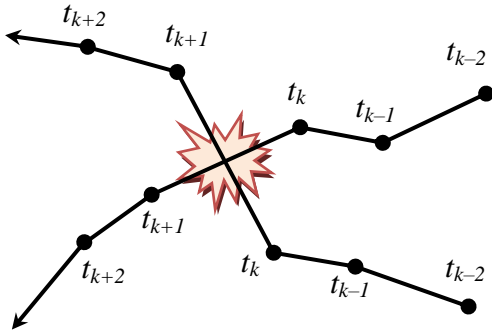
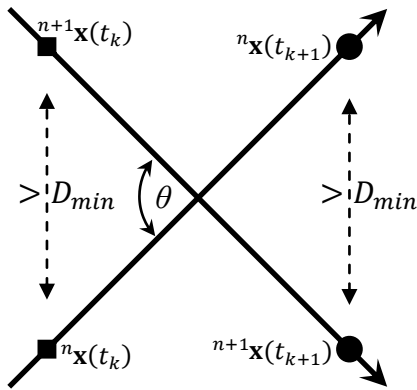


Figure 4.2.8: Inter Sample-Point Conflict


 Figure 4.2.9: Worst case scenario $\theta \rightarrow 180^\circ$

The following analysis can therefore form the basis for deciding the minimum sample rate for such trajectories. **Figure 4.2.9** shows the worst case scenario of two trajectories intersecting at the midpoint between their respective sample points. The worst case scenario arises when the angle of approach reaches 180° - a head on collision. The CATM optimiser algorithm is designed to penalise trajectory pairs only when the Euclidian distance between corresponding time samples differs by less than a $D_{min} \cong 5km$

If we assume that this condition is observed by the optimiser, then we must also ensure that no two flights can get closer than D_{min} in the inter-sample point gap. It is simple to see that if the distance between sample points is less than D_{min} , then this scenario will not arise. If two flights flying on a head-on collision course fail to breach this minimum before the collision point, they will certainly breach it at the subsequent sample point if the sampling resolution is better than D_{min} . This will be thus be detected by the optimiser.

However, this again assumes a spherical buffer zone and thus Euclidian distances are being used by the optimiser, but it turns out that the Euclidian norm (Eq 4.2.25) is computationally intensive (Eq. 4.2.26) and it is desirable to simplify such distance calculations as much as possible, given that they must be performed billion of times.

$$\|n\mathbf{x}_{t_k} - n^+1\mathbf{x}_{t_k}\| > D_{min} \quad (4.2.25)$$

$$\sqrt{\left(nx_{t_k} - n^+1x_{t_k}\right)^2 + \left(ny_{t_k} - n^+1y_{t_k}\right)^2 + \left(nz_{t_k} - n^+1z_{t_k}\right)^2} > D_{min} \quad (4.2.26)$$

We can obtain a significant simplification by using octahedral buffer zones instead of spherical ones. This is shown in **Figure 4.2.10** where it can be seen that if the aircraft lies at the centre of the octahedron, the closest points between its surface and the aircraft are in the middle of the octahedron's faces. This is taken into account using basic geometry in equation 4.2.27 where it can be seen that the octahedron buffer zone relies on simple arithmetic such as subtraction and modulus, that are much faster to compute than squares and square roots. Therefore, satisfying this inequality means that the aircraft are at least D_{min} apart. If not, a penalty will be applied to the two offending flights.

$$\left(\sqrt{3} \cdot D_{min} - \left|nx_{t_k} - n^+1x_{t_k}\right| - \left|ny_{t_k} - n^+1y_{t_k}\right| - \left|nz_{t_k} - n^+1z_{t_k}\right|\right) > 0 \quad (4.2.27)$$

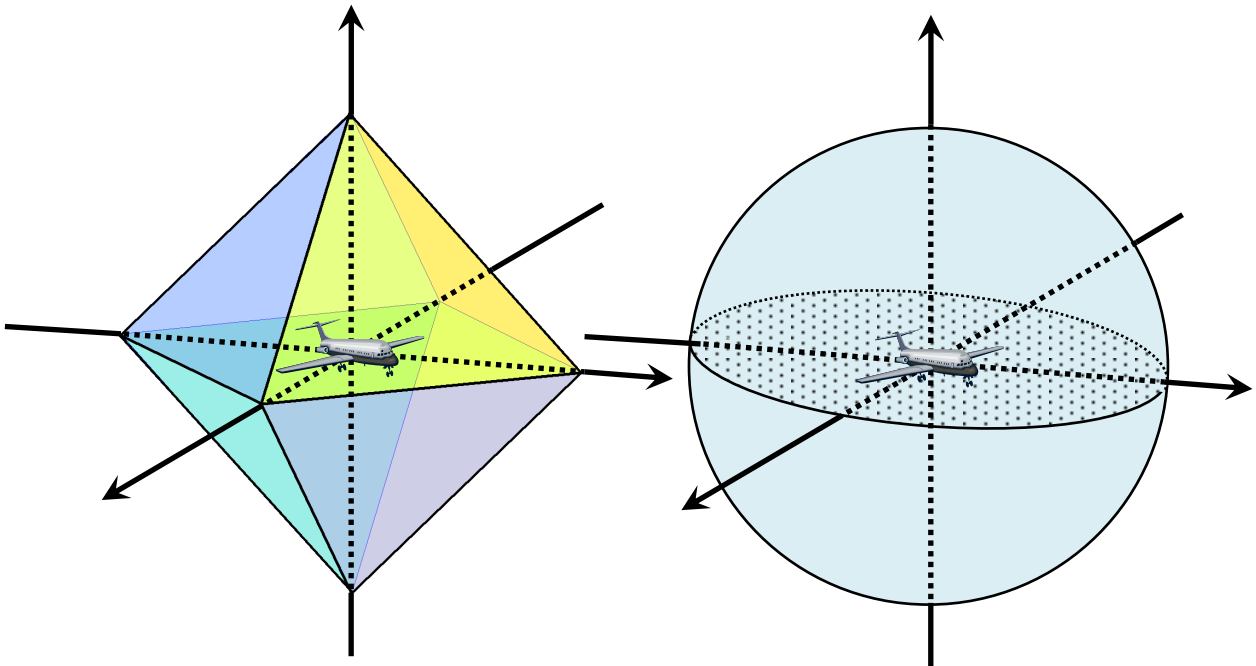


Figure 4.2.10: Octahedral and Spherical Buffer Zones Surrounding Aircraft Positions

Using 5km separation limits implies that the trajectory discretization resolution must exceed 2000 points for a 10,000km journey. This is computationally unfeasible given the large number of flights involved. However, much of the flight consists of straight lines and a few smooth curves, mostly circular due to coordinated-flight requirements. Thus, interpolation using smooth functions such as NURBs is a practical alternative that should result in high accuracy results. A 20:1 up-sampling ratio results in just 100 points per flight that need to be determined by the optimiser, while the NURB interpolator fills-in the rest. In order to ensure that the interpolated trajectory meets all the constraints, all fitness or cost evaluations are conducted on the interpolated version of each flight while the optimiser determines the knot values defining the NURB.

4.2.10.1 Combination Techniques

In summary, after having discussed the various ways of handling high dimensionality problems, it is worth noting that these are not mutually exclusive ideas. In fact a combination of techniques is a very promising way forward. MRP already has a lot in common with evolutionary algorithms as well as swarm intelligence. The latter methods start with large populations spanning the entire search space and with every iteration they converge to examine ever smaller search-space volumes. So it may be interesting to explore the idea of explicitly contracting population diversity over each iteration, in order to better control the intensification process. Conversely, it would also be interesting to explore the possibility of using DP as a mechanism of optimal crossover in genetic algorithms. Parameterisation would be desired in either case.

4.3 GLOBAL OPTIMISATION PHASE

We shall now proceed to describe in some detail some of the algorithms that have been explored in this work. These have been implemented and adapted for the combination of characteristics that arise in CATM.

4.3.1 Search Techniques

With optimisation being one of the most heavily researched subjects ever, a full taxonomy of optimisation methods would be beyond the scope of this work. However, after substantial review, we lay our focus on two important classes; Deterministic single solution searches and meta-heuristic population-based searches. We further narrow our focus into three important sub-classes of global optimisation algorithms, as shown in **Figure 4.3.1**

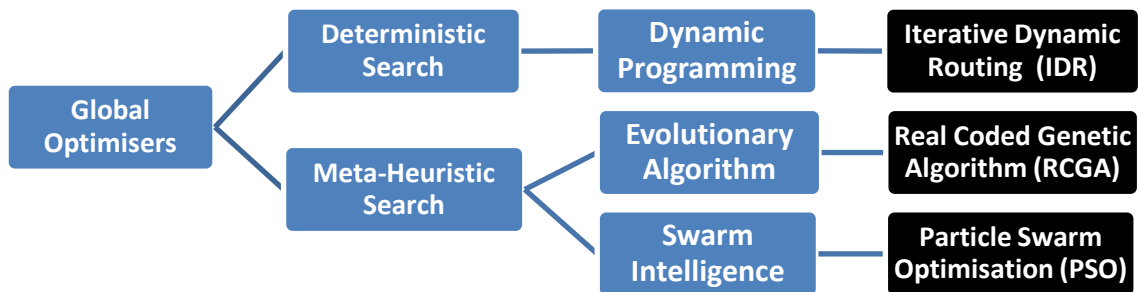


Figure 4.3.1: Algorithms considered for CATM global search

4.3.1.1 Deterministic Single Solution Searches

A deterministic single-solution search constructs (or performs incremental improvements on) one candidate solution. If exhaustive, it is guaranteed to eventually find the optimal solution in any multimodal problem. Continuous search spaces, may be discretized as finely as necessary to convert them into a combinatorial optimisation problems. One can then transcribe all the possible choices, into a tree graph, and search it for the best outcome. However, for obvious reasons it is impractical to naively scan every branch, except for very small problems. Instead, one of a number of techniques (or heuristics) can help organise, and prioritise the search as efficiently as possible. Branch and bound, greedy search, A*, B* and D* are just some of the many possibilities.

4.3.1.2 Meta-Heuristic Population-based Searches

Meta-heuristics have witnessed a dramatic increase in popularity [4.48], and yet a unifying definition of this class of algorithms remains elusive. A recent definition was given by Sörensen and Glover:

“A metaheuristic is a high-level problem-independent algorithmic framework that provides a set of guidelines or strategies to develop heuristic optimization algorithms.... Metaheuristics are therefore developed specifically to find a solution that is “good enough” in a computing time that is “small enough” [4.49]

Population based methods, iteratively generate a large number of candidate solutions, part stochastically, part heuristically, to sample the search space as widely as possible.

4.4 DYNAMIC PROGRAMMING

An exhaustive search called dynamic programming (DP) was developed by Richard Bellman in 1953. This provides a comprehensive solution to multistage optimisation problems which can be transcribed into a number of sequential decisions in which the outcomes of those preceding, bear an effect on future ones [4.50]. This is particularly useful when a decision tree diagram can be folded into a trellis-shaped graph, in which case the problem can be termed as having an optimal substructure. In this case, DP reduces the number of operations, compared to a brute-force tree search, as we have seen earlier.

4.4.1 Bellman's Principle of Optimality

“An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision.” [4.51]

Thus restating the principle of optimality in mathematical terms, the Bellman equation is given recursively by 4.4.1 for combinatorial optimisation problems with optimal substructure. As commonly formulated, this is solved by backward induction, starting from the desired terminal state \mathbf{x}_f and ending with the initial state \mathbf{x}_0 . However, by simple symmetry, we can also opt for the forward recursive version as in equation 4.4.2

$$\text{Reverse:} \quad V_R(\mathbf{x}_k) = \min_{\mathbf{a} \in \Gamma(\mathbf{x}_k)} [U(\mathbf{x}_k, \mathbf{a}_k) + \beta V_R(\mathbf{x}_{k+1})] \quad (4.4.1)$$

$$\text{Forward:} \quad V_F(\mathbf{x}_{k+1}) = \min_{\mathbf{a} \in \Gamma(\mathbf{x}_k)} [U(\mathbf{x}_k, \mathbf{a}_k) + \beta V_F(\mathbf{x}_k)] \quad (4.4.2)$$

where: \mathbf{x}_k is the system state vector at step k

\mathbf{a}_k is the action vector at step k (also called the control policy function)

$\beta_k \in [0,1]$ is a discount factor to optionally modulate the cost of early controls

$\Gamma(\mathbf{x}_k)$ is the feasible space for choosing \mathbf{a}_k , as a function of state \mathbf{x}_k (4.4.3)

$V_R(\mathbf{x}_k)$ is the objective/value function, ie: the cost-to-go from \mathbf{x}_k to \mathbf{x}_f

$V_F(\mathbf{x}_k)$ is the objective/value function, ie: the cost incurred from \mathbf{x}_0 to \mathbf{x}_k

$U(\mathbf{x}_k, \mathbf{a}_k)$ is the utility cost of taking action \mathbf{a}_k to change state $\mathbf{x}_k \rightarrow \mathbf{x}_{k+1}$

Similarly, the same ideas can be further extended to the continuous time domain as expressed in the reverse and forward versions of the Hamilton–Jacobi–Bellman (HJB) equation of 4.4.4 and 4.4.5 respectively.

$$\text{Forward HJB:} \quad V_R(\mathbf{x}_t, t) = \min_{\mathbf{a} \in \Gamma(\mathbf{x}_t)} \left[\int_t^{t+dt} U(\mathbf{x}_t, \mathbf{a}_t) dt + \beta V_R(\mathbf{x}_{t+dt}, t + dt) \right] \quad (4.4.4)$$

$$\text{Forward HJB:} \quad V_F(\mathbf{x}_{t+dt}, t + dt) = \min_{\mathbf{a} \in \Gamma(\mathbf{x}_t)} \left[\int_t^{t+dt} U(\mathbf{x}_t, \mathbf{a}_t) dt + \beta V_F(\mathbf{x}_t, t) \right] \quad (4.4.5)$$

The Principle of Optimality and the Bellman equation are put into practice through dynamic programming. Using spatial path planning as an example, **Figure 4.4.1** shows how a weighted, directed, decision graph representing all the possible

trajectories towards a goal, can be traversed. In this routing problem, each node represents a spatial location and is associated with a cost. This cost is the minimum cost incurred to get to that location.

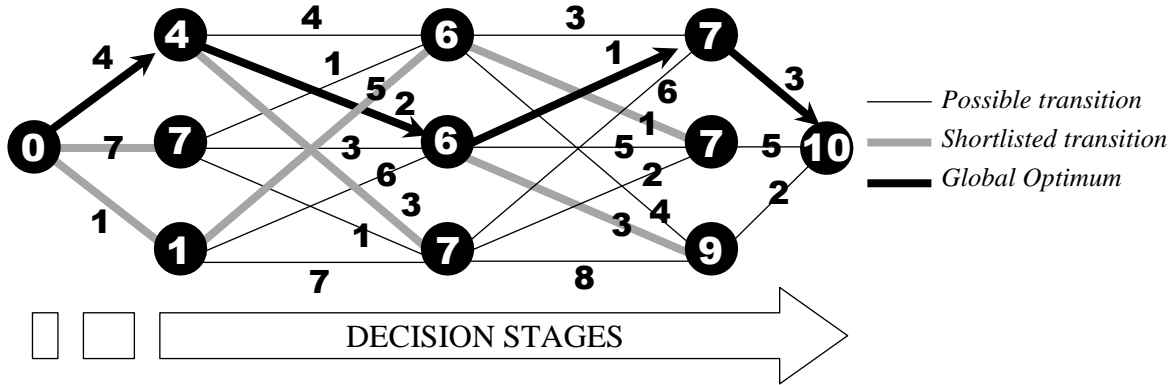


Figure 4.4.1: Forward Induction Dynamic Programming

The nodes are arranged in stages, which can either represent snapshots in time, or if time itself is among the searched dimensions, then each stage is some dimensionless ordered quantity. Each edge represents a feasible transition between two locations, and carries an associated transition cost. The initial state (or location), shown to the left carries a cost of zero. The DP algorithm proceeds to compare the incoming transitions at each node, by adding each transition cost to its source node cost, respectively. The minimum cost branch entering each node is thus selected and forms the basis of assigning cost to the current node. The process continues at each node until every node has been assigned a cost. In line with Bellman’s principle, at any stage one need not consider what prior decisions have been taken at any prior stage. One is only concerned with minimising the incremental cost associated with the current stage. When the final stage is reached, the same process is repeated. The last decision taken between the incoming transitions, effectively determines which of all the incoming routes was globally optimal.

4.4.2 Structured Dynamic Programming

It is clear, from the forgoing that this broad-spectrum optimisation procedure, is exhaustive, and makes no assumptions of convexity. However, the single greatest limitation is related to the size of the state vector and the associated resolution of discretization.

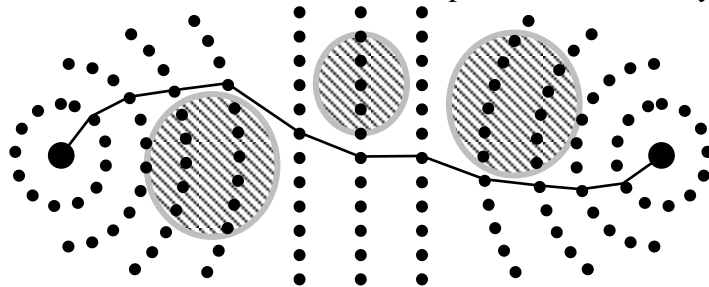


Figure 4.4.2: Structured Output Search Space with Obstacles

A coarse grid gives meaningless results, while a fine grid would make the problem intractable when the state vector is large. In order to reduce problem size somewhat, one may

heuristically structure the grid (Figure 4.4.2) such that it loosely follows the expected “reasonable solution” space. This can greatly trim down the search space but may reintroduce some bias, if not done carefully. This is in fact one way of applying path constraints to the problem.

4.4.3 Iterative Dynamic Routing

Iterative dynamic routing (IDR) is a multi resolution dynamic programming technique inspired by IDP, but specific to spatial, time-variant path planning problems. The same concept can be augmented to include the entire ATS system state vector coupled with time. With the addition of aircraft models, multiple traffic and airspace restrictions, the edges of the graph can be assigned an appropriate cost, and these algorithms can be used for full 4D trajectory generation. However, for clarity we are hereby limiting diagrams and descriptions to a simple 2D or 3D spatial path planning.

In IDR the search space is contracted gradually over several passes, while retaining the same number discretization levels per dimension. Each contraction is kept centred around the optimal result of the previous pass. This increases the effective spatial resolution of the grid, while avoiding problem growth, thus allowing for more refined searching in the vicinity of the expected global optimum. **Figure 4.4.3** shows the result of a first pass through the structured search space, while **Figure 4.4.4** shows the second pass after contraction. This goes on as long as necessary until some convergence criteria are met.

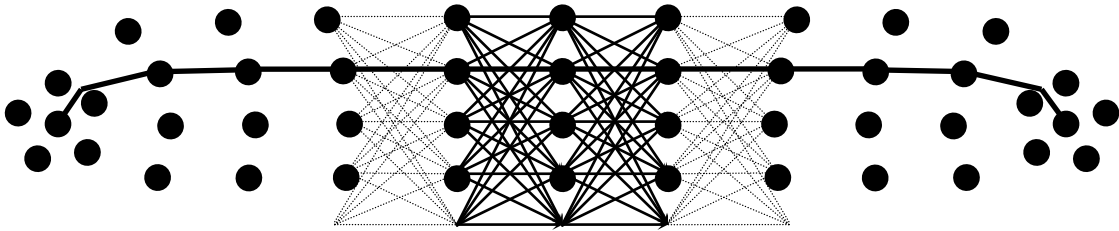


Figure 4.4.3: First Pass of a structured ITR problem

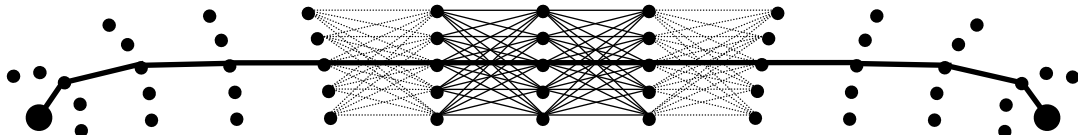


Figure 4.4.4: Second pass: Search Space Contracts by a constant factor γ

This algorithm was tested in a number of scenarios and several observations can be made. **Figure 4.4.5** shows a single aircraft negotiating a large obstacle in 2D space. The objective is finding the minimum distance path between the given terminals while clearing the obstacle. This problem is clearly non-convex with two obvious local minima. The aircraft may either fly above or below the obstacle. The feasibility constraints caused by the presence of the obstacle are handled using a hard penalty function $P(x, y)$ of the form:

$$\text{obstacle:} \quad P(x, y) = \text{Mag} \cdot \left[\text{Rad} > \left(\frac{(x - c_x)^\alpha}{w_x} + \frac{(y - c_y)^\alpha}{w_y} \right)^{1/\alpha} \right] \quad (4.4.6)$$

where: c_x, c_y define the geometric centre of the object
 w_x, w_y define the relative width and height of the object
 $\alpha > 1$ defines the degree of eccentricity of the object
 $\alpha = 2$ creates an ellipsoid, while $\alpha > 2$ creates a rounded rectangle
 $P(x, y)$ gives the penalty associated with spatial location (x, y)
 Mag represents the magnitude of the penalty when (x, y) overlaps the object
 Rad defines the perimeter of the object

(4.4.7)

In **Figure 4.4.5 (a)** one can observe the initial structured grid that explores the airspace in coarse manner. In this implementation, the grid spacing is cast randomly rather than uniformly as described earlier. This seems to improve exploration. As can be seen, on the very first iteration the algorithm determines which of the two possible minima seems most promising. All paths that terminate within the unfeasible space are not pursued further due the high cost incurred by the penalty function. The selected path in **(a)** forms the basis for casting the subsequent contracted grid in **(b)** and in **(c)** the collapsed grid is shown.

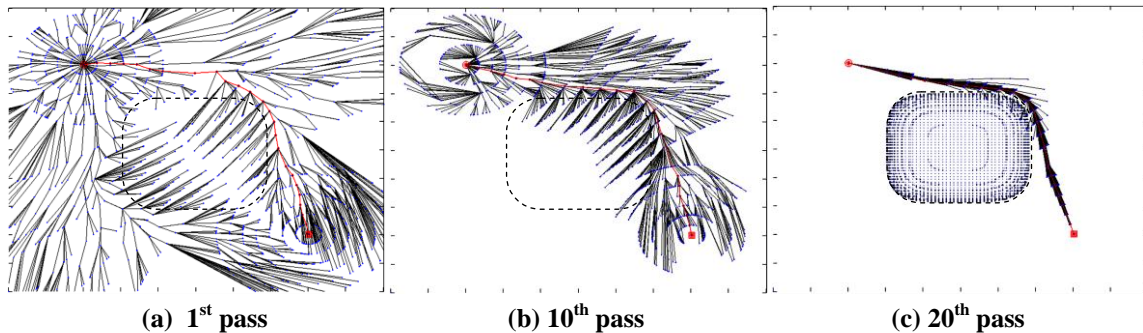


Figure 4.4.5: IDR Path Planning progress in 2D with static obstacle

4.4.4 Counteracting Collapse

The defining feature of IDR also generates its major flaw: As described so far, it only works with static value functions. The gradual contraction of the search space on every iteration, restricts the algorithm from being able to adapt to any changes in the search space. If after initial convergence, in the global optimum shifts to a region beyond the volume of the currently searched space, as would happen in a dynamic problem, the algorithm will fail to re-converge to this new optimum. A mechanism is therefore required to allow the system to detect and track changes.

A technique was developed and tested as part of this work to address this problem. It allows the search space contraction to be reversed and hence re-expanded when dynamic behaviour is detected. Taking a 3D problem as an example, it involves dividing the grid into three zones:

- The inner contractile search grid surrounding each optimal node
- Two concentric auxiliary ring grids also centred round the optimum
 - Ring A: Also contractile, however it uses a smaller contraction factor, γ
 - Ring B: Set at an *absolute* distance from the contractile grid perimeter

Figure 4.4.6 demonstrates this arrangement graphically for one stage. These additional rings contribute towards the local discretization of the search space, and over each pass DP works as usual to determine the optima path through the grid. However, if the optimum is time invariant, then these rings are not likely to ever coincide with the optimum, since the direction of grid contraction between subsequent passes is always centred *around* the optimum. But this will change if the location of the optimum begins

to shift. In this scenario, the optimum might move closer towards Rings B or A. When this is detected, by DP, the contraction is reversed, on the assumption that the optimum may have moved beyond the current search space. The expansion continues for as long as the optimum appears to strike the rings.

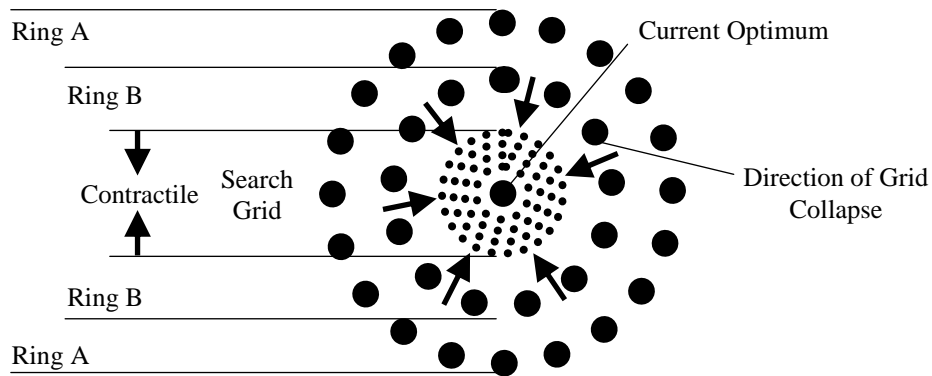


Figure 4.4.6: Countering Collapse in IDR

This arrangement has been empirically demonstrated to be effective to deal with dynamic problem where the global optimum shifts slowly with respect to the time it takes to compute a single IDR pass, as shown progressively in **Figure 4.4.7 (a) and (b)**

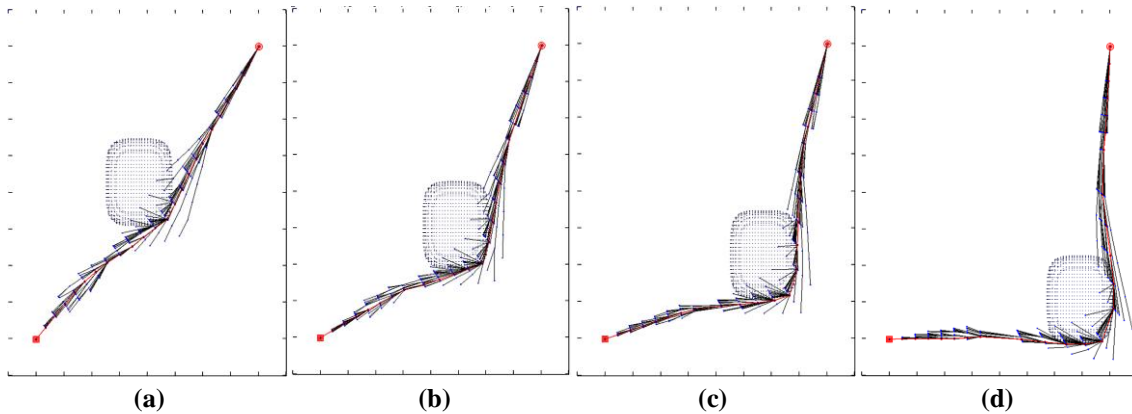


Figure 4.4.7: IDR Path Planning, after initial convergence in 2D with a dynamic obstacle

However, another problem remains. Although the augmented algorithm is able to track optima, it is unable to determine whether the tracked regional optimum is still the global optimum. For this scenario as shown in **Figure 4.4.7 (c) and (d)**, the result is that the optimiser remains trapped in the local minimum and there seems to be no better option other than to restart the IDR regularly, to discard previous assumptions, but this is computationally costly.

4.4.5 Multi Aircraft Approaches with IDR

Handling multiple aircraft with IDR is not a straight forward issue. Naively concatenating the state vectors into a single all-encompassing problem leads to unacceptable search space growth. So there are three other possible ways worth exploring:

- 1) One approach recognises the role of locality. Under this paradigm, trajectories are ordinarily solved independently. However, when it is established that any two or more trajectories are likely to intersect in 4D space, those trajectories are then solved collectively using IDR after concatenating their state vectors. This has the potential to greatly reduce the size of the problem by limiting combinatorial growth to isolated clusters involving flights with direct interaction, but there is one caveat: Trajectories can form chains of dependencies, through their interactions. This makes the size of the problem potentially unbounded within a cluster.
- 2) Another approach involves the use of a *polygrid*. This is when the discretization grid changes dimensionality along the trajectories. For segments of flight with no interaction, flights are optimised in an isolated manner. However, when two or more flights interact, a portion of trajectory surrounding that interaction is solved by concatenating state vectors. This limits the interaction to just small sections of the trajectories, much like CD&R, and hence reduces the risk of unbounded cluster growth. However, some issues remain on how to handle the changeover between grids of differing dimensionality.
- 3) A third approach is that of representing flight interactions using penalty functions. This simplifies the problem, by allowing each aircraft to essentially solve single flight problems in a fixed obstacle field. However, at every iteration, the remaining flights are treated as static obstacles that collectively generate an additive penalty-field that can be used to calculate the cumulative penalty of taking any particular route. The objective of this *Reduced Order IDR* (ROIDR) is to find the path with minimum interaction through the field. The disadvantage however, is that, like all penalty functions, the outcome is highly dependent on the many parameters that would define such penalty functions. **Figure 4.4.8**, shows a simulation result of a typical pathological scenario. This involves three flights that would ordinarily intersect at a point as shown in (a). The ROIDR successfully finds a good solution (b). However, it is difficult to ascertain whether this is indeed optimal. It can be argued that the calculated manoeuvres are exaggerated. Hence, this would require substantial manual fine tuning to achieve optimal results.

