

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

ProQuest Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

UMI[®]

TOWARD REAL-TIME EN ROUTE AIR TRAFFIC CONTROL OPTIMIZATION

**A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF AERONAUTICS AND ASTRONAUTICS
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY**

**Matthew R. Jardin
March 2003**

UMI Number: 3085194

Copyright 2003 by
Jardin, Matthew Robert

All rights reserved.

UMI[®]

UMI Microform 3085194

Copyright 2003 by ProQuest Information and Learning Company.
All rights reserved. This microform edition is protected against
unauthorized copying under Title 17, United States Code.

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

© Copyright by Matthew R. Jardin 2003

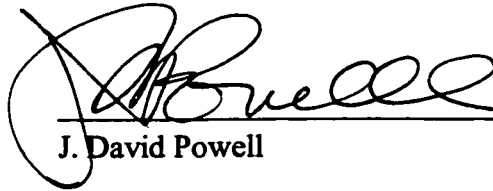
All Rights Reserved

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.



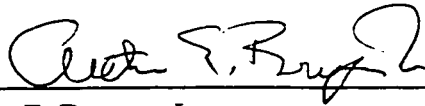
Stephen M. Rock, Principal Adviser

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.



J. David Powell

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.



Arthur E. Bryson, Jr.

Approved for the University Committee on Graduate Studies:

Abstract

The increase in air traffic along the existing jet route structure has led to inefficiencies and frequent congestion in en route airspace. Analysis of air-traffic data suggests that direct operating costs might be reduced by about 4.5%, or \$500 million per year, if aircraft were permitted to fly optimal wind-routes instead of the structured routes allowed today. To enable aircraft to safely fly along unstructured optimal routes, automation is required to aid air-traffic controllers. This requires the global solution for conflict-free optimal routes for many aircraft in real time. The constraint that all aircraft must maintain adequate separation from one another results in a greater-than-exponential increase in the complexity of the multi-aircraft optimization problem. The main challenges addressed in this dissertation are in the areas of optimal wind-routing, computationally efficient aircraft conflict detection, and efficient conflict resolution. A core contribution is the derivation of an analytical neighboring optimal control solution for the efficient computation of optimal wind-routes. The neighboring optimal control algorithm uses an order of magnitude less computational effort to achieve the same performance as existing algorithms, and is easily extended to compute near-optimal conflict-free trajectories. A conflict detection algorithm has been developed which eliminates the need to compute inter-aircraft distances. Simulation results are presented to demonstrate an integrated horizontal route-optimization and conflict-resolution method for air-traffic control. Conflict-free solutions have been computed for roughly double the current-day traffic density for a single flight level (over 600 aircraft) in less than 1 minute on a 450-MHz UNIX work station. This corresponds to a computation rate of better than 25 optimal routes per second. Extrapolation of the two-dimensional results to the multi-flight-level domain suggests that the complete solution for optimal conflict-free routes can be achieved in about 1 min. using currently available hardware.

Acknowledgments

The first class I took at Stanford (nonlinear control) was taught by Professor Steve Rock, who fortunately agreed to be my thesis adviser. His enthusiasm, insightful advice, and patience were essential to my completing this dissertation. Professor Rock worked patiently with me to distill this large problem into a manageable and focused research effort. I had a great deal to learn in this area, and I am grateful to Professor Rock for all of his help.

Professor Dave Powell stepped in as my co-adviser about halfway through my studies. He made sure that I considered my research from the perspective of a pilot. Professor Powell's insistence that I address the many-aircraft conflict resolution problem led to many of the key developments and contributions in this dissertation, and I am grateful to him for not letting me off the hook in this regard.

My dissertation reading committee was completed by the addition of Professor Arthur Bryson. What began for me as a reluctant foray into optimization became one of the most rewarding aspects of this dissertation. Professor Bryson was always available to meet with me to discuss and improve my dissertation work, right up until the final week before publication. I look forward to continuing our collaboration once this dissertation is completed.

I have earned this degree as a part-time student while working full-time at the NASA Ames Research Center. My collaboration with the research communities at both NASA and Stanford has been a rewarding experience, and I sincerely hope that NASA's support of employee degree programs will continue. I will try to provide NASA with a good return on their investment in me.

I was hired at NASA by Dr. Dallas Denery and Dr. Heinz Erzberger to work on air-traffic control concepts as a newly-graduated Master's degree student. From the time I began, I was encouraged by both of these gentlemen to continue my education.

Dr. Denery's understanding from having gone through the same education and career path that I would go through was very important to me and I am grateful for his mentorship and unwavering support over the years. Dr. Erzberger's enthusiasm for his work has been an inspiration for me. I found that whenever I had a new idea to try for my dissertation work, my first thought was to run it by Heinz to see what he'd think of it. The ensuing discussions and encouragement he gave were invaluable. I am fortunate to have been able to work so closely with Heinz over the years.

I would also like to thank Dr. Banavar Sridhar and Dr. Tom Edwards, my Branch and Division Chiefs at NASA, for being so supportive of my continuing Ph.D. studies. Fortunately for me, they understood the time pressures that often arose because of conflicts between my studies and my NASA duties, and they gave me enough latitude to see my Ph.D. through to its completion.

Some of my NASA colleagues (and friends), Dr. Gano Chatterji, Dr. Kapil Sheth, Dr. Shon Grabbe, and Michael Jastrzebski, endured many requests from me for results from the FACET software tool. They brought this upon themselves by being so knowledgeable and capable, and I greatly appreciate their help. Michael Jastrzebski was instrumental in generating some of the real-world examples in this dissertation by porting my optimization algorithms into the FACET software tool. I very much appreciated Michael's ability to take my terse and uncommented MATLAB code and to convert it into C and Java.

Finally, I'd like to express my gratitude for the support I've received from my family over the years. Whenever I was in need of encouragement, they were always there for me, cheering me on. I cannot thank my family enough for being there for me.

For encouraging my interest in science and engineering (remember the trips to Kittyhawk and to Cape Kennedy to see the Shuttle Columbia on the launchpad?), for supporting my educational pursuits, and for everything else too, I dedicate this dissertation to my father, Robert Jardin.

Contents

| | |
|---|-------------|
| Abstract..... | iv |
| Acknowledgments | v |
| Contents | vii |
| List of Tables | xi |
| List of Figures..... | xiii |
| Nomenclature | xvii |
| Chapter 1 | |
| Introduction..... | 1 |
| 1.1 Motivation..... | 2 |
| 1.2 Challenges..... | 2 |
| 1.3 Contributions | 5 |
| 1.4 Dissertation Outline | 7 |
| Chapter 2 | |
| Background | 9 |
| 2.1 Introduction..... | 9 |
| 2.2 Current En Route Air-Traffic Control | 10 |
| 2.3 Mathematical Problem Statement | 14 |

| | | |
|----------------------|--|------------|
| 2.4 | Prior Art: 4-Dimensional System Concepts..... | 18 |
| 2.5 | Analysis and Simplifying Assumptions..... | 25 |
| 2.6 | Sequential Trajectory Optimization..... | 33 |
| 2.7 | A Practical Bound on the Optimum Solution | 35 |
| 2.8 | Summary | 36 |
| Chapter 3 | | |
| | System Concept..... | 39 |
| 3.1 | Introduction..... | 39 |
| 3.2 | Sequential Optimization Algorithm..... | 40 |
| 3.3 | Computational Requirements of the Sequential Optimization Algorithm..... | 50 |
| 3.4 | Summary | 61 |
| Chapter 4 | | |
| | Optimal Wind Routing..... | 63 |
| 4.1 | Introduction..... | 63 |
| 4.2 | Motivation: The Potential Benefits of Optimal Wind-Routing | 66 |
| 4.3 | Prior Art: Optimal Aircraft Trajectories in Winds..... | 69 |
| 4.4 | Neighboring Optimal Wind Routing | 81 |
| 4.5 | Application of Neighboring Optimal Wind Routing | 98 |
| 4.6 | NOWR Examples | 103 |
| 4.7 | NOWR Performance Analysis..... | 110 |
| 4.8 | Benefits and Enhancements of NOWR Over Prior Art | 124 |
| 4.9 | A Few Words on Optimal Routes..... | 126 |
| 4.10 | Summary | 129 |
| Chapter 5 | | |
| | Strategic Conflict Detection and Resolution | 131 |
| 5.1 | Introduction..... | 131 |
| 5.2 | Conflict Detection Background | 136 |
| 5.3 | Conflict Grid Method..... | 140 |
| 5.4 | Conflict-Resolution Background | 151 |
| 5.5 | Neighboring Optimal Wind-Routing Perturbation Conflict Resolution..... | 155 |

| | | |
|--------------------|---|------------|
| 5.6 | Summary | 158 |
| Chapter 6 | | |
| | Simulation and Performance Evaluation | 161 |
| 6.1 | Introduction..... | 161 |
| 6.2 | Simulation Design..... | 161 |
| 6.3 | Analysis and Parametric Studies..... | 169 |
| 6.4 | Summary | 186 |
| Chapter 7 | | |
| | Conclusion | 187 |
| 7.1 | Contributions | 188 |
| 7.2 | Recommendations for Future Work | 191 |
| 7.3 | Concluding Remarks..... | 193 |
| Appendix 1: | | |
| | Solutions for the Neighboring Optimal Wind Routing Gains | 195 |
| A1.1 | Analytical Solution for NOC Feedback Gains..... | 195 |
| A1.2 | Numerical Solution for NOC Feedback Gains | 210 |
| Appendix 2: | | |
| | The Rapid Update Cycle | 221 |
| | References..... | 223 |

List of Tables

| | |
|---|-----|
| Table 2.1. Conflict statistics from binomial random variable model. | 31 |
| Table 4.1. RUC data files used in optimization performance analysis. | 68 |
| Table 4.2. NOC feedback gains for 3 grid points. | 104 |
| Table 4.3. Combinations of DP grid resolution. | 118 |
| Table 6.1. Simulation parameters for conflict perturbation analysis. | 171 |
| Table 6.2. Simulation parameters for airspace capacity analysis. | 175 |
| Table 6.3. Empirically-determined conflict iteration parameters. | 183 |

List of Figures

| | | |
|-------------|---|----|
| Figure 1.1. | Two possible solution sets (crossing orders) for two aircraft..... | 3 |
| Figure 1.2. | Eight possible solution sets (crossing orders) for three aircraft..... | 4 |
| Figure 2.1. | Airspace classifications in the United States. | 10 |
| Figure 2.2. | Maximum aircraft count per flight level over the U.S. on 18 Mar. 2000. ... | 12 |
| Figure 2.3. | An ATC sector map of the United States. | 13 |
| Figure 2.4. | High level concept diagram of 4-D time-based air-traffic control. | 19 |
| Figure 2.5. | Block diagrams of 4-D ATC and the current ATC system..... | 20 |
| Figure 2.6. | 4-D trajectory control and 4-D STA control..... | 22 |
| Figure 2.7. | Optimal cruise-climb and step-climb profiles. | 26 |
| Figure 2.8. | Conflict counts for structured and great-circle routes. | 30 |
| Figure 3.1. | Sequential route optimization algorithm flowchart. | 41 |
| Figure 3.2. | Resolving conflicts earlier in a deterministic system is more efficient. | 43 |
| Figure 3.3. | Plot of additional fuel use vs. regulated position error. | 49 |
| Figure 3.4. | Conflict probability is hypothesized to be memoryless..... | 52 |
| Figure 3.5. | Expected No. of conflict iterations and approximation errors..... | 57 |
| Figure 3.6. | Two illustrations of airspace capacity limits. | 58 |
| Figure 4.1. | A representative set of long-range routes. | 67 |
| Figure 4.2. | Zermelo's problem: minimum-time paths through varying winds..... | 70 |
| Figure 4.3. | Zermelo's problem with linear wind shear in the cross-track direction. | 71 |
| Figure 4.4. | Discrete Dynamic Programming. | 73 |
| Figure 4.5. | The NOC perturbation feedback algorithm. | 82 |
| Figure 4.6. | Coordinate rotation from SLL to RLL coordinates. | 84 |

| | |
|--|-----|
| Figure 4.7. Transformation from RLL to NLL coordinates. | 88 |
| Figure 4.8. A graphical depiction of a piecewise-linear function of | 92 |
| Figure 4.9. Graphical depiction of the neighboring optimal wind routing problem. | 94 |
| Figure 4.10. Normalized neighboring optimal feedback gains. | 97 |
| Figure 4.11. Block diagram of the NOWR algorithm. | 99 |
| Figure 4.12. NOC feedback gains for the case of three grid points. | 105 |
| Figure 4.13. Illustrative example-1 wind parameters. | 106 |
| Figure 4.14. Illustrative example-1 trajectory plot and plot of optimal heading vs. x... | 107 |
| Figure 4.15. Illustrative example-2 wind parameters. | 108 |
| Figure 4.16. Illustrative example-2 trajectory plot and plot of optimal heading vs. x... | 109 |
| Figure 4.17. Illustrative example-3 wind parameters. | 110 |
| Figure 4.18. Illustrative example-3 trajectory plot and plot of optimal heading vs. x... | 111 |
| Figure 4.19. FACET example-1: NOWR route from JFK to SFO (2/14/2001). | 112 |
| Figure 4.20. FACET example-2: NOWR route from Miami to Seattle (2/14/2001). | 113 |
| Figure 4.21. Illustration of the Dynamic Programming grid. | 114 |
| Figure 4.22. A Dynamic Programming solution. | 115 |
| Figure 4.23. Performance analysis of NOWR and DP. | 120 |
| Figure 4.24. An example of highly nonlinear wind conditions. | 122 |
| Figure 4.25. Illustration of time variations in optimal wind routes. | 129 |
| Figure 5.1. A simple conflict-resolution model. | 133 |
| Figure 5.2. The Conflict Grid method of conflict detection. | 141 |
| Figure 5.3. Flowchart of the sequential Conflict Grid algorithm. | 144 |
| Figure 5.4. Illustration of a missed conflict at a grid cell boundary. | 145 |
| Figure 5.5. Illustration of a missed conflict resulting from time discretization. | 145 |
| Figure 5.6. Grid-cell buffering to eliminate missed conflicts. | 146 |
| Figure 5.7. Example of low predicted conflict probability for two aircraft. | 149 |
| Figure 5.8. Example of higher conflict probability with multiple aircraft. | 150 |
| Figure 5.9. Conflict probabilities between storm cells and aircraft. | 151 |
| Figure 5.10. Introducing pseudo-shears for conflict resolution. | 157 |
| Figure 6.1. Histogram of the number of aircraft scheduled to depart for FL350. | 163 |
| Figure 6.2. Generating random routes based on empirical data. | 165 |

| | |
|---|-----|
| Figure 6.3. High-level simulation flowchart. | 168 |
| Figure 6.4. Optimization performance vs. perturbation parameter. | 170 |
| Figure 6.5. Number of onerous conflicts vs. conflict-resolution parameter. | 172 |
| Figure 6.6. Flight-level capacity vs. conflict-resolution parameter. | 173 |
| Figure 6.7. Varying average airspace sector density. | 174 |
| Figure 6.8. Curve-fitting empirical conflict-iteration data. | 176 |
| Figure 6.9. Curve fit example of conflict-iteration data. | 177 |
| Figure 6.10. Airspace capacity (per flight level) vs. average conflict-grid-cell size. | 178 |
| Figure 6.11. Extrapolated 3-D capacity for 17 flight levels from FL180 to FL390. | 179 |
| Figure 6.12. Estimating required recomputation rate from grid-cell transit time. | 181 |
| Figure 6.13. Required computation rate vs. conflict grid spacing. | 184 |
| Figure 6.14. 2-D results extrapolated to the 3-D case. | 185 |
| Figure A1.1. Nominal optimal trajectory in zero winds. | 196 |
| Figure A1.2. Graphical depiction of a piecewise linear function of x | 198 |
| Figure A1.3. Piecewise linear perturbations in the wind parameters at point i | 200 |
| Figure A1.4. Normalized neighboring optimal feedback gains. | 209 |
| Figure A1.5. Nominal optimal solution in normalized coordinates. | 216 |

Nomenclature

Roman Symbols

| | |
|---------------------|---|
| AC_i | aircraft i |
| C_0 | GRV model parameter: predicted airspace capacity (2-D) |
| \bar{C}_0 | GRV model parameter: predicted airspace capacity (3-D) |
| C_1 | GRV model parameter (2-D) |
| \bar{C}_1 | GRV model parameter (3-D) |
| C_f | fuel cost |
| CI | cost index |
| $E[\]$ | expectation operator |
| f_s | rate of change of fuel weight with path distance |
| H | Hamiltonian |
| H_x | partial derivative of Hamiltonian with respect to x |
| i_w | nearest wind-model grid point to a detected conflict |
| J_1 | single-aircraft cost function |
| J_{ATM} | multi-aircraft ATM cost function |
| J_{DOC} | direct operating cost |
| J_{NOWR}^* | minimum total flight time of all NOWR solutions, ignoring conflicts |
| J_T^* | cost of minimum-time solution |

| | |
|-------------------|--|
| K_{NOWR} | time to compute a neighboring optimal wind route |
| L | flight range |
| L_1 | single-aircraft Lagrangian |
| N | number of aircraft |
| N_c | number of conflicts |
| N_{cr} | number of conflict resolution maneuvers |
| N_{FL} | number of flight levels |
| N_{ss} | number of solution sets |
| N_w | number of wind-model grid points |
| p | probability |
| P | state error covariance matrix |
| P_i | probability of a conflict-free trajectory for aircraft i (geometric random variable model) |
| \bar{P}_i | probability that at least one conflict constraint will be active |
| p_{ik} | probability of resolving a conflict in k iterations for aircraft i (Bernoulli trial) |
| R | maximum arrival rate per runway; radius of Earth |
| R_i | number of conflict resolution iterations for aircraft i |
| t | time |
| T | normalized time |
| t_0 | initial time |
| t_f | final time |
| u | normalized along-track wind vector component |
| u_1 | single-aircraft control vector |
| u_b | bias along-track wind |
| u_{ps} | pseudo wind shear |

| | |
|----------|---|
| u_w | east wind vector component |
| u_y | partial derivative of u with respect to y |
| u_{yi} | u_y at wind-model grid point i |
| V | airspeed |
| v | normalized cross-track wind vector component |
| v_b | bias cross-track wind |
| v_{bi} | bias cross-track wind at wind-model grid point i |
| v_w | north wind vector component |
| V_{ws} | wind shear parameter |
| v_y | partial derivative of v with respect to y |
| x | east position; normalized along-track position |
| x_1 | single-aircraft state vector |
| y | north position; normalized cross-track position |
| Y_{2D} | expected number of conflict-resolution iterations for 2-D problem |
| Y_{3D} | expected number of conflict-resolution iterations for 3-D problem |
| y_s | distance for computing approximate wind shear |

Greek Symbols

| | |
|-------------------|--|
| Δx_{DP} | dynamic programming along-track grid spacing |
| Δx_{grid} | conflict grid spacing |
| Δx_{ij} | separation distance between aircraft i and j |
| H | Hamiltonian |
| η_{ATC} | air traffic control optimization efficiency |
| η_c | ratio of trajectory communication time to trajectory optimization time |
| H_x | partial derivative of Hamiltonian with respect to x |
| λ | latitude |
| μ_{CG} | conflict grid memory |
| μ_s | maximum perturbation metric |
| ρ_{NOWR} | number of optimal routes computed per unit time |
| τ | longitude |
| θ | heading angle |
| θ_{DP} | dynamic programming angular grid resolution |
| θ_{nom} | nominal heading angle |
| u_w | east wind vector component |
| v_w | north wind vector component |
| V_{ws} | wind shear parameter |
| ξ_{cd} | number of computations for one conflict detection operation |
| ξ_i | number of computations for aircraft i |
| ξ_{wo} | number of computations for computing a wind-optimal route |

Acronyms

| | |
|-----------------|---|
| AAL | Active Aircraft List |
| ADS-B | Automatic Dependent Surveillance Mode-B |
| AIRINC | Aeronautical Radio, Inc. |
| ARTCC | Air Route Traffic Control Center |
| ATC | air-traffic control |
| ATM | air-traffic management |
| CD&R | conflict detection and resolution |
| CFD | computational fluid dynamics |
| CG | Conflict Grid |
| CTAS | Center-TRACON Automation System |
| DCM | direction cosine matrix |
| DOC | direct operating cost |
| DP | Dynamic Programming |
| ETMS | Enhanced Traffic Management System |
| FACET | Future ATM Concepts Evaluation Tool |
| FL | flight level |
| FSFS | first-scheduled-first-served |
| GA | Genetic Algorithm |
| GARTEUR | Group for ATM R&D Technology in Europe |
| GRIB | Gridded Binary |
| GRV | geometric random variable |
| IFR | Instrument Flight Rules |
| LQ | Linear Quadratic |
| MAPS | Mesoscale Analysis and Prediction System |
| NAS | National Airspace System |
| NLL | normalized longitude and latitude |
| NOC | Neighboring Optimal Control |
| NOWR | Neighboring Optimal Wind Routing |
| NP | Nondeterministic Polynomial-time |
| PHARE | Programme for Harmonised ATM Research |
| RLL | rotated longitude and latitude |

| | |
|---------------|---|
| RUC | Rapid Update Cycle |
| RVSM | Reduced Vertical Separation Minimum |
| SCG | Stochastic Conflict Grid |
| SLL | standard longitude and latitude |
| STA | scheduled time of arrival |
| SUA | special-use airspace |
| TRACON | Terminal Radar Approach Control |
| WAAS | Wide Area Augmentation System |
| WARP | Worldwide Aeronautical Route Planner |

Chapter 1

Introduction

The objective of this dissertation is to work toward a real-time method to optimize and deconflict en route trajectories of all aircraft in flight on a continental scale. Under certain practical assumptions, optimization of en route aircraft trajectories is primarily a problem of finding minimum-time horizontal routes through a varying wind field. A practical real-time optimization method would enable the development of a system to perform strategic optimization of air traffic within the National Airspace System (NAS) of the United States, the largest and busiest air transportation system in the world. This has not been possible using existing approaches to air-traffic control optimization.

Much of the en route air-traffic control research of the past decade has focused on allowing aircraft to fly along less constrained routes than is possible today by relying on distributed control to guarantee safe conflict resolution. Free Flight is the general term that has been applied to these concepts. But the ultimate goal of Free Flight is to remove route constraints so that the routes traveled are close to optimum (minimum fuel, minimum time, minimum deviation from schedule, etc.), not to effect distributed control as an end in itself. The theoretical optimum solution for all aircraft is generally considered to be impossible to compute using current state-of-the-art optimization algorithms and computer hardware. The development of a feasible approach to providing near-optimal conflict-free air-traffic control solutions in real-time would constitute an important contribution to the field of air-traffic control.

1.1 Motivation

By creating a system that can safely and efficiently optimize aircraft routes in real time, it becomes possible to move toward a system free of the many route constraints imposed in the current structured routing system. An analysis of actual traffic data has shown that a reduction of at least 4.5% in total flight time (or fuel use) is possible when aircraft are permitted to fly unconstrained optimal routes. In real terms, this is a reduction of about 500 hr per day in flight time, which translates to nearly \$1 million per day (\$360 million annually). In addition to direct cost savings, the air-traffic density would be reduced for a given number of aircraft by spreading them over a larger airspace than is possible in a structured routing system. This should improve safety by lessening the need for tactical collision avoidance.

1.2 Challenges

The greatest challenge to real-time air-traffic control optimization is system complexity. The airspace air transportation systems in the world. At any given time, there may be as many as 3,000 aircraft operating in Class A airspace over the continental United States, making it among the world's busiest [1]. Just computing an optimal route for a single aircraft is a difficult problem because of the varying wind field, but the great increase in complexity begins when multiple aircraft and trajectory conflicts are considered.

The solution complexity becomes apparent when considering all the possible solution sets, or orderings, of all aircraft. In this context, a solution set refers to all solutions to one side or the other of a conflict-resolution maneuver in a horizontal plane. For N aircraft traveling in a horizontal plane, the number of possible discrete solution sets of the aircraft is given in general by

$$N_{ss} = 2^{((N(N-1))/2)} \quad (1.1)$$

The ordering of aircraft in the horizontal plane may be achieved by changes in speed or heading, or by a combination of the two. For example, in the case of two aircraft at a

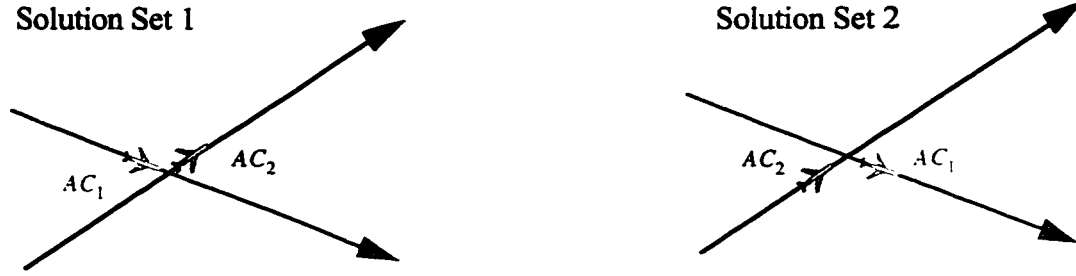


Figure 1.1. Two possible solution sets (crossing orders) for two aircraft.

constant altitude there are two possible solution sets (fig. 1.1). In the first solution set, aircraft AC_2 crosses ahead of aircraft AC_1 , with the order being reversed in the second set of solutions. For three aircraft, the number of possible solution sets increases to eight (fig. 1.2). When the current-day maximum number of aircraft at any given flight level is considered (about 500 aircraft at FL330 or FL350), the number of possible solution sets explodes to more than $10^{37,500}$. By examining equation (1.1), one sees that adding just one additional aircraft leads to 2^N times the number of possible solution sets. This greater-than-exponential increase in computational complexity is typical of optimization problems involving many independent vehicles or agents [2-5].

The optimum result within each solution set must still be computed and compared with each other set to determine the global optimum solution. For the air-traffic control problem, these individual optimization problems are essentially wind optimal routing calculations. Computing wind optimal routes for a single aircraft is nontrivial because the problem is both nonlinear and nonconvex. Efficient grid-search techniques such as discrete dynamic programming (DP), and randomized search techniques such as Genetic Algorithms (GA) have been applied to optimal wind routing, but they are still too computationally intensive to solve the many-aircraft optimal conflict free routing problem in real time [6-9].

Another subtle complexity of the air-traffic control optimization problem is that there is no clear beginning or end. With the rare exception of the terrorist attacks of 11

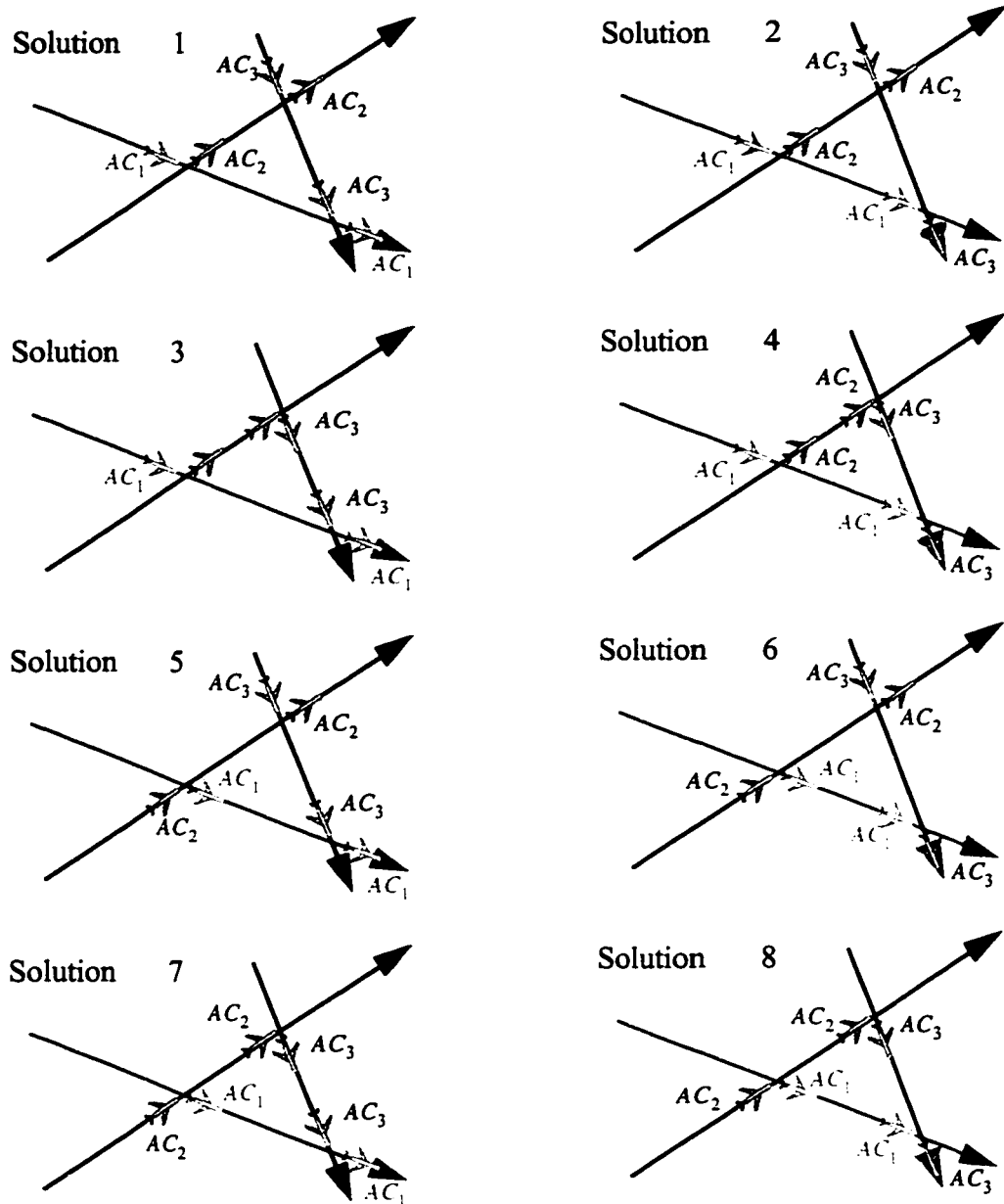


Figure 1.2. Eight possible solution sets (crossing orders) for three aircraft.

September 2001, there is always a significant number of active aircraft in the United States airspace. This is not a difficulty in practice, as evidenced by the fact that the system performs adequately today, but when one tries to compute an optimal solution instead of just a feasible solution, the continuous nature of air traffic becomes vexing. The difficulty

arises because inter-aircraft separation constraints couple the optimal solutions of each aircraft to the controls of each other aircraft for all time.

An additional factor that contributes to the complexity is that significant unmodeled disturbances exist, such as misforecast atmospheric conditions and aircraft flight technical errors. These disturbances push the system well beyond linear perturbation limits and ultimately require that optimal solutions be recomputed in real-time.

1.3 Contributions

The main contribution of this dissertation is the design of a system that computes a complete set of conflict-free wind-optimal routes for double the current-day air-traffic density at the busiest flight level in less than 1 minute on a 450-MHz UNIX work station. This is orders of magnitude beyond the capabilities of any currently available algorithms, and is a major contribution to the field of air-traffic control. To achieve this real-time, conflict free optimal-route planning capability, several contributions are claimed in this dissertation.

1.3.1 Neighboring Optimal Wind Routing

The first contribution is the development of a computationally efficient wind-optimal routing algorithm based on neighboring optimal control techniques. The resulting algorithm, called Neighboring Optimal Wind Routing (NOWR), computes near-optimal routes in an amount of time that is proportional to the length of the route (this is referred to as “order n ,” or $O(n)$) with a relatively small proportionality constant. The best prior techniques have been shown to be $O(n \log n)$, with very large proportionality constants. Aside from the improvement in computational efficiency, the NOWR algorithm has been shown to be well suited to conflict resolution in a sequential optimization system.

1.3.2 Efficient Strategic Conflict Detection and Resolution

The next contributions are in the areas of aircraft conflict detection and resolution. The problem of conflict detection is to predict whether the distance between any pair of aircraft will be less than the allowable legal separation (generally 5 n.mi. for commercial

aircraft), and the best existing conflict detection algorithms have been shown to be $O(n \log n)$ [10]. A new approach to conflict-detection that is coupled with the conflict-resolution process is developed. The technique is called the Conflict Grid (CG) and is shown to require a negligible amount of computation relative to that required to compute an optimal trajectory (less than 0.1%). An enhancement to the NOWR algorithm is made to enable the efficient computation of perturbed wind optimal routes. A very efficient conflict detection and resolution strategy is developed based on the perturbation NOWR algorithm for which the expected number of computations is proportional to $\log(C_0/(C_0 - n))$, where n is the number of aircraft in the system, and C_0 is a constant parameter used to fit the computational model to empirical data. This is the first technique to resolve conflicts while maintaining a sense of wind optimality for each aircraft route. All previous approaches have either neglected winds entirely, or have assumed that some nominal ground path was the desired objective so that conflict resolutions were computed to minimize deviations from the nominal path [11].

Although there can be no claim of global optimality for the resulting solutions using the techniques presented in this dissertation (an NP-hard problem cannot be solved in polynomial time), the solutions are compared to a theoretical bound on the optimum solution to give real-time measures of the optimization performance.

1.3.3 Airspace Capacity Modeling

Additional contributions have been made in system analysis using the techniques developed in this dissertation. The first is the derivation of a practical airspace-capacity model. Until now, there has never been a practical method for accurately estimating airspace capacity for a given air-traffic-control strategy. The technique is derived by hypothesizing a probabilistic model of the expected number of conflicts as a function of the number of aircraft in the airspace.

Using the probabilistic airspace-capacity model, simulation studies have been conducted to determine the relationship between legal aircraft separation standards and airspace capacity. This relationship may be used to estimate the benefits to be gained by a

reduction in minimum separation standards that will be possible with the introduction of new surveillance technologies like the Wide Area Augmentation System (WAAS) and Automatic Dependent Surveillance Mode-B (ADS-B). Until now, there has been no practical or meaningful way to translate navigational accuracy improvements into air-traffic capacity and efficiency gains.

By reasoning that the ultimate requirement for real-time recomputation of optimal trajectories for all aircraft is driven by consideration of anomalies such as in-flight emergencies, a computer performance requirement has been derived. The resulting relationship gives the required computation rate as a function of the minimum separation distance between aircraft.

1.4 Dissertation Outline

1.4.1 Background

Background on prior art in 4-D air-traffic control system concepts dating back to the early 1970s is presented in chapter 2. The background discussion is followed by a statement of the air-traffic-control optimization problem and a discussion of the simplifying assumptions that are made to arrive at a practical method to achieve near-optimum performance in real time.

1.4.2 System Concept

The system architecture is presented in chapter 3, followed by an analysis of the computational requirements and performance capabilities of the chosen system. The system concept is based on the strategic optimization of all aircraft trajectories. Aircraft then would follow the optimal trajectory clearances using closed-loop 4-D guidance and control. The optimal solution is approximated by solving for the optimal conflict-free trajectories of each aircraft sequentially, while holding previously computed trajectories fixed. This approximation reduces problem complexity to that which may be solved in polynomial time.

1.4.3 Optimal Wind Routing

Prior art in the computation of optimal wind routes is presented in chapter 4. This is followed by the development of an algorithm called Neighboring Optimal Wind Routing (NOWR), which is shown to be an efficient method for computing optimal wind routes. The chapter concludes with a discussion of the benefits of NOWR over those of existing approaches, and of recommendations for future to further improve the performance of NOWR.

1.4.4 Strategic Conflict Detection and Resolution

The prior art in solving the related problems of strategic conflict detection and resolution is discussed in chapter 5, followed by the development of more efficient solution approaches that retain optimality in winds. For conflict detection, an algorithm is developed that is essentially free, from a computational standpoint. For conflict-resolution, the NOWR algorithm is enhanced so that it may be used in an iterative manner for conflict-resolution.

1.4.5 Simulation and Performance Evaluation

The algorithms developed in this dissertation are integrated into a complete simulation system in chapter 6. The design of the simulator is presented, including a pseudo-code listing of the simulator source code. The results of a set of parametric studies are then presented to demonstrate how the optimization algorithms might be used in practice, and to demonstrate the kinds of analyses that may be conducted with the simulator. The chapter concludes with a computational timing analysis of the neighboring optimal wind routing functions to show that real-time optimization has been achieved.

1.4.6 Conclusion

The final chapter presents a summary of the main conclusions and contributions of the dissertation. Recommendations for future research are also presented.

Chapter 2

Background

2.1 Introduction

The goal of en route air-traffic-control (ATC) optimization is to find the most efficient trajectories for all aircraft operating in the national airspace system (NAS) while maintaining a safe separation distance between all aircraft. This is a complex dynamic optimization problem with inequality constraints on the separation between aircraft. The optimization criterion is to minimize the direct operating cost (DOC) for each aircraft in the system, where DOC is a weighted combination of fuel and time costs.

The large number of constraints has led to the classification of this problem as a nondeterministic polynomial-time hard (NP-hard) or NP-complete problem. This means that the problem is characterized by an inherent complexity that causes the solution effort to increase at a rate greater than can be expressed by a polynomial function. Two excellent sources for non-mathematicians on computational complexity are found in references 2 and 3, and a more rigorous description may be found in reference 4. The NP-hard nature has been proven for some abstract forms of the aircraft conflict-resolution problem and will be assumed true for the purposes of this dissertation [5]. Because the complexity of this problem precludes its rigorous solution, approximate solutions are required.

There are many approximate solution approaches to the ATC optimization problem. To explore each of them is beyond the scope of this work. Instead, the goal is to use practical system analysis to help guide the choices and approximations made to obtain

near-optimal solutions in real-time. To find a feasible solution, the problem must be transformed into one that can be solved in polynomial time. The challenge is to make approximations that do not significantly degrade the optimality of the solution.

In this chapter, the en route ATC optimization problem is posed as a dynamic optimization problem so that the complexity of the solution may be discussed. The complexity will be shown to originate from the aircraft separation constraints. Prior approaches to solving this problem are then discussed. The remainder of this chapter introduces simplifying approximations that enable solutions in polynomial time without greatly sacrificing optimization performance. A brief description of the current en route air-traffic control system is now presented to provide perspective for the work in this dissertation.

2.2 Current En Route Air-Traffic Control

Airspace over the continental United States is classified as either Class A, B, C, D, E, or G according to the type of air-traffic services provided (fig. 2.1). Class A airspace

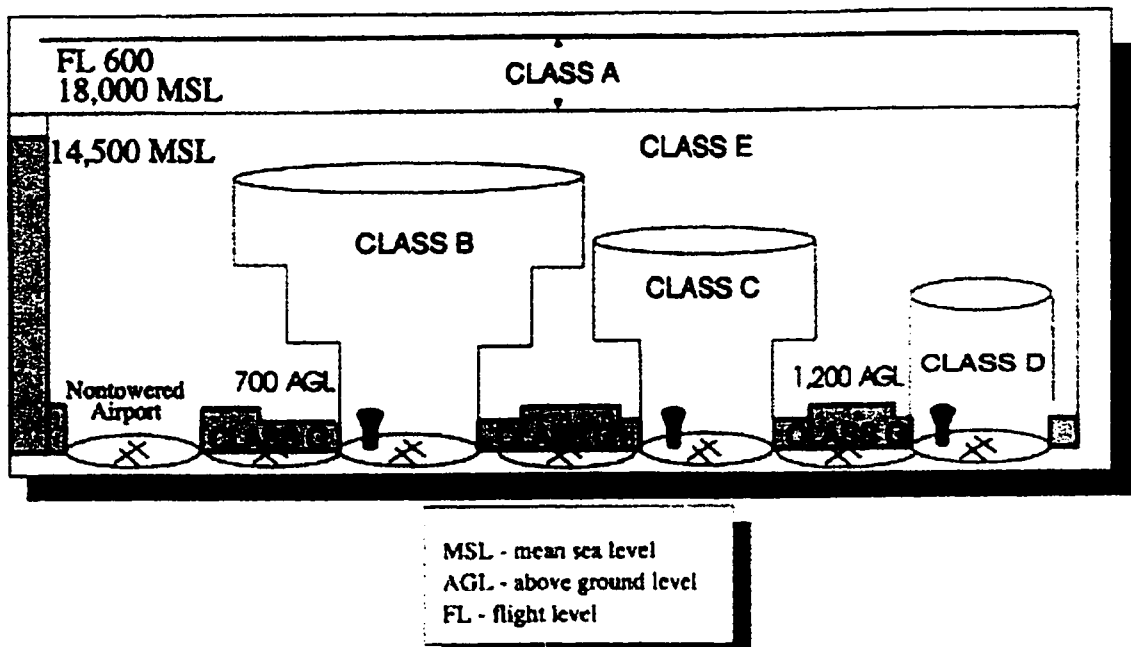


Figure 2.1. Airspace classifications in the United States.

extends from 18,000 ft above mean sea level (MSL) up to 60,000 ft, and is the domain of interest for en route air-traffic control. Altitudes in Class A airspace are discretized by flight level, where a flight level (FL) is defined as a surface of constant atmospheric pressure which is related to a specific standard pressure datum (standard pressure: 29.92 in. Hg or 1,013 hPa) and is separated from other such surfaces by specific pressure intervals which are stated in three digits that represent hundreds of feet. For example, FL600 is 60,000 feet MSL. The distinction is made between a flight level and a geopotential altitude in air-traffic control so that aircraft operating on barometric altimeters will all have common altitude references in any local region. In this way, aircraft can maintain unambiguous altitude references while following isobars. Note that standard pressure altitude may differ from true geopotential altitude by a few thousand feet as atmospheric conditions change.

Within Class A airspace from FL180 up to FL390 there are 17 distinct flight levels. From FL180 through FL290, the flight levels are separated by 1,000 ft and alternate westbound and eastbound traffic, with FL180 being a westbound level (headings from 000° through 179°). Above FL290, the flight levels are separated by 2,000 ft, again with alternating westbound and eastbound traffic. FL290 is an eastbound flight level. With the desire to move toward free flight, changes to the flight-level rules are planned. Under a program called Reduced Vertical Separation Minimum (RVSM), the separation between flight levels over the North Atlantic has already been reduced to 1000 foot increments. The same change is due to be phased in over the continental United States in January 2005. More fundamental changes in the flight-level definitions have also been considered, such as using a finer quantization of flight levels depending on a more detailed quantization of the direction of each aircraft [12].

A recent study of air-traffic conflicts for free and structured routing reported that of the approximately 57,000 aircraft flying during a typical 24-hr period in the U.S., about 38,000 operated in Class A airspace [1]. The peak number of aircraft operating at any one time in Class A airspace is about 3,000, with a little more than half of that traffic operating

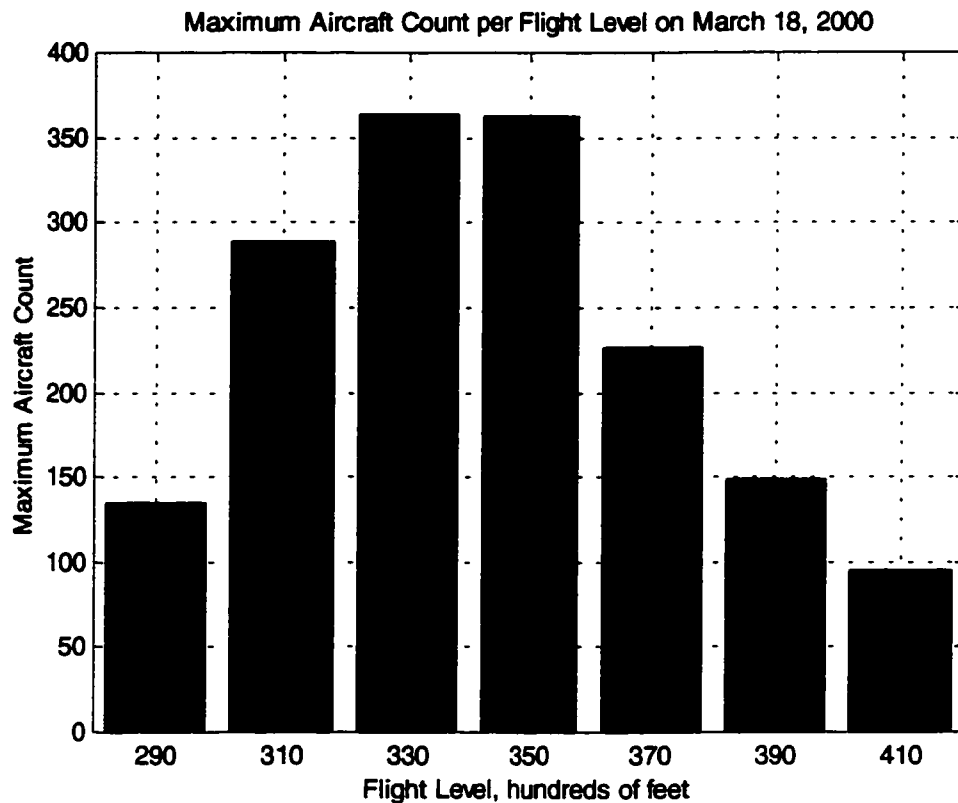


Figure 2.2. Maximum aircraft count per flight level over the U.S. on 18 Mar. 2000.

above FL290. The maximum number of aircraft operating at any given flight level is about 360 at FL330 and FL350 (fig. 2.2).

The airspace over the United States is divided into 20 Air Route Traffic Control Centers (ARTCC, pronounced “art-see,” or referred to as a “Center”) (fig. 2.3). A Center is a facility established to provide air-traffic control service to aircraft operating on Instrument Flight Rule (IFR) flight plans within controlled airspace, principally during the en route phase of flight. The Centers are further divided into sectors, and the sectors are partitioned into low-altitude, high-altitude, and sometimes super-high-altitude sectors. The size and dimensions of the sectors are determined by the traffic patterns and radar surveillance coverage in the area, and are each nominally controlled by one air-traffic controller.

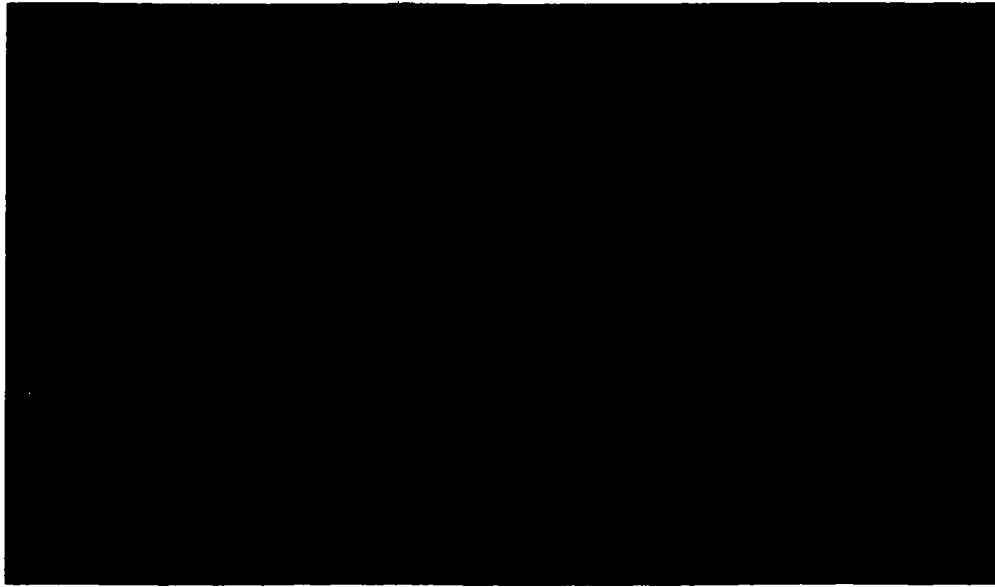


Figure 2.3. An ATC sector map of the United States.

In the current system, many additional constraints are placed upon aircraft trajectories in order to make tactical decisions tractable for human controllers and pilots. Among these constraints is a system of defined jet routes and navigation way points which aircraft must adhere to under normal circumstances. Within these constraints, aircraft must file flight plans prior to departure to specify their desired altitude, airspeed, and 2-dimensional (2-D) route of flight.

Airlines and business-jet operators use methods of varying levels of sophistication to choose routes that take into account the predicted wind patterns. They also consider constraints such as regions of predicted bad weather or turbulence. Each aircraft operator solves the optimization problem for himself without considering aircraft separation constraints. Separation constraints are handled by air-traffic controllers and pilots as they arise during flight. The current system performs reasonably well because the en route airspace is relatively sparsely populated. Air-traffic control also imposes flow constraints as required by weather or traffic congestion. Flow constraints are imposed by ground holds or ground stops, or through what are known as playbook routes. Playbook routes are

general sets of routes that are used to quickly route traffic around particular regions and that are implemented when bad weather conditions are encountered.

The problem with the current system is that it is inflexible and overconstrained. Requests for flight-plan changes are often not granted because controller work load is too high, and sometimes the desired route cannot be well approximated within the route constraints. Sometimes a desired route straddles sector boundaries in a way that is not permitted. If automation tools could be introduced that would enable controllers to use a longer-term, or strategic, approach to resolving inter-aircraft conflicts, efficiency would improve. These improvements would come from resolving conflicts more efficiently, and by permitting a reduction in the number of trajectory constraints.

2.3 Mathematical Problem Statement

Here, an abstract version of the optimal air-traffic control problem is posed to help introduce the assumptions and solution techniques that lead to the system concept. The main assumption is that the airspace is relatively sparsely occupied so that an approximate solution approach is expected to exhibit good performance.

Because the ATC system is so complex, one cannot write out a complete optimal control problem statement in functional form; however, much can be learned by abstracting and examining the properties of the general problem. For instance, it becomes clear why the optimal air-traffic control problem is considered to be NP-hard. The single-aircraft trajectory optimization problem is stated first, and then expanded to the multi-aircraft case by summing together the many individual trajectory optimization cost functions and adding the inter-aircraft separation constraints.

A general cost function for a single aircraft with adjoined final-state constraints may be expressed as

$$J_1 = \int_{t_{10}}^{t_{1f}} L_1[x_1(t), u_1(t), t] dt \quad (2.1)$$

where x_1 is the state vector, t is time, L_1 is the Lagrangian function for this aircraft, and u_1 is the control vector. One may put any desired terms in the Lagrangian, but for commercial aircraft one typically includes fuel-flow rate and total flight velocity terms in the Lagrangian.

The constraints consist of aircraft and other system state dynamic constraints (such as atmospheric parameters and winds):

$$\dot{x}_1(t) = f(x_1, u_1, t) \quad (2.2)$$

initial-state constraints:

$$x_1(t_{10}) - x_{10} = 0 \quad (2.3)$$

and final-state constraints:

$$x_1(t_{1f}) - x_{1f} = 0 \quad (2.4)$$

The objective is to choose the controls, u_1 , and the final time, t_{1f} , to minimize equation (2.1) subject to the constraints in equations (2.2) through (2.4).

This single aircraft optimization problem is challenging, but it has been solved for many common flight scenarios. In the absence of winds, the optimum horizontal route is simply a great circle so that only the optimum vertical profile need be calculated. A great circle is the shortest distance between two points on the surface of a sphere, the spherical equivalent of a straight line in a Cartesian system. The goal of the vertical profile optimization problem is to find the optimal speed and altitude profiles to achieve minimum fuel or the minimum direct operating cost (DOC) for a specified range. A discussion of the solution of the vertical-profile optimization problem is deferred until later in this chapter when a justification is presented for decoupling the vertical and horizontal trajectory optimizations.

Horizontal path optimization by means of equations (2.1) through (2.4) is generally not solved using calculus of variations techniques because of the excessive number of computations required, and because the solutions obtained may only be local minima. Discrete dynamic programming and randomized search algorithms have been used to obtain solutions. These are discussed in greater detail in the next chapter, where a neighboring optimal control solution is developed to solve the optimal horizontal route problem efficiently.

The problem becomes considerably more difficult when more aircraft are added. It is a challenge just to write the multi-aircraft cost function because different aircraft are continually entering and leaving the problem domain. One straightforward approach is simply to sum the individual cost functions for each aircraft:

$$J_{\text{ATM}} = \sum_{i=1}^N J_i = \sum_{i=1}^N \left[\int_{t_{i0}}^{t_{if}} L_i[x_i(t), u_i(t), t] dt \right] \quad (2.5)$$

The constraints are:

$$\dot{x}_i(t) = f(x_i, u_i, t) \quad 1 \leq i \leq N \quad (2.6)$$

$$x_i(t_{i0}) - x_{i0} = 0 \quad 1 \leq i \leq N \quad (2.7)$$

$$x_i(t_{if}) - x_{if} = 0 \quad 1 \leq i \leq N \quad (2.8)$$

If these were the only constraints, then the multi-aircraft optimization problem would simply be N decoupled single-aircraft optimization problems. However, there are more constraints to add that create the air-traffic control problem, that each aircraft must

maintain a minimum distance of separation from each other aircraft. These inequality constraints may be expressed as

$$\Delta x_{ij} \geq d_{\min} \quad \begin{cases} 1 \leq i \leq N \\ 1 \leq j \leq N \\ i \neq j \end{cases} \quad (2.9)$$

where Δx_{ij} is the distance between the i th and j th aircraft, and d_{\min} is the minimum allowable separation. When expanded, equation (2.9) represents $N(N-1)/2$ constraint equations that couple the optimal controls of each aircraft to those of each other aircraft. For 500 aircraft, that corresponds to almost 125,000 constraint equations. As discussed in chapter 1, there would be nearly $10^{37,500}$ possible aircraft orderings to examine in order to guarantee that the optimal solution had been found. That this is a difficult problem is an understatement.

Most of these constraints would not be active, because many of the aircraft would never physically encounter one another, but trajectory modifications for an aircraft may indirectly affect the optimal controls of any other aircraft. An interesting feature of the air transportation system is that there is no clearly defined beginning and end time for performing conflict detection and trajectory optimization. With the rare exception of the aftermath of the terrorist attacks on 11 September 2001, when all flights were grounded, there are commercial flights operating in the United States airspace at all times [13]. Because of the continuous nature of air transportation, the separation constraints effectively couple every trajectory in the airspace for all time. When performing strategic optimization, a flight that will occur 10 days or even 10 years from now can be shown to be coupled with a flight today by a continuous chain of separation constraints. A simple thought experiment shows why this is the case.

Imagine computing optimal trajectories for a set of aircraft that are scheduled to depart some time in the future. An aircraft (AC_1) is scheduled to depart from New York at 0000 Universal Coordinated Time (UTC) on 1 January 2000, and to arrive at San Francisco at

0600 UTC. Five hours later, at 0500 UTC, another aircraft (AC_2) takes off from San Francisco heading for New York along a trajectory that conflicts with AC_1 so that a conflict-resolution maneuver must be made. In general, an optimal maneuver would require that both aircraft must adjust their trajectories so that AC_1 would have to alter its trajectory beginning at 0000 UTC. Then, at 1100 UTC, well after AC_1 is scheduled to have landed, another aircraft (AC_3) is scheduled to depart New York for San Francisco, and this new aircraft conflicts with AC_2 which is arriving from San Francisco. The optimal resolution would require that both AC_2 and AC_3 adjust their trajectories, which in turn would affect the optimal resolution of the first predicted conflict between AC_1 and AC_2 . In this way, the optimal trajectory for AC_1 is shown to be dependent upon the trajectory of AC_3 , which is not scheduled to depart until after AC_1 has arrived at its destination. By induction, this process continues as long as there is no gap in air traffic.

This fact does not have disastrous consequences for finding feasible solutions in practice, but it does make strategic optimization extremely challenging. The couplings among aircraft become weak with time so that the look-ahead time for trajectory optimization may be safely cut off, with the caveat that the globally optimal solution to the real problem is being abandoned.