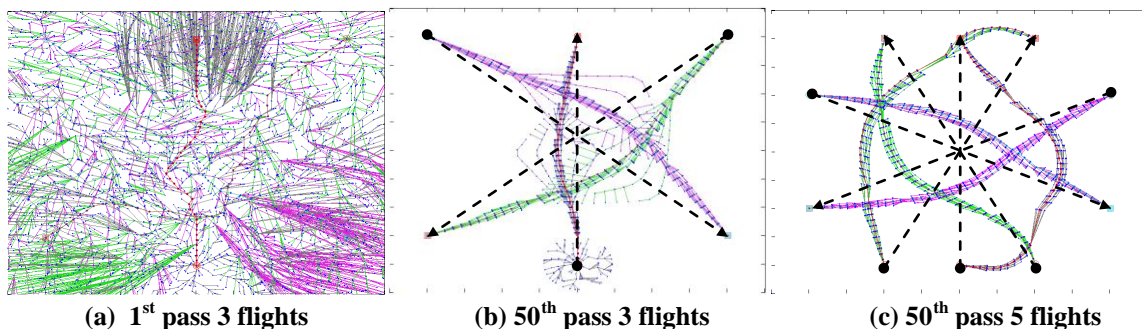


Figure 4.4.8: ROIDR Path Planning progress with 3 or 5 interacting flights in 2D space

Although, a substantial improvement in convergence rate can be obtained with IDR, enough to optimise one or a few aircraft trajectories at a time, this is not quite sufficient to deal with the multi-aircraft problem holistically, particularly when this involves thousands of flights. Not only does it not scale too well, but the techniques used to simplify the problem, also defeat DP's resilience from getting trapped in local optima. This is clear in **Figure 4.4.8 (c)** when the number of aircraft was increased to five. The departure from the shortest path is quite stark and the level of sub optimality is uneven between aircraft. This can be attributed to insufficient exploration of the search space and premature convergence. A different approach is therefore required, that is better suited at exploration without the associated combinatorial dimensionality expansion. We now move to probabilistic metaheuristic techniques.

4.5 EVOLUTIONARY ALGORITHMS

Evolutionary algorithms (EA) are metaheuristics inspired by the Darwinian theory of natural selection. They were first proposed by Alan Turing in the 1950s in a philosophical article about computing machinery and intelligence [4.52], but they only gained popularity as a machine learning tool following the work by John Holland in the 1970s [4.53].

4.5.1 Genetic Algorithms and Evolutionary Programming

The main concept of Evolutionary Programming (EP) revolves around maintaining a diverse simulated population of *individuals* whose *genetic identity*, is subject to *mutation*, and *selection* based on the *suitability* of each individual to *survive* in a given *environment*. Genetic Algorithms (GA) add the notion of *mating* and *reproduction* to the EP metaphor.

In optimal control terms, an *individual* represents a candidate solution to the optimisation problem. Its *genetic identity* represents the associated control policy. *Mutation* is the introduction of random variations into each solution. *Suitability* is defined as the ability to score highly when evaluated by the objective function. *Selection* is the process of discriminating between the strongest (most suitable) and the weakest (least suitable) solutions from the population. *Surviving* relates to being favoured in terms of reproductive prospects. *Mating* is the process of randomly combining traits of different candidate solutions, and as a result of *reproduction*, new candidate solutions are generated with increased diversity. The *environment* represents the problem to be solved. The algorithm is run over several iterations called *generations* until some stopping condition is met.

4.5.2 The Fitness Function

The fitness function is a scalar mapping between the objective (cost) function to a real valued figure of merit that indicates the relative suitability of a solution with respect to other candidates. The mapping need not be linear, but it should be monotonic. Every individual is evaluated by this user-defined application-specific operator that assigns fitness to each member in the population. For multi-objective problems, several fitness functions are used; one for each objective. These functions are evaluated millions of times in the inner loop of the GA and thus execution speed is of the essence. Great care is expended in making these functions as lightweight and optimized as possible.

4.5.3 Chromosome Encoding

In a genetic algorithm, *the chromosome* is a digital schemata [4.56] or a multi-part blueprint that holds all the information related to any member in the evolving population. There are a number of ways of constructing such chromosomes and this depends on the underlying nature of the optimisation variables. Thus, chromosome construction is problem specific. In the early years of the field there was much debate over the coding methods for GAs, with one faction favouring optimisation at the *genotype* level and others directly at the *phenotype*. Many real world problems benefit from the latter paradigm because there is less risk of losing information in the translation from real-domain to representation-domain.

4.5.3.1 Binary Coded Genetic Algorithms (BCGA)

For discrete domains, the chromosomes might consist in the concatenation of binary values representing a sequence of choices, where each bit represents a gene in the chromosome. This is called *binary encoding* and has been shown to represent the maximum size of schemata per bit of information [4.56]. This is useful for representing discrete combinatorial problems, and for many years this was the default implementation.

4.5.3.2 Real Coded Genetic Algorithms (RCGA)

For continuous variables (like waypoint coordinates) binary coding is not appropriate, since it would imply that some bit changes will have a much higher significance than other bits in the chromosome, causing *Hamming cliff* artefacts, particularly if the variables are represented as floating point numbers. This happens when a bit flip has a disproportionate effect on a variable, which is undesirable during the final intensification stages of the optimisation run. Efficiency is also impaired, because in the initial stages, the optimiser finds itself wasting far too much effort exploring the effect of bit flips of little significance. Thus, for a real-coded genetic algorithm (RCGA [4.56], [4.57]), the chromosomes typically consist of vectors of real numbers, where one gene is represented by one variable. In the case of trajectory optimisation these might be waypoint vectors, or aircraft control input vectors. This is called *value encoding* [4.58].

In this work, trajectory optimisation is being carried out in the output domain and therefore phenotype representation is used, where chromosomes simply consist of real-valued vectors of genes that define the spatial 3D knots locations along the trajectories. However, in order to allow better compatibility during crossover, these vectors define the spatial knots indirectly. Each vector element defines the difference between knots in the trajectory rather than the absolute knot locations. Although the goal is to generate 4D trajectories, time is not included in the search space, because this can be used as a free variable to synchronise all flight knots and simplify fitness functions considerably. These vectors are initialised quasi randomly with the observance of some basic constraints such as terminal constraints. Equations 4.5.1 describe the process.

$$\begin{aligned} \text{initialisation: } \quad & {}^n\mathbf{R}^x = \mathbf{rand}(K), \quad {}^n\mathbf{C}^x = ({}^nx_K - {}^nx_0) \cdot {}^n\mathbf{R}^x \cdot \left(\sum_{k=0}^{K-1} {}^nR_k^x\right)^{-1} \\ & {}^n\mathbf{R}^y = \mathbf{rand}(K), \quad {}^n\mathbf{C}^y = ({}^ny_K - {}^ny_0) \cdot {}^n\mathbf{R}^y \cdot \left(\sum_{k=0}^{K-1} {}^nR_k^y\right)^{-1} \\ & {}^n\mathbf{R}^h = \mathbf{rand}(K), \quad {}^n\mathbf{C}^h = ({}^nh_K - {}^nh_0) \cdot {}^n\mathbf{R}^h \cdot \left(\sum_{k=0}^{K-1} {}^nR_k^h\right)^{-1} \end{aligned} \quad (4.5.1)$$

$$\text{chromosome:} \quad {}^n\mathbf{C} = \{{}^n\mathbf{C}^x, {}^n\mathbf{C}^y, {}^n\mathbf{C}^h\} \quad (4.5.2)$$

K is the numbers of knots (or genes) in a trajectory.

$\mathbf{rand}(K)$ is a vector valued function that generates K random variables.

n is the chromosome number.

${}^n\mathbf{C}$ represents the initialised chromosome for the n^{th} member in population

$$\text{where:} \quad {}^n\mathbf{C}^x, {}^n\mathbf{C}^y, {}^n\mathbf{C}^h \text{ represent the initialisation value for each dimension} \quad (4.5.3)$$

${}^n\mathbf{R}^x, {}^n\mathbf{R}^y, {}^n\mathbf{R}^h$ represent three vectors of random numbers

${}^nR_k^x, {}^nR_k^y, {}^nR_k^h$ are the k^{th} elements of their respective vectors.

${}^nx_0, {}^ny_0, {}^nh_0$ represent origin of each trajectory in 3D space

${}^nx_K, {}^ny_K, {}^nh_K$ represent destination of each trajectory in 3D space

4.5.4 Selection

Selection is the process of favouring the fittest members in a population for survival and further dissemination of their genetic information. Selection also discards the weakest members, making way for offspring of the best parents. This process is inspired by natural selection whereby the fittest members of any species survive to procreate, thereby carrying over the genes that gave them the very advantage that allowed them to survive. In a GA, selection is enforced artificially using any one of various schemes. The key parameter here is called the *selection probability* and is computed as a function of fitness.

4.5.4.1 Selection Pressure

This *selection pressure* is what guides the GA optimizer, and without which GAs would be little more than glorified random searches. However, a balance must be struck. If too much selection pressure is applied (by weeding out too many of the weaklings) the GA will fail to suitably explore the search landscape by discarding potentially important information too early. This leads to premature convergence and is likely to get the GA trapped in the nearest local optimum. Population *diversity* is the principle resource that the GA draws upon to make progress.

On the other hand, too little selection pressure will result in far too much exploration and far too little exploitation. Exploitation is the term used in this context to signify refinement of solutions. This *exploration v.s. exploitation* balance is central to the idea of evolutionary metaheuristics. The focus of the optimizer tends to shift over the time, in general it is desirable to perform as much exploration as possible in the initial stages to identify the most promising regions of the search space. The objective shifts somewhat towards the end of the run, where the focus is on establishing the exact optimum in the selected well.

The main problem lies in the deception that some search spaces pose. What may initially seem to be a very promising region might turn out to be less optimal than others after exploitation is finished. Conversely, what may initially seem to be an uninteresting area to exploit may actually contain the global optimum. This highlights the needs to maintain sufficient diversity in the population, such that exploration never halts entirely.

4.5.4.2 Selection Techniques

Proportional Selection [4.53], is the classic method originally proposed by Holland. The selection is random. However, a selection probability is assigned in proportion to the fitness of the population members in relation to the rest as shown in Equation 4.5.5 below. After the probabilities have been assigned, a pie chart of all population members is constructed using their relative fitness value as the chart metric. Roulette Wheel Selection, (as seen before in **Figure 3.8.7** in chapter 3) is used as many times as necessary to select parents for carryover or crossover.

$$P_S(i\mathcal{C}) = \frac{f(i\mathcal{C})}{\sum_{n=1}^N f(n\mathcal{C})} \quad \text{where:} \quad \begin{cases} P_S(i\mathcal{C}) & \text{is the selection probability of Chromosome } i \\ f(i\mathcal{C}) & \text{is the fitness of Chromosome } i \\ n\mathcal{C} & \text{is the Chromosome of population member } n \\ N & \text{is the size of the population} \end{cases} \quad (4.5.4)$$

Rank Selection sorts all the members of the population by their fitness value. The probability of selection is then increased linearly according to the rank order in the list. Roulette Wheel Selection is then used as before. This method of selection favours weaker members by assigning them a probability disproportionately large compared to their raw fitness value. This method slows down convergence but allows greater exploration. However, selection pressure can be adjusted by truncating the ranked list as required.

Tournament Selection [4.54], is a popular technique that uses very few comparisons (logarithmic) to select members from the population. A number of random members are paired and compared pairwise in terms of fitness. The winner of each pair is paired with the winner of another pair and the process repeats similar to a football tournament, until a winner is declared. The tournament is conducted as many times as there are parents. Selection pressure can be adjusted by changing the tournament size, with small tournaments favouring weaker members.

Steady-State Selection, also called *Parental Persistence*, is the process of allowing the majority of the parents to survive to the next generation, with a few offspring generated from the best members replacing a few of the weakest members of the previous generation. This requires a technique to avoid wasting computational resources on reevaluating the same members all over again.

Elitism is the process of setting aside a few of the very best members of the population for repeated inclusion into the subsequent populations. This is a very effective technique that guarantees that the GA never loses good solutions once they are found.

Idealism is an intuitive but very effective technique introduced by the author specifically for trajectory optimisation work. Genetic algorithms have been heavily criticised in the past for their inability to fully exploit the tremendous quantity of information present in the search landscape [4.55]. This makes them very inefficient compared to gradient methods for solving convex problems (albeit CATM being non-convex, which rules out gradient methods anyway).

However, in ATM, the ideal (but impractical) solution is typically known. It is obviously the straight line joining the departure to destination points. It so happens that this naive solution already contains most of the information needed to solve the problem. It simply needs to be adjusted to take into account realistic constraints. Even though it is clearly unfeasible, it can provide significant suggestive information on where the global solution might be located. By including the ideal solution into the population, the GA will occasionally crossover this ideal member with other members to generate offspring that are superior to both. Thus, the GA is immediately guided towards regions in proximity of the global optimum and this increases the rate of convergence.

4.5.5 Crossover

Very often population members have complementary strengths and weaknesses. This is also true in the case of biological systems. Mating is the process of mingling genetic material from the parents in various ways to generate unique offspring with new genetic identities arising from different combinations of the parent’s genetic heritage. The process of combining genetic blueprints is called crossover. Several offspring can thus be produced from a highly varied range of possible parental gene permutations.













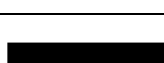

Crossover Techniques	Parents	Offspring
Single Point (Simple) Crossover [4.60]: uses a single crossover point at a random position along the parent chromosomes. These are then sliced at the same relative point and the two parts (called <i>oligomers</i>) are swapped and spliced to generate two new offspring		
Two Point Crossover: extends single point crossover with a second random crossover point. Each chromosome is split into three oligomers which are then alternately recombined into two complementary offspring.		
Multipoint Crossover: generalises two point crossover to multiple points and disperses parental contribution. It gives good results with trajectory optimisation. Two complementary offspring are generated as a result.		
Uniform (Discrete) Crossover [4.62]: consist of fine grained random swaps of genetic data between the parents to produce offspring with highly uniform contributions from both		
Flat Crossover [4.59]: is another fine grained method where the offspring receive randomly weighted sums of individual genes taken between the parents to produce offspring with highly uniform contributions from both.		
Arithmetic Crossover [4.60][4.61]: combines parents using a weighted average (or other math operator) and will produce a single offspring. However, more offspring can be produced by modifying the arithmetic operator or weighting parameters.		
Multi-parent Crossover [4.63]: extends several of the aforementioned techniques to more than two parents at once, resulting in a broader variety in the offspring produced. However, the advantage of having multiple parents depends on the context		

Table 4.5.1: Some Crossover Techniques Applicable to Real Coded Genetic Algorithms

There are numerous ways of performing crossover and they are highly dependent on the type of encoding that is adopted by the chromosomes. The interested reader is referred to various surveys on the subject [4.57]. Several techniques that are applicable to value encoding were implemented and tested and are described in **Table 4.5.1**.

The random confluence of genes from the parents results in some combinations that might be detrimental to the resulting offspring, particularly when unfavourable genes converge into the same chromosome. However, it is equally likely that enhanced offspring, that are better than either parent, are generated by combining the best characteristics of each. This makes crossover one of the most powerful operators in GAs and is particularly important in the case of trajectory optimisation. The process of

crossover is governed by a factor called the crossover probability that determines the likelihood of crossover occurring between the parents. This is usually high (85% to 95%) as failing to crossover would generate offspring that are essentially replicas of the parents.

4.5.6 Mutation

While crossover combines traits across individuals to trade diversity for fitness, mutation is the process of introducing new diversity into the population to partly counteract the loss of it. This can be achieved in ways related to the manner of encoding. In the case of binary encoding, a few bits are flipped randomly along the chromosomes of some individuals. In the case of value encoding, real valued noise replaces or is arithmetically added to some genes in the chromosomes.

Mutation Technique	Originals	Mutants
Random Uniform Mutation [4.60]: replaces a randomly selected gene in a chromosome and replaces it with a random real number taken from a uniform distribution restricted to the ranges imposed by the problem constraints. The number of chromosomes affected is governed by the mutation rate that is kept constant throughout the problem.		
Non-Uniform Mutation[4.60]: adds a random real number to a randomly selected gene in a chromosome. This number is taken from a uniform distribution restricted to ever decreasing ranges depending on the generation count of the algorithm. The number of chromosomes affected is governed by the mutation rate that is kept constant throughout the problem.		
Scheduled Mutation: adds a random real number to a randomly selected gene in a chromosome. This number is taken from a uniform distribution restricted to ever decreasing ranges depending on a schedule. The number of chromosomes affected is governed by the mutation rate that decreases from 50% to 1% by generation number according to a hand tuned schedule.		

Table 4.5.2: Some Mutation Techniques Applicable to Real Coded Genetic Algorithms

The probability of mutation (or amplitude of noise) is usually kept very low, in the order of 0.5% to 2%. Low rates of mutation help the optimiser refine the final solution, reaching better outcomes over time. High mutation rates cause disruption and tend to damage good solutions, thereby hampering convergence. While a zero mutation rate would limit the optimiser's convergence asymptote by the diversity of the initial genetic pool. There again are a number of ways in which mutation can be applied in GAs. **Table 4.5.2** lists but a few. For a more detailed comparison see [4.57]. In this work, scheduled mutation was used because of its flexibility. **Table 4.5.3** shows a typical mutation schedule being used for trajectory optimisation problems.

Gen.	Rate	Magnitude
0+	10%	0.95
10+	8%	0.96
50+	6%	0.97
100+	4%	0.98
150+	2%	0.98
200+	1%	0.99

Table 4.5.3: Mutation Schedule

4.5.7 Speciation

One of the ways of safeguarding against diversity collapse is to maintain separate populations. This results in speciation since *genetic drift* causes different genes to be lost or magnified when evolution goes down different paths in isolation. This is a well known effect in biological systems that is observed when organisms are marooned on separate islands. With fitness sharing [4.64], each species settles at a separate local optimum. This is analogous to organisms adapting to separate niches in an ecosystem.

However, in a multi-objective problem, if selection pressure favours one objective over another, the subspecies evolves to optimise that objective. For a large multi-objective optimisation problem, this can be done with each objective in turn having a separate population biased towards each objective [4.65], thereby increasing overall diversity by virtue of isolation. Genetic material from each subpopulation can then be crossed over to generate new individuals with separately optimised traits. An alternative involves migration, where a few individuals from the subpopulations are swapped at random.

A well known related technique (NSGA-II) proposed in 2002 by Deb et al. [4.30] uses non-dominated sorting to avoid diversity collapse for multi-objective problems. This type of sorting favours Pareto optimal solutions and ranks all other population members into concentric Pareto shells, thus removing all selection pressure between Pareto optimal solutions. It uses an elitist approach which preserves Pareto optimal members from one generation to the next. New members are generated by crossing over members lying on the Pareto front, which intrinsically guarantees complementary diversity of the parents.

4.5.8 Coevolution

In CATM, the problem arises on how to strike a balance between competing flights vying for the same airspace. Optimising one flight at the expense of another is not a good option. A global solution is desirable such that minimizes the cost (or maximises fitness) across the entire fleet. A logical way of attempting to reach such a solution is by concatenating the genetic material representing each flight into a single super-chromosome. This would conceptually automatically deal with minimising overall cost by evaluating fitness over the whole fleet. The problem with this approach is one of dimensionality. The resulting exponential growth of the search space dilutes the GA efforts to the point of zero progress. The GA will then fail to converge (sensibly) due to the lack of coverage.

It turns out that a very effective technique to deal with high dimensionality can again be drawn from nature [4.66]. Different species in nature share the same environment and thus tend to affect each other's evolutionary path, particularly if there is a predator-prey relationship between them. A change in one species is soon reflected with a corresponding counter change in the neighbouring species, which may in turn re-affect the first species and so on. This was recognised by Danny Hillis [4.67] of MIT in 1990 who developed the first co-evolutionary algorithm for use on his 65,536 processor Connection Machine[®] (CM1). The technique was successfully applied to the search of efficient sorting networks and was adapted by Husbans [4.68] for solving harder variants of job shop scheduling.

4.5.8.1 Cooperative Coevolution

Cooperative co-evolution is an improved technique for dealing with high dimensionality [4.34],[4.36], [4.69], which again makes a strong case for speciation. Each species represents and optimises a different portion of the super-chromosome. This leads to good results on condition that highly interacting genes are co-located within the genome of the same species. When information about interaction is not known, or the problem is not easily separable, random gene grouping has been proposed with very good results (CCPSO-1,2 [4.70], [4.71]). This allows the GA to search for any interaction between genes.

In ATM the boundaries between aircraft are fairly distinct, and the obvious starting point is to include all genes related to one aircraft into a single species, such that one aircraft is represented by one population. The effectiveness of this approach was demonstrated by Gao et al. [4.72], who used coevolution to efficiently resolve pathological multi-way air traffic conflicts.

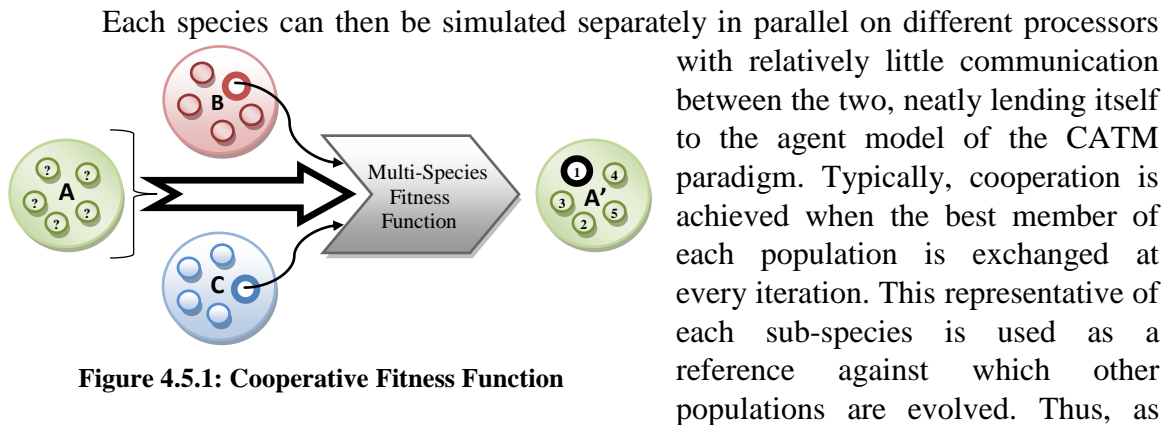


Figure 4.5.1: Cooperative Fitness Function

shown in **Figure 4.5.1**, each member of species ‘A’ is evaluated by feeding it into a multi-species fitness function together with the current best members from every other species, (B and C). The fitness function is designed to give maximum credit to *cooperation* between members of the species that result in the most efficient, lowest-conflict set of flights. This same is performed by every other species.

4.5.9 Clustering

With multi-species GAs using cooperative co-evolution, distributed computing requirements can be conveniently met by evolving each species on hardware located in the respective cockpit. However, when many aircraft are involved, the combinatorial interaction between species leads to a large quadratic problem. This may be alleviated by limiting the interaction by dividing grouping interacting flights into separate clusters. This leads to good results except when strongly interacting flights end up in separate clusters, resulting in unresolved conflicts.

Very recently, Zhang et al. [4.73], proposed a coarsely grained scheme that dynamically detects conflicts as they appear at every iteration, and groups the conflicted flights into isolated subgroups [4.74]. The flight variables are concatenated into super-chromosomes for optimisation with a simple GA. The unevenly coagulated airspace is

allowed to evolve. However, the separate groups will easily generate new conflicts with flights from other groups and this is where cooperative co-evolution is used to ward off these secondary conflicts. Neighbouring groups routinely share their best solutions which are then used as fixed constraints by their neighbours.

Here we propose a different fine-grained technique that involves creating one group of species (flights) for every flight in the system. This results in as many clusters as there are flights and can therefore be used in a grid avionics CATM system where each aircraft is responsible for its trajectory. The highly overlapping clusters share information implicitly by virtue of their overlap. Static clustering is preferred to dynamic clustering as it gives very similar results using far less computational effort, and bears no risk of unstable oscillator behaviour. Interaction can be conveniently predicted to a high degree at the very start by calculating the distance of closest approach using the end points of each flight. By including the flights that are most likely to interact into each cluster, the quadratic problem that arises when searching for interaction, can be kept small. Dynamic clustering can then be used for “finishing touches” to resolve the rare remaining case of undetected interaction.

Cooperative coevolution with fine grained clustering draws many parallels with simulated annealing or the crystallisation of solids. These are reliable processes that are guaranteed to converge with mathematical proofs to this effect [4.75]. However, unlike simulated annealing, cooperative co-evolution can be applied in ways (discussed later) that reach a dynamic equilibrium. This allows the optimiser to rapidly adapt to changing conditions. A simple demonstration can be set up using a space filling algorithm. A number of agents placed in a confined space are tasked with filling all available space evenly by maximising the distance between nearest neighbours. Each agent only interacts and measures its distance from 10 or so nearest neighbours and the cluster overlap does the rest.

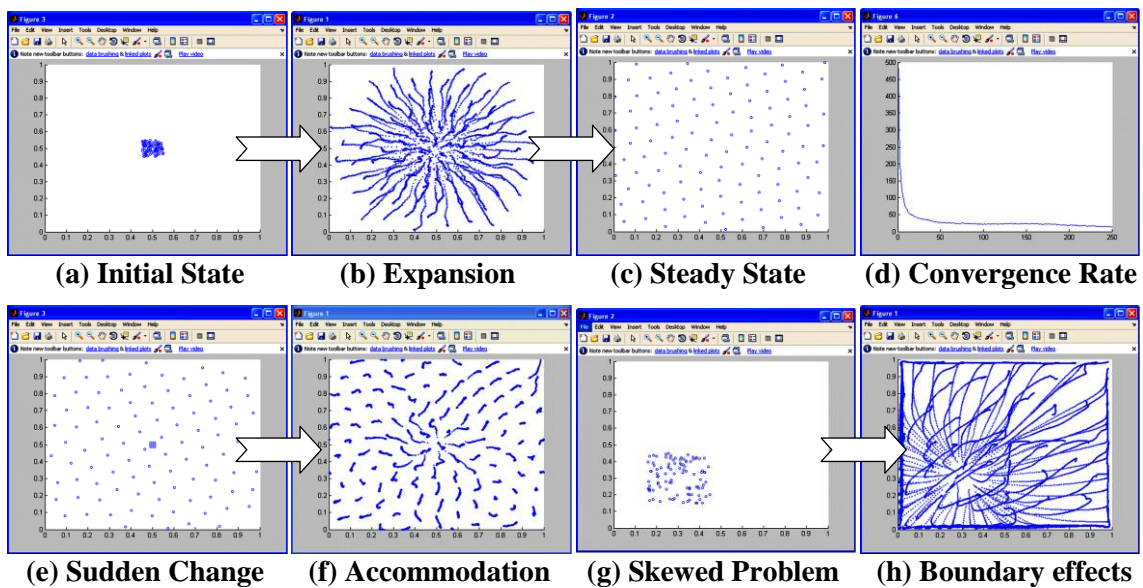


Figure 4.5.2: Simple Demonstration of Clustering Effectiveness with Dynamic Behaviour

The test problem is initialised by placing one hundred agents in random close proximity in the middle of a given 2D space as shown in **Figure 4.5.2 (a)**. The 2D space is topologically toroidal and hence wraps around indefinitely without the need for explicit boundaries. Each agent assumes that its nearest neighbours' current position is fixed and adjust its own position using an electrostatic repulsion function that generates a force in a direction that will maximise separation from the neighbours. Each agent observes the same simple rules of engagement and while none of them is aware of the full picture, the swarm expands as in **(b)** to find the lowest energy state in the system as in **(c)**. A convergence metric, proportional to the energy in the system, is plotted in **(d)**.

The system is now in dynamic equilibrium at its lowest energy state with the agents assuming a lattice-like positioning. The dynamic nature of the equilibrium is demonstrated by placing an additional batch of agents in the middle of the converged system as in **(e)**. The system quickly reacts to the sudden change and proceeds to disperse the new energy impulse as in **(f)**. In **(g)** and **(h)**, the system is shown behaving correctly in the presence of boundary constraints and an asymmetric start. This kind of behaviour is highly desirable in an ATM system and will be our objective.

4.5.10 Diversity and Dynamic Behaviour of GAs

If the initial population constituting the species is sufficiently large and diverse, crossover is alone able to generate such a rich variety of individuals, that together with the selection operator, results in fairly rapid optimisation convergence. This places great emphasis on the entropy present in the original population. In the absence of better knowledge, populations are thus initialised randomly to maximise diversity and avoid skewing the process in ways that might be detrimental down the line.

In biology, this is studied in the realms of population genetics, and is the mechanism that allows a species to rapidly adapt to new circumstances at the expense of some diversity. Over the course of just a few thousand generations, humans have exploited this technique to breed an impressive variety of dogs by decomposing the diversity naturally contained in the populations of wolves, coyotes and hyenas.

The downside of this process is that the resulting sub species are significantly less diverse or resilient than the original populations and this means that they are much less able to adapt to new circumstances unless combined with other existing sub species containing complementary traits. If the diversity is lost, it becomes very difficult to generate new combinations by crossover, and the optimisation stalls. If the genetic pool becomes too restricted, the inbreeding merely swaps virtually identical genetic information which can hardly generate new individuals. This creates difficulties when optimising in highly non-convex search spaces, because premature diversity collapse limits the exploratory behaviour of the algorithm making it highly prone to get trapped inside local minima. Over-reliance on the mutation operator is misguided as low mutation rates are unable to generate sufficient diversity once this is lost. On the other hand, high mutation rates are highly disruptive and hamper convergence.

Diversity collapse is particularly relevant for dynamic optimisation in which the goal posts keep changing from time to time. Unless diversity is maintained, optimisation by crossover quickly stagnates and is then unable to track changing optima. Adding new random individuals to a converged population is not very effective either. The large differences in fitness quickly works to eliminate the new individual and any offspring it might create, making its contribution very short lived. Diversity management is a challenging problem in evolutionary algorithms and is one of the main reasons why GAs are typically assumed to be one-way optimisers.

The trivial method of achieving dynamic optimisation is by frequently resetting the population and restarting the whole process. This essentially guarantees good dynamic behaviour. However, the down side of this approach is that no information is carried over between restarts making the process approach the inefficiency of a random search. This makes it prohibitively expensive.

A second technique has already been mentioned and involves maintaining numerous small populations in several niches located in the non-convex search space. This is achieved using fitness sharing [4.64].

4.5.11 Crossover Scheme Effects

Multi-parent multipoint crossover was considered for this work on the basis of some tests that indicate that two parents for crossover is not necessarily the best option. **Table 4.5.4** shows a selection test results that highlight this finding.

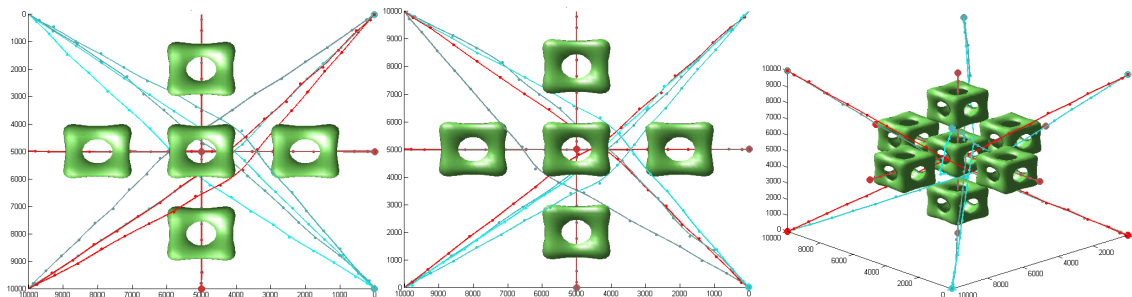


Figure 4.5.3: GA Optimisation of Pathological Non-Convex case with Obstacles

A pathological multi trajectory test with obstacles was used as a reference problem exhibiting highly non-convex behaviour. This involves ‘aircraft’ departing from all eight corners of a cube to intersect at the middle and reach the opposite corner. Six additional flights from the middle of opposing faces join the rest. Left to their own devices, this would result in a 14 way mid air collision at the centre of the cube. The objective is finding the shortest path across the cube while keeping adequate separation from all other flights. This is a particularly difficult problem since it is symmetric and also highly dynamically non-convex. Six cubes surrounding the intersection point, with circularly open faces, were added to further increase the complexity. The total of the distances travelled per aircraft serving as a metric for minimisation. The converged result is shown in **Figure 4.5.3**. Elitism (5 member) and idealism is being used in this example.

Gen.	Population Size	Trajectory Segments	Crossover Points	Sexes	Time (s)	Distance Minimisation Metric \pm Stderr
200	2000	200	11	3	269.36	1.96 \pm 4.83%
200	200	2000	11	3	263.89	1.91 \pm 0.08%
200	200	200	11	3	27.71	1.95 \pm 0.28%
200	200	200	11	6	27.78	1.95 \pm 0.49%
200	200	200	11	2	27.46	3.59 \pm 78.11%
200	200	200	21	11	35.71	4.75 \pm 50.82%
200	200	200	21	2	35.73	14.88 \pm 16.49%
200	200	200	9	5	28.27	1.97 \pm 0.06%
200	200	200	9	2	26.22	1.96 \pm 0.15%
200	200	200	5	3	23.64	1.97 \pm 0.31%
200	200	200	5	2	23.50	1.97 \pm 0.23%
200	200	200	5	6	23.29	1.97 \pm 0.18%
200	200	200	2	3	20.66	65.55 \pm 9.35%

Table 4.5.4: Effect of Multipoint Crossover and multiple sexes

Oligomers are short segments of chromosomes. Thus there is always one more oligomer than there are crossover points. When there are more oligomers than there are sexes, the contributions of the sexes are selected in a round robin fashion. The first two tests were used to establish the problem's convergence asymptotes: one with a very large population and another with a large number of segments. The tests were run 20 times and the table shows the final result and time taken to complete 200 generations. The standard deviation (expressed as %) is given as a measure of repeatability.

It is clear from these results that multiple crossover points are beneficial up to a point. Beyond 11 crossover points is detrimental to both outcome and iteration rate. The probable answer lies in building block theory [4.56], and is likely due to the short oligomers being unable to represent full schema. Too few crossover points is also detrimental probably due to insufficient diversity of parental contribution. The choice of the number of sexes is also not a foregone conclusion, as it has little effect in some cases and substantial effect in others. However, it appears that having more than two sexes is beneficial to the algorithm.

4.5.12 Population Size Effects

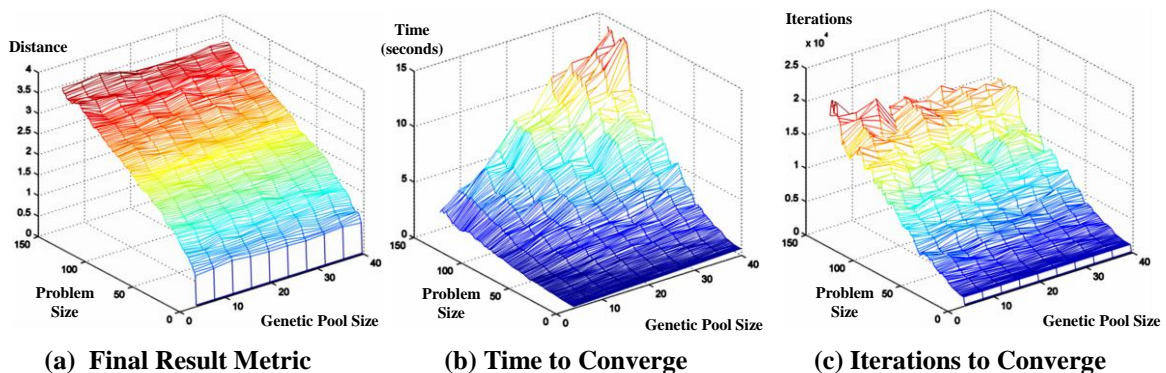


Figure 4.5.4: The effect of Genetic Pool Size on convergence Time and Outcome

The size of the population (genetic pool) is one parameter that has substantial repercussions on the performance of the algorithm. A small parametric test was conducted for a well known NP-hard travelling salesman problem for a range of problem sizes (10 to 150 aircraft) and a range of population sizes (5 to 40) test. The number of nodes per trajectory was kept small (~10) to reduce computation time. Results are shown in **Figure 4.5.4**. Small populations require far less computation resources to maintain. This results in a fast iteration rate and will result in far more exploitation in any given execution time. However, the power of genetic algorithm, when solving non-convex problems, lies in having a large population that suitably samples the search landscape. The larger the population, the better the exploration, but the slower the iteration rate. This is counteracted by somewhat higher improvement per iteration.

4.5.13 Elitism and Idealism effects

Idealism and elitism have complementary effects on the convergence behaviour of the GA. Elitism has negligible effect on the initial convergence rate however it makes a big difference during the final exploitation phase of the algorithm. The ability to retain in memory a group of top performing solutions, ensures that the GA never lapses to lesser solutions, but continuously ratchets for better results. This is very apparent in **Table 4.5.5**, which shows very pronounced improvement in steady state behaviour as elitism is introduced. The effect seems to saturate and does not exhibit any improvement beyond having 15 elitist members.

Idealism is implemented by always making the naive solution available for crossover, bypassing the fitness function and the normal selection process. It gives the GA a shortcut by making available a fairly good set of schemata for admixing with other parents, and indeed results in faster initial GA convergence by wasting less time exploring highly suboptimal solutions. This can be observed in **Figure 4.5.5** which shows the convergence plots of all the trials in the table. The plots clearly cluster into two batches. However, idealism is, on the other hand, markedly detrimental to the final steady state solution, since the naive solution tends to distract the GA from progress beyond the ideal. However, this can be easily solved by turning it off after it completes its role during the initial stages.

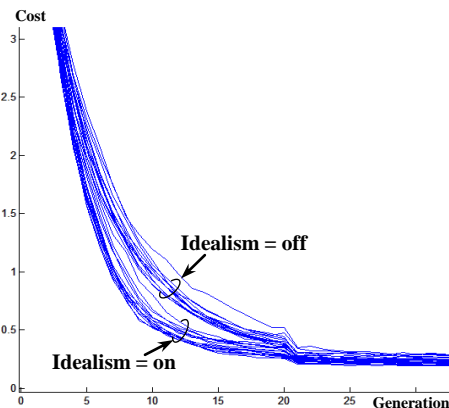


Figure 4.5.5: The effect of idealism

Gen.	Time (s)	Elitism	Idealism	Final Systemic Inefficiency (% ± Stderr)
100	25.85	25	On	5.7% ±11.4% ²
100	25.99	25	Off	4.9% ±12.1% ²
100	26.46	15	On	5.4% ±8.1% ²
100	26.51	15	Off	5.4% ±11.2% ²
100	26.50	5	On	9.4% ±8.1% ²
100	26.57	5	Off	6.6% ±8.7% ²
100	24.98	0	On	28.4% ±8.7% ²
100	24.85	0	Off	17.6% ±9.0% ²

Pool = 200; Segments = 21; Crssvr pts = 11; Sexes = 6, N = 42

Table 4.5.5: Effects of Elitism and Idealism

4.6 SWARM INTELLIGENCE

Swarm intelligence (SI) is another bio-inspired class of optimisation schemes [4.12]. They are similar to GAs in that they are population based. However, the update mechanism is markedly different from evolution and relies instead on mimicking the behavioural traits of social animals and insects. The intricate information processing network formed by a collection of like-minded creatures results in *emergent behaviour* that benefits the swarm in terms of survival. This survival advantage is of course an attribute that is itself the result of millions of years of evolution. Therefore, swarm intelligence can be seen as the result of nature's attempt at meta-optimisation, also called hyper-heuristics. Meta-optimisation is the process of using an upper-tier of optimisation to design and enhance the performance of another lower-tier optimiser [4.76]. If millions of years of natural selection converged on swarm intelligence as the instrument of choice for solving difficult problems, such as foraging for food or building adaptive immune systems, nature makes a strong case to mimic these algorithms to solve engineering problems – such as transport. Several studies indeed show that SI outperforms EAs in many non-convex problems [4.77].

Swarm intelligence is oftentimes the unintended consequence of ultimately egocentric behaviour of the many members making up a society. It results in a decrease in entropy, where order emerges from disorder. This is a characteristic of many life forms including, ants, termites, bees and indeed humans! One of the best and most succinct descriptions of the underlying philosophy behind swarm intelligence is given by Engelbrecht in his book on the subject:

"No individual has to accomplish much or understand the whole problem, but collectively they can perform incredible engineering feats." [4.21]

The field of artificial swarm intelligence grew in popularity following seminal work by the Italian computer scientist, Marco Dorigo, who in 1992 completed his PhD thesis on Ant Colony Optimisation [4.78]. Dorigo drew his inspiration from work published by entomologist Jean-Louis Deneubourg who described "the self-organizing exploratory pattern of the Argentine Ant" [4.79]. Using a very simple path planning

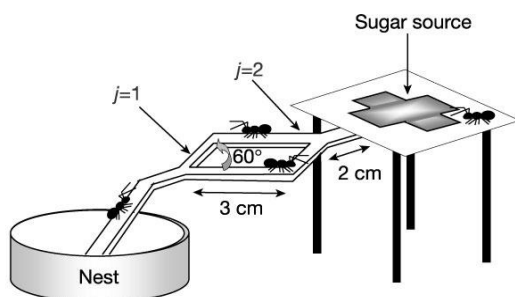


Figure 4.6.1: Argentine Ant Experiment [4.79]

experiment (Figure 4.6.1), Deneubourg showed that the ants were consistently capable of determining the shortest path towards food. The social insect colony behaved like a super-organism, which although lacking a central nervous system, or any intelligence on an individual basis, was capable of complex, collective behaviour. Dorigo emulated the behaviour and communications medium between the ants using *digital pheromone* trails and before

long he was able to solve the well known (NP-complete) travelling salesman problem, using the Ant Colony Optimisation (ACO) algorithm. The algorithm and its variants had a number of other successes including applications in network packet routing.

Over the span of a few years, such was the interest kindled by swarm intelligence that several other bio-inspired meta-heuristics were proposed and successfully tested. Each metaphor presents unique strengths and weaknesses and different balance between exploration and exploitation, and matching the application to the algorithm remains an important consideration. The following are but a few of the more popular schemes:

- Ant Colony Optimisation (ACO) [4.78]
- Artificial Bee Colony (ABC) [4.80]
- Artificial Immune Systems (AIS) [4.81]
- Bat algorithm (BA) [4.82]
- Firefly Algorithm (FA) [4.77]
- Glow-worm Swarm Optimisation (GSO) [4.83]
- Particle Swarm Optimisation (PSO) [4.84]
- River Formation Dynamics (RFD) [4.85]

Selecting an appropriate metaheuristic for CATM requires a focussed understanding of what is required to solve it. Essentially any metaheuristic solution of the CATM problem reduces to these main operations:

- OP-1) First, the problem needs to be discretized. This maps a continuous domain onto a discrete graph with enumerable nodes which can be searched algorithmically.
- OP-2) The second step involves evaluating each node and path in the graph to allow searching it in an informed manner.
- OP-3) The third step involves down-selecting the most promising search areas such that future search effort can be better directed.
- OP-4) The last step involves updating the graph to re-discretise the problem with better focus around these areas to gradually increase the resolving power of the search.
- OP-5) The process repeats in an iterative manner.

With this in mind it is immediately apparent that algorithms like ACO are inappropriate because they have no inbuilt ability of updating the graph they are searching. Given the continuous real valued nature of trajectory optimisation, an algorithm that lends itself well to continuous variables is also required. Other algorithms like ABC, GSO or FA spread their resolving power too thinly to allow meaningful progress in a high dimensional problem. The CATM problem is also highly non-convex with many locally-convex local minima, therefore a particular balance is required between initial exploration to reliably seek the location of the global optimum and final exploitation to give a clean result that is well contained in the global optimum well. PSO seems particularly adept in this regard with good computational efficiency and guaranteed convergence characteristics.

However, the candidate algorithm must also exhibit good dynamic behaviour in its converged state, in order to track and monitor the time-variant search landscape. ACO, ABC, GSO and FA have some strength in this regard. However, none of these algorithms are well suited in their standard form to deal with the real-time nature of this high dimensionality application. Some modification is required.

4.6.1 Particle Swarm Optimization (PSO)

PSO was developed by James Kennedy (a social psychologist), and Russell Eberhart (and electrical engineer) in 1995 as a method for simulating social behaviour but its ability for robust global optimization was soon recognized [4.86]. It uses a swarm of particles to loosely mimic herding/flocking/schooling behaviour of a number of organisms, but the similarities end there. In standard PSO [4.87], each particle is modelled as a moving point mass that dynamically scans the problem domain in D dimensional space while sampling the objective function and recording the resulting cost history along its path.

The particles are made to move in a damped oscillatory fashion towards updatable attractors that consist of the best solution found thus far. (See **Figure 4.6.2**) This performs very thorough scanning of the neighbourhood of good solutions which greatly improves the exploitation phase of the algorithm. At the same time, the quasi-random motion of the swarm provides good coverage of the entire search space, thereby

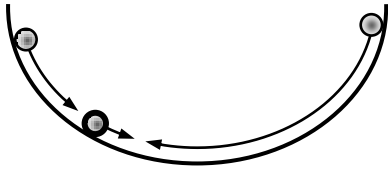


Figure 4.6.2: Damped SHM in PSO among other things.

efficiently identifying most if not all of the local minima. If designed properly the damped simple harmonic motion (SHM) of the particles will rapidly converge onto the global optimum with arbitrarily high probability, depending on the size of the swarm,

For every dimension d and iteration i , the lowest cost point found thus far along the trajectory of each particle j at is called the *personal best* point or P_{Bi}^{jd} . The lowest cost point found thus far by its two nearest neighbours is called the *local best* point or P_{Li}^{jd} . Finally, the lowest cost point found thus far by all particles in the swarm is called the *global best* point or P_{Gi}^d . Every dimension is dealt with totally independently in PSO.

The particles move about the search space according to a position and velocity update equation. However, there are two variants of PSO. *Gbest* PSO uses the velocity update equation 4.6.1 and offers fast convergence characteristics, while *Lbest* PSO uses the velocity update equation 4.6.2 and offers better exploratory behaviour at the expense of convergence rate. Both variants use the same position update equation 4.6.3.

$$\text{Gbest velocity: } \dot{x}_{i+1}^{jd} = m\dot{x}_i^{jd} + c_1\epsilon_1^d(P_{Bi}^{jd} - x_i^{jd}) + c_2\epsilon_2^d(P_{Gi}^d - x_i^{jd}) \quad (4.6.1)$$

$$\text{Lbest velocity: } \dot{x}_{i+1}^{jd} = m\dot{x}_i^{jd} + c_1\epsilon_1^d(P_{Bi}^{jd} - x_i^{jd}) + c_2\epsilon_2^d(P_{Li}^d - x_i^{jd}) \quad (4.6.2)$$

$$\text{position: } x_{i+1}^{jd} = x_i^{jd} + \dot{x}_i^{jd} \quad (4.6.3)$$

where: m is an inertia / mass parameter that modulates sensitivity to velocity updates
 c_1, c_2 are constants with a value around 2.0
 $\epsilon_1^d, \epsilon_2^d$ are independent random numbers separately generated at every update
 x_i^{jd} is the position of particle j at time step i along dimension d
 \dot{x}_i^{jd} is the velocity of particle j at time step i along dimension d (4.6.4)

4.6.2 Stability and PSO Constriction

Since particle motion in Standard PSO (SPSO) resembles a harmonic oscillator, stability is an important matter related to the damping factor or rate of energy loss. As per the previous equations, the particles gain velocity in relation to their distance from the attractors and this may result in very high velocities that can propel particles far out of the feasible zone. If coefficients m , c_1 and c_2 are chosen carelessly, the rate of energy loss might turn negative, resulting in explosive divergence rather than convergence. Early empirical observation had shown that reliable operation was associated with $m = 1$ and with an attraction coefficient $\varphi = (c_1 + c_2) \cong 4$, but $\varphi \neq 4$, as discussed by Eberhart in [4.88].

The underlying particle dynamics were studied at some length by Clerc in 2002 and using a linear state-space representation of a deterministic version of PSO, much better insight on stability was obtained [4.39]. It turns out that the attraction coefficient φ is related to the Eigen values of the system as shown in equations 4.6.5:

$$\text{PSO Eigen values: } \begin{cases} e_1 = 1 - \frac{\varphi}{2} + \frac{\sqrt{\varphi^2 - 4\varphi}}{2} \\ e_2 = 1 - \frac{\varphi}{2} - \frac{\sqrt{\varphi^2 - 4\varphi}}{2} \end{cases} \quad (4.6.5)$$

Further analysis shows that when $\varphi < 4$, (when the Eigen values are complex), the particle motion tends to be periodic, and circular on the complex phase plane and spiral slowly towards the attractors, while when $\varphi > 4$, there is no periodic behaviour, Eigen values are real, and in fact convergence is fast and nearly linear, but if $\varphi \gg 4$, the rate of convergence deteriorates to the point of instability. Moreover, when $\varphi \cong 4$, there is very slow convergence, and when $\varphi = 4$ exactly, there is no convergence whatsoever. In summary, Clerc defines a stability relationship in the form of constriction parameter χ (4.6.6) which is used to modify the velocity update equations as shown in (4.6.7), (4.6.8)

$$\text{Constriction: } \chi = \frac{2}{|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}|}, \varphi = c_1 + c_2 \quad (4.6.6)$$

$$\text{Gbest velocity: } \dot{x}_{i+1}^{jd} = \chi \cdot [m\dot{x}_i^{jd} + c_1\epsilon_1^d (P_{Bi}^{jd} - x_i^{jd}) + c_2\epsilon_2^d (P_{Gi}^d - x_i^{jd})] \quad (4.6.7)$$

$$\text{Lbest velocity: } \dot{x}_{i+1}^{jd} = \chi \cdot [m\dot{x}_i^{jd} + c_1\epsilon_1^d (P_{Bi}^{jd} - x_i^{jd}) + c_2\epsilon_2^d (P_{Li}^d - x_i^{jd})] \quad (4.6.8)$$

where:

$$\begin{aligned} m &\text{ is now taken as 1.00} \\ \varphi &> 4 \text{ but not much larger for guaranteed fast convergence} \\ &\text{and typically } c_1 = c_2 = \varphi/2 \end{aligned} \quad (4.6.9)$$

The position update equation remains unchanged and using this simple modification, much of the trial and error parameter tuning that is so typical of metaheuristics is obviated. Thus a good choice for $\varphi = 4.1$ as this provides the best balance between exploration and fast reliable convergence.

4.6.3 Ideal Attractors (GPSO)

Similar to the approach mentioned in Section 4.5.4.2 and 4.5.13 with respect to GAs and idealism, the same ideas can be transposed to PSO in the form of ideal naive attractors. This can be done by modifying the velocity update equation quite simply by introducing a new term that measures the distance between each particle position x_i^{jd} and the fixed ideal attractor P_I^d as shown in equations 4.6.10 or 4.6.11. The constriction term is now based on an updated attraction coefficient $\varphi = (c_1 + c_2 + c_3)$ which must still exceed 4.

$$\begin{array}{l} Gbest \\ \text{velocity:} \end{array} \quad \dot{x}_{i+1}^{jd} = \chi \cdot \begin{bmatrix} m\dot{x}_i^{jd} + c_1\epsilon_1^d(P_{Bi}^{jd} - x_i^{jd}) \\ + c_2\epsilon_2^d(P_{Gi}^d - x_i^{jd}) \\ + c_3\epsilon_3^d(P_I^d - x_i^{jd}) \end{bmatrix} \quad (4.6.10)$$

$$\begin{array}{l} Lbest \\ \text{velocity:} \end{array} \quad \dot{x}_{i+1}^{jd} = \chi \cdot \begin{bmatrix} m\dot{x}_i^{jd} + c_1\epsilon_1^d(P_{Bi}^{jd} - x_i^{jd}) \\ + c_2\epsilon_2^d(P_{Li}^d - x_i^{jd}) \\ + c_3\epsilon_3^d(P_I^d - x_i^{jd}) \end{bmatrix} \quad (4.6.11)$$

where:

$$\chi = \frac{2}{|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}|}, \varphi = c_1 + c_2 + c_3 \quad (4.6.12)$$

where:

$$\begin{array}{l} m \text{ is now taken as } 1.00 \\ \varphi > 4 \text{ but not much larger for guaranteed fast convergence} \\ c_1 = c_2 = (\varphi - c_3)/2 \\ \text{and } c_3 \text{ can be quite small eg: } 0.01 \end{array} \quad (4.6.13)$$

Introducing fixed ideal solutions into the swarm has a much bigger positive effect in PSO than it has in GAs. This is because unlike in GAs, PSO particles are not being crossed over. There is no portion of the naive attractor that is overwriting portions of the particles in the swarm. Swarm particles retain their freedom to take into account any constraint penalty functions imposed by the problem (and ignored by the ideal attractor). Thus, even when highly converged, the ideal attractor continues to contribute positively and does not disrupt the best particles other than attracting them to the ideal point in the search space.

The attraction coefficient c_3 , must be kept low enough to ensure that the exploratory behaviour of the swarm is not unduly hampered, but must still be

Algorithm	m	φ	c_1	c_2	c_3
SPSO	0.9375	4.100	2.0500	2.0500	0
SPSO	1.0000	4.117	2.0585	2.0585	0
GPSO	1.0000	4.117	2.0000	2.0000	0.117
GPSO	1.0000	4.117	2.0500	2.0500	0.017

Table 4.6.1: Constricted PSO Attraction Coefficients

high enough to create a gentle drift in the swarm towards the region surrounding the global optimum. We refer to this variation of particle swarm optimisation as Guided PSO (GPSO) and given the very substantial improvement it affords, this forms the basis of all subsequent work. **Table 4.6.1** shows some of the coefficients that give good results with PSO and GPSO.

4.6.4 Particle Initialisation

In order to focus on the performance and characteristics of the distributed PSO optimisation algorithm, the simplest possible 3-DOF dynamic model was adopted to avoid distracting the investigation with the computational penalty of observing nonlinear fixed-wing aircraft constraints. This is fair enough given that the methodology for including model dynamics is well established and need not be investigated further in this work.

In addition, since optimisation is being anyway conducted in the output space, this choice has little impact. The chosen model is given as follows in equations 4.6.14, where $\mathbf{x}(t)$ is the state vector, $\mathbf{u}(t)$ is the input velocity vector, $\mathbf{y}(t)$ is the output position vector and m is a constant of proportionality, typically chosen to be unity.

$$\begin{aligned}\dot{\mathbf{x}}(t) &= m \cdot \mathbf{u}(t) \\ \mathbf{y}(t) &= \mathbf{x}(t)\end{aligned}\tag{4.6.14}$$

$$\begin{aligned}\mathbf{x}_{k+1} &= \mathbf{x}_k + m \cdot \mathbf{u}_k \\ \mathbf{y}_k &= \mathbf{x}_k\end{aligned}\tag{4.6.15}$$

In this simplified model, the linear one-to-one relationship between the output and input implies that there is little difference whether \mathbf{u}_k or \mathbf{y}_k are taken as optimisation variables. However, using \mathbf{u}_k , allows the terminal constraints to be enforced by simply scaling the input in every dimension such that the sum of inputs matches the difference between the terminal constraints (like in eq. 4.5.1). The resulting \mathbf{y}_k can then be mapped to the input domain of the fixed-wing model using the coordinated flight assumptions.

4.6.5 Upgrading PSO for Dynamic Behaviour (GPSO-D)

In GAs a ratcheting mechanism can be introduced with the use of elitism that ensures that the best fitness found is never lost, but rather becomes the benchmark for subsequent progress. This is highly beneficial to GAs as was shown in section 4.5.13. In SPSO the same technique is implicitly embodied by the *Pbest*, *Gbest* and *Lbest* position records that correspond to the best results found by the swarm.

However, these techniques introduce a problem when faced with dynamic changes in the search space. Any change in the problem's goal posts invalidates the recorded fitness at *Pbest*, *Gbest* and *Lbest*, but these will only be replaced if better solutions are found. Yet in the meantime the algorithm continues to be attracted to these positions and in doing so is unlikely to find any better solutions than those associated with the expired attractors. This causes SPSO to stall and is unable to track changes in the search space.

Luckily with PSO this is easily resolved by regularly re-evaluating the fitness of *Pbest*, *Gbest* and *Lbest* with respect to the most recent objective functions and constraints. This effectively doubles the computational effort per iteration since there are as many *Pbest* or *Lbest* values as there are particles in the swarm. However, this can be greatly mitigated by conditionally updating *Pbest* or *Lbest* depending on whether any decreases in *Gbest* are detected. This makes the computational implications of making PSO dynamic, nearly negligible since decreases in *Gbest* are fairly rare. In the

meantime, the ideal attractor need only be re-evaluated if there are any changes in the endpoints of any trajectory. These small modifications make PSO or GPSO capable of adapting to dynamic problems such as CATM. We thereby label this variant of PSO as GPSO-D.

4.6.6 Upgrading PSO for High dimensionality (CCGPSO-D)

In order to solve multi trajectory problems the dimensionality is multiplied by the number of aircraft, with the associated exponential rise in search volume. PSO is no better than GA or any other metaheuristic when it comes to dealing with large search spaces. It is a fundamental problem after all.

However, like GAs, a divide and conquer approach may be employed to make the problem tractable. The search space may be divided into multiple swarms, one per group of dimensions. Dimensions are grouped by flight. The ideas of cooperative co-evolution (CC) may be imported and applied to PSO [4.35], just as well as GAs [4.36]. Thus if each single flight employs a separate swarm of particles to optimise its trajectory, they can share their *Gbest* particles and use them as fixed trajectories in each other's sub-search spaces. Fitness or cost calculations in each of the sub-swarms take into account all of these fixed members. This process works remarkably well and there are some very good results in other high dimensionality contexts in the literature. [4.38], [4.40].

4.6.7 Fine Grained Clustering (C-CCGPSO-D)

The final addition to the modified PSO algorithm is clustering. Clustering of aircraft is essential for reducing the order of the problem on the local scale. This proceeds in a fashion that is very similar to that used with GAs. Aircraft are grouped according to their level of interaction. We call this final variant of PSO, Clustered Cooperatively Co-evolving Guided PSO with Dynamic behaviour or C-CCGPSO-D

4.6.7.1 Defining Interaction

Interaction is invariably based on the Euclidean distance between interacting aircraft. If the Euclidean distance, between two aircraft reduces below a certain threshold then those aircraft are automatically paired into a cluster. Other aircraft may join the same cluster in the case of multi-aircraft interactions. There are as many clusters as there are aircraft because each aircraft is necessarily at the centre of its own cluster.

The principle of clustering is central to solving the large scale problem. However, the real-link with the large scale problem comes through the intrinsic overlap that exists between clusters. There are several ways of defining interaction between trajectories;

- **Instantaneous Clustering:** The first method involves segregating flights into clusters depending on their instantaneous position with respect to each other. This implies that aircraft enter and exit clusters frequently and membership in a cluster becomes time-dependent. This would result in tactical behaviour similar to Collision Detection and Resolution (CD&R).

- **Dynamic Clustering:** A second method of defining a cluster involves collecting aircraft that interact (in the Euclidean sense) at any point along their 4D trajectories. This will in all likelihood involve a much greater number of aircraft per cluster but membership would be more stable throughout a flight, allowing for holistic optimization. There is also a greater degree of overlap and this type of overlap is much more meaningful to the global problem. This stems from the fact that in this case the overlap can occur along the entire time axis-therby linking parts of the problem that bear effect on each other over much greater time-scales. However, as optimisation progresses, the interaction changes as a result and this requires regular redefinition of the clusters which is computationally expensive. As a side effect, this could also result in oscillatory behaviour whereby interacting aircraft within a cluster are moved apart by the optimiser to the extent of them leaving each other's cluster, only to return to their initial positions that caused the interaction to begin with.

- **Static Clustering:** A third method involves clustering on the basis of expected interaction based on the closest proximity of the idealised straight lines joining the endpoints of each 4D trajectory. This offers still higher stability apart from simplifying calculations with respect to method 2, because cluster membership is not affected by the outcome of optimizations. The associated caveat is that some interactions could be underestimated, because the optimiser might push some trajectories far out of their expected bounds. This can be tested by evaluating the elongation of all trajectories. Trajectories that are within a few percent of their minimum lengths are safely predicted.

4.6.8 Multi-Swarms, Stagnation and Population Diversity Collapse

When only one swarm is used, the diversity collapse that frequently accompanies convergence would result in just one optimum being tracked – hopefully the global optimum. However, to better improve dynamic behaviour, multiple swarms can be used to track more than one local minimum at a time. This avoids the undesirable behaviour shown in **Figure 4.4.7 (d)**, where the global optimum gradually loses its status while it is being tracked. Multiple swarms are an excellent way to counteract stagnation by increasing diversity in a managed fashion.

4.6.9 A Hybrid: Iterative Dynamic Evolutionary Programming

Close observation of the way the particle swarm scans the search space reveals behaviour that is not too different from the multi resolution version of Dynamic Programming. The swarm gradually occupies smaller and smaller volumes, thereby increasing the resolution of sample points surrounding the global minimum. This appears to be a stochastic variant of IDR and one idea that is being mulled for future investigation is the combination of swarm intelligence with iterative dynamic programming.

This approach would join ranks with the growing field of the so-called *Matheuristics* [4.89], whereby complementary strengths of metaheuristics and mathematical programming techniques are combined.

4.7 HIERARCHICAL OPTIMISATION FOR CATM

Hierarchical optimisation is a technique that combines the strengths of local optimisers with those of global optimisers in a bid to reduce the computational effort and speed up to the time it takes for the duo to reach acceptable results. This was proposed number of times in the literature [4.90], [4.91] even in the aeronautics domain [4.92], [4.93], and is considered a gold standard approach.