2.4 Prior Art: 4-Dimensional System Concepts

The air-traffic-control optimization concept presented in this dissertation is based on an idea that has been around for several decades. This is the idea that aircraft would be assigned 4-dimensional (4-D) flight plans designed to be conflict-free, and that they would use closed-loop 4-D control to follow the assigned flight plans without requiring regular tactical instructions from air-traffic control (fig. 2.4). If each aircraft guarantees that it is following its assigned 4-D trajectory, and if the trajectories are designed to be conflict-free, then tactical conflicts will not arise. Verification for this algorithm is achieved by verifying each individual 4-D control system rather than by verifying a complex system of stochastic multi-aircraft tactical conflict resolution. This concept is referred to as 4-D Time-Based Control, or simply 4-D Control. In addition to determining feasible conflict-

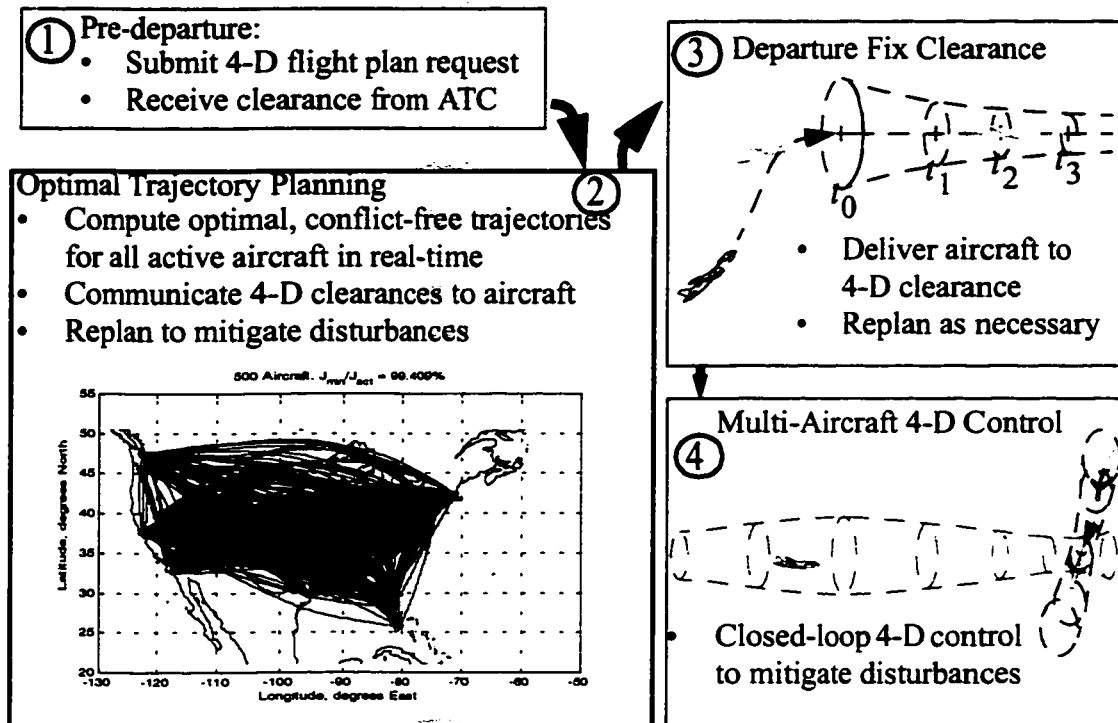


Figure 2.4. High level concept diagram of 4-D time-based air-traffic control.

free 4-D trajectories for all aircraft, a further goal is to find trajectories that are efficient. This is conceptually different from the current air-traffic control system in which much more use is made of tactical conflict resolution.

The current air-traffic control system may be thought of as an approximate solution to the ATC optimization problem, although it has probably never been formally considered in this way. In the current ATC system, aircraft operators file a flight plan which includes a proposed departure time, a coarsely defined horizontal flight route, an intended flight-level (altitude), and an intended airspeed. The flight plan does not include intended time along the proposed flight route. Air-traffic controllers monitor aircraft positions and make adjustments to aircraft speeds, flight-level assignments, and headings to keep aircraft safely separated from one another.

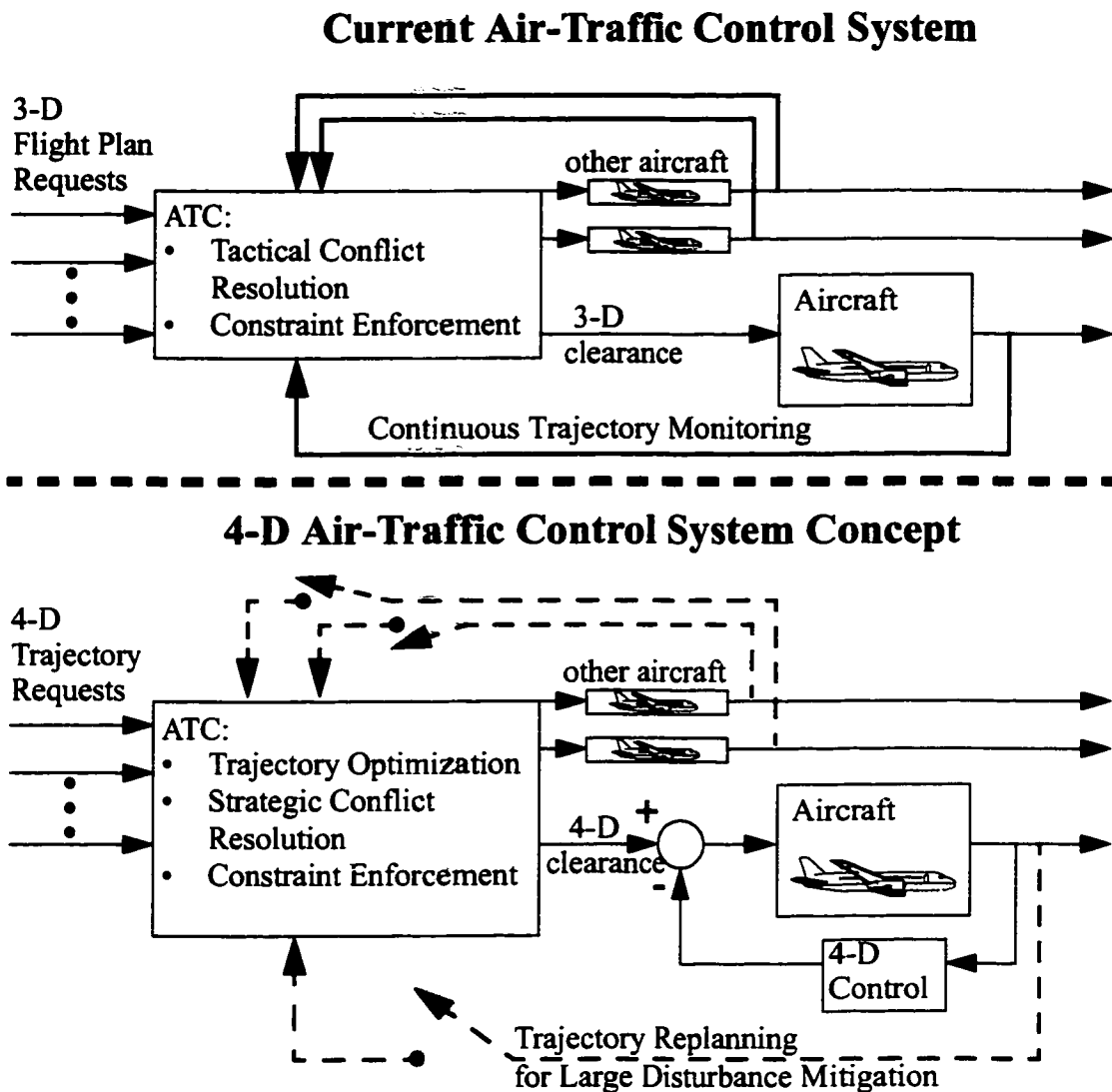


Figure 2.5. Block diagrams of 4-D ATC and the current ATC system.

The conceptual difference between the current ATC system and a 4-D ATC system is depicted in figure 2.5. The current air-traffic control system is shown as a feedback control system in which 3-D trajectory clearances are input to the aircraft, and the control loop is closed through the control actions of air-traffic controllers. The aircraft are monitored along their clearances by air-traffic controllers who issue tactical advisories as required to maintain safe separation among the aircraft. In the 4-D air-traffic control system concept, inputs to the aircraft are 4-D trajectories that specify the clearance as

position versus time. The aircraft close the control loop locally using 4-D control to maintain their positions along their assigned 4-D clearances. Occasional trajectory replanning is required for large disturbance mitigation.

Among the first 4-D air-traffic control concepts were those proposed by Erzberger beginning in the early 1970s. One of the first such research studies was to be applied to short takeoff and landing (STOL) vehicles [14].

Just a few years later, a time-based ATC system was proposed by a group headed by Ralph Erwin at Boeing Commercial Airplane Co. [15]. Again, the basic concept was that ATC would determine non-conflicting 4-D flight paths for all aircraft and that the pilots would execute the clearances by accurately flying the assigned 4-D paths. It was suggested that speed be used to adjust aircraft spacing rather than path-stretching. The 4-D trajectory clearances would be computed to meet assigned arrival times at the respective arrival runways. The paper concluded with the following statement: *"It seems inevitable that this system is coming. The question is, how soon will it get here?"*

It has been 25 years, and although many advances have been made in guidance and navigation, weather prediction, and other air-traffic control technologies, the implementation of a time-based air-traffic control system seems still to be far off. One of the primary reasons for the delay is that the computation of efficient conflict-free 4-D trajectories for all aircraft in real-time has remained an unsolved problem.

The next advances in 4-D concepts were made in the development of ground-based automation tools. It was realized that airborne 4-D control concepts could not be fully exploited unless air-traffic control could accommodate 4-D capabilities. By equipping the ATC system with accurate 4-D trajectory planning, effectively giving the system the capability to serve as a flight management system for all aircraft in its domain, the hope was that time-based ATC would move closer to reality. Several concepts were developed to varying degrees of technological readiness [16-20]. The most highly developed system, called the Center TRACON Automation System (CTAS), was developed by a team of researchers led by Erzberger at the NASA Ames Research Center from the mid-1980s

through the present day [17]. The CTAS concept uses scheduled time-of-arrival (STA) control under which aircraft are given clearances to arrive at a meter fix or runway at a scheduled time. The precise trajectory leading to the final condition is not specified, and tactical advisories are to be issued for conflict resolution (fig. 2.6). Prototypes of CTAS

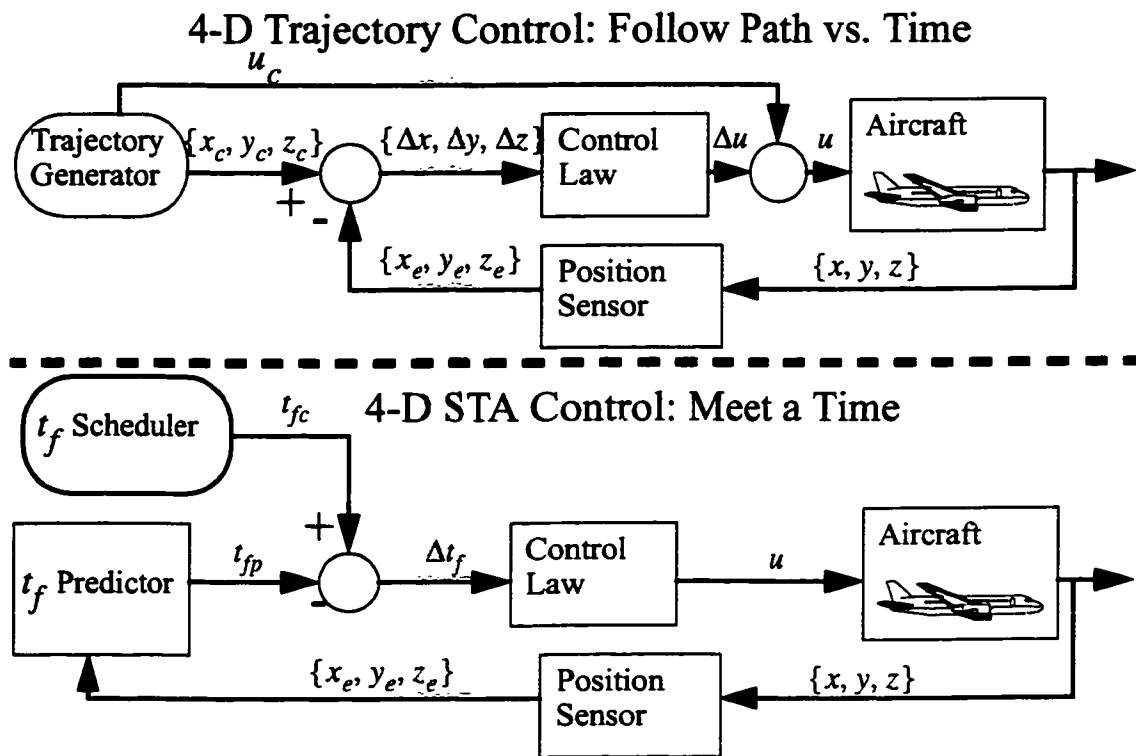


Figure 2.6. 4-D trajectory control and 4-D STA control.

have been successfully field tested at several Air Route Traffic Control Centers (ARTCC) and Terminal Radar Approach Control (TRACON) facilities, but the active 4-D control aspects have not yet been adopted. Many of the reasons are human-factors related, but there have also been some difficult practical problems to overcome when trying to use time-based control in the extremely busy terminal-area environment. Some of the human-factors difficulties stem from deciding whether or not air-traffic controllers are legally responsible for aircraft separation assurance when advisories are being provided by automation.

The Group for ATM R&D Technology in Europe (GARTEUR) conducted research in the late 1980s on a time-based air-traffic control concept [19]. The GARTEUR concept was to provide aircraft with trajectory constraint boundaries, within which they could self-optimize. The GARTEUR approach was superseded by a new initiative called the Programme for Harmonised ATM Research in Europe (PHARE). The PHARE concept was to develop and implement 4-D time-based control using conflict-free 4-D tubes [19, 20] negotiated between aircraft operators and air-traffic control. Tubes are volumes of airspace that are to be assigned to aircraft as functions of time so that a clearance would consist of both a nominal 4-D trajectory and a buffer region around the trajectory. The program developed to the point where detailed tube specifications were created for communicating tube clearances between ATC and an advanced new 4-D flight management system. The PHARE program ended without the implementation of the tube concept. Anecdotal evidence suggests that the assigned 4-D trajectories were deemed to be too restrictive and cumbersome by both pilots and controllers. The fundamental problem of safely and reliably computing conflict-free 4-D tube updates in real time was never solved.

A concept called “Trajectory Space” was proposed in the early 1990s. Trajectory Space was conceptually similar to the GARTEUR concept developed in Europe [19]. The idea was for ATC to communicate constraint regions to aircraft which would then self-optimize within those constraints. Only a brief concept description was presented. Although conceptually appealing, Trajectory Space would be extremely difficult to implement in the suggested manner. The specification of regions of unavailable airspace and the coordination among aircraft would be difficult to code into a feasible algorithm. Many other difficult problems arise, such as guaranteeing that a feasible trajectory even exists for a given set of constraints.

The Free-flight concept was proposed beginning in the mid-1990s [21]. The initial idea was that aircraft could use advanced conflict-detection and resolution technologies to perform tactical conflict resolution without any direction from ATC. The hope was that aircraft could then fly on free routes, unrestricted by the jet route structure in the NAS

today. Over time, a more sober view of free flight emerged in which ATC would have a role in monitoring for conflicts and providing separation services when resolution maneuvers could not be worked out by the aircraft involved. As research has progressed, free routing still remains a distant hope; technologies have not yet been developed that can assure that multi-aircraft conflict situations can always be resolved tactically. Significant theoretical advances have been made in this area using hybrid control theory [22, 23], but the application of these ideas to practical air-traffic control systems remains for the subject of future work.

Recently, work has begun on the design of practical automated airspace architectures that would enable implementation of some of the key optimization and conflict resolution functions developed over the past decade [24]. The goal is to use automation to lower air-traffic controller work load so that en route airspace capacity may be increased while enabling aircraft to fly along unrestricted trajectories. Particular attention is given to the development of a core system to guarantee safe separation for all aircraft. This is an approach that may enable the implementation of free-flight concepts. The development of automated airspace architectures will create the infrastructure required for trajectory optimization approaches such as the one offered in this dissertation.

The concepts of free flight and 4-D time-based control are not mutually exclusive. The goal of free flight is not to enable airborne anarchy; the goal is to remove the restrictions of the jet route structure and to allow aircraft to fly along more efficient trajectories. As is pointed out in this dissertation, when free routing is permitted, the airspace is relatively sparsely occupied so that the trajectories that would result from tactical conflict resolutions proposed under free flight might still be quite close to a theoretical global optimum solution. However, if any degree of congestion is encountered, one must rely on strategic optimization to achieve the most efficient solution. The strategic optimization of trajectories is what is enabled by 4-D time-based control. One might think of 4-D time-based control as a free-flight system run in fast-time to determine a set of optimal trajectories that would then be followed by aircraft.

2.5 Analysis and Simplifying Assumptions

Although the en route air-traffic control problem may be posed fairly concisely, it appears to be too complex to solve in real-time. One must make insightful simplifying assumptions so that the solution to the approximate problem is close to the solution to the original optimization problem. Three main simplifications that lead to a feasible solution approach with near-optimum performance are discussed here. The first is that aircraft trajectory optimization may be decoupled into vertical profile optimization and horizontal route optimization. The second assumption is that the airspace is relatively sparsely occupied. This leads to the third simplification, the hypothesis that a sequential optimization and deconfliction approach could produce feasible near-optimum solutions. Air-traffic control by means of a sequential optimization approach is shown to be a problem that can be solved in polynomial time. This means that as the number of aircraft increases, the amount of time required to solve the problem may be expressed as a finite polynomial function of the number of aircraft rather than as an exponential function.

2.5.1 Horizontal/Vertical Optimization Decoupling

The optimization of vertical profiles (airspeed and altitude schedules) for commercial jet transport aircraft has received considerable attention over the past 40 years [25-34]. The essence of this problem is to balance the time and fuel costs to travel between a specified origin and destination. Fuel use is a function of airspeed and ambient pressure and temperature. These latter two quantities may be expressed as functions of altitude for an assumed standard atmosphere.

The basic vertical optimization problem is to minimize the direct operating cost (DOC) expressed in dollars:

$$J_{\text{DOC}} = C_f \left\{ \frac{\$}{\text{lb}} \right\} \int_{t_0}^{t_f} \left(\dot{V} \left(\frac{\text{lb}}{\text{hr}} \right) + CI \left[\frac{\$/\text{hr}}{\$/\text{lb}} \right] \right) dt \quad (2.10)$$

where J_{DOC} is a measure of the total cost for a given aircraft trajectory from the specified initial time, t_0 , to the free final time, t_f . The cost of fuel, expressed in dollars per pound, is

given by C_f , and the fuel-flow rate for the particular aircraft/engine combination is given as a function of airspeed and altitude by $\dot{f}(V, h)$, expressed in pounds per hour. The Cost Index, CI , represents the cost of time relative to the cost of fuel. Calculus of variations techniques have been successfully applied to this problem for a fixed range and free final time in a standard atmosphere using an energy-state formulation of the problem [25, 26]. In the energy-state formulation, the integral cost function of equation (2.10) is converted into an equivalent form by defining an energy state as the sum of kinetic and potential energy of the aircraft and integrating over that variable instead of over time. One of the benefits of converting to the energy state formulation is that the free final-time problem is converted to a problem with a fixed terminal condition. These methods were extended to the computation of optimal vertical profiles for problems with fixed final time. In this case, the cost index is varied iteratively until the desired final time is achieved [27, 28]. The effects of winds on optimum vertical profiles has also been examined, with practical constraints on commercial aircraft flights being considered [29-33]. Research has also been conducted on the use of soft dynamic programming approaches to minimizing equation (2.10) when there are multiple time constraints [34].

The main conclusion of vertical-profile optimization for long-range commercial jet aircraft is that optimal profiles consist of short climb and descent segments and a long cruise-climb segment with a shallow climb angle (fig. 2.7). The shallow climb segment

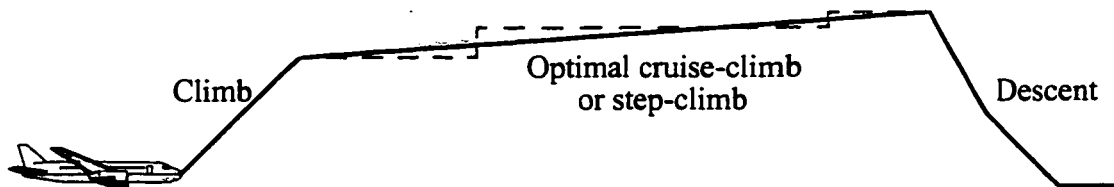


Figure 2.7. Optimal cruise-climb and step-climb profiles.

results because the optimal altitude slowly increases as the aircraft burns off fuel. In the current constrained air-traffic control environment where aircraft must operate at flight

levels that are separated by 4,000 ft, the optimal cruise climb segments have been shown to consist of a stair-step segments that are about 500 n. mi. long [32]. This may be considered as a constant altitude for the purpose of determining optimum horizontal routes.

Since the winds vary in a nonlinear fashion, the optimization problem with wind may have many local minima, and is more difficult to solve. Fortunately, the range of efficient altitudes for contemporary commercial jet aircraft is narrow, so that a practical approach is to examine one or two altitudes near the optimal no-wind altitude and then to choose the one that gives the minimum cost. This trial and error search method is reasonably efficient and is what is used today by airlines and commercial flight planning services.

There will ordinarily be some coupling between the optimal vertical profile and the optimal horizontal route owing to variations in the winds. If an optimal vertical profile is computed for a great-circle route using the procedure outlined above, it may be possible that shifting the horizontal route to take advantage of horizontal wind shear might take the aircraft to a region of airspace where a different altitude would be more efficient. In mathematical terms, the rate of change in the wind vector with altitude is also a function of horizontal position. Empirical evidence suggests that, except near the boundaries of a jet stream, the coupling of vertical and horizontal wind shear is weak. The practical approach to finding a feasible optimum solution is again to perform a local iterative search of altitudes and routes nearby the optimum no-wind solution. The iterative procedure would be as follows:

1. Compute optimum altitude and airspeed profile assuming zero winds
2. Compute the cost of flying at the nearest legal altitudes above and below the zero-wind optimum altitude and choose the one with minimum cost
3. Fix the altitude and airspeed and compute the optimum horizontal route for the given wind conditions
4. Fix the horizontal route and search the nearest legal altitudes above and below the altitude chosen in step 2; choose the one with minimum cost
5. Iterate until desired convergence is achieved

The vertical-profile optimization and horizontal-route optimization may therefore be precomputed in parallel by aircraft operators. This is a significant simplification in the aircraft trajectory optimization problem, because it decouples the fairly straightforward vertical-profile optimization from the much more challenging horizontal-route optimization problem. Efficient algorithms for computing the optimum horizontal route are still required; this problem is addressed in chapter 3.

2.5.2 Sparse Airspace Assumption

A review of recent ATC research literature suggests that en route air-traffic congestion is a severe problem. In the vicinity of a busy commercial hub airport, one might be able to count a steady stream of arrival and departure aircraft, but these aircraft would occupy linear regions of airspace, and they would still be separated by more than 5 n. mi. Except for the choke points created by air-traffic-control route restrictions, the en route airspace is actually sparsely occupied and will continue to be so until many more airports are built. This has important implications for the computation of approximate optimal air-traffic-control solutions.

A recent study examined the properties of air-traffic conflicts for both structured routing and great-circle routing [1]. Flight plan data were taken from the Enhanced Traffic Management System (ETMS) for a 24-hr period in March 2000. The Future Air Traffic Management Concepts Evaluation Tool (FACET) [44] was used to simulate aircraft flying

either along the filed flight plans or along great-circle routes between the scheduled origin and destination airports within Class A airspace (above FL180). At each 15-sec integration time-step, the number of active conflicts in the airspace was recorded, as was the total number of aircraft in the airspace at that time.

These conflict data may be used to evaluate the aircraft density of the airspace environment. A high number of conflicts and a steep rate of growth of conflicts would suggest that the airspace was nearing saturation. Conversely, a low number of conflicts and a shallow growth rate would suggest that the airspace still had plenty of extra maneuvering volume remaining.

The plot of the number of instantaneous conflicts versus the number of aircraft flying shows that for both structured routing and great-circle routing, the number of conflicts is quite low, and the growth rate is still shallow (fig. 2.8) (Note: The study [1] referred to great-circle routing as “free routing,” but as shown in chapter 4, optimal routes may be quite different than great-circle routes). Without assuming any prior knowledge of aircraft paths, it is equally likely that one aircraft will be in conflict with any other aircraft. This suggests that the number of conflicts, X , for any aircraft may be modeled as a binomial random variable such that the probability mass function is given by

$$P(X = x) = \binom{n}{x} p^x (1 - p)^{n-x} \quad \begin{cases} 0 \leq p \leq 1 \\ x = 0, 1, \dots, n \end{cases} \quad (2.11)$$

where $n \equiv (N - 1)$, N is the number of aircraft flying, and

$$\binom{n}{x} \Rightarrow n \text{ choose } x \Rightarrow \frac{n!}{x!(n-x)!} \quad (2.12)$$

With this model, the expected number of conflicts for a single aircraft in a field of N aircraft is given by

$$E[X_N] = (N - 1)p \quad (2.13)$$

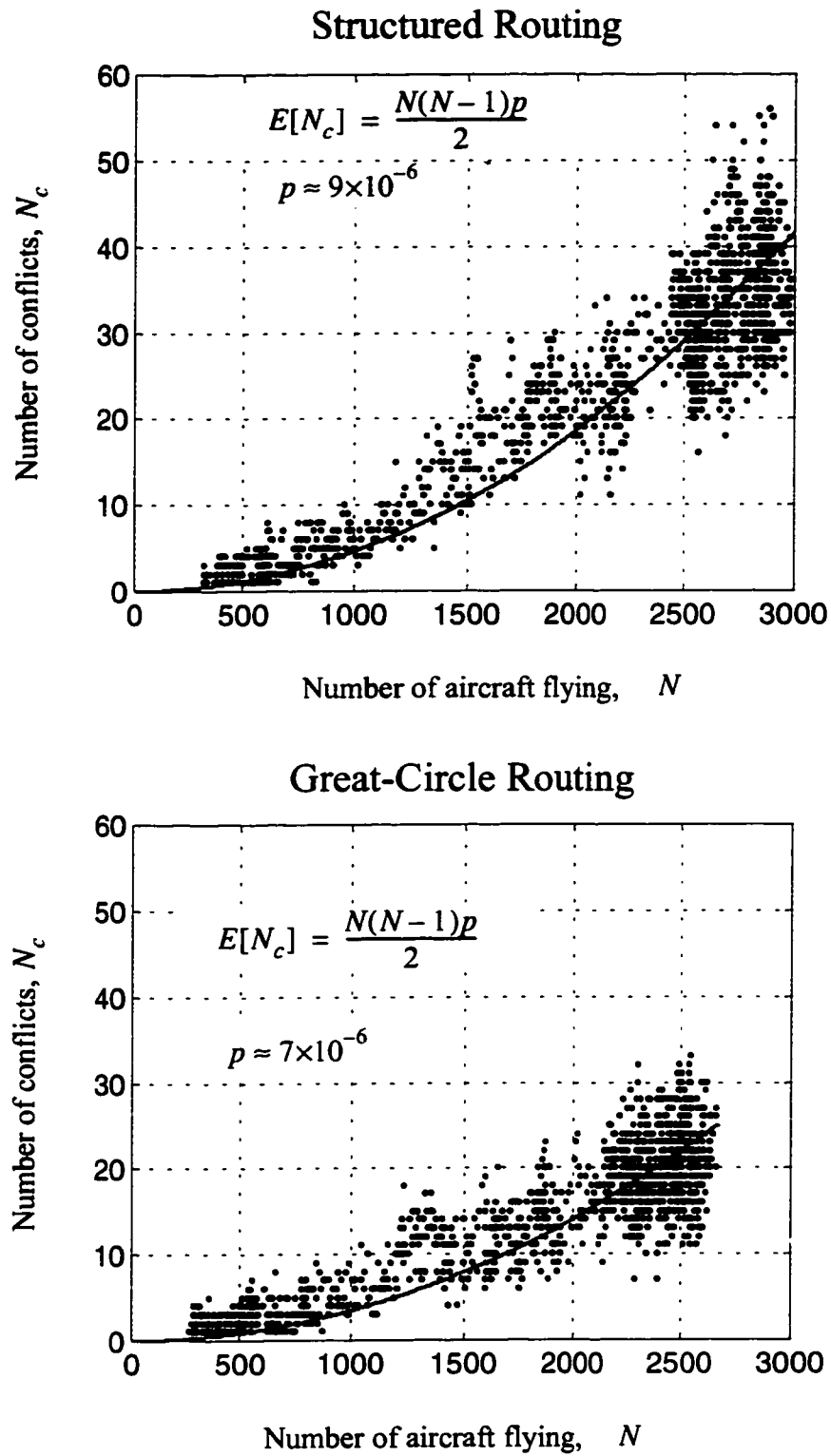


Figure 2.8. Conflict counts for structured and great-circle routes.

and the expected sum total number of conflicts (divided in half so that conflicts are not counted twice) is given by

$$N_c \equiv \sum_{i=1}^N \frac{E[X]}{2} = \frac{N(N-1)p}{2} \quad (2.14)$$

By choosing the aircraft-to-aircraft conflict probability, p , to fit the data in a least square error sense, the following probabilities and expected values are computed for structured or great-circle routing (table 2.1):

Table 2.1. Conflict statistics from binomial random variable model.

| Statistic | Description | Flight plan | Great circle |
|-------------------------------------|---|--------------------|--------------------|
| p | Probability of aircraft i conflicting with any other aircraft j | 9×10^{-6} | 7×10^{-6} |
| $E[X_{3000}]$ | Expected number of conflicts per aircraft in Class A airspace with 3,000 aircraft | 0.027 | 0.021 |
| $\sum_{i=1}^{3000} \frac{E[X]}{2}$ | Expected sum total number of conflicts in Class A airspace with 3,000 aircraft | 40.5 | 31.5 |
| $E[X_{30000}]$ | Expected number of conflicts per aircraft in Class A airspace with 30,000 aircraft (extrapolated) | 0.27 | 0.21 |
| $\sum_{i=1}^{30000} \frac{E[X]}{2}$ | Expected sum total number of conflicts in Class A airspace with 30,000 aircraft (extrapolated) | 4050 | 3150 |

These probabilities and expected numbers of conflicts are exceedingly low. In a field of 3,000 aircraft, only about 1% of the aircraft are ever expected to experience a conflict. Conflict probabilities are somewhat lower for great-circle routing than for flight plan routing because aircraft are able to utilize a greater amount of airspace.

For a free routing system, the airspace at current levels of operation appears to be quite sparsely occupied, with plenty of airspace available for additional aircraft operations. But what about in the future? How many more aircraft can operate in the airspace before conflict probabilities increase to an unmanageable level? An interesting way to examine these questions is to estimate the average number of aircraft that can be expected in the airspace if all current airports are operating at maximum arrival and departure rates.

The same study [1] showed that there are approximately 40,000 daily flights and that at any instant in time there are 3,000 or fewer aircraft in Class A airspace. Further examination of the data shows that there are only about 200 airports serving Class A airspace on a regular basis. The average number of active runways for these 200 airports is difficult to compute precisely, but two active runways per airport is a reasonable assumption.

The maximum arrival rate per runway, R , may be estimated by landing aircraft at 5 n.mi. separation at a typical landing speed of about 160 knots:

$$R = \frac{1 \text{ aircraft}}{5 \text{ nmi-rwy}} \cdot \frac{160 \text{ nmi}}{\text{hr}} = \frac{32 \text{ aircraft}}{\text{hr-rwy}} \quad (2.15)$$

so that the average acceptance rate per airport at full capacity is approximately 65 aircraft per hour.

Analysis of daily air traffic operating at flight levels 330 and 350 (approximately 33,000 ft and 35,000 ft MSL, respectively) shows that the average flight time is about 2.4 hr. By computing the number of aircraft that would be in the airspace after operating for 2.4 hr at the maximum average acceptance rate, R , the maximum number of en route aircraft, N_{\max} , may be approximated as

$$N_{\max} \approx 200 \text{ airports} \cdot 65 \text{ ac/hr-airport} \cdot 2.4 \text{ hr} \approx 30000 \text{ aircraft} \quad (2.16)$$

This is a factor of 10 greater than the number of aircraft operating today. By equations (2.13) and (2.14), this means that one would expect 10 times the number of conflicts for

each aircraft, and 100 times the sum total number of conflicts (table 2.1). This would be a significant increase over current operations, but the conflict probabilities would still be low.

The implication is that even if all the current airports ran continuously at full capacity, the airspace would still be relatively sparsely occupied. For this to occur, many more aircraft would need to be manufactured as well. This means that a sparse airspace assumption is not only valid for current traffic levels, but will continue to be valid for many years to come until more airports and runways are built, or until other existing airports are utilized for high-altitude air traffic.

2.6 Sequential Trajectory Optimization

The sparse airspace assumption turns out to be one of the keys to achieving fast and efficient solutions to the air-traffic control optimization problem. Because of this assumption, the use of an iterative trial-and-error approach to finding conflict-free routes is possible. The importance of this must not be underestimated. By the sparse airspace assumption, one only need know that there is a conflict. It is not important with which constraint the conflict is taking place, or even the particular geometry of the conflict. Without this assumption, one must compute precise conflict-resolution maneuvers using information about all aircraft involved in each specific conflict situation. The computational overhead required can overwhelm even the computations required for route optimization.

The sparse airspace assumption implies that the cost function in equation (2.5) is flat near the optimum solution. This means that of the many conflicts that might arise, one would obtain nearly the same total cost no matter how the conflicts were resolved as long as the initial trajectories were nearly optimal. Many researchers have examined approaches to conflict resolution (for small numbers of aircraft) that are in some way optimal. These approaches are discussed in greater detail in chapter 5. When extended to the case of many aircraft, the attempt to find optimal resolution maneuvers contributes to the explosion in complexity. This happens because each conflict-resolution maneuver

changes the constraints faced by every other aircraft. To reduce the complexity from NP to polynomial-time, the problem solution must be constrained. A reasonable place to start is with the optimal trajectories for each aircraft independent of any conflicts that might occur. The computation complexity for this task, using the optimization algorithm described in chapter 4, is just N , where N is the number of aircraft. The next step is to detect any conflicts and to make adjustments in the aircraft trajectories to resolve those conflicts. It is already known that if all possible resolution maneuvers are examined, the problem complexity is probably NP-hard, so a much smaller subset of resolution maneuvers needs to be selected for examination.

One approach to limiting the number of possible conflict-resolution maneuvers is to allow only heading maneuvers to one side or the other until all conflicts for that aircraft have been resolved. Similar approximations have been suggested in previous research, also for the purpose of reducing the complexity of the problem [45]. If this procedure is followed in a sequential manner, with trajectories of previously planned aircraft being held fixed, then the worst-case situation would occur if each aircraft had a conflict with each previously planned aircraft. For a set of N aircraft, the maximum number of resolution maneuvers would be

$$N_{cr} = \sum_{i=1}^N (i-1) = \frac{N(N-1)}{2} \quad (2.17)$$

This problem may be solved in polynomial-time, even in the worst case.

There are potentially many different ways to compute feasible trajectory solutions that meet all the separation constraints, but to explore each of these would be well beyond the scope of this dissertation. Instead, only this sequential optimization approach is developed. No claims are made regarding whether the sequential optimization approach is the best possible, only that solutions may be computed in polynomial time. This approximate approach will be used to justify several other algorithm choices. The guiding

principle is to keep the computations as simple as possible so that near-optimum performance can be achieved in real-time.

2.7 A Practical Bound on the Optimum Solution

Although it is not currently possible to compute the true optimum conflict-free solution in a reasonable amount of time, a conservative bound on the minimum cost can be computed by summing the optimum costs for all aircraft while ignoring any conflicts or other constraints that might arise.

Aircraft trajectory optimization is usually only concerned with minimizing direct operating costs. If vertical-profile optimization is decoupled from horizontal-route optimization, then the optimal aircraft speed and altitude profiles are fixed. This turns the horizontal-route optimization problem into a minimum-time problem for fixed aircraft airspeed.

The minimum flight time for aircraft i , without regard to separation constraints, is denoted by J^*_i . This is computed by minimizing equation (2.1) with the Lagrangian function set to unity so that the cost function is simply the total flight time, and by constraining the aircraft to fly at a specified altitude and airspeed profile. The minimization is subject to the initial and final state, and the dynamic constraints of equation (2.2) through (2.4). The essence of this optimization problem is to compute the heading commands that will cause the aircraft to travel between the initial and final-state constraints in the minimum time.

The total minimum time for a set of N aircraft is given by

$$J^*_T = \sum_{i=1}^N J^*_i \quad (2.18)$$

In any situation where additional constraints may be placed upon the aircraft, the actual total flight time for a set of N aircraft is given by

$$J_T = \sum_{i=1}^N J_i \quad (2.19)$$

The efficiency of any particular solution to the air-traffic control optimization problem is now defined as

$$\eta_{\text{ATC}} \equiv \frac{J_T^*}{J_T} \quad (2.20)$$

By definition, the actual cost must always be greater than or equal to the minimum cost. At least one feasible solution to the air-traffic control optimization problem can always be shown to exist (e.g., all aircraft fly their routes sequentially, with only one aircraft being allowed in the air at any given time) so that the actual cost is finite. This leads to the following bounds on the efficiency parameter:

$$0 < \eta_{\text{ATC}} \leq 1 \quad (2.21)$$

This efficiency parameter is relatively easy to compute, so it provides a practical means of evaluating optimization performance. The efficiency parameter also provides an excellent means of evaluating the sparse airspace assumption. If feasible solutions to the air-traffic control optimization problem have low efficiencies, it would suggest that the sparse airspace assumption is invalid. Conversely, if feasible solutions with high efficiencies are commonly found by using simple algorithms, the sparse airspace assumption would then be strongly substantiated.

2.8 Summary

This chapter introduced the en route air-traffic control optimization problem, including a high-level description of how the air-traffic control system handles en route traffic today. The problem was posed as one of minimizing the sum of N cost functions

subject to up to $N((N - 1)/2)$ inter-aircraft separation constraints. These constraints were identified as the reason why this problem is considered to be NP-hard, and therefore intractable. Prior solution approaches and their limitations were then presented.

Analysis of the aircraft trajectory optimization problem was used to justify decoupling the vertical-profile and horizontal-route optimization functions, which greatly simplifies the trajectory optimization problem. A probabilistic model of the expected number of aircraft conflicts was derived and fitted to empirical data to support the assumption that en route airspace at present traffic levels is sparsely occupied. Further analysis was used to show that the airspace will continue to be sparsely occupied well into the future.

The sparse airspace assumption led to the choice of a sequential optimization algorithm using an optimal wind-routing algorithm with a guided trial-and-error approach to conflict resolution. A simple optimization performance metric was derived to enable the real-time monitoring of optimization performance for the sequential optimization algorithm.

The next chapter defines the detailed system concept. After presenting the high-level concept, the component algorithms are identified for further description in the following chapters. Among these component algorithms are an optimal wind-routing algorithm, a conflict-detection algorithm, and a conflict-resolution algorithm. A computational analysis of the sequential optimization algorithm is also presented.

Chapter 3

System Concept

3.1 Introduction

The need to compute feasible sets of conflict-free trajectories for all aircraft in real time has been identified as a major challenge to the implementation of 4-dimensional (4-D) time-based air-traffic control in the National Airspace System (NAS). This problem is compounded by the fact that aircraft operators (primarily commercial airlines) will not accept merely feasible routes, they require efficient (optimal) routes. In the current tactical system, airlines file flight plans along routes that are predicted to be wind-optimal within the constraints of the NAS, but these flight plans are not checked for predicted conflicts. Instead, aircraft routes are significantly constrained so that tactical conflict-resolution maneuvers may safely be used. Tactical conflict-resolution maneuvers degrade the performance of aircraft routes, and a system that relies on tactical conflict-resolution cannot be proven to be safe. In an automated 4-D time-based system where flight plans must be guaranteed to be conflict-free, some form of strategic trajectory optimization that satisfies the separation constraints is required.

The air-traffic control optimization problem introduced in chapter 2 has been categorized as NP-hard [5]. Analysis of the practical problem led to a few key simplifying assumptions so that feasible solutions might be achieved without greatly sacrificing optimality. In summary, these simplifying assumptions and their implications are as follows:

1. Horizontal- and vertical-trajectory optimization are loosely coupled: The implication is that horizontal- and vertical-trajectory optimization may be computed independently. This greatly simplifies the task of trajectory optimization.
2. The airspace is relatively sparsely occupied, and will continue to be so: The sparse airspace assumption implies that the air-traffic control cost function is flat near the optimum solution. This suggests that sequential trajectory optimization by means of a trial and error approach should lead to feasible near-optimum solutions.

The simplifications and approximations used to solve the optimal air-traffic control problem are necessary to enable real-time computation because of the prohibitive complexity of the original problem. Throughout the development of this approach, an appeal is made to the two key simplifying assumptions listed above. The sequential optimization approach introduced in this chapter is shown to produce near optimum solutions in polynomial time.

3.2 Sequential Optimization Algorithm

A high-level flowchart for the sequential optimization algorithm is presented in figure 3.1. The algorithm begins by putting all scheduled aircraft into an ordered list called the Active Aircraft List (AAL), which is to be described in more detail. The optimal vertical profiles for each aircraft are assumed to have been chosen, and they are inputs to the horizontal-route optimization algorithm. This was enabled by the decoupling of horizontal and vertical trajectory optimization. The optimal horizontal route for the first aircraft on the AAL is computed and checked for conflicts. Note that there will be no conflicts with other aircraft for the first aircraft on the AAL, but conflicts with regions of bad weather or with special-use airspace may occur. If any conflicts are found, the trajectory is iteratively modified until a conflict-free trajectory results. The algorithm proceeds through all aircraft on the AAL until all have optimal conflict-free trajectories. At this point, the trajectories may be communicated to the aircraft as clearances, and the optimization procedure may be restarted as often as required. In between recomputation cycles, it may be advantageous or

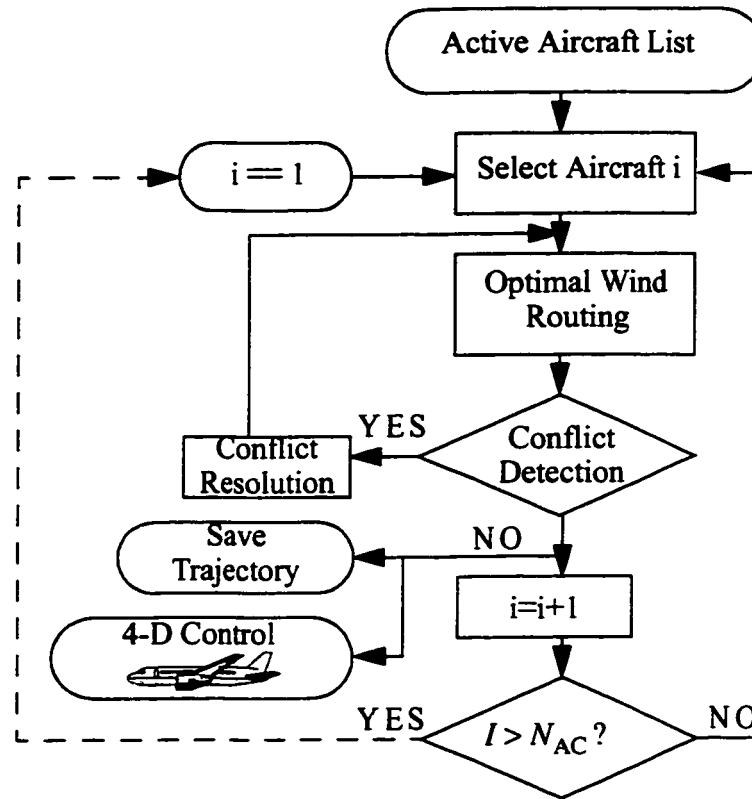


Figure 3.1. Sequential route optimization algorithm flowchart.

even necessary for aircraft to use active control to follow their assigned 4-D trajectories to ensure conflict-free operations. The main component blocks depicted in the flowchart are now introduced, but detailed discussion is deferred to later chapters.

3.2.1 The Active Aircraft List

The Active Aircraft List (AAL) is defined as all aircraft currently in flight in the airspace domain of interest, plus all aircraft scheduled to enter the domain within the next ΔT_B time period. Most aircraft entering the domain will enter after departing from their respective airports, but aircraft may also enter through any arbitrarily chosen airspace boundaries. For example, if Class A airspace over the continental United States is the domain of interest, then arriving international flights will enter the domain at points not associated with any particular airport.

The purpose of ΔT_B is to provide for the computational latency for a complete pass through the AAL. If the latency were not accounted for, then new aircraft would enter the airspace domain before conflict-free trajectories had been computed for them. This would be unacceptable. Therefore, ΔT_B must be set according to the expected maximum time to complete a pass through the AAL. Some additional time should be allotted in practice to provide ample time to address any potential optimization or conflict-resolution difficulties before aircraft enter the airspace domain. The value of ΔT_B is left as a variable parameter so that the processing speeds of different hardware systems may be evaluated in simulation.

The ordering of the AAL still must be chosen. Many different factors might be used to order the aircraft. Among these are the following:

- Random selection
- Market-driven (aircraft operators pay for the right to higher priority in trajectory planning)
- First-scheduled-first-served

Without performing complete parametric studies on actual traffic patterns and accurate models of system uncertainties, the first-scheduled-first-served (FSFS) option makes intuitive sense and is used here. Studies have indicated that when uncertainty is considered, there is an optimal look-ahead time for resolving conflicts because there is a trade-off between solution efficiency and the probability that a conflict will actually occur [46]. For a given level of uncertainty, there is a theoretical optimum time to perform conflict-resolution, and this suggests that conflicts should be resolved in chronological order; this makes intuitive sense because, on average, aircraft that have entered the airspace domain earlier will have less time remaining on their trajectories, and therefore any conflict-resolution maneuvers will increase costs more than resolution maneuvers for aircraft with more of their trajectories remaining. A simple example shows why this is the case.

For an aircraft traveling from point *A* to point *B*, imagine that there is a conflict that will occur somewhere along the trajectory at point *C* (fig. 3.2). The resolution of the conflict requires that the aircraft deviate from the straight-line path to point *E* before

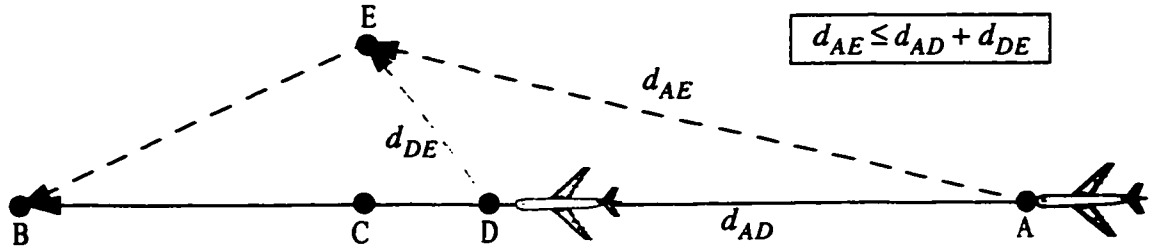


Figure 3.2. Resolving conflicts earlier in a deterministic system is more efficient.

resuming course to point *B*. If the conflict is resolved from point *A*, the total additional flight distance for this first resolution case is given by

$$d_{r1} = d_{AE} + d_{EB} \quad (3.1)$$

The total additional flight distance for a second resolution case where the aircraft waits until a later point, *D*, to begin the resolution maneuver is given by

$$d_{r2} = d_{AD} + d_{DE} + d_{EB} \quad (3.2)$$

The additional distance for the second resolution is always greater than for the first resolution maneuver. One caveat here is that the assumption has been made that minimizing distance is more efficient. In the absence of wind gradients, this is true, but when wind gradients are present, the shortest distance between two points is not always the most efficient option for minimizing either flight time or fuel use.

The vertical profiles of each aircraft are assumed to have been provided as inputs to this optimization algorithm. Presumably, aircraft operators would compute optimal airspeeds and flight levels at which to operate their aircraft for the chosen flight route, but other profiles may be used, such as a fastest-speed profile. Many practical constraints may also be considered when choosing a vertical profile, such as regions of bad weather or turbulence. The important point is that the chosen airspeed and flight level for each aircraft are to be given as an input to the sequential optimization algorithm.

3.2.2 Optimal Wind Routing

The optimal wind routing function must take the desired vertical profile (speeds and altitudes) for each aircraft as inputs, and must return a complete 4-D trajectory. At this stage, the trajectory has not yet been checked for conflicts or for the satisfaction of other constraints. The accuracy and resolution of the output trajectory must be such that it can be adequately compared against other trajectories or airspace constraints. Approximate straight-line segments with assumed average speeds will not suffice. The quantitative computational requirements are derived in more detail later in this chapter, but it is already clear that the route-optimization function must be efficient because it is called on each pass through the loop. The computation of optimal routes is addressed in much more detail in chapter 4.

3.2.3 Conflict Detection

The conflict-detection function must accept a complete 4-D trajectory for the current aircraft and must determine whether or not the given trajectory is in conflict with any previously planned trajectories. A generalization of the conflict-detection function is to consider the satisfaction of additional constraints such as air-traffic control sector loading limits, special use airspace definitions, or regions of bad weather. Depending upon what kind of conflict-resolution function is used, the conflict-detection algorithm must return more or less information about conflicts. If a precise pairwise optimal conflict-resolution function is to be used, then the conflict-detection function must return detailed information about the location and geometry of each conflict, including information about the other aircraft or constraints that are involved. Multi-aircraft conflict-resolution schemes would require even more detailed information. Another approach is to just determine that a conflict is predicted without considering any of the specific details. This approach has the benefit of greatly simplifying the conflict-detection task and is the basic approach taken in this dissertation. The details are presented in chapter 5.

3.2.4 Conflict Resolution

The conflict-resolution function may be a sophisticated optimal-resolution algorithm, or a simple random-perturbation algorithm. A sophisticated algorithm might be able to resolve detected conflicts while minimizing added path length, or flight time, but these algorithms would require a significant amount of computational resources. A simpler approach is to request that the 4-D trajectory be randomly perturbed away from the potential conflict. This is justified based on the sparse-airspace assumption because one would expect to find many conflict-free routes near the optimum solution. A modified version of this approach is explored in this dissertation. Instead of making a completely random resolution perturbation, a perturbation is made in the general vicinity of the conflict such that the perturbed trajectory resolves the conflict while retaining wind optimality. This increases the chances of finding an efficient conflict-free resolution in a small number of iterations. More details are presented in chapter 5.

3.2.5 4-D Control

Once a conflict-free trajectory has been computed for an aircraft, a 4-D clearance may be sent to the aircraft. Depending on the recomputation rate of 4-D trajectories, it may be advantageous or even necessary for the aircraft to use closed-loop 4-D trajectory control (fig. 2.6) to mitigate the effects of small disturbances in between recomputation cycles. Many techniques have been explored for 4-D control and may be adapted for this purpose [30, 47-61]. However, 4-D control should be used only to the extent required to maintain conflict-free trajectories, because unnecessary use of 4-D control increases fuel use. This is intuitive since an aircraft flying at an optimal airspeed that uses airspeed perturbations to overcome disturbances can only increase its costs over the optimum. An analysis of the costs of 4-D control is now undertaken to quantify the additional cost of 4-D control. The goal is to show that 4-D control should be used as sparingly as possible. The following analysis leads to a relationship between the longitudinal position error variance of an aircraft using closed-loop 4-D control, and the increased fuel cost over the optimum result.

The longitudinal perturbation dynamic model of an aircraft traveling through a wind field with stochastic errors is given by

$$\underbrace{\begin{bmatrix} \Delta \dot{x} \\ \Delta \dot{u}_w \end{bmatrix}}_{\dot{\bar{x}} = F \bar{x} + L w} = \underbrace{\begin{bmatrix} -k & 1 \\ 0 & a_w \end{bmatrix}}_{F} \underbrace{\begin{bmatrix} \Delta x \\ \Delta u_w \end{bmatrix}}_{\bar{x}} + \underbrace{\begin{bmatrix} 0 \\ a_w \end{bmatrix}}_{L} w \quad \left\{ \begin{array}{l} w \sim N(0, 1) \end{array} \right. \quad (3.3)$$

where Δx is the longitudinal-position perturbation from the optimum position (from flying at the optimum airspeed), Δu_w is the perturbation in longitudinal winds from the predicted winds, k is a feedback gain parameter that may be adjusted to either tighten or loosen the 4-D control loop, a_w is the time constant of the wind-error model, and w is zero-mean Gaussian white noise that drives the wind-error model. The value of a_w based on analyses of Rapid Update Cycle (RUC) wind-model errors is approximately $a_w \approx (1/300) \text{ sec}^{-1}$. This corresponds to a time constant of 5 min., which is the approximate correlation constant of RUC wind model errors [62]. Because this time constant is so much longer than the speed control dynamics of the aircraft, the speed control dynamics for the aircraft can be safely neglected in this analysis.

For the system in equation (3.3), the state-error covariance matrix is given by

$$\dot{P} = FP + PF^T + LWL^T \quad (3.4)$$

where W is the error variance of the wind-model error. Substituting in the matrix values from equation (3.3) leads to the following expressions for the variance of the position

error, the covariance of the position error with the wind model error, and the wind-model error variance:

$$\begin{aligned} P_{\Delta x} &\equiv E[(\Delta x)^2] = \frac{W^2}{k(a_w + k)} \\ P_{\Delta x \Delta u_w} &\equiv E[(\Delta x)(\Delta u_w)] = \frac{W^2}{(a_w + k)} \\ P_{\Delta u_w} &\equiv E[(\Delta u_w)^2] = W^2 \end{aligned} \quad (3.5)$$

The fuel cost of an aircraft is simply the integral of the fuel flow rate, either specified as a function of time or of path distance:

$$J_{\text{fuel}} = \int_{t_0}^{t_f} \frac{\partial f}{\partial t} dt = \int_{s_0}^{s_f} \frac{\partial f}{\partial s} ds \quad (3.6)$$

where $\partial f / \partial t$ is the time rate of change of fuel weight, and $\partial f / \partial s$ is the equivalent path-distance rate of change of fuel weight. The path-distance expression is used in the remaining discussion.

The fuel-flow rate of an aircraft is a function of atmospheric parameters and aircraft speed, but for an analysis of perturbations near the optimum fuel flow rate for a given altitude and set of atmospheric conditions, the fuel-flow rate may be adequately modeled as a quadratic function of airspeed. The following is an approximate model for the Boeing 767 aircraft operating at optimum cruising altitude:

$$f_s \approx b_0 u_a^2 + b_1 u_a + b_2 \quad \begin{cases} b_0 = 4140 (\text{lb}/(\text{n.mi.}/\text{sec})^2) / \text{sec} \\ b_1 = -995.8 (\text{lb}/(\text{n.mi.}/\text{sec})) / \text{sec} \\ b_2 = 78.27 \text{ lb}/\text{sec} \end{cases} \quad (3.7)$$

where u_a is the airspeed of the aircraft in nautical miles per second, and f_s is the rate of change of fuel weight with path distance [30].