4.7.1 Meta-Heuristics: Exploration and Exploitation

As was pointed out earlier, metaheuristics are always faced with the conflicting requirement of exploration v.s. exploitation [4.94]. Yet both activities are important to yield good results. The former searches for local minima (or maxima) while the latter measures their depth (or height). In order to negotiate the conflict, typically metaheuristics are designed to adaptively change focus during the optimisation run. However, this comes at a cost. As the optimiser restricts its view to smaller search spaces it is also increasingly less able to identify the emergence of new global optima outside its “field of view”. This damages its ability to adapt to dynamic scenarios.

4.7.2 Cascaded Optimisation

The classic approach to meet both requirements is to cascade a local optimiser after a global optimiser (**Figure 4.7.1**). This is also called High Level Relay Hybrids (HLH) [4.95]. The first run of the global optimiser locates the whereabouts of the global optimum and this approximate information is then passed-on to a local optimiser that refines the result by exploiting any gradient information present in the immediate vicinity of the approximate result. Clearly this assumes that the objective function is well behaved, smooth and differentiable in the neighbourhood of the global optimum. The global optimiser can therefore be spared the substantial time it takes for it to inefficiently search (in a probabilistic fashion) for the very best point in the global optimum neighbourhood. Local optimisers are much better suited for this task.

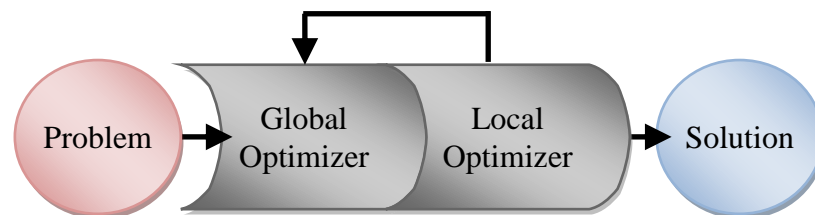


Figure 4.7.1: Cascaded Optimization

Cascaded optimisation is usually unidirectional and as such any changes in the problem would not be accommodated once a solution is found. However, dynamic behaviour can be re-introduced by including a feedback path between the local optimiser and the global optimiser, such that global optimiser iteratively tracks and reevaluates the state of the local optimiser and compares it to other freshly-found local optima.

4.7.3 Nested Optimization

Local optimisers use gradient information very efficiently to locate the minimum point within a locally convex neighbourhood, as long as they are seeded with an initial value that is well within the neighbourhood. This perfectly complements the strengths and weaknesses of the global optimiser that is typically good at finding approximate solutions but inefficient at exploiting them. Therefore, other very effective way of addressing this dilemma is to embed a local optimiser *within* the global optimiser as an “oracle” which it consults as shown in **Figure 4.7.2** [4.96]. This kind of symbiotic relationship is often called Low-level Teamwork Hybrid (LTH) [4.95].

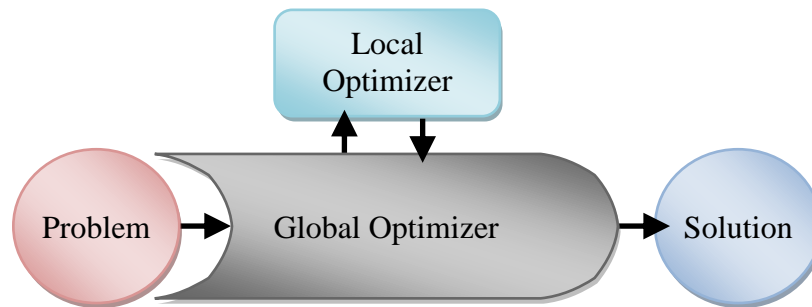


Figure 4.7.2: Embedded Local Optimization

If a local optimiser is included within the inner loop of the global optimiser, every exploratory sample point can be taken down to its limits to determine if the neighbourhood is worth exploring further. This can be implemented by replacing one of the population replacement operators in the metaheuristic. For instance, in GAs, a local optimiser can be used to replace the mutation operator to give much better improvements than random perturbation. This means that far less time is spent chasing dead ends to instead dedicate more effort looking in other promising areas. Such techniques are sometimes also called *memetic algorithms*:

Memetic algorithms are population-based metaheuristics composed of an evolutionary framework and a set of local search algorithms which are activated within the generation cycle of the external framework. [4.97]

This is a very appealing approach that deserves further investigation for CATM, considering that some impressive results have been achieved in this context [4.96]. It however requires a very stable and fast implementation of a local optimiser that can be easily interfaced with the upper-level global optimizer metaheuristic. Placing the local optimiser in the inner loop, implies millions of executions, thus the implementation of local optimiser must be streamlined before such tight coupling can be attempted.

4.7.4 Cooperative Local and Global Optimization

One other commonly used technique involves several parallel efforts from local optimisers and global optimisers selected for complementary performance, which regularly share or pool information such as upper or lower bounds on the partial results. This kind of collaboration is sometimes called a High-level Teamwork Hybrid (HTH).

4.8 LOCAL OPTIMISATION PHASE

Local optimisation of trajectories is a highly established subject based on discretized versions of variational calculus. Therefore the following will not expand on this subject beyond the few simple techniques that were employed as part of this work. For a detailed treatise on this subject the interested reader is referred to the surveys by Becerra [4.3] and Betts [4.5] and the book by Bryson et al. [4.1]. We will begin the section by defining a point mass model which will be used as a simple and clean placeholder for trajectory optimisation using NLP solvers. Without loss of generality we shall assume 2D space for this example, to simplify the diagrams and matrices.

4.8.1 A Simple Point Mass Model

For a simple point mass particle travelling in 2D space \mathbb{S} with a trajectory uniformly discretized in time as shown in **Figure 4.8.1** with predefined terminal constraints at both ends of the trajectory, a discrete time model can be expressed as shown in eq: 4.8.1 using a trapezoidal implicit difference equation. The terminal constraints are given in eq: 4.8.2 where \mathbf{x}_k is the position vector of the particle, $\dot{\mathbf{x}}_k$ the velocity and $\ddot{\mathbf{x}}_k$ the acceleration vector. \mathbf{u}_k is the force input, m is the mass, T is the time step and, \mathbf{y}_k is the output which is taken equal to the position vector \mathbf{x}_k .

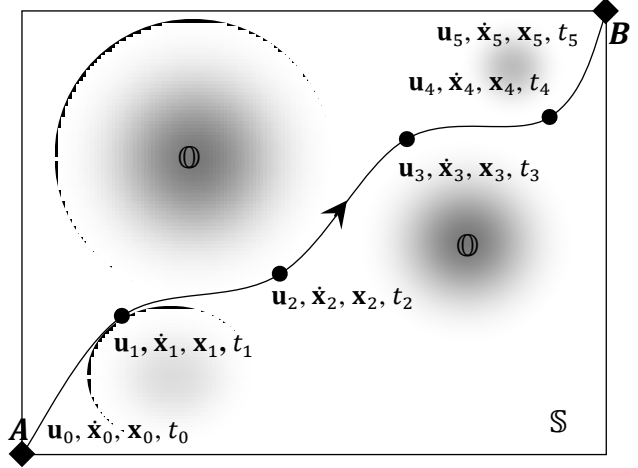


Figure 4.8.1: Trajectory of a Newtonian Point Mass

$$\begin{array}{l} \text{Point} \\ \text{Mass:} \\ \text{Model:} \end{array} \left\{ \begin{array}{l} \ddot{\mathbf{x}}_k = \frac{1}{m} \cdot \mathbf{u}_k \\ \dot{\mathbf{x}}_{k+1} = \dot{\mathbf{x}}_k + T \cdot (\ddot{\mathbf{x}}_k + \ddot{\mathbf{x}}_{k+1})/2 \\ \mathbf{x}_{k+1} = \mathbf{x}_k + T \cdot (\dot{\mathbf{x}}_k + \dot{\mathbf{x}}_{k+1})/2 \\ \mathbf{y}_{k+1} = \mathbf{x}_{k+1} \end{array} \right. \quad (4.8.1)$$

$$\begin{array}{l} \text{Initial:} \\ \text{Conditions:} \end{array} \left\{ \begin{array}{l} \mathbf{u}_0 = \mathbf{u}_i \\ \dot{\mathbf{x}}_0 = \dot{\mathbf{x}}_i \\ \mathbf{x}_0 = \mathbf{x}_i \\ t_0 = t_i \end{array} \right. \quad \text{Final:} \left\{ \begin{array}{l} \mathbf{u}_5 = \mathbf{u}_f \\ \dot{\mathbf{x}}_5 = \dot{\mathbf{x}}_f \\ \mathbf{x}_5 = \mathbf{x}_f \\ t_5 = t_f \end{array} \right. \quad (4.8.2)$$

The particle is required to negotiate the given landscape \mathbb{S} , by finding the lowest cost trajectory between A and B where the cost J_x is given as a function $\mathcal{F}(\bullet, \bullet)$ of the position of the particle and the of all the obstacles \mathbb{O} , as shown in eq. 4.8.3

$$J_x = \mathcal{F}(\mathbf{x}, \mathbb{O}) \Rightarrow J_{x_k} = \mathcal{F}(\mathbf{x}_k, \mathbb{O}) \quad (4.8.3)$$

4.8.2 Direct Collocation

We can formulate the problem into a form suitable for feeding into an NLP solver by writing down a series of expressions as per a strict format that poses the system's dynamics a series of constraints enforcing the relationship between the particle's position and its derivatives. For the six node problem described above, using eq 4.8.1, we iteratively write the following equality conditions to enforce trajectory and velocity continuity at each node:

$$\begin{aligned}
 \mathbf{x}_1 &= \mathbf{x}_0 + T(\dot{\mathbf{x}}_0 + \dot{\mathbf{x}}_1)/2 & \dot{\mathbf{x}}_1 &= \dot{\mathbf{x}}_0 + T(\ddot{\mathbf{x}}_0 + \ddot{\mathbf{x}}_1)/2 \\
 \mathbf{x}_2 &= \mathbf{x}_1 + T(\dot{\mathbf{x}}_1 + \dot{\mathbf{x}}_2)/2 & \dot{\mathbf{x}}_2 &= \dot{\mathbf{x}}_1 + T(\ddot{\mathbf{x}}_1 + \ddot{\mathbf{x}}_2)/2 \\
 \mathbf{x}_3 &= \mathbf{x}_2 + T(\dot{\mathbf{x}}_2 + \dot{\mathbf{x}}_3)/2 & \dot{\mathbf{x}}_3 &= \dot{\mathbf{x}}_2 + T(\ddot{\mathbf{x}}_2 + \ddot{\mathbf{x}}_3)/2 \\
 \mathbf{x}_4 &= \mathbf{x}_3 + T(\dot{\mathbf{x}}_3 + \dot{\mathbf{x}}_4)/2 & \dot{\mathbf{x}}_4 &= \dot{\mathbf{x}}_3 + T(\ddot{\mathbf{x}}_3 + \ddot{\mathbf{x}}_4)/2 \\
 \mathbf{x}_5 &= \mathbf{x}_4 + T(\dot{\mathbf{x}}_4 + \dot{\mathbf{x}}_5)/2 & \dot{\mathbf{x}}_5 &= \dot{\mathbf{x}}_4 + T(\ddot{\mathbf{x}}_4 + \ddot{\mathbf{x}}_5)/2
 \end{aligned}
 \tag{4.8.4}$$

...and after rearranging and taking into account the input equation by replacing $\ddot{\mathbf{x}}_k$ with $m^{-1} \cdot \mathbf{u}_k$ we get:

$$\begin{aligned}
 0 &= 2\mathbf{x}_0 - 2\mathbf{x}_1 + T\dot{\mathbf{x}}_0 + T\dot{\mathbf{x}}_1 & 0 &= 2\dot{\mathbf{x}}_0 - 2\dot{\mathbf{x}}_1 + Tm^{-1}\mathbf{u}_0 + Tm^{-1}\mathbf{u}_1 \\
 0 &= 2\mathbf{x}_1 - 2\mathbf{x}_2 + T\dot{\mathbf{x}}_1 + T\dot{\mathbf{x}}_2 & 0 &= 2\dot{\mathbf{x}}_1 - 2\dot{\mathbf{x}}_2 + Tm^{-1}\mathbf{u}_1 + Tm^{-1}\mathbf{u}_2 \\
 0 &= 2\mathbf{x}_2 - 2\mathbf{x}_3 + T\dot{\mathbf{x}}_2 + T\dot{\mathbf{x}}_3 & 0 &= 2\dot{\mathbf{x}}_2 - 2\dot{\mathbf{x}}_3 + Tm^{-1}\mathbf{u}_2 + Tm^{-1}\mathbf{u}_3 \\
 0 &= 2\mathbf{x}_3 - 2\mathbf{x}_4 + T\dot{\mathbf{x}}_3 + T\dot{\mathbf{x}}_4 & 0 &= 2\dot{\mathbf{x}}_3 - 2\dot{\mathbf{x}}_4 + Tm^{-1}\mathbf{u}_3 + Tm^{-1}\mathbf{u}_4 \\
 0 &= 2\mathbf{x}_4 - 2\mathbf{x}_5 + T\dot{\mathbf{x}}_4 + T\dot{\mathbf{x}}_5 & 0 &= 2\dot{\mathbf{x}}_4 - 2\dot{\mathbf{x}}_5 + Tm^{-1}\mathbf{u}_4 + Tm^{-1}\mathbf{u}_5
 \end{aligned}
 \tag{4.8.5}$$

...including the terminal constraints this may now be written in matrix form as follows:

$$\begin{bmatrix}
 2 & -2 & 0 & 0 & 0 & 0 & T & T & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 2 & -2 & 0 & 0 & 0 & 0 & T & T & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 2 & -2 & 0 & 0 & 0 & 0 & T & T & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 2 & -2 & 0 & 0 & 0 & 0 & T & T & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 2 & -2 & 0 & 0 & 0 & 0 & T & T/m & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & -2 & 0 & 0 & 0 & T & T/m & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & -2 & 0 & 0 & 0 & 0 & T & T/m & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & -2 & 0 & 0 & 0 & 0 & 0 & T & T/m & 0 & 0 \\
 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{x}_0 \\ \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \\ \mathbf{x}_4 \\ \mathbf{x}_5 \\ \dot{\mathbf{x}}_0 \\ \dot{\mathbf{x}}_1 \\ \dot{\mathbf{x}}_2 \\ \dot{\mathbf{x}}_3 \\ \dot{\mathbf{x}}_4 \\ \dot{\mathbf{x}}_5 \\ \mathbf{u}_0 \\ \mathbf{u}_1 \\ \mathbf{u}_2 \\ \mathbf{u}_3 \\ \mathbf{u}_4 \\ \mathbf{u}_5 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

...and in a more compact form:

$$\mathbf{A}_{eq} \cdot \mathbf{X} = \mathbf{B}_{eq}
 \tag{4.8.6}$$

The pattern in the above matrix equation is a predictable one and may be algorithmically generated for any number of nodes in the trajectory, making this technique more versatile than it initially seems. In this case, terminal constraints have been imposed which removes the need to apply terminal costs. Therefore the above problem takes the Lagrange form shown in eq. 4.8.7. Upper and lower bounds need to be defined to impose path constraints. They can also be used to limit velocity and input thrust to realistically achievable values.

$$\text{Lagrange Cost: } J \approx \min_{\mathbf{x}_k, \tilde{\mathbf{x}}_k, \mathbf{u}_k} \left\{ \sum_{k=0}^{K-1} \mathcal{F}(\mathbf{x}_k, \mathbb{O}) \right\} \quad (4.8.7)$$

Subject to:

$$- \text{ State transition constraints: } \mathbf{A}_{eq} \cdot \mathbf{X} = \mathbf{B}_{eq} \quad (4.8.8)$$

$$- \text{ Boundary conditions:}$$

$$- \text{ Inequality constraints: } \mathbf{A}_{eq} \cdot \mathbf{X} \leq \mathbf{B}_{eq} \quad (4.8.9)$$

$$- \text{ Trajectory constraints: } \mathbf{X}_{lb} \leq \mathbf{X} \leq \mathbf{X}_{ub} \quad (4.8.10)$$

$$- \text{ Initial approximate guess: } \tilde{\mathbf{X}} \quad (4.8.11)$$

where: \mathbf{X}_{lb} is the lower bound vector for all variables in \mathbf{X}
 \mathbf{X}_{ub} is the upper bound vector for all variables in \mathbf{X}
 \mathbf{A}_{eq} is a matrix that defines the equality constraints
 \mathbf{B}_{eq} is a vector that defines the equality constraints
 \mathbf{A}_{ineq} is a matrix that defines the inequality constraints (this is not used in this example)
 \mathbf{B}_{ineq} is a vector that defines the inequality constraints (this is not used in this example)

After the problem has been converted into the prescribed form this is then fed into any of a wide variety of NLP solvers such as PSOPT [4.14], SNOPT [4.15], NPSOL, COMSOL, FMINCON, IPOPT or many others that tend to use a very similar interface. An approximate guess ($\tilde{\mathbf{X}}$) is usually required and has a great bearing on the outcome of the solver. This is typically the interface between the global optimizer and the local optimizer. An example is given hereunder using FMINCON from Matlab:

$$[\mathbf{X}^*, J^*] = \text{fmincon}(@\text{Cost}, \tilde{\mathbf{X}}, \mathbf{A}_{ineq}, \mathbf{B}_{ineq}, \mathbf{A}_{eq}, \mathbf{B}_{eq}, \mathbf{X}_{lb}, \mathbf{X}_{ub});$$

The solver then returns the closest local optimum to the “guess” provided, (and the associated cost), and if this was in the vicinity of the global optimum, the problem is solved. In **Figure 4.8.2**, three example results are provided, where (a) shows a 2D path clearly trapped in a local minimum, while (b) and (c) show a generalization of the same process in 3D, this time using potential fields surrounding cuboid obstacles. A NURB interpolator was included when evaluating the cost metric in this example, to ensure that the knots found correspond to the optimum interpolated trajectory.

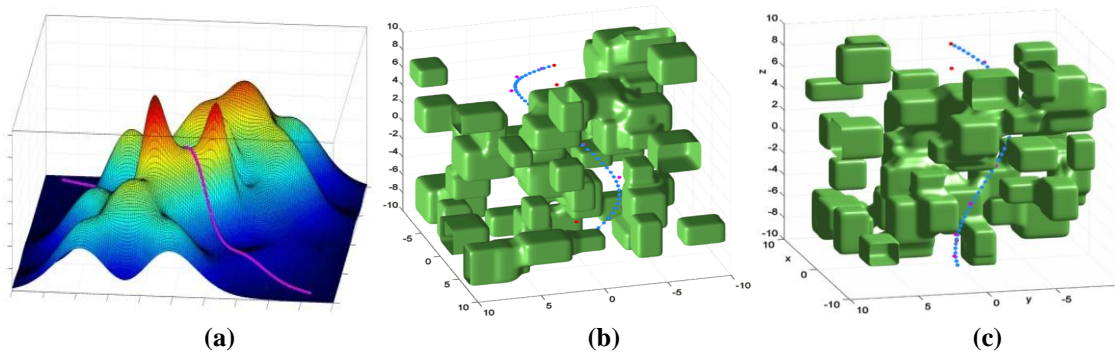


Figure 4.8.2: Locally Optimized Trajectory of a Newtonian Point Mass

This method of collocation is one of the simpler ways for performing trajectory optimization with gradient methods. A much more accurate and generally more reliable way of performing trajectory optimization is with Pseudo-Spectral methods that use a superposition of functions that span the entire domain together with Gaussian quadrature for numerical integration instead of the simple trapezoidal rule.

4.9 IN SUMMARY

A broad comparison between a number of different global optimisation approaches for CATM was provided. Some detail was given in respect of three methods that were chosen and adapted during this study to address the particular combination of challenges presented by this formidable problem. CATM appears to be feasible with no fundamental obstacles, except for thousands of man-years of engineering effort. **Table 4.9.1** summarises these findings and scores the algorithms in accordance to their suitability.

CLASS	GUIDE	ALGORITHM	Search Space				Multi Resolution	Multimodal	Dynamic Adaptation	Pipelining	Parallel Scaling	Distributable	Locality Principle	Agent Symmetric	High-Dimension.	SCORE	
			Unstructured	Real valued	Tree	Graph											
Deterministic	Heuristic	B&B, A*, B*, D*	x	✓	✓	✓	x	x	x	D	D	x	x	x	x	0	
	Optimal Sub-Structure	Dynamic Programming (DP)	x	✓	x	✓	x	x	x	✓	D	x	x	x	x	1	
		Iterative DP (IDP)	x	✓	x	✓	✓	x	x	✓	D	x	x	x	x	2	
		Iterative Dynamic Routing (IDR)	x	✓	x	✓	✓	x	L	✓	D	x	x	x	x	3	
Probabilistic	Random	Monte Carlo Search (MCS)	✓	✓	✓	✓	x	✓	✓	x	✓	✓	x	✓	x	5	
	Meta-Heuristic	Q/R	Contractile Monte Carlo Search (CMCS)	✓	✓	✓	✓	✓	x	✓	C	✓	✓	x	✓	x	6
			Simulated Annealing (SA)	✓	✓	✓	✓	x	x	x	C	✓	✓	x	✓	x	4
	Evolutionary	Real Coded Evolutionary Prog. (RCEP)	✓	✓	✓	✓	✓	x	L	C	✓	x	x	x	x	4	
		Binary Coded Genetic Algor. (BCGA)	x	x	✓	✓	✓	x	x	C	✓	x	x	x	x	3	
		Real Coded Genetic Algorithm (RCGA)	✓	x	✓	✓	✓	x	x	C	✓	x	x	x	x	3	
		Cooperative Coevolutionary RCGA	✓	x	✓	✓	✓	x	x	C	✓	✓	✓	✓	✓	7	
	Swarm Intelligence	Ant Colony Optimisation (ACO)	x	✓	✓	✓	x	x	✓	C	✓	x	x	x	x	3	
		Artificial Bee Colony (ABC)	✓	✓	x	x	x	✓	✓	C	✓	x	x	x	x	4	
		Firefly Algorithm (FA)	✓	✓	x	x	x	✓	✓	C	✓	x	x	x	x	4	
		Glow-worm Swarm Optimisation (GSO)	✓	✓	x	x	x	✓	✓	C	✓	x	x	x	x	4	
		Particle Swarm Optimisation (PSO)	✓	✓	x	x	✓	x	x	C	✓	x	x	x	x	3	
		Cooperative Coevolution PSO (CCPSO)	✓	✓	x	x	✓	x	x	C	✓	x	x	x	x	3	
Dynamic Guided PSO (GPSO-D)		✓	✓	x	x	✓	x	✓	C	✓	x	x	x	x	4		
Clustered CCGPSO-D	✓	✓	x	x	✓	x	✓	C	✓	✓	✓	✓	✓	8			

C = Circular Pipeline, L= Local Reconvergence, Q/R = Quasi Random, D = Difficult

Table 4.9.1: Comparison between some Global Search Techniques

4.10 REFERENCES

- [4.1] Bryson, A.E., Ho, Y.C., 1975, "Applied Optimal Control: Optimization, Estimation, and Control," John Wiley & Sons, Incorporated, USA
- [4.2] Pontryagin L.S, Boltyanskii V.G, Gamkrelidze R. V, Mishchenko E. F, 1962, "The Mathematical Theory of Optimal Processes", English translation: Interscience, New York, USA
- [4.3] Becerra, V.M., 2004, Aug, "Solving Optimal Control Problems with State Constraints Using Nonlinear Programming and Simulation Tools", IEEE Transactions on Education, vol. 47, no. 3, pp. 377
- [4.4] Kuhn, H. W., Tucker, A. W., 1951, "Nonlinear programming," Proceedings of 2nd Berkeley Symposium, Berkeley, University of California Press, pp. 481–492
- [4.5] Betts, J. T., 1998, "Survey of numerical methods for trajectory optimization," Journal of Guidance, Control, and Dynamics, AIAA, vol. 21, no. 2, pp 193-207
- [4.6] Fliess, M., Lévine, P., Martin, J., Rouchon, P., 1995, "Flatness and defect of nonlinear systems: Introductory theory and examples," International Journal of Control, vol. 61 no. 6, pp 1327-1361.
- [4.7] Martin, P., Devasia, S., Paden, B. 1994, Dec., "A different look at output tracking: control of a VTOL aircraft." in Proceedings of the 33rd IEEE Conference on Decision and Control, IEEE, vol. 3, pp. 2376-2381
- [4.8] Koo, T. J., & Sastry, S. 1999, "Differential flatness based full authority helicopter control design," In Proceedings of the 38th IEEE Conference on Decision and Control, IEEE, vol. 2, pp. 1982-1987
- [4.9] Cowling, I. D., Yakimenko, O. A., Whidborne, J. F., & Cooke, A. K. 2007, Jul. "A prototype of an autonomous controller for a quadrotor UAV", In European Control Conference, Kos, Greece, pp. 1-8
- [4.10] Sira-Ramirez, H., & Agrawal, S. K., 2004. "Differentially flat systems," Marcel, Dekker, Inc., New York, USA
- [4.11] Hauser, J., & Hindman, R. 1997, Dec., "Aggressive flight maneuvers," In Proceedings of the 36th IEEE Conference on Decision and Control, IEEE, vol.5, pp.4186-91
- [4.12] Kennedy, J., Eberhart, R.C. & Shi, Y., 2001, "Swarm Intelligence," The Morgan Kaufmann Series in Evolutionary Computation, Morgan Kaufmann Publishers, San Francisco, CA, USA
- [4.13] Panigrahi, B. K., Shi, Y., & Lim, M. H., 2011. "Handbook of Swarm Intelligence". Springer-Verlag, Berlin, Heidelberg, Germany
- [4.14] Becerra, V. M. 2010, September. "Solving complex optimal control problems at no cost with PSOPT," In Computer-Aided Control System Design (CACSD), 2010 IEEE International Symposium on, IEEE, pp. 1391-1396
- [4.15] Gill, P. E., Murray, W., & Saunders, M. A. 2002. "SNOPT: An SQP algorithm for large-scale constrained optimization," SIAM Journal on Optimization, vol.12, no.4, pp 979-1006
- [4.16] Spanier, E.H., 1994, "Algebraic topology". vol. 55. no. 1, Springer-Verlag, Berlin, Heidelberg, Germany
- [4.17] Nguyen, T. T., Yang, S., & Branke, J., 2012, "Evolutionary dynamic optimization: A survey of the state of the art." in Swarm and Evolutionary Computation, vol. 6, pp. 1-24.
- [4.18] Alba, E., Nakib, A., & Siarry, P. 2013, "Metaheuristics for Dynamic Optimization", in Studies in Computational Intelligence, Springer-Verlag, Berlin, Heidelberg, Germany, vol. 433
- [4.19] Gill, P. E., Murray, W., & Saunders, M. A. 2008, "User's guide for SNOPT version 7: Software for large-scale nonlinear programming", Systems Optimization Laboratory, Stanford University
- [4.20] Michalewicz, Z., Fogel, D.B., 2004, "How to Solve it: Modern Heuristics", 2nd ed., Springer-Verlag, Berlin, Heidelberg, Germany
- [4.21] Engelbrecht, Andries P., 2005, "Fundamentals of Computational Swarm Intelligence," John Wiley and Sons Ltd, Pretoria, South Africa
- [4.22] Bäck, T., Hoffmeister, F. & Schwefel, H.P., "A survey of evolution strategies", in: Proceedings of the Fourth International Conference on Genetic Algorithms, Morgan Kaufmann Publishers, San Mateo, California, 1991, pp. 2-9

- [4.23] Mezura-Montes, E., & Coello Coello, C. A. 2011. "Constraint-handling in nature-inspired numerical optimization: past, present and future. *Swarm and Evolutionary Computation*", vol. 1 no. 4, pp 173-194
- [4.24] Tessema, B. & Yen, G.G., 2009, "An adaptive penalty formulation for constrained evolutionary optimization", *IEEE Transactions on Systems, Man & Cybernetics, Part A (Systems & Humans)* vol. 39, no. 3, pp 565-578
- [4.25] Ray, T, Singh, H.K., Isaacs, A. & Smith, W., 2009, "Infeasibility driven evolutionary algorithm for constrained optimization", in: E. Mezura-Montes (Ed.), *Constraint- Handling in Evolutionary Optimization*, in: *Studies in Computational Intelligence Series*, vol. 198, Springer-Verlag, Berlin, Heidelberg, Germany, pp. 145–165.
- [4.26] Kim, D.G. & Husbands, P., 1998, "Mapping based constraint handling for evolutionary search: Thurston's circle packing and grid generation", in: I. Parmee (Ed.), *The Integration of Evolutionary and Adaptive Computing Technologies with Product/System Design and Realisation*, Springer-Verlag, Plymouth, United Kingdom, pp. 161–173
- [4.27] Michalewicz, Z., 1996, "Genetic Algorithms+Data Structures = Evolution Programs", third ed., Springer-Verlag, Berlin, Heidelberg, Germany.
- [4.28] Deb, K., 2000, "An efficient constraint handling method for genetic algorithms", in *Computer Methods in Applied Mechanics and Engineering*, vol. 186, pp 311–338
- [4.29] Marler, R.T., & Arora, J.S., 2004, "Survey of multi-objective optimization methods for engineering." *Structural and multidisciplinary optimization*, vol. 26, no. 6, pp. 369-395.
- [4.30] Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. A. M. T., 2002, "A fast and elitist multiobjective genetic algorithm: NSGA-II." *Evolutionary Computation*, *IEEE Transactions on*, vol. 6, no. 2, pp. 182-197
- [4.31] Li, H., & Zhang, Q., 2009, "Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II" *Evolutionary Computation*, *IEEE Transactions on*, vol. 13 no. 2, pp. 284-302
- [4.32] Schandl, B., Klamroth, K., & Wiecek, M. M., 2002, "Norm-based approximation in multicriteria programming," *Computers & Mathematics with Applications*, vol. 44 no.7, pp. 925-942.
- [4.33] Krozel, J., Peters, M., Bilimoria, K. D., Lee, C., & Mitchell, J., 2001, Dec., "System performance characteristics of centralized and decentralized air traffic separation strategies" In *Fourth USA/Europe Air Traffic Management Research and Development Seminar*.
- [4.34] Mahdavi, S., Shiri, M. E., & Rahnamayan, S., 2014, "Metaheuristics in large-scale global continues optimization: A survey." *Information Sciences*, Vol. 295 pp. 407-428
- [4.35] Yang, Z., Tang, K., & Yao, X., 2008, "Large scale evolutionary optimization using cooperative co-evolution," *Information Sciences*, Elsevier B.V., vol.178, no.15, pp.2986-99
- [4.36] Potter, M. A., & De Jong, K. A., 1994, "A cooperative coevolutionary approach to function optimization." In *Parallel Problem Solving from Nature, PPSN III*, Springer-Verlag, Berlin, Heidelberg, Germany, pp. 249-257
- [4.37] Van den Bergh, F. & Engelbrecht, A., 2004, "A cooperative approach to particle swarm optimization," *IEEE Transactions on Evolutionary Computation*, vol.8, no.3, pp. 225-39
- [4.38] Li, X., & Yao, X. 2009, May. "Tackling high dimensional nonseparable optimization problems by cooperatively coevolving particle swarms." In *Evolutionary Computation, 2009. CEC'09. IEEE Congress on*, pp. 1546-1553
- [4.39] Clerc, M., & Kennedy, J., 2002, "The particle swarm-explosion, stability, and convergence in a multidimensional complex space." *Evolutionary Computation*, *IEEE Transactions on*, vol. 6, no. 1, pp. 58-73.
- [4.40] Li, X., & Yao, X., 2012, "Cooperatively coevolving particle swarms for large scale optimization," *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 2, pp. 210-224
- [4.41] Luus, R., 2000, "Iterative dynamic programming," *Monographs and Surveys in Pure and Applied Mathematics*, Chapman & Hall/CRC Press LLC, Boca Raton, Florida
- [4.42] Luus, R. 1990, "Optimal control by dynamic programming using systematic reduction in grid size". *International Journal of Control*, Taylor & Francis Publishers, vol. 51 no. 5, pp. 995-1013

- [4.43] Luus, R., 2009, "Handling Inequality Constraints in Optimal Control by Problem Reformulation," *Industrial & Engineering Chemistry Research*, vol.48, no.21, pp.9622-30
- [4.44] Luus, R., 2001, Aug. 26-29, "Iterative dynamic programming: From curiosity to a practical optimization procedure," *40th Annual Conference of Metallurgists of CIM: International symposium on Computer Applications in Metals Processing*, Toronto, Ontario, Canada, pp. 3-13
- [4.45] Stanford Research Systems, 2013, "PRS10 — Rubidium frequency standard with low phase noise", datasheet, Stanford Research Systems, Inc., Sunnyvale, California, <http://www.thinksrs.com/downloads/PDFs/Catalog/PRS10c.pdf>, accessed: Jul 2014.
- [4.46] Whidborne, J. F., Cowling, I. D., & Yakimenko, O. A. 2008, April. "A direct method for UAV guidance and control." In Proceedings of 23rd International Unmanned Air Vehicle Systems (UAVS) Conference, University of Bristol, Bristol, UK
- [4.47] De Boor, C., 2001, "A Practical Guide to Splines (revised edition)", Series in Applied Mathematical Sciences, vol. 27, Springer-Verlag, New York, USA
- [4.48] Sörensen, K., Feb 2013, "Metaheuristics - The metaphor exposed," *International Transactions in Operational Research*, John Wiley & Sons.
- [4.49] Sörensen, K., & Glover, F.W., 2013, "Metaheuristics." In *Encyclopedia of Operations Research and Management Science*, Springer USA, pp. 960-970
- [4.50] Bellman, R.E., 1953, "An introduction to the theory of dynamic programming," Research Report, Rand Corporation, Santa Monica, California.
- [4.51] Bellman, R.E., 1957, "Dynamic Programming," Princeton University Press, Princeton, NJ, Republished 2003, Dover
- [4.52] Turing, Alan M. 1950, Oct, "Computing machinery and intelligence". *Mind – Quarterly Review*, vol. 59, no. 238 pp. 433–460
- [4.53] Holland, J. H., 1973. "Genetic algorithms and the optimal allocation of trials." *SIAM Journal on Computing*, vol. 2, no. 2, pp. 88-105.
- [4.54] Miller, B. L., & Goldberg, D. E., 1995, "Genetic algorithms, tournament selection, and the effects of noise." *Complex Systems*, vol. 9 no. 3, pp. 193-212.
- [4.55] Ross, I.M., and Gong, Q., 2008, "Emerging Principles in Fast Trajectory Optimization", Elissar Publications, Monterey, CA
- [4.56] Goldberg, D. E., 1990, "Real-coded genetic algorithms, virtual alphabets, and blocking." *Urbana*, vol. 51, Report 61801
- [4.57] Herrera, F., Lozano, M., & Verdegay, J. L., 1998, "Tackling real-coded genetic algorithms: Operators and tools for behavioural analysis." *Artificial Intelligence Review*, vol. 12n no. 4, pp. 265-319.
- [4.58] Chelouah, R., & Siarry, P., 2000, "A continuous genetic algorithm designed for the global optimization of multimodal functions." *Journal of Heuristics*, vol.6, no.2, pp.191-213.
- [4.59] Radcliffe N.J., 1991, "Equivalence Class Analysis of Genetic Algorithms." *Complex Systems*, vol 5, no. 2, pp 183–205.
- [4.60] Michalewicz, Z., 1992, "Genetic Algorithms + Data Structures = Evolution Programs." SpringerVerlag, New York., NJ
- [4.61] Wright, A., 1991, "Genetic Algorithms for Real Parameter Optimization." *Foundations of Genetic Algorithms 1*, G.J.E Rawlin (Ed.), Morgan Kaufmann, San Mateo, pp205–218.
- [4.62] Muhlenbein H. & Schlierkamp-Voosen D., 1993, "Predictive Models for the Breeder Genetic Algorithm I. Continuous Parameter Optimization." *Evolutionary Computation*, vol 1, pp. 25–49.
- [4.63] Elsayed, S. M., Sarker, R. A., & Essam, D. L. 2011, June, "GA with a new multi-parent crossover for solving IEEE-CEC2011 competition problems." In *Evolutionary Computation (CEC), 2011 IEEE Congress on pp. 1034-1040*
- [4.64] Sareni, B., & Krahenbuhl, L., 1998, "Fitness sharing and niching methods revisited." *Evolutionary Computation, IEEE Transactions on*, vol. 2 no.3, pp. 97-106

- [4.65] Zhan, Z. H., Li, J., Cao, J., Zhang, J., Chung, H. H., & Shi, Y. H., 2013, "Multiple populations for multiple objectives: a coevolutionary technique for solving multiobjective optimization problems." *Cybernetics, IEEE Transactions on*, vol. 43, no. 2, pp. 445-463
- [4.66] M. J. Wade, 2007. Mar., "The co-evolutionary genetics of ecological communities," *Nature Review Genetics*, vol. 8, no. 3, pp. 185-195
- [4.67] Hillis, W. D., 1990, "Co-evolving parasites improve simulated evolution as an optimization procedure." *Physica D: Nonlinear Phenomena*, vol. 42, no. 1, pp. 228-234
- [4.68] Husbands, P., 1994, "Distributed coevolutionary genetic algorithms for multi-criteria and multi-constraint optimization." In Fogarty, T.C. (ed.) *Evolutionary Computing (Selected papers from AISB workshop, Leeds, UK)*, pp. 150-165. Springer-Verlag, Berlin
- [4.69] Potter, M., 1997, "The Design and Analysis, of a Computational Model of Cooperative CoEvolution." PhD thesis, George Mason University, Fairfax, Virginia
- [4.70] Yang, Z., Tang, K., & Yao, X., 2008, "Large scale evolutionary optimization using cooperative coevolution." *Information Sciences*, vol. 178, no. 15, pp. 2985-2999
- [4.71] Li, X., & Yao, X., 2012, "Cooperatively coevolving particle swarms for large scale optimization." *Evolutionary Computation, IEEE Transactions on*, vol.16, no.2, pp. 210-224
- [4.72] Gao, Y., Zhang, X., & Guan, X., 2012, Aug., "Cooperative multi-aircraft conflict resolution based on co-evolution." In *Instrumentation & Measurement, Sensor Network and Automation (IMSNA), 2012 International Symposium on*, vol. 1, pp. 310-313
- [4.73] Zhang, X., Guan, X., Hwang, I., & Cai, K., 2013, "A hybrid distributed-centralized conflict resolution approach for multi-aircraft based on cooperative co-evolutionary." *Science China Information Sciences*, vol. 56, no. 12, pp. 1-16
- [4.74] Liu, E. Y., Guan, X. M., Zhang, X. J., & Zeng, J., 2013, "Conflict Resolution Based on Cooperative Coevolutionary with Dynamic Grouping Strategy for Multi-Aircraft." *Applied Mechanics and Materials*, vol. 333, pp. 1251-1255
- [4.75] Granville, V.; Krivanek, M.; Rasson, J.P., 1994, "Simulated annealing: A proof of convergence." *IEEE Transactions on Pattern Analysis and Machine Intelligence* vol 16 no. 6 pp. 652-656
- [4.76] Pedersen, M.E.H., 2010, Jan, "Tuning & Simplifying Heuristic Optimization," *Computational Engineering and Design Group, PhD Thesis, School of Engineering Sciences, University of Southampton, UK*
- [4.77] Yang, X. S., 2009, "Firefly algorithms for multimodal optimization," In *Stochastic algorithms: foundations and applications*, Springer Berlin Heidelberg, pp. 169-178
- [4.78] Dorigo, Marco, 1992, "Optimization, Learning, and Natural Algorithms," Dipartimento di Elettronica, PhD Thesis, Politecnico di Milano, Milano (Italy),
- [4.79] Deneubourg, J.-L., S. Aron, S. Goss and J.M. Pasteels, 1990, "The self-organizing exploratory pattern of the Argentine Ant," *Journal of Insect Behavior*, vol. 3, no. 2, pp. 159-168,
- [4.80] Karaboga, D., & Basturk, B., 2007, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm." *Journal of Global optimization, Springer BV*, vol. 39 no. 3, pp. 459-471
- [4.81] de Castro, L. N. & Timmis, J., 2002, "Artificial Immune Systems: A New Computational Intelligence Approach," Springer. pp. 57-58
- [4.82] Yang, X. S., 2010, "A New Metaheuristic Bat-Inspired Algorithm," in: *Nature Inspired Cooperative Strategies for Optimization (NISCO 2010)* (Eds. J. R. Gonzalez et al.), *Studies in Computational Intelligence*, Springer Verlag Berlin, vol. 284, pp. 65-74
- [4.83] Krishnanand, K.N. & Ghose, D., 2009, "Glowworm swarm optimization for simultaneous capture of multiple local optima of multimodal functions". *Swarm Intelligence, Springer BV*, vol. 3, no. 2, pp.87-124
- [4.84] Kennedy, J.; Eberhart, R., 1995, "Particle Swarm Optimization." *Proceedings of IEEE International Conference on Neural Networks* vol. IV. pp. 1942-1948
- [4.85] P. Rabanal, I. Rodríguez, & F. Rubio , 2007, "Using River Formation Dynamics to Design Heuristic Algorithms". *Unconventional Computation, Springer Berlin Heidelberg, LNCS 4616*, pp. 163-177.

-
- [4.86] Eberhart, R., Kennedy, J., 1995, 4-6 October, "A new optimizer using particle swarm theory," IEEE Sixth International Symposium on Micro Machine and Human Science, MHS, pp. 39-43
- [4.87] Bratton, D., Kennedy, J., 2007, Apr, "Defining a Standard for Particle Swarm Optimization," in Proceedings of the 2007 IEEE Swarm Intelligence Symposium (SIS 2007), IEEE, pp. 120-127
- [4.88] Eberhart, R. C., & Shi, Y., 2001, "Particle swarm optimization: developments, applications and resources," In Evolutionary Computation, Proceedings of the 2001 Congress on, IEEE, vol. 1, pp. 81-86
- [4.89] Maniezzo, V., 2009, "Hybridizing Metaheuristics and Mathematical Programming," Series: Annals of Information Systems, Vol. 10; Stützle, T.; Voß, S. (Eds.), Springer
- [4.90] Rinnooy-Kan, A.H.G. & Timmer, G.T., 1984, "Stochastic methods for global optimization," American Journal of Mathematical Management Science, vol. 4, pp. 740
- [4.91] Goldberg, D. E., & Voessner, S., 1999, "Optimizing global-local search hybrids." Urbana, vol. 51, Report 61801
- [4.92] Castellini, F., Riccardi, A., Lavagna, M., & Büskens, C., 2011, "Global and Local Multidisciplinary Design Optimization of Expendable Launch Vehicles," In 52nd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference 19th, 4-7 April 2011, Denver, Colorado, AIAA, 2011-1901
- [4.93] Melton, R. G., 2014, "Hybrid methods for determining time-optimal, constrained spacecraft reorientation maneuvers," Acta Astronautica, vol. 94, no. 1, pp. 294-301
- [4.94] Renders, J. M., & Flasse, S. P., 1996, "Hybrid methods using genetic algorithms for global optimization." Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, vol. 26, no. 2, pp. 243-258
- [4.95] Jourdan, L., Basseur, M., & Talbi, E. G., 2009, "Hybridizing exact methods and metaheuristics: A taxonomy." European Journal of Operational Research, vol. 199, no. 3, pp. 620-629
- [4.96] Cotta, C., & Troya, J. M., 2003, "Embedding branch and bound within evolutionary algorithms." Applied Intelligence, vol. 18 no. 2, pp. 137-153
- [4.97] Neri, F., & Cotta, C., 2012, "Memetic algorithms and memetic computing optimization: A literature review." Swarm and Evolutionary Computation, vol. 2, 1-14.
-

Chapter 5

Simulation, Results and Discussion

In this chapter we will discuss the simulation paradigm, the testing methodology, the resources used, the simplifications adopted and a number of practical considerations that must respect the technical limitations of the underlying simulation platform. Results are discussed in the light of the techniques described and objectives set out in earlier chapters.

5.1 THE SIMULATION PARADIGM

Running CATM simulations of continental levels of traffic involved substantial computational resources that were not easily afforded on any single desktop machine. The CATM system described, simultaneously requires one independent computing element per flight agent. However, for practical purposes, during simulation this could not be accommodated using discrete hardware elements. Therefore, in order to preserve and test the distributed nature of the algorithms, an efficient, multi-node simulation environment had to be devised. See **Figure 5.1.1**.

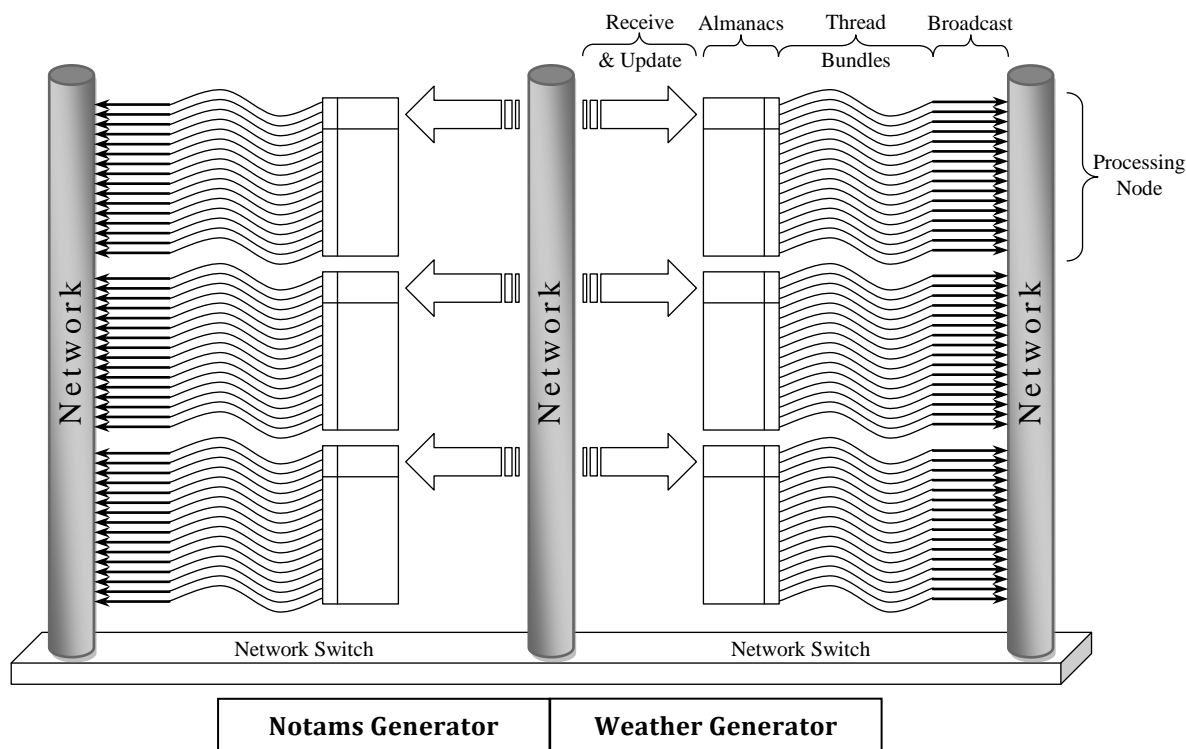


Figure 5.1.1: The CATM Simulation System Architecture

5.1.1 Scaling-up to the Continental CATM Problem

In order to simulate CATM faithfully, the architecture employed for simulation had to be fully *agent symmetric*. This is to say that all agents in the system had to be identical with no preferential treatment. Processing resources had to also be allocated to each agent independently and the coupling between agents had to be as loose as practically possible, save for the essential trajectory transfer required for agent cooperation. The intra-cluster architecture is envisaged as shown in **Figure 5.1.2**, where within each cluster of M aircraft, one is a master and the rest are slaves. Keeping in mind that there are as many clusters as there are flights, symmetric operation was still achieved by replicating the same arrangement for every agent in the system. Thus every flight was a master of some cluster.

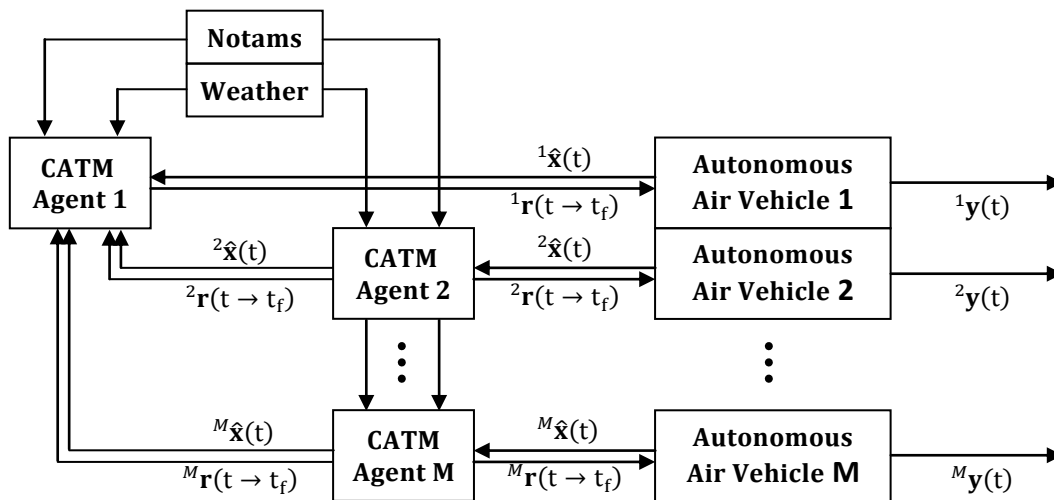


Figure 5.1.2: The CATM System Hierarchy within a Single Flight Cluster

The CATM master controller in each cluster receives the best known remaining trajectories ${}^m\mathbf{r}(t \rightarrow t_f)$ and current system states ${}^m\hat{\mathbf{x}}(t)$ from each of the slaves. The CATM master controller updates the optimal trajectory for its own aircraft, taking into account all the latest information gathered from the cluster and any weather and NOTAM obstacle database information. The process is repeated at every time-step and if the flights are airborne and being executed in real-time, a receding horizon methodology can be used to keep the master optimiser current with the unfolding state trajectory.

5.1.2 Multi -Threaded Approach

The agent approach proposed for CATM is best embodied using a multithreaded approach where each aircraft is represented with a single computing thread and each thread is interconnected over a high capacity network infrastructure. Therefore a number of threads were assigned to a simulation process which was run on a computing node consisting of a network-connected, high performance blade server. Gigabit Ethernet was selected to emulate the CATM network infrastructure due to its reliability, performance and simplicity. **Figure 5.1.1** shows the data flow patterns in the architecture employed.

5.1.3 The Shared Almanac

Each thread is essentially an independent PSO optimizer which is tasked with generating an optimal trajectory for the flight agent it is hosting. Nothing is shared among threads which operate on separate memory areas, except for the flight *almanac*. For memory efficiency purposes a single copy of the almanac is maintained per process and each thread has asynchronous access to this shared memory area. This saves substantial memory which is understandably limited in the simulator.

Each thread interacts with a multitude of other threads residing anywhere over the network infrastructure by broadcasting self-contained packets of trajectory waypoint data. These are received at each node and used to update the local copies of the almanac. No direct communication between the threads is allowed, if not through the almanacs. This way, it was ensured that each thread has an identical and symmetric relationship to every other thread in the entire system, irrespective of where the threads were physically hosted. This is essential to maintain the validity of the distributed computing simulation.

5.1.4 Transmission Filtering

The network forms the backbone of the system and substantial thought was dedicated to maximising performance. Gigabit Ethernet provides a low cost method to interconnect high performance computing nodes. Even with 1530 byte packets, (the longest supported by standard Ethernet), latency is of the order of 50 μ s. Ethernet switches that can support billions of network transactions per second are also readily available. A 24 port switch from NetGear (GS724TS) was used with an internal switching bandwidth of 48GB/sec. To further improve throughput and minimise delays, two network interfaces were provided per node and these were aggregated for a total bandwidth of 2Gbits per node in each direction.

However, with thousands of agents sending trajectory data over the network at every iteration, it is easy to overload the network infrastructure. As it happens, most of the time, especially after initial convergence, the threads have no new information to transmit other than repeating the same packets over and over. This is very wasteful of resources and tends to crowd-out important information on the network. One neat solution to the problem is to self-censor all the threads from transmission unless significant changes are detected. Since all aircraft have access to a local copy of the latest almanac this does not impact the algorithm's performance but significantly reduces network traffic.

5.1.5 Software Architecture

In order to run simulations smoothly, a Master-Slave-Monitor approach was used. The slaves contain all the processing threads and almanacs. The master issues network commands to start, stop, initialise and clear the slaves of simulation data, thereby setting up and coordinating simulation scenarios. The monitor passively listens on the network and records all traffic which is used to calculate performance statistics. The slaves are all uniquely identified by the MAC address of the respective machine's network interface, which simplifies any future addition of more nodes to increase simulation capacity.

5.2 CONSTRUCTION OF A CATM SIMULATION RIG

The thousands of concurrent optimizer threads required some rather hefty computational hardware to accommodate. For this purpose, a dedicated 12 node computing blade server rig was constructed using dual processor motherboards from Supermicro® computer and Intel Xeon® Harpertown 8-core CPUs. The memory of the system was sized to accommodate several thousand threads per blade. **Figure 5.2.1** describes the setup.

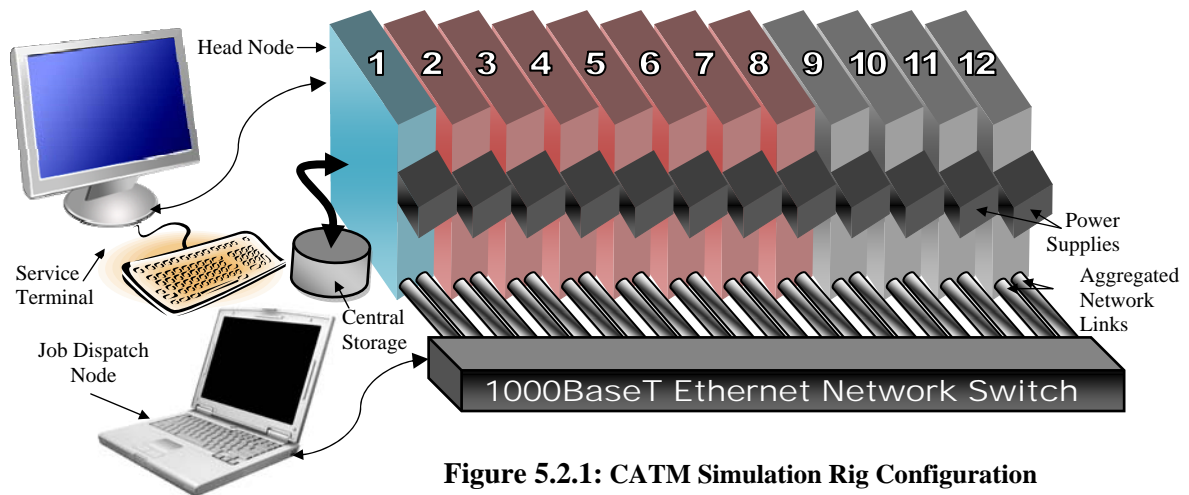


Figure 5.2.1: CATM Simulation Rig Configuration

Special consideration was taken to ensure adequate cooling of the resulting multi kW system. Water cooling and high capacity vehicle radiators were used to dispose of the waste heat from all 24 CPUs, and forced air cooling was used for the memory and the rest of the system. The processors were also over-clocked to around 130% to extract maximum performance, and their numerical reliability was exhaustively re-tested using Intel's Linpack V11.1.1 benchmarking tools. The aggregate performance of the system reaches 1.0 Tflop and is capable of traversing 80k PSO epochs per second for a 12,288 flight air traffic system. The system is divided into a 3.8GHz head node and 11 slaves. **Figure 5.2.2** shows a photograph of the assembled computing rig running some CATM tests, while **Table 5.2.1** shows a summary of the system's configuration and performance, together with an HP Pavilion DV6 i7 laptop PC, also listed here for comparison.



Figure 5.2.2: CATM Simulation Rig

NODE (#)	PROCESSOR (INTEL Xeon)	ECC MEMORY (GB)	OVERCLOCK SPEED (GHz)	CPU CORES (#)	INTEL LINPACK [†] BENCHMARK (GFLOP)	PEAK CCGPSO PERFORMANCE [‡] ITERATIONS/S	PEAK CPU THERMAL POWER (W)	PEAK INPUT POWER (W)
0*	i7-720QM	8	1.728	4	14.47	2834	45	91.6
1	Dual X5460	32	3.8	4x2	102.42	7792	163x2	642
2	Dual E5430	8	3.2	4x2	89.13	7158	142x2	399
3	Dual E5430	8	3.2	4x2	89.13	7158	142x2	399
4	Dual E5430	8	3.2	4x2	89.13	7158	142x2	399
5	Dual E5430	8	3.2	4x2	89.13	7158	142x2	399
6	Dual E5430	8	3.2	4x2	89.13	7158	142x2	399
7	Dual E5430	8	3.2	4x2	89.13	7158	142x2	399
8	Dual E5430	8	3.2	4x2	89.13	7158	142x2	399
9	Dual 5405	8	2.4	4x2	68.04	5456	109x2	311
10	Dual 5405	8	2.4	4x2	68.04	5456	109x2	311
11	Dual 5405	8	2.4	4x2	68.04	5456	109x2	311
12	Dual 5405	8	2.4	4x2	68.04	5456	109x2	311
Rig Totals:		120	37.528	96	1012.96	82556	3276	4770.6

*Laptop, † INTEL LINPACK V11.1.1, ‡ PSO Based on Flights No.=1024x12, Cluster Size=16, Swarm Size=16

Table 5.2.1: CATM Simulation Rig Configuration

The slave nodes are dedicated for numerical processing and centrally managed from the head node to which a user interface is connected. The head node also handles a centralised file system and this explains the higher specification. However, this node contributes to the processing effort like the rest. Four of the nodes are lower specified than the rest and these were used to conserve energy during the long development times. These can be upgraded later if necessary with a simple processor swap.

The motherboards were selected to accommodate future expansion of processing capacity in the form of GP-GPU co-processors. This will allow a hundred-fold further increase in processing power to accelerate selected portions of the optimizer algorithms, and complex objective functions when higher fidelity aircraft models are introduced.

5.3 CATM: PSO C++ IMPLEMENTATION

After having understood, implemented and compared the basic candidate algorithms in Matlab, PSO was eventually selected as the most promising technique for the global search portion of the CATM problem. The search mechanism seems particularly well adapted to non-convex optimisation of continuous-domain problems such as multi-trajectory optimisation. It has well understood convergence criteria and strikes a good balance between exploration and exploitation, and with the modifications made, can switch between the two modes relatively easily to provide the algorithm with a capability to cater for dynamic scenarios. Clustering was also relatively straight forward to implement.

During development, Matlab served its purpose well as a rapid design tool for exploring and comparing the strengths and weaknesses of numerous algorithm variants and enhancements. Compiled Mex C++ inline functions were used to improve performance of critical sections of the code. However, despite writing very careful Matlab code and frequent profiling, even with Mex, performance, stability and scalability using this vector-based interpreted language is very limited. The decision

was ultimately made to port the enhanced PSO solver to C++ in order to enable scaling up the problem to practical sizes.

C++ brings with it several essential tools that facilitate low level networking (WinPcap), high speed graphics (OpenGL) and low level access to CPU SIMD instructions (SSE4). C++ also brings with it several challenges that were hidden when using Matlab. Vector algebra is no longer a single line affair but must be diligently programmed element by element. There are also very few safeguards from buffer overflows and memory leakages. This required a much more disciplined approach when dealing with processes involving countless arrays that can reach several Gigabytes of memory. However, this low level access brings with it the ability to optimise the code to take advantage of the system's architectural features and the ability to use certain computational/memory efficient techniques that require "promiscuous" access to memory.

5.3.1 Fast Random Number Generators

All probabilistic metaheuristics employ pseudo-random number generators (PRNG) to guide their stochastic behaviour. Such numbers are required during mutation, selection and crossover in GAs and while computing particle velocities in the PSO variants. The large quantity of random numbers required makes the performance of the PRNG particularly critical. This is one of the most frequently called functions in the system. Typically the PRNGs included with most C++ libraries are based on the MT19937 Mersenne Twister algorithm [5.1] and although they have an excellent statistical distribution and a period length of $(2^{19937} - 1)$, they are far too slow for this kind of work. The MT19937 is still the algorithm of choice for the initialisation of the airspace and initial particle positions or initial chromosome definitions. However, when used for the inner loops such as particle velocity updates, a profile of the optimiser revealed that up to 30% of the computing effort was being spent on MT19937 PRNG calls alone.

A simpler PRNG using much fast integer arithmetic was therefore implemented using the Multiply With Carry (MWC1616) algorithm proposed by George Marsaglia [5.2] as a generalisation of prior work [5.3], which concatenates two 16bit MWC results to produce a 32 bit integer result which can then be converted to a 32 bit float. The recursive equations are given in 4.7.1 and provide random number sequences with periods roughly 2^{59} .

The resulting sequence is not cryptographically perfect and shares many similarities with Linear Congruential Generators (LCG), which have some sequential correlation properties (as per Marsaglia's Theorem [5.4]) that make them unsuitable for Monte Carlo methods.