By taking the derivative of equation (3.7) with respect to u_a , setting to zero, and solving for the airspeed that gives minimum fuel-flow rate, the following result is obtained:

$$u_{\text{opt}} = \frac{-b_1}{2b_0} \quad (3.8)$$

The additional fuel-flow rate caused by flying at a non-optimum airspeed may be written as

$$\begin{aligned} \Delta f_s &= f_s - f_{s_{\text{opt}}} \\ &= b_0(u_{\text{opt}} + \Delta u_a)^2 + b_1(u_{\text{opt}} + \Delta u_a) + b_2 - [b_0 u_{\text{opt}}^2 + b_1 u_{\text{opt}} + b_2] \end{aligned} \quad (3.9)$$

This expression simplifies to

$$\Delta f_s = b_0 \Delta u_a^2 + 2b_0 u_{\text{opt}} \Delta u_a + b_1 \Delta u_a \quad (3.10)$$

The airspeed difference, Δu_a , is simply the feedback perturbation of the airspeed as shown in equation (3.3) to be given by

$$\Delta u_a \equiv -k \Delta x \quad (3.11)$$

This is a stochastic perturbation that is driven by the white noise in the wind-error model, also shown in equation (3.3), so the mathematical expectation must now be used to determine the expected additional cost. Introducing the expectation operator, substituting equation (3.11) into equation (3.10), and simplifying leads to

$$E[\Delta f_s] = -2b_0 k u_{\text{opt}} E[\Delta x] + b_0 k^2 E[\Delta x^2] - b_1 k E[\Delta x] \quad (3.12)$$

This expression is simplified further by noting that the average position error is zero, so that the expected additional fuel flow rate is given by

$$E[\Delta f_s] = b_0 k^2 P_{\Delta x} = \frac{b_0 k}{(k + a_w)} W^2 \quad (3.13)$$

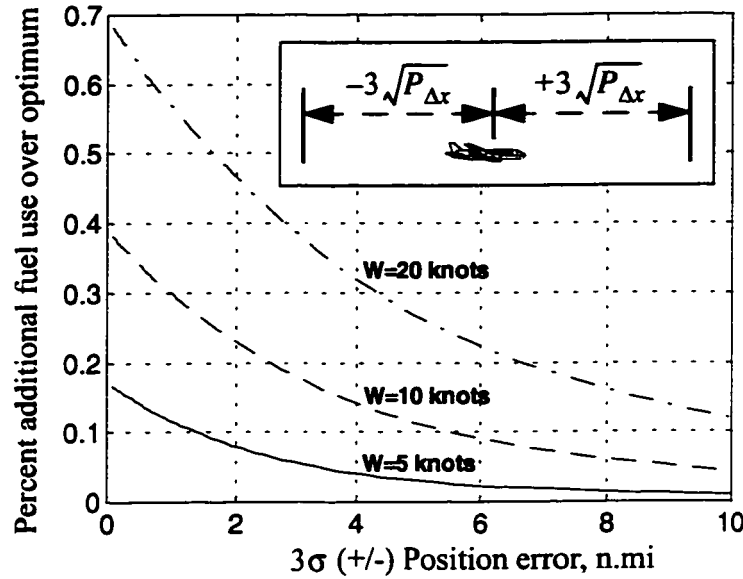


Figure 3.3. Plot of additional fuel use vs. regulated position error.

By solving equation (3.5) for k in terms of $P_{\Delta x}$ and substituting into equation (3.13), the following expression for the expected additional fuel-flow rate as a function of the position error variance is obtained:

$$E[\Delta f_s] = b_0 W^2 \left[\frac{\sqrt{a_w^2 + 4(W^2/P_{\Delta x})} - a_w}{\sqrt{a_w^2 + 4(W^2/P_{\Delta x})} + a_w} \right] \quad (3.14)$$

This relationship has been plotted for $a_w = (1/300) \text{ sec}^{-1}$ and three different values of W , the RMS wind-model error (fig. 3.3).

Two qualitative trends are apparent from the plot. The first is that reducing wind modeling errors, or other equivalent longitudinal position errors such as navigation and flight technical errors, will directly reduce costs. The second is that 4-D control clearly induces increased fuel costs.

Although the magnitude of the fuel cost increase is modest (of the order of a few tenths of a percent), even this is important to commercial aircraft operations where

notoriously tight profit margins cause airlines to increase efficiency wherever possible. In this case, where no additional computational effort is required on the part of the sequential optimization system to achieve 4-D control on the aircraft, and where the aircraft would be using some form of 4-D control anyway, it would make sense to use as efficient 4-D control as possible.

3.3 Computational Requirements of the Sequential Optimization Algorithm

The computational requirements of the sequential optimization algorithm are derived here. The results of this analysis establish the need for a more efficient optimal wind-routing algorithm.

Even without trajectory prediction uncertainty, the number of conflicts that may arise during the computations is not known *a priori* so that a probabilistic approach must be taken. This difficulty arises because of the NP-hard nature of the problem. As is to be shown, deterministic bounds for this problem are unrealistically conservative and not of much practical use. By following a probabilistic approach, one gives up the elegance of a deterministic solution or bound, but one gains a much more practical computation bound in the process. The probabilistic approach leads to a parametric expression for the expected number of computations. The parameters of the resulting expression may be adjusted to match observed data for any problem of interest. In addition to the computational estimate, a useful feature of the derived expression for the expected number of computations is that it also provides an estimate of the maximum airspace capacity.

In the discussion that follows, the term “computations” represents consistent units for expressing computational effort, such as floating point operations or computer clock cycles. These units are all interchangeable through proportionality constants.

The expected number of computations to arrive at a minimum-time sequential optimization solution for the i th aircraft, is given by

$$E[\xi_i] = E[R_i \cdot (\xi_{w0} + \xi_{cd})] \quad (3.15)$$

where $E[\]$ is the expectation operator, ξ_i is the total number of computations for aircraft i , R_i is the number of conflict-resolution iterations for aircraft i , ξ_{wo} is the number of computations required to compute a single wind-optimal route, and ξ_{cd} is the number of computations required to check a single route for conflicts. Since the number of conflicts that may arise is independent of the computations required to compute optimal routes or to detect conflicts, equation (3.15) becomes

$$E[\xi_i] = E[R_i] \cdot (E[\xi_{wo}] + E[\xi_{cd}]) \quad (3.16)$$

Similarly, the expected total number of computations over all N aircraft is given by

$$E\left[\sum_{i=1}^N \xi_i\right] = \left[\sum_{i=1}^N E[R_i]\right] \cdot (E[\xi_{wo}] + E[\xi_{cd}]) \quad (3.17)$$

As seen in equation (3.17), the expected numbers of computations to compute optimal routes or to detect conflicts are both multiplied by the total number of conflict-resolution iterations. At least one iteration is required for all aircraft, so that the total number of conflict-resolution iterations is greater than or equal to the number of aircraft:

$$\sum_{i=1}^N E[R_i] \geq N \quad (3.18)$$

One may theorize that the actual number of iterations will depend upon the following quantitative and qualitative parameters:

- Size of the airspace
- Structure of allowable aircraft routes
- Number and distribution of aircraft
- Specific restricted airspace owing to statutory regulations or bad weather
- Conflict-resolution algorithm details

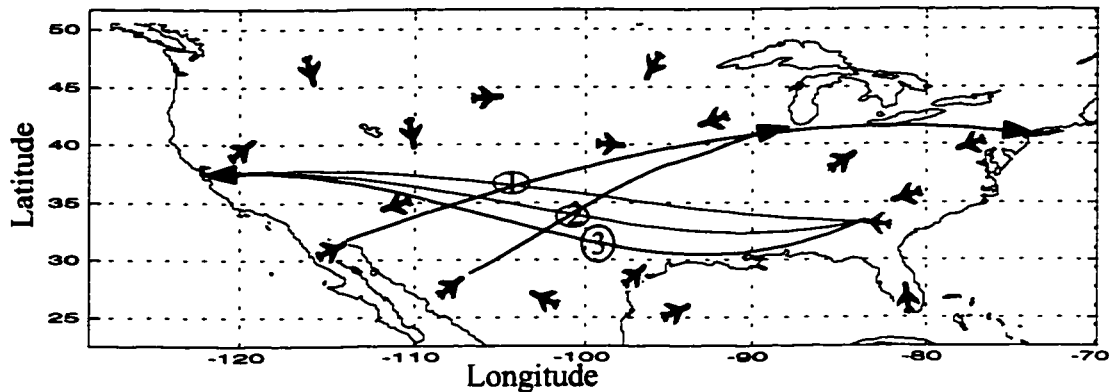


Figure 3.4. Conflict probability is hypothesized to be memoryless.

These parameters do not easily lead to a closed-form theoretical expression for the expected number of conflict-resolution iterations. Therefore a parametric model is derived which can be fitted to empirical data for different conditions.

3.3.1 Conflict Iteration and Capacity Model

The goal here is to derive a practical model to predict the computations required to obtain conflict-free trajectories for some given number of aircraft. The hypothesis is that for a sequential conflict-resolution strategy, it is equally likely at each iteration that another conflict may be encountered (fig. 3.4). This may be described as a *memoryless property*, and suggests the use of the geometric random variable (GRV) for the conflict iteration model, because the GRV is the only discrete random variable with the memoryless property.

If R_i is modeled as a GRV representing the number of iterations required to resolve all conflicts for the i th aircraft, where each resolution iteration is considered to be an

independent Bernoulli trial with probability P_i of being conflict-free, then the probability mass function (pmf) for R_i is given by

$$p_{ik} = P_i(1 - P_i)^{(k-1)} \begin{cases} i = 1, 2, \dots, N \\ k = 1, 2, \dots \end{cases} \quad (3.19)$$

where p_{ik} is the probability of resolving a conflict in k iterations for the i th aircraft. Typical values of P_i are close to unity so that the probability of finding a conflict-free solution during the first iteration is high, and the probability that a conflict-free trajectory will not be found until a later iteration decreases rapidly.

The expected value of the GRV, R_i , is

$$E[R_i] = \frac{1}{P_i} \quad (3.20)$$

As an extension to the standard GRV model, P_i is modeled as a function of the aircraft number. The reason for doing so is that the probability that a particular aircraft trajectory will be conflict-free decreases as the number of aircraft increases. The first aircraft will have a conflict-free trajectory with probability 1, while later aircraft will have increasing conflict probabilities.

In the interest of developing a simple model with a small number of parameters, a linear form for P_i is chosen as

$$P_i = \frac{(C_0 + 1)}{C_1} - \frac{1}{C_1}i \quad (3.21)$$

where C_0 and C_1 are parameters that are to be determined to best fit observed data. The form for the coefficients of P_i was chosen to simplify the final results.

Substituting equation (3.21) into equation (3.20) leads to

$$E[R_i] = \frac{C_1}{(C_0 + 1) - i} \quad (3.22)$$

As a practical model, equation (3.22) is not yet convenient. To generate enough data to curve-fit equation (3.22), one would have to perform multiple simulations or experiments to generate many data points at each value of i so that the expected number of resolution iterations could be determined to some degree of statistical significance. A curve-fit of these expected values as a function of i could then be used to determine C_0 and C_1 in a least square error sense.

A better approach is to derive an expression for the sum of equation (3.22). By doing so, only one simulation need be run while maintaining a running total of the number of conflict iterations. Each element of the sum is an independent measurement so that many independent measurements contribute to the sum as a function of the number of aircraft. A curve-fit of the summation function may then be used to obtain values for C_0 and C_1 .

The summation of equation (3.22) leads to the following analytical expression:

$$\begin{aligned} Y_N &\equiv \sum_{i=1}^N E[R_i] \\ Y_N &= \sum_{i=1}^N \frac{C_1}{(1 + C_0) - i} = C_1 \left(\Psi(C_0) - \Psi(C_0 - N) - \frac{N}{C_0(C_0 - N)} \right) \end{aligned} \quad (3.23)$$

where $\Psi(x)$ is the *digamma* function, defined as

$$\Psi(x) \equiv \frac{d}{dx} \ln(\Gamma(x)) \quad (3.24)$$

and the *gamma* function, $\Gamma(x)$, is defined as

$$\Gamma(x) \equiv \int_0^{\infty} t^{(x-1)} e^{-t} dt \quad (3.25)$$

Although equation (3.23) is quite compact, it would be inconvenient to leave the expression in this form. Routines for computing the *digamma* function are becoming more common, but the *digamma* function is not particularly well-known. In the region of interest for this problem, the *digamma* function is asymptotically close to the natural logarithm, $\ln(x)$. This leads to the following approximate form of equation (3.23):

$$\begin{aligned} Y_N \approx Y'_N &= C_1 \left(\ln(C_0) - \ln(C_0 - N) - \frac{N}{C_0(C_0 - N)} \right) \\ &= C_1 \ln \left(\frac{C_0}{C_0 - N} \right) - \frac{C_1 N}{C_0(C_0 - N)} \end{aligned} \quad (3.26)$$

A discussion of the physical nature of equation (3.26) is now presented, during which the second term will be shown to be negligible in the region of interest so that the expected total number of conflict iterations for a given number of aircraft is well approximated by

$$Y_N \approx \hat{Y}_N \equiv C_1 \ln \left(\frac{C_0}{C_0 - N} \right) \quad (3.27)$$

By examining equation (3.21), one can determine some properties of the conflict-iteration model parameters. The first aircraft will only require one resolution iteration (ignoring special-use airspace and weather cells for the moment), with probability 1, leading to the following relation:

$$P_1 = \frac{(C_0 + 1)}{C_1} - \frac{1}{C_1} = 1 \quad (3.28)$$

so one would expect

$$C_0 \approx C_1 \quad (3.29)$$

When curve-fitting actual data, the fits obtained using two parameters are much better than single-parameter curve-fits because the data are not absolutely random; as a result, both parameters are retained. However, the analysis above shows that one might expect the values for C_0 and C_1 to be numerically close. In real air-traffic data, there are often concentrations of aircraft traveling along the more popular city-pair routes at peak travel times so that the air traffic is not randomly distributed as presupposed by the GRV model assumption. Because of these unmodeled properties in the real air-traffic data, the GRV model is not a perfect fit, but it is close and leads to a practical method for estimating computation requirements for conflict resolution.

Restricting attention to optimization in the horizontal plane, for free-routed air traffic at the busiest flight levels over the continental United States (flight levels 330 and 350), C_0 and C_1 are both around 1,200 (e.g., fig. 6.8, chapter 6). The maximum number of aircraft found at these flight levels is about 500. For these parameter values, the two terms in equation (3.26) are given by

$$\begin{aligned} C_1 \ln\left(\frac{C_0}{C_0 - N}\right) &= 1200 \ln\left(\frac{1200}{1200 - 500}\right) = 646.8 \\ \frac{C_1 N}{C_0(C_0 - N)} &= \frac{1200 \cdot 500}{1200(1200 - 500)} = 0.71 \end{aligned} \quad (3.30)$$

The second term is negligible when compared to the first, and is always less than 0.015% of the expected total number of conflict iterations up to the discontinuity at $N = C_0 = 1,200$. The magnitude of the difference between the logarithm approximation (eq. (3.27)) and the exact version (eq. (3.23)) is less than 0.12% up to $N = 1,000$, and is always less than 4.1% up to the discontinuity (fig. 3.5).

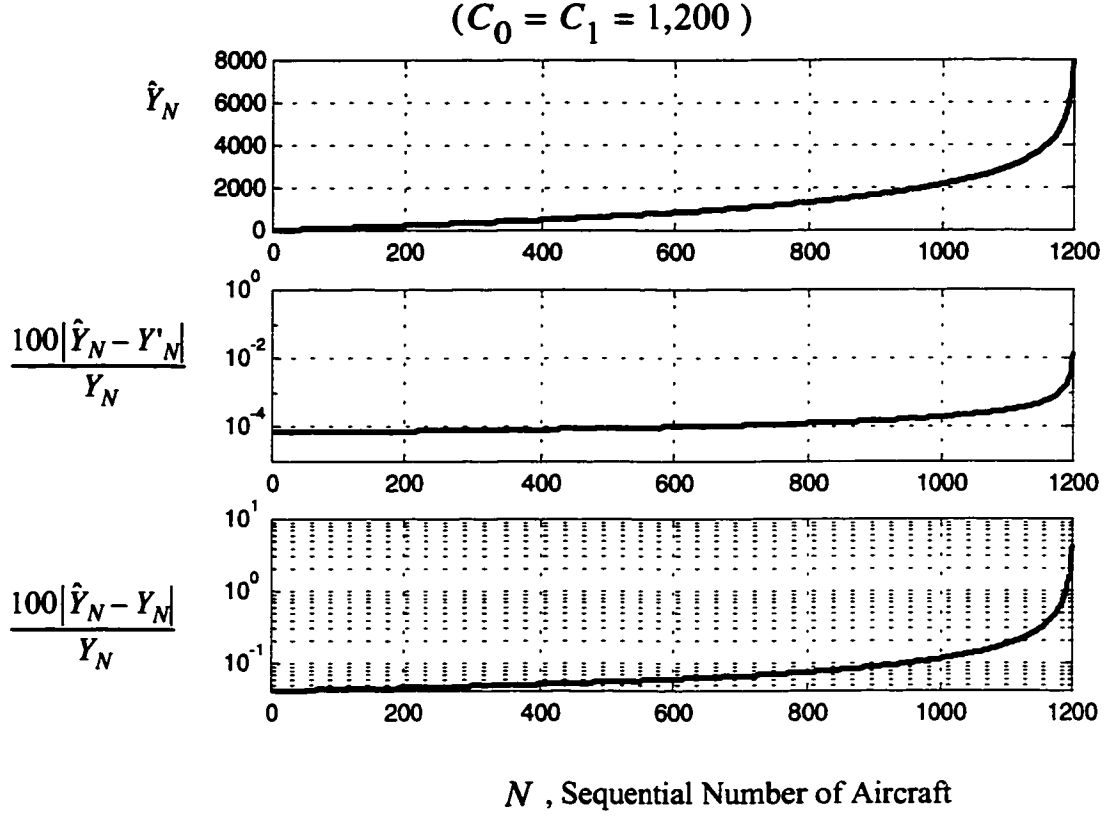


Figure 3.5. Expected No. of conflict iterations and approximation errors.

An important property of the derived conflict-iteration model is that there is a discontinuity when C_0 equals N . This discontinuity occurs in the same place in each of the various approximations of the model and is related to a physical result: it is not merely an abstract mathematical anomaly. It is instructive to consider the underlying reason for this discontinuity.

For a given situation, including the traffic patterns, the weather situation, and the chosen conflict-resolution algorithm, there is a theoretical maximum number of aircraft that can fit in the allotted airspace. Without any empirical data, one might try to determine a bound for the maximum number of aircraft by simply dividing the airspace area by the legally required area per aircraft. This is what has typically been done in the past. In en route airspace, aircraft must always maintain at least a 5-n.mi separation between

themselves and all other aircraft, so a reasonable approximation is to allow one aircraft per 25 square nautical miles. The continental United States covers approximately 3.8 million square nautical miles, so this simple bound would predict that 150,000 aircraft could fit at each flight level. To remain conflict-free, this would necessarily have to correspond to the situation where all aircraft were traveling in exactly the same direction at exactly the same speed (fig. 3.6). This is clearly not of much use in practice.

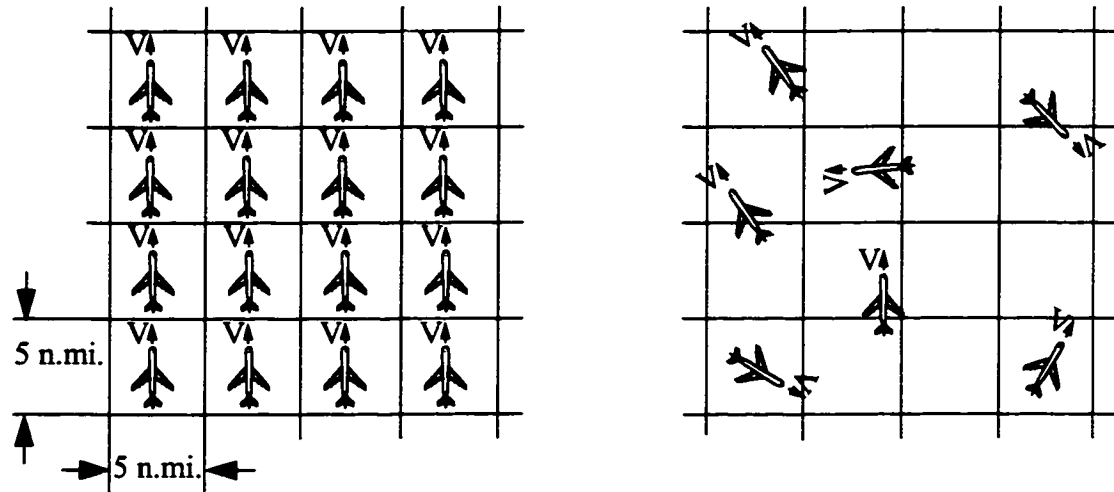


Figure 3.6. Two illustrations of airspace capacity limits.

When used to model a particular air-traffic system, the value of C_0 that is determined by curve-fitting actual data is a prediction of the maximum airspace capacity. This is an important result, because it provides, for the first time, a practical means of evaluating different conflict-resolution algorithms for their effect on airspace capacity. Much use of this capacity metric is made in this dissertation.

3.3.2 Extrapolating 2-D Solutions to the Full 3-D Problem

The complexity of the en route air-traffic-control optimization problem lies primarily in the area of horizontal route optimization and conflict resolution. Even though the vertical and speed profiles are computed separately in this concept, conflicts need to be checked over all altitudes. This adds a greater volume of airspace and a greater number of aircraft, but no inherent increase in solution complexity. Although possible on higher-end

current-day computer hardware, solving the air-traffic-control optimization problem for all of Class A airspace exceeds the amount of RAM available on the computer that has been used for algorithm development and simulation (256 MByte). This is largely because of the method used to achieve efficient conflict detection. Also, since MATLAB is used for algorithm development, simulation, and presentation of results, the processing speed is not as fast as would be the case in a compiled-code version of the algorithm. This is because MATLAB is partially an interpreted computer language. Because of the reduced processing speed, the solution for the typical numbers of aircraft in all of Class A airspace (3,000 - 5,000) would result in excessive simulation times. For these reasons, it is desirable during the algorithm development stage to address the reduced-scope problem of optimization in a horizontal plane and to relate the results of the reduced-scope problem to the solution of the full Class A problem.

As discussed in chapter 2, optimal vertical profiles consist of essentially constant altitude cruise segments at constant airspeed (fig. 2.7). This implies that most of the conflicts that can be expected will be between aircraft at the same altitude. Therefore, it is hypothesized that solution properties for all aircraft at a single flight level will scale linearly with the number of flight levels in Class A airspace.

The number of conflict iterations for the full Class A problem (3-D) is given by

$$Y_{3D} = \bar{C}_1 \ln \left(\frac{\bar{C}_0}{\bar{C}_0 - N_{3D}} \right) \quad (3.31)$$

where N_{3D} is a variable representing the number of aircraft for the 3-D Class A airspace problem. Both \bar{C}_0 and \bar{C}_1 are model parameters for the 3-D problem that are to be determined through simulation.

If the assumption is made that there are relatively few conflicts in en route airspace between climbing/descending aircraft and aircraft in level flight, the expected number of conflict iterations at any one flight level out of N_{FL} flight levels is then given by

$$Y_{2D} = Y_{3D}/N_{FL} \quad (3.32)$$

and the number of aircraft at any one flight level is given by

$$N = N_{3D}/N_{FL} \quad (3.33)$$

Substituting equations (3.32) and (3.33) into equation (3.31) and solving for Y_{2D} leads to the following expression for the expected number of conflicts at any one flight level:

$$Y_{2D} = C_1 \ln\left(\frac{C_0}{C_0 - N}\right) \quad (3.34)$$

where

$$C_0 \equiv \bar{C}_0/N_{FL} \quad (3.35)$$

and

$$C_1 \equiv \bar{C}_1/N_{FL} \quad (3.36)$$

Equation (3.34) has exactly the same form as equation (3.31) so that under the given assumptions, the number of aircraft at a single flight level is expected to follow the same functional form as for the full Class A airspace problem. The model parameters C_0 and C_1 are simply scaled by the number of flight levels.

This leads to a dramatic reduction in the effort required to obtain results that apply to the full Class A problem. In the common flight altitudes of Class A airspace (FL180 through FL390), there are 17 distinct flight levels at which up to 3,000 aircraft may be found at any instant in time (chapter 2). Instead of running simulations of 3,000 or more aircraft to determine the model parameters of equation (3.31), one may run much simpler

simulations of about 175 aircraft ($3000/17$) at constant altitude. Air traffic is not evenly distributed among the flight levels (fig. 2.2) because it is more efficient for most large jet aircraft to operate at about 35,000 ft. Therefore, constant-altitude simulations must use a few more aircraft than prescribed by equation (3.33) (about 400 is the maximum number of aircraft found at FL330 or FL350 at any instant in time). Simulating several times this value is easily achieved on the development platform, an IBM ThinkPad 750-MHz laptop with 256 MB RAM, running MATLAB. The values of C_0 and C_1 may be determined by curve-fitting the simulation data, and then equations (3.35) and (3.36) may be used to determine the equivalent values for the full Class A problem (e.g. fig. 6.8 & fig. 6.10, chapter 6). This is a significant reduction in the effort required to estimate these values.

3.4 Summary

Equation (3.17) showed that both the wind-optimal routing calculations and the conflict-detection calculations are computational primitives for this system concept. Both of these quantities are multiplied by the expected number of conflict iterations, which may be a large number. This requires that both the optimization and conflict-detection functions be made as efficient as possible so that the algorithm may be run in real-time.

A parametric model of the expected number of conflict iterations as a function of the number of aircraft has been derived based on a geometric random variable conflict model. The model has been derived for the general 3-D problem for all of Class A airspace, and then a similar model for the equivalent 2-D case was then derived. The equivalent 2-D model allows much simpler simulations to be conducted to obtain computational results for the full 3-D problem. This simplification is required to make algorithm development and simulation possible on a typical 750-MHz laptop computer.

Chapter 4

Optimal Wind Routing

4.1 Introduction

This chapter introduces the concept of computing minimum-time routes through wind fields, and a new solution technique called Neighboring Optimal Wind Routing (NOWR). Optimal wind-routing is a core function of the sequential optimization algorithm, and therefore must be computationally efficient (fig. 3.1, chap. 3). The NOWR technique is shown to have excellent optimization performance characteristics, and is computationally efficient when compared to existing algorithms. This makes NOWR ideally suited for computing conflict-free optimal routing solutions in real-time with the sequential optimization technique.

The reason that NOWR is so efficient is that it is a linear feedback algorithm. Perturbations in the winds *along the entire nominal trajectory* are fed back to perturb the aircraft heading such that the resulting trajectories are optimal to second order. Although this is mathematically no different than any other formulation of time-varying linear state feedback, NOWR is unusual because of the feedback of perturbations in the future wind states. The NOWR algorithm anticipates the winds that will be experienced at later stages in the flight and modifies the aircraft heading accordingly. This is made apparent through several illustrative examples in this chapter.

As noted in chapter 2, vertical-profile optimization and horizontal-route optimization are often decoupled. Although some optimization performance may be lost, the

computational simplification achieved is a significant benefit. The solution to the vertical-profile optimization problem with the goal of minimizing direct operating cost (DOC) yields a schedule of altitudes and airspeeds for a particular aircraft and particular route. The optimal vertical profiles for typical commercial jet aircraft are accurately characterized as constant-altitude or piecewise constant-altitude segments at constant or piecewise constant airspeed (fig. 2.7). The corresponding horizontal-route optimization problem becomes a minimum-time problem since the speed of the aircraft is fixed. In the absence of winds, or in a constant wind field, the optimum solution is a straight line between the origin and destination (a great-circle route on a spherical surface). In varying winds, the horizontal-route optimization problem is essentially that of finding minimum time routes through a varying wind field. This is referred to as an optimal wind-routing problem.

In regions of strong winds and horizontal wind shear, optimal wind-routes save an average of about 1.6% of flight time (or fuel burn) over the corresponding great-circle routes on long-distance flights (500 n.mi. or greater). An even greater savings, nearly 4.5%, can be achieved over the corresponding constrained flight-plan routes in use today. This is a reduction of about 500 hr per day in flight time, which translates to nearly \$1 million per day (\$360 million annually). Large savings on the order of 12% have been observed during strong wind conditions typical of the winter months. More is said about these statistics later in this chapter. In addition to direct cost savings, the air-traffic density would be reduced for a given number of aircraft by spreading aircraft over a greater volume of airspace than is possible in a structured routing system. This may enhance safety by lessening the chance for collisions. Because of the clear economic advantage and potential system benefits offered by optimal wind routes, it is not surprising that airlines attempt to operate aircraft along optimal wind routes as nearly as possible within the constraints of the current air-traffic control system. Airlines will continue to fly optimal wind routes as the air-traffic control system transitions to a more flexible system based on Free Flight concepts [21].

Although airlines currently compute approximate optimal wind routes for their aircraft, the techniques they use are not well-suited to real-time operation in the computation of conflict-free optimal wind paths for a large number of aircraft. As shown in chapter 3 (equation (3.17)), when computing conflict free optimal routes for a large number of aircraft with a sequential optimization approach, the wind-optimal computations must be efficient because they are repeated many times.

This chapter continues with a general discussion of the potential benefits of optimal wind routes. Next, a discussion of the computation of minimum-time paths for aircraft traveling at a constant speed and altitude through a wind field is presented. After introducing this problem and the prior art applied to its solution, the neighboring optimal control solution is presented. The neighboring optimal control solution is derived for the general case of an aircraft flying through a generally-varying wind field in spherical coordinates. The resulting algorithm is called Neighboring Optimal Wind Routing (NOWR).

A simulation study is conducted to evaluate the performance of NOWR. A discrete dynamic programming (DP) algorithm is used to compute true optimal routes to a fairly high degree of certainty. As with any directed graph optimization algorithm, there is a trade-off between optimization performance and computational speed. A comparison of NOWR and DP solutions is conducted to evaluate the performance/computation speed trade-off, and the results are presented.

One of the most important attributes of the NOWR algorithm developed in this dissertation is that it eliminates the difficulty of computing the nominal optimal trajectories that usually must be computed for neighboring optimal control. Because of this contribution, NOWR is an $O(N)$ algorithm, meaning that the computational effort required increases at a rate proportional to the length of the trajectory being computed. Actually, the normalized implementation of the NOWR algorithm has made it an $O(k)$ algorithm, where k is a constant. Since the trajectories are normalized, the amount of computation required to compute a neighboring optimal route between any two points is

independent of the trajectory length. All previous optimal wind-routing algorithms exhibit computational complexity of $O(N \log N)$ or higher.

4.2 Motivation: The Potential Benefits of Optimal Wind-Routing

The potential benefits of optimal wind-routing may be estimated by comparing flight times for optimal wind-routes with the corresponding times for filed flight-plan routes and great-circle routes. Obtaining complete data sets including winds, convective weather and turbulence, and aircraft schedule and track information is challenging because it requires vast amounts of computer memory and patience while awaiting the fulfillment of requests for data from various sources. Blending data from disparate sources is also a challenge. A comprehensive data analysis spanning over at least 1 year would be ideal, but this level of effort is beyond the scope of this dissertation. Instead, a limited analysis is conducted across a set of representative days so that the magnitude of potential benefits may be estimated. This demonstrates how more detailed data analyses might be conducted.

4.2.1 Optimal Wind Routes versus Great-circle Routes

A representative selection of 42 common long-range routes across the United States is examined for this study (fig. 4.1). The flight time along each of these routes is computed by integrating each trajectory through six different measured wind fields taken from RUC data (table 4.1) [63]. The RUC Gridded Binary (GRIB) files are processed using the WGRIB utility, version 1.7, the source code of which has been made freely available at several sites on the internet. The file names shown in the table are in a format found on the National Center for Environmental Prediction (NCEP) internet site (<http://www.nco.ncep.noaa.gov/pmb/products/ruc2/>). The file names describe the RUC model version (e.g. “ruc2”), the model analysis time in hours Universal Coordinated Time (UTC, or “Zulu”: e.g. “T21Z”), the GRIB format (e.g. “grb2”), and the forecast time in hours (e.g. “f02”). The wind data (u and v components) at the 225-mbar level are extracted and written out as new GRIB files. These files are then converted to an ASCII format suitable for MATLAB analysis. The 225-mbar level corresponds to approximately 36,000 ft MSL. More information about the RUC wind model may be found in appendix

2. Both the great-circle routes and the optimal wind-routes are computed. The optimal wind-routes are computed using the NOWR algorithm presented in more detail in this chapter.

The results are that, on average, optimal-route total flight times are about 1.6% less than those for the corresponding great-circle routes. The optimal wind-routes are geometrically quite different from the corresponding great-circle routes. In terms of airline cost savings, this seemingly small improvement of 1.6% is significant and demonstrates why airlines do not typically fly great-circle routes even today.

4.2.2 Optimal Wind Routes versus Filed Flight Plan

The comparison of optimal routes versus great-circle routes shows the benefit of minimizing flight time in winds over minimizing path distance. Another important result to determine is how much time would be saved by flying optimal wind-routes versus flying along the routes permitted in the NAS today. This demonstrates the benefit to be achieved by allowing aircraft to fly unconstrained optimal routes rather than the jet routes they are constrained to in the current air-traffic control system. When performing this

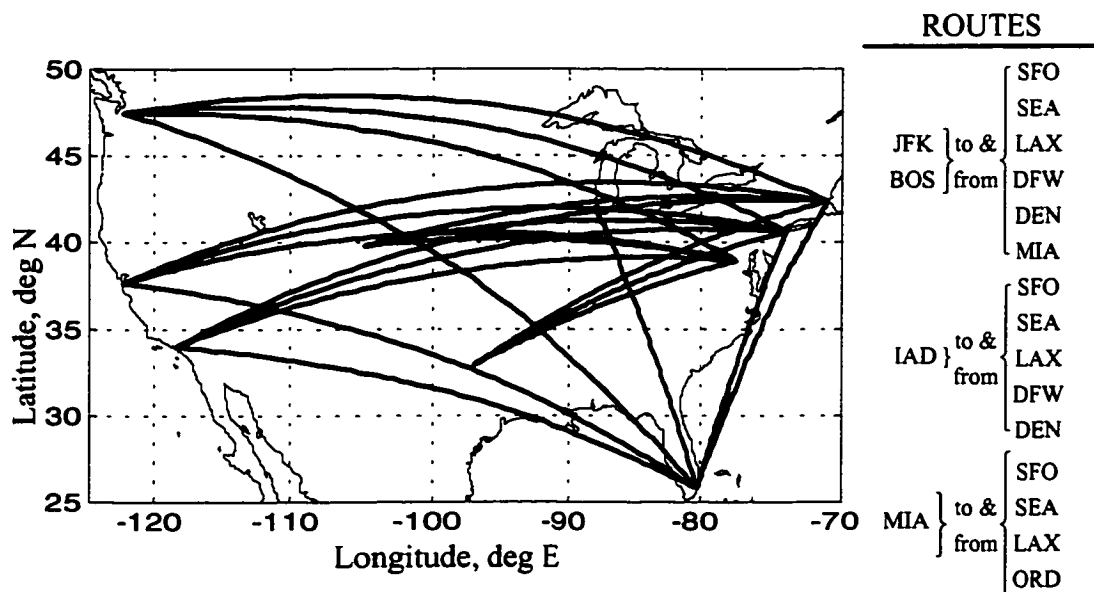


Figure 4.1. A representative set of long-range routes.

Table 4.1. RUC data files used in optimization performance analysis.

| Date (UTC) | Filename | Description |
|------------|-------------------|--|
| 2/14/2001 | ruc2.T21Z.grb2f02 | <ul style="list-style-type: none"> • RUC version 2 file • Vertical Coordinate: pressure level, millibars • GRIB Spec: Grid 211 (80 km & 25 mbar resolution) • Analysis time: 2100 universal coordinated time (UTC) • Altitude: Constant at 225 mbar pressure level (approximately 36,000 ft MSL)) |
| 2/11/2002 | ruc2.T19Z.grb2f02 | <ul style="list-style-type: none"> • Analysis time: 1900 UTC (other parameters same as previous) |
| 2/12/2002 | ruc2.T19Z.grb2f02 | (same as previous) |
| 2/13/2002 | ruc2.T07Z.grb2f02 | <ul style="list-style-type: none"> • Analysis time: 0700 UTC (other parameters same as previous) |
| 2/14/2002 | ruc2.T19Z.grb2f02 | <ul style="list-style-type: none"> • Analysis time: 1900 UTC (other parameters same as previous) |
| 2/20/2002 | ruc2.T21Z.grb2f02 | <ul style="list-style-type: none"> • Analysis time: 2100 UTC (other parameters same as previous) |

analysis, it is important to compare optimal wind-routes using the same wind data used by the airlines when they selected the flight plans.

For this study, the Future ATM Concepts Evaluation Tool (FACET) [44] is used to compute the flight times for a day's worth of real en route air traffic. The data are taken from archived Enhanced Traffic Management System (ETMS) files that span the period from midnight UTC on 12 February 2000 to midnight UTC on 13 February 2000. All flights at FL330 and FL350 are selected. This amounts to just over 4,000 aircraft for the chosen data set. The origin and destination airports, the filed airspeed, and the filed flight route are parsed from the ETMS data files and input into FACET for simulation. The corresponding RUC data files are obtained for use in the simulations as well. The FACET tool is used to integrate the trajectories of each aircraft along its respective flight plan and to compute the flight time of each aircraft and the sum total flight time for all aircraft. The

same data are used to drive a simulation in which each aircraft instead follows an optimal wind-route computed using the NOWR algorithm.

The total flight time savings over all aircraft is about 243 hr for the optimal wind-routes versus the filed flight-plan routes. This is a time savings of 4.5%. The cost to operate a commercial aircraft is of the order of \$30/min. Using this figure, the financial savings at these two flight levels would be about \$437,000 per day. This corresponds to just over \$100 per flight, which is roughly the cost of one revenue-paying passenger per aircraft.

These results indicate the financial benefit to be gained by permitting aircraft to fly along optimal wind-routes. The airline industry and air-traffic control research community have been aware of the potential benefits of optimal wind routes for some time now; as a result there has been a great deal of research regarding their accurate and efficient computation. Some of the key innovations and prior art in the field of optimal wind routing will now be discussed.

4.3 Prior Art: Optimal Aircraft Trajectories in Winds

Ernst Zermelo (1871 - 1953), a German mathematician, made important contributions to the calculus of variations, statistical mechanics, and later to set theory. In 1931, he studied the problem of determining minimum-time paths for an aircraft with constant altitude and airspeed traveling through a region of varying winds. This has become known as the *Zermelo problem*, or as *Zermelo's problem* (fig. 4.2) [36, 37]. Note that this is identical to the problem of determining minimum-time paths for a ship traveling through a region of varying currents [39].

The equations of motion for Zermelo's problem are given by

$$\dot{x} = V\cos\theta + u_w(x, y) \quad (4.1)$$

$$\dot{y} = V\sin\theta + v_w(x, y) \quad (4.2)$$

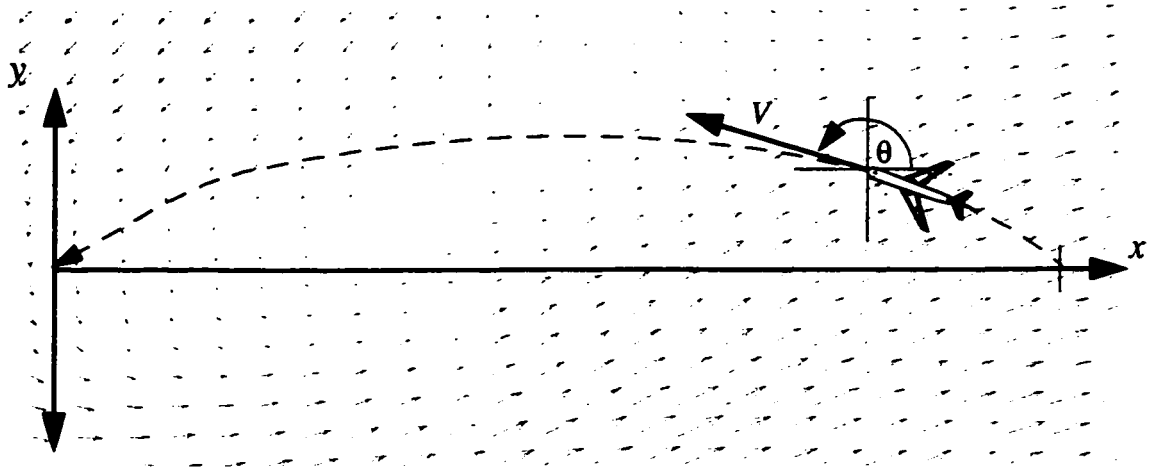


Figure 4.2. Zermelo's problem: minimum-time paths through varying winds.

where x and y are Cartesian position coordinates, and V is the airplane velocity relative to the air mass (the airspeed). The x -component of the wind velocity is $u_w(x, y)$, and the y -component is $v_w(x, y)$. The heading angle, θ , is the control available for achieving the minimum-time objective, with the airspeed usually being considered fixed. Some of the past approaches to solving this problem are now described.

4.3.1 Graphical Optimization

An interesting discussion of the early history of optimal ship routing, long before the aircraft had been invented, can be found in the introduction of reference [35]. Early attempts at computing optimal ship routes were manual methods based on the computation of time fronts, which are analogous to the wave fronts of geometrical optics. The first computer methods for optimal ship routing were then based on automating the manual methods.

4.3.2 Calculus of Variations

The calculus of variations was applied to the solution of the aircraft optimal wind routing problem by Zermelo in 1931 [36,37]. Zermelo derived a first-order differential equation describing the time rate-of-change of the optimal heading angle. The practical

computation of optimal trajectories would have to wait a few decades for the digital computer because the solution still required the numerical integration of three first-order nonlinear differential equations. The first two are the dynamic system equations (eqs. 4.1 & 4.2); the third equation to determine the optimal heading angle is given by

$$\dot{\theta} = v_x \sin^2 \theta + (u_x - v_y) \cos \theta \sin \theta - u_y \cos^2 \theta \quad (4.3)$$

where the subscripts x and y denote partial differentiation of the wind components with respect to the x and y coordinates. One common solution approach is to guess initial values of θ and then to iterate until the integrated solution trajectory passes through the desired final point. Families of solutions may be generated in this way.

Bryson and Ho [39] derived an implicit analytical solution for the case where the x -direction wind speed varies linearly in the y -direction (fig. 4.3). For $u_w(x, y) = -V_{ws}y$

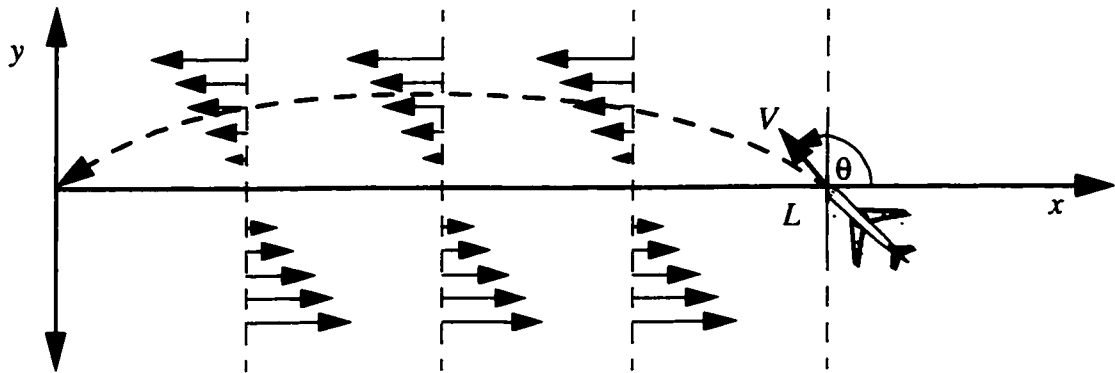


Figure 4.3. Zermelo's problem with linear wind shear in the cross-track direction.

and $v_w(x, y) = 0$, the heading, θ , at the point $\{x, y\}$, in order to go to the origin in minimum time, is computed implicitly from the following equations:

$$y = \frac{V}{V_{ws}}(\sec \theta - \sec \theta_f) \quad (4.4)$$

$$x = \frac{V}{2V_{ws}} \cdot [\operatorname{asinh}(\tan \theta_f) - \operatorname{asinh}(\tan \theta) + \tan \theta(\sec \theta_f - \sec \theta) - \sec \theta_f(\tan \theta_f - \tan \theta)] \quad (4.5)$$

where V_{ws} is the constant wind shear parameter and θ_f is the optimal heading angle at the final position. The time-to-go is given by

$$T = \left(\frac{1}{V_{ws}} \right) (\tan \theta_f - \tan \theta) \quad (4.6)$$

In addition to the inclusion of the V/V_{ws} terms in these expressions, several minor typographical errors in reference [39] have been corrected in equations (4.4) through (4.6).

4.3.3 Heuristic Dynamic Programming

Dynamic Programming (DP) is a directed graph optimization technique that is a straightforward approach to solving practical optimal routing problems (fig. 4.4) [39, 64]. In the basic DP approach, a discretized grid is overlaid on the domain of interest. The grid need not be uniform. The direction of travel is usually constrained to always flow from origin toward destination so that self-intersecting paths are eliminated from consideration. The flight time along each segment of the graph must be computed, which is what results in the $O(n^2)$ performance of the algorithm. The selection of the minimum-time (or minimum-cost) route may either be computed backward from the final point to the initial point, or forward from the initial point to the final point. The backward method results in a family of solutions from each grid point in the graph to the final point, while the forward method results in a family of solutions from the initial point to each grid point in the graph (including the final point). The winds may significantly vary over the time of an aircraft

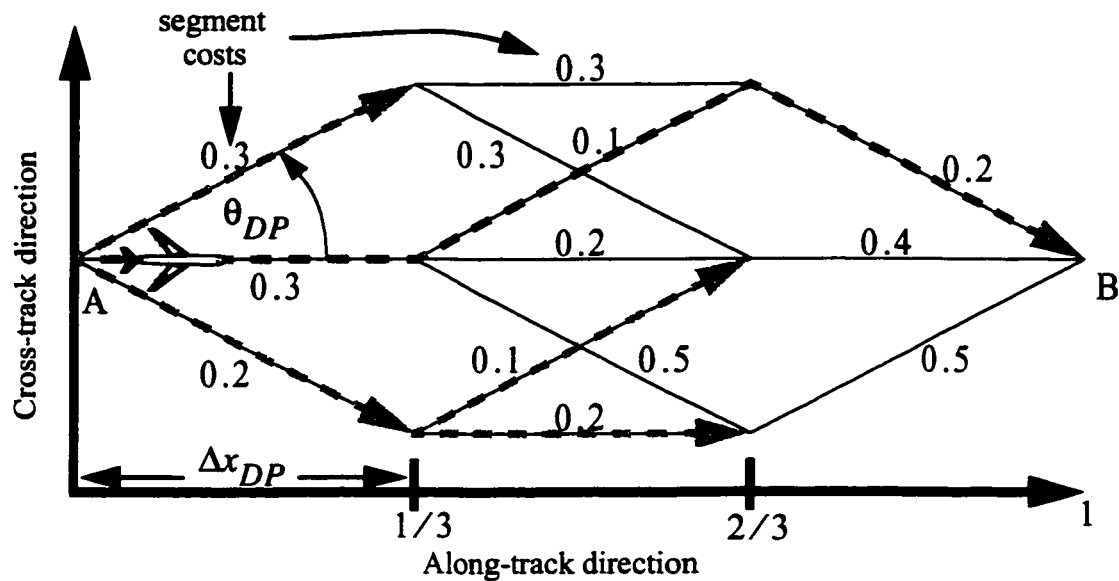


Figure 4.4. Discrete Dynamic Programming.

flight, so the forward method must be used for aircraft route optimization. This results from the fact that one must know the time at which the aircraft will be at each segment so that the proper winds may be selected from forecast data.

Although DP suffers from $O(n^2)$ complexity, it is well suited to aircraft trajectory-optimization problems because constraints that are difficult to quantify may be easily incorporated into the solution. The DP technique is well understood, but practical implementation details of DP for aircraft route optimization are often proprietary so that documentation is not readily available. Practical route-optimization software systems were developed in the late 1960s and early 1970s by R. Dixon Speas of R. Dixon Speas Associates, and by a group at Lockheed headed by Reinkins [43]. The group at Lockheed was later sold and became a part of Jeppesen Dataplan.

The Lockheed group found that straight DP was too computationally intensive for then-available computer hardware (IBM 360) so that heuristics were needed to reduce computational effort to manageable levels. The adopted procedure was to optimize the horizontal route over a discrete grid within a reduced elliptical domain around the great-circle route. Analysis showed that this approach was 99.9% accurate in locating the

optimum route while greatly reducing computational effort. After the horizontal route was determined, a constrained optimum vertical/speed profile was determined by searching over a few candidate cruise altitudes. Climbs were only permitted over nav aids to further reduce the search space. It was found that higher altitudes were nearly always more efficient for commercial jet traffic, regardless of the vertical wind shear conditions.

The implementation of Lockheed's optimization system used digitized aircraft performance data and tabular atmospheric data from the National Weather Service. The atmospheric data were in Aeronautical Radio, Inc. (ARINC) format, were updated twice daily, and were presented in 6-hr increments up to a 30-hr forecast. Analysis of their system demonstrated that aircraft range could be greatly extended through route-optimization. One interesting application of the route-optimization algorithms was to aid in-flight refueling operations during the Falkland Islands crisis in 1982.

The Lockheed system has evolved into a number of commercial products and services offered by Jeppesen Dataplan [41], and similar proprietary systems are in use by a number of different airlines and large corporate fleet operators. These systems have been manually tuned to work within the constraints of the NAS to produce near-optimum flight plans, even considering such constraints as the fees incurred by crossing into different international airspace domains. The computation speed is difficult to assess and to compare with other computational approaches to optimization because the details are proprietary, but the fact is that any DP system will exhibit $O(n \log n)$ or $O(n^2)$ computational behavior. This can be improved through heuristics, but there will always be a trade-off between computational efficiency and optimization performance.

A DP-like system called the Worldwide Aeronautical Route Planner (WARP), based on a search algorithm called *Multi-Pass A**, was reported to have relatively good computational and optimization properties, though detailed quantitative results were not made available [42]. It was reported that the WARP system produced optimal routes with fuel savings of the order of 1% to 8% compared to great-circle routes, and that these routes took approximately 30 sec to compute. This is many orders of magnitude slower than what

is required for a system that must compute conflict-free optimal routes for thousands of aircraft in less than a minute or so.

Later in this chapter, a computational and performance comparison of a particular DP algorithm and the neighboring optimal wind-routing algorithm is presented. The analysis suggests that DP requires approximately an order of magnitude more computational effort than the neighboring optimal scheme to achieve the same performance. An additional drawback of DP solutions is that they are not easy to perturb in order to find nearby trajectories when trajectory conflicts must be resolved.

4.3.4 Genetic Algorithms

Genetic Algorithms (GAs) and other randomized search techniques have been the subject of much research over the past decade for their potential to obtain feasible near-optimum solutions to NP-hard problems. The essence of GAs is to randomly perturb trajectory parameters, to evaluate each solution for constraint satisfaction and optimization performance, and then to select a subset of the random solutions at each iteration with the best performance. Heuristics are used to occasionally select solutions that are different from the current best solutions to avoid locally optimal solutions.

In reference [9], a GA approach is used to determine single-aircraft paths that minimize some specified index. The algorithm depends on ATC first computing and communicating all regions of unavailable airspace. Using sparse structured airspace, real-time performance was achieved from the perspective of the needs of a single aircraft, but the results were considerably sub-optimal. Using what was referred to as a time-slice approach, much better optimal paths were achieved, but not in real-time. Conflicts among aircraft do not appear to be directly accounted for in the optimization, other than if potential conflicts are identified by a centralized air-traffic control system and provided to the GA system. Winds were modeled as constant.

Other applications of GAs to air-traffic-control optimization have concentrated primarily on the conflict resolution function and have not directly included variable winds

[45, 65, 8]. In principle, the inclusion of variable winds should not greatly affect the performance of GAs.

Genetic Algorithms still suffer from the same $O(n^2)$ type of performance as DP, but the proportionality constants are potentially much lower for GA so that near-optimum results may be obtained quickly. Among the problems with GAs is that there are no guarantees on the optimality of the solution, and that the solutions are so coarse that once a solution is found, a more detailed optimum result still must be computed, usually using a gradient optimization algorithm.

4.3.5 Effect of Winds on Optimal Vertical Profiles

The effect of winds on optimal vertical and speed profiles has received considerable attention because winds can have considerable effect on optimal trajectory solutions [30, 31, 32]. In this case, the goal is to find the optimum combinations of speed and altitude to minimize the direct operating cost, which is a combination of fuel and time costs (eq. (2.10)). This is in contrast to the horizontal-plane optimization problem where speed is considered fixed.

The studies that have been conducted examine the general effects of winds on the optimal trajectory solutions. Winds are shown to have a measurable effect on optimal-speed profiles, indicating that it is important to account for the changing nature of the winds. Since wind forecasting is an inexact science, the implication is that one should recompute optimal speed and altitude profiles regularly as wind forecasts change. This is one of the primary reasons why optimal conflict-free trajectories must be recomputed often, to allow flexibility as wind conditions change.

4.3.6 Effect of Constant Winds on Constrained Horizontal Routes

Minimum-time solutions to the horizontal-trajectory optimization problem in zero winds with hard state constraints have been shown to consist of straight-line segments connected by minimum radius turns [38]. The optimum zero-wind solutions are

characterized as bang-singular-bang solutions, and have typically been derived based on the minimum principle.

A parameterization technique was applied to the trajectory optimization in the horizontal plane in the presence of winds with fixed airspeed [66]. Even a constant wind field prevents the derivation of a closed-form solution to the problem. Results are specific to each wind scenario and to each set of initial- and final-state constraints, so that generalizations cannot be made. The main conclusion is that optimum trajectories in the horizontal plane in the presence of winds are qualitatively different from the zero-wind solutions, and sometimes dramatically so.

4.3.7 Neighboring Optimal Control: A Brief Introduction

Neighboring optimal control (NOC) is a time-varying linear feedback control algorithm that minimizes a performance index to second order by regulating small perturbations around a nominal optimal trajectory. NOC has been successfully applied to the solution of a simple case of the Zermelo problem where the winds are modeled as varying linearly in the cross-track direction only.

The NOC regulator is closely related to the better known Linear Quadratic (LQ) regulator. In LQ design, the control engineer adjusts terms in the cost-function (J) weighting matrices (Q , R , and N) to achieve good regulator performance:

$$\Delta J = \int_0^\infty \left(\begin{bmatrix} \Delta \bar{x}^T & \Delta \bar{u}^T \end{bmatrix} \begin{bmatrix} Q & N \\ N^T & R \end{bmatrix} \begin{bmatrix} \Delta \bar{x} \\ \Delta \bar{u} \end{bmatrix} \right) dt \quad (4.7)$$

Adjusting these weighting matrices achieves a balance among the state perturbations and the control perturbations used to eliminate them. The weighting matrices may be constants or may be time-varying as required by the particular system being regulated. Constant weighting matrices are used in most cases for aircraft control. A time-varying gain gives better results than the constant gain LQ regulator solution.

The difference between NOC and LQ is that in NOC, the weighting matrices are elements of the Hessian matrix of an optimizing cost-function as follows:

$$\Delta J = \int_0^{t_f} \left(\begin{bmatrix} \Delta \bar{x}^T & \Delta \bar{u}^T \end{bmatrix} \begin{bmatrix} H_{xx}(t) & H_{xu}(t) \\ H_{ux}(t) & H_{uu}(t) \end{bmatrix} \begin{bmatrix} \Delta \bar{x} \\ \Delta \bar{u} \end{bmatrix} \right) dt \quad (4.8)$$

where H is the Hamiltonian of the dynamic system being optimized. The subscripts x and u denote partial differentiation with respect to the states and controls, respectively. In this case, the weighting matrices are time-varying (shown explicitly in eq. (4.8)), so the resulting optimal feedback gains are also time-varying. These feedback gains may be tabulated as functions of time-to-go, and the perturbation in the final time may be estimated so that an NOC regulator can be used to achieve minimum-time solutions.

The details of computing gains for NOC are presented in references [39] and [64]. More details on the computation of NOC are also presented later in this chapter. The important result with NOC is that it is a time-varying linear feedback algorithm where the perturbation controller has the following form:

$$\delta u(T) = -K_{ux}(T) \delta x(T). \quad (4.9)$$

Here, $\delta u(T)$ is the perturbation from the nominal control vector, $\delta x(T)$ is the perturbation from the nominal state vector, $K_{ux}(T)$ is the neighboring optimal feedback gain matrix, and T is the time-to-go on the nominal optimal path. The appropriate nominal states, controls, and feedback gains are selected based on finding the same time-to-go on the actual path and the nominal path by using the following NOC feedback relation to update the optimal final time, t_f :

$$dt_f = -K_{tx}(T) \delta x(T) \quad (4.10)$$

Neighboring optimal controls are generated by feed-forward of the nominal optimal controls and linear feedback of perturbations from the nominal optimal state trajectory. This should not be mistaken for simply regulating a system to follow a nominal trajectory.

Use of the NOC gains will minimize the original cost function to second order. For most problems, nominal optimal solutions and NOC feedback gains must be computed numerically because complexity precludes analytical solutions.

4.3.8 NOC Solution to the Zermelo Problem for Constant Wind Shear

The NOC solution for the Zermelo problem with a constant cross-track wind shear has been computed analytically [39, 40]. A numerical solution to this problem has also been derived and compared with the analytical solution [40]. This solution is for the general case of perturbations about any nominal trajectory.