In fact if sequences of N LCG numbers are plotted in N -dimensional space the numbers would align on a series of hyper-planes. Therefore, in respect of the high dimensionality of the problems being addressed, it is important to minimise this risk by decoupling the correlation between sequential draws. The generators were thus reseeded using a Mersenne Twister at every iteration. Separately seeded MWC1616 had to be also used for each dimension. This provision gave perfectly adequate numbers for randomising PSO progress, which should itself be rather insensitive to correlation

anyway due to the intrinsically chaotic nature of its cumulative velocity and position integration equations.

$$\begin{aligned} x_{n+1} &= (Ax_n + C) \bmod 2^{16} \\ y_{n+1} &= (By_n + C) \bmod 2^{16} \\ z &= (x_n * 2^{16} + y_n) / 2^{32} \end{aligned} \quad (4.7.1)$$

where: A is 18000 chosen from a table [18000 18030 18273 18513 18879 19074 19098]
 B is 30903 chosen from a table [29379 29889 30135 30345 30459 30714 30903] (4.7.2)
 x_o, y_o must be seeded with unique numbers

C++ code:

```
unsigned long X = seed1; unsigned long Y = seed2;

float MWC16(void){ X = 18000 * (X & 0xffff) + (X >> 16);
                  Y = 30903 * (Y & 0xffff) + (Y >> 16);
                  return (float)((X << 16) + (Y & 0xffff))/4294967296.0f;};
```

(4.7.3)

5.3.1.1 Uncorrelated Seeding

PRNGs generate numbers that are completely predictable if the seed is known. This is not usually a problem, however in large multi-threaded applications, like the one proposed, each thread needs to have a separate PRNG and these must not generate numbers that are correlated between them in order to ensure that no bias is introduced that would affect the validity of the test results. Therefore it is imperative that each thread is supplied with a unique seed. A single, randomly initialised, Mersenne Twister in the main thread can generate all the seeds for the other threads during spawning. Alternatively, Visual Studio provides a *random_device* class that guarantees uncorrelated random numbers from a non-deterministic source. This is slow, but would only be required once during start-up.

5.3.2 Single Instruction Multiple Data (SIMD) Acceleration

Modern Intel CPUs are equipped with Single Instruction Multiple Data (SIMD) instructions that greatly enhance the throughput of both the arithmetic and floating point unit by allowing groups of 2x or 4x single precision floats or 32-bit integers to be grouped, transferred and processed as a single entity. This is especially convenient when dealing with 4D coordinates and streamlines memory transfers over the 64-bit memory bus and improves cache utilisation. These instructions were chronologically added by Intel to the x86 instruction set in the following sets: MMX, SSE2, SSE3 and SSE4 for multimedia acceleration. Maximum use of such commands was made to leverage these benefits for a 4 to 5 fold improvement in performance. The C++ code excerpt in **Figure 5.3.1** below shows how the PRNG code described in eq. 4.7.3 was implemented using the SSE4 extensions to generate 4 uncorrelated random numbers per function call. Each SSE4 function (shown in purple) corresponds to a single assembly 128 bit SIMD instruction.

```
X = _mm_add_epi32(_mm_mullo_epi32((_mm_and_si128(X,
                                _mm_set1_epi32(0xFFFF))),
                                _mm_set1_epi32(18000))),
               _mm_srli_epi32(X, 0xF));
```

Figure 5.3.1: SSE4 Code excerpt showing the MWC1616 Algorithm

5.4 CATM: C-CCGPSO-D SIMULATION RESULTS

We now move onto discussing some simulation results that have been achieved using the preferred metaheuristic algorithm, C-CCGPSO-D, which was devised by combining a number of ideas that appear suited to solving the CATM problem. We organise the test results as follows, to highlight the salient features of the algorithm:

5.5 Qualitative Results

- 5.5.1 *Cooperative Coevolution in Pathological Cases*
- 5.5.2 *Visualisation of Swarm Collapse*
- 5.5.3 *Visualisation of Convergence in Large Scale Traffic Scenarios*
- 5.5.4 *Visualisation of Convergence with Obstacle Constraints*
- 5.5.5 *Visualisation of Dynamic Optimisation*

5.6 Quantitative Results

- 5.6.1 *Demonstrating Scalability*
- 5.6.2 *Communication Overheads*
- 5.6.3 *The Effect of Transmission filtering*
- 5.6.4 *Clustering Efficacy*
- 5.6.5 *Swarm Size Effects*
- 5.6.6 *Constriction Effects*
- 5.6.7 *Delay Distribution*

The following abbreviations will apply throughout the discussion.

CCGPSO: Cooperative Coevolution Guided Particle Swarm Optimisation
 C-CCGPSO: Clustered Cooperative Coevolution Guided PSO
 C-CCPSO-D : Clustered Cooperative Coevolution PSO with Dynamic behaviour
 C-CCGPSO-D: Clustered Cooperative Coevolution Guided PSO for Dynamic problems

The key technical achievements are:

- 1) Demonstration of a linear relationship between problem size and computation required using the proposed C-CCGPSO-D CATM algorithm.
- 2) Demonstration of dynamic optimisation with moving obstacles
- 3) Demonstration of effective inter-aircraft collision avoidance while optimising trajectories with respect to length.
- 4) Demonstration of effective obstacle avoidance, while still maintaining separation minima and optimising trajectories with respect to length.
- 5) Demonstration of far higher achievable traffic densities (24-fold) with less delay than incurred today, while still maintaining acceptable flight separation.
- 6) Demonstration of clustering as an effective means of dividing the problem space
- 7) Demonstration of effects of swarm size, ideal attracts and optimal constriction
- 8) Demonstration of modest communication requirements by today's standards.

5.5 QUALITATIVE RESULTS

These qualitative results demonstrate the behaviour of the C-CCGPSO-D under specific demanding conditions that were especially designed to stress the dynamic solver with particularly difficult cases.

5.5.1 Cooperative Coevolution in Pathological Cases

In order to test the equitable self-organisational features of the algorithm, a perfectly symmetric pathological scenario was set-up involving 14 flights from all opposing corners and faces of a test cube, of sides 10 km. The flights travel in both directions, flying through a central choke point to the diametrically opposite position. No obstacles are involved.

This kind of scenario involves difficult decision making that should ideally result in a perfectly balanced result, where all the flights contribute with comparable elongation and where no flight is unduly penalized, such that the overall system cost is minimised. System cost is measured in terms of total percentage elongation across all 14 flights.

The separation constraints of the aircraft are set such that the octahedral buffer zones around each aircraft have sides of 1000 meters. This should result in the aircraft settling at a minimum separation of $1000/\sqrt{3} = 577.35\text{m}$ apart.

For this case, a swarm of 64 particles was used, trajectories were discretized with 32 knots, and all clusters were sized to the whole fleet, which translates to CCGPSO-D. As can be seen in **Figure 5.5.1**, the system converged to the expected ideal solution, where each flight deviated by just the right amount to avoid colliding with any other flight. The process of convergence is shown in three steps, where **(a)** shows the result during early exploration, **(b)** already shows a very good approximation after just 1.25 seconds and **(c)** shows the asymptotic converged state after 8,000 iterations that take about 20 seconds to complete on a 1.6GHz i7 Laptop. **Figure 5.5.2** shows the progression of convergence. Systemic inefficiency dropped to just 0.5% as shown in **(a)**, while the separation constraints were, met nearly exactly with a minimum final separation of 577.71m

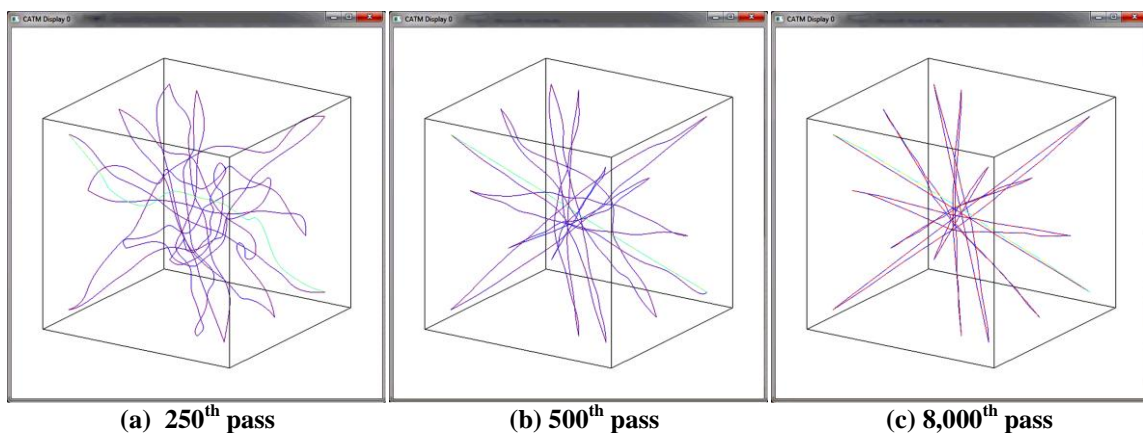


Figure 5.5.1: Convergence of CCGPSO with a 14-way choke-point scenario

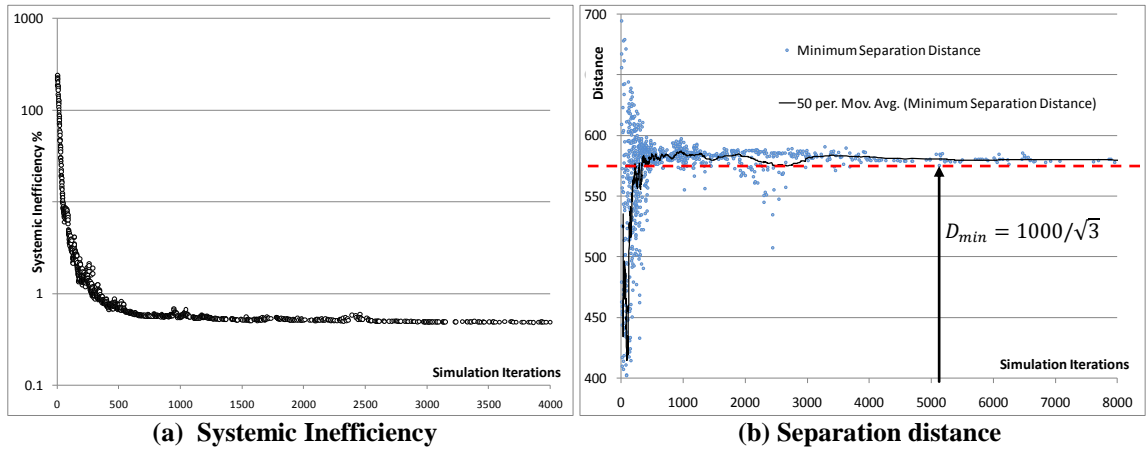


Figure 5.5.2: Convergence Charts of CCGPSO with a 14-way choke-point scenario

5.5.2 Visualisation of Swarm Collapse

The process of convergence is best appreciated when the actual particle swarm is visualised at various points in the process. In this case, each “particle” is in effect a candidate trajectory joining the departure point to the destination. Therefore in **Figure 5.5.3 (a)** a swarm of particles can be seen sampling the relevant search space in all sorts of directions. By the 50th pass **(b)**, the swarm quickly begins to take shape around the regions containing the global optimum. By the 200th pass **(c)**, the swarm has identified the neighbourhood of the global optimum, but is still significantly in breach the separation constraints. By the 1000th pass **(d)**, the intensification process is well under way and the particles crowd very densely around the global-best particle as they seek the lowest cost set of trajectories. The optimum relative positioning of each trajectory has been determined and it is now a question of refinement. By the 4000th pass **(e)**, the swarm has practically collapsed. The problem is essentially solved, although there is still some slight room for improvement.

At this point the metaheuristic became very ineffective, and progress slowed down to a crawl. At this stage, the solution found was well within a locally convex domain so there was little point continuing with probabilistic techniques. The solution is ideally transferred to a local optimiser, that starting with this seed, can reliably improve the result to perfection, limited only by the numerical accuracy of computing platform being used.

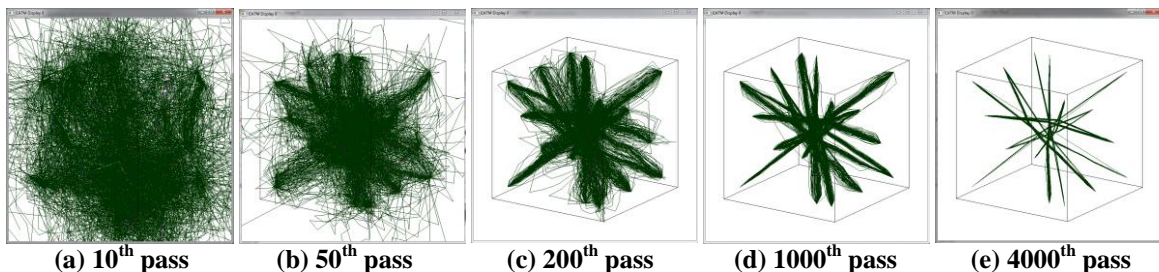


Figure 5.5.3: Visualization of Swarm collapse a 14-way choke-point scenario

5.5.3 Visualisation of Convergence in large scale traffic scenarios

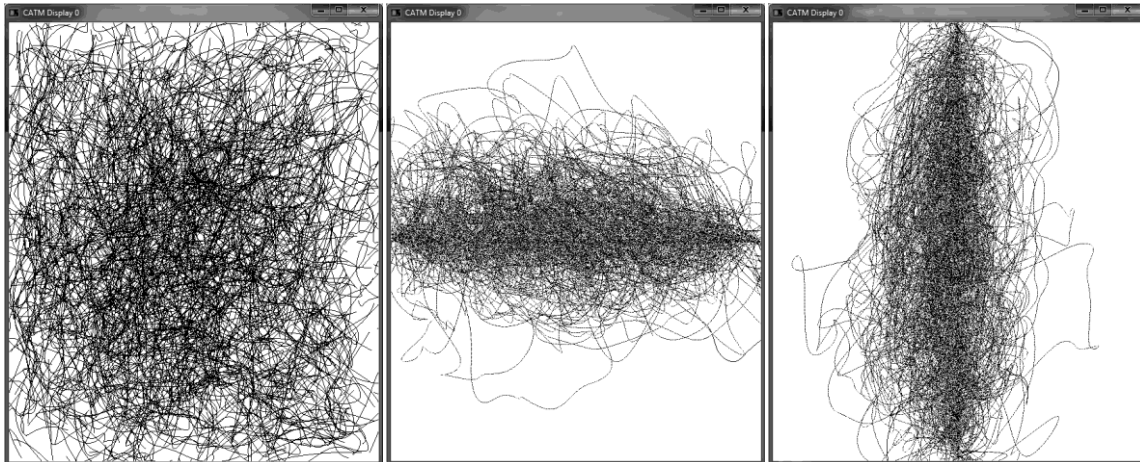


Figure 5.5.4 (a): After 2000 iterations: 232% Systemic Inefficiency, Closest Encounter = 2.05km

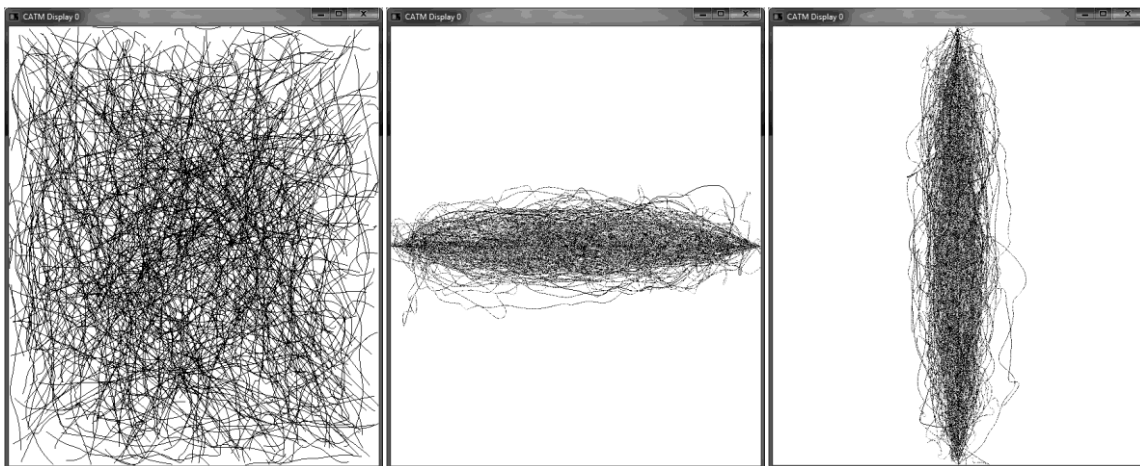


Figure 5.5.4 (b): After 40,000 iterations: 42.4% Systemic Inefficiency, Closest Encounter = 4.98km

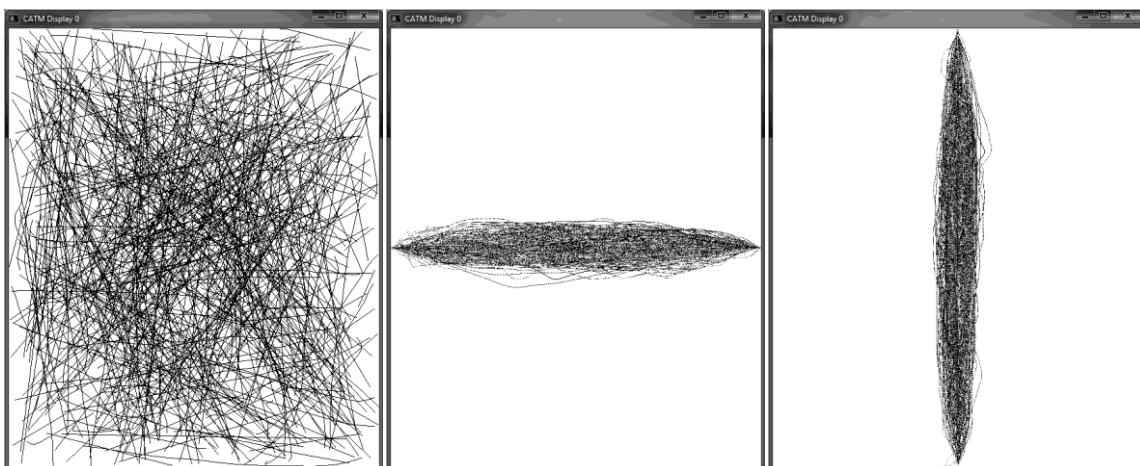


Figure 5.5.4 (c): After 1,000,000 Iterations: 6.01% Systemic Inefficiency, Closest Encounter = 5.58km

The next stage of testing involved high density traffic between points lying on a plane. This is much more representative of real traffic and in order to keep the simulation unbiased and independent of specific circumstances, a random set of airport locations was generated at every run of the optimiser. This ensured that uncorrelated trials were used when running Monte-Carlo simulations, which is essential for the validity of the results.

Airports were uniformly distributed in a square plane of 1000km x 1000km. However, a number of exclusion rules were enforced to avoid contentious or unrealistic scenarios that would unreasonably skew the results:

- 1) No airports were placed closer than 10km apart.
- 2) No airports were placed within restricted airspace.
- 3) No flights between airports closer than 50km were generated.

The visualisations shown in **Figure 5.5.4 (a)-(c)**, depict the afore-mentioned scenario, with 512 random flights, with 16 knots per flight and 4 fold NURB interpolation. The flights are clustered in groups of 16, where each flight is optimised using a swarm of 16 particles. No restricted airspace was defined in this case, but separation minima were set at 5km. Optimal constriction was used with ideal attractors, which translates to C-CCGPSO-D.

The first set of images in **(a)**, shows the state of the best solution found after 2000 iterations from three different viewpoints. It is clear that progress is much slower in this case than it was in the simpler 14 flight choke-point scenario. After 2000 iterations the metaheuristic is still in exploratory phase, but this is hardly surprising given the hundreds of thousands of individual local minima that arise in such a large problem. The systemic inefficiency is still at 232% in relation to the ideal straight line trajectories, while the closest pair of trajectories are 2.05 km apart. The non-linear conflict penalty function, ensures that close encounters are associated with disproportionately high cost, thereby ensuring that breaches of separation are given high priority.

After 40,000 iterations **(b)**, the systemic inefficiency drops to 42.4%, and the closest encounter is at 4.98km, just 0.02km short of the 5km constraint. Further exploration is necessary to reorder the flights more optimally. At 1,000,000 iterations **(c)**, the systemic inefficiency drops to 6.01%, at which point progress is very slow and the closest encounter is at 5.58km. This is probably not the global optimum but it is unlikely that much better results will be obtained from this point onwards since the swarm is now fully collapsed and the optimizer is trapped in some local minimum. However, this result is also probably quite close to the global optimum and any further improvement is probably not worth the effort.

5.5.4 Visualisation of Convergence with obstacle constraints

When obstacles were introduced to simulate the effect of restricted airspace the resulting systemic inefficiency increased. But this is totally understandable given the need for many detours. Five regular octahedral obstacles with sides of 150km were introduced into the middle of the previously described airspace as shown in **Figure 5.5.5**. The same optimisation conditions and size of problem were retained as in the previous case.

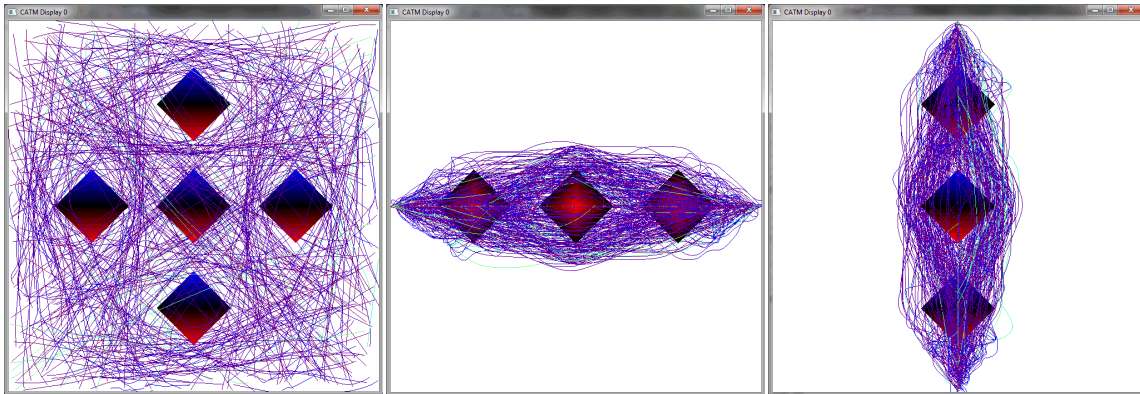


Figure 5.5.5: After 5,000,000 Iterations: 17.8% Systemic Inefficiency, Closest Encounter = 4.30km

The converged steady-state solution after 5 million iterations is shown in three different views in **Figure 5.5.5**. The obstacles caused flights to concentrate between them, and this resulted in higher regional density. In this case the metaheuristic, sought to balance meeting the separation constraints, the obstacle constraints and the systemic inefficiency through the cost weighting function. Since separation criteria were expressed as *soft constraints* using a non-linear function, some degree of flexibility was retained, and the optimizer traded off some inter-flight separation for lower systemic inefficiency and obstacle interaction.

5.5.5 Visualisation of Dynamic Optimisation

One of the key elements that make PSO attractive for this kind of work is the relative ease with which the basic algorithm can be adapted for dynamic behaviour. This was tested using a simple experiment consisting of a single tetrahedral object at one side of a dynamically converged set of trajectories as shown in **Figure 5.5.6 (a)**. The object was plunged through the airspace, while the behaviour of the swarm was closely analysed. This scenario represents the motion a non-traversable weather cell through the airspace, which would typically force many aircraft to re-route reactively.

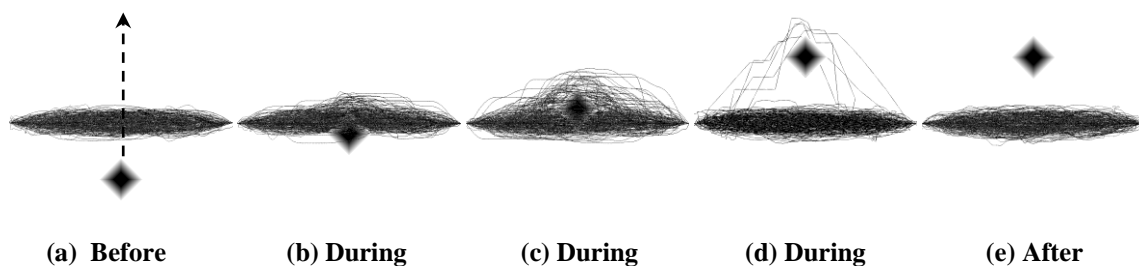


Figure 5.5.6: Effects of an object collision with a dynamically converged solution

After the fleet of flights reached dynamic equilibrium with their constraints, the trajectories behaved like elastic bands stretched between their terminals. As the object collided with the fleet **(b)**, the swarm re-expanded. The converged *G_{best}* particle was suddenly no longer considered ‘best’ in these circumstances in view of the high cost incurred due to the collision. The underlying swarm dispersed, in search for a new optimum that kept changing as the object moved through the fleet **(c)**. This re-acquired diversity was what allowed the swarm to dynamically adjust to the infolding scenario.

One advantage of using octahedral buffer zones is that they do not usually create non-convexities in 3D space. The trajectories tended to slide off them quite easily due to the gradient (c). However, a few flights took longer to re-settle as they got stretched around the flat faces of the object (d). This is undesirable and is symptomatic of insufficient swarm diversity. If the swarm diversity were high, this local optimum would quickly be abandoned in favour of much better solutions. Regular resets would solve the problem trivially (e) but incur a high performance penalty as the metaheuristic loses all its memory to start afresh. A much better solution involved the continuous replacement of the most converged particles with new random particles, such that the swarm was continually regenerated.

The effect of collisions during final convergence is beautifully depicted in the chart shown in **Figure 5.5.7**. Systemic inefficiency increased temporarily as the trajectories rerouted with longer paths round the moving object. However, this was temporary, because as soon as the object moved out of the centre of the airspace the swarm quickly reconverged to the optimal state and returned back from where it left off along the exponential convergence decay curve. The effect was repeated when the object retraced its path back to where it started.

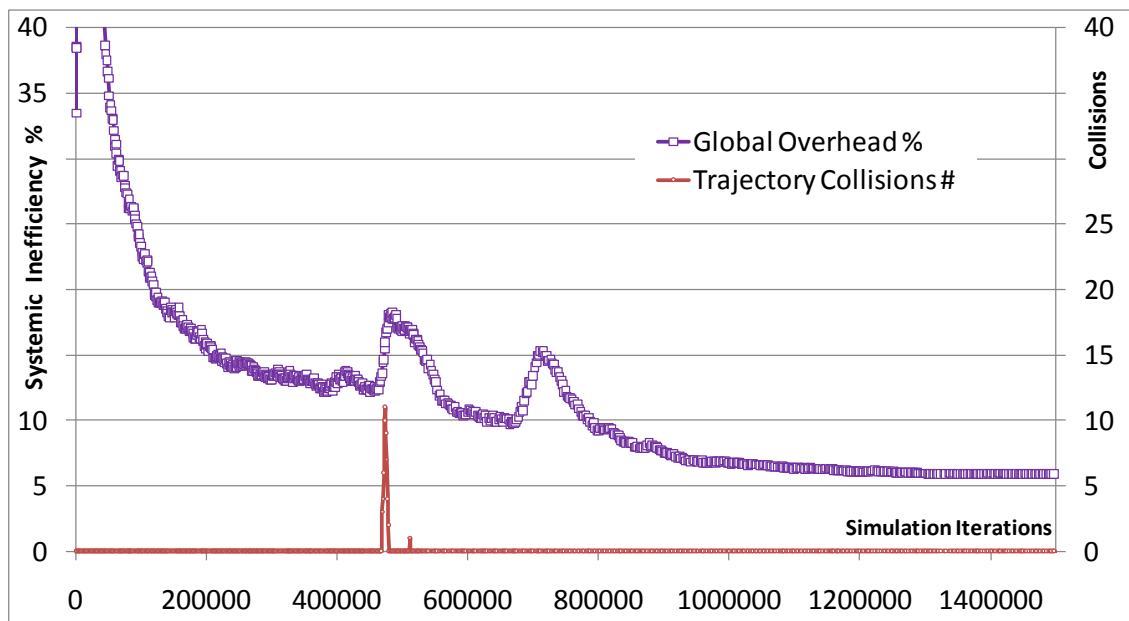


Figure 5.5.7: Transient caused by an object collision during convergence

Also shown in the chart, is the tally of trajectory collisions, which suffered a brief transient, due to the initial compression, only to return back to zero very soon after. Collisions were defined as breaches of the 5km buffer zone. In reality, the trajectories remained a substantial distance apart throughout the event, and the probability of collision was still very small. Besides, this was a worst case scenario that unfolded over a matter of seconds, when in reality weather cells take days to form, and move about, and although the aircraft models would need to be much more sophisticated, a CATM system would have vastly more time and computing resources at its disposal than was afforded in this small simulator. In addition, trajectories would be calculated well in

advance of actual flying. This would ensure that it would have re-converged fast enough to avoid all breaches of separation at the outset.

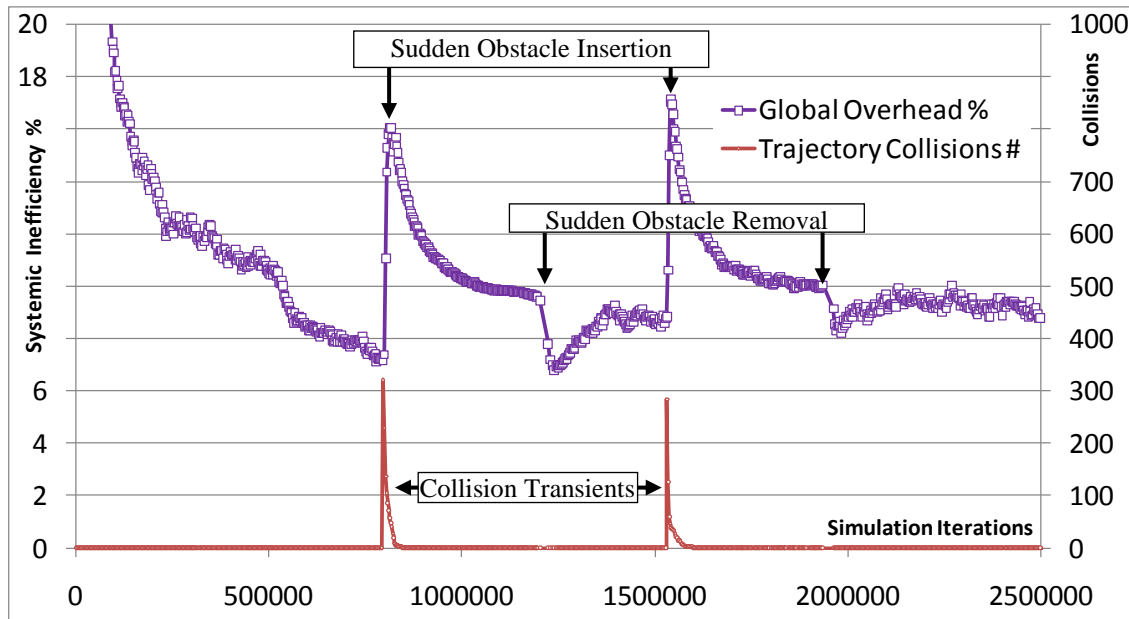


Figure 5.5.8: Transient due to a sudden object collision with a dynamically converged solution

The chart in **Figure 5.5.8** shows the effect of much harder scenarios. The obstacles were in this case introduced suddenly into the middle of the airspace after initial convergence of the swarm. This was performed twice in a row, removing the obstacle before re-introducing it and removing it again. It simulated the effects of catastrophic disruptions. As can be seen, there was again a brief transient in the number of trajectory collisions, but this was not repeated when the obstacles were removed.

Systemic inefficiency also suffered a sharp transient as the swarm rapidly rerouted to accommodate the offending obstacle. It then went through a period of re-optimization during which it attempted to find a new optimum taking into account the obstacle. On removal of the obstacles, the efficiency improved very rapidly.

All the tests used to show dynamic behavior were conducted with 512 random flights, using 16 knots per flight and 4 fold NURB interpolation. The flights were clustered in groups of 16, where each flight was optimised using a swarm of 16 particles. Optimal constriction ($m = 1.0$; $C_1 = C_2 = 2.00$; $C_3 = 0.117$) was used with ideal attractors, which translates to C-CCGPSO-D.

5.6 QUANTITATIVE RESULTS

A number of quantitative tests were performed to show the relative performance of the algorithm under different operating conditions and problem sizes. Although direct comparisons are hard to make when there are so many influences that need to be taken into account, the achieved results compare favourably with the traffic statistics extracted in Chapter 3. Higher fidelity simulations are obviously possible at the expense of far more engineering effort and computational load. In the absence of that, it was decided to standardise the tests in a way that can make performance comparisons between algorithms easier and statistically valid.

5.6.1 Demonstrating Scalability

One of the primary objectives of this project was to demonstrate scalability, which is to say that the algorithm must be able to operate and properly converge whatever the size of the problem. This would obviously require an appropriate increase in processing power. However, this must also scale linearly (or polynomially) with the problem size in order to prove practical in the real domain.

Thus an experiment was devised, whereby the simulation was run a number of times with an exponentially increasing number of flights as shown in **Table 5.6.1**. This covered 8 octaves, starting with 32 flights all the way to 4096 flights. The flights were confined to a square of 1000 x 1000 km. However, the processing power was kept constant, by sharing the same 3.8GHz machine over the ever increasing number of threads. Therefore, if it can be shown that the time to converge increases linearly with the number of flights, then scalability is proved and assured. Any increase in complexity to the individual agents such as more sophisticated aircraft models will also contribute linearly to the problem.

<i>CONCURRENT FLIGHTS (#/10⁶km²)</i>	<i>TRAFFIC DENSITY 10⁴NM²</i>	<i>CONVERGENCE TIME TO 110% OF BEST (SECONDS)</i>	<i>MEMORY FOOTPRINT (MB)†</i>	<i>ITERATION RATE (ITER/S)</i>	<i>BEST RESULT (%)</i>	<i>FINAL MIN. SEPARATION (KM)</i>	<i>FINAL AVE. SEPARATION (KM)</i>
32	1.1	4.1	25	3200	0.38	45.1	466
64	2.2	13.7	29	6400	0.63	40.2	531
128	4.4	13.3	35	11104	1.88	11.7	547
256	8.8	84.3	49	11071	3.42	14.4	534
512	17.6	203.3	78	10546	6.60	8.1	533
1024	35.1	198.6	140	8457	8.50	4.7	543
2048	70.2	355.3	291	5250	8.57	2.9	545
4096	140.5	1289.9	652	2877	8.55	1.8	537

Using Node at 3.8GHz: Trajectory Knots = 16, Clusters = 16, Swarm Size = 16, Constriction = E, †per Node

Table 5.6.1: Scalability Tests Spaced over 8 octaves

The linearity between problem size and execution time is best appreciated in graphical form as can be seen in the log-log chart given in **Figure 5.6.1 (a)**. However, this convergence time assumes constant CPU capacity. If computing capacity were added to the system in proportion to the number of flights, by housing a complete CATM computing node within each aircraft, the execution time to convergence will be of the order of what it takes to run a single thread: ie: ~ 0.3 seconds in this case, using a single 3.8GHz 8-core Xeon per flight.

The memory footprint also scales approximately linearly with the number of flights and as can be seen the memory requirements are not very taxing on modern computing hardware. In a typical application, the largest data structure would be the almanac, that must allocate space for as many flights as there are aircraft in the system. A local copy of the almanac would be required per flight. However, in this case one is used per node. The iteration rate is also charted, and as expected this also drops with large problem sizes. However, it is interesting to note that the iteration rate was in practice limited by the communications latency for the smallest problem sizes, where a peak iteration rate occurred when there were around 256 flights in the system.

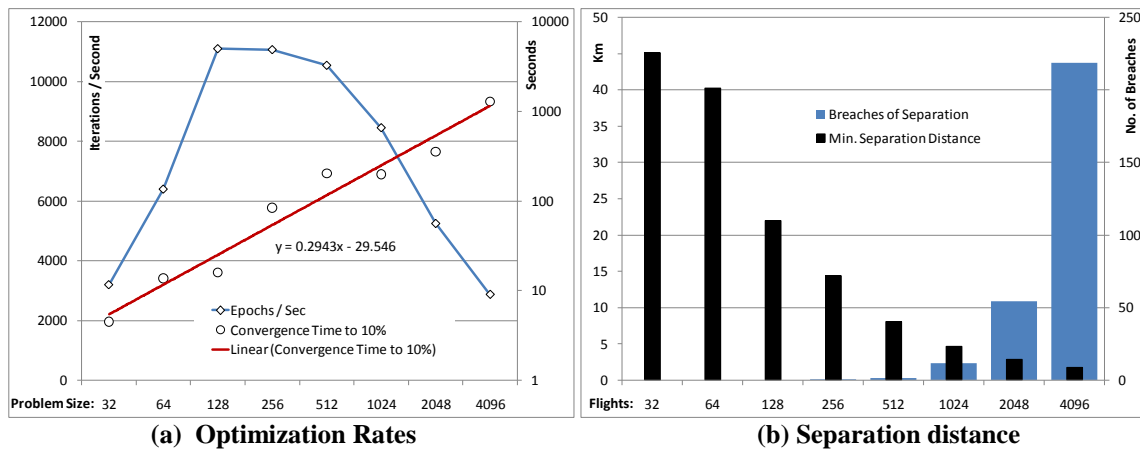


Figure 5.6.1: Performance of C-CCGPSO-D with Variable Problem sizes

The effect of problem size on the separation criteria can be observed in **(b)**, where it can be seen that the C-CCGPSO-D algorithm was able to fully meet the separation constraints when the traffic density was less than 512 flights per 10^6 km². Below this density there were no breaches of separation. Although the number of breaches increased substantially beyond this threshold density, the separation metric decreased gradually. Note that this density is triple the peak density that is found over Europe. This takes place at 2:00pm on the busiest days of summer.

Also note, that a breach of the 5km buffer zone does not imply a collision. It may in fact be desirable to reduce separation minima to increase capacity, so long as higher performance ATM algorithms can be found. We have thus shown that a **24-fold increase** in traffic density over the European *peak* can still be accommodated safely with a comfortable separation distance of 1.8km between the closest pair trajectories, and this with a peak systemic inefficiency (of 8.55%). This is still far better than what is achievable today in Europe, where the *24-hour average* systemic inefficiency for domestic European flights is at around 9.15%.

Reliability of convergence is another important criterion for a practical CATM algorithm. However, this can be amply observed from the very well behaved convergence charts shown in **Figure 5.6.2**. Based on empirical data, irrespective of problem size, C-CCGPSO-D appears to be unconditionally stable. The difference in convergence rates can be explained by the fact that a single machine's processing power was being shared among as many threads as there were aircraft. This should not be the case in practice.

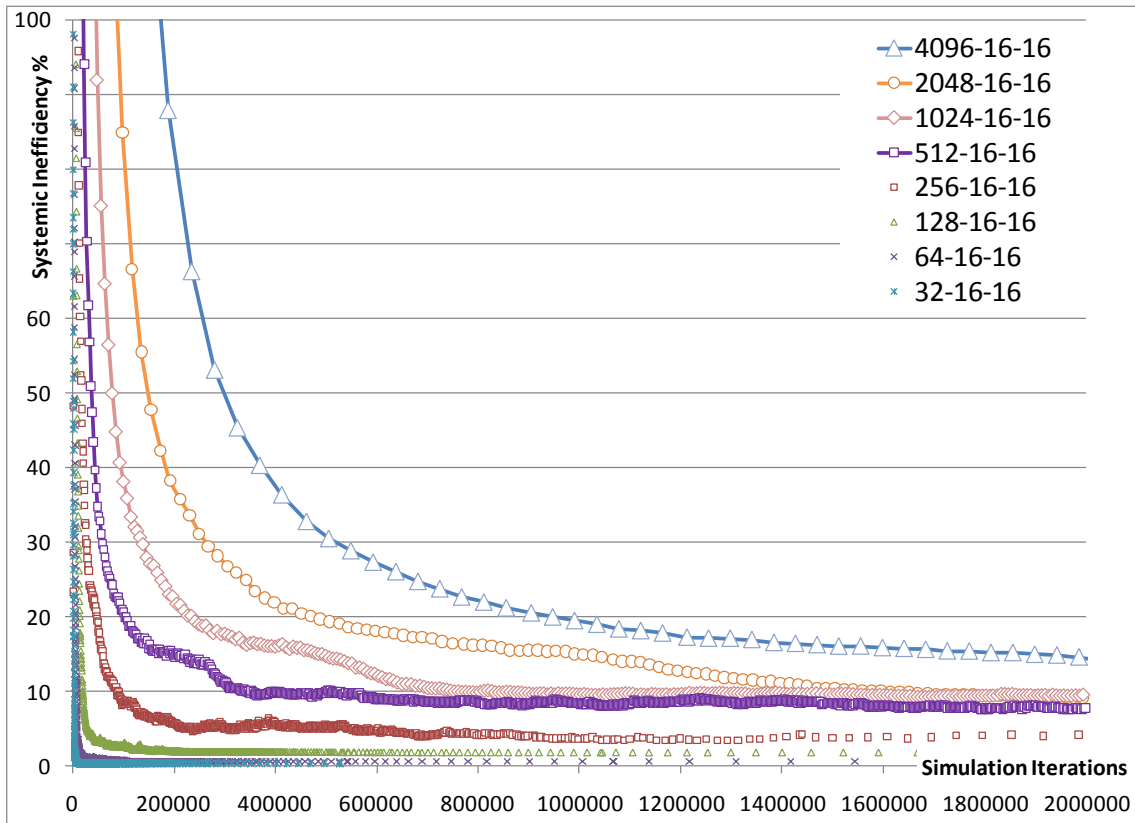


Figure 5.6.2: Reliable Convergence with Variable Problem Sizes

5.6.2 Communication Overheads

One can get an indicative idea of the data rates involved during convergence by looking at **Table 5.6.2**. Here the packets transferred were counted throughout the optimisation run and given the fixed size of each packet (302 bytes for 16 knots) the total data transferred was calculated. The peak data rate was approximately equal across the tests, which makes sense given that identical processing power was allocated. For the larger problems, which took much longer to converge, the difference in cumulative data transferred was quite substantial.

CONCURRENT FLIGHTS (#)	MEMORY FOOTPRINT (MB) [†]	ITERATION RATE (ITER/S)	BEST RESULT (%)	PEAK DATA TRANSFER RATE (KB/S)	TO 110% OF BEST RESULT		
					CONVERGENCE TIME (SECONDS)	FINAL DATA TRANSFER RATE (KB/S)	CUMULATIVE DATA TRANSFER (MB)
512	152	40,920	6.205	6,022	20.2	2,951	149,847
1024	245	26,711	7.773	6,151	71.1	1,545	269,499
2048	458	15,639	8.001	6,728	191.7	817	414,133
4096	1,306	8,511	7.726	5,780	663.4	469	755,358
8192	4,131	4,323	8.125	6,382	2,801.3	81	1,595,323
16384	14,448	2,002	9.073	6,443	14,949.4	11	3,921,156

Using 4 Nodes at 3.2GHz: Trajectory Knots = 16, Clusters = 16, Swarm Size = 16, [†]per System

Table 5.6.2: Communication Overheads

5.6.3 The Effect of Transmission filtering

Transmission filtering had an important role in limiting the total data transferred during the optimisation run. The effect can be seen by comparing the peak data transfer rate with the final data transfer rate. As the problem reached convergence, updates to trajectories became increasingly less frequent and small in magnitude. With transmission filtering, this resulted in the data rate falling to very low levels which is ideal.

This shows that in order to preserve the dynamic equilibrium the sustained data rate need not be very high. Substantial data transfers are only needed when the system is actively adapting to changes in the constraints. Luckily since most of these changes are slow and gradual, such as weather changes, the associated data rate would still not be too high. It is true that the network must be sized for peak transfer rate, but this can be calculated on the basis of local disruption to leave enough capacity for relaying information across the network. There is clearly much more room for research in this area.

5.6.4 Clustering Efficacy

Clustering is another principal feature of the C-CCGPSO-D algorithm that needed to be thoroughly investigated. Two aspects deserved most attention, convergence rate and the quality of the final outcome. For this purpose, a series of tests involving varying cluster sizes were conducted. All tests were based on 512 random flights, using 16 knots per flight, 4-fold NURB interpolation and no obstacles. Each flight was optimised using a swarm of 64 particles. Optimal constriction ($m = 1.0$; $C_1 = C_2 = 2.00$; $C_3 = 0.117$) was used with ideal attractors. The tests started with the single flight cluster, (which meant no cooperative coevolution or clustering at all and corresponds to GPSO-D), and proceeded all the way to 512 aircraft per cluster, which meant all the flights were included into a single cluster, which corresponds to cooperative coevolution without clustering, or CCGPSO-D.

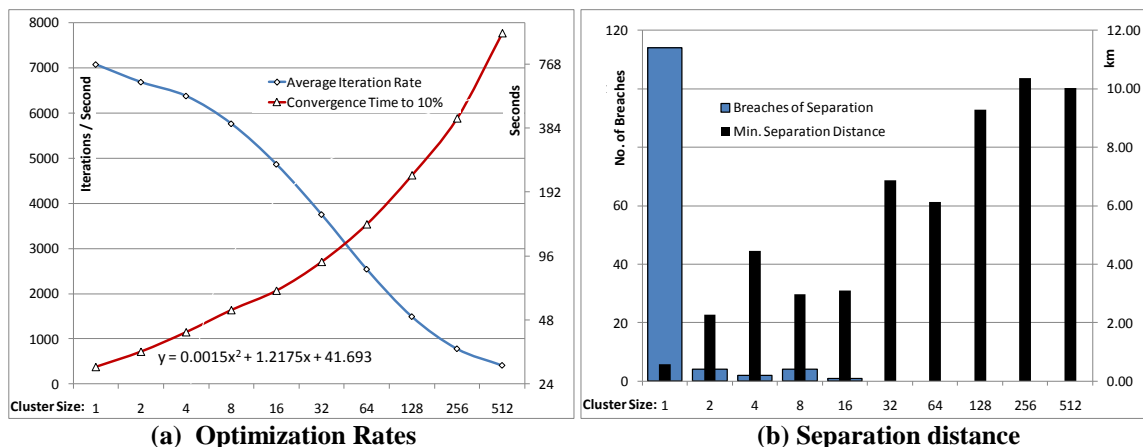


Figure 5.6.3: Performance of C-CCGPSO-D with Variable Cluster sizes

The optimisation rate results are shown in **Figure 5.6.3 (a)**. Convergence time was taken as the time it took for the cost metric to fall to within 10% of the asymptotic final result. This made it much less sensitive to random variations. In **(a)**, it can be seen that the iteration rate drops approximately logarithmically with cluster size, while the

convergence time is quadratically related. This must be interpreted with reference to the observance of separation criteria as shown in **(b)**. It is immediately clear that with no cooperative coevolution, the minimum separation criteria were not adhered to, registering nearly 120 breaches of separation. This improved dramatically with clusters of just two flights until there were no more breaches with clusters larger than 32. When all flights were included into one cluster, very good results were obtained, but this came at a cost of convergence time. A good compromise was attained with clusters of about 32 flights.

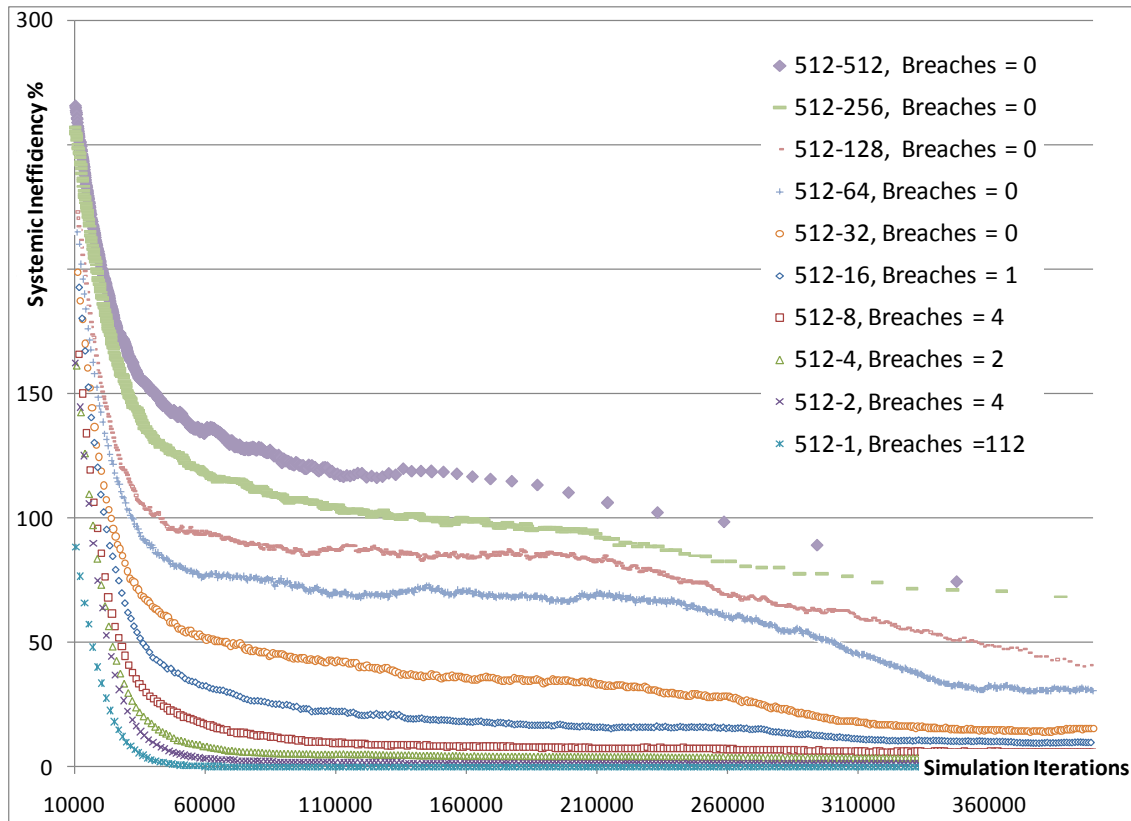


Figure 5.6.4: Reliable Convergence with Variable Cluster sizes

The convergence plots shown in **Figure 5.6.4**, empirically indicate that the C-CCGPSO-D metaheuristic remained unconditionally stable with respect to cluster size. However, it also shows that convergence reached a better result, quicker with smaller cluster sizes, making C-CCGPSO-D superior to CCGPSO-D and superior to GPSO-D with respect to separation.

Therefore, in summary we can conclude that clustering is an effective means of limiting dimensionality growth, while still effective at relaying information throughout the system, such that separation criteria are observed. It appears that clusters of 16 flights strike a good balance with some bias towards good/fast convergence, while using clusters of 32, sacrifices some performance to ensure lower probabilities of buffer zone incursions.

5.6.5 Swarm Size effects

The size of the swarm is related to the searching power of any swarm intelligence metaheuristic. However, large swarms incur heavier memory usage with slower iteration rates. It was therefore interesting to test whether large or small swarm sizes should be used in these kinds of problems.

For this, a series of experiments were run with variable swarm sizes ranging from one particle all the way to 512 particles. The following conditions applied: All tests were based on 512 random flights, using 16 knots per flight, 4-fold NURB interpolation and no obstacles. Flights were clustered in sets of 16 and optimal constriction ($m = 1.0$; $C_1 = C_2 = 2.00$; $C_3 = 0.117$) was used with ideal attractors. In order to avoid getting distracted with the significant variability between individual results, each test was repeated 20 times, and the results were averaged. A measure of variance in the form of relative standard error was provided as a percentage of the quantities shown. **Table 5.6.3** shows a summary of the results obtained.

SWARM SIZE (#)	CONVERGENCE TIME TO 110% OF BEST (SECONDS)	MEMORY FOOTPRINT (MB)	ITERATION RATE (ITER/S)	BEST RESULT [†] (%)	FINAL MINIMUM SEPARATION (KM)	FINAL AVERAGE SEPARATION (KM)
1	80.6 (±4.0%)	222	18117 (±2.9%)	10.97 (±1.0% ²)	3.30 (±9.9%)	545 (±0.4%)
2	69.7 (±3.7%)	223	17320 (±3.0%)	9.72 (±1.3% ²)	4.11 (±8.9%)	539 (±0.5%)
4	70.8 (±4.2%)	225	15796 (±2.9%)	8.58 (±1.4% ²)	4.75 (±8.2%)	539 (±0.5%)
8	68.3 (±5.1%)	229	13109 (±3.0%)	7.80 (±1.8% ²)	4.71 (±10.4%)	544 (±0.4%)
16	81.6 (±6.5%)	238	9611 (±2.9%)	7.09 (±4.5% ²)	5.33 (±7.7%)	540 (±0.5%)
32	105.4 (±3.5%)	256	6330 (±3.1%)	6.57 (±2.1% ²)	5.67 (±5.5%)	540 (±0.4%)
64	126.1 (±4.4%)	290	3922 (±2.8%)	6.48 (±2.0% ²)	5.66 (±8.5%)	544 (±0.4%)
128	156.8 (±5.7%)	357	2132 (±3.1%)	6.03 (±2.2% ²)	5.65 (±9.0%)	548 (±0.4%)
256	211.9 (±4.3%)	495	1128 (±3.2%)	5.76 (±1.9% ²)	5.95 (±7.2%)	541 (±0.5%)
512	224.3 (±8.0%)	770	606 (±3.0%)	6.85 (±4.3% ²)	6.71 (±6.5%)	547 (±0.5%)

Tests run for 20 times, results expressed as mean Qty (± Std. err %), † Systemic Inefficiency
Using 3.2GHz Nodes: Flights = 512, Trajectory Knots = 16, Clusters = 16, Constriction = E

Table 5.6.3: The Effect of Swarm Size

These results are better appreciated graphically as shown in **Figure 5.6.5 (a)**, where a nearly linear relationship can be observed between convergence time and population size. This held true for much of the range and only broke down with very small population sizes, where the time consumed in overheads greatly exceeded the time spent processing the miniscule swarm at each iteration. The iteration rate fell quasi-logarithmically with increasing population size. At the same time, a quick look at **Figure 5.6.6**, reveals that larger swarms were conducive to better final results, with the lowest values of systemic inefficiency achieved using the 256 particle swarm. This was coupled with significantly better performance in terms of meeting separation constraints, where a trend was very clearly shown. The larger swarms lead to higher final separation distance and fewer breaches. This leads us to the conclusion that swarms somewhat larger than 32 particles result in the best compromise between the conflicting requirements of convergence time vs performance.

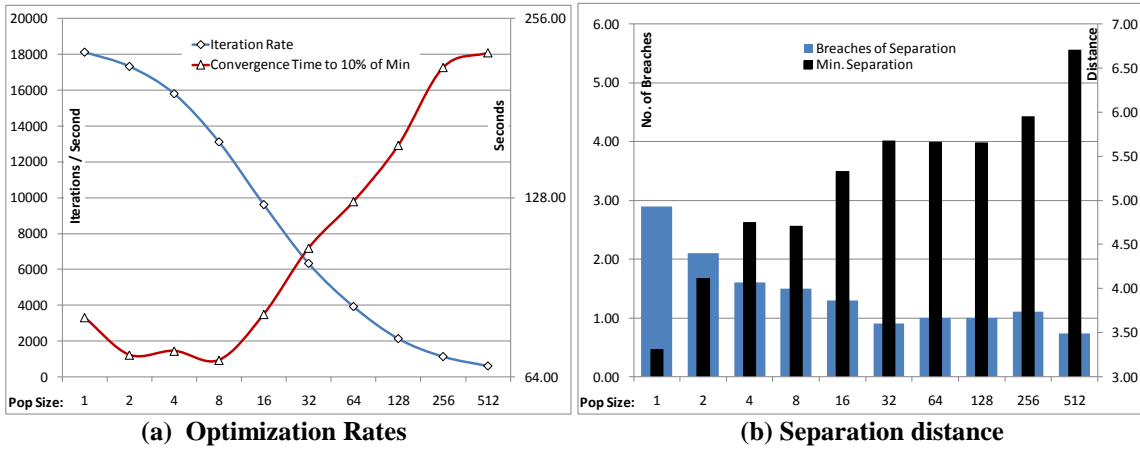


Figure 5.6.5: Performance of C-CCGPSO-D with Variable Swarm sizes

The plots shown in **Figure 5.6.6** are also the result of 20-fold averaging, such that the relative trends shown are truly representative of the likelihood of obtaining such a result. This was in effect equivalent to a small Monte-Carlo simulation, where each trial was attempted using a different random set of flights and airports.

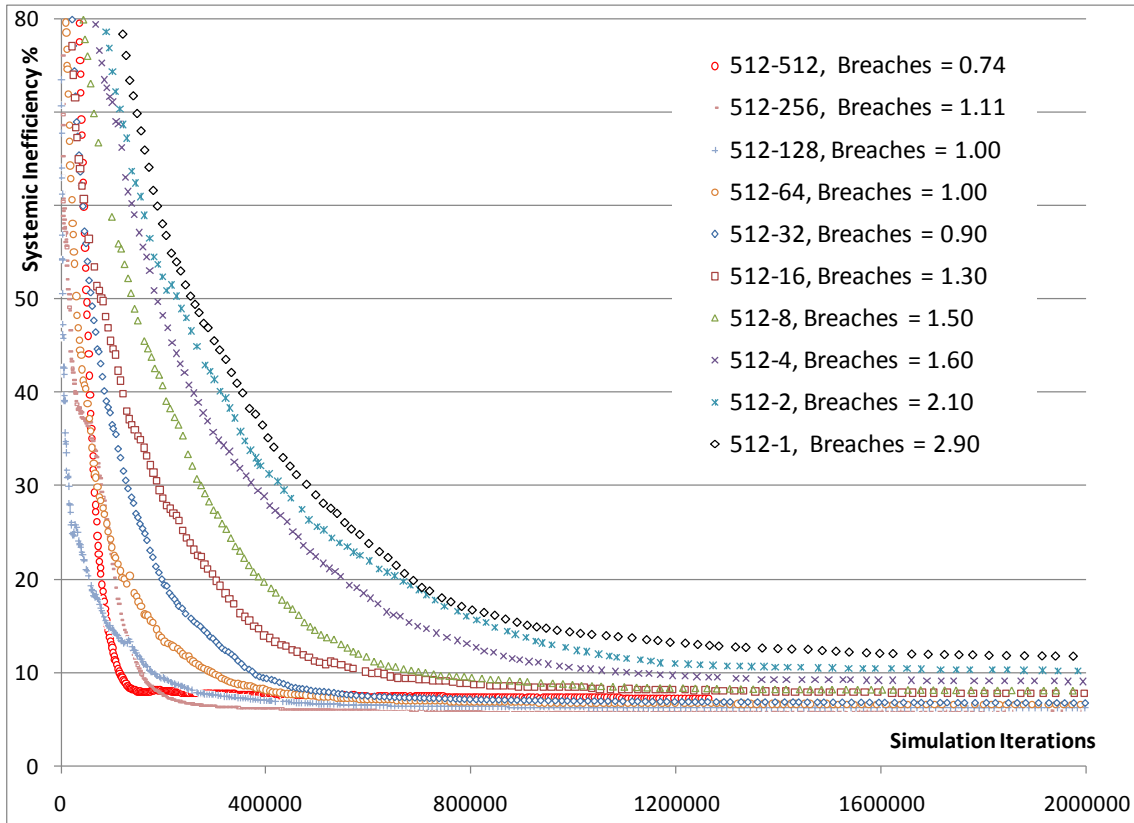


Figure 5.6.6: Reliable Convergence with Variable Swarm sizes

5.6.6 Constriction effects

Constriction parameters control the convergence properties of PSO swarms in accordance to a well defined theoretical framework. However, there remains some degree of freedom in the choice of these parameters. The addition of ideal attractors, changed the standard PSO formula and a new parameter was added. In order to ensure that the best choice of parameters was selected, the following tests, as shown in **Table 5.6.4**, were conducted.

#	m	c_1	c_2	c_3	CONVERGENCE TIME TO 110% OF BEST (SECONDS)	BEST RESULT (%) [†]	FINAL MINIMUM SEPARATION (KM)	FINAL AVERAGE SEPARATION (KM)
A	0.9375	2.0500	2.0500	0.000	181 ($\pm 4.4\%$)	9.52 ($\pm 2.2\%^2$)	6.24 ($\pm 12.4\%$)	539 ($\pm 0.5\%$)
B	1.0000	2.0585	2.0585	0.000	187 ($\pm 2.5\%$)	9.43 ($\pm 2.5\%^2$)	6.02 ($\pm 13.1\%$)	544 ($\pm 0.5\%$)
C	1.0000	2.0550	2.0550	0.007	189 ($\pm 3.0\%$)	7.79 ($\pm 1.9\%^2$)	5.32 ($\pm 12.5\%$)	551 ($\pm 0.5\%$)
D	1.0000	2.0500	2.0500	0.017	163 ($\pm 3.2\%$)	7.17 ($\pm 2.5\%^2$)	8.32 ($\pm 5.4\%$)	544 ($\pm 0.5\%$)
E	1.0000	2.0000	2.0000	0.117	125 ($\pm 3.7\%$)	7.19 ($\pm 2.6\%^2$)	7.42 ($\pm 7.6\%$)	553 ($\pm 0.6\%$)
F	1.0000	1.7500	1.7500	0.617	64 ($\pm 46.3\%$)	11.11 ($\pm 3.1\%^2$)	6.71 ($\pm 11.4\%$)	551 ($\pm 0.5\%$)

Tests run for 10 times, results expressed as mean Qty (\pm Std. err %), [†] Systemic Inefficiency

Using all 3.2GHz Nodes: Flights = 512, Trajectory Knots = 16, Clusters = 32, Swarm Size = 64

Table 5.6.4: Swarm Constriction Effects

Six scenarios were devised. The first two {A, B}, pertained to standard PSO and included no ideal attractor term (which is therefore zero). The next 4 scenarios {C, D, E, F}, progressively increased the ideal attractor term, at the expense of the other terms, while ensuring that the total constriction parameter remained well above 2.00. From the table and **Figure 5.6.7**, it can be seen that the best performance is achieved with option E.