The perturbation feedback laws for θ and θ_f are given by

$$\begin{bmatrix} \delta\theta \\ \delta\theta_f \end{bmatrix} = \begin{bmatrix} \frac{\partial\theta}{\partial x} & \frac{\partial\theta}{\partial y} \\ \frac{\partial\theta_f}{\partial x} & \frac{\partial\theta_f}{\partial y} \end{bmatrix} \begin{bmatrix} \delta x \\ \delta y \end{bmatrix} \quad (4.11)$$

where the perturbation of any variable, A , is defined by $\delta(A) \equiv (A - A_{\text{nominal}})$, and the partial derivatives of θ and θ_f are determined by inverting equation (4.11):

$$\begin{bmatrix} \frac{\partial\theta}{\partial x} & \frac{\partial\theta}{\partial y} \\ \frac{\partial\theta_f}{\partial x} & \frac{\partial\theta_f}{\partial y} \end{bmatrix} = \frac{1}{\frac{\partial x}{\partial\theta} \cdot \frac{\partial y}{\partial\theta_f} - \frac{\partial x}{\partial\theta_f} \cdot \frac{\partial y}{\partial\theta}} \begin{bmatrix} \frac{\partial y}{\partial\theta_f} & -\frac{\partial x}{\partial\theta_f} \\ -\frac{\partial y}{\partial\theta} & \frac{\partial x}{\partial\theta} \end{bmatrix} \quad (4.12)$$

The partial derivatives of x and y with respect to θ and θ_f are computed from the implicit analytical solution for θ in equations (4.4) through (4.6) and are given by

$$\frac{\partial x}{\partial \theta} = \frac{V[2\sec^2\theta(\sec\theta_f - \sec\theta) + \sec\theta - |\sec\theta|]}{2V_{ws}} \quad (4.13)$$

$$\frac{\partial y}{\partial \theta} = \frac{V}{V_{ws}} \sec\theta \tan\theta \quad (4.14)$$

$$\frac{\partial x}{\partial \theta_f} = \frac{1}{2} \frac{V}{V_{ws}} [2\sec\theta_f(\tan\theta_f \tan\theta - \sec^2\theta_f) + \sec\theta_f + |\sec\theta_f|] \quad (4.15)$$

and

$$\frac{\partial y}{\partial \theta_f} = -\frac{V}{V_{ws}} \sec\theta_f \tan\theta_f \quad (4.16)$$

The reference point is determined by the point with the same time-to-go as the point on the nominal path. Differential changes in T are given by

$$\begin{aligned} dT &= \frac{\partial T}{\partial x} \delta x + \frac{\partial T}{\partial y} \delta y \\ &= \left(\frac{\partial T}{\partial \theta} \cdot \frac{\partial \theta}{\partial x} + \frac{\partial T}{\partial \theta_f} \cdot \frac{\partial \theta_f}{\partial x} \right) \delta x + \left(\frac{\partial T}{\partial \theta} \cdot \frac{\partial \theta}{\partial y} + \frac{\partial T}{\partial \theta_f} \cdot \frac{\partial \theta_f}{\partial y} \right) \delta y \end{aligned} \quad (4.17)$$

where

$$\frac{\partial T}{\partial \theta} = -\frac{\sec^2\theta}{V_{ws}} \quad (4.18)$$

$$\frac{\partial T}{\partial \theta_f} = \frac{\sec^2\theta_f}{V_{ws}} \quad (4.19)$$

When given the current coordinates, $\{x, y\}$, the appropriate value of T is found by solving equation (4.17) for $dT = 0$. By doing so, the location on the actual path with the

same time-to-go as the corresponding point on the nominal path is determined. With this value of T , the appropriate values of the feedback gains and nominal controls are obtained from lookup tables. The optimal heading is computed as the sum of the nominal optimal heading and the neighboring optimal perturbation:

$$\theta = \theta(T)_{\text{nom}} + \frac{\partial}{\partial x}\theta(T)(x - x_{\text{nom}}(T)) + \frac{\partial}{\partial y}\theta(T)(y - y_{\text{nom}}(T)) \quad (4.20)$$

This NOC solution to the Zermelo problem is intended to be used to regulate small position perturbations away from a nominal optimal route for some fixed value of cross-track wind shear, V_{ws} . The solution procedure is to set V_{ws} , compute the nominal optimal route, and then use the NOC regulator to find neighboring solutions when the aircraft position deviates from the nominal route. This is a computationally efficient method for computing neighboring optimal solutions. The NOC approach works well when the wind shear really is constant as modeled, but when the wind shear changes, one must recompute a new nominal optimal trajectory or else severely sub-optimal performance may result. For this reason, a more flexible solution technique is required. In addition to accounting for perturbations in the wind shear, it would be desirable to allow more general wind models than just a simple linear wind shear. A more realistic wind model is needed. These shortcomings led to the development of a more general NOC solution in this dissertation called Neighboring Optimal Wind Routing (NOWR). In NOWR, the wind field is modeled by piecewise-linearly-varying functions of position, and the dynamic system equations are augmented with new states for the various wind terms. The solution to the augmented problem produces feedback terms for perturbations in the winds.

4.4 Neighboring Optimal Wind Routing

The NOWR algorithm is an application of NOC to the problem of optimizing aircraft trajectories in a general wind field. It includes the required coordinate rotations, normalizations, and other augmentations required for application to real aircraft trajectory optimization problems. An important extension over prior applications of NOC to the optimal wind-routing problem is the modeling of the wind field by piecewise-linear

functions. The system dynamic model is augmented with bias states for the wind-model parameter values at a set of measurement points. This allows the winds to be nominally set to zero so that the nominal optimal solution is simply a straight line (or great circle on the sphere) between origin and destination. From the NOC solution, perturbations in the winds along the route are fed back to modify the aircraft heading to produce near-optimal trajectories. This leads to a practical and efficient means of computing near-optimal routes in general wind fields using linear feedback.

The NOWR algorithm is a time-varying linear perturbation feedback algorithm, but some nonlinear elements such as coordinate rotations to spherical-Earth coordinates must be included for practical use. NOWR is similar to other perturbation feedback schemes in which nominal controls are input to the dynamic system being controlled, and then perturbations from the nominal state, including wind perturbations, are multiplied by neighboring optimal feedback gains to modify the nominal aircraft heading (fig. 4.5). The

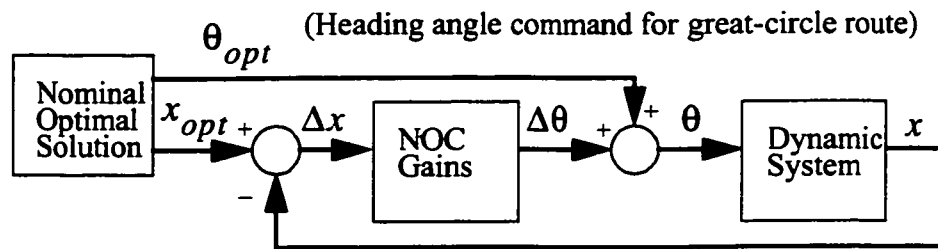


Figure 4.5. The NOC perturbation feedback algorithm.

difference between NOWR and other perturbation feedback schemes is that the use of the NOC gains results in a near minimum-time path for the given wind conditions.

The following sections present the derivation of the NOWR algorithm, including dynamic modeling, coordinate transformations, wind modeling, and solution for the neighboring optimal control feedback gains. This is followed by a short discussion of the benefits and possible enhancements of NOWR. The outline of the derivations is as follows:

- **Dynamic Modeling:** Definition of the point-mass kinematic model of an aircraft in spherical-Earth coordinates
- **Coordinate Rotation:** Rotation of coordinates to one in which linearization is possible
- **Wind Vector Rotation:** Rotation of a vector from one spherical-Earth coordinate system to another
- **Nondimensionalization of Equations of Motion**
- **General Wind Field Modeling:** Modeling the wind field with piecewise-linear functions
- **Analytical Neighboring Optimal Control Solution**

4.4.1 Dynamic Modeling

A mathematical model of the real system (an aircraft in a wind field in this case) is presented. Although the model exhibits enough fidelity and complexity for accurate trajectory optimization, it is not any more complex than required.

A nonlinear kinematic model of the motion of an aircraft at a constant airspeed and altitude in a spherical-earth latitude and longitude coordinate system is given by

$$\frac{d\tau}{dt} = \frac{V \cos \theta + u_w(\tau, \lambda)}{R \cos \lambda} \quad (4.21)$$

$$\frac{d\lambda}{dt} = \frac{V \sin \theta + v_w(\tau, \lambda)}{R} \quad (4.22)$$

where τ is the longitude, λ is the latitude, V is the constant airspeed, R is the distance from the center of Earth to the current aircraft position, and $\{u_w, v_w\}$ are the wind vector components in the east and north directions, respectively.

The dynamic model is nonlinear through the $\cos \lambda$ term in the denominator of equation (4.21) and also through the potentially nonlinear variation of the winds with position. The next step is to perform a coordinate rotation that eliminates the $\cos \lambda$ term. The normalization can be achieved so that ultimately a single solution can be applied to many different optimal-routing problems. Development of the mathematical model of the winds is deferred until after linearization and normalization of the equations of motion.

4.4.2 Coordinate Rotation

The equations of motion, (4.21) and (4.22), are nonlinearly coupled by the $\cos(\lambda)$ term in the denominator of the longitude rate equation. By judicious choice of coordinates, this term can be neglected for small perturbations from a nominal route so that the equations of motion may be linearized and decoupled from one another. This is achieved through a coordinate rotation that places any desired great-circle route along the equator of a rotated coordinate system (fig. 4.6). Along with the linearization, a normalization of

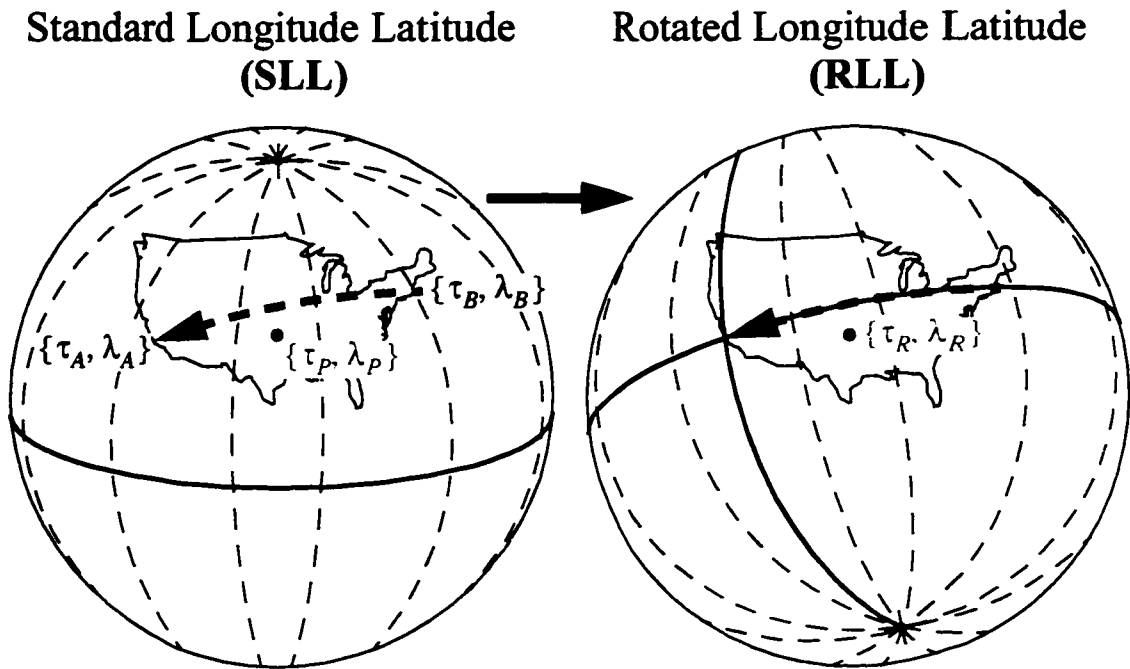


Figure 4.6. Coordinate rotation from SLL to RLL coordinates.

coordinates may be performed which leads to a great simplification in the application NOC for optimal wind routing. More is said on this later.

The spherical-Earth latitude and longitude coordinates in common use, with the equatorial plane normal to Earth's rotation axis and the prime meridian passing through Greenwich, are referred to here as Standard Longitude and Latitude (SLL) coordinates. The rotated coordinates are referred to as Rotated Longitude and Latitude (RLL) coordinates.

The derivation of the following rotations is achieved by equating sets of Euler angle rotations in both coordinate systems. Other relations or simplifications of these expressions may be possible, but these have been demonstrated to work well.

The rotation from one set of latitude/longitude coordinates to another is achieved as follows:

1. Compute ξ , an intermediate rotation angle:

$$\xi = \text{atan2}(-[\cos\lambda_A \sin\lambda_B - \sin\lambda_A \cos\lambda_B \cos(\tau_A - \tau_B)], -[\sin(\tau_A - \tau_B) \cos\lambda_B]) \quad (4.23)$$

where $\{\tau_A, \lambda_A\}$ are the longitude and latitude coordinates of the final point of travel in SLL coordinates, and $\{\tau_B, \lambda_B\}$ are the longitude and latitude of the initial point of travel in SLL coordinates (fig. 4.6).

2. Calculate additional intermediate variables:

$$\begin{aligned} \phi_1 &\equiv \sin(\tau_A - \tau_P) \cos\lambda_P \\ \phi_2 &\equiv \cos\lambda_A \sin\lambda_P - \sin\lambda_A \cos(\tau_A - \tau_P) \cos\lambda_P \\ x_{PA} &\equiv -(\phi_1 \cos\alpha + \phi_2 \sin\alpha) \\ y_{PA} &\equiv -(-\phi_1 \sin\alpha + \phi_2 \cos\alpha) \\ z_{PA} &\equiv -(\sin\lambda_A \sin\lambda_P + \cos\lambda_A \cos(\tau_A - \tau_P) \cos\lambda_P) \end{aligned} \quad (4.24)$$

where $\{\tau_P, \lambda_P\}$ are the longitude and latitude in SLL coordinates of the point for which RLL coordinates are being computed.

3. Compute the RLL longitude and latitude coordinates from the intermediate variables:

$$\begin{aligned} \lambda_R &= \text{asin}(-y_{PA}) \\ \tau_R &= \text{atan2}(x_{PA}, -z_{PA}) \end{aligned} \quad (4.25)$$

With this rotation, and with the assumption that perturbations from the nominal great-circle path will be small, the equations of motion, (4.21) and (4.22), become

$$\left. \begin{aligned} \frac{dx_r}{dt} &= V \cos \theta + u_{wr} \\ \frac{dy_r}{dt} &= V \sin \theta + v_{wr} \end{aligned} \right\} \begin{aligned} x_r &\equiv R \tau_R \\ y_r &\equiv R \lambda_R \end{aligned} \quad (4.26)$$

The rotated wind vector components, $\{u_{wr}, v_{wr}\}$, must be obtained by rotating the wind vector from the SLL system to the RLL system, the derivation of which is now presented.

4.4.3 Wind Vector Rotation

The wind components are typically resolved along the SLL east and north directions, respectively, so they must be rotated to be along the x_r and y_r directions in the RLL system. This rotation is nontrivial since the rotation angle is a function of latitude and longitude.

The goal of the following derivation is to rotate an angle, ψ_{SLL} , which is known at the point $\{\tau_p, \lambda_p\}$ in the SLL system, to the equivalent angle, ψ_{RLL} , in the RLL system. In this case, the angle to be rotated is given by

$$\psi_{SLL} = \text{atan2}(v_w, u_w) \quad (4.27)$$

where u_w and v_w are the wind-vector components in the SLL coordinate system.

The first step is to convert the coordinates of the base of the wind vector from SLL to RLL coordinates. This is exactly the same conversion as given by equations (4.23) through (4.25), and the same variable definitions are used here.

A direction cosine matrix (DCM) has been derived from a series of Euler angle rotations to perform the rest of the desired angle conversion. The DCM is given by

$$C = \begin{bmatrix} \left(\frac{-z_{PA}a_2 - x_{PA} \cos(\lambda_{0R}) \sin(\tau_{0R} - \tau_P)}{\sqrt{1 - y_{PA}^2}} \right) & \left(\frac{-z_{PA}a_4 - x_{PA}a_3}{\sqrt{1 - y_{PA}^2}} \right) \\ a_5\sqrt{1 - y_{PA}^2} + y_{PA}a_8 & a_7\sqrt{1 - y_{PA}^2} + y_{PA}a_8 \end{bmatrix} \quad (4.28)$$

where

$$\begin{aligned} a_1 &\equiv \cos(\lambda_A) \cos(\lambda_P) + \sin(\lambda_A) \cos(\tau_A - \tau_P) \\ a_2 &\equiv \cos(\alpha) \cos(\tau_A - \tau_P) + \sin(\alpha) \sin(\lambda_A) \sin(\tau_A - \tau_P) \\ a_3 &\equiv \sin(\lambda_A) \cos(\lambda_P) - \cos(\lambda_A) \cos(\tau_A - \tau_P) \sin(\lambda_P) \\ a_4 &\equiv \cos(\alpha) \sin(\tau_A - \tau_P) \sin(\lambda_P) - \sin(\alpha) a_1 \\ a_5 &\equiv \sin(\alpha) \cos(\tau_A - \tau_P) - \cos(\alpha) \sin(\lambda_A) \\ a_6 &\equiv \frac{x_{PA}a_2 - z_{PA} \cos(\lambda_A) \sin(\tau_A - \tau_P)}{\sqrt{1 - y_{PA}^2}} \\ a_7 &\equiv \sin(\alpha) \sin(\tau_A - \tau_P) \sin(\lambda_P) + \cos(\alpha) a_1 \\ a_8 &\equiv \frac{x_{PA}a_4 - z_{PA}a_3}{\sqrt{1 - y_{PA}^2}} \end{aligned} \quad (4.29)$$

The rotated angle in the RLL coordinate system is then given by

$$\psi_{RLL} = \text{atan2}(z_x, z_y)$$

where

$$\begin{bmatrix} z_x \\ z_y \end{bmatrix} = C \begin{bmatrix} \sin \psi_{SLL} \\ \cos \psi_{SLL} \end{bmatrix} \quad (4.30)$$

In the final step, the rotated wind-vector components in the RLL system are given by

$$\begin{aligned} u_{wr} &= \cos(\psi_{RLL}) \sqrt{(u_w)^2 + (v_w)^2} \\ v_{wr} &= \sin(\psi_{RLL}) \sqrt{(u_w)^2 + (v_w)^2} \end{aligned} \quad (4.31)$$

4.4.4 Nondimensionalization of Equations of Motion

By introducing a new unit of time and making a few algebraic manipulations, the equations of motion may now be nondimensionalized and put into a form that is common to flights between any two points at any airspeed. The nondimensionalization converts the linearized equations of motion in RLL coordinates to a set of normalized coordinates, referred to as NLL coordinates (fig. 4.7). This allows one NOC solution to be applied to a

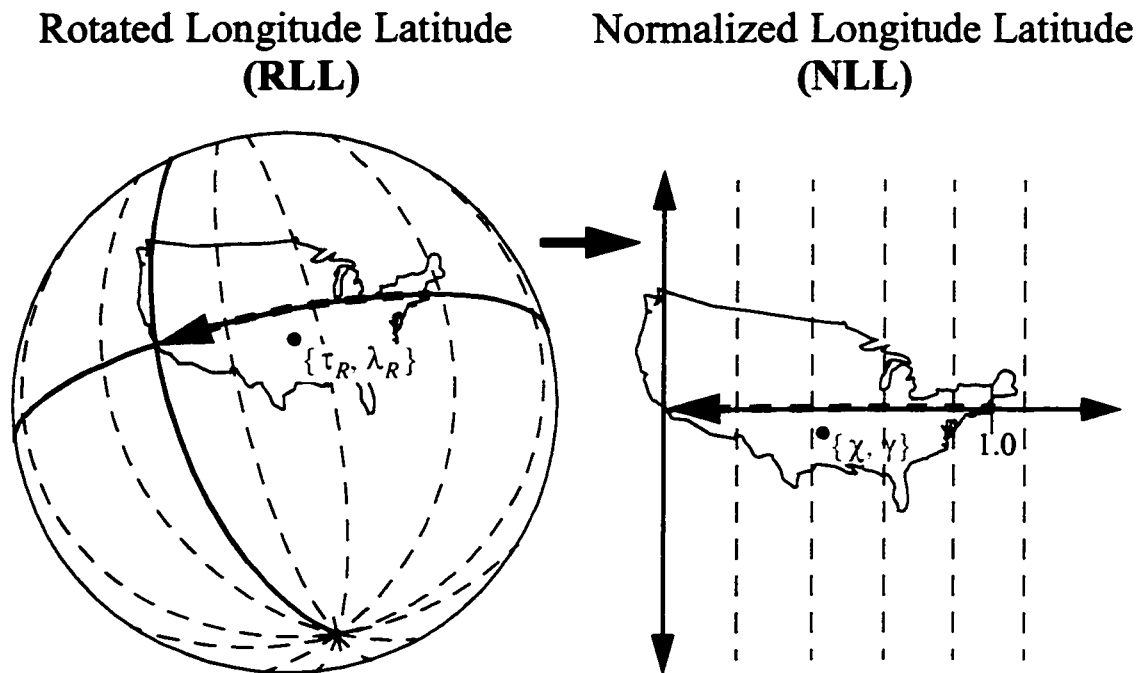


Figure 4.7. Transformation from RLL to NLL coordinates.

wide range of different problems. The utility of having to compute only one NOC solution is that the NOC solution is difficult to compute. If it can be computed once and stored for use, this is clearly a benefit. Similarly, memory is conserved by only requiring that one set of feedback gains be stored.

Nondimensional time is related to standard time by the ratio of the airspeed, V , and the nominal flight distance, L , and is defined by

$$T \equiv \left(\frac{V}{L}\right)t \quad (4.32)$$

The derivative of any variable, A , with respect to standard time is then related to the derivative with respect to nondimensional time, T , by the following equation:

$$\frac{dA}{dt} = \frac{dA}{dT} \cdot \frac{dT}{dt} = \left(\frac{V}{L}\right) \frac{dA}{dT} \quad (4.33)$$

Substituting for the time derivatives in equation (4.26) and dividing through by V leads to the following equations of motion:

$$\frac{dx}{dT} = \cos\theta + u(x, y) \quad (4.34)$$

$$\frac{dy}{dT} = \sin\theta + v(x, y) \quad (4.35)$$

where the following nondimensional variables have been introduced:

$$\begin{aligned} x &\equiv (x_r/L) & y &\equiv (y_r/L) \\ u &\equiv (u_{wr}/V) & v &\equiv (v_{wr}/V) \end{aligned} \quad (4.36)$$

4.4.5 General Wind Field Modeling

Prior NOC solutions to the optimal wind-routing problem modeled the along-track winds as constant in the along-track direction and as linearly varying in the cross-track direction. This allowed analytical solutions which provided useful insight into the nature of optimal wind routes, but it was not accurate enough for the real problem of optimal aircraft routing. A more general wind model is needed, one that can model general changes in the winds.

It must be emphasized that, in the spirit of dynamic modeling for control systems, the chosen wind model is just a convenient mathematical description of the dynamical system *for the purposes of NOC feedback gain computation only*. In practice, once the NOC feedback gains have been computed, a more accurate description of the wind field may be used when computing optimal routes. In fact, if the NOC solution is used for real-time feedback control, then the exact wind field is encountered as an aircraft travels along its trajectory. As in all linear feedback control, if the linear model is reasonably accurate, then the results obtained with the linearized NOC controller will be adequate given the real, nonlinear, system.

Many different forms are possible for modeling the winds as functions of position. Linear or piecewise-linear functions are chosen for this application because they can be made to match the real winds arbitrarily closely by increasing the resolution of the modeling grid [67]. Another reason for using piecewise-linear models is that the NOC technique effectively linearizes the equations of motion so that if more general wind models are used, only linear terms remain after linearization.

The nominal winds are set to zero for the computation of the nominal optimal trajectory, which is simply a straight line from the initial position to the final position, with $0 \leq x_{\text{nom}} \leq 1$ and $y_{\text{nom}} = 0$. Since x_{nom} varies across a range of values, the winds are modeled as piecewise-linear functions of x . Since y_{nom} is a constant (zero in this case), the winds need only be a linear function of y instead of piecewise linear. This is because only the variation in the winds near the nominal trajectory affects the neighboring optimal gain solution. This suggests the following form for the piecewise-linear wind model:

$$u(x, y) = y \cdot u_y + u_b \quad (4.37)$$

$$v(x, y) = y \cdot v_y + v_b \quad (4.38)$$

where the shear terms are piecewise linear in x and are defined as

$$u_y = \left. \frac{\partial}{\partial y} u(x, y) \right|_{y=0} \quad (4.39)$$

$$v_y = \left. \frac{\partial}{\partial y} v(x, y) \right|_{y=0} \quad (4.40)$$

The bias wind terms are also piecewise-linear functions of x and are defined as

$$u_b = u(x, 0) \quad (4.41)$$

$$v_b = v(x, 0) \quad (4.42)$$

When the optimal solution to the Zermelo problem is linearized for small perturbations in the wind parameters, it is easily shown that only the u_y and v_b terms remain (appendix 1). The complete wind model therefore simplifies to:

$$u(x, y) = y \cdot u_y \quad (4.43)$$

$$v(x, y) = v_b \quad (4.44)$$

If the nominal winds are nonzero, then a more detailed wind model is required. Once gains have been computed, more detailed wind models should be used when applying the NOC gains to compute neighboring-optimal trajectories.

The wind parameters, u_y and v_b , are modeled as piecewise-linear functions of x , the normalized along-track position. These functions take on specified values at each of N_w grid points along the x axis, and vary linearly in between grid points (fig. 4.8). Given x , it is straightforward to find x_i and x_{i+1} , and therefore i . Between any two points, x_i and x_{i+1} , the value of a piecewise-linear function, z , is simply

$$z(x) = z_i + \frac{(x - x_i)}{(x_{i+1} - x_i)}(z_{i+1} - z_i) \quad (4.45)$$

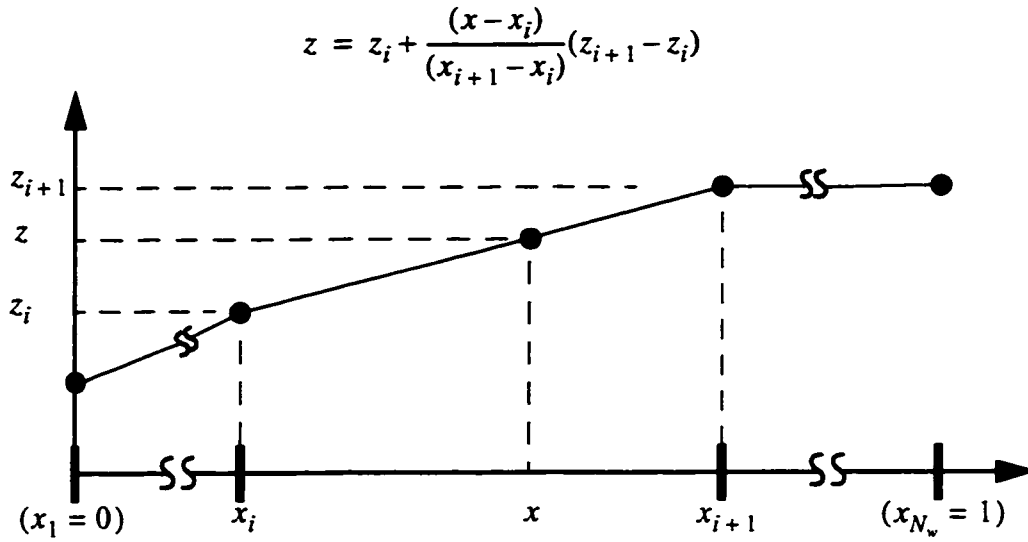


Figure 4.8. A graphical depiction of a piecewise-linear function of x .

4.4.6 Analytical Neighboring Optimal Control Solution

Neighboring optimal control solutions usually must be obtained via numerical methods. In the present case, the fact that the NOC solution is to be computed for the case of nominally zero winds enables the derivation of an analytical solution. The analytical solution is a significant improvement over numerical solutions because it eliminates the need to employ table look-up to find the gains during implementation of the neighboring optimal control law. The analytical solution has been obtained for the general case where the winds are modeled as piecewise linear on a grid of N_w points. Solutions when N_w becomes large (greater than about 15) are not even possible via numerical methods because of computational difficulties.

The derivation of the analytical solution is straightforward, but tedious because of the piecewise-linear wind models. Therefore, the solution is presented here, with the derivation being deferred to appendix 1. The derivation of the numerical solution via the

backward sweep method is also presented in appendix 1 for comparison, and to show how NOC gains might be computed for more complicated systems.

The equations of motion for the neighboring optimal wind routing problem are given by

$$\dot{x} = \cos \theta + y \cdot u_y \quad (4.46)$$

$$\dot{y} = \sin \theta + v_b \quad (4.47)$$

The parameters u_y and v_b are piecewise-linear functions of x as shown in fig. 4.8, and are specified at N_w grid points along the x -axis. Note that the value of x at any grid point, i is given by

$$x_i = \frac{i-1}{N_w-1} \quad (4.48)$$

The initial conditions are:

$$x(0) = 1 \quad (4.49)$$

$$y(0) = 0 \quad (4.50)$$

and the final conditions are:

$$x(t_f) = 0 \quad (4.51)$$

$$y(t_f) = 0 \quad (4.52)$$

The nominal optimal heading is obtained for the case of zero winds, and is given by

$$\theta_{\text{nom}} = \pi \quad (4.53)$$

The situation is depicted graphically in fig. 4.9.

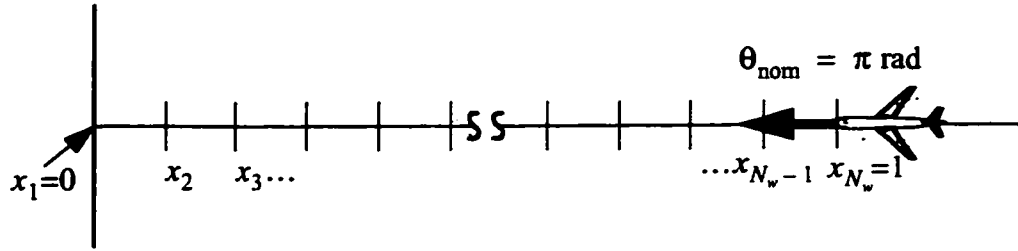


Figure 4.9. Graphical depiction of the neighboring optimal wind routing problem.

The neighboring optimal feedback law is given by

$$\theta = \pi + \frac{\partial \theta}{\partial y} \cdot y + \sum_{i=1}^{N_w} \left(\frac{\partial \theta}{\partial u_{yi}} \cdot u_{yi} + \frac{\partial \theta}{\partial v_{bi}} \cdot v_{bi} \right) \quad (4.54)$$

where the partial derivatives of θ with respect to each of the state variables are the NOC gains, which are now defined.

The neighboring optimal perturbation in heading for perturbations in cross-track position is

$$\frac{\partial \theta}{\partial y} = \frac{1}{x} \quad 0 \leq x \leq 1 \quad (4.55)$$

Note that this linearized feedback gain approaches infinity as the aircraft nears $x = 0$. In practice, a mode switch to a heading regulator is used to avoid this singularity. Another option is to use the optimal nonlinear feedback rule for perturbations in y as x approaches zero, which is

$$\Delta \theta_y = \tan^{-1} \left(\frac{y}{x} \right) \quad (4.56)$$

Note that equation (4.56) reduces to equation (4.55) for small perturbations in y .

Because of the piecewise-linear nature of the wind model, the NOC gains related to the wind parameters must be expressed as piecewise-continuous functions of x . The NOC gain for perturbations in the along-track shear at the first grid point, $i = 1$, is given by

$$\frac{\partial \theta}{\partial u_{y1}} = \begin{cases} \frac{1}{6(N_w - 1)^2} \cdot \frac{1}{x} & \frac{1}{(N_w - 1)} < x \leq 1 \\ \frac{1}{6(N_w - 1)^2} \cdot \frac{1}{x} \cdot [1 - (1 - x \cdot (N_w - 1))^2(2x \cdot (N_w - 1) + 1)] & 0 < x \leq \frac{1}{(N_w - 1)} \\ 0 & \lim x \rightarrow 0 \end{cases} \quad (4.57)$$

All other NOC gains for perturbations in the along-track shear over the range $2 \leq i \leq N_w$ are given by

$$\frac{\partial \theta}{\partial u_{yi}} = \begin{cases} \frac{(i-1)}{(N_w - 1)^2} \cdot \frac{1}{x} & \{x_{i+1} < x \leq 1\} \\ \frac{6(i-1) - (i - x \cdot (N_w - 1))^2 \cdot (2x \cdot (N_w - 1) + i)}{6x \cdot (N_w - 1)^2} & \{x_i < x \leq x_{i+1}\} \\ \frac{[3i - 4 - 3((i-1)^2 - x^2(N_w - 1)^2) + (i-1 - x \cdot (N_w - 1))^2(2x \cdot (N_w - 1) + i - 1)]}{6x \cdot (N_w - 1)^2} & \{x_{i-1} < x \leq x_i\} \\ 0 & \{0 \leq x \leq x_{i-1}\} \end{cases} \quad (4.58)$$

The NOC gain for perturbations in the cross-track bias wind at $i = 1$ is given by

$$\frac{\partial \theta}{\partial v_{b1}} = \begin{cases} \frac{1}{2(N_w - 1)} \cdot \frac{1}{x} & \frac{1}{(N_w - 1)} < x \leq 1 \\ 1 - \frac{(N_w - 1)}{2} \cdot x & 0 \leq x \leq \frac{1}{(N_w - 1)} \end{cases} \quad (4.59)$$

All other NOC gains for perturbations in the cross-track bias wind over the range $2 \leq i \leq N_w$ are given by

$$\frac{\partial \theta}{\partial v_{bi}} = \begin{cases} \frac{1}{(N_w - 1)} \cdot \frac{1}{x} & x_{i+1} < x \leq 1 \\ \frac{2 - (i - x \cdot (N_w - 1))^2}{2x \cdot (N_w - 1)} & x_i < x \leq x_{i+1} \\ \frac{(i - 2 - x \cdot (N_w - 1))^2}{2x \cdot (N_w - 1)} & x_{i-1} < x \leq x_i \\ 0 & 0 \leq x \leq x_{i-1} \end{cases} \quad (4.60)$$

The NOC gains for perturbations in y , u_{yi} , and v_{bi} have been plotted for the specific case of $N_w = 13$ to illustrate their qualitative nature (fig. 4.10).

In prior NOC solutions to the Zermelo problem, NOC gains were tabulated as functions of time-to-go ($t_f - t$). In addition to the neighboring optimal heading perturbation equations, a set of neighboring optimal perturbations in the optimal final time were derived and used to determine the point on the nominal path with the same time-to-go as on the neighboring optimal path. Using time-to-go as the independent variable rather than time ensures that the NOC gains do not run out if the final time on the neighboring optimal path happens to be greater than the nominal final time.

In the present application, the use of a nominal path which is a straight line enables the gains to be expressed as a function of x , which is the distance-to-go along the nominal path. The same benefit of not having the NOC gains run out along the trajectory is

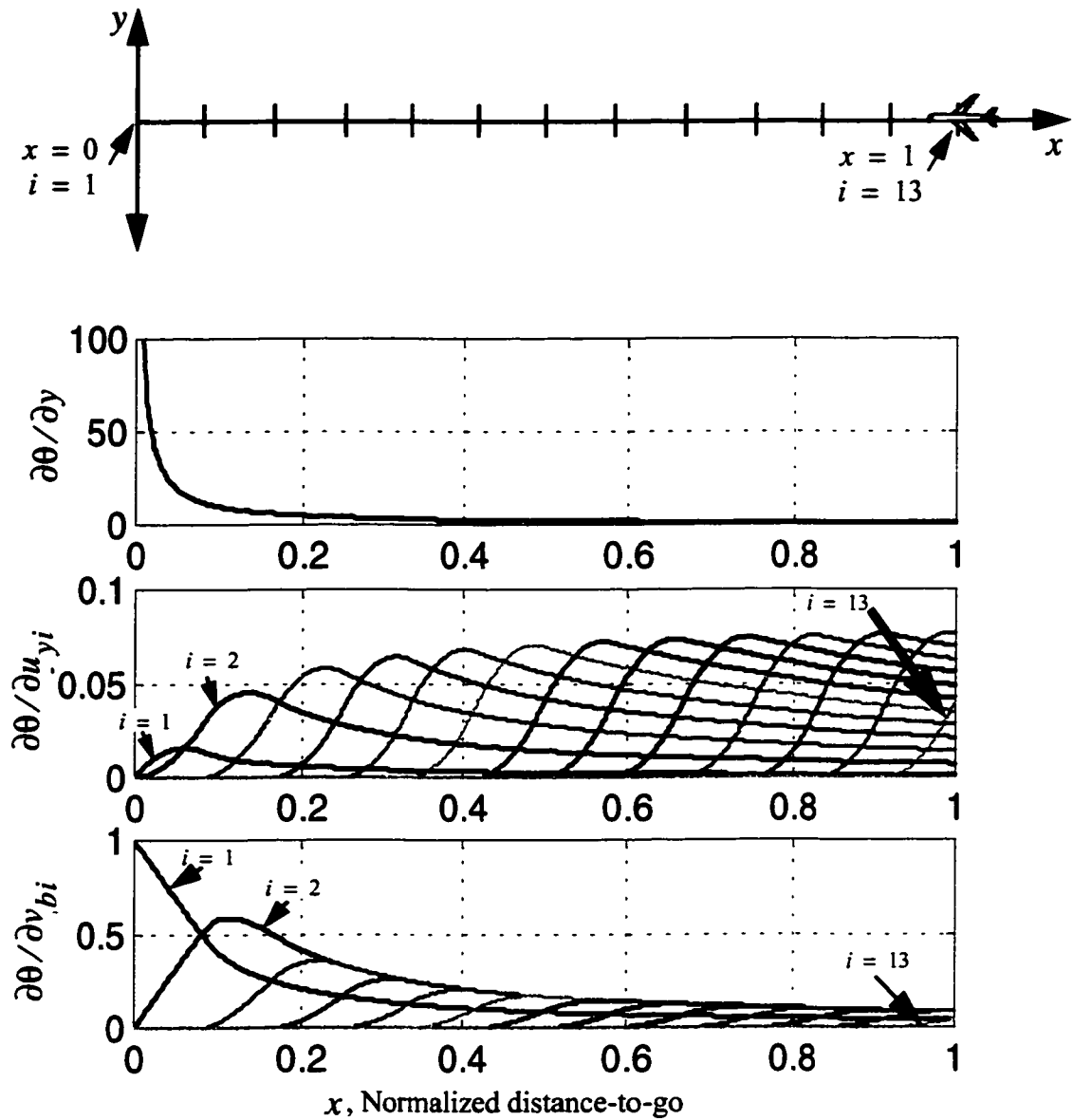


Figure 4.10. Normalized neighboring optimal feedback gains.

achieved with this method. The analytical solution for the NOC gains has eliminated the table look-up procedure, which simplifies the application of NOC. Since the winds are modeled as a function of x , the resulting neighboring-optimal solutions have better performance than if the NOC gains are obtained as a function of time-to-go. This is because the effect of winds along the route on the optimal trajectory is a function of x only.

Although it is straightforward to compute NOC gains for any value of N_w with the analytical solution, N_w should not be made any larger than required for the accurate modeling of the wind field. Each new wind measurement point adds another $2N_w$ feedback gains to the problem, which can slow down the implementation. Winds across the United States are adequately modeled for the purpose of trajectory optimization using points spaced by about 200 nmi. Increasing grid spacing too far can result in inaccurate weather models by under-sampling the winds, possibly resulting in aliasing. Decreasing the grid spacing may lead to excessive computation without any significant gain in optimization performance. Using this as a guideline, $N_w = 13$ was determined to be adequate for the longest routes in the continental United States which are about 2500 nautical miles.

Careful examination of the normalized feedback gains in fig. 4.10 demonstrates how wind perturbations affect the optimal heading along the route. In particular, note how feedback gains associated with perturbations in the winds at distant locations affects the current optimal heading. Also note that as an aircraft passes a point, i , the effect of the feedback gain associated with the point $i + 1$ drops to zero as one would expect.

4.5 Application of Neighboring Optimal Wind Routing

Optimal wind-routes may be computed in real-time by using the feedback law of equation (4.54) to guide the aircraft, or they may be computed using a fast-time simulation so that the entire route may be precomputed and given as a clearance to the aircraft. The best option depends on the particular application; both are now presented.

4.5.1 Real-Time Neighboring Optimal Wind Routing

The application of the NOWR feedback gains to the computation of the neighboring-optimal heading perturbation is performed in the nondimensional rotated (NLL) coordinate frame. Additional computations are required to convert the resulting solutions back to the SLL coordinate frame (fig. 4.11). These steps are as follows:

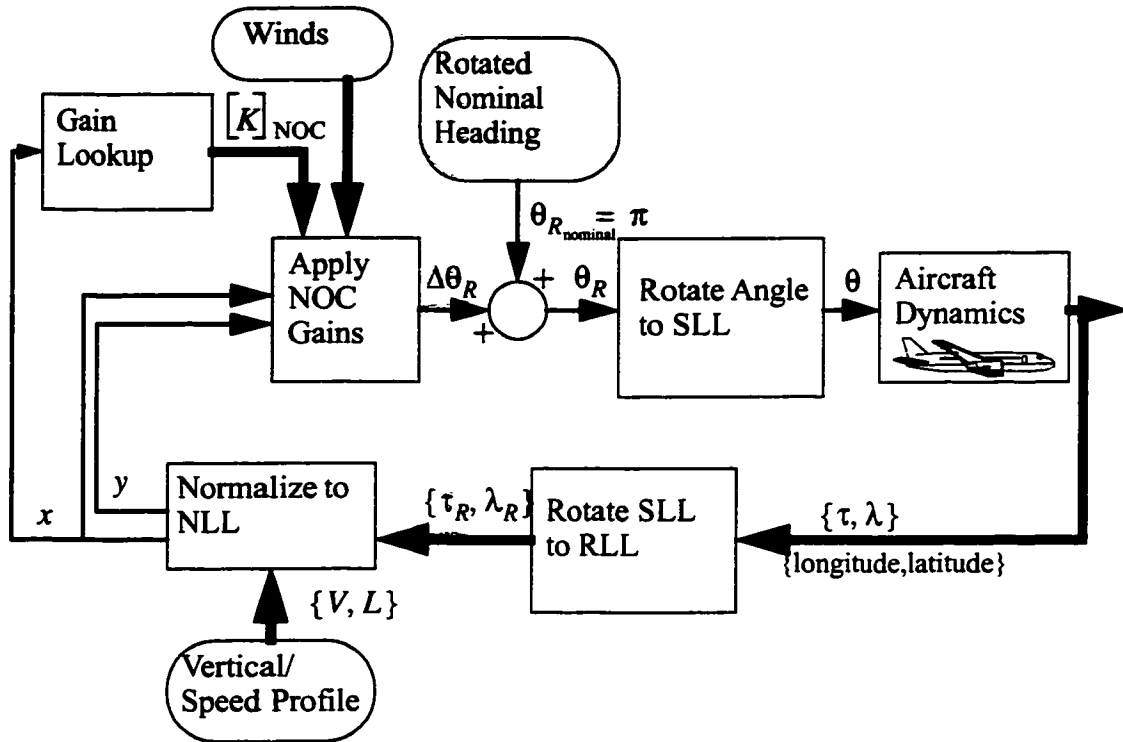


Figure 4.11. Block diagram of the NOWR algorithm.

1. Rotate Aircraft Coordinates from SLL to RLL: Apply the rotation algorithm in equations (4.23) through (4.25) to the current aircraft coordinates.
2. Normalize to obtain NLL Coordinates: Apply the normalization as shown in equation (4.26) and equation (4.36) to obtain the current aircraft coordinates in NLL coordinates, $\{x_c, y_c\}$.
3. Compute Wind Perturbations: The wind data are usually obtained by interpolating from a gridded model in SLL coordinates. Several options are available for obtaining the SLL coordinates. If the nominal great-circle route is already known in SLL coordinates, one may simply divide the SLL great-circle route into $N_w - 1$ equally spaced sections and then look up the u_{yi} and v_{bi} terms at the N_w connecting points. Another option is to begin with the NLL coordinates where the nominal path is a straight line from the point $\{1, 0\}$

to $\{0, 0\}$. The N_w sets of coordinates are obtained from equation (4.48) and then rotated to their equivalent SLL coordinates for interpolation of the u_w and v_w data.

Once the u_w and v_w data have been interpolated, they are rotated to RLL coordinates by means of equation (4.31). The v_{bi} values may be computed directly from v_{wR} as follows:

$$v_{bi} = \frac{(v_{wR})_i}{V} \quad i = 1, 2, \dots, N_w \quad (4.61)$$

The shear terms, u_{yi} , still must be computed from the $(u_{wR})_i$ values. There are many possible methods for computing the shear terms, two of which are described here.

The first method for computing the shear terms is to choose a value of $\pm y_s$ at which to compute u_{wRi} . The shear vector is then computed by the following formula:

$$u_{yi} = \frac{u_{wR}(x_i, y_s) - u_{wR}(x_i, -y_s)}{2V \cdot y_s} \quad i = 1, 2, \dots, N_w \quad (4.62)$$

The second method is to choose a range of cross-track values $(-y_s \leq y \leq y_s)$. The shear is then obtained by computing the slope of the line that best fits the wind data over that range of y in a least-square error sense, or in some other optimized sense.

Either of these methods may be modified by iterating on the choice of y_s and then computing the optimal wind trajectory by means of fast-time simulation (described below). After computing the optimal flight time, y_s is varied until the value is found that leads to the minimum predicted flight time. This would typically be done as an analysis study, and the best average value would then become a fixed parameter for application of the NOWR algorithm. As shown later in this chapter, the best value is $y_s = 0.08$.

4. Compute Neighboring Optimal Heading: After the wind terms have been computed, the optimal heading in the RLL system is computed from equation (4.54).

To rotate θ_{RLL} to the SLL system, one must implement the inverse rotation given in equation (4.23) through equation (4.25) and equation (4.28) through equation (4.30). When rotating from the SLL system to the RLL system, the coordinates of $\{\tau_A, \lambda_A\}$ and $\{\tau_B, \lambda_B\}$, which were the coordinates of the final point of travel and the initial point of travel in the SLL system, were required. The equivalent of these coordinates in the RLL system is now needed.

To accomplish the reverse rotation, one must first convert the origin of the SLL coordinate system (the intersection of the equator with the prime meridian, $\{0, 0\}$) into the RLL system. A second point along the SLL equator is also needed in the RLL system for the reverse rotation algorithm. Any point along the SLL equator will do: for example, one may use the point $\{0, 45^\circ\}$. These rotations produce the following two sets of coordinates:

$$\begin{aligned} \{0, 0\}|_{SLL} &\Rightarrow \{\tau_{0R}, \lambda_{0R}\}|_{RLL} \\ &\text{and} \\ \{0, \lambda_{\text{equator}}\}|_{SLL} &\Rightarrow \{\tau_{0eR}, \lambda_{0eR}\}|_{RLL} \end{aligned} \quad (4.63)$$

One may now use the rotation algorithm in equation (4.23) through (4.25) and equation (4.28) through (4.30) by making the following substitutions:

$$\begin{aligned} \tau_A &\equiv \tau_{0R} & \lambda_A &\equiv \lambda_{0R} \\ \tau_B &\equiv \tau_{0eR} & \lambda_B &\equiv \lambda_{0eR} \end{aligned} \quad (4.64)$$

In these reverse rotations, the point $\{\tau_p, \lambda_p\}$ now represents the location of the aircraft in the RLL coordinates. After completing these rotations, the total neighboring optimal heading command in SLL coordinates is obtained:

$$\theta_{RLL} \Rightarrow \theta_{SLL} \quad (4.65)$$

For the real-time system, this heading command must be updated continuously to guide the aircraft along the neighboring optimal trajectory.

4.5.2 Fast-Time Simulation for Neighboring Optimal Wind Routing

In this mode, neighboring optimal wind routes are precomputed by means of fast-time simulation using a computer to integrate the equations of motion for the aircraft. This produces an entire flight trajectory, which is then given to the aircraft to follow. The trajectory may be updated whenever desired (when new wind data become available, for example). The steps are as follows.

1. Rotate from SLL coordinates to NLL coordinates and Compute Wind Parameters: The rotation formula to go from SLL coordinates to NLL coordinates is exactly the same as was outlined in equation (4.23) through (4.25) and equation (4.27) through (4.36). The wind parameters are also computed just as in the real-time algorithm.

2. Integrate Equations of Motion: The equations of motion to be integrated in the NLL system are slightly different from those used for the neighboring optimal gain computations. This is because now the equations of motion are being integrated to obtain an accurate trajectory so that the decoupling simplification necessary for gain computation is not required. The equations of motion are now given by

$$\begin{aligned}\dot{x} &= \frac{\cos\theta + u(x, y)}{\cos(y \cdot \tau_{R0})} \\ \dot{y} &= \sin\theta + v(x, y)\end{aligned}\tag{4.66}$$

where τ_{R0} is the initial longitude in RLL coordinates (recall that $\tau_R = y \cdot \tau_{R0}$). Notice that the spherical-earth equations are used, as are the actual wind values. The piecewise-linear form of the winds is not needed during this step. One may choose to use more detailed equations of motion if more accuracy is required.

3. Compute Neighboring Optimal Heading: The neighboring optimal heading only needs to be computed in the RLL system according to equation (4.54). The rotation of the

heading command to the SLL system is not required in this case since the equations of motion are integrated in the RLL system and the resulting trajectory is converted from RLL to SLL coordinates.

4. Convert Resulting Trajectory from RLL to SLL Coordinates: The reverse conversion of RLL latitude and longitude to SLL latitude and longitude is achieved using equation (4.23) through (4.25) along with the substitutions shown in equations (4.63) and (4.64). This leads to the desired neighboring optimal trajectory in SLL coordinates.

4.6 NOWR Examples

Several examples are now presented to demonstrate the NOWR algorithm. The first set of examples are devised to illustrate how linear feedback operates on perturbations in the winds to achieve neighboring optimal trajectories. These examples are generated for the case of $N_w = 3$ so that one may more easily follow the effects of the feedback gains on the neighboring optimal solution. For these cases, the actual winds are also piecewise linear as modeled for the computation of the NOWR gains. A second set of examples is presented to demonstrate the NOWR algorithm in a more realistic application. The NOWR algorithm has been installed in the Future ATM Concept Evaluation Tool (FACET) so that neighboring optimal wind-routes may be computed within a spherical-Earth model using RUC atmospheric data.

4.6.1 Illustrative Examples

In these examples, normalized coordinates are used. Dimensionalizing the results to any desired range and airspeed may be achieved via equation (4.32) and (4.36). The aircraft begins at the initial point, $\{x = 1, y = 0\}$, and proceeds at unit velocity to the final point, $\{x = 0, y = 0\}$. The number of wind grid points is $N_w = 3$, and winds are

modeled as piecewise linear in between grid points. The feedback gains for these examples are listed in table 4.2 and are presented graphically in figure 4.12.

Table 4.2. NOC feedback gains for 3 grid points.

| NOC Gain | $0 \leq x \leq \frac{1}{2}$ | $\frac{1}{2} < x \leq 1$ |
|----------------------------------|----------------------------------|-----------------------------------|
| $\partial\theta/\partial y$ | $1/x$ | $1/x$ |
| $\partial\theta/\partial u_{y1}$ | $-\frac{2}{3}x^2 + \frac{1}{2}x$ | $\frac{1}{24x}$ |
| $\partial\theta/\partial u_{y2}$ | $\frac{2x^2}{3}$ | $\frac{(-1 + 12x^2 - 8x^3)}{12x}$ |
| $\partial\theta/\partial u_{y3}$ | 0 | $\frac{(1 - 12x^2 + 16x^3)}{24x}$ |
| $\partial\theta/\partial v_{b1}$ | $1 - x$ | $\frac{1}{4x}$ |
| $\partial\theta/\partial v_{b2}$ | x | $\frac{(-1 + 4x - 2x^2)}{2x}$ |
| $\partial\theta/\partial v_{b3}$ | 0 | $\frac{(1 - 2x)^2}{4x}$ |

The winds for the first example are zero along the nominal route, but have a cross-track shear in the along-track wind at the point $x_1 = 0$ (fig. 4.13) such that $u_{y1} = -1$, $u_{y2} = u_{y3} = 0$. The goal of this example is to demonstrate how perturbations in the winds at a distant location have a measurable effect on the optimal heading. In this case, the winds are zero throughout the first half of the trajectory, but the optimal heading angle perturbation is clearly nonzero in this region. Both the neighboring optimal solution and the exact optimal solution have been computed for these wind conditions, and are nearly identical to one another (fig. 4.14). The aircraft symbols depicted on the trajectory plot are shown at approximately the correct heading angle, but note that the angles are distorted because of the different scales on the x and y axes. Also note that the heading angle is relative to the air mass, and is therefore different than the course angle in nonzero winds.

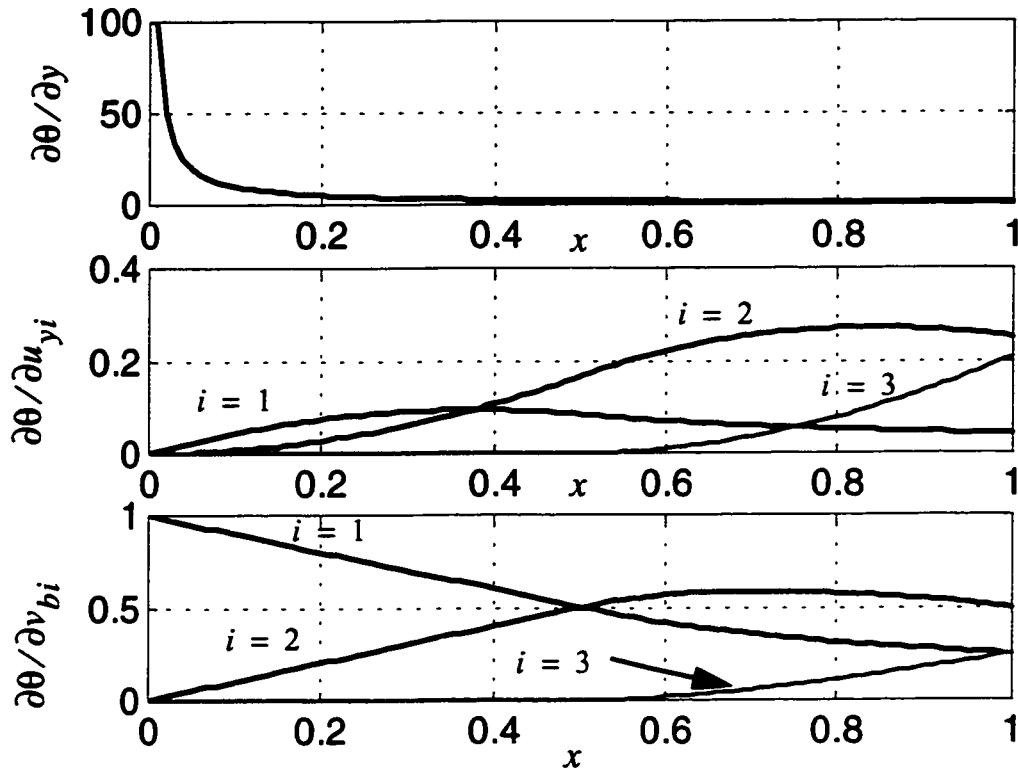


Figure 4.12. NOC feedback gains for the case of three grid points.

The only discernible difference in the neighboring-optimal and exact-optimal solutions is the small perturbation in the neighboring optimal heading as the aircraft nears the final position. This is due to a mode switch to a heading capture command near the final point to avoid the singularity in the NOC solution as discussed earlier in this chapter. The minimum flight time in this example is only about 0.22% less than the nominal flight time. If this example were scaled to a nominal range of 2500 n.mi. and an airspeed of 480 kn, then the nominal flight time would be 312.5 min, and the optimal flight time would be about 1 minute less. For these dimensions, the shear at $x = 0$ is $u_{y1} = 0.192$ kn/n.mi. and the maximum trajectory deviation is about 66 n.mi. This is not a particularly severe wind shear, as much higher values may be encountered in the real atmosphere.

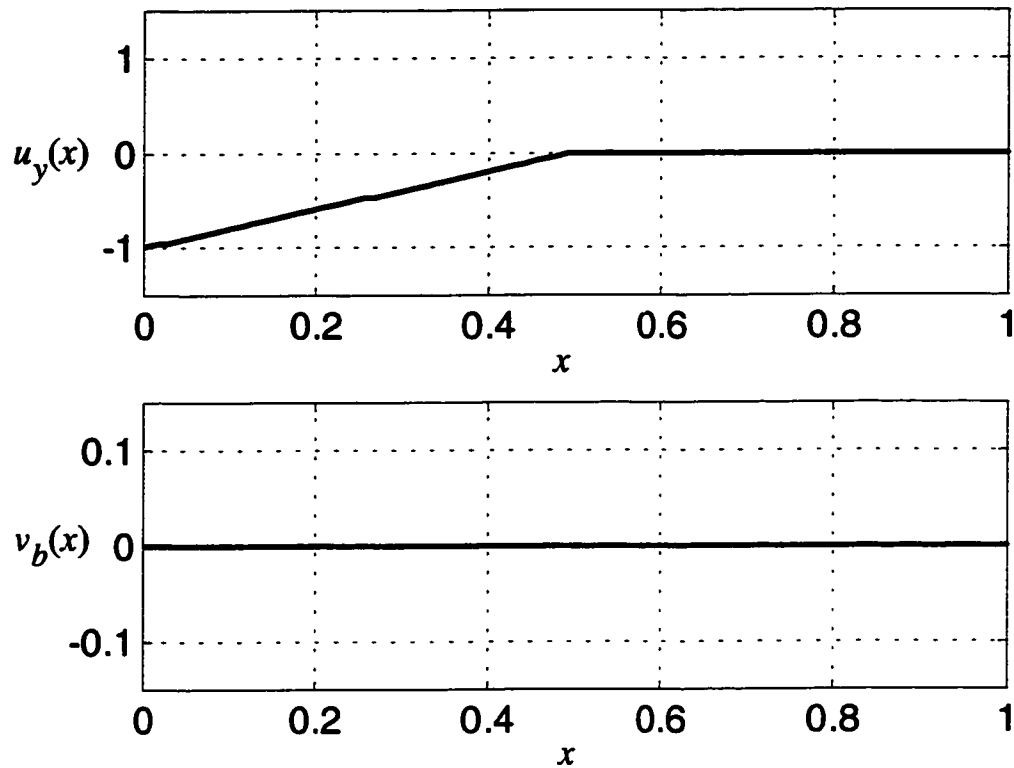


Figure 4.13. Illustrative example-1 wind parameters.

In the second example, the cross-track shear in the along-track direction is modeled as a linearly varying function of x (fig. 4.15). In this case, $u_{y1} = -0.1$, $u_{y2} = 0$, and $u_{y3} = 0.1$. The resulting trajectory (fig. 4.16) hints at the rich structure that may be achieved in neighboring optimal trajectories when the winds are modeled as functions of x . Again, the neighboring-optimal and exact-optimal solutions are nearly identical. In this second example, the minimum time is about 0.23% of the nominal flight time.

The goal of the third example is to demonstrate the effect of the cross-track winds on the optimal trajectory, and to again show how winds along the entire route affect the optimal heading. In this example, the along-track winds are zero and the cross-track winds are modeled as a linear function of x . In this case, $v_{b1} = -0.1$, $v_{b2} = 0$, and $v_{b3} = 0.1$ (fig. 4.17). The interesting feature in this example is that the contributions of the

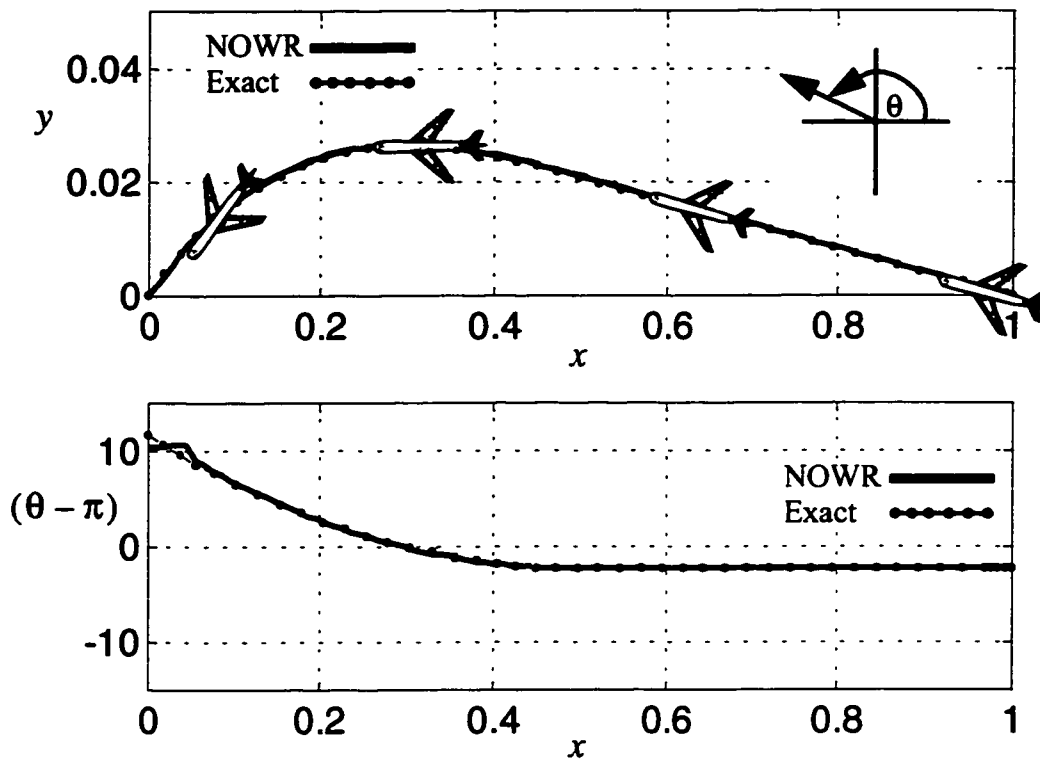


Figure 4.14. Illustrative example-1 trajectory plot and plot of optimal heading vs. x .

perturbations in y and in the two cross-track wind terms exactly cancel along the entire trajectory for these particular perturbations so that the aircraft maintains a constant heading relative to the air (fig. 4.18). The NOC solution accounts for the fact that the cross-track winds will change sign. The optimal strategy is to maintain a constant heading relative to the air because ultimately the aircraft will drift back to the correct final position. The optimal flight time with the perturbed cross winds is identical to the nominal flight time in zero winds even though the flight distance is increased. This is possible because the ground speed increases from $V_g = 1$ to $V_g = \sqrt{1 + v_b^2}$. The reader is encouraged to examine this case in greater detail by computing the individual contributions of the y , v_{b1} , and v_{b3} terms to the neighboring optimal heading perturbation to see that they do indeed cancel.

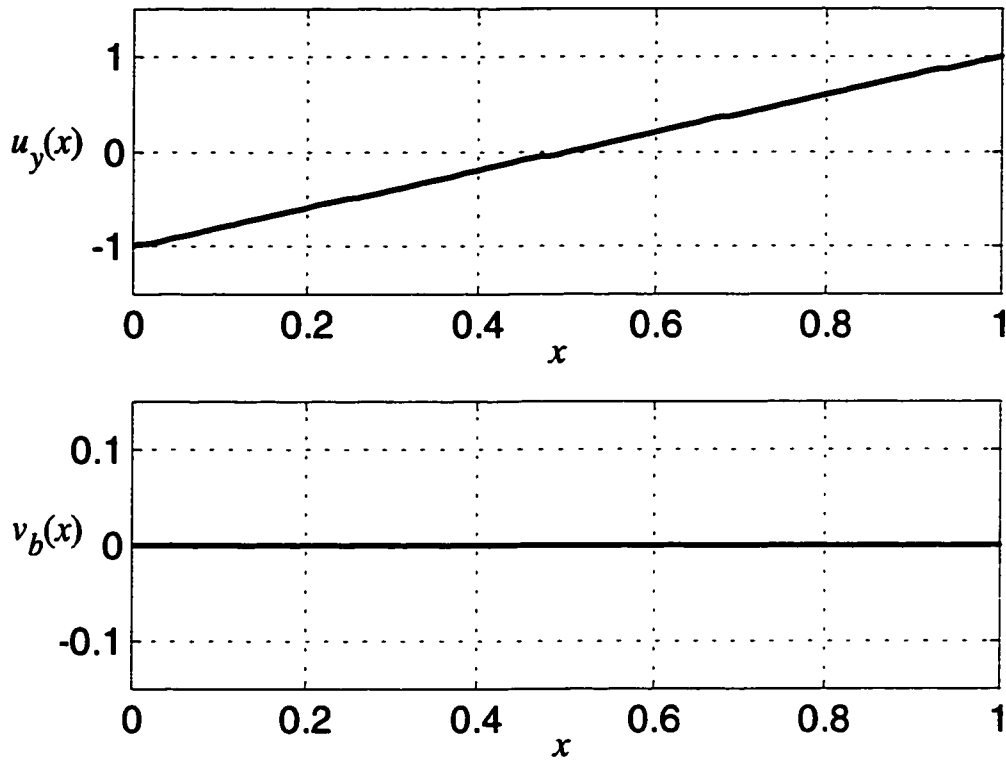


Figure 4.15. Illustrative example-2 wind parameters.

4.6.2 FACET Simulation Examples

The NOWR algorithm has been coded into the Future ATM Concept Evaluation Tool (FACET) to examine neighboring optimal wind routes in a real air-traffic control environment. FACET integrates the aircraft kinematic equations over the spherical Earth and utilizes detailed wind data from the Rapid Update Cycle (RUC) [63]. It is emphasized that in these examples, the aircraft trajectory is integrated through a real wind field along spherical-earth coordinates even though the NOC feedback law was derived using simpler linearized models. The NOWR gains are implemented in FACET using 13 grid points so that $N_w = 13$ (fig. 4.10).

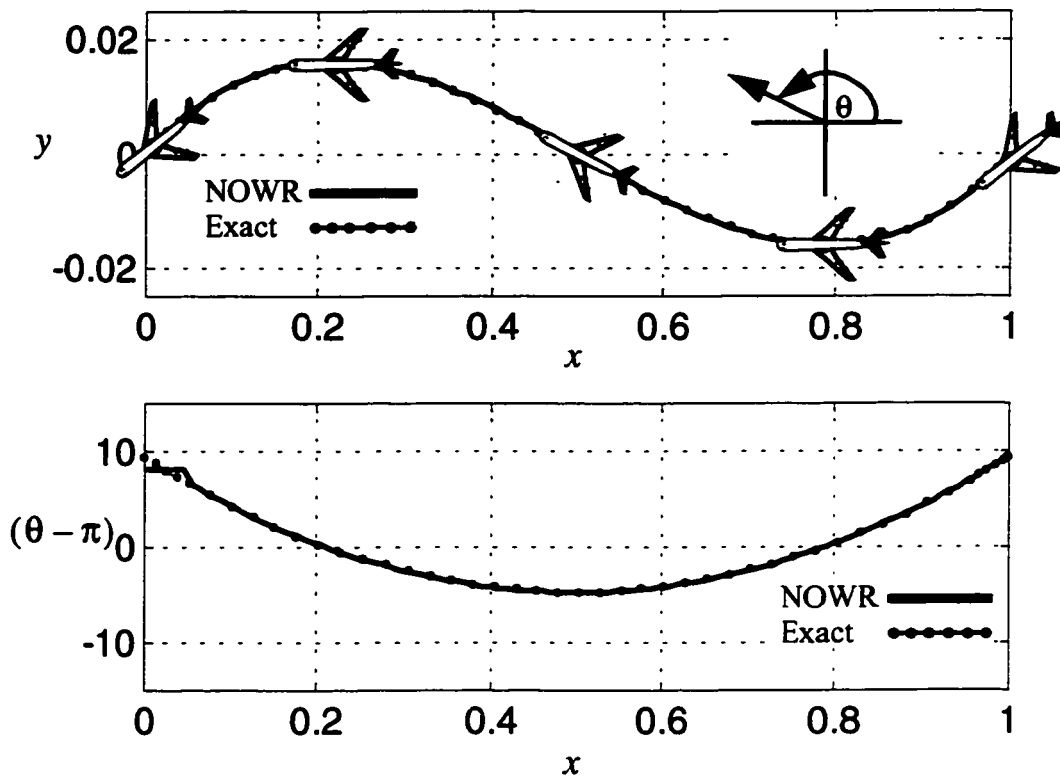


Figure 4.16. Illustrative example-2 trajectory plot and plot of optimal heading vs. x .

The first FACET example (fig. 4.19) shows a flight from New York (JFK) to San Francisco (SFO) using RUC wind data from 14 February 2001 at approximately 36,000 ft MSL. The great-circle flight distance is $L = 2242$ n.mi., and the airspeed is $V = 450$ kn. The cross-track shear in the along-track winds on this day was large because of the presence of a strong south-westerly jet stream, which is typical in the winter months. The time saved over the corresponding great-circle route was about 20 min, and the maximum deviation from the great-circle route was more than 300 n.mi.

The second FACET example (fig. 4.20) shows a flight from Miami, Florida (MIA) to Seattle, Washington (SEA) through the same wind field as in the first example. The great-circle flight distance is $L = 2365$ n.mi., and the airspeed is again $V = 450$ kn. The time saved over the great-circle is only about 7 minutes because the shear is less than for the

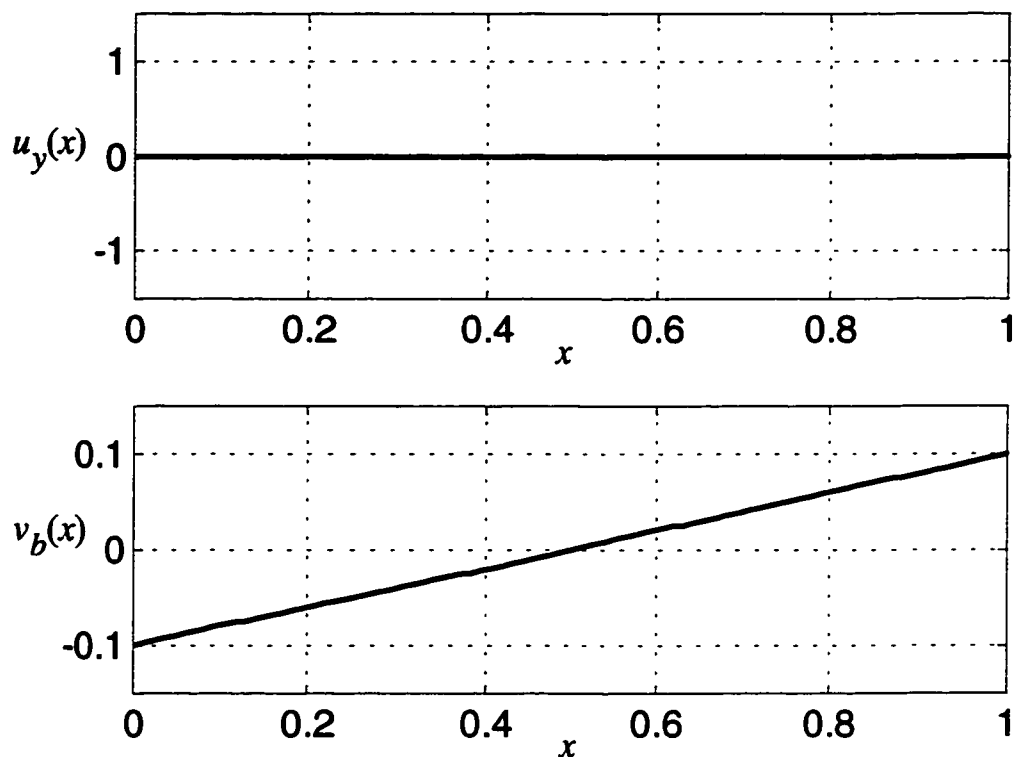


Figure 4.17. Illustrative example-3 wind parameters.

JFK to SFO route. The important feature of this second example is that the neighboring optimal wind route was computed using exactly the same normalized feedback gains as in the first example. The difference in trajectories is a result of the different coordinate rotations, dimensional scaling, and the different winds along each route.

4.7 NOWR Performance Analysis

How close to optimal are the NOWR solutions, and under what conditions can the performance be bounded? How computationally efficient is the NOWR algorithm when compared to a DP algorithm? These questions are examined by comparing optimal routes computed with NOWR to those computed by a representative DP algorithm. The computational effort and optimization performance of each is evaluated.

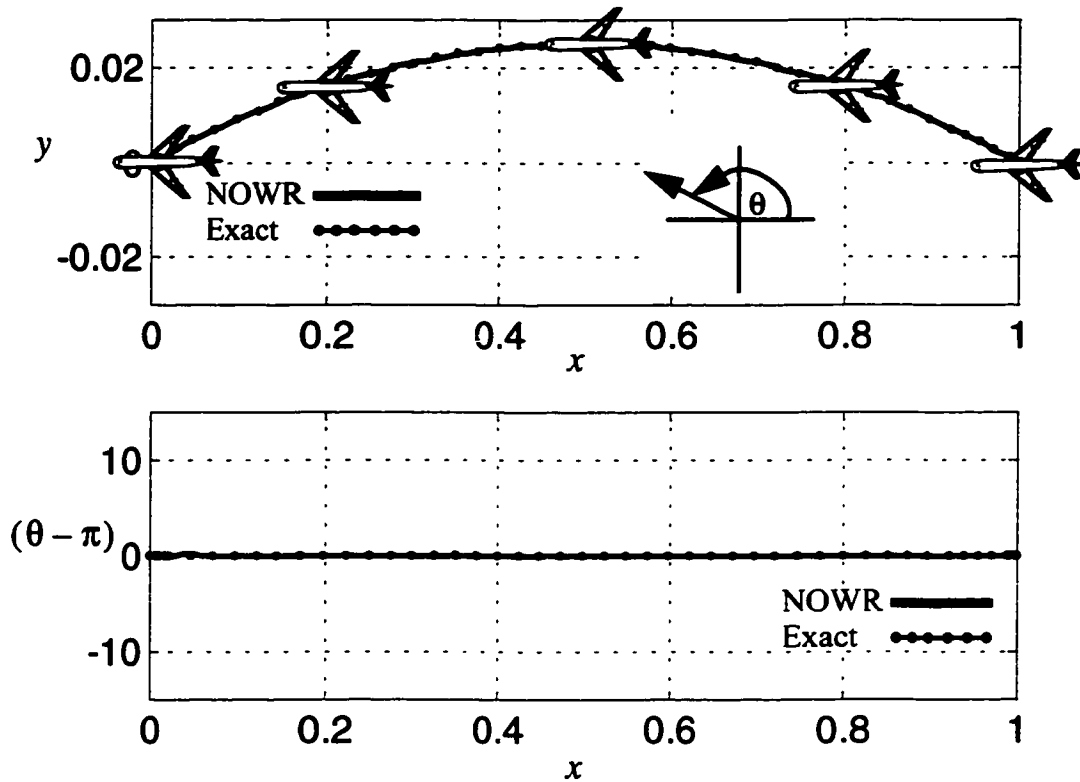


Figure 4.18. Illustrative example-3 trajectory plot and plot of optimal heading vs. x.

A generally varying wind field is not easily modeled as a deterministic analytical function. At the scale of importance to aircraft trajectory optimization, the wind field may contain significant areas of nonlinearity such as jet streams and circulation zones. While improbable, it is possible that the linearizing assumptions underlying the NOWR algorithm might produce trajectories that are completely out of phase with the real nonlinear wind field. These worst-case trajectories could have significant performance deficits when compared to the true optimal solution which takes into account the real system dynamics. These differences in performance for the worst-case scenarios could be so large that they would not be useful as practical bounds. This suggests that an empirical approach might be the best way to compute practical bounds on NOWR performance. To compute empirical bounds on NOWR performance requires that both the NOWR solution

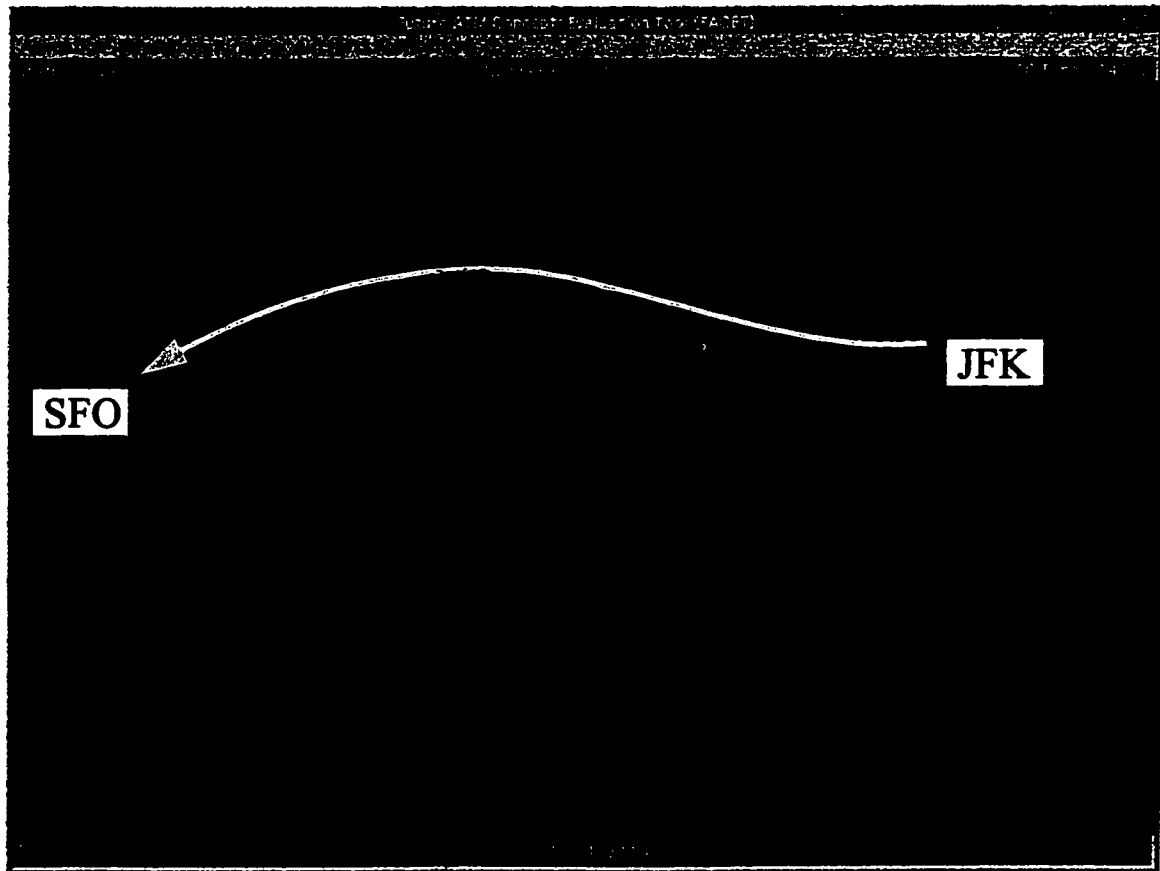


Figure 4.19. FACET example-1: NOWR route from JFK to SFO (2/14/2001).

and some close approximation to the true optimum solution be computed for a statistically significant set of wind conditions and flight paths. A comprehensive analysis might lead to NOWR optimization performance results that could be reported as a function of different parameters such as time-of-day, time-of-year, altitude, and region. The goal here is to show how such a comprehensive analysis might be achieved by demonstrating the concept for a small subset of wind conditions and routes.

4.7.1 Heuristic Dynamic Programming Algorithm

The optimal solution is to be computed with a discrete Dynamic Programming (DP) algorithm, the basics of which were described in section 4.3. Discrete DP is essentially an exhaustive search technique whereby the flight times along each segment of the DP grid

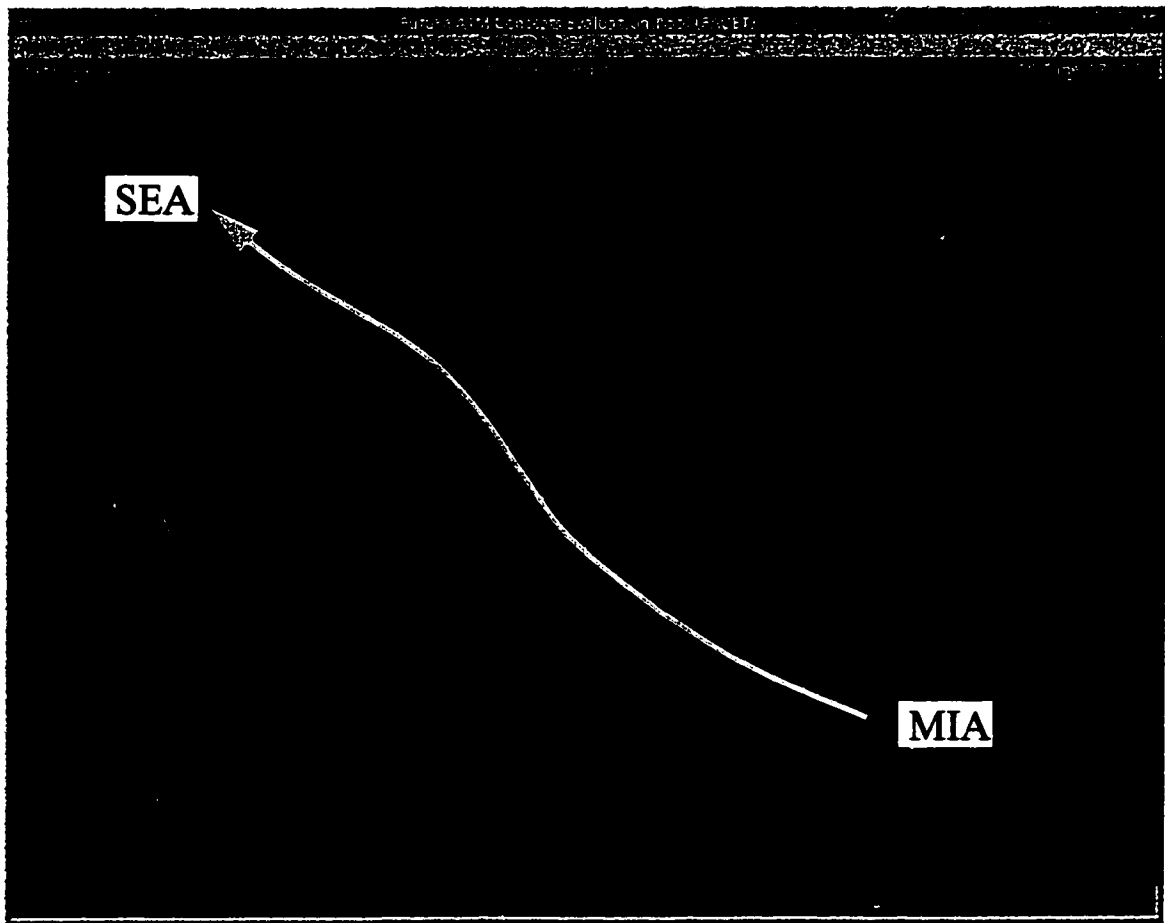


Figure 4.20. FACET example-2: NOWR route from Miami to Seattle (2/14/2001).

are computed and the minimum-time path selected. A DP algorithm offers the advantage of producing the true optimal solution for a given grid network. Other techniques either suffer from the tendency to become stuck at a local minimum solution or, in the case of randomized search techniques, they cannot guarantee that the global optimum solution has been found. To find the approximate optimal solution, the resolution of the DP grid is increased until the computation time and computer memory limits are reached. The optimization performance at several different grid resolutions may be examined to verify that the minimum time solution is converging. This value is then taken as an approximation of the globally optimal route.

Since the computational effort of the DP algorithm will be compared with that of the NOWR algorithm, an effort was made to use reasonable heuristics to limit the search

space of the DP algorithm. However, since there are practically an infinite number of ways to implement a heuristic DP algorithm, no claim can be made regarding the fundamental and absolute comparison of NOWR and DP. For that, a more rigorous abstract computational analysis would be required, which is beyond the scope of what is attempted here.

To simplify and speed up the DP computations, the same coordinate system rotations were used as for the NOWR computations (figs. 4.6 and 4.7). A simple Cartesian grid was then defined, with the departure angle from each point to the next point and the distance between along-track stations being variable parameters (fig. 4.21). For a given resolution,

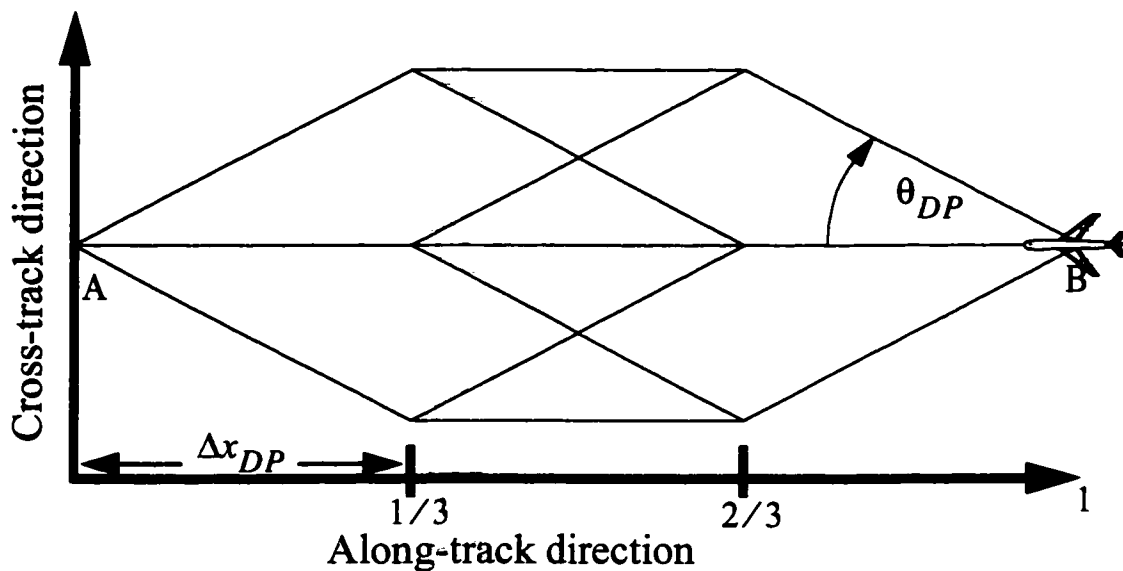


Figure 4.21. Illustration of the Dynamic Programming grid.

values for θ_{DP} and Δx_{DP} are chosen, the optimal route is computed, and then the resulting route is dimensionalized and rotated back to SLL coordinates for output. The range of the grid was limited such that departure angles from the initial point were less than or equal to 45° , and the maximum cross-track deviation allowed was 25% of the great-circle path distance. An example grid and optimal route solution are shown in figure 4.22.

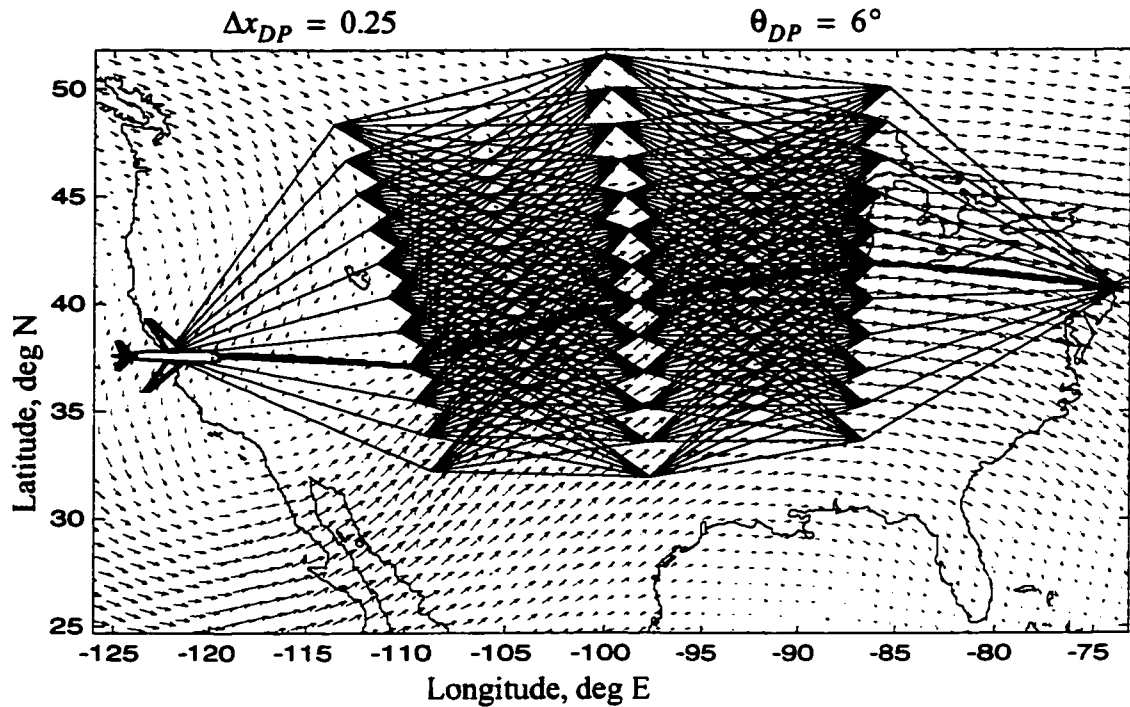


Figure 4.22. A Dynamic Programming solution.

One simplification that greatly speeds up the computations is to use simple rules for integrating the flight time along each segment of the DP grid. When the grid segments are fairly short, the winds along that segment are approximately constant. Therefore, the segment ground speed is computed at the midpoint of each segment. The great-circle distance for each segment is computed and divided by the average ground speed to obtain the approximate segment flight time. Once the route with the minimum time is selected in this way, a more accurate integration is carried out by causing the aircraft to fly great-circle routes between each segment point along the selected minimum-time route. The details of this simplified DP method are now presented.

For each segment consisting of initial SLL coordinates $\{\tau_0, \lambda_0\}$ and final SLL coordinates $\{\tau_f, \lambda_f\}$, the great-circle distance on a spherical Earth is given by

$$L_{gc} = 2R \sqrt{\sin^2\left(\frac{\lambda_0 - \lambda_f}{2}\right) + \cos \lambda_0 \cos \lambda_f \sin^2\left(\frac{\tau_0 - \tau_f}{2}\right)} \quad (4.67)$$

Other more compact formulas may be used, but the one in equation (4.67) is less susceptible to rounding errors than others.

The average longitude and latitude along the segment are computed as simple averages:

$$\tau_m \equiv (\tau_0 + \tau_f)/2 \quad (4.68)$$

$$\lambda_m \equiv (\lambda_0 + \lambda_f)/2 \quad (4.69)$$

The winds are interpolated from the provided RUC data at the $\{\tau_m, \lambda_m\}$ coordinates. The ground speed at this point must be computed accounting for the crab angle required to maintain a great-circle course from $\{\tau_m, \lambda_m\}$ to the final point, $\{\tau_f, \lambda_f\}$. The great-circle ground speed at this point is given by

$$V_{gm} = V \cdot \cos\left(\sin^{-1}\left(\frac{V_w}{V} \sin(\chi_w - \chi_g)\right)\right) + V_w \cdot \cos(\chi_w - \chi_g) \quad (4.70)$$

where V is the airspeed, V_w is the wind magnitude, and the wind vector angle, χ_w , and aircraft course angle, χ_g , are defined as

$$\chi_w \equiv \text{atan2}(u_w, v_w) \quad (4.71)$$

$$\chi_g \equiv \text{atan2}[\sin(\tau_f - \tau_m) \cos \lambda_f \sin \lambda_f \cos \lambda_m \cos(\tau_f - \tau_m)] \quad (4.72)$$

and u_w and v_w are the east and north wind vector components in the SLL coordinate system.

The approximate segment flight time is then computed as

$$t_{\text{seg}} = \frac{L_{gc}}{V_{gm}} \quad (4.73)$$

Once the flight time along each segment has been computed, the algorithm determines the minimum-time route to each node in the graph from the initial point to the final point. After the minimum-time route has been selected using the approximate segment times, a more accurate trajectory is numerically integrated assuming great-circle routes in between grid points. This final trajectory is of the same level of fidelity as those computed using the NOWR algorithm. Note that because of this final numerical integration, the DP algorithm will always take at least as long as the NOWR algorithm because the NOWR algorithm is essentially just a numerical integration of the trajectory.

The advantage of going through this approximation scheme is that it eliminates the need to perform computationally expensive numerical integration along each segment. The time saved is dramatic, and when segment sizes are small, there is no difference in the resulting solutions.

4.7.2 Empirical Study Data Set: Winds and Routes

High-wind-shear days were chosen for this analysis because high-wind-shear conditions have the highest potential for NOWR performance difficulties. Since the highest shear conditions tend to happen in the winter months, the data for this empirical study were chosen from several days in February 2002, and from one additional high-wind-shear day in February 2001. The wind data and data processing used for this study are the same as those presented earlier in this chapter for the comparison of great-circle routes to optimal wind routes. The specific data files used for this study are listed in table 4.1. The same set of long-distance routes is used for this study as for the comparison of optimal wind routes with great-circle routes (fig. 4.1).

An initial analysis of these data was performed to determine the value of y_s that gave the best average performance for the NOWR algorithm. The best value was determined to be

$$y_s|_{\text{best}} = 0.08 \quad (4.74)$$

This best value of y_s has been used to compute all of the NOWR results in this chapter, and in the rest of the dissertation.

4.7.3 NOWR versus DP Comparison Results

The comparisons have been computed in MATLAB on a Sun Ultra 450-MHz work station. Using the DP algorithm defined in this chapter, the grid resolution is varied and the DP optimal routes are computed for each of the 42 routes for each of the different weather files for a total of 252 different routes at each DP resolution. The DP resolution is varied according to the values shown in table 4.3. The along-track grid resolution, Δx_{DP} ,

Table 4.3. Combinations of DP grid resolution.

| $\Delta x_{DP} \backslash \theta_{DP}$ | 2 deg | 3 deg | 4 deg | 5 deg | 10 deg | 20 deg | 45 deg |
|--|-------|-------|-------|-------|--------|--------|--------|
| 5% | | | | | | X | X |
| 10% | X | X | | X | X | X | X |
| 25% | X | X | X | X | X | X | X |
| 33% | X | | | X | | | |
| 50% | | | | | | X | X |

is expressed as a percentage of the total great-circle-route length, and the angular resolution, θ_{DP} , is expressed in degrees. Not every combination of these two parameters has been tested, either because the combined grid resolution clearly became too coarse to achieve meaningful results, or because the combined resolution became so high that the computation would take an exceedingly long time or would exceed memory capacity on the computer hardware.

The highest combined resolution achieved with the best average optimization performance is at $\Delta x_{DP} = 0.10$ and $\theta_{DP} = 2^\circ$. At this resolution, the total computation time for the 252 optimal-route computations exceeded 15 hr. The lower resolution simulations all took correspondingly less time, and the NOWR simulations required two orders of magnitude less time to complete.

The computational effort was measured in terms of floating point operations, or FLOPS, using MATLAB's built-in FLOP estimation function. This is not an exact measurement of the total number of operations required, but is a good approximation. For more details, one may refer to the MATLAB User's Guide or on-line help [68].

The optimization performance is measured relative to the DP solution with the highest grid resolution ($\Delta x_{DP} = 0.10$, $\theta_{DP} = 2^\circ$), which is referred to as the benchmark DP solution. The results were evaluated in terms of the percentage of additional total flight time required over all 252 routes relative to the benchmark DP solution.

The additional total flight time over 252 routes normalized by the benchmark DP solution is given by

$$\Delta T \equiv \frac{J - J_{\text{benchmark}}}{J_{\text{benchmark}}} \quad (4.75)$$

The cost function, J , is defined as

$$J \equiv \sum_{i=1}^{N_R} (t_f)_i \quad (4.76)$$

where $(t_f)_i$ is the flight time for route i , and N_R is the total number of routes (252 in this case).

This value is plotted as a percent versus the DP grid departure angle, θ_{DP} for a family of different values of Δx_{DP} in the upper plot of figure 4.23. For comparison, the NOWR and great-circle solutions are shown on the same plot. The total additional flight time of

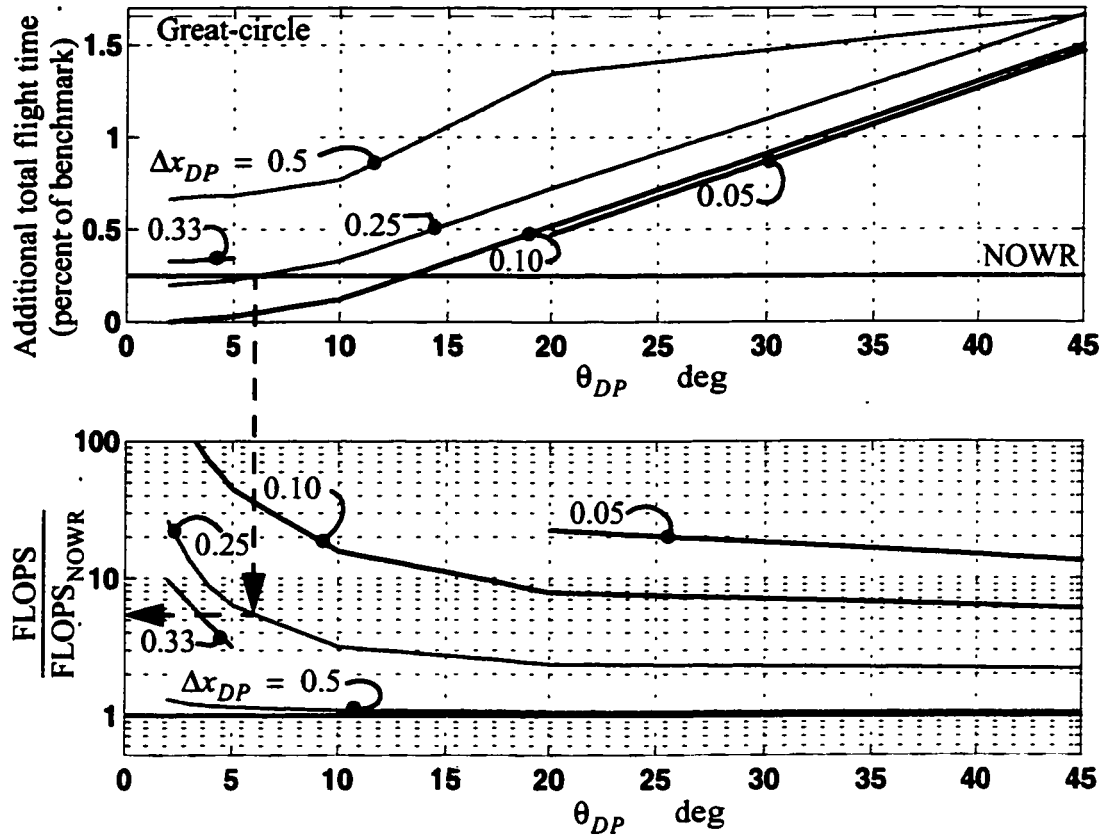


Figure 4.23. Performance analysis of NOWR and DP.

the NOWR solution over all 252 routes is shown to be just 0.25% above the benchmark DP solution. The great-circle solution is more than 1.6% above the benchmark. All of the various DP solutions computed according to table 4.3 exhibit the expected behavior as a function of both Δx_{DP} and θ_{DP} . Note that as the along-track grid resolution is increased, the solutions appear to asymptotically approach a common value. This is most clearly noticed when examining the small difference between the solution curve for $\Delta x_{DP} = 0.10$ and $\Delta x_{DP} = 0.05$. This suggests that the benchmark DP solution is near the true optimum solution.

The lower plot in figure 4.23 shows the total floating point operations, or FLOPS, versus θ_{DP} , and again for a family of values of Δx_{DP} . The FLOPS have been normalized

by $\text{FLOPS}_{\text{NOWR}}$, the number of FLOPS required for the NOWR solutions. Therefore, the NOWR solution is at unity on the vertical axis. The great-circle FLOPS are nearly identical to the NOWR FLOPS.

Each of the values of Δx_{DP} corresponds to a value that divides the nominal route into an integer number of segments: 2 ($\Delta x_{DP} = 0.5$), 3 ($\Delta x_{DP} = 0.33$), 4 ($\Delta x_{DP} = 0.25$), 10 ($\Delta x_{DP} = 0.1$), or 20 ($\Delta x_{DP} = 0.05$). The two plots may be used to determine the grid resolution that achieves the same optimization performance (on the upper plot in figure 4.23) for the lowest computational effort. The most efficient DP grid resolution that achieved the same average performance as NOWR is $\Delta x_{DP} = 0.25$ and $\theta_{DP} = 6^\circ$, which is the same resolution shown in fig. 4.22. The additional computational effort for this solution is more than 5 times greater than that required for the NOWR or for great-circle solutions. All other grid resolutions with the same performance as NOWR required even greater computational effort.

Qualitatively, these results show that even fairly coarse DP grids can achieve good optimization performance. This suggests that the minimum time cost function is flat near the optimum solution. Coarse solutions are good because they require less computational effort, but on the other hand, they still require more than 5 times the computational effort of the NOWR algorithm. Another problem is that it is not easy to perturb the solutions to achieve nearby optimum solutions. This is a significant drawback because the trajectories will need to be slightly perturbed during conflict resolution, as will be shown in chapter 5.

4.7.4 Limitations of NOWR

Analysis showed that, on average, NOWR exhibits excellent optimization performance and is computationally efficient. Part of the credit for this good performance goes to the mesoscale wind field, which is smoothly varying most of the time. Under some conditions, however, a strong jet stream or localized region of strong circulation can push the NOWR algorithm beyond the linearizing assumptions and lead to substantially suboptimal performance. The good average performance suggests that such cases are rare.

Among the routes used in the NOWR versus DP comparisons, one such situation was noticed. On 14 February 2001, there was a strong southwesterly jet stream flowing across the United States with winds exceeding 170 kn. The NOWR solution for a route from New York (JFK) to San Francisco (SFO) at FL350 showed a rather large deviation from the great-circle route to achieve more than 20 min in flight-time savings. This is good, but the DP solution was able to save more than 50 min by making an even larger deviation. Examining the plot of these routes and the associated wind field shows how this occurred (fig. 4.24). Near JFK, the jet stream straddled the great-circle route symmetrically so that

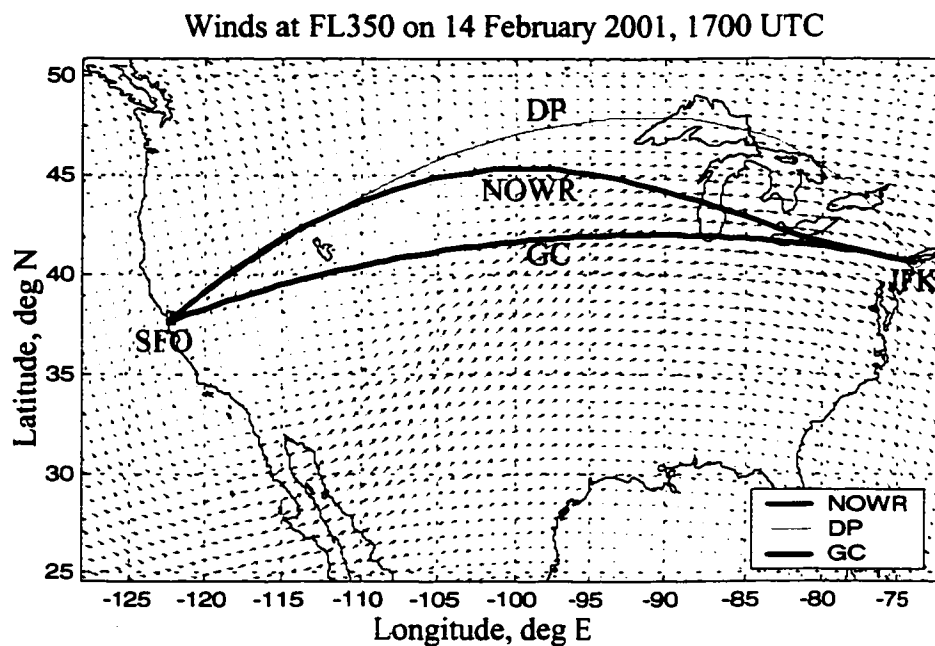


Figure 4.24. An example of highly nonlinear wind conditions.

the local change in wind speed with cross-track deviation was nearly zero in that area. Therefore, the NOWR algorithm didn't "see" any advantage in deviating in that region. The DP search algorithm was not limited in this way and found that a large immediate deviation to the north could get around the jet stream entirely. Toward the latter half of the trajectory when the cross-track wind shear became apparent to the NOWR algorithm, the DP and NOWR solutions began to match up closely.

On that particular day (i.e., 14 Feb. 2001), the JFK-SFO route was one of the only routes that exhibited this type of behavior. If the initial trajectory point was moved just slightly to the North, the NOWR algorithm could begin to see the significant cross-track shear so the DP and NOWR solutions matched much more closely.

This kind of behavior is to be expected since the assumptions underlying the NOWR algorithm are only valid for small perturbations from a locally optimum solution. But in practice, large deviations are usually handled gracefully by the NOWR algorithm. Even in the case shown above, the penalty was that the NOWR algorithm only saved 20 minutes instead of the possible 50 min. To account for such occurrences in practice, one might occasionally compute DP solutions between common city pairs to determine when these nonlinear conditions are prevalent. In these cases, it may be advantageous to use some form of perturbation guidance other than NOWR. One might modify the NOWR algorithm to control around perturbations from an arbitrarily specified route. This is left as a subject for future research.

A simple metric has been defined to help identify when the NOWR route may have exceeded the small perturbation assumption. While the NOWR route is being computed, one may record the maximum cross-track deviation, y_{\max} . This value may be compared to the cross-track deviation used to compute the discrete wind shear as given by y_s in equation (4.62). This maximum perturbation metric, μ_s , is defined as

$$\mu_s \equiv \frac{y_{\max}}{y_s} \quad (4.77)$$

If μ_s is much greater than 1, then it is possible the NOWR route has gone outside of the small perturbation limit. Conversely, if μ_s is much less than 1, the value of y_s should be adjusted lower. On average, $y_s = 0.08$ has been shown to be a good value.

4.8 Benefits and Enhancements of NOWR Over Prior Art

4.8.1 Modeling for Perturbations in Along-Track Wind Shear

Referring to Bryson & Ho [39], the Zermelo problem was previously solved using neighboring optimal control by fixing the value of the wind shear parameter (rate of change of along-track wind in the cross-track direction) and then solving for the nominal optimal path for the fixed conditions. NOC feedback gains were computed only for perturbations in position, but not for perturbations in the wind shear. If the wind shear changed from the modeled value, then the nominal optimization computations had to be redone or substantially sub-optimal performance would result.

The nominal optimal solution is for zero winds in the current formulation, and is therefore trivial. This, in addition to the nondimensionalization of the solution, is what allows a single nondimensional solution to be applied to flights between any two points and at any flight speed. This is a significant improvement over the prior art because it produces good performance using linear feedback.

4.8.2 Incorporating Varying Along-Track Wind Shear

Prior work in neighboring optimal control for the Zermelo problem has treated the wind shear as constant in the along-track direction. Modeling the shear parameter as linearly varying between chosen grid points has allowed neighboring optimal control gains to be computed for perturbations in the wind shear parameters. This is another significant improvement over the prior art.

4.8.3 Incorporating Perturbations in Cross-Track Winds

Prior work on the Zermelo problem did not incorporate cross-track winds. The cross-track winds do affect the optimal solution and are therefore important to include in the neighboring optimal control formulation. This is easily understood by imagining a case where the cross-track winds are positive during the first half of the flight, and equal in magnitude but negative during the second half of the flight. By incorporating the cross-track perturbation terms in the neighboring optimal control solution, the aircraft is allowed

to drift in the cross-track direction during the first half of the flight because the solution “knows” that the aircraft will drift back to the final point during the second half of the flight. Without including the cross-track wind perturbations in the solution, the aircraft would be directed to fight against the cross-track winds, resulting in a longer flight time.

4.8.4 Analytical Solution for the NOC Gains

The derivation of the analytical solution for the NOC gains for the general case of piecewise-linear winds modeled at N_w grid points constitutes a significant advance over prior art. The analytical solution eliminates the need for tedious and computationally difficult numerical solutions. When implementing the NOC feedback law, the analytical solution eliminates the need for tabulating the NOC feedback gains and using table look-up. This reduces any memory burden that would have been required to store a table of gains, and eliminates the computational effort required to interpolate gains from the table.

4.8.5 3D with Performance Optimization

The neighboring optimal wind routing algorithm can be extended to include variations in the winds in the vertical coordinate. The difficulty with including the vertical coordinate is that commercial jet aircraft performance is also coupled with altitude. Although the goal of neighboring optimal wind routing is to find minimum-time paths, the overriding goal is to find minimum-fuel paths. The fuel minimization is assumed to have been performed by choosing the appropriate flight altitude and airspeed. By then performing time-minimization for fixed altitude and airspeed, the total fuel use is further reduced.

The simplest extension would be to include commanded altitude as a new control and to use a 3-dimensional wind model. The resulting vertical profile would need to be evaluated for its practicality and efficiency for the given aircraft and engine performance, and for any ATC constraints.

The next level of complexity would be to include state models of the fuel flow rate for the aircraft as a function of airspeed and altitude when computing the neighboring optimal control gains.

4.9 A Few Words on Optimal Routes

Through limited simulation analysis, optimal routes have been shown to be different from great-circle routes. Because of the interesting variations in the winds, a similarly interesting variation in the shapes of optimal routes is exhibited. Some optimal routes just deviate to one side or the other of the corresponding great-circle routes; others may cross the great-circle route several times along the trajectory. The shape of any particular optimal route depends on the wind conditions at the time.

Observing optimal wind routes during different wind conditions can be fascinating as new patterns and subtleties emerge. A complete analysis of the atmosphere and related optimal wind-routes would make an interesting study. A few qualitative observations on the nature of optimal routes follow.

4.9.1 Slower Aircraft Benefit More from Optimal Wind Routing

Greater time savings are possible for slower aircraft, simply because the nominal flight times of the aircraft are longer. The nominal flight time, t_{nom} , on a route of length L at ground speed V_g is given by

$$t_{\text{nom}} = \frac{L}{V_g} \quad (4.78)$$

The perturbed flight time on the same route at a ground speed increased by ΔV_g is given by

$$t_p = \frac{L}{(V_g + \Delta V_g)} \quad (4.79)$$

The magnitude of the time difference between these two flight times as a fraction of the nominal flight time is simply

$$\frac{\Delta t}{t_{\text{nom}}} = \frac{|t_p - t_{\text{nom}}|}{t_{\text{nom}}} = \frac{|\Delta V_g|}{|V_g + \Delta V_g|} \quad (4.80)$$

This expression shows that a flight that reduces its ground speed will experience a greater time savings as a percentage of its nominal flight time than will a flight which increases its ground speed by the same magnitude. Therefore, an aircraft flying at a slower ground speed will experience a greater percentage of time savings than will a faster aircraft.

4.9.2 Greater Time Savings in Head Winds (East-to-West Routes)

This is a corollary of the previous observation. Over the United States, and in the Northern Hemisphere in general, prevailing winds are from west to east so that aircraft traveling in the opposite direction will usually encounter head winds. This means that east-to-west traffic can expect to benefit more from optimal wind routing than west-to-east traffic.

4.9.3 Similarity Scaling of Optimal Wind Routing

Aircraft that tend to fly at higher altitudes are generally faster than aircraft that fly at lower altitudes. On average, winds increase linearly with altitude up through the troposphere so that the ratio of average aircraft airspeed to average wind speed has a small range, or may even be considered constant. This suggests that the optimal routing problem is scaled such that flight time savings as a percentage of the great-circle time should be similar for jet traffic at 35,000 ft or for a single-engine propeller-driven plane at 5,000 ft. The scaled distance perturbations from the great-circle route would also be expected to be similar.

4.9.4 Horizontal Stratification of Optimal Routes Between City Pairs

When conducting simulation analyses of free-flight concepts, nearly all researchers have used great-circle routes to approximate free-flight routes. As has become clear in this chapter, this is simply not the case. The difference is often large in magnitude and must be accounted for if simulation results are to be applicable to the real world.

One interesting implication of the difference between great-circle routes and optimal wind-routes is that aircraft traveling between city pairs in different directions tend to be naturally separated when aircraft follow optimal wind routes. If it is optimal for an aircraft traveling from New York to San Francisco to deviate to the north to minimize the head wind, then it is optimal for an aircraft traveling from San Francisco to New York to deviate to the south to maximize tail wind. If great-circle routes are assumed, then clearly there will be many conflicts developing among aircraft traveling between any given city pair, but this will generally not happen if the aircraft travel along optimal wind routes.

This implies that in a future free-flight system in which aircraft are permitted to fly along optimal wind-routes, the current flight-level restrictions on eastbound and westbound traffic may be relaxed or eliminated because these two traffic flows will tend to be naturally separated.

4.9.5 Significant Variation in Optimal Wind Routes with Time

As a final observation, optimal wind-routes may vary significantly on time scales of hours. During the day, a jet stream may meander across a significant distance so that the shapes of optimal wind-routes may change dramatically. This reinforces the notion that optimal wind-routes must take into account both the time-varying nature of the winds and the need to frequently recompute the optimal wind-routes as new wind predictions become available.

On one of the days for which wind data were used for the NOWR versus DP comparisons, a significant time-variation in the optimal route from New York (JFK) to San Francisco (SFO) was noticed (fig. 4.25). The time savings for the optimal route at

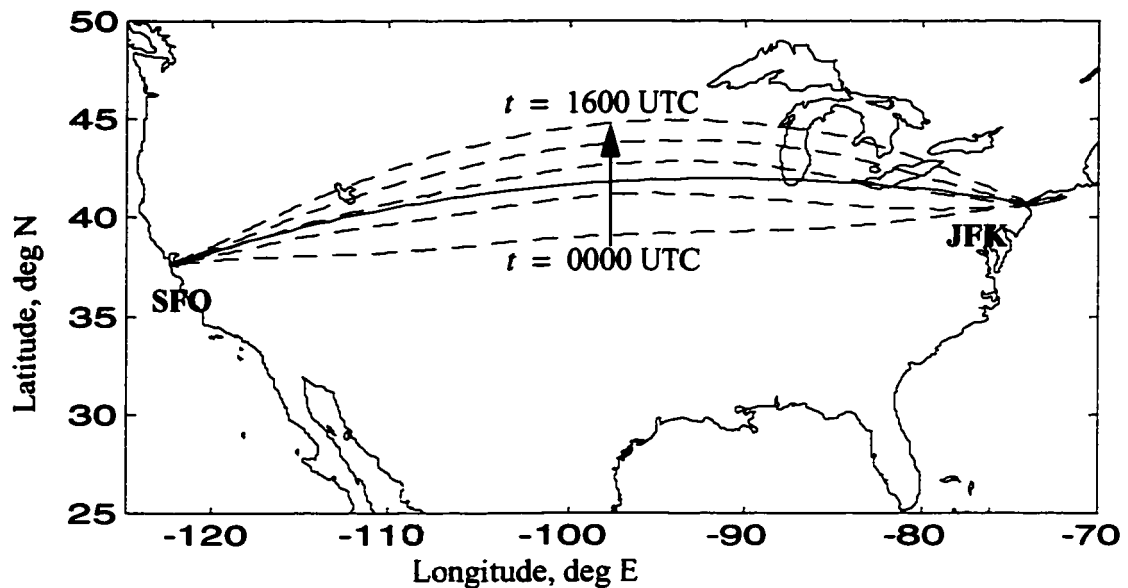


Figure 4.25. Illustration of time variations in optimal wind routes.

0000 UTC (the most southerly route) was just over 17 min. The optimal route at 1600 UTC (the most northerly route) saved about 17 min over the corresponding great-circle route. The maximum distance between these two routes is about 6.5° of latitude, or nearly 400 n.mi.

4.10 Summary

This chapter has introduced the notion of optimal wind-routing and gave some background on many of the prior approaches to solving this problem. In the context of computing optimal wind-routes for many aircraft in real-time, it was shown that an algorithm for optimal wind-routing must be fast while retaining good average optimization performance. To this end, the Neighboring Optimal Wind Routing (NOWR) algorithm was developed. The derivation of the NOWR algorithm from calculus of variations principles was presented, along with the derivation of additional details required to compute optimal wind routes in practice. The derivation of an analytical solution for the NOC feedback gains for a piecewise-linear wind model was identified as a significant advancement over prior art.

The performance and computational properties of the NOWR algorithm were compared to solutions obtained from a Dynamic Programming (DP) algorithm, and NOWR was shown to exhibit superior computational performance with excellent optimization performance. Some cases were identified for which NOWR performed suboptimally, but those cases are thought to occur infrequently. Techniques to identify such situations were suggested for future research.

The chapter concluded with a qualitative discussion on the nature of optimal wind routing, and many interesting properties of optimal wind-routes were identified.

Chapter 5

Strategic Conflict Detection and Resolution

5.1 Introduction

Strategic conflict-detection and resolution (CD&R) algorithms for air-traffic control automation have been under development since the mid-1990's. The term *strategic* means that conflict-detection and resolution are to be performed with a significant look-ahead time considering the effects on the entire trajectory. This is in contrast with tactical conflict-detection and resolution, which would be performed on a shorter time scale and would only consider the immediate effects of the conflict-resolution maneuver. The boundary between tactical and strategic concepts is usually considered to be at about 20 min. Algorithms for both strategic conflict detection and strategic conflict resolution are presented in this chapter that are much more efficient than their predecessors. These functions are both identified in the high-level block diagram (fig. 3.1).

Ground-based ATC automation was the first application to require large-scale CD&R algorithms. As free-flight concepts emerged, airborne-automation tool developers also needed algorithms to efficiently detect and resolve conflicts over a fairly large airspace domain. Conflict detection is usually more computationally demanding than conflict resolution, especially since each trial conflict-resolution maneuver initiates a new round of conflict detection. This can be avoided by using conflict-resolution algorithms that locally resolve conflicts while leaving the rest of the trajectories unchanged, but this is not always

possible or desirable. If localized resolution algorithms are not used, then resolving a local conflict may induce one or more conflicts along the rest of the trajectory. This has been referred to as the “domino-effect” [1]. Another issue is that most conflict-resolution algorithms have been tactical in scope, and have not taken into account trajectory optimization on a global scale. A technique will be introduced in this chapter to address these problems. By modifying the neighboring optimal wind-routing (NOWR) algorithm, it is possible to achieve efficient conflict resolutions that are close to being globally optimal.

In most prior work, conflict-detection and resolution functions are distinctly separate operations, whereas here they are intimately coupled. The approach is to detect and resolve conflicts as the trajectories are computed rather than first computing a set of trajectories and then trying to detect and resolve them simultaneously. This is the essence of performing sequential trajectory optimization while enforcing separation constraints, which has the benefit of greatly reducing the number of computations required to obtain a conflict-free solution. The resulting solution will generally not be the global optimal solution, but it will be a feasible near-optimum solution. Analysis results are presented in chapter 6 to support this claim.

One of the important implications of the sparse-airspace assumption for the air-traffic control problem is that many feasible solutions near the globally optimum solution will have nearly the same cost. This means that the computation of any kind of optimal conflict-resolution maneuvers may not be worth the effort. It is instructive to examine a simple conflict-resolution example to see why this is the case.

For this simple example, the effects of winds, aircraft turn dynamics, and spherical coordinates are not important to the results and are neglected. A kinematic model of an aircraft flying along a straight line from origin to destination at a constant airspeed is adopted. The total flight distance is set at 1,100 n.mi., which is the approximate average flight distance for commercial aircraft flying at 35,000 ft. Conflicts are to be resolved solely by heading changes, and maneuvers are only to be made by one aircraft. The

resolution maneuver model is for the aircraft to change heading at the point where the conflict is detected and then to turn back at an equal angle after the conflict in order to rejoin the original trajectory (fig. 5.1). A minimum separation of 5 n.mi. is required to

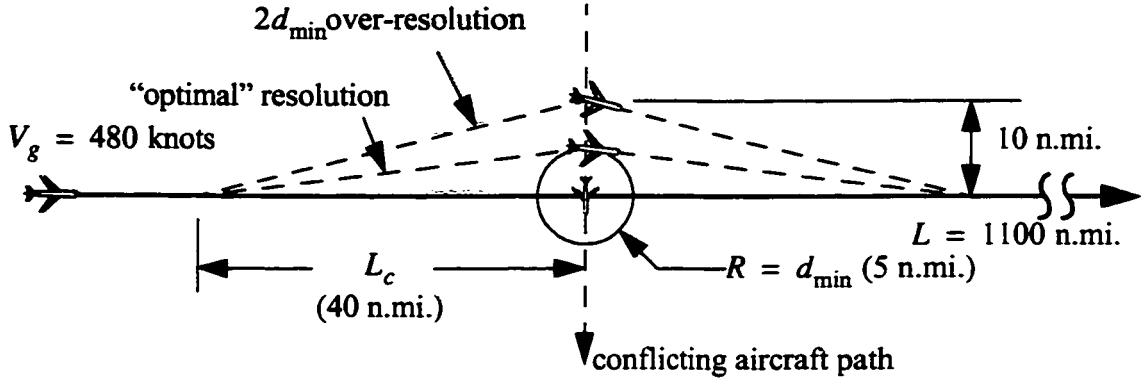


Figure 5.1. A simple conflict-resolution model.

resolve a conflict. Under normal circumstances, a conflict would be detected and a resolution maneuver initiated at a minimum of about 5 min before its predicted occurrence. At 480 knots, this corresponds to 40 n.mi., or 8 times the required separation distance.

The optimal resolution maneuver in terms of minimizing additional path length (or, equivalently, flight time or fuel use for this constant-speed case) is for the aircraft to turn such that the minimum predicted separation is d_{\min} (5 n.mi. in this example). A second suboptimal maneuver is made such that the minimum predicted separation is twice the minimum amount, or $2d_{\min}$ (10 n.mi.). The additional path length for the optimal resolution maneuver over the nominal path is given by

$$L_{\text{opt}} = 2[\sqrt{L_c^2 + d_{\text{opt}}^2} - L_c] \quad (5.1)$$

which, for this example, is $L_{\text{opt}} = 0.62$ n.mi. . The additional path length for the suboptimal maneuver with twice the required separation is given by

$$L_{2d_{\min}} = 2[\sqrt{L_c^2 + 4d_{\text{opt}}^2} - L_c] \quad (5.2)$$

which, for this example, is $L_{2d_{\min}} = 2.5$ n.mi. .

Therefore, a maneuver to create twice the required separation results in nearly four times the additional path distance of the optimal resolution maneuver. These kinds of statistics are often cited to show the potential benefits of optimal conflict resolution. A potential for a 400% reduction in additional path length might seem worth the effort, but taking a broader view of the problem leads to a much different conclusion about the utility of optimal resolution maneuvers.

The additional path length of the suboptimal maneuver over the optimal maneuver, normalized by the total trajectory path length, is given by

$$\frac{\Delta L}{L} \equiv \frac{L_{2d_{\min}} - L_{\text{opt}}}{L} = \frac{\sqrt{(L_c/d_{\text{opt}})^2 + 4} - \sqrt{(L_c/d_{\text{opt}})^2 + 1}}{(L/2d_{\text{opt}})} \quad (5.3)$$

For typical en route trajectories, the denominator of equation (5.3) is large (in this case: $1100/10 = 110$). The additional path length of the suboptimal maneuver over the optimal maneuver is just 1.84 n.mi. over a trajectory of 1100 n.mi., or 0.17%. The corresponding additional flight time of the suboptimal maneuver over the optimal maneuver at a speed of 480 knots is just under 14 sec for a nominal trajectory time of more than 2 hr. This is negligible, especially when compared to the savings that result from the optimization of the nominal trajectory in the first place, as described in detail in chapter 4, where time savings of the order of several percent are common. Note that this is a rather extreme example used to show that even an inefficient resolution maneuver does not result in much of a penalty. In practice, even less of a penalty would be expected.

The conclusion is that specific resolution maneuvers have almost no effect on the optimality of a trajectory. Therefore, one should not worry too much about computing optimal maneuvers if significant penalties in computational effort would be incurred. This lesson is given considerable weight during the development of a fast trial-and error approach to conflict resolution in this chapter.

Conflict-resolution algorithms have received considerable attention over the past decade because of the interest in the free-flight concept [21]. Most research has focused on optimal tactical conflict resolution between two aircraft, where the term *tactical* implies a time scale of about 20 min or less. Some research has examined the more strategic problem of conflict resolution among a set of many aircraft trajectories. Many good approaches have been developed, but none has yet achieved anything close to real-time conflict-free solutions for hundreds or thousands of aircraft. Another drawback to state-of-the-art conflict-resolution concepts is that almost none of them consider the effects of winds, or of optimal wind-routing.

A serious issue with prior approaches to conflict resolution is that none can guarantee that safe solutions are possible under all circumstances [69]. Even the work in verification of conflict-resolution maneuvers by means of hybrid control techniques can only claim that safe resolution maneuvers are possible for a small number of aircraft, and only under simple air-traffic modeling assumptions [70, 23]. The problem of verification is a fundamental challenge for automated conflict-resolution algorithms. The root of the problem lies in the complexity of the air-traffic control system, and in the uncertainty inherent in the system. It is simply not possible at this time to ensure that all possible conflict scenarios can be safely handled; as a result an appeal must be made to probabilistic verification techniques such as Monte Carlo simulation. No one can offer a formal proof that any concept is fail-safe.

The current air-traffic control system suffers from the same problem. The only guarantee of air-traffic control safety in the current system is that it has been demonstrated to be safe empirically: en route mid-air collisions are exceedingly rare. The excellent

safety record of the air-traffic control system has been achieved by heavily constraining air traffic operations so that the conflict situations that arise are easily handled by air-traffic controllers and pilots well in advance. The constraints in place effectively give controllers and pilots plenty of time to resolve any conflicts that arise, and this suggests that a prudent approach to automated conflict resolution should also operate strategically rather than tactically. Even though tactical conflict-resolution algorithms can be simulated and shown to “almost always” be safe, the fact that this cannot be verified will probably preclude the implementation of any tactical conflict-resolution concepts. Instead, a strategic approach must be adopted so that time is allowed in which to handle anomalous situations *when* they occur. This is one of the guiding principles of the strategic approach taken in this dissertation, where conflict free trajectories are computed on a strategic time scale and then presented as flight plan clearances to aircraft.

A review of prior work on the conflict detection problem and the development of a new and efficient approach to conflict detection called the Conflict Grid is presented below. After that, some background on conflict resolution, and the development of a strategic conflict-resolution algorithm based on NOWR are presented.

5.2 Conflict Detection Background

The task of conflict detection is to determine whether all aircraft within a specified spatial and time domain will maintain adequate spatial separation. In general, this requires that aircraft trajectories be discretized as a function of time so that the distances between each aircraft can be computed at each discrete instant of time and compared to the allowable separation distance. Mathematically, the number of point-to-point distance comparisons is expressed as

$$N_{\text{comparisons}} = N_T \cdot \sum_{i=1}^{N_{AC}} (N_{AC} - i) = \frac{N_T(N_{AC}^2 - N_{AC})}{2} \quad (5.4)$$

where N_T is the number of time-steps in the conflict-detection domain, and N_{AC} is the number of aircraft in the domain. The separation comparisons may be expressed as

$$\|\bar{x}_2 - \bar{x}_1\| > d_{min} \quad (5.5)$$

where \bar{x}_i is the position vector of the i th aircraft, $\|\Delta\bar{x}\|$ is the 2-Norm of $\Delta\bar{x}$, and d_{min} is the minimum separation distance.

For small numbers of aircraft, this is not a problem, but as N_{AC} increases, the computational burden grows quadratically. The quadratic growth rate of the brute-force conflict search has led to a search for more efficient conflict-detection algorithms.

Some of the first attempts at developing efficient conflict-resolution algorithms began with the development of the Center TRACON Automation System (CTAS) at the NASA Ames Research Center [17]. Since CTAS was one of the first practical large-scale ATC automation tools to be developed, it was the first to have a need for efficient conflict-detection algorithms.

A paper by Issacson and Erzberger [71] describes a practical approach to conflict detection for CTAS using heuristics such as altitude pruning and time-skipping to limit the search space when looking for conflicts. Aircraft trajectories are first computed in time and space using 10-sec time-steps before applying heuristics to limit the conflict search. At 480 knots there are about 1.3 n.mi. between steps. Altitude pruning is the process of eliminating potential conflict pairs that are never flying at the same altitude. The use of altitude pruning typically removed 60-80% of all possible trajectory pairs from the detailed conflict search. Time-skipping is used to limit the number of points that need to be checked for conflicts. Time skipping uses the fact that if two aircraft are separated by a large distance at some point in time, then there is no need to check every 10 sec for a conflict because it would be physically impossible for a conflict to occur until the aircraft could travel the current separation distance. A further reduction in computation is achieved by only computing the sum of the squares of Δx and Δy when both of those terms are individually less than the required separation.

The computational performance of this algorithm still grows as $O(n^2)$, but the actual number of computations is proportionally reduced through the applied heuristics so that on a Sun Ultra workstation (340 MIPS), about 800 aircraft could be checked for conflicts within 10 sec.

The paper by Sridhar and Chatterji [10] looks at aircraft conflict detection using principles from computer science search and sort algorithms. A sorting-based algorithm first partitions the airspace into a Cartesian grid. Aircraft are stored in the grid cells by using a hashing function to convert the x - and y -coordinates into grid indices. The grid matrix is then unwrapped into a single vector of bin numbers. The vector of bin numbers is sorted using an efficient sorting algorithm (e.g., Heapsort, Quicksort) so that repeated bin numbers may be easily located. From the repeated bin numbers, the corresponding grid locations and the associated aircraft numbers are uniquely determined. This algorithm is shown to be $O(n \log n)$.

For the Quicksort method, the average number of computations, C_q was shown to be

$$C_q = 8(M \cdot n + 1) \log_2(M \cdot n) \quad (5.6)$$

where M is the maximum number of bins occupied by any single aircraft trajectory and n is the number of aircraft. In reference [10], an example is given where the average ground speed is assumed to be 500 knots, and the conflict look-ahead time is 20 min so that the maximum number of bins per aircraft for 5n.mi. by 5n.mi. grid cells is given by

$$M = \left\lceil \frac{500 \text{ n.mi./hr} \cdot (20/60) \text{ hr}}{5 \text{ n.mi.}} \right\rceil = \lceil 33.33 \rceil = 34 \quad (5.7)$$

For $n = 100$ and $M = 34$, the proportionality factor in equation (5.6) is 2,708, while the logarithm term is only 11.73 for a total number of computations equal to 31,765. Even though this is an $n \log_2(n)$ algorithm, the proportionality constant in the area of interest still makes this computationally intensive.

The next algorithm considered in reference [10] is accumulator-based. The mapping of aircraft location to bin number is the same as in the sorting algorithm, but in this case, each time an aircraft is mapped into a bin, the number of aircraft in that bin is incremented by one. The search for conflicts can then be limited to those bins with more than one aircraft. Once the potential conflict bins have been located, the corresponding aircraft are directly checked for conflict.

The average number of computations required by the accumulator algorithm was shown to be

$$C_a = M \cdot n + (I_{max} + 1)(J_{max} + 1) \quad (5.8)$$

where M and n are as previously defined, and I_{max} and J_{max} are the maximum number of grid cells in the airspace domain of interest. In reference [10], an 800-n.mi. square region is examined for a 20-min conflict look-ahead time so that $M = 34$ once again. In this case, I_{max} and J_{max} are both 160 for 5-n.mi. grid cells. For 100 aircraft, this requires 29,321 computations, which is only marginally better than the sorting algorithm. The real benefit of the accumulator algorithm is realized for greater numbers of aircraft since the number of computations grows as $O(n)$ instead of as $O(n \log n)$.

The approach taken in reference [72] is to use a geometric hashing algorithm to identify clusters of aircraft that may be in conflict with one another. Once the clusters have been identified, they are sent as inputs to the conflict-resolution algorithm, which then uses brute-force conflict-search techniques to check potential resolution maneuvers for conflicts. The geometric hashing algorithm used for identifying clusters is similar to the accumulator method of reference [10], but with the additional discretization of the time dimension to create a 4-D grid of the airspace. The grid cells are sized according to the minimum separation requirement (5 n.mi.), and presumably the time-grid size is to be set short enough that the fastest aircraft would not travel through more than the minimum separation distance during one time interval (a value of 0.1 min was used in the example given in the paper). Since conflicts might occur with aircraft in neighboring grid cells, of which there are 27 in 3-D space or 81 in 4-D space (adding the time dimension to the three

spatial dimensions), the neighboring cells are also checked. In addition to looking at the immediately neighboring grid cells, certain other non-conflict aircraft are identified that are sufficiently close to the cluster that should be treated as constraints during the conflict-resolution phase. In reference [10] it is suggested that the cluster identification algorithm tends to take time linear in the number of aircraft, but the conflict detection in the cluster conflict-resolution phase is still $O(n^2)$, where n is the number of aircraft in a cluster.

Additional research has been conducted in the field of conflict-probability prediction which is closely related to conflict detection [46]. Instead of making binary decisions on whether a conflict will occur, a probability is assigned to each potential conflict so that optimal decisions can be made regarding when to initiate conflict-resolution maneuvers. The system design approach taken in this dissertation uses several forms of feedback to mitigate the effects of disturbances on trajectory-prediction accuracy so that the effects of trajectory-prediction uncertainties are greatly reduced. To account for remaining system uncertainty, the principles of two-aircraft conflict-probability prediction are generalized to the case of multiple aircraft conflicts, and conflicts with uncertain weather constraints.

5.3 Conflict Grid Method

An efficient conflict-detection algorithm called the Conflict Grid (CG) method is now introduced. The CG method is similar to the algorithm presented in Ref. [72], though a few key differences result in the CG method being essentially computationally free in the context of sequential route optimization. The deterministic version of the CG is first developed, followed by the introduction of a stochastic version of the CG.

5.3.1 Deterministic Conflict Grid

The basic idea behind the CG method is to store optimal aircraft trajectories in a 4-D grid space (three spatial dimensions and time) as they are computed by setting the values of the corresponding grid cells to one (binary “on,” or “true”) (fig. 5.2). After the development of the deterministic CG algorithm, an extension to stochastic conflict detection is presented where grid cell values represent the probability of an active constraint in that cell and may take on any value between zero and one. The grid cell

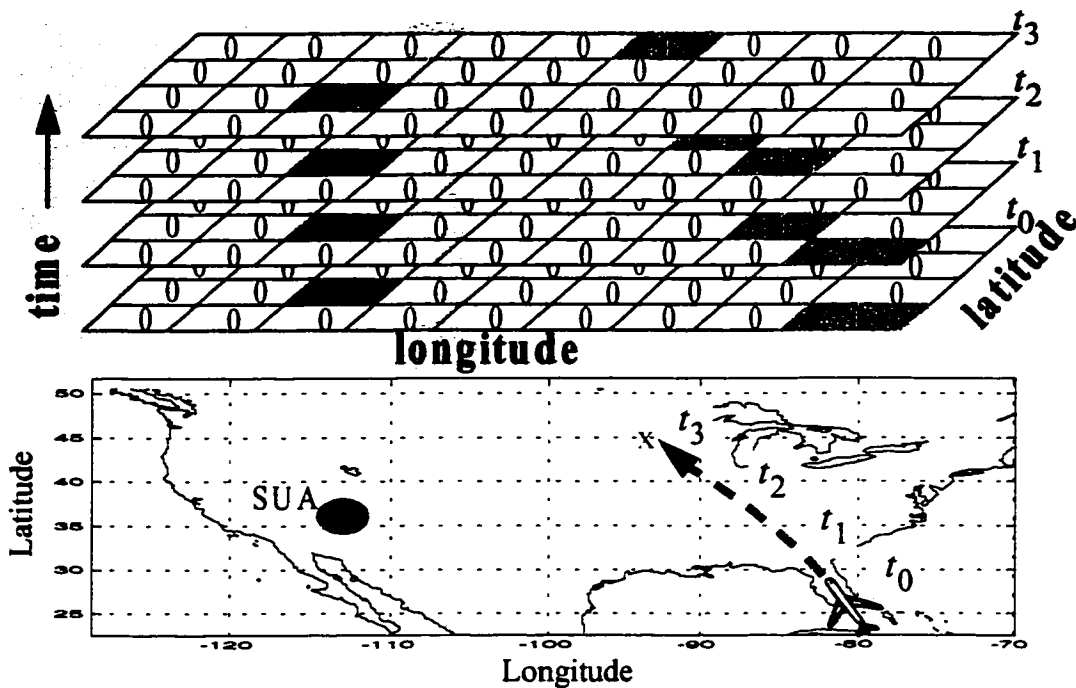


Figure 5.2. The Conflict Grid method of conflict detection.

dimensions are set according to the allowable aircraft separation limits. If a grid cell is found to be already occupied, then it is immediately known that the current trajectory will be in conflict, and conflict-resolution maneuvers are initiated. Note that regions of bad weather and special-use airspace may easily be incorporated into the conflict grid by setting the corresponding grid cell values to one for any of these areas of restricted airspace. The computational benefit of the CG approach is that it completely eliminates pairwise distance computations.

The airspace is first partitioned into a 4-D grid of space and time. The coordinates used here are longitude, latitude, altitude, and time, but any convenient set of independent coordinates may be used. The airspace is partitioned by creating a set of separate 3-D grids (longitude, latitude, time) for each discrete flight level. If the flight level structure of the current air transportation system is used, then each flight level extends vertically by 2,000 ft, but plans for the reduced vertical separation minimum (RVSM) will likely reduce the size of each flight level to 1,000 ft. The spatial longitude and latitude grid dimensions are

left as a variable parameter so that simulation studies can be made to determine such things as optimization performance and airspace capacity as a function of the grid dimensions. The time grid size, Δt , is linked to the spatial grid size, Δx , such that an aircraft flying at the fastest expected ground speed would not travel more than Δx distance in Δt time. For example, if the fastest expected ground speed was 600 knots, and $\Delta x = 5$ n.mi. , then the time resolution would be set to $\Delta t \leq 30$ sec .

Clearly, a conflict grid cannot extend from the current time to infinity. This is solved by creating the CG with a rolling time window. The window is chosen to span the appropriate amount of conflict look-ahead time. The maximum flight time for an aircraft across the continental United States is less than 7 hr, so this can be used as the maximum bound on the range of the CG time-window. With these definitions, the CG may be mathematically represented by a 3-D matrix for each flight level, FL_p , as follows

$$CG_{FL_p} = CG(i, j, k) \quad \begin{cases} 0 \leq i \leq \left\lceil \frac{(\tau_{max} - \tau_{min})}{\Delta \tau} \right\rceil \\ 0 \leq j \leq \left\lceil \frac{(\lambda_{max} - \lambda_{min})}{\Delta \lambda} \right\rceil \\ 0 \leq k \leq \left\lceil \frac{(t_{max} - t_{min})}{\Delta t} \right\rceil \end{cases} \quad (5.9)$$

$$\Delta \tau \equiv \frac{K_x \cdot \Delta x}{\cos(\max(\lambda))}$$

$$\Delta \lambda \equiv K_x \cdot \Delta x$$

$$\Delta t \equiv \frac{(t_{max} - t_{min})}{\left(\left\lceil \frac{(t_{max} - t_{min}) \cdot V_{g_{max}}}{\Delta x} \right\rceil + 1 \right)}$$

where τ and λ are longitude and latitude, respectively, t is time, Δx is the spatial grid dimension, K_x is a unit conversion constant, $V_{g_{max}}$ is the maximum anticipated ground speed, and $\lceil a \rceil$ is defined as the nearest integer to a rounded up toward positive infinity. The maximum and minimum grid dimensions are then chosen for the particular needs of the problem being solved.

The amount of memory required for the CG is now approximated. The Continental United States extends approximately 2,500 n.mi. from east to west, and 1,500 n.mi. from north to south. For a 7-hr conflict look-ahead time, a time grid resolution of 30 sec, and grid spacing of 5 n.mi., the memory required for the Conflict Grid per flight level is given by

$$\mu_{CG} = \frac{2,500 \text{ n.mi.}}{5 \text{ n.mi.}} \cdot \frac{1,500 \text{ n.mi.}}{5 \text{ n.mi.}} \cdot \frac{(7 \text{ hr} \cdot 3600 \text{ sec/hr})}{30 \text{ sec}} \cdot \frac{1 \text{ byte}}{8 \text{ bits}} = 15.75 \text{ Mbyte} \quad (5.10)$$

Therefore, the amount of memory required for the conflict grid at each flight level is less than 16 megabytes, which is not particularly challenging for current-day technology. Inexpensive computers are available with gigabytes of RAM.

The procedure for conflict detection is now described (fig. 5.3). At the beginning of the conflict-detection loop, the CG is cleared so that the value of each grid cell is set to zero. The next step is to store any weather constraints or special-use airspace constraints in the CG by setting the corresponding constrained airspace grid cell values to one. Next, the Active Aircraft List (AAL) is passed through sequentially to compute a predicted trajectory for each aircraft. The trajectory is generally computed and stored as a set of vectors of three spatial coordinates versus time. These vectors are then interpolated to the discrete time values of the conflict grid. As the values are interpolated, the corresponding values of the conflict grid are checked. If the grid cell values are zero, then that means there are no prior aircraft occupying that cell, and that the airspace of that cell is not restricted by bad weather or other regulation. In that case, the value of that grid cell is set to 1 to signify that it is occupied by the current aircraft. If any of the trajectory points for the current aircraft are found to be in conflict at any of the grid cells, then the trajectory must be modified and checked again. The process continues until all aircraft in the active list have been given conflict free trajectories. At the conclusion of the AAL, the whole process may be restarted.

The CG algorithm as discussed to this point has a subtle feature that must be addressed. If just the occupied grid cells are marked as such, it may occur that aircraft in

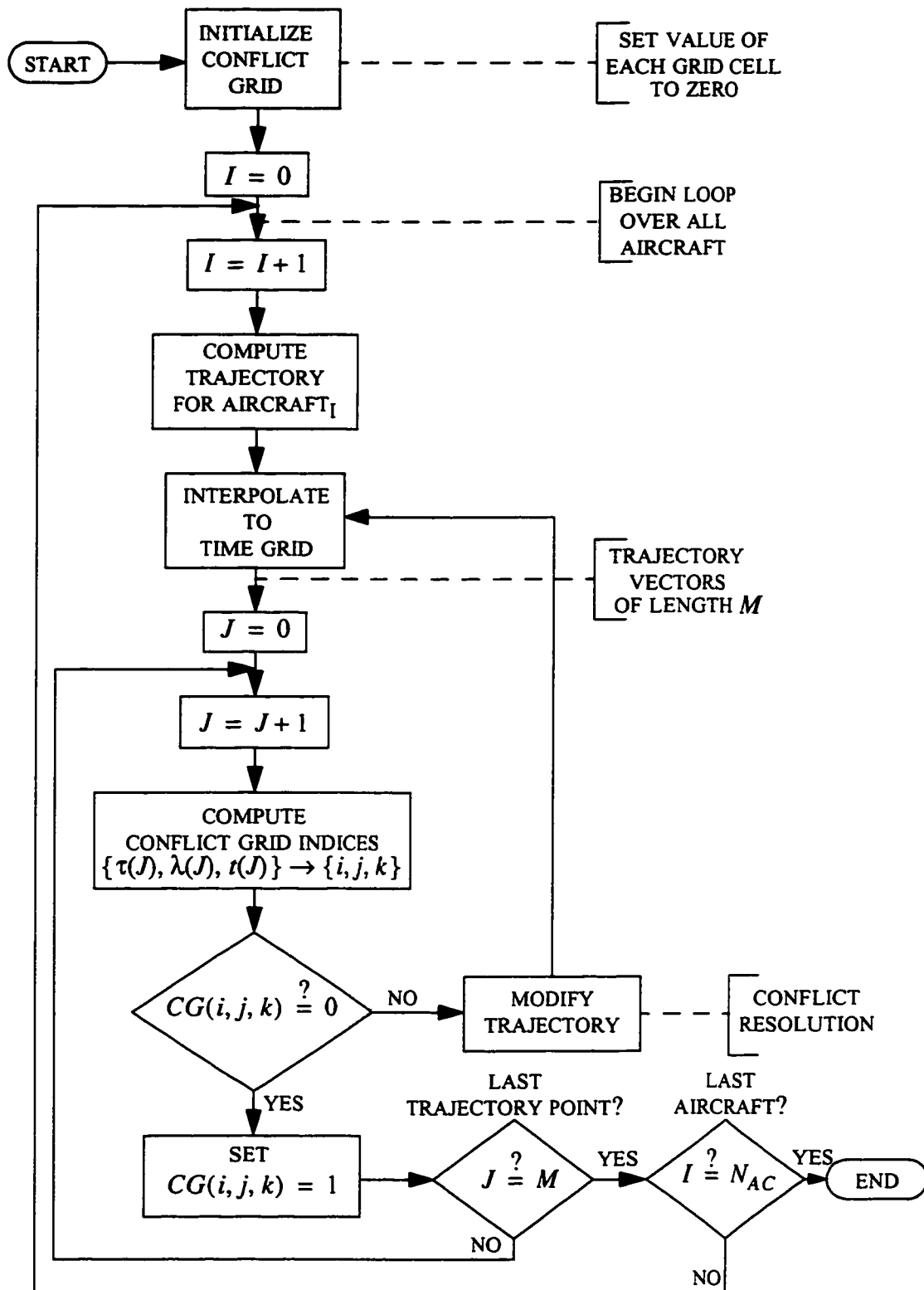


Figure 5.3. Flowchart of the sequential Conflict Grid algorithm.

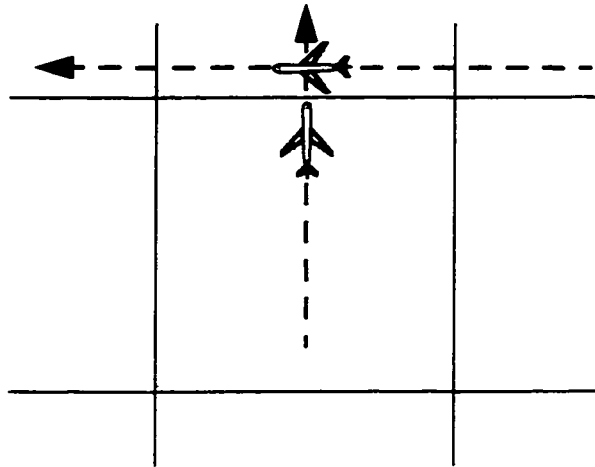


Figure 5.4. Illustration of a missed conflict at a grid cell boundary.

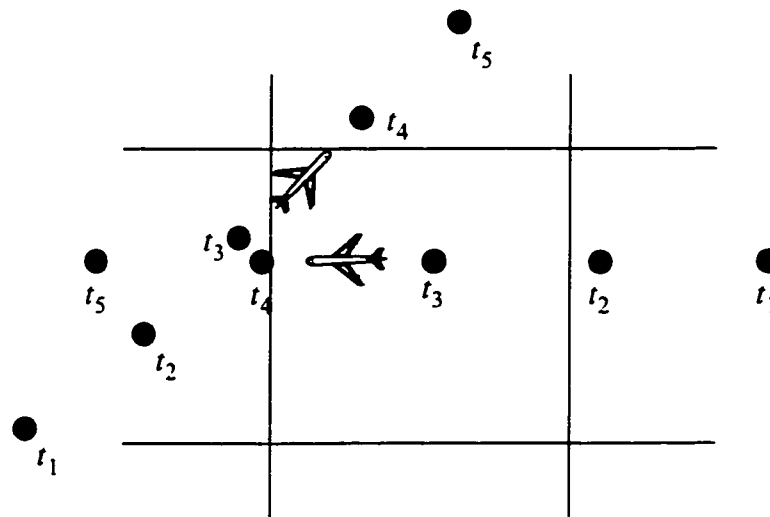


Figure 5.5. Illustration of a missed conflict resulting from time discretization.

neighboring grid cells are in conflict with one another because the basic CG algorithm does not include any space between neighboring grid cells (fig. 5.4). Another situation that may occur is that the discretized time-steps of a trajectory may overstep a grid cell so that a real conflict is not identified (fig. 5.5). These subtleties are not particularly difficult to address.

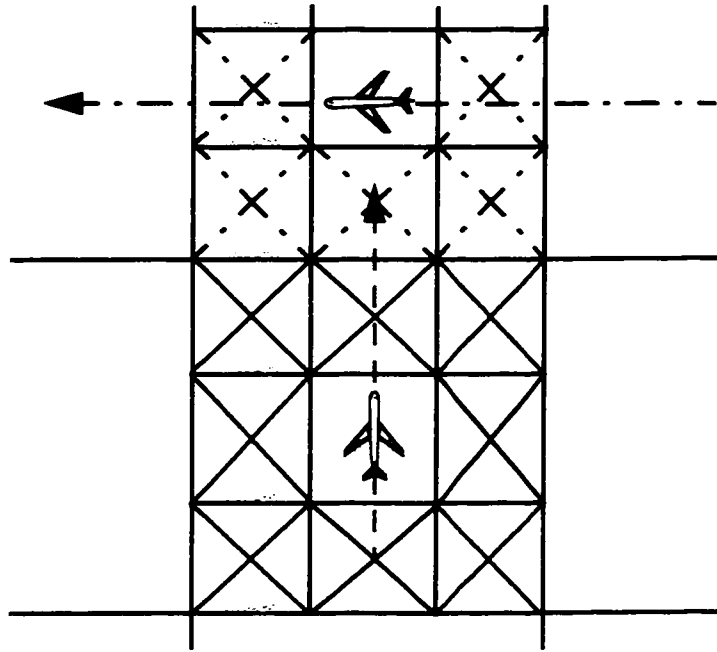


Figure 5.6. Grid-cell buffering to eliminate missed conflicts.

One of the simplest algorithmic solutions to these kinds of conflicts is to use a grid cell buffering technique. With grid-cell buffering, the CG cells would be made half the desired size, but then each time a trajectory point was stored in the grid, the neighboring cells would also be marked as being occupied (fig. 5.6). By doing this in all 3 dimensions for the single flight-level problem (x, y, t) , the types of conflicts mentioned above would all be detected. The cost of grid cell buffering is that the amount of memory required for the CG for any given aircraft spacing would increase by a factor of 8 (2^3). For the example given in equation (5.10), the amount of memory required would increase from 15.75 Mbytes to 126 Mbytes, which is still well within the memory capabilities of current-day computers. There would also be some minor computational costs incurred by requiring that additional grid cells be set for each trajectory point, but these costs would still be negligible when compared with the other computational costs involved in trajectory optimization.

Another valid approach is to accept that these types of scenarios might occasionally surface, and to resolve the conflicts tactically as they occur. Even though inter-aircraft separation may be predicted to be less than the minimum allowable separation, the CG method ensures that, on average, there is enough airspace available to support all of the aircraft.

Finally, a stochastic extension would generalize the basic CG method and would also eliminate many of the subtle types of missed conflict alerts addressed above without having to resort to grid cell buffering. This stochastic extension is now presented.

5.3.2 Stochastic Conflict Grid

Aircraft trajectory prediction, atmospheric prediction, and many other facets of the NAS are better characterized as stochastic systems than as deterministic systems. Errors in wind models can lead to rather large errors in down-range trajectory predictions. Storm forecast errors can either completely miss actual storms that develop, or may predict storms that never materialize. This suggests that the air-traffic control optimization problem should be treated as a stochastic optimization problem and that the probabilistic nature of conflict detection should be considered. In a complete treatment, this would involve recasting the cost function in equation (2.5) as an expected value function.

Instead of completely reformulating the optimization problem, it is possible to consider the probabilistic nature of predicted conflicts with a fairly simple extension to the deterministic CG algorithm. The extended algorithm is referred to as the Stochastic Conflict Grid (SCG). The basic idea is to store the probability that *at least one* active constraint exists in any given grid cell rather than using a binary *yes* or *no* value. An active constraint is any entity that requires exclusive use of the airspace, such as an aircraft, a weather storm cell, special-use airspace, or even an aircraft trailing wake vortex. This generalizes the CG technique by incorporating important conflict-probability concepts from prior research on pairwise conflict-probability estimation [73, 46]. Some minor increase in computational overhead is incurred for the CG method itself, but as with the grid cell buffering technique, the overhead is negligible when compared with the other

computational costs involved in conflict-free trajectory optimization. Apart from the conflict-detection algorithm, computing and maintaining models of aircraft trajectory-prediction accuracy might add significant overhead. The additional computational burden may be reduced by using simple probability models for aircraft trajectory-prediction uncertainty.

The probability that a constraint, c_i , will be active in a particular grid cell may be modeled as a Bernoulli trial so that the probability is given by

$$P_i \equiv p(\text{constraint } i \text{ is active}) \quad (5.11)$$

and the probability that the constraint will not be active is simply

$$p(\text{constraint } i \text{ is not active}) = (1 - P_i) \quad (5.12)$$

For a set of n possible constraints, an easy way to compute the probability that any one of those constraints will be active is to compute the probability that no constraints will be active and then to subtract this from one, as follows:

$$\bar{P} \equiv p(\text{at least one active constraint}) = 1 - \prod_{i=1}^n (1 - P_i) \quad (5.13)$$

where Π is the serial product operator.

Rather than storing the individual probabilities for all of the potential constraints, it may be desirable to maintain a running total probability that at least one constraint will be active. This is easily shown to be given by

$$\bar{P}_i = 1 - (1 - P_i)(1 - \bar{P}_{i-1}) \quad (5.14)$$

where \bar{P}_i is the running total probability that any constraint up to and including constraint c_i will be active, and \bar{P}_{i-1} is the running total probability that any constraint will be active before considering the new constraint, c_i .

If trajectory-prediction errors are modeled as having Gaussian distributions, then the case of a two-aircraft conflict using the SCG would be identical to the method developed by Erzberger et al. [46, 73] for pairwise conflict-probability estimation. Following this method, the choice of whether to make a conflict-resolution maneuver would be based on the minimization of the expected cost of resolution. In essence, low-probability conflicts are ignored because the expected cost of resolving a conflict that may not occur is too high. Over time, the probability increases to the point that delaying a conflict-resolution maneuver would result in a higher expected cost because short-term conflict-resolution maneuvers are less efficient than strategic resolution maneuvers.

By the use of the SCG, this same conflict-probability method is generalized to the case of multiple aircraft and to the case of conflicts with any other type of constraint. Imagine a case where two aircraft are predicted to converge at some instant an hour in the future (fig.

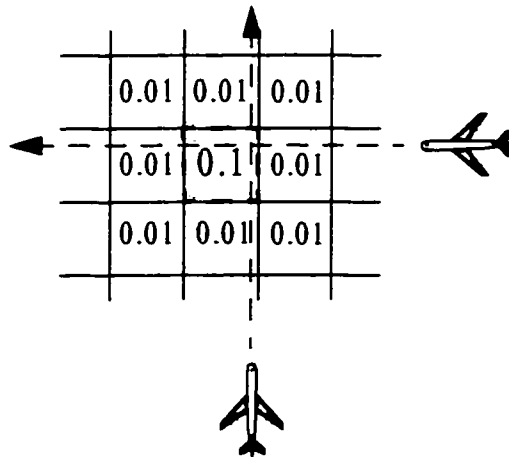


Figure 5.7. Example of low predicted conflict probability for two aircraft.

5.7). The probability that either aircraft will be in that particular grid cell (at that instant) is then quite low because of the growth rate of trajectory-prediction error. In this case, it would be more efficient to wait to see how the potential conflict develops. Now imagine that many aircraft are predicted to converge at the same point (fig. 5.8). The probability that *any one aircraft* will be in that grid cell now increases, possibly to the point that the next aircraft predicted to be in that grid cell would have to make a conflict-resolution

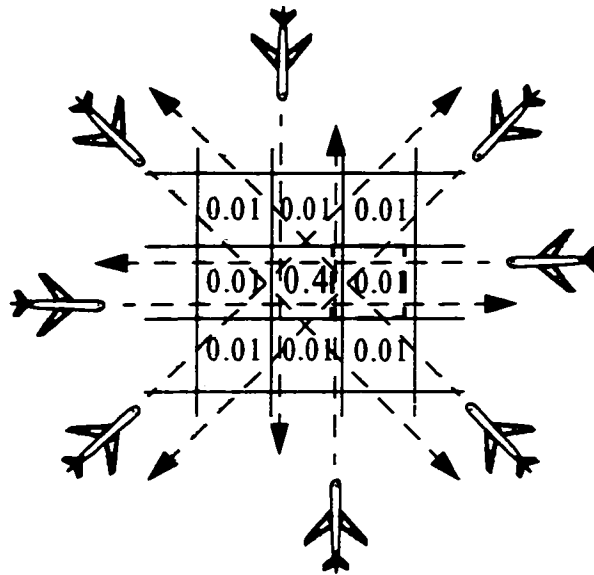


Figure 5.8. Example of higher conflict probability with multiple aircraft.

maneuver. In this way, the SCG method handles the case of predicted en route aircraft congestion naturally. This generalizes the notion of traffic flow management to a concept that covers all of the en route airspace, not just the terminal airspace near airports.

Weather storm cells are notoriously difficult to predict with any accuracy until they have actually developed into storms, and even then their prediction beyond 30 min is not accurate. Storm predictions are usually made such that a region of airspace can be identified as having some heightened potential for storm development. This heightened potential could be stored in the CG cells in a region (fig. 5.9). This would result in fewer aircraft being permitted to pass through that region because of the elevated probability of storm conflict. As the time of the storm prediction neared and the prediction became more accurate, the conflict probabilities in the corresponding grid cells would increase to the point where no aircraft would be permitted in those cells. This would be a gradual process so that sudden drastic maneuvers around unpredictable storm cells would be greatly reduced. The resolution thresholds could be modified empirically to obtain good performance.

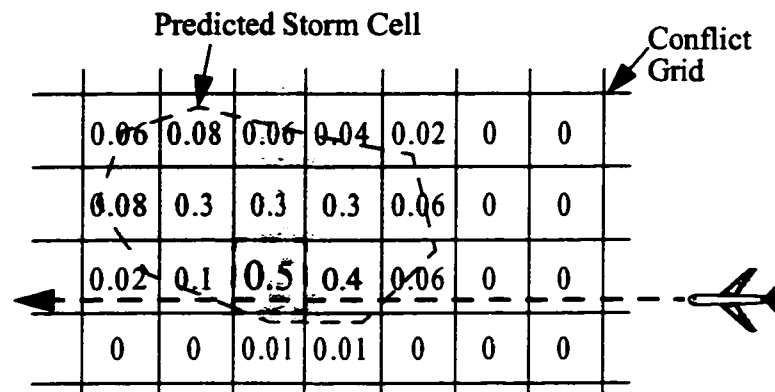


Figure 5.9. Conflict probabilities between storm cells and aircraft.

The case of special-use airspace is even easier to consider. Once special-use airspace is activated, it is known with certainty that a constraint is active in that region so the probabilities of the corresponding grid cells would be set to 1. This would block those cells from being occupied by any aircraft. The same principle applies to any regions of blocked airspace, whether it be for security, noise abatement, or other purpose.

The SCG method has been introduced, but will not be demonstrated as part of this dissertation. Although conceptually straightforward, the added software complexity for implementation would go beyond the scope of this research. The added complexity does not come from the SGC itself, but from the maintenance of trajectory-prediction probabilities for all aircraft and for weather cells. Techniques for computing trajectory-prediction probabilities are well known, but they require considerable additional software implementation effort for simulation. Instead, the deterministic CG technique is used for simulations and analysis to demonstrate how the CG method might work in practice, but it is anticipated that the SGC method would be a superior choice for practical implementation.

5.4 Conflict-Resolution Background

As discussed in the Introduction, many approaches to automated conflict resolution are tactical methods for computing optimal resolution maneuvers between a pair of

aircraft or a small number of aircraft [7, 11, 23, 46, 70, 74-76]. These optimal-resolution algorithms vary in dynamic model complexity, optimization goals, and the controls utilized for conflict resolution, but they are all intended for tactical conflict resolution in free-flight systems and do not consider strategic trajectory optimization. Winds, and their effects on optimal trajectories, are essentially neglected in all of these tactical conflict-resolution methods. A good survey of much of this work up to 1997 is presented in reference [69].

Any of these tactical conflict-resolution algorithms might be converted into strategic algorithms by using them within a fast-time air-traffic simulation to generate sets of conflict-free optimal routes which would be given to aircraft as trajectory clearances. If such conflict-free sets of trajectories could be computed in real-time, and with a significant amount of lead-time, then they might be useful in a practical air-traffic control system. However, many of the most difficult issues such as trajectory optimization, efficient conflict detection, and real-time computation would still have to be addressed.

Potential-field or force-field techniques adapted from robotic path planning have been applied to aircraft conflict resolution [76, 77]. The essence of these techniques is to model aircraft as potential sources which are repelled by other aircraft or blocked airspace. These techniques have been used to compute basis resolution maneuvers for other conflict-resolution or verification techniques [23]. One significant problem with potential field methods is that they do not account for trajectory optimization in variable winds. Perhaps the techniques could be extended to account for optimal wind-routing, but this has not yet been attempted. Another problem is that cases may arise where non-realizable solutions are obtained, so some form of post-processing would be required.

Various strategic conflict-resolution algorithms for the multi-aircraft problem have also been examined. One approach to limiting the complexity of conflict-detection and resolution is to apply the concept of *clustering* in order to identify smaller geographical subsets of aircraft for detailed conflict resolution [6, 45, 72]. Each cluster is then examined in more detail for specific aircraft conflicts. Without system uncertainty, this approach can

greatly reduce the $O(n^2)$ performance of conflict-detection and resolution. As uncertainty is introduced, the clusters tend to join together until ultimately the entire airspace must be considered as one large cluster, and their computational benefit is significantly reduced.

Genetic Algorithms (GA) have been applied to conflict-detection and resolution in an attempt to compute near-optimum solutions to the air-traffic control optimization problem in real-time. The benefit of GAs is that they can quickly identify good candidate solutions to the multi-aircraft conflict-resolution problem. The challenge is to increase the precision of the search space without increasing the computation time to such an extent that real-time solutions are not possible. Because of this trade-off with GAs, all past approaches have greatly limited the scope of the problems that were solved. For example, conflict look-ahead times have been set to 30 min or less, the airspace domains have been much smaller than the continental United States, winds (and therefore optimal wind-routing) have been neglected, dynamic models have been either simplified or neglected entirely by using straight-line segment trajectories, and reduced traffic levels have been assumed. Even with these simplifying assumptions, real-time performance has not been achieved with GA.

Another approach to multi-aircraft conflict resolution has been proposed based on the solution to an approximate convex semi-definite program [78]. In this approach, the air-traffic control problem with separation constraints is posed as a nonconvex, quadratically constrained quadratic optimization problem, and then approximated by a relaxed problem. In the relaxed problem, instead of looking for a specific decision variable to minimize the original nonconvex problem, a random decision variable is considered and the optimization goal is modified such that the expected cost over the random decision variable is minimized. The chief benefits of this approach are that it leads to computable lower bounds on the solution to the original problem; it also provides direction for searching for random feasible solutions near the optimal solution to the original problem.

The philosophy of the semi-definite programming approach is similar to that followed in this dissertation: determine an optimal solution independent of the constraints, then

look for feasible solutions near the unconstrained optimum. However, there are some issues not addressed by the semi-definite programming approach. The most serious of these is that the optimization of aircraft trajectories in winds has not been accounted for so that straight-line trajectory solutions are assumed to be optimal and the optimization goal is considered to be the minimization of deviations from requested speeds and headings. This has the effect of greatly reducing the number of decision variables, but comes with the price of excluding optimal wind-route solutions. Another drawback is that the computational effort apparently still grows as $O(n^2)$ so that large numbers of aircraft cannot be handled by this approach. The semi-definite programming approach appears to be well suited to finding optimal resolution maneuvers for a relatively small number of aircraft, but as already discussed, the optimization of short-term conflict-resolution maneuvers is not relevant to the strategic optimization of trajectories.

Hybrid control techniques have been applied to the development and verification of provably safe conflict-resolution maneuvers [23, 70]. The objective of these approaches is to address the problem of guaranteeing that conflicts can safely be resolved in tactical situations when uncertainties are present. Much progress has been made, but the complexity of the air-traffic control problem has limited the application of hybrid verification techniques to conflict scenarios involving only a few aircraft. The hybrid automaton models have also been limited to simple kinematics and basic conflict-resolution maneuvers. Hybrid techniques hold promise for providing a rigorous proof of safety for free-flight concepts, but much work still remains for the practical application of these ideas to air-traffic control.

A discussion of prior art in conflict resolution has been presented here to give a broad perspective on some of the various approaches to solving this problem. The classification of different solution approaches is difficult since many of them are combinations of several different techniques. The main conclusion to be drawn from the results of prior work in conflict-resolution techniques is that all are too computationally limited to achieve real-time optimization for hundreds or thousands of aircraft over a domain as large as the United States. Because of the computational limitations, each approach has

made simplifying assumptions that neglect important aspects of the problem such as optimal wind-routing.

The goal of this dissertation is to develop a real-time method for finding conflict-free optimal-wind routes for all aircraft in the continental United States domain. This means that a conflict-resolution method which can compute optimal wind-route resolution maneuvers is required. The NOWR algorithm presented in chapter 4 has properties that make it ideally suited to this purpose. By perturbing unconstrained NOWR solutions, it is possible to find conflict-free solutions that are close to being wind-optimal. The NOWR perturbation technique is now described in more detail.

5.5 Neighboring Optimal Wind-Routing Perturbation Conflict Resolution

In the approaches to conflict resolution discussed above, two main complications were responsible for causing conflict resolution to be a computationally intensive function. The first is that maneuvers are optimized by varying the trajectories of all aircraft involved. The second is that conflict-resolution maneuvers are optimized for specific conflict geometries. The first complication has already been avoided by taking a sequential optimization approach so that previously planned aircraft trajectories remain fixed. This greatly reduces the number of possible resolution maneuvers. A corollary to the fact that the airspace is relatively sparse is that many trajectory perturbations will resolve a given conflict and will have almost the same cost. This suggests that one need not take into account the specific geometry of a conflict situation in order to get close to the globally optimal solution. Instead, one must only find a feasible solution nearby the unconstrained optimal solution. This is a reasonable hypothesis that is confirmed later with simulation results. Applying this knowledge, the approach to conflict resolution is to compute perturbations to the optimal wind trajectory for a conflicting aircraft until one is found to be conflict free.

Once a conflict has been detected by means of the CG method, possible conflict resolutions may be computed in parallel if so desired. In the most basic version of the CG method, when a conflict is detected, it is not known with which aircraft there is a conflict,

nor is the particular geometry known. The approach is to perturb the current aircraft trajectory in both of the possible horizontal directions until a resolution is found. If resolutions in these two directions are computed in parallel, one may keep track of the lowest cost maneuver that resolves the conflict so that it may be chosen as the solution. In this dissertation, only heading changes in the horizontal plane have been examined, but it is also possible to look for speed perturbations or altitude perturbations in parallel. The primary use of horizontal conflict resolution at constant speed for en route aircraft is justified based upon fuel efficiency and passenger comfort and considerations.

With the NOWR algorithm, there is a natural mechanism in place to generate smooth perturbation trajectories that are close approximations to a wind-optimal resolution maneuver. For reference, the relationship for computing the wind-optimal heading, equation (4.54), is repeated here

$$\theta_{\text{opt}}|_{\text{RLL}} = \theta_{\text{nom}} + \delta\theta = \pi + \frac{\partial\theta}{\partial y} \cdot y + \sum_{i=1}^{N_w} \left(\frac{\partial\theta}{\partial u_{yi}} \cdot u_{yi} + \frac{\partial\theta}{\partial v_{bi}} \cdot v_{bi} \right) \quad (5.15)$$

where $\delta\theta$ is the heading perturbation, the various partial derivative terms are the neighboring optimal feedback gains, the u_{yi} terms are the along-track wind shear in the cross-track direction at each grid point, i , and the v_{bi} are the normalized cross-track winds at each grid point, i .

By introducing the concept of the *pseudo-shear*, a control is put in place to cause perturbations in the computed heading command. The concept is simply to identify the nearest along-track wind shear term (u_{yi}) to the conflict location and then to modify the wind shear at that location by adding a pseudo-shear (fig. 5.10). If parallel processors are used, then perturbations in both directions may be examined at the same time until a resolution is found. If a resolution is found in both directions, the one with the lowest flight time is selected. If parallel processing is not available, one may compute perturbations in each direction sequentially until a solution is found. This is the approach that has been taken here.

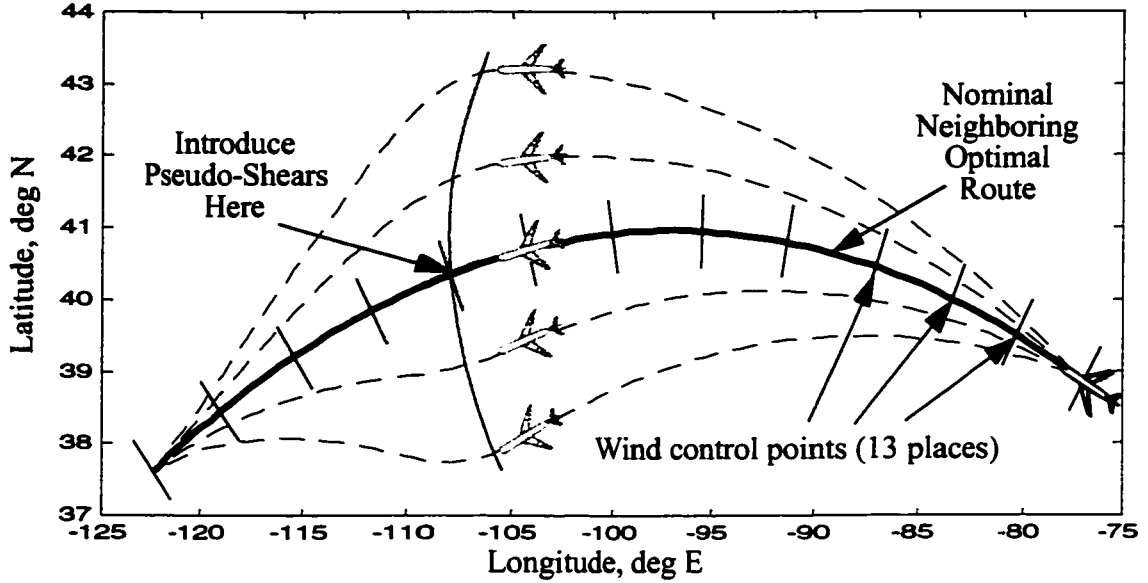


Figure 5.10. Introducing pseudo-shears for conflict resolution.

The nearest wind control point is approximated by using the time at which the conflict is detected by the CG. The time is divided by the total trajectory time to determine the proportion of the trajectory that has already been travelled. Since the wind control points are evenly spaced along the nominal trajectory, the closest wind control point can be approximated using the trajectory time ratio. For N_w wind control points, the following expression gives the index of the wind control point nearest to a conflict that is to occur at t_{conflict} along a trajectory that is predicted to take t_{total} time

$$i_w = N_w - \text{round} \left[(N_w - 1) \left(\frac{t_{\text{conflict}}}{t_{\text{total}}} \right) \right] \quad (5.16)$$

Other approximations may be used, but the one shown in equation (5.16) has been demonstrated to work well. In practice, one may choose to limit the value of i_w so that wind shears are not introduced at either the first or the last wind control point. The reason

for doing so is that the effect of shears at these points is not as strong as at other points and may cause excessive iteration. In that case, the following value would be used

$$\bar{i}_w \equiv \begin{cases} 2 & i_w = 1 \\ i_w & 2 \leq i_w \leq (N_w - 1) \\ (N_w - 1) & i_w = N_w \end{cases} \quad (5.17)$$

Once the nearest wind control point is selected, the total shear at that location is given by

$$\hat{u}_{yi} = u_{yi} + u_{ps} \quad (5.18)$$

where u_{ps} is the pseudo-shear value.

The value of u_{ps} to use while iterating to find a conflict free trajectory must be determined. If too small of a value is chosen, then excessive iterations would result. If too large of a value is chosen, then the first resolution maneuver that is found may be too large, and would be inefficient. Because of the lack of a priori knowledge about conflicts, it is not possible to determine an exact pseudo-shear value to use. Ideally, each different conflict situation would use a different amount of pseudo-shear during conflict-resolution iterations. In order to determine a good average value to use, parametric studies may be performed on real air-traffic data. This has been done, and the results are presented later in chapter 6. One of the benefits of using a normalized problem formulation is that a single average pseudo-shear value tends to work well over a wide range of problems.

5.6 Summary

Methods for real-time strategic conflict-detection and resolution were presented in this chapter. The CG technique presented was shown to be virtually computationally free for use in a sequential optimization algorithm. The SCG was introduced as an extension to the basic CG technique to generalize the concept of conflict detection to constraints with uncertainty.

The optimization of tactical conflict-resolution maneuvers was shown to be a negligible factor in the context of strategic trajectory optimization so that much simpler resolution techniques may be used instead of computationally expensive optimal techniques. This led to the development of a trial-and-error approach to conflict resolution using the NOWR algorithm as the basis for conflict-free trajectory generation. The performance of this approach is evaluated in chapter 6.

Chapter 6

Simulation and Performance Evaluation

6.1 Introduction

This chapter presents the simulation system that has been developed to demonstrate many of the algorithms in this dissertation, and to perform simulation analyses. After describing the design and operation of the simulation system, some results of two parametric studies will be given as examples of what can be done with the algorithms that have been developed. The first study is an analysis of the effect of changing aircraft separation limits on airspace capacity. The next study determines the required computation rate to achieve real-time performance, also as a function of changing aircraft separation limits.

The approach is to develop algorithms and perform simulation studies in MATLAB. Then, by porting the core NOWR functions to the C-language, it is possible to determine the compiled speed that could be achieved in a compiled version of the entire simulation. This approach is possible because the NOWR functions form the core computational effort required by the sequential optimization approach as shown in equation (3.17).

6.2 Simulation Design

The simulation is designed to be able to determine the computational requirements of the complete optimization system, including optimal wind-routing and conflict-detection and resolution. The simulation is all conducted in a horizontal plane to keep the amount of

data and computation manageable for the development system running in MATLAB either on an IBM ThinkPad T20 notebook computer (750 MHz, 256Mb RAM), or on a SUN UltraSparc 60 workstation (450 MHz, 256Mb RAM). As discussed in chapter 3, the 2-D results can be extrapolated to assess the performance of a full 3-D system. Spherical-Earth coordinates are used to ensure that any computations required to convert between local normalized coordinates and the global spherical coordinates are taken into account. Winds are modeled by importing data from the RUC (appendix 2). This ensures that the same level of fidelity of wind data are used in simulation as are available for real air-traffic control applications [67]. Because of the strong effect that weather storm cells and special-use airspace (SUA) might have on system performance, the capability to model these phenomena is included in the simulation. Finally, since uncertainty in weather models and aircraft positions has a strong effect on optimization and computational performance, these uncertainties can also be modeled within the simulation.

6.2.1 Empirical Air-Traffic Data

Input data for the simulation have been taken directly from recordings of aircraft traffic by means of Enhanced Traffic Management System (ETMS) data files. Obtaining complete sets of ETMS data for any particular day can be a challenging process because of the difficulty in acquiring access to the data, the difficulty of storing a large amount of data (hundreds of Megabytes per day), and the difficulty of processing the large data sets for use in a particular simulation. For these reasons, primarily one complete data set has been collected, processed, and used to run the simulation studies.

The ETMS data were collected in the form of what are called *orig* files. A custom processing routine, written by NASA colleague Jim Murphy, was run on the *orig* files to select the data fields and particular aircraft of interest. Murphy's code examined the *orig* files and pulled out the aircraft identification number, aircraft type, departure time, origination airport, destination airport, filed altitude, and filed airspeed. Because of the manner in which filed flight-plan messages are entered into the ETMS data stream, it may take several hours from the start time of data collection before the filed flight plans for all

active aircraft have been obtained. This is because aircraft already in-flight at the start of data collection will have already filed flight plan messages prior to the start of the data collection.

The collected data set began at 0000 UTC on 10 August, 2001 (2000 EDT on 9 August 2001) and continued through 2345 UTC on 10 August, 2001 (1900 EDT on 10 August 2001). The data included flights at all altitudes. The complete data set was filtered to produce a file with only those aircraft filed to fly at FL350, and only those aircraft traveling between U.S. origins and destinations. The international flights were filtered out because they typically only reside in U.S. airspace only a short time after departure or before arrival and would not affect the simulation results. A histogram of the number of scheduled departures for those aircraft flying at FL350 is given in figure 6.1. The initial

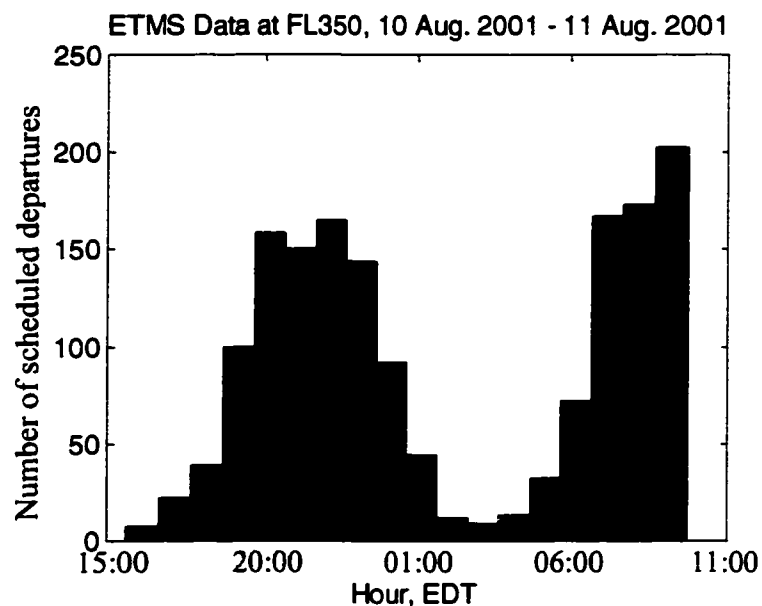


Figure 6.1. Histogram of the number of aircraft scheduled to depart for FL350.

ramp-up of departures is an artifact of the data processing mentioned earlier, but within a few hours of the data collection start time, the data are representative of the scheduled departures.

6.2.2 Simulated Air Traffic

Simulated traffic can be randomly generated to match the statistical traffic profiles between different city pairs found in the ETMS data set discussed above. Some researchers have used randomly generated traffic based on uniform distributions or other artificial distributions, but the unrealistic traffic patterns resulting from these data sets would not be applicable to real-world air-traffic analysis. Real air-traffic patterns involve local concentrations of aircraft at peak traffic times, and some city pairs are much more congested than others. Because of these concentrations, real air-traffic patterns tend to be more computationally challenging for conflict detection and resolution algorithms than uniformly random traffic patterns. A histogram of origin/destination pairs may be determined from the empirical ETMS data. A random route-generation function is easily created to reproduce the same histogram as is now explained.

From the ETMS data, each flight is numbered consecutively from 1 to N , where N is the number of unique flights during the given input hour. These numbers are then normalized by dividing through by N . A uniform random number is then generated between 0 and 1, and the nearest value in the flight list is selected. This technique produces a random flight distribution that matches the empirical ETMS data (fig. 6.2). Larger amounts of traffic could be simulated by increasing the number of flights between city pairs at the same distribution as is found at the normal traffic levels.

6.2.3 Simulator Code

The simulator software was written in MATLAB m-functions and m-files to take advantage of the rich set of high-level functions and overloaded operators in MATLAB, especially those that perform vectorized operations. Since MATLAB is partly an interpretive language, the computation time results for a MATLAB simulation can be expected to be much slower than for an equivalent simulation run in a compiled version of the code. For this reason, the measurement of computational effort was abstracted from the particular machine and software implementation by using the FLOPS function to estimate the floating point operations required for each run. The FLOPS estimates were

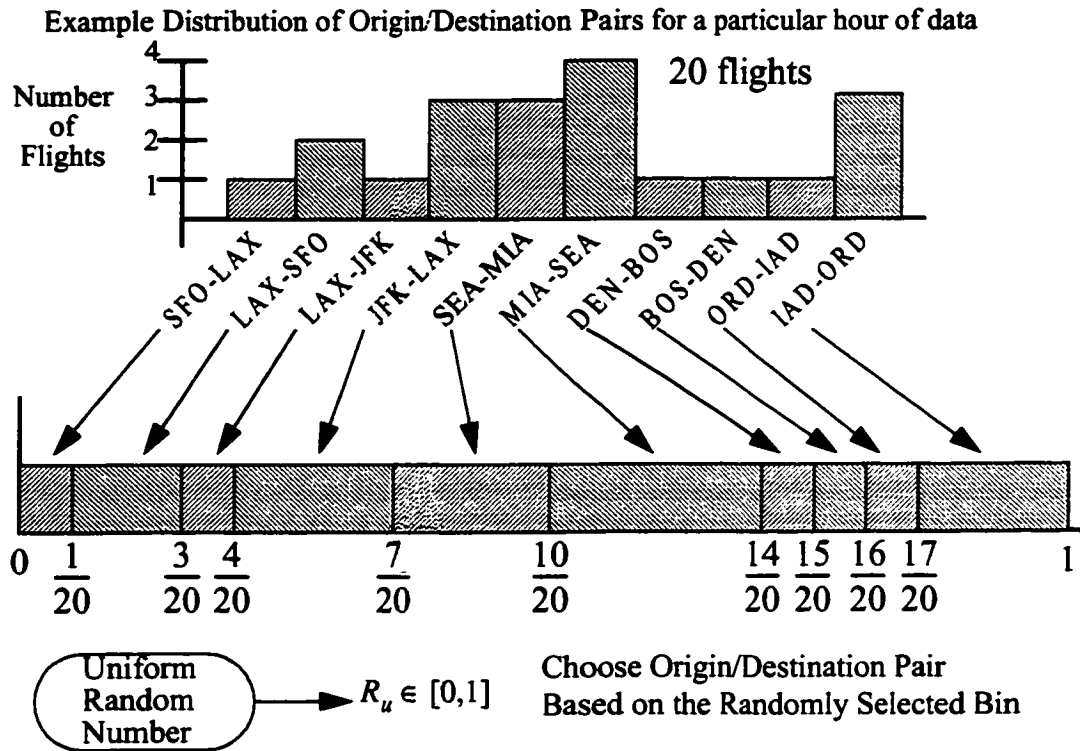


Figure 6.2. Generating random routes based on empirical data.

broken down for each subsection of the simulation program so that results dependent on random variables, such as the number of conflicts experienced during a simulation run, could still be computed.

The following is a narrative of the simulation m-file. The narrative is given to explain the process and hopefully not get lost in the details of variable initialization and other trivialities.

- 1) Initialize Variables and Functions
 - Set Buffer Time, T_{buffer}
 - Set Conflict Look-ahead Time, T_{conflict}
 - Initialize Wind File
- 2) Initialize Aircraft List
 - Load Data Directly from ETMS File
 - or, Generate Random Aircraft Based on ETMS Histogram

```
3]   Pre-Generate Aircraft Trajectories [Some aircraft will have been
    scheduled to be enroute at the start of the simulation. These
    aircraft are required to populate the airspace as it would be
    populated in the real system when the optimization code would be
    turned on.]

4]   Initialize the Conflict Grid
    •   Compute and Set the Conflict Grid Dimensions
    •   Initialize Memory

5]   Initialize Simulated Weather Cells & Special-use Airspace

6]   BEGIN REAL TIME LOOP

7]       Clear Conflict Grid Memory

8]       Update Positions of Wx Cells and SUA

9]       Store Wx Cells and SUA in the Conflict Grid

10]      Clear Wx Cells and SUA that have Dissipated

11]      BEGIN ACTIVE AIRCRAFT LOOP

12]          Update Active Aircraft List (AAL) -- Add new aircraft
            that are scheduled to depart within the next TBUFFER
            time, and remove aircraft that have arrived at
            destination.

13]          Update Current Aircraft Positions with Simulated
            Random Longitudinal Perturbations

14]          Initialize Conflict Resolution Loop Variables

15]          BEGIN CONFLICT RESOLUTION LOOP

16]              Generate Neighboring Optimal Wind Route with
                Requested Conflict Resolution Perturbation

17]              Interpolate Trajectory onto Time-Grid
                Coordinates [only need to interpolate from
                current time out to current time plus TCONFLICT,
                the conflict look-ahead time]

18]              Compute Conflict Grid Indices for Each
                Trajectory Point [use simple hashing function to
                compute grid indices as a function of position
                and time coordinates]

19]              Determine Conflict List [make a list of
                trajectory points for which the conflict grid
                has more aircraft than are allowed in each cell,
                or is occupied by a Wx cell or SUA]

20]              BEGIN CONFLICT LIST LOOP

21]                  Filter Conflict List [Ignore terminal area
                    conflicts. Mark certain conflicts as
                    unresolvable by horizontal means if the
                    resolution delay is larger than a chosen
                    threshold]
```

```

22]          Compute Nearest Wind Control Point [find
              the nearest wind control point to the
              conflict so that the trajectory may be
              perturbed at that location]

23]          Set Conflict Resolution Perturbation
              [Note: The simulation software currently
              runs in series, so it iterates between
              starboard and port resolution maneuvers
              until one is found that resolves the
              conflict. With a parallel processor, both
              starboard and port resolutions could be
              computed at the same time. In a further
              generalization, many other types of
              resolution maneuvers could be computed in
              parallel so that the minimum-cost
              resolution could be selected.]

24]          If a Conflict Has Been Detected, Break Out
              of Conflict List Loop ==> GOTO Line 16 [If
              a conflict has been detected, we don't
              care to look for additional conflicts
              along the same trajectory]

25]          END CONFLICT LIST LOOP

26]          Store Trajectory in Conflict Grid [Increment
              values of corresponding Conflict Grid cells by
              one]

27]          Increment Aircraft Counter

28]          END CONFLICT RESOLUTION LOOP

29]          Increment Simulation-Time
              [The time is incremented such that it will take just under
               $T_{\text{buffer}}$  time to run through the entire active aircraft list]

30]          If at End of Simulation, Quit.

31]  END REAL-TIME LOOP

```

6.2.4 Qualitative Simulation Description

The simulation proceeds through the provided set of aircraft data according to the basic flowchart presented in chapter 3, which is repeated here for convenience (fig. 6.3). There are some additional subtleties which must be addressed in order to maintain a reference to simulation time.

The T_{buffer} parameter sets the amount of time in the future for which aircraft not yet scheduled to depart will be included in the active aircraft list. This buffer is added to allow for the time to run through the complete active aircraft list, which will be a finite amount

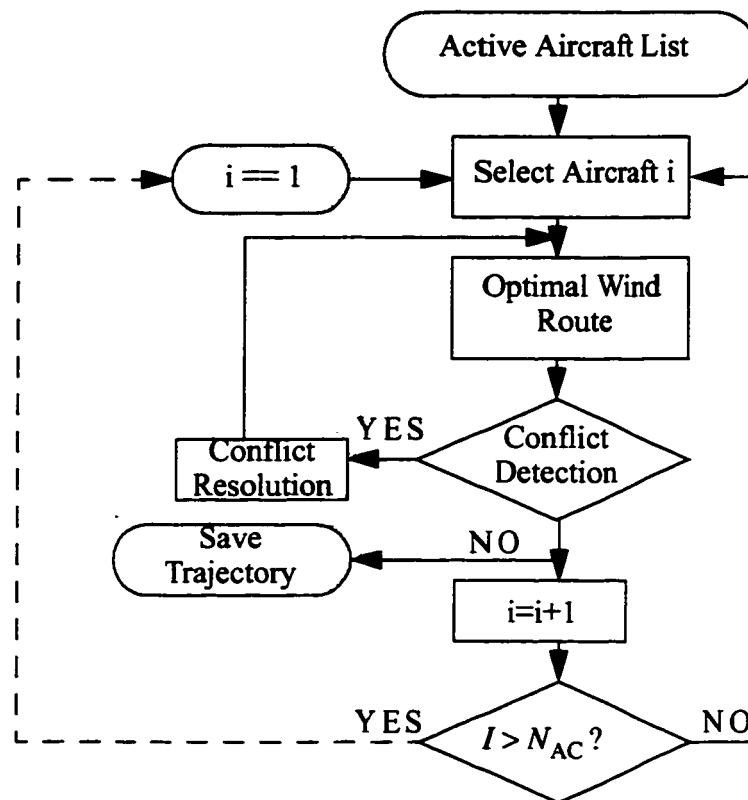


Figure 6.3. High-level simulation flowchart.

of time in practice. If the aircraft scheduled to depart in this buffer time were not included in the active list, then these aircraft would end up departing before being assigned a trajectory. In all the simulation studies presented here, the buffer time has been set to $T_{\text{buffer}} = 0.5 \text{ hr}$. The simulation is then set to run through the active aircraft list in just under T_{buffer} so that it runs as fast as possible, but does not take longer to run through the active aircraft list than T_{buffer} .

Once the simulation has run through a complete pass of the active aircraft list, which has been set to take just under T_{buffer} time, the active aircraft list is refreshed by adding all aircraft that have now come within the T_{buffer} time window, and by removing all aircraft that have arrived at their respective destinations. The computation process through the refreshed active aircraft list begins anew, and continues until the specified simulation end has been reached. This means that during any given simulation, multiple passes are made

through the active aircraft list, which is always changing as aircraft are added and removed from the list. This process can be observed in the above pseudo-code listing.

6.3 Analysis and Parametric Studies

The simulation code has been designed to easily allow algorithm parameters to be varied to study their effects on the optimization performance and computational efficiency of the algorithm. Among the variable parameters are the following:

- Allowable minimum aircraft separation distance
- Number of aircraft (may be varied using random route generation function)
- Conflict time horizon (look-ahead time)
- Ground speed uncertainty (wind errors, 4-D control errors, etc.)

The following outputs are of interest in evaluating the sequential optimization approach:

- Optimization performance: actual cost versus minimum cost
- Airspace capacity
- Computational effort: required safe recomputation rate

The results of a few analysis studies are presented here in order to demonstrate the capabilities of the simulation system, and to present some new results obtained by simulating the algorithms and concepts developed in this dissertation. The purpose of the first study is to determine the conflict-resolution perturbation parameter that best balances computational efficiency and optimization performance. Once this basic algorithm parameter has been set, two analysis studies are presented. The goal of the first study is to show the effects of reducing legal separation limits on airspace capacity. The goal of the second study is to analyze the computational performance of the entire optimization system.

6.3.1 Adjustment of Conflict-Resolution Perturbation Parameters

The first simulation study has been performed to determine the best value of pseudo-shear perturbation, u_{ps} , to use in the rest of the simulations (refer to sec. 5.5). The procedure is to vary the amount of conflict-resolution perturbation used on each conflict iteration and to evaluate the effects on computational effort, optimization performance,

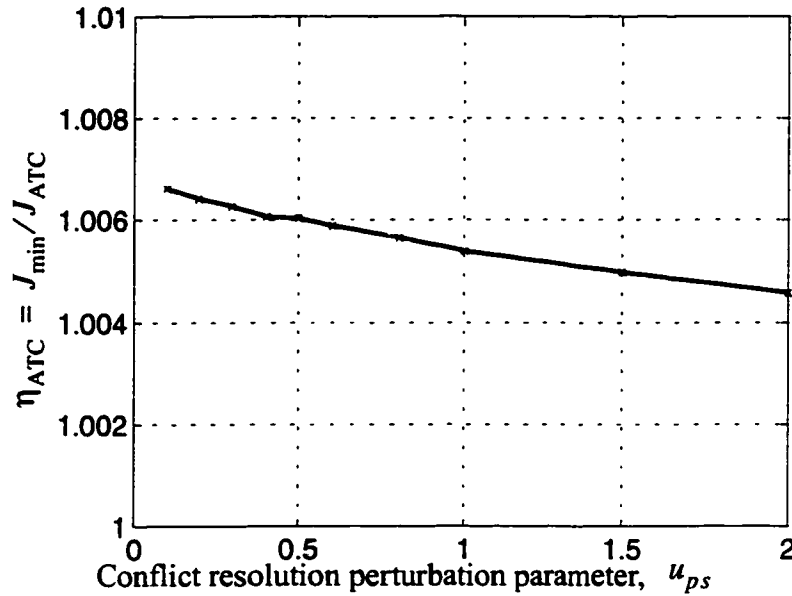


Figure 6.4. Optimization performance vs. perturbation parameter.

and predicted airspace capacity. If the perturbation is too small, excessive iterations would result; if the perturbation is too large, the efficiency of the resulting solutions would be expected to suffer. The specific parameters and data used for this study are presented in table 6.1. The first plot (fig. 6.4) is of the optimization performance versus u_{ps} . The performance is measured in terms of the efficiency parameter, defined as

$$\eta_{ATC} \equiv \frac{J}{J^*_{NOWR}} \quad (6.1)$$

where J is the total flight time over all aircraft in the simulation after conflict resolution has been performed, and J^*_{NOWR} is the sum total flight time of neighboring optimal wind routes for all aircraft when conflicts are ignored. Notice that the values of the efficiency parameter are greater than 1. This is because the value of J^*_{NOWR} is computed from the trajectories of each aircraft as the aircraft are first introduced in the simulation. Recall that the NOWR trajectories are normalized, and a fixed number of wind control points are used to compute the neighboring optimal solutions for any path distance. This results in higher-

resolution wind data being used for shorter trajectories. Therefore, as an aircraft follows the original NOWR trajectory, successively higher-resolution data are used when the solutions are recomputed. On average, this results in better optimization performance being achieved than is predicted from the initial NOWR solution computed for each aircraft. In other words, the ultimate flight time recorded for an aircraft is typically less than that predicted by the first NOWR solution; this is why the efficiency parameter may be greater than 1.

Table 6.1. Simulation parameters for conflict perturbation analysis.

| Parameter | Description | Value(s) |
|--------------------------|--|------------|
| u_{ps} | Normalized pseudo-shear perturbation | 0.1 -- 2.0 |
| T_{conflict} | Conflict look-ahead-time | 6.5 hr |
| T_{buffer} | Take-off time buffer. Aircraft scheduled to take off within the next T_{buffer} time are included in the Active Aircraft List to permit time for the processing of the list | 0.5 hr |
| Δx_{grid} | Average size of conflict grid cell on a side | 10 n.mi. |
| y_s | Normalized shear computation distance | 0.08 |
| Wx File | ruc2.T21Z.grb2f02 (RUC2, 2100 UTC, GRIB format, 2 hr forecast) Date: 14-Feb-2001 u & v wind component data from 225mb (225 mb is approximately 36,000 ft) | N/A |
| ETMS Data | Enhanced Traffic Management System data. All aircraft with flight plans filed at FL350 between US origins and destinations were selected from the data file. Data selected were origin, destination, filed airspeed, filed departure time. Date: Friday, August 10, 2001 Total # of Aircraft: 1283 | N/A |

The important trend in figure 6.4 is that the optimization performance degrades as the resolution parameter increases. This is to be expected since larger trial perturbations may over-resolve conflicts to some degree. The decline in optimization performance is approximately linear with increasing u_{ps} , but with perhaps a slight steepening of the slope after a value of $u_{ps} = 0.5$.

The number of onerous conflicts versus the perturbation parameter is a good measure of the computational effort required for any given value of u_{ps} (fig. 6.5). An onerous

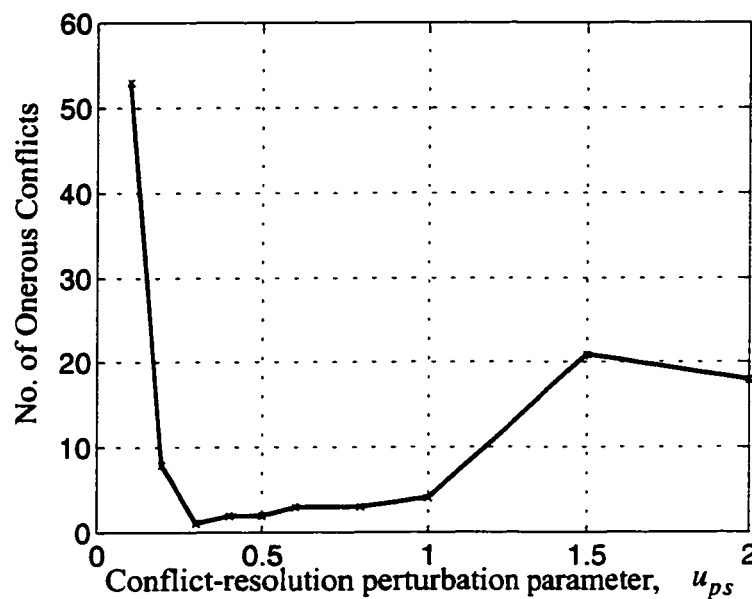


Figure 6.5. Number of onerous conflicts vs. conflict-resolution parameter.

conflict is one that has exceeded a threshold number of conflict iterations and has been deemed to be impractical to resolve by horizontal maneuvers alone. In these cases, which are few, other resolution means would be used, such as speed variations, altitude variations, departure holds, or some combination of these. Simulating the details of these different resolution maneuvers is not difficult, but falls outside the scope of this dissertation. In this simulation study, when the perturbation parameter is set too low, the number of onerous conflicts climbs rapidly because too many iterations are required to

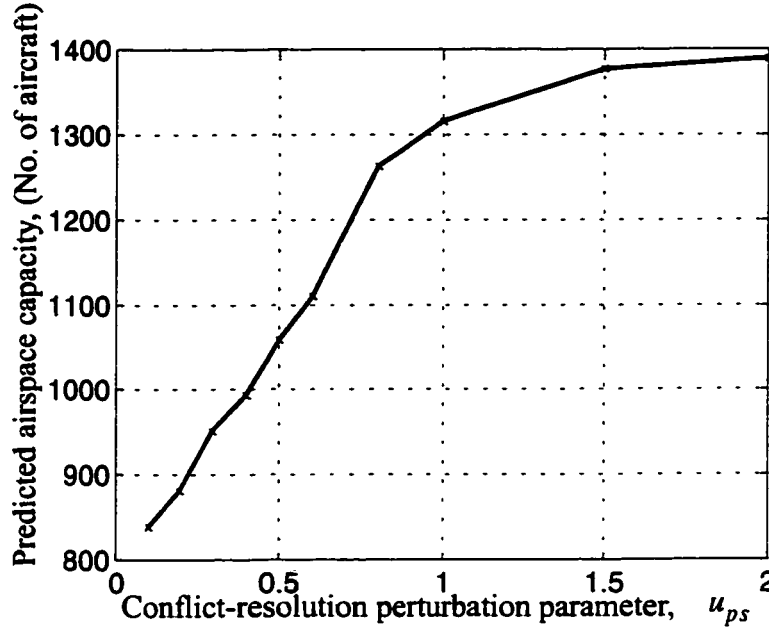


Figure 6.6. Flight-level capacity vs. conflict-resolution parameter.

resolve each conflict. The point where the number of onerous conflicts clearly begins to rise is for values of $u_{ps} < 0.3$. Most values above this range appear to be acceptable, but there is a noticeable increase above $u_{ps} = 1.0$.

Finally, the predicted capacity shows a rather steep increase as a function of u_{ps} (fig. 6.6). This plot is another measure of the computational effort required to obtain a solution. As the computational effort increases, the airspace capacity metric (see sec. 3.3) predicts lower capacity limits. The steep drop in predicted capacity ends at values above $u_{ps} = 0.7$. Note that in this case, the capacity metric is not being used as a representation of actual physical capacity, but instead is being used as a measure of computational effort. The apparent drop in capacity as measured by the capacity metric occurs because the number of conflict iterations increases when the perturbation parameter is too small.

Balancing optimization and computational performance leads to an acceptable range of values for u_{ps} between about 0.3 and 0.7. A value in the middle of this range of

$u_{ps} = 0.5$ has been set for use in the remaining simulation studies because it has a good blend of optimization performance and computational efficiency.

6.3.2 Airspace Capacity versus Separation Limits

An analysis study has been conducted to determine the potential airspace capacity as a function of required aircraft separation. The conflict-grid dimensions are used as a means of adjusting the aircraft separation limits since only one aircraft is permitted within each conflict grid cell (sec. 5.3). The conflict-grid dimensions are varied, and then simulations of the same ETMS data set discussed in the previous section are conducted. The airspace capacity model can then be used to estimate the airspace capacity for any given conflict-grid dimension.

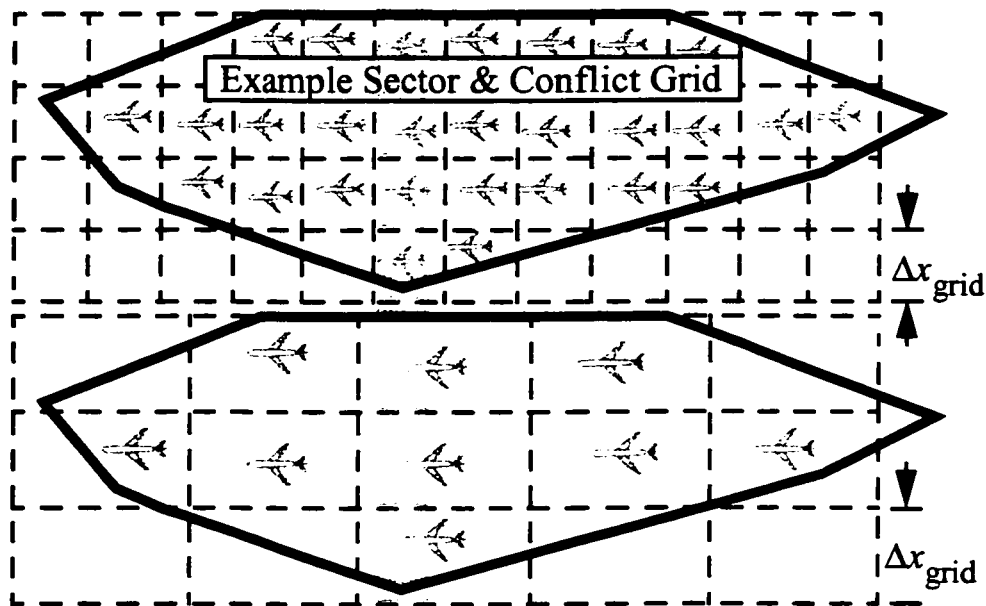


Figure 6.7. Varying average airspace sector density.

An equivalent interpretation of the results is to examine the effect of average airspace sector loading on capacity (fig. 6.7). In the current air-traffic control system, each airspace sector is permitted to handle a certain maximum number of aircraft because an air-traffic controller is only expected to handle so many aircraft within a given airspace region. The

threshold depends on a variety of factors, but is essentially an airspace density limit. This interpretation of the results demonstrates how the simulation capability developed in this dissertation may be applied to the benefit of the current air-traffic control system.

Table 6.2. Simulation parameters for airspace capacity analysis.

| Parameter | Description | Value(s) |
|--------------------------|--|------------|
| u_{ps} | Normalized pseudo-shear perturbation | 0.5 |
| T_{conflict} | Conflict look-ahead-time | 6.5 hr |
| T_{buffer} | Takeoff time buffer. Aircraft scheduled to take off within the next T_{buffer} time are included in the Active Aircraft List to permit time for the processing of the list. | 0.5 hr |
| Δx_{grid} | Average size of conflict grid cell on a side | 5 - 15 nmi |
| y_s | Normalized shear computation distance | 0.08 |
| Wx File | ruc2.T21Z.grb2f02 (RUC2, 2100 UTC, GRIB format, 2-hr forecast) Date: 14-Feb-2001 u & v wind component data from 225 mb (225 mb is approximately 36,000 ft) | N/A |
| ETMS Data | Enhanced Traffic Management System data. All aircraft with flight plans filed at FL350 between U.S. origins and destinations were selected from the data file. Data selected were origin, destination, filed airspeed, filed departure time. Date: Friday, 10 August 2001 Total No. of Aircraft: 1,283 | N/A |

The parameter values and data sources used in this study are given in table 6.2. The average conflict grid square dimensions are varied over the range $5 \text{ n.mi.} \leq \Delta x_{\text{grid}} \leq 15 \text{ n.mi.}$. This is referred to as an average grid dimension because the actual conflict grid is set in terms of degrees of latitude or longitude, and the longitude dimensions vary as a function of latitude.

The capacity model was described in detail in sec. 3.3. The model describes the expected total number of conflict-resolution iterations, \hat{Y}_N , as a function of the number of aircraft, N . Two parameters, C_0 and C_1 , must be determined based on fitting the model to empirical simulation data. The model is repeated here for convenience:

$$\hat{Y}_N \equiv C_1 \ln \left(\frac{C_0}{C_0 - N} \right) \quad (6.2)$$

As described, the parameters of the model in equation (6.2) are to be adjusted to fit simulation data in a minimum least-square error sense. An example of a fit to empirical data for a single pass through the active aircraft list is shown in fig. 6.8.

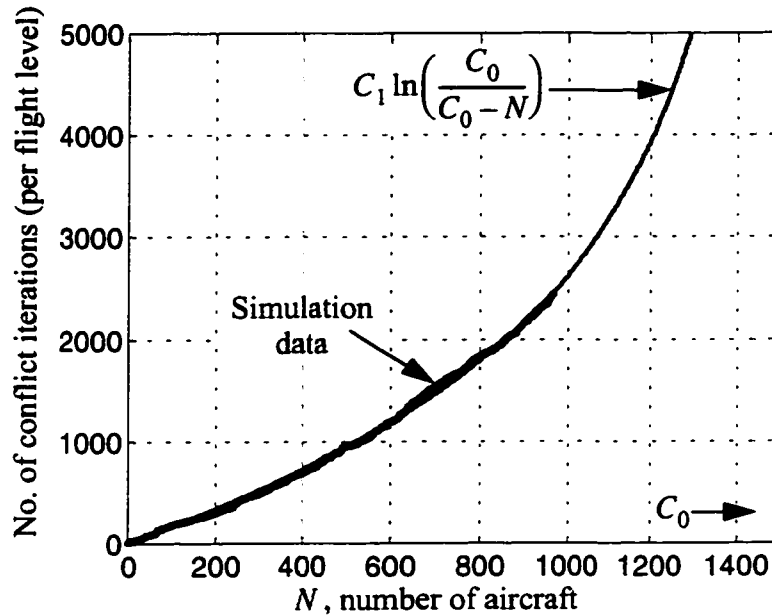


Figure 6.8. Curve-fitting empirical conflict-iteration data.

As discussed previously, each simulation consists of multiple passes through the active aircraft list as aircraft enter and leave the system. This produces multiple sets of simulation data for adjusting the conflict-iteration model parameters. A plot of all passes through the active aircraft list for the data set described in table 6.2 for the specific case of $\Delta x_{\text{grid}} = 15$ n.mi. shows how the capacity parameter, C_0 , is determined (fig. 6.9).

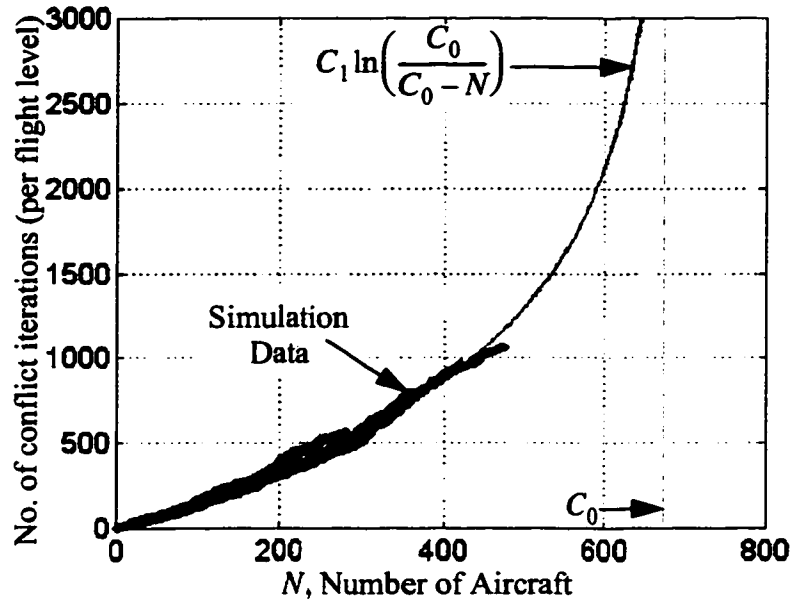


Figure 6.9. Curve fit example of conflict-iteration data.

The capacity limits determined for each value of Δx_{grid} have been determined through simulation and plotted together to determine the overall effects on airspace capacity of varying aircraft density (fig. 6.10). The maximum number of aircraft found at any flight level in the current air-traffic control system, including aircraft predicted to enter the airspace within the next 30 min (to account for T_{buffer}), is about 650 and is shown in figure 6.9. This corresponds to an average aircraft spacing for the optimized system of about 15 n.mi. This means that the current traffic levels could be accommodated by the optimization system, even if the inter-aircraft separation standard were increased. Conversely, if the inter-aircraft separation standard remains at 5 n.mi. as in the current air-traffic control system, an increase by a factor of 8 to nearly 4800 aircraft per flight level is predicted.

As derived in chapter 3, the full 3-D Class A capacity results are related to the 2-D results by simply scaling the 2-D results by the number of flight levels (eq. (3.35)). The scaled results over the 17 flight levels from FL180 through FL390 are shown in figure

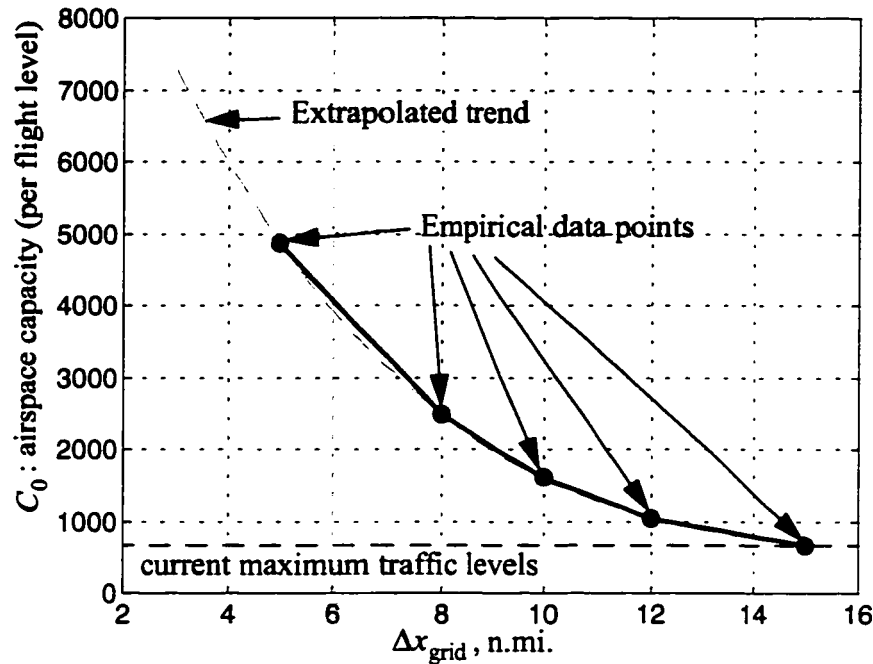


Figure 6.10. Airspace capacity (per flight level) vs. average conflict-grid-cell size.

6.11. Note that these scaled results assume that traffic would be equally distributed across all flight levels. Unless the airspace was truly at capacity, the distribution of traffic would be weighted toward the higher altitude flight levels near FL330 and FL350 (fig. 2.2).

6.3.3 Real-Time Computational Analysis

The computational performance of the optimization system and algorithms developed in this dissertation is of great interest. A simulation study and related analysis have been conducted to determine the computational properties of the sequential optimization approach. An effort has been made to generalize the computational results so that they may be applicable to different computer hardware implementations with different processing speeds.

The reason that real-time recomputation of optimal trajectories is required is to make large-scale corrections for large disturbances, as discussed in chapter 3. If winds and storms were predictable on the time-scale of a typical cross-country flight (about 6 hr), if

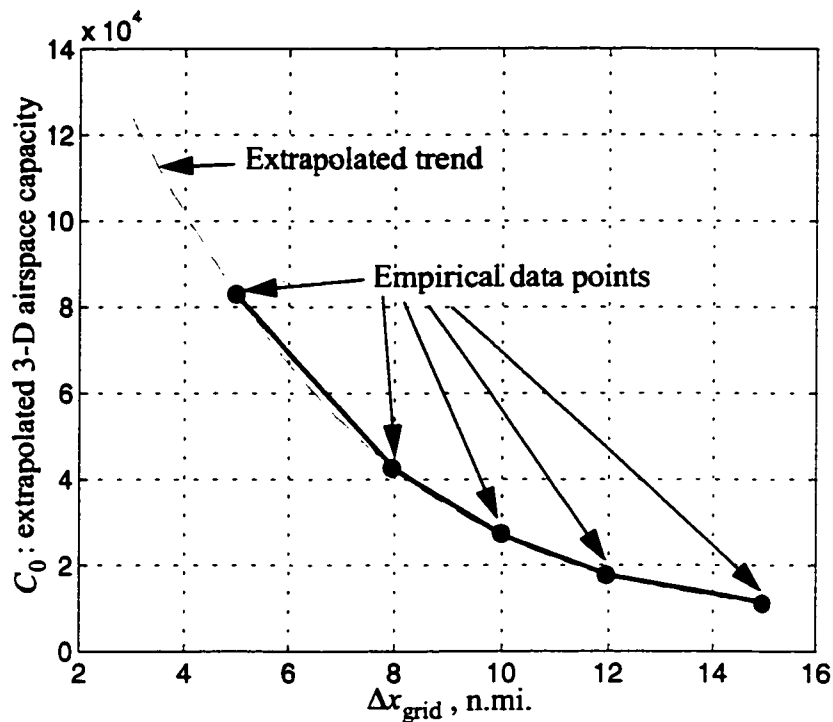


Figure 6.11. Extrapolated 3-D capacity for 17 flight levels from FL180 to FL390.

aircraft could depart exactly on schedule, and if aircraft preferences did not change, then one could compute an optimal set of trajectories, perhaps once per day, and then use closed-loop 4-D guidance and control to mitigate the effects of small disturbances. Although much progress has been made, winds and weather systems are notoriously difficult to accurately predict. The only thing that is certain is that unpredictable errors will occur, so recomputation of optimal trajectory solutions will always be required. Even without the possibility of mismodeled storms, simple errors in the predicted winds necessitate the occasional update of trajectory solutions so that aircraft are not forced to fly for extended periods of time at inefficient airspeeds. At efficient flight altitudes, commercial jet aircraft have a narrow airspeed flight envelope to work within; as a result, wind errors can quickly lead to situations in which aircraft come up against physical flight limitations, and 4-D trajectory recomputation would be required. Other sources of uncertainty that would require occasional trajectory updates are aircraft flight technical errors, navigation errors (probably not much of a problem with the advent of global

navigation systems like the Wide Area Augmentation System, or WAAS), and changes in aircraft flight preferences owing to internal airline scheduling changes.

These nominal uncertainties are the driving force behind required nominal recomputation rates. The largest nominal uncertainty, by orders of magnitude, is that of wind-modeling accuracy. The average performance of state-of-the-art continental-scale wind models is between 5 and 7 knots rms wind vector error [62, 63, 79]. Assuming a Gaussian distribution of wind-prediction errors, the variance in aircraft position error will grow linearly at a rate of between 5 and 7 nautical miles each hour. A reasonable goal is to maintain aircraft position errors within about 1 n.mi. To achieve this without requiring modifications in airspeed suggests that trajectory recomputation is needed on the order of every 10 to 12 min. Similarly, accurate storm predictions beyond about 30 min are not currently available. This suggests that trajectory recomputation under nominal conditions should be performed every 10 to 30 min. This will continue to be the case until the accuracy of weather prediction is improved.

A different source of uncertainty that is important enough to address separately is that of in-flight emergencies, including aircraft flight hardware problems, navigation or communication systems outages, medical emergencies, and hijackings. These are referred to as off-nominal system uncertainties. Any of these emergencies can occur at any time, so their effects and demands on a trajectory optimization system *must* be considered. Dealing with the detailed system engineering issues of severely off-nominal conditions like this is far beyond the scope of this dissertation, but the issue will be addressed at a high level. A brief analysis is now given to see how off-nominal system uncertainties may be used to derive a required recomputation rate for system-wide optimization.

With the conflict-grid method, trajectory solutions are designed to maintain inter-aircraft separation equal to the size of a conflict-grid cell. This leads to a means of deriving a reasonable recomputation limit. By assuming that the optimal conflict-free trajectories for all aircraft must be computed in the amount of time it takes a typical

aircraft to transit a conflict-grid cell, the amount of time required for recomputation, T_R , is simply given by

$$T_R = \frac{\Delta x_{\text{grid}}}{V_{\text{max}}} \quad (6.3)$$

where Δx_{grid} is the conflict-grid dimension, and V_{max} is the maximum expected aircraft ground speed (fig. 6.12). The maximum ground speed can be computed by adding the

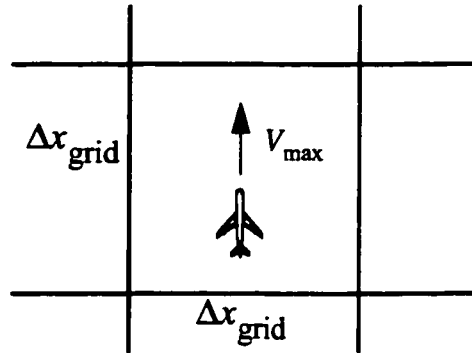


Figure 6.12. Estimating required recomputation rate from grid-cell transit time.

maximum typical airspeed for commercial jet aircraft to the maximum expected wind speed. Values in the range of 600 to 650 knots are typical.

The expected time to compute optimal conflict-free trajectories for a set of N aircraft may be determined based on equations (3.17), (3.23), and (3.27) by applying constants to relate the number of computations to time. The time to optimize all trajectories, T_O , is then given by

$$T_O = C_1 \ln\left(\frac{C_0}{C_0 - N}\right) K_{\text{NOWR}} \quad (6.4)$$

where K_{NOWR} is a hardware-dependent parameter to express the amount of time required to compute a neighboring optimal wind route.

In addition to the computation time, time must be provided for communicating trajectories to all of the aircraft. Currently, no dedicated high-bandwidth trajectory communications systems are in place; they would therefore have to be designed and fielded for practical implementation of the ideas in this dissertation. For analysis purposes, the communication time is left as a parameter so that requirements on the bandwidth of the communication system can be developed. The communication time is defined as T_C so that the total computation and communication time, T_T , is given by

$$T_T = C_1 \ln\left(\frac{C_0}{C_0 - N}\right) K_{\text{NOWR}} + T_C \quad (6.5)$$

In order to meet the emergency recomputation time, an inequality between equation (6.3) and equation (6.5) is formed to give

$$C_1 \ln\left(\frac{C_0}{C_0 - N}\right) K_{\text{NOWR}} + T_C \leq \frac{\Delta x_{\text{grid}}}{V_{\text{max}}} \quad (6.6)$$

Solving equation (6.6) for K_{NOWR} leads to the following requirement on the time per neighboring optimal wind route:

$$K_{\text{NOWR}} \leq \frac{\Delta x_{\text{grid}}}{V_{\text{max}}} (1 - \eta_c) \left[C_1 \ln\left(\frac{C_0}{C_0 - N}\right) \right]^{-1} \quad (6.7)$$

where η_c is the ratio of trajectory communication time to optimization time and is defined as

$$\eta_c \equiv T_C / T_R = \frac{T_C}{(\Delta x_{\text{grid}} / V_{\text{max}})} \quad (6.8)$$

An equivalent requirement on the number of neighboring optimal routes that must be computed per unit time may be derived by inverting equation (6.7) to give

$$\rho_{\text{NOWR}} \geq \left[\left(\frac{\Delta x_{\text{grid}}}{V_{\text{max}}} \right) (1 - \eta_c) \right]^{-1} \cdot C_1 \ln\left(\frac{C_0}{C_0 - N}\right) \quad (6.9)$$

where

$$\rho_{\text{NOWR}} \equiv \left(\frac{1}{K_{\text{NOWR}}} \right) \quad (6.10)$$

The derived expressions, equations (6.7) and (6.9), may be applied to either the full 3-D Class A problem or to the 2-D single-flight-level problem by appropriate scaling of the C_0 and C_1 parameters. As before, 2-D simulation results will be used to determine C_0 and C_1 directly for the 2-D case, and these will be scaled to determine the 3-D computation requirements.

An interesting aspect of this problem is that as Δx_{grid} increases, the amount of time available for computation increases because transit times across a grid cell are longer. But when Δx_{grid} increases, so does the expected number of conflicts, because more airspace is required per aircraft. This suggests that some optimum grid spacing exists from a computational standpoint.

Table 6.3. Empirically-determined conflict iteration parameters.

| Δx_{grid} | C_0 | C_1 |
|--------------------------|-------|-------|
| 15 | 675 | 962 |
| 12 | 1038 | 1552 |
| 10 | 1597 | 2462 |
| 8 | 2483 | 3566 |
| 5 | 4865 | 6599 |

The same simulation results for the previous simulation study (table 6.2) may be used to evaluate the computational requirements derived here. The values of C_0 and C_1 from those results are used here (table 6.3). The number of aircraft per flight level is set to 600, which is nearly double the current-day traffic levels. A value of 600 knots is assumed for V_{max} . The plot of ρ_{NOWR} versus Δx_{grid} for the 2-D case for three different values of η_c shows that there is, indeed, a minimum required computation rate at about a 12 n.mi. grid

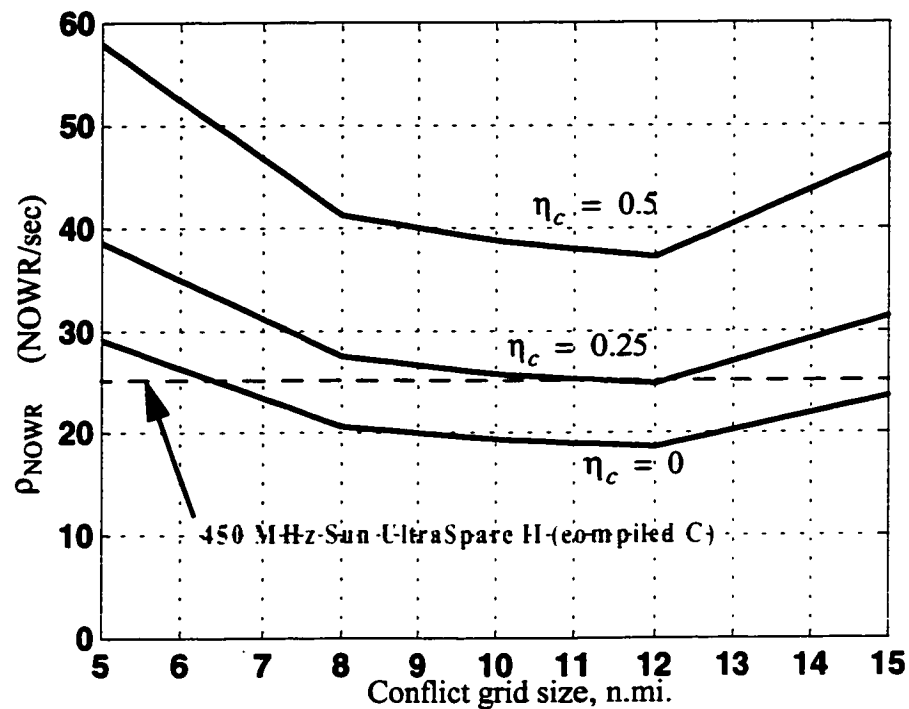


Figure 6.13. Required computation rate vs. conflict grid spacing.

spacing (fig. 6.13). This result is related to the typical speeds involved in commercial aircraft travel.

Porting the NOWR functions over to the C-language, compiling, and running a timing analysis on just that function allows the simulation timing results from the MATLAB system to be extrapolated to that which would be expected from a fully compiled version of the simulator. On the same plot (fig. 6.13), the computation rate achievable on an average Sun Ultra workstation (450 MHz, 256 Mb RAM) is shown for reference. The compiled version runs at a rate of about 25 optimal routes per second, or 0.04 msec per optimal route.

The equivalent plot has been generated for the full 3-D case to show what is required for solution of the complete Class A problem (fig. 6.14). For extrapolation of the results,

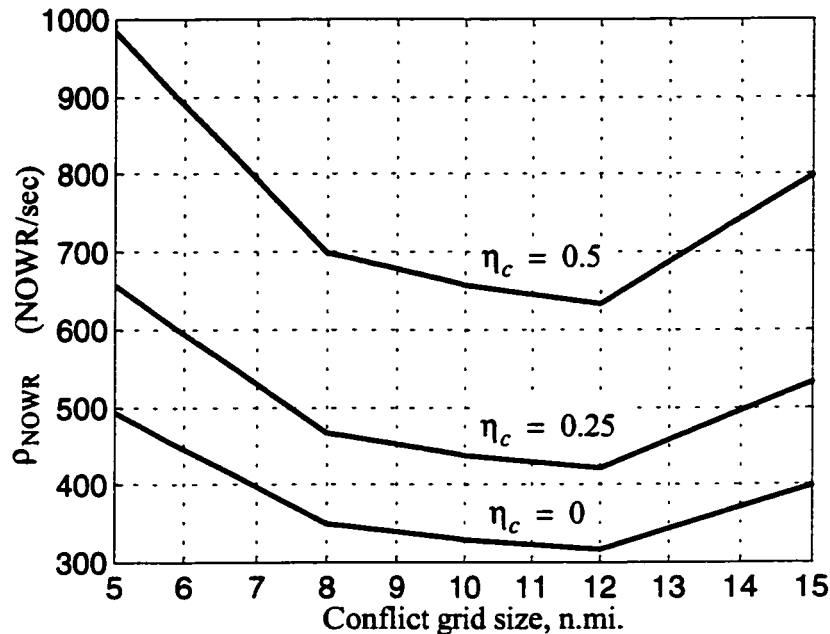


Figure 6.14. 2-D results extrapolated to the 3-D case.

the parameters N , C_0 , and C_1 are multiplied by the number of flight levels in Class A airspace, or 17. The absolute minimum optimal wind-route computation rate in the 3-D case is still at a conflict grid size of 12 n.mi., and is about 310 optimal routes per second. This is about a factor of 12 faster than has been measured with the compiled code on the 450-MHz UltraSparc work station. This is not a difficult gap to close, particularly since off-the-shelf computers operating at 3 GHz (3,000 MHz) are now available for under \$5,000. This means that the computational gap is already down to about a factor of 2. With attention paid to code optimization and with the regular improvements in computing power, this computational gap should already be considered closed. The next step in this work is to move from the 2-D development system to a full demonstration of the 3-D system on faster hardware, but the basic demonstration of the method has been achieved.

6.4 Summary

The simulation system that has been developed to test the algorithms and concepts in this dissertation has been presented. The results of several studies and analyses have also been presented to demonstrate how the simulation system can be used to obtain such results. The results obtained from the first study showed how airspace capacity might be expected to increase if the conflict-free optimization algorithms are adopted. The second study was a computational performance analysis that showed what is required of computer hardware to achieve real-time performance. Real-time performance for a single flight level has already been achieved on average off-the-shelf computer hardware, and, with high-end hardware, optimization of the complete Class A airspace is probably already possible too.

Chapter 7

Conclusion

The goal of this dissertation was to develop an approach and algorithms to enable the real-time conflict-free optimization of all aircraft in the en route airspace over the continental United States. Within the constraints and limitations of the computer and software development platform used for this work, that goal has been achieved.

A sequential optimization approach with iterative trajectory conflict resolution was adopted based on analysis showing that the en route airspace is, and will continue to be, sparsely populated with aircraft. The sequential optimization approach was shown to be a problem that could be solved in polynomial time. An analysis of the computational effort required by the sequential optimization approach showed that the core functions of optimal wind-routing and conflict detection were computational primitives and that they would need to be efficient to enable real-time performance. New approaches to both optimal wind-routing and conflict detection were developed to address this need for real-time performance.

A simulation system was developed to test the sequential optimization approach and associated algorithms. The simulation was developed for the 2-D problem, and analysis was used to extrapolate the 2-D results to the full 3-D Class A airspace problem. The simulation was used to perform several studies of the performance of the sequential optimization approach, including a computational analysis. The requirement for real-time recomputation of optimal conflict-free trajectories was related to the time required for an

aircraft to travel the minimum inter-aircraft separation distance. For the 2-D problem, this was shown to be equivalent to requiring optimal routes to be computed at a rate faster than 20 optimal routes per second. The equivalent 3-D requirement on optimal routes was shown to be a rate faster than about 300 optimal routes per second. A rate of 25 optimal routes per second was demonstrated on a Sun Ultra 450-MHz work station so that real-time performance has been demonstrated for the 2-D problem. It was reasoned that a rate of 300 optimal routes should be attainable with currently available computer hardware, though this has not yet been demonstrated.

7.1 Contributions

7.1.1 Neighboring Optimal Wind Routing

A new approach to computing minimum-time wind-routes based on the technique of neighboring optimal control (NOC) has been introduced. The NOC formulation produces a time-varying linear feedback law for computing near-optimal aircraft heading perturbations for given perturbations in the winds along a nominal route. Although this is mathematically similar to other linear feedback implementations, the ability to adjust the current aircraft heading as a function of perturbations in the winds at distant locations is conceptually unique.

The winds are modeled as piecewise linear functions of the along-track coordinate, specified at N_w grid points. The analytical solution for the NOC feedback gains has been determined as a function of N_w . The analytical solution for a general wind model enables accurate NOC gains to be computed even for a large number of grid points, and eliminates the need to tabulate feedback gains for look-up during implementation.

In addition to the NOC feedback law, normalization and transformation algorithms for applying the NOC solutions to problems in spherical-Earth coordinates have been developed. The combined NOC and transformation algorithms are called Neighboring Optimal Wind Routing (NOWR). The NOWR algorithm achieves near-minimum-time routes in an amount of time that is proportional to the length of the route (this is referred to as “order n ,” or $O(n)$). The NOWR algorithm has been demonstrated to produce optimal

wind-routes in about 10^5 floating point operations, which corresponds to about 40 milliseconds on a 450-MHz UNIX work station. This is an order of magnitude faster than existing optimization algorithms with similar optimization performance.

7.1.2 Strategic Conflict Detection and Resolution

Efficient approaches to strategic conflict-detection and resolution have been introduced. The problem of conflict detection is to predict whether the distance between any pair of aircraft will be less than the allowable legal separation (generally 5 n.mi. for commercial aircraft), and the best existing conflict-detection algorithms have been shown to be $O(n \log n)$. A new approach to conflict detection that is coupled with the conflict-resolution process has been developed. The Conflict Grid (CG) method involves storing aircraft trajectory data in a position versus time grid. A conflict is detected while storing trajectory data whenever any particular grid cell is seen to already be occupied by another aircraft or other airspace constraint (e.g., bad weather). The CG technique was shown to require less than 10^2 floating point operations to test a trajectory for conflicts. This is less than 0.1% additional computational effort over that required to compute an optimal wind-route. The CG method achieves this efficient performance by eliminating the need for pair-wise inter-aircraft distance computations. An extension to the CG method called the Stochastic Conflict Grid (SCG) was proposed to account for uncertainty in aircraft trajectory prediction and in the prediction of bad weather. Instead of setting grid cell values to binary 1 (occupied by a constraint) or 0 (no active constraint), the values in the SCG concept may take on any value between 0 and 1 to represent the probability that at least one constraint is active in that grid cell.

An enhancement to the NOWR algorithm was made to enable the efficient computation of perturbed optimal wind-routes. An efficient conflict-detection and resolution strategy was developed based on the perturbation NOWR algorithm for which the expected number of computations is proportional to $\log(C_0/(C_0 - N))$, where N is the number of aircraft in the system, and C_0 is a constant parameter used to fit the computational model to empirical data. This is the first technique proposed to resolve

conflicts while maintaining wind optimality for each aircraft route. All previous approaches have either neglected winds entirely, or have assumed that some nominal ground path was the desired objective so that conflict resolutions were computed to minimize deviations from the nominal path.

The resulting conflict-free solutions are not claimed to be globally optimal, but have been compared to a bound on the globally optimal solution and shown to exhibit performance within 0.25% of the global optimum for the real air-traffic and wind data used in the study.

7.1.3 Airspace Capacity Modeling

A model of airspace capacity was derived based on the total number of iterations required to determine optimal conflict-free trajectories for a set of N aircraft. The derived expression for the number of conflict-resolution iterations, \hat{Y}_N , is given by

$$\hat{Y}_N = C_1 \ln \left(\frac{C_0}{C_0 - N} \right) \quad (7.1)$$

where C_0 and C_1 are model parameters that are adjusted to fit empirical data. It was reasoned that C_0 may be interpreted as a measure of predicted airspace capacity. The use of this metric was demonstrated with simulation data. This is likely the first practical method of approximating en route airspace capacity for systems with unconstrained routing.

Using the capacity model, simulation studies were conducted to determine the relationship between legal aircraft separation standards and airspace capacity. This relationship may be used to estimate the potential benefits of reducing the minimum separation standards for en route aircraft. Reductions in the separation standards may now become possible with the advent of high-accuracy navigation and surveillance systems like the Wide Area Augmentation System (WAAS) and Automatic Dependent

Surveillance Mode-B (ADS-B). The proposed capacity model allows improvements in navigational performance to be related directly to improvements in airspace capacity.

7.2 Recommendations for Future Work

7.2.1 Extend NOWR to 3-D

Extending the 2-D NOWR algorithm to 3-D would permit the same rapid computation of 3-D routes as has been demonstrated with 2-D routes without having to decouple the horizontal and vertical trajectory optimization functions. This would permit greater flexibility in the sequential optimization system, and should lead to even better optimization performance. The challenge is to make the extension to 3-D while retaining the efficient properties of the 2-D NOWR algorithm. One of the great advantages of the NOWR algorithm is that, through normalization, it may be applied to the computation of optimal routes between any two points.

The vertical-profile optimization problem involves balancing engine fuel efficiency and aircraft drag. Higher altitudes lead to reduced drag, but also reduced engine performance. Airspeed also must be adjusted to achieve an optimal balance between drag and engine performance. Extending NOWR to 3-D means that the dynamic model would have to include a fuel-flow-rate model and an airframe-drag model. The variation in winds with altitude would also have to be included. In the spirit of NOWR, it is possible that the fuel flow rate could be modeled as a function of altitude (or temperature) with several nominal parameters. The nominal parameters would be modeled as constants so that perturbations in the parameters would be introduced into the neighboring optimal solution in the same manner as wind perturbations were introduced. The nominal optimal en route trajectory would be a normalized cruise-climb segment along a great circle. Perturbations in winds along the nominal route would be fed back as before, but now with additional perturbations in airspeed, engine performance model parameters, and airframe-drag model parameters. The utility of this approach would depend on how well engine performance and airframe drag could be modeled with linearized perturbations.

7.2.2 Demonstrate Stochastic Conflict Grid

The stochastic version of the conflict-grid concept was introduced in this dissertation, but not demonstrated. To do so would require the capability to compute position error estimates for aircraft trajectories and weather systems. This is not conceptually difficult, but would require a considerable amount of software-engineering effort to implement. It is anticipated that the stochastic conflict-grid concept would achieve good optimization performance in the uncertain national airspace system environment.

7.2.3 Demonstrate Solution to Complete 3-D Class A Airspace Problem

As with the stochastic conflict grid, the demonstration of the complete 3-D Class A airspace problem is not conceptually difficult, but would require considerable software engineering beyond what has been done in this dissertation for the 2-D simulation system. A computer platform with greater than 3-GHz performance (currently available) should be able to demonstrate complete 3-D solutions for over 10,000 aircraft in less than 1 min. In addition to the faster computation speed, enough RAM or virtual memory would be required to implement the conflict grid concept in 3-D. A conflict grid with 1 byte for each 5 n.mi. square grid cell spanning across the continental United States from FL180 up to FL390, and extending for 7 hr would require approximately 2-Gbyte of memory. UNIX work stations are already available with up to 8-Gbyte of RAM.

7.2.4 Develop a Perturbation Conflict-Resolution Algorithm for Arbitrary Trajectories

A practical ATC system will have to accept requests for arbitrary trajectories in addition to requests for optimal wind-routes. This will require a perturbation conflict-resolution algorithm to be developed for arbitrary trajectories similar to the perturbation NOWR algorithm developed here. Conceptually, this should not be difficult, but will require some development effort.

7.3 Concluding Remarks

If it were possible, the ultimate air-traffic control system would continuously compute optimal controls for all aircraft and communicate those controls to aircraft in real-time. This would permit the use of the most current system information and would allow the system to adapt to real-time uncertainties and emergencies. The optimal solution would unfold as new information became available. The system engineering and procedural changes required to implement such a system would be considerable, but in principle, this would be the ultimate air-traffic control system.

The computation of near-minimum-time conflict-free trajectories for the tens of thousands of aircraft that operate daily in the United States has remained a distant objective because of the extremely high problem complexity. Because the solution has seemed so distant, system concepts have generally not considered the use of real-time system-wide optimization.

The goal of this dissertation was to demonstrate the feasibility of a real-time system-wide optimization concept. A solution approach has been developed and demonstrated in simulation to bring real-time system-wide optimization into the realm of possibility. The system engineering required to take the concepts of this dissertation into the real world would be an immense undertaking. But now that the basic computational algorithms supporting the system-wide optimization concept have been shown to be feasible, implementation of the system-wide optimization concept in some form seems inevitable.

Appendix 1

Solutions for the Neighboring Optimal Wind Routing Gains

A1.1 Analytical Solution for NOC Feedback Gains

By examining small perturbations in the solution for minimum-time paths through varying winds, an analytical neighboring optimal control solution has been derived. The analytical solution is derived as a function of N_w , the number of grid points used to model the winds as piecewise linear functions of x . This enables NOC feedback gains to be computed for many different modeling situations much more quickly and precisely than has been possible with numerical methods.

In the following sections, the model of the minimum-time aircraft problem in varying winds is presented, followed by the derivation of the analytical neighboring optimal control feedback gains for the case of piecewise linearly varying winds.

A1.1.1 Linearized System Model

For zero winds, the nominal optimal heading angle from $\{x = 1, y = 0\}$ to the origin is $\theta_{\text{nom}} = \pi$ (fig. A1.1). With the intent of linearizing the equations of motion

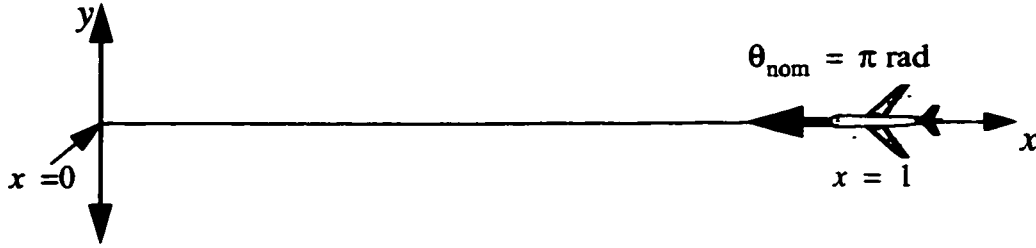


Figure A1.1. Nominal optimal trajectory in zero winds.

about this straight-line trajectory, the wind components, $u(x, y)$, and $v(x, y)$, are modeled as the sum of bias and cross-track shear terms in the following way:

$$u = u_b(x) + y \cdot u_y(x) \quad (\text{A1.1})$$

$$v = v_b(x) + y \cdot v_y(x) \quad (\text{A1.2})$$

When the linearization is performed, more general descriptions of the winds would reduce to these linear components, so the wind model given in equations (A1.1) and (A1.2) is as general as required for the derivation of NOC feedback gains. Further simplification can be achieved because, as is shown in a moment, the u_b and v_y terms are eliminated from the problem during the linearization process.

The equations of motion, including the differential equation derived by Zermelo for the minimum-time heading as a function of the wind perturbations [36, 37, 39], are given in this case by

$$\dot{x} = \cos\theta + u_b + y \cdot u_y \quad (\text{A1.3})$$

$$\dot{y} = \sin\theta + v_b + y \cdot v_y \quad (\text{A1.4})$$

$$\dot{\theta} = \frac{\partial v_b}{\partial x} \cdot \sin^2\theta + \left(\frac{\partial u_b}{\partial x} - v_y \right) \sin\theta \cos\theta - u_y \cdot \cos^2\theta \quad (\text{A1.5})$$

Perturbations in each of the wind terms may be considered independently in order to compute the corresponding perturbations in θ . For small perturbations in any of the wind terms, the corresponding perturbations in y and $\theta - \pi$ are also small. For such perturbations, $\sin \theta \rightarrow -(\theta - \pi)$, $\cos \theta \rightarrow -1$, and all remaining terms that are higher than first order in perturbation quantities reduce to zero. Making these substitutions leads to the following linearized equations of motion for minimum-time trajectories in winds:

$$x = 1 - t \quad (\text{A1.6})$$

$$\frac{dy}{dx} = (\theta - \pi) - v_b \quad (\text{A1.7})$$

$$\frac{d\theta}{dx} = u_y \quad (\text{A1.8})$$

where the differential equations in y and θ have been converted from t to x using equation (A1.6).

A1.1.2 Analytical NOC Feedback Gains for Piecewise Linear Winds

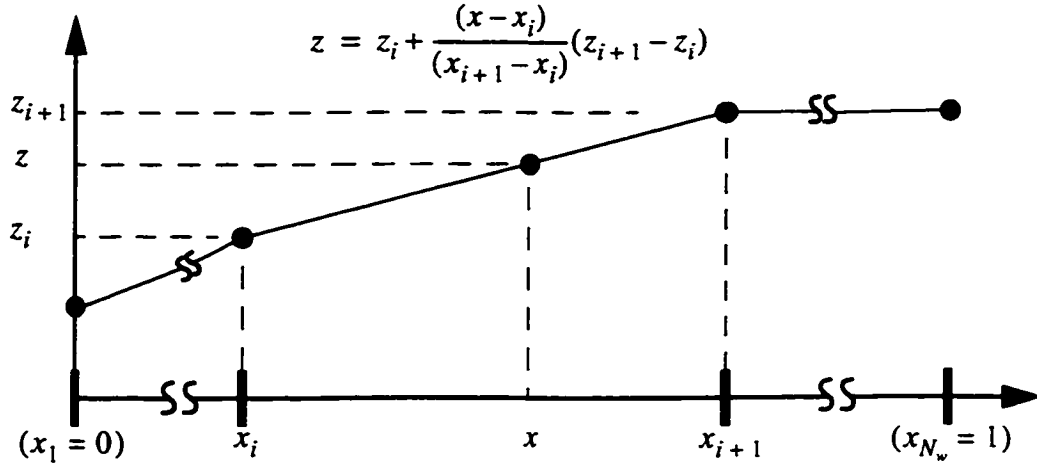
The terms u_y and v_b are now modeled as piecewise linear functions of x (fig. A1.2) and are specified by the parameters u_{yi} and v_{bi} at N_w grid points along the x -axis. For the linearized model, θ may be written as

$$\theta = f(x, y, u_{y1}, u_{y2}, \dots, u_{yN_w}, v_{b1}, v_{b2}, \dots, v_{bN_w}) \quad (\text{A1.9})$$

so that small perturbations in θ are given by

$$d\theta = \frac{\partial \theta}{\partial x} dx + \frac{\partial \theta}{\partial y} dy + \sum_{i=1}^{N_w} \left(\frac{\partial \theta}{\partial u_{yi}} du_{yi} + \frac{\partial \theta}{\partial v_{bi}} dv_{bi} \right) \quad (\text{A1.10})$$

Note that the partial derivative terms represent the change of θ with respect to the particular parameter or state while holding all other parameters and states at their nominal values. These are the neighboring optimal control feedback gains.

Figure A1.2. Graphical depiction of a piecewise linear function of x .

By equation (A1.8), $\partial\theta/\partial x = 0$ when all the wind parameters are zero. The perturbation in θ with respect to y is easily computed from the linearized equations of motion and is given by

$$\frac{\partial\theta}{\partial y} = \frac{1}{x} \quad (\text{A1.11})$$

Perturbing each of the wind parameters independently produces perturbations in both y and θ so that equation (A1.10) must be used to determine the contributions of just the wind parameters to perturbations in θ . The partial derivative of θ with respect to perturbations in parameter u_{yi} is given by

$$\frac{\partial\theta}{\partial u_{yi}} = \frac{d\theta}{du_{yi}} - \frac{\partial\theta}{\partial y} \cdot \frac{dy}{du_{yi}} \quad (\text{A1.12})$$

Similarly, perturbations in v_{bi} lead to

$$\frac{\partial\theta}{\partial v_{bi}} = \frac{d\theta}{dv_{bi}} - \frac{\partial\theta}{\partial y} \cdot \frac{dy}{dv_{bi}} \quad (\text{A1.13})$$

The partial derivative of θ with respect to y is given by equation (A1.11), but the other derivatives remain to be determined from the linearized equations and the piecewise linear models of the winds.

Integrating equation (A1.8) over θ and x , and taking the derivative with respect to u_{yi} produces the following expression for the total derivative of θ with respect to u_{yi} :

$$\frac{d\theta}{du_{yi}} = \frac{d\theta_0}{du_{yi}} + \int_{x_0}^x \frac{du_y}{du_{yi}} dx \quad (\text{A1.14})$$

Similarly, setting $v_b = 0$ and integrating equation (A1.7) over θ and y leads to

$$\frac{dy}{du_{yi}} = \frac{d\theta_0}{du_{yi}} \cdot (x - x_0) + \int_{x_0}^x \left[\int_{x_0}^x \frac{du_y}{du_{yi}} dx \right] dx \quad (\text{A1.15})$$

The corresponding expressions for the total derivatives of θ and y with respect to v_b are derived in a similar manner, but now with $u_y = 0$. The derivative terms are given by

$$\frac{d\theta}{dv_{bi}} = \frac{d\theta_0}{dv_{bi}} \quad (\text{A1.16})$$

$$\frac{dy}{dv_{bi}} = \frac{d\theta_0}{dv_{bi}} \cdot (x - x_0) - \int_{x_0}^x \frac{dv_b}{dv_{bi}} dx \quad (\text{A1.17})$$

The integrations in equations (A1.14), (A1.15), and (A1.17) are now computed. Because of the piecewise linear definitions of the winds, these integrals must be computed in a piecewise fashion. In general, for a perturbation in any parameter u_{yi} or v_{bi} , there are up to four integration regions (fig. A1.3). Fewer integration regions are required both at the first two and the last two grid points. Integrating the piecewise linear functions of x is straightforward but tedious because one must be careful to carry through boundary conditions to match the integral over each integration region. These solutions are now presented.

Since the initial condition is at $x = x_{N_w}$, the integrals are carried out from $x = x_{N_w}$ to $x = x_1$. As depicted in fig. A1.3, the integration regions are defined as follows:

$$\text{Region } R_I \quad x_{i+1} \leq x \leq x_{N_w} \quad (\text{A1.18})$$

$$\text{Region } R_{II} \quad x_i \leq x \leq x_{i+1} \quad (\text{A1.19})$$

$$\text{Region } R_{III} \quad x_{i-1} \leq x \leq x_i \quad (\text{A1.20})$$

$$\text{Region } R_{IV} \quad x_1 \leq x \leq x_{i-1} \quad (\text{A1.21})$$

Note that some of these regions do not exist for certain values of i . For instance, Region R_I does not exist for either $i = N_w - 1$ or $i = N_w$. The integrations in each of

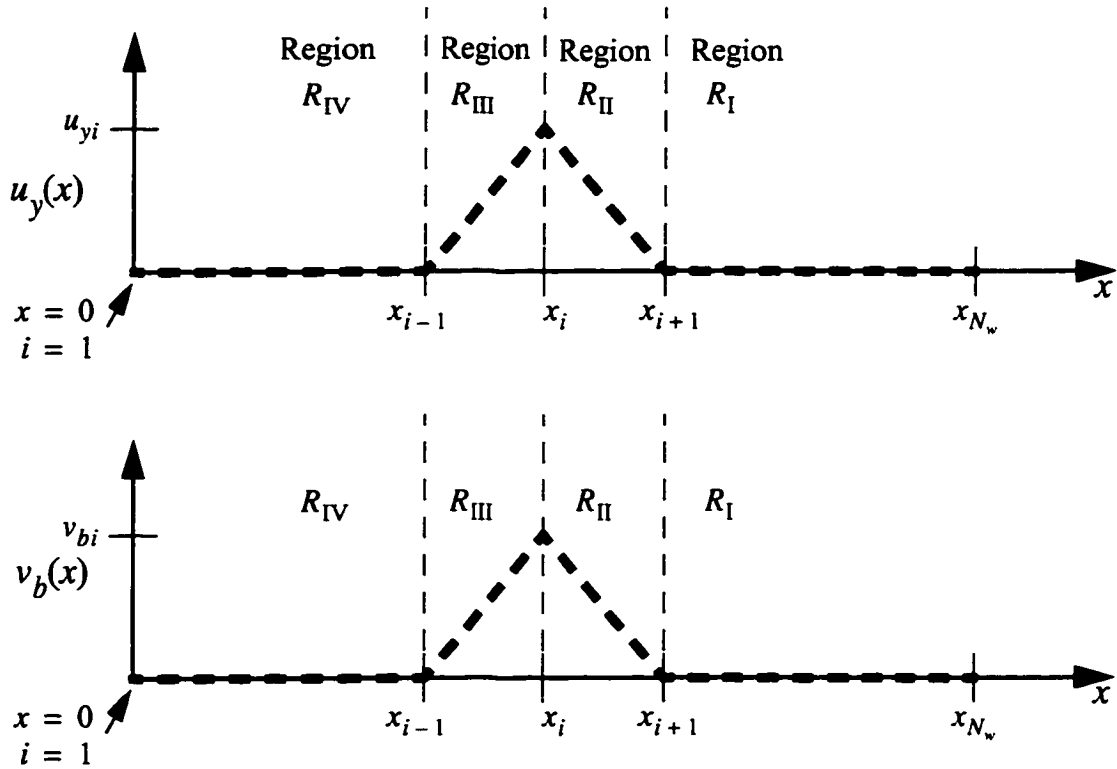


Figure A1.3. Piecewise linear perturbations in the wind parameters at point i .

these regions for the piecewise linear functions shown in fig. A1.3 are now presented, and the applicable boundary conditions are identified.

Region R_I

The integrals over du_y/du_{yi} in R_I are given by

$$\int_{x_{N_w}}^x \frac{du_y}{du_{yi}} dx = \int_{x_{N_w}}^x 0 dx = 0 \quad (\text{A1.22})$$

$$\int_{x_{N_w}}^x \left[\int_{x_{N_w}}^x \frac{du_y}{du_{yi}} dx \right] dx = 0 \quad (\text{A1.23})$$

so that equation (A1.14) becomes

$$\left. \frac{d\theta}{du_{yi}} \right|^{R_I} = \frac{d\theta_{N_w}}{du_{yi}} \quad (\text{A1.24})$$

and equation (A1.15) becomes

$$\left. \frac{dy}{du_{yi}} \right|^{R_I} = \frac{d\theta_{N_w}}{du_{yi}} \cdot (x - x_{N_w}) \quad (\text{A1.25})$$

Substituting these expressions along with equation (A1.11) into equation (A1.12) and simplifying leads to the following expression for the NOC feedback gain for perturbations in u_{yi} :

$$\left. \frac{\partial \theta}{\partial u_{yi}} \right|^{R_I} = \frac{d\theta_{N_w}}{du_{yi}} \cdot \left(\frac{x_{N_w}}{x} \right) \quad (\text{A1.26})$$

with the value of $d\theta_{N_w}/du_{yi}$ remaining to be determined by the following boundary condition:

$$\left. \frac{\partial \theta}{\partial u_{yi}} \right|_{x=x_{i+1}}^{R_I} = \left. \frac{\partial \theta}{\partial u_{yi}} \right|_{x=x_{i+1}}^{R_{II}} \quad (A1.27)$$

By a similar process, the NOC feedback gain for perturbations in v_{bi} is given by

$$\left. \frac{\partial \theta}{\partial v_{bi}} \right|^{R_I} = \frac{d\theta_{N_w}}{dv_{bi}} \cdot \left(\frac{x_{N_w}}{x} \right) \quad (A1.28)$$

with boundary condition

$$\left. \frac{\partial \theta}{\partial v_{bi}} \right|_{x=x_{i+1}}^{R_I} = \left. \frac{\partial \theta}{\partial v_{bi}} \right|_{x=x_{i+1}}^{R_{II}} \quad (A1.29)$$

Region R_{II}

The integrals over du_y/du_{yi} in R_{II} are given by

$$\int_{x_{i+1}}^x \frac{du_y}{du_{yi}} dx = \int_{x_{i+1}}^x \frac{(x-x_{i+1})}{(x_i-x_{i+1})} dx = \frac{(x-x_{i+1})^2}{2(x_i-x_{i+1})} \quad (A1.30)$$

$$\int_{x_{i+1}}^x \left[\int_{x_{i+1}}^x \frac{du_y}{du_{yi}} dx \right] dx = \frac{(x_{i+1}-x)^3}{6(x_{i+1}-x_i)} \quad (A1.31)$$

so that equation (A1.14) becomes

$$\left. \frac{d\theta}{du_{yi}} \right|^{R_{II}} = \frac{d\theta_{i+1}}{du_{yi}} + \frac{(x-x_{i+1})^2}{2(x_i-x_{i+1})} \quad (A1.32)$$

and equation (A1.15) becomes

$$\left. \frac{dy}{du_{yi}} \right|^{R_{II}} = \frac{d\theta_{i+1}}{du_{yi}} \cdot (x - x_{i+1}) + \frac{(x_{i+1} - x)^3}{6(x_{i+1} - x_i)} \quad (\text{A1.33})$$

Substituting these expressions along with equation (A1.11) into equation (A1.12) and simplifying leads to the following expression for the NOC feedback gain for perturbations in u_{yi} :

$$\left. \frac{\partial \theta}{\partial u_{yi}} \right|^{R_{II}} = \frac{d\theta_{i+1}}{du_{yi}} \cdot \left(\frac{x_{i+1}}{x} \right) - \frac{(x_{i+1} - x)^2}{6x \cdot (x_{i+1} - x_i)} (2x + x_{i+1}) \quad (\text{A1.34})$$

with the value of $d\theta_{i+1}/du_{yi}$ remaining to be determined by the following boundary conditions:

$$\begin{aligned} \left. \frac{\partial \theta}{\partial u_{yi}} \right|_{x=x_i}^{R_{II}} &= \left. \frac{\partial \theta}{\partial u_{yi}} \right|_{x=x_i}^{R_{III}} & 2 \leq i \\ \left. \frac{\partial \theta}{\partial u_{yi}} \right|_{x=x_1}^{R_{II}} &= 0 & i = 1 \end{aligned} \quad (\text{A1.35})$$

The separate boundary condition is required because there is no R_{III} for $i = 1$.

By a similar process, the NOC feedback gain for perturbations in v_{bi} is given by

$$\left. \frac{\partial \theta}{\partial v_{bi}} \right|^{R_{II}} = \frac{d\theta_{i+1}}{dv_{bi}} \cdot \left(\frac{x_{i+1}}{x} \right) - \frac{(x_{i+1} - x)^2}{2x \cdot (x_{i+1} - x_i)} \quad (\text{A1.36})$$

with boundary conditions

$$\begin{aligned} \frac{\partial \theta}{\partial v_{bi}} \Big|_{x=x_i}^{R_{II}} &= \frac{\partial \theta}{\partial v_{bi}} \Big|_{x=x_i}^{R_{III}} & 2 \leq i \\ \frac{\partial \theta}{\partial v_{b1}} \Big|_{x=x_1}^{R_{II}} &= 1 & i = 1 \end{aligned} \quad (A1.37)$$

Note that in this case, for $i = 1$, the boundary conditions are determined by applying equation (A1.36) as a limiting case as $x \rightarrow 0$. By doing so, it is determined that $d\theta_2/dv_{b1} = 1/2$.

Region R_{III}

The integrals over du_y/du_{yi} in R_{III} are given by

$$\int_{x_i}^x \frac{du_y}{du_{yi}} dx = \int_{x_i}^x \left(1 - \frac{(x_i - x)}{(x_i - x_{i-1})} \right) dx = -(x_i - x) + \frac{(x_i - x)^2}{2(x_i - x_{i-1})} \quad (A1.38)$$

$$\int_{x_i}^x \left[\int_{x_i}^x \frac{du_y}{du_{yi}} dx \right] dx = \frac{(x_i - x)^2(2(x_i - x_{i-1}) + (x - x_{i-1}))}{6(x_i - x_{i-1})} \quad (A1.39)$$

so that equation (A1.14) becomes

$$\frac{d\theta}{du_{yi}} \Big|^{R_{III}} = \frac{d\theta_i}{du_{yi}} - (x_i - x) + \frac{(x_i - x)^2}{2(x_i - x_{i-1})} \quad (A1.40)$$

and equation (A1.15) becomes

$$\frac{dy}{du_{yi}} \Big|^{R_{III}} = \frac{d\theta_i}{du_{yi}} \cdot (x - x_i) + \frac{(x_i - x)^2(2(x_i - x_{i-1}) + (x - x_{i-1}))}{6(x_i - x_{i-1})} \quad (A1.41)$$

Substituting these expressions along with equation (A1.11) into equation (A1.12) and simplifying leads to the following expression for the NOC feedback gain for perturbations in u_{yi} :

$$\left. \frac{\partial \theta}{\partial u_{yi}} \right|^{R_{III}} = \frac{d\theta_i}{du_{yi}} \cdot \left(\frac{x_i}{x} \right) - \frac{(x_i^2 - x^2)}{2x} + \frac{(x_i - x)^2(2x + x_i)}{6x \cdot (x_i - x_{i-1})} \quad (\text{A1.42})$$

with the value of $d\theta_i/du_{yi}$ remaining to be determined by the following boundary condition:

$$\begin{aligned} \left. \frac{\partial \theta}{\partial u_{yi}} \right|_{x=x_{i-1}}^{R_{III}} &= 0 & 2 \leq i \\ \left. \frac{\partial \theta}{\partial u_{yi}} \right|_{x=x_1}^{R_{III}} &= \text{N/A} & i = 1 \end{aligned} \quad (\text{A1.43})$$

The gain in R_{III} is left undefined when $i = 1$ because there is no R_{III} in that case.

By a similar process, the NOC feedback gain for perturbations in v_{bi} is given by

$$\left. \frac{\partial \theta}{\partial v_{bi}} \right|^{R_{III}} = \frac{d\theta_i}{dv_{bi}} \cdot \left(\frac{x_i}{x} \right) + \frac{(x - x_{i-1})^2}{2x \cdot (x_i - x_{i-1})} \quad (\text{A1.44})$$

with boundary condition

$$\begin{aligned} \left. \frac{\partial \theta}{\partial v_{bi}} \right|_{x=x_{i-1}}^{R_{III}} &= 0 & 2 \leq i \\ \left. \frac{\partial \theta}{\partial v_{bi}} \right|_{x=x_1}^{R_{III}} &= \text{N/A} & i = 1 \end{aligned} \quad (\text{A1.45})$$

Region R_{IV}

In Region R_{IV} , it is straightforward to show that

$$\begin{aligned} \left. \frac{\partial \theta}{\partial u_{yi}} \right|_{R_{IV}} &= 0 & 3 \leq i \\ \left. \frac{\partial \theta}{\partial u_{yi}} \right|_{R_{IV}} &= \text{N/A} & i = 1, 2 \end{aligned} \quad (\text{A1.46})$$

$$\begin{aligned} \left. \frac{\partial \theta}{\partial v_{bi}} \right|_{R_{IV}} &= 0 & 3 \leq i \\ \left. \frac{\partial \theta}{\partial v_{bi}} \right|_{R_{IV}} &= \text{N/A} & i = 1, 2 \end{aligned} \quad (\text{A1.47})$$

A1.1.3 Analytical NOC Solution Summary

Applying the boundary conditions determines the various constants in the gain expressions. A summary of the analytical NOC feedback gains is now presented.

Perturbations in y

The neighboring optimal perturbation in heading for perturbations in cross-track position is

$$\frac{\partial \theta}{\partial y} = \frac{1}{x} \quad 0 \leq x \leq 1 \quad (\text{A1.48})$$

Note that this linearized feedback gain approaches infinity as the aircraft nears $x = 0$. In practice, a mode switch to a heading regulator is used to avoid this singularity. Another option is to use the optimal nonlinear feedback rule for perturbations in y , which is

$$\Delta \theta_y = \tan^{-1} \left(\frac{y}{x} \right) \quad (\text{A1.49})$$

Note that equation (A1.49) reduces to equation (A1.48) for small perturbations in y .

which is precisely the linearized NOC feedback law for perturbations in y .

Perturbations in u_{yi}

The NOC gain for perturbations in the along-track shear at $i = 1$ is given by

$$\frac{\partial \theta}{\partial u_{y1}} = \begin{cases} \frac{1}{6(N_w - 1)^2} \cdot \frac{1}{x} & \frac{1}{(N_w - 1)} < x \leq 1 \\ \frac{1}{6(N_w - 1)^2} \cdot \frac{1}{x} \cdot [1 - (1 - x \cdot (N_w - 1))^2(2x \cdot (N_w - 1) + 1)] & 0 < x \leq \frac{1}{(N_w - 1)} \\ 0 & \lim x \rightarrow 0 \end{cases} \quad (\text{A1.50})$$

where it is noted that

$$x_i = \frac{i - 1}{N_w - 1} \quad (\text{A1.51})$$

All other NOC gains for perturbations in the along-track shear over the range $2 \leq i \leq N_w$ are given by

$$\frac{\partial \theta}{\partial u_{yi}} = \begin{cases} \frac{(i - 1)}{(N_w - 1)^2} \cdot \frac{1}{x} & \{x_{i+1} < x \leq 1\} \\ \frac{6(i - 1) - (i - x \cdot (N_w - 1))^2 \cdot (2x \cdot (N_w - 1) + i)}{6x \cdot (N_w - 1)^2} & \{x_i < x \leq x_{i+1}\} \\ \frac{[3i - 4 - 3((i - 1)^2 - x^2(N_w - 1)^2) + (i - 1 - x \cdot (N_w - 1))^2(2x \cdot (N_w - 1) + i - 1)]}{6x \cdot (N_w - 1)^2} & \{x_{i-1} < x \leq x_i\} \\ 0 & \{0 \leq x \leq x_{i-1}\} \end{cases} \quad (\text{A1.52})$$

Perturbations in v_{bi}

The NOC gain for perturbations in the cross-track bias wind at $i = 1$ is given by

$$\frac{\partial \theta}{\partial v_{b1}} = \begin{cases} \frac{1}{2(N_w - 1)} \cdot \frac{1}{x} & \frac{1}{(N_w - 1)} < x \leq 1 \\ 1 - \frac{(N_w - 1)}{2} \cdot x & 0 \leq x \leq \frac{1}{(N_w - 1)} \end{cases} \quad (\text{A1.53})$$

All other NOC gains for perturbations in the cross-track bias wind over the range $2 \leq i \leq N_w$ are given by

$$\frac{\partial \theta}{\partial v_{bi}} = \begin{cases} \frac{1}{(N_w - 1)} \cdot \frac{1}{x} & x_{i+1} < x \leq 1 \\ \frac{2 - (i - x \cdot (N_w - 1))^2}{2x \cdot (N_w - 1)} & x_i < x \leq x_{i+1} \\ \frac{(i - 2 - x \cdot (N_w - 1))^2}{2x \cdot (N_w - 1)} & x_{i-1} < x \leq x_i \\ 0 & 0 \leq x \leq x_{i-1} \end{cases} \quad (\text{A1.54})$$

The NOC gains for perturbations in y , u_{yi} , and v_{bi} have been plotted for the specific case of $N_w = 13$ to illustrate their qualitative nature (fig. A1.4).

Finally, the neighboring optimal feedback law is determined from equation (A1.10) and the nominal solution ($\theta = \pi$) to be

$$\theta = \pi + \frac{\partial \theta}{\partial y} \cdot y + \sum_{i=1}^{N_w} \left(\frac{\partial \theta}{\partial u_{yi}} \cdot u_{yi} + \frac{\partial \theta}{\partial v_{bi}} \cdot v_{bi} \right) \quad (\text{A1.55})$$

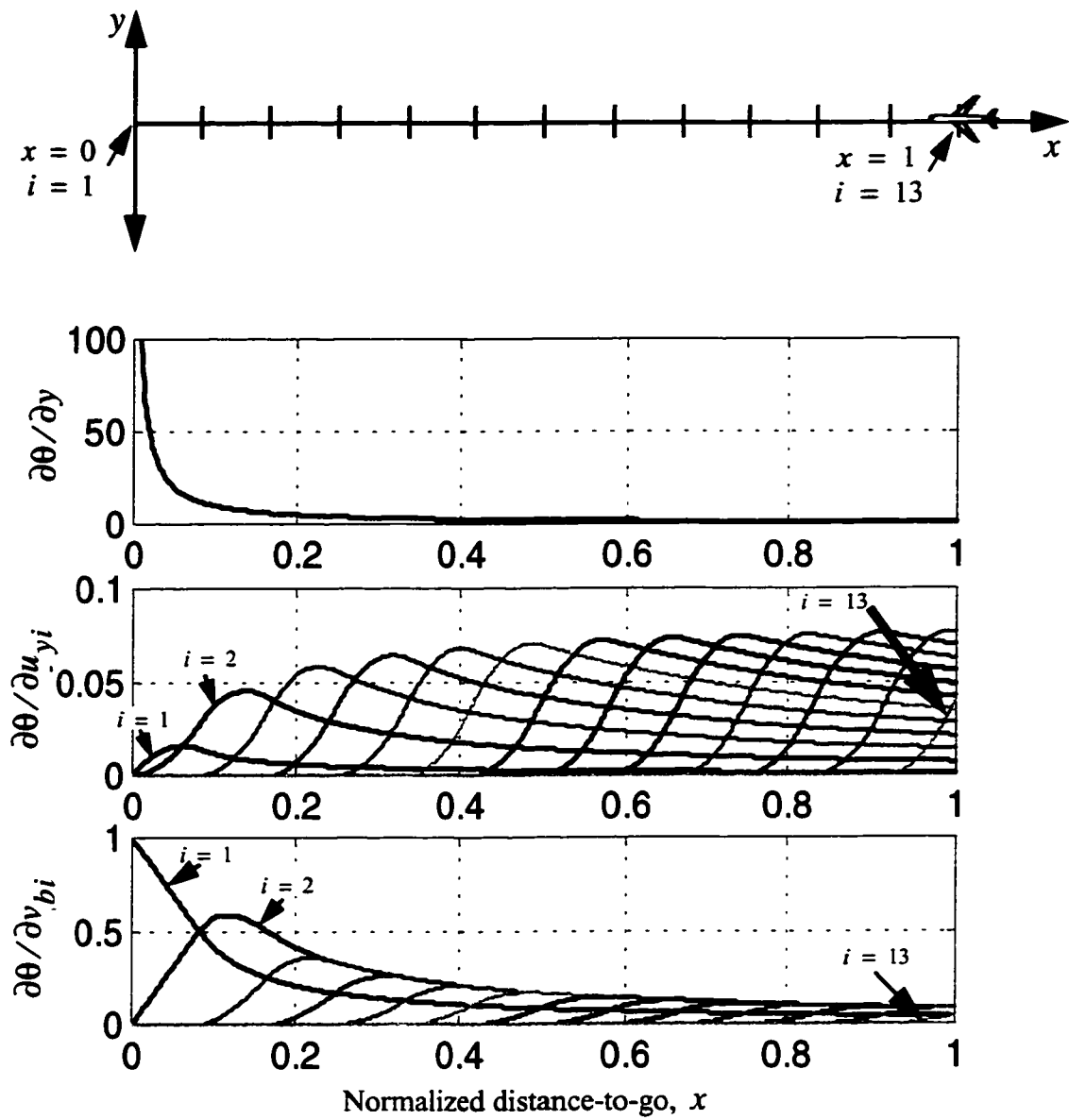


Figure A1.4. Normalized neighboring optimal feedback gains.

A1.2 Numerical Solution for NOC Feedback Gains

The numerical solution for the neighboring optimal control gains via the backward sweep method [39] is presented in this section. Although the analytical solution for the gains has been derived in this case, the numerical solution is useful because it demonstrates how NOC gains may be computed for other systems with varying system parameters when the analytical solution is not feasible. The technique is to model the varying parameters as bias states of the system. By doing so, the NOC solution via the backward sweep method produces NOC feedback gains for perturbations in the system parameters.

A1.2.1 Dynamic Model

The augmented state vector, s , is given by

$$s \equiv \begin{bmatrix} x & y & u_{y1} & \dots & u_{yN_w} & v_{b1} & \dots & v_{bN_w} \end{bmatrix}^T \in R^{n \times 1} \quad (\text{A1.56})$$

The state equations of motion are presented in vector form as

$$\dot{s} = f[s(t), \theta(t)] = \begin{bmatrix} \cos\theta + a(x)^T(y \cdot U_y) \\ \sin\theta + a(x)^T V_b \\ 0^{N_w \times 1} \\ 0^{N_w \times 1} \end{bmatrix} \quad (\text{A1.57})$$

where $0^{N_w \times 1}$ is an $N_w \times 1$ vector of zeros, and the wind parameter vectors are defined as

$$U_y \equiv \begin{bmatrix} u_{y1} & u_{y2} & \dots & u_{yN_w} \end{bmatrix}^T \quad U_y \in R^{N_w \times 1} \quad (\text{A1.58})$$

$$V_b \equiv \begin{bmatrix} V_{b1} & V_{b2} & \dots & V_{bN_w} \end{bmatrix}^T \quad V_b \in R^{N_w \times 1} \quad (\text{A1.59})$$

The elements of the piecewise linear vector function, $a(x)$, are defined by the following equations:

$$a_i(x) = i - x(N_w - 1) \quad (\text{A1.60})$$

$$a_{i+1}(x) = x(N_w - 1) - (i - 1) \quad (\text{A1.61})$$

$$a_j(x) = 0 \begin{cases} j = 1, 2, \dots, (i - 1) \\ j = (i + 2), (i + 3), \dots, N_w \end{cases} \quad (\text{A1.62})$$

The integer value of i is determined based on the position, x , from the following rule:

$$i = \min[(1 + \lfloor x \cdot (N_w - 1) \rfloor), (N_w - 1)] \quad (\text{A1.63})$$

where $\min[c, d]$ is the minimum of c and d , and $\lfloor c \rfloor$ is the floor of c , defined as the largest integer smaller than c .

Note that none of the wind states are controllable. The model is augmented with these states so that the effects of perturbations in the winds on the optimal aircraft heading can be accounted for. The models for the wind parameters need not be constants. Other dynamic models may be used. Instead of modeling the wind parameter as constants, one might choose to model them by linear dynamic processes. For example, the following dynamic model of U_y might be used:

$$\begin{aligned} \dot{U}_y &= -A_s \cdot U_y + B_s \cdot U_{yc} \\ \dot{U}_{yc} &= 0 \end{aligned} \quad (\text{A1.64})$$

where A_s and B_s are the state transition and control matrices, and U_{yc} is a constant command parameter vector. An exponential decay process would be modeled by choosing the following form for A_s and B_s :

$$A_s = B_s = \begin{bmatrix} a_{s1} & 0 & \dots & 0 & 0 \\ 0 & a_{s2} & 0 & \dots & 0 \\ \dots & 0 & \dots & 0 & 0 \\ 0 & \dots & 0 & a_{s(N_w-1)} & 0 \\ 0 & 0 & 0 & 0 & a_{sN_w} \end{bmatrix} \quad (\text{A1.65})$$

This approach would allow predicted time variations in the winds to be modeled explicitly.

A1.2.2 The Optimal Control Problem

In this section, the optimal wind-routing problem is formally posed as an optimization problem in the calculus of variations. The cost function, dynamic constraints, and initial and final state constraints are defined. This is done so that the nominal optimal solution may be explicitly identified before the neighboring optimal control solution is derived. The notation and definitions here owe much to Refs. [39] and [64].

The goal is to choose the heading angle, $\theta(t)$, to minimize

$$J = \phi[s(t_f), t_f] = t_f \quad (\text{A1.66})$$

where t_f is the free final time.

The dynamic constraints are given by equation (A1.57) with initial conditions

$$s(t_0) = s_0 = \begin{bmatrix} 1 & 0 & 0^{1 \times N_w} & 0^{1 \times N_w} \end{bmatrix}^T \quad (\text{A1.67})$$

The final state constraint is that the aircraft must arrive at the origin, so the final state constraints are given by

$$\psi(s(t_f)) = \begin{bmatrix} x \\ y \end{bmatrix}_{t=t_f} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (\text{A1.68})$$

The Hamiltonian for this problem is

$$H = \lambda(t)^T f[s, \theta] \quad (\text{A1.69})$$

where $\lambda(t)$ is an n -row column vector of to-be-determined Lagrange multipliers defined as

$$\lambda(t) \equiv \begin{bmatrix} \lambda_x & \lambda_y & \lambda_{uy1} & \dots & \lambda_{uyN_w} & \lambda_{vb1} & \dots & \lambda_{vbN_w} \end{bmatrix}^T \quad (\text{A1.70})$$

All of the Lagrange multipliers associated with the constant wind parameters vanish so that the Hamiltonian simplifies to

$$H = \lambda_x(\cos\theta + a(x)^T(y \cdot U_y)) + \lambda_y(\sin\theta + a(x)^T V_b) \quad (\text{A1.71})$$

The function Φ is defined here as

$$\Phi \equiv t_f + v^T \psi \quad (\text{A1.72})$$

where v is a vector of constant Lagrange multipliers that adjoin the final state constraints to the cost function, and is defined by

$$v \equiv \begin{bmatrix} v_x \\ v_y \end{bmatrix} \quad (\text{A1.73})$$

The Euler-Lagrange equations for this problem are

$$\dot{\lambda} = -H_s = -f_s \cdot \lambda \quad (\text{A1.74})$$

where H_s is the partial derivative of H with respect to the state vector. The partial derivative of the dynamic constraint vector is given by

$$f_s \equiv \frac{\partial f}{\partial s} \quad (\text{A1.75})$$

which expands to

$$f_s = \begin{bmatrix} y \cdot a^T(U_y)_s + (U_y)_s^T(a \cdot y_s^T + y \cdot a_s) \\ a^T(V_b)_s + V_b^T a_s \\ 0^{N_w \times n} \\ 0^{N_w \times n} \end{bmatrix} \in R^{n \times n} \quad (\text{A1.76})$$

The derivative of a vector, $B(x) \in R^{m \times 1}$, with respect to $s \in R^{n \times 1}$, is defined as

$$B_s \equiv \begin{bmatrix} \frac{\partial B_1}{\partial s_1} & \frac{\partial B_1}{\partial s_2} & \cdots & \frac{\partial B_1}{\partial s_n} \\ \frac{\partial B_2}{\partial s_1} & \frac{\partial B_2}{\partial s_2} & \cdots & \frac{\partial B_2}{\partial s_n} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial B_m}{\partial s_1} & \frac{\partial B_m}{\partial s_2} & \cdots & \frac{\partial B_m}{\partial s_n} \end{bmatrix} \quad (\text{A1.77})$$

Following this definition, the derivative of U_y with respect to the state vector is

$$(U_y)_s = \begin{bmatrix} [0^{N_w \times 2}] & [I^{N_w \times N_w}] & [0^{N_w \times N_w}] \end{bmatrix} \quad (\text{A1.78})$$

where $I^{m \times m} \in R^{m \times m}$ is a square identity matrix.

Similarly, $(V_b)_s$ is given by

$$(V_b)_s = \begin{bmatrix} [0^{N_w \times 2}] & [0^{N_w \times N_w}] & [I^{N_w \times N_w}] \end{bmatrix} \quad (\text{A1.79})$$

The derivative of the piecewise varying function $a(x)$ with respect to s is a piecewise varying matrix given by

$$a_s(i, 1) = -(N_w - 1) \quad (\text{A1.80})$$

$$a_s((i + 1), 1) = (N_w - 1) \quad (\text{A1.81})$$

$$a_s(j, k) = 0 \begin{cases} j = 1, 2, \dots, (i - 1) \\ j = (i + 2), (i + 3), \dots, N_w \\ k = 1, 2, \dots, n \end{cases} \quad (\text{A1.82})$$

where i is determined by equation (A1.63).

The necessary condition for a stationary solution is

$$0 = H_\theta = f_\theta^T \cdot \lambda \quad (\text{A1.83})$$

where

$$f_\theta = \frac{\partial f}{\partial \theta} \quad (\text{A1.84})$$

which expands to

$$f_\theta = \begin{bmatrix} -\sin \theta & \cos \theta & 0^{1 \times N_w} & 0^{1 \times N_w} \end{bmatrix}^T \quad (\text{A1.85})$$

Substituting into equation (A1.83) leads to

$$H_\theta = -\lambda_x \sin \theta + \lambda_y \cos \theta = 0 \quad (\text{A1.86})$$

The final condition on $\lambda(t_f)$ is given by

$$\lambda(t_f) = \psi_s^T|_{t=t_f} \cdot \mathbf{v} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{v}_x \\ \mathbf{v}_y \end{bmatrix} = \begin{bmatrix} \mathbf{v}_x \\ \mathbf{v}_y \end{bmatrix} \quad (\text{A1.87})$$

The transversality condition that determines the optimal final time is

$$\begin{aligned} 0 &= \Omega = H(t_f) + 1 \\ &= [\lambda_x(\cos \theta + a(x)^T(y \cdot U_y)) + \lambda_y(\sin \theta + a(x)^T V_b) + 1]_{t=t_f} \end{aligned} \quad (\text{A1.88})$$

The nominal winds are set to zero so that the nominal optimal path is a straight line from the initial point to the final point at $\theta = \pi$ rad (fig. A1.5).

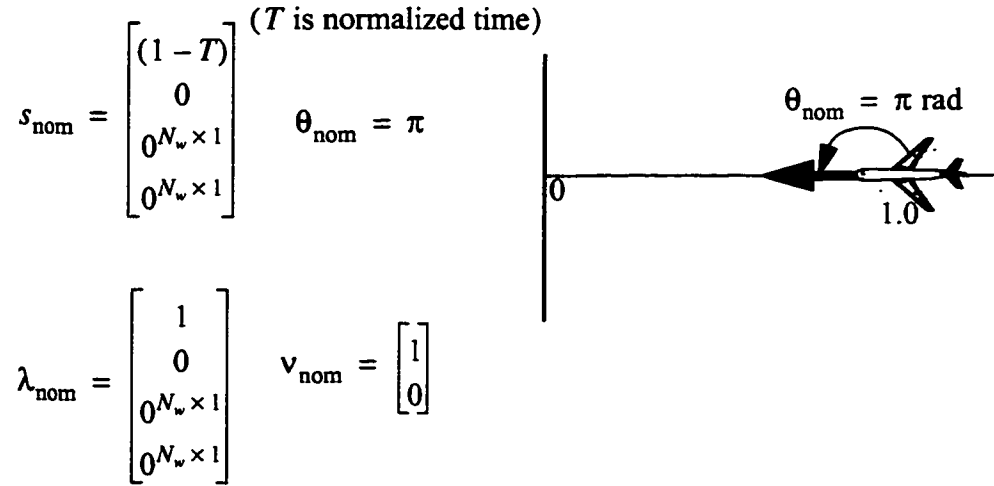


Figure A1.5. Nominal optimal solution in normalized coordinates.

A1.2.3 Computing NOC Gains via Backward Sweep Method

The neighboring optimal control gains are computed numerically using the Backward Sweep method [39]. The gains are then tabulated for look-up during the application of the neighboring optimal control law.

The solution for the neighboring optimal perturbation in heading angle is given by

$$\delta\theta(t) = -K_s\delta s - K_\psi d\psi \quad (\text{A1.89})$$

where the neighboring optimal feedback gain vectors are defined as

$$K_s(t) \equiv H_{\theta\theta}^{-1} [H_{\theta s} + f_{\theta}^T (\bar{S} - \bar{R}\bar{Q}^{-1}\bar{R}^T)] \quad (\text{A1.90})$$

$$K_\psi(t) \equiv H_{\theta\theta}^{-1} [f_{\theta}^T \bar{R}\bar{Q}^{-1}] \quad (\text{A1.91})$$

and δs is the state perturbation vector given by

$$\delta s \equiv s(t) - s_{\text{nom}}(t) \quad (\text{A1.92})$$

and $d\psi$ is the perturbation in the desired final state, given by

$$d\psi \equiv \psi - \psi_{\text{nom}} = \psi \quad (\text{A1.93})$$

The perturbation in the final time is computed