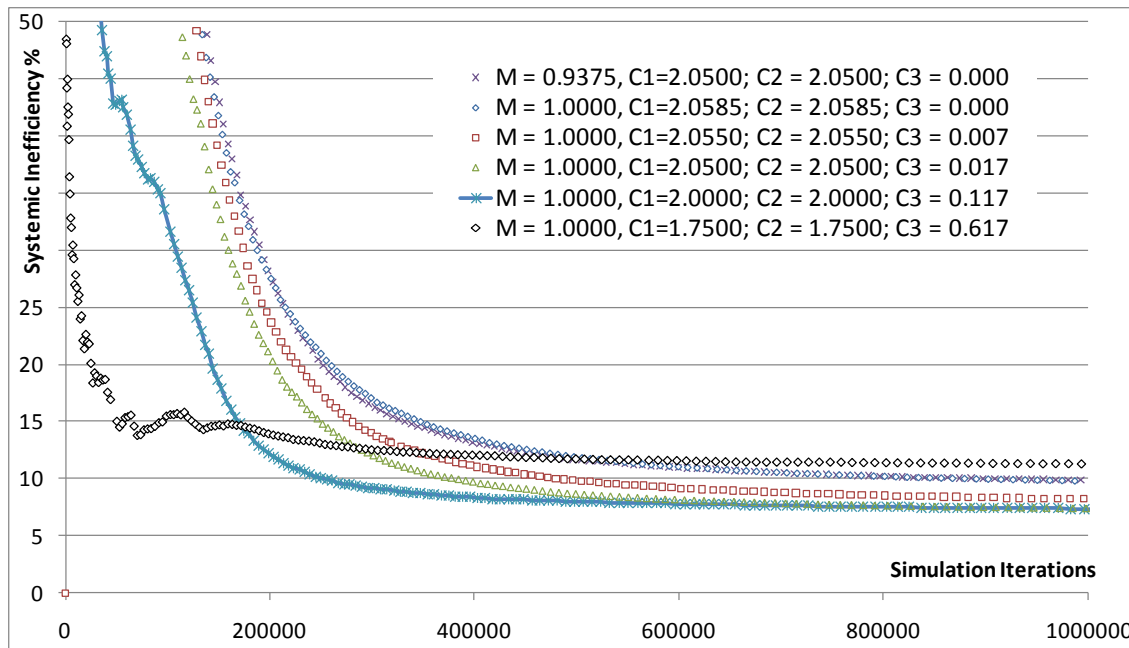


Figure 5.6.7: Performance of C-CCGPSO-D with Variable Constriction

For this series of experiments, the swarm size was fixed to 64. All tests were based on 512 random flights, using 16 knots per flight, 4-fold NURB interpolation and no obstacles. Flights were clustered in sets of 32 and the type of constriction was varied. In order to avoid getting distracted with the significant variability between individual results, each test was repeated 10 times, where each trial was attempted using a different random set of flights and airports.

Options {A, B}, offered no guidance to the swarm, thereby missing out on the opportunity to impart important information to the metaheuristic. Progress was slower than the other options and the swarm was ultimately trapped into a local minimum that was substantially less optimal than what could be achieved in practice. **Table 5.6.4** shows how marginally acceptable separation was achieved at a relatively high cost of nearly 10% average trajectory elongation. Options {C, D, E, F}, imparted a progressively stronger bias towards the ideal solution. However, too much bias lead to fast, but premature, convergence to a suboptimal local minimum, as can be seen in **Figure 5.6.7**.

Option E, achieved the best balance between exploration and exploitation, by guiding the swarm towards a global best region, without unduly hampering it from searching for the best balance with the path constraints.

5.6.7 Delay Distribution

As a measure of equitability, delay or flight elongation histograms were used to measure the distribution of inefficiency in the system.

With a swarm size of 64 particles, all tests were based on 512 random flights spread over 10^6 km², which corresponds to 3 times the current summer peak European density of traffic. This can be considered as representative of 2035 traffic density. 16 knots per flight were used with 4-fold NURB interpolation and no obstacles. Flights were clustered in sets of 16 and constriction type E as used. At various points during the convergence, the best trajectories found for each aircraft in the fleet were binned in terms of their individual inefficiency, measured as the percentage elongation of each flight. Given, that aircraft tend to travel at quasi-constant cruising speed, this is a good measure of the delay incurred.

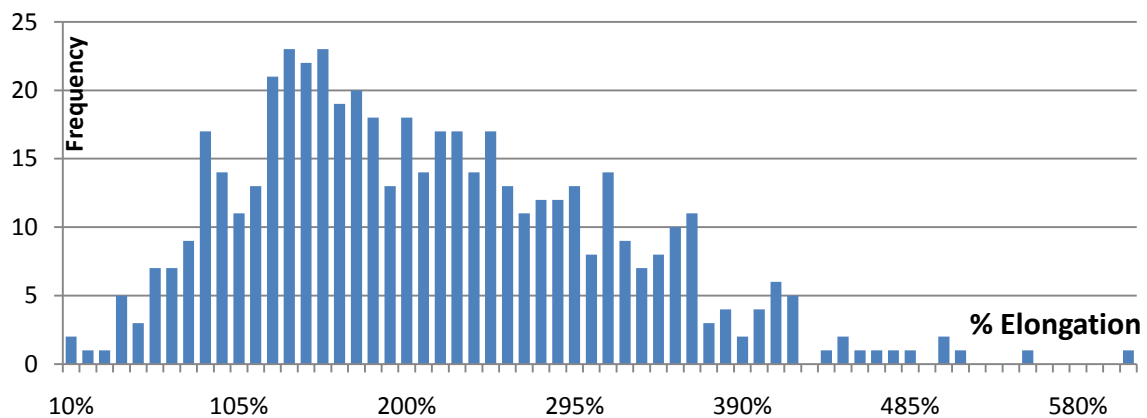


Figure 5.6.8: Delay distribution of flights after 10,000 iterations.

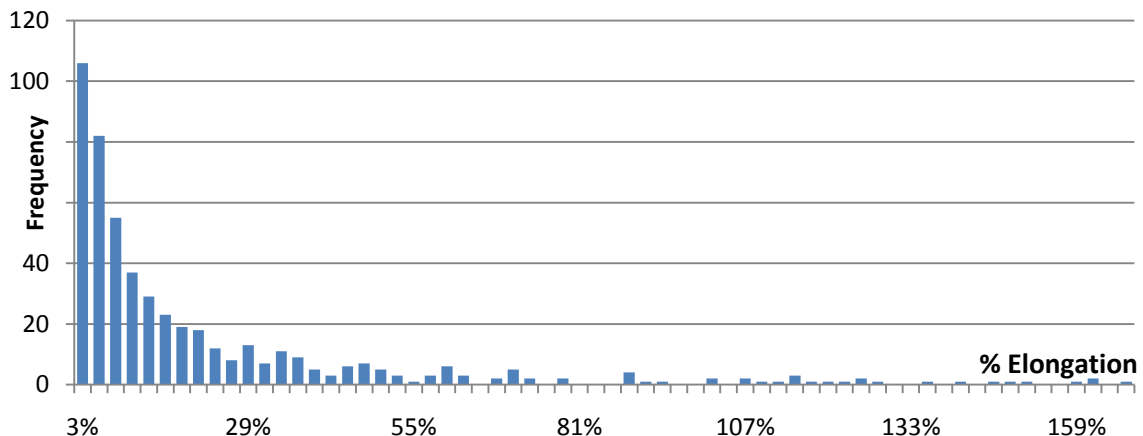


Figure 5.6.9: Delay distribution of flights after 100,000 iterations.

After the first 10,000 iterations, the flights were still influenced by their random initialisation and **Figure 5.6.8**, shows a roughly Gaussian distribution of flights in terms of elongation. This was expected from the central limit theorem that states that any summation of uncorrelated quantities drawn from a random source will tend towards Gaussian distributions, when the number of elements is large. In this case, each flight consisted of the concatenation of 16 random segments.

As the optimisation proceeded, the distribution changed as flights were progressively shortened towards their optimum lengths. **Figure 5.6.9**, shows the effect of optimization on the distribution after 100,000 iterations, where the spread of delay is greatly reduced and most flights are now experiencing less than 30% elongation.

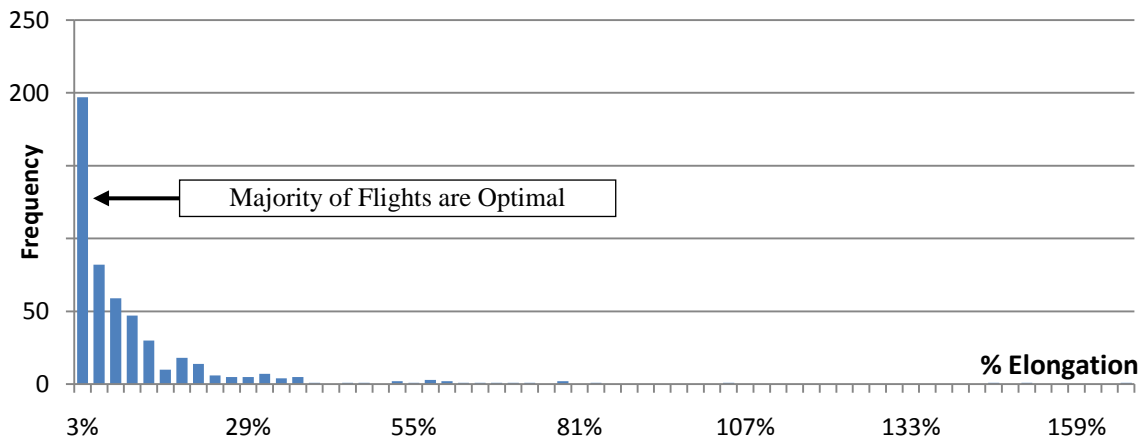


Figure 5.6.10: Delay distribution of flights after 1,000,000 iterations.

Asymptotic convergence was almost achieved by 1,000,000 iterations, where the distribution of flight delay looked exponential, as can be seen in **Figure 5.6.10**. For a more representative sample, this test was repeated 20 times to 5,000,000 iterations, and the accumulated data was then statistically analyzed (see **Figure 5.6.11**). It was found that a Burr distribution (Type XII) fitted the data very closely, with the fit parameters given in **Table 5.6.5**.

#		26 th Jun 2009 Traffic	5,000,000 [‡] ITERATIONS OF C-CCGPSO-D	
Fit Parameters	Distribution	Lognormal Distribution	Burr Distribution (Type XII)	
	Log Likelihood	34986.8	19443.4	
	Mean	11.462%	6.4318%	
	Variance	1.411%	5.9896%	
	μ	-2.5308 (± 0.00514474)	α	0.0736585 (± 0.00565)
	σ	0.8540 (± 0.00363798)	k	1.0020100 (± 0.01401)
			c	2.1434300 (± 0.11268)
	50 th percentile [†]	8.68%	3.19%	
	25 th percentile [†]	4.91%	1.30%	
	10 th percentile [†]	2.57%	0.47%	
	5 th percentile [†]	1.61%	0.21%	
	Maximum Delay	239%	185%	
	Minimum Delay	0%	0%	
	Sample Size	27559	512x20	

[‡]Tests run for 20 times, [†] Systemic Inefficiency, Flights = 512, Knots = 16, Clusters = 16, Swarm = 64

Table 5.6.5: Comparison and fit of Delay Distributions

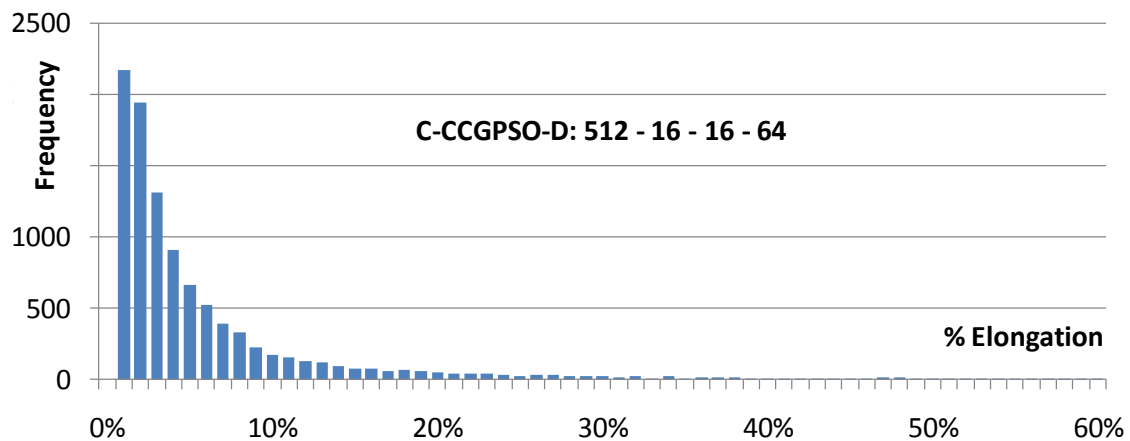


Figure 5.6.11: Delay distribution of flights after 5,000,000 iterations, (20 trials)

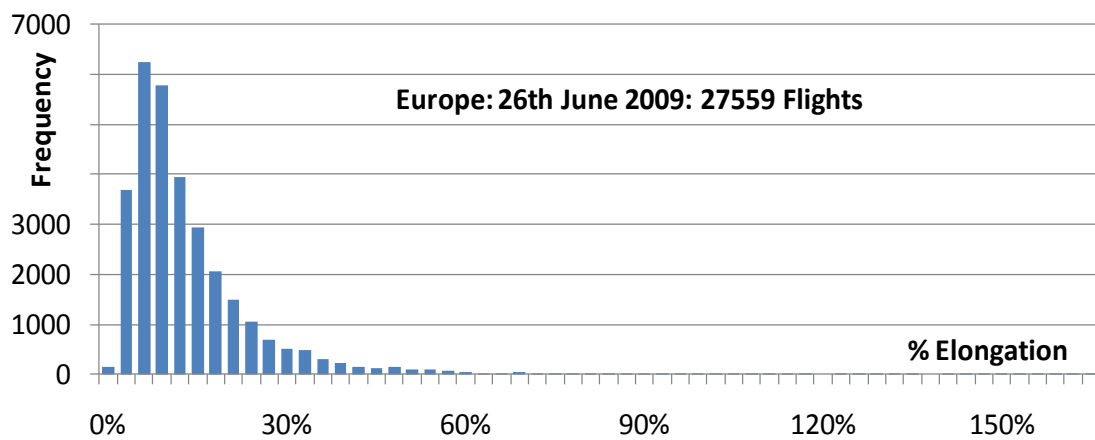


Figure 5.6.12: Delay distribution in the June 2009 flight sample analysed.

After 5 million iterations elapsed, the vast majority of flights (90%) were less than 0.5% delayed in relation to their optimum journey length. This, while fully complying with the separation minima of 5km, with no breaches at all, as given in the 64 particle entry of **Table 5.6.3**.

These results are best compared with actual statistics collected by Eurocontrol. In **Figure 5.6.12**, the distribution of flight delays shown, was extracted from the available ATM dataset, after removing all flights between airports closer than 250km, to eliminate most pleasure flights. The data fits almost precisely a lognormal distribution with the fit parameters shown in **Table 5.6.5**. The 50th, 25th, 10th and 5th percentiles are also given in both cases. In either case, a small proportion of flights were significantly delayed with respect to their shortest direct route. However a significant number of flights were able to fly direct routes with zero overhead.

Finally it is worth recalling Eurocontrol's own 2035 delay projection compared to 2012 data. Eurocontrol predict a sustained 5% growth for the foreseeable future, and this takes us to *triple* the current level of traffic density by 2035. This is given in **Figure 5.6.13**, where the delay scenario is expected to substantially worsen if conventional ATM technology prevails. The simulation results given in **Figure 5.6.11** and **Table 5.6.5**, represent this 2035 level of traffic density and the delay profile achievable using CATM.

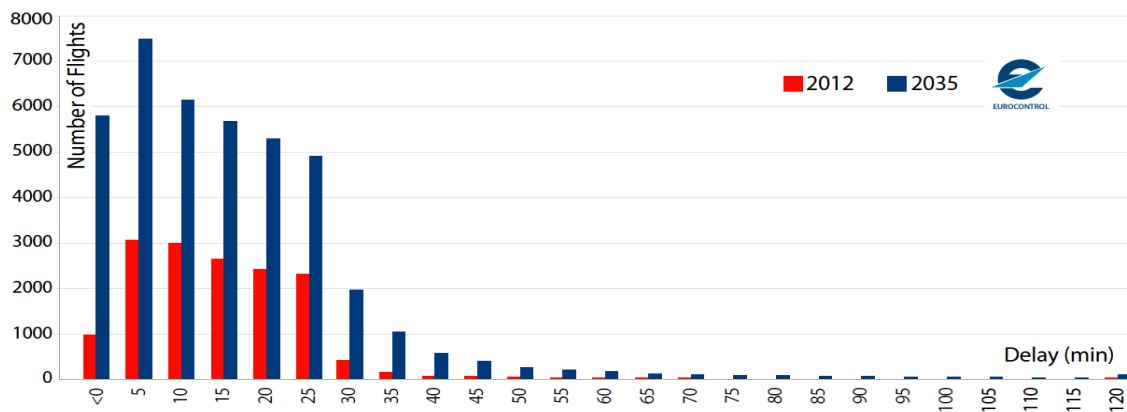


Figure 5.6.13: Eurocontrol's Current and Projected Delay Distribution [5.5]

5.7 REFERENCES

- [5.1] Matsumoto, M.; Nishimura, T., 1998, "Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator," *ACM Transactions on Modeling and Computer Simulation* vol. 8 no. 1 pp. 3–30.
 - [5.2] Marsaglia, G. 1994, Aug 1, "Yet another RNG", Posted to electronic bulletin board on sci.stat.math.
 - [5.3] Marsaglia, G. & Zaman, A., 1991., "A new class of random number generators," *Annals of Applied Probability* vol. 1 no. 3, pp. 462–480
 - [5.4] Marsaglia, G., 1968, "Random numbers fall mainly in the planes", *Proc. Natl. Acad. Sci.* vol. 61, no. 1, pp. 25-28
 - [5.5] Eurocontrol, 2013, Oct., "Challenges of Growth 2013 - Task 6: The Effect of Air Traffic Network Congestion in 2035," Report, European Organisation for the Safety of Air Navigation (Eurocontrol)
-

5.2 CONSTRUCTION OF A CATM SIMULATION RIG

The thousands of concurrent optimizer threads required some rather hefty computational hardware to accommodate. For this purpose, a dedicated 12 node computing blade server rig was constructed using dual processor motherboards from Supermicro® computer and Intel Xeon® Harpertown 8-core CPUs. The memory of the system was sized to accommodate several thousand threads per blade. **Figure 5.2.1** describes the setup.

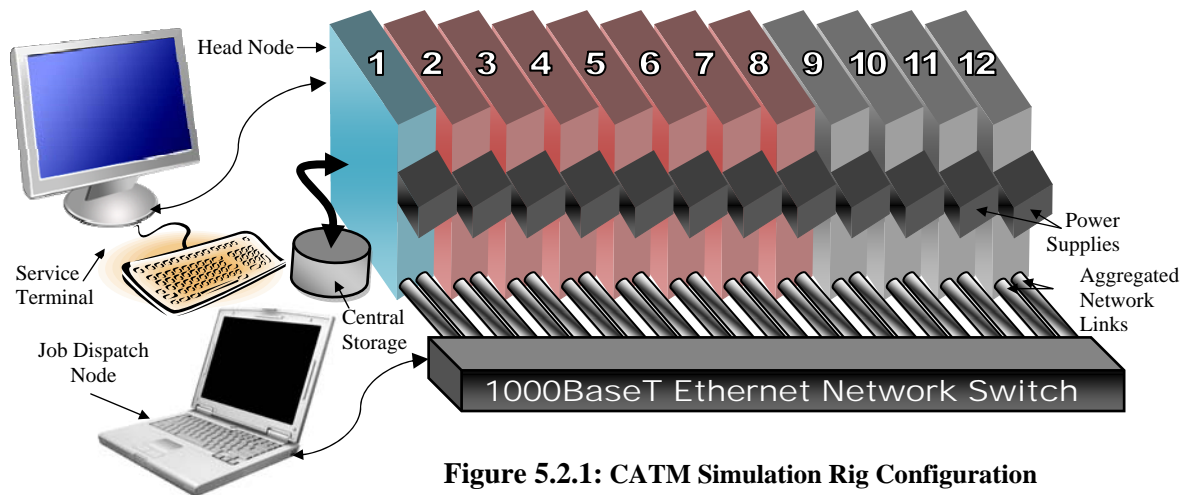


Figure 5.2.1: CATM Simulation Rig Configuration

Special consideration was taken to ensure adequate cooling of the resulting multi kW system. Water cooling and high capacity vehicle radiators were used to dispose of the waste heat from all 24 CPUs, and forced air cooling was used for the memory and the rest of the system. The processors were also over-clocked to around 130% to extract maximum performance, and their numerical reliability was exhaustively re-tested using Intel's Linpack V11.1.1 benchmarking tools. The aggregate performance of the system reaches 1.0 Tflop and is capable of traversing 80k PSO epochs per second for a 12,288 flight air traffic system. The system is divided into a 3.8GHz head node and 11 slaves. **Figure 5.2.2** shows a photograph of the assembled computing rig running some CATM tests, while **Table 5.2.1** shows a summary of the system's configuration and performance, together with an HP Pavilion DV6 i7 laptop PC, also listed here for comparison.

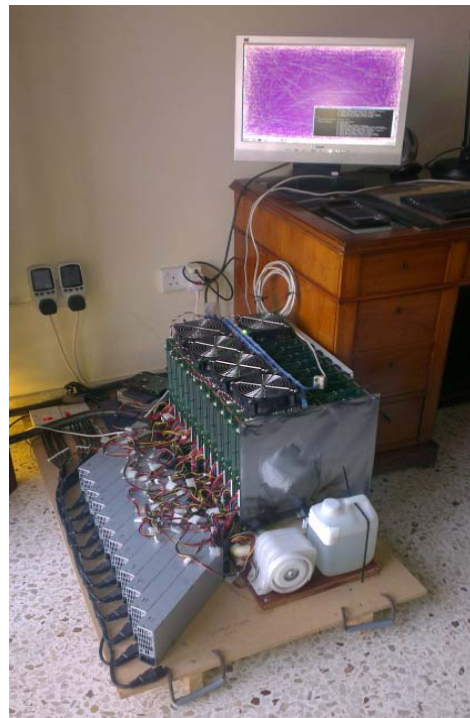


Figure 5.2.2: CATM Simulation Rig

Chapter 6

Conclusions and Future Work

This dissertation is concluded with a summary of the motivating objectives and the work undertaken. After listing the principal contributions, a number of recommendations for further work are made.

6.1 SUMMARY

“Global demand for motor vehicles will not exceed one million, due to the limited availability of professional chauffeurs,” said Gottlieb Daimler, inventor of the automobile in 1900 and we all know how true that turned out to be. The democratization of personal aviation, with very light jets (VLJ), coupled with commercial unmanned flight have the potential to repeat in aviation, what has happened to road transport.

World air transport has been on a steady exponential rise since the 1940’s and the trend has shown remarkable resilience to external shocks. The level of air traffic has greatly exceeded the wildest expectations of the air traffic management pioneers that originally defined the basic precepts of ATM that persist till today. This has stretched conventional ATM to a point where it is starting to show signs of ineffectiveness in the face of ever increasing congestion. Delays are on the rise, flights are being elongated unnecessarily, the system is becoming increasingly susceptible to disruption, and the high environmental impact of aviation is being compounded by the inability of air traffic controllers to optimise ATM operation in real-time. If these trends are not reversed, ATM could eventually face instability. The conservative, self-preserving outlook of the ATM community has confined progress to relatively minor tweaks of a tired human-centric paradigm. However, the diverging gap between ATM performance and fundamental requirements indicates the need for a step change.

In this work, the traditionally incremental approach to ATM research was broken to favour a more exploratory mindset. As a result, a new discipline called Computational Air Traffic Management has been defined to address the unique set of challenges presented by the ATM problem, by taking a more objective scientific approach.

A specific embodiment of a CATM system was designed, constructed, simulated and tested and shown to be a significant step towards demonstrating the feasibility of a fully autonomous multi-agent-based air transportation system based on optimisation principles. The system offers unique advantages in terms of resilience to disruption, efficiency and future scalability. The traffic density using such a system can be realistically increased many times higher than current levels while significantly improving on the current levels of safety, operating cost, environmental impact and flight delays.

6.2 CONCLUSIONS

Equipped with the latest developments in science, mathematics, computing and engineering, the ATM problem was framed in mathematical terms and tackled in a structured fashion in relation to the key performance indicators. The overarching recommended solution to the ATM problem is given as follows:

- 1) Computational Air Traffic Management is presented as a new engineering discipline focussed on holistically solving the global air traffic management problem in an efficient and autonomous, cost-effective fashion. It replaces today's fragmented partial automation with unified overarching autonomy by drawing on the latest results from the science of systems and complexity, optimal control, communications, and computational intelligence.
- 2) ATM was expressed as a large, decentralized, dynamic, variable size, infinite horizon, multi-parameter, constrained, nonlinear, non-causal, non-convex, multi-objective, high-dimensionality, hybrid (continuous-and-combinatorial), optimal control problem.
- 3) Grid Avionics was suggested as the preferred CATM embodiment, that relies on a self-organising distributed architecture that seeks to create a parallel-computing grid by internetworking all aircraft through using high speed links.
- 4) Swarm Intelligence, biomimetics, and other related metaheuristics are proposed as the underlying cooperative, but distributed, real-time optimisation algorithms that can be integrated into a receding horizon controller.
- 5) Multipath Free Space Optical Links between aircraft and static hubs are recommended as an appropriate medium to form a suitable CATM network that is both fast and resilient.
- 6) NURBS are proposed as an appropriate parameterisation for describing trajectories as well as universal language of interchange.
- 7) Hierarchical optimisation is suggested as the means of handling the non-convexity of ATM, by matching the intertwined local and global optimisation aspects of the problem to the complementary strengths of probabilistic and deterministic gradient driven optimisers.
- 8) Dynamic optimisation is suggested as the technique to make optimisers adaptive to the inevitable variability in the problem's constraints.
- 9) Continuous optimisation is used to capitalise on past optimisation effort to increase the system update rate, and reduce the effects of uncertainty.
- 10) Static clustering is recommended as a viable and stable method to mitigate problem growth when thousands of aircraft are involved.
- 11) Ideal attractors are recommended to accelerate optimisation, by directing probabilistic searches towards the best solutions, thereby conserving computational resources.

6.2.1 Key Contributions

There are a number of novel contributions of various magnitudes that are described in this dissertation. The most important of these are highly conceptual. However, in addition to the primary to the ATM domain (which is of a conceptual nature), a number of other secondary contributions are offered as supporting evidence in favour of the central concepts. The following specific contributions to the fields of distributed computational intelligence, optimisation science and ATM were made:

- 1) The ATM problem was given a fundamental rethink, starting from basic requirements. Continuity with the past was a subordinate objective and therefore no constraints were carried over including the highly onerous constraint of keeping the human decision maker in the loop. Full autonomy was thereby given serious thought, together with the implications that it brings.
- 2) Through a fully symmetric divide-and-conquer technique called clustering, an effective multi-trajectory optimisation technique based on Particle Swarm Optimisation, and Cooperative Coevolution was presented as a practical means of equitably distributing computational intelligence over all the aircraft in the ATM system such that a unique solution to the global ATM problem is found in real-time by capitalizing on the aggregate computational resources made available by all participating agents.
- 3) Several candidate automation algorithms from the wide spectrum of optimisation techniques were investigated and their unique strengths and weaknesses were compared. A number of novel problem simplifications and dimensionality mitigating enhancements were suggested and implemented to make the algorithms fast enough for practical applications.
- 4) A linear relationship between problem size and the computational resources required for a given level of performance was demonstrated empirically. This settles the debate on whether a CATM system would be scalable to deal with the global scale ATM problem. A CATM-based system is technically possible.
- 5) The convergence properties of the preferred algorithm were tested with respect to a number of parameters such as swarm size, cluster size and constriction configuration and parameters that strike a good compromise between exploration and exploitation were identified. The algorithms were shown to be unconditionally convergent irrespective of the parameters chosen (so long as fundamental constriction condition are observed).
- 6) Important preliminary insights about the communication requirements of these kind of algorithms were given to help select suitable communication links for CATM.
- 7) Developed a CATM simulator that can be further extended in a modular fashion using object oriented techniques in C++. An associated, extendable computing hardware platform was also devised to match the unique requirements of CATM multi-agent simulation.

6.3 FUTURE WORK

The following recommendations can be made for future investigations:

- 1) A Model Predictive version of the system needs to be simulated to show the system working in an ongoing fashion with new flights generated, executed and completed in real-time, while the optimiser tracks any disturbances to which the system is subjected.
 - 2) Realistic aircraft and atmospheric models and the associated inverse-dynamics constraints need to be integrated into the inner loop penalty function evaluation. This means that the BADA, ISA and other environmental impact models need to be cut down to the bare minimum computational footprint that still allows good fidelity. These models have to be rewritten in CUDA to allow GPGPU acceleration of the inner loop of the optimiser. The computing hardware platform needs to be upgraded accordingly.
 - 3) The European traffic models need to be integrated into simulator in order to generate directly comparable results with the measured performance figures of current ATM system. This is not difficult but requires some additional programming effort.
 - 4) The simulations then need to be scaled to even larger sizes, until a truly global self contained solution can be evaluated and compared to real world data. This needs to be enhanced with the WGS84 coordinate system to take into account geodetic effects.
 - 5) Various multi-swarm configurations of the algorithms need to be tested with respect to their ability to enhance the dynamic behaviour of the optimiser.
 - 6) A Pseudospectral local optimiser needs to be integrated with the global optimiser such that a more reliable hierarchical optimisation hybrid is created. This will require some thought on how to re-space the knots to match the Gauss-Lobatto or Gauss-Chebyshev collocation points.
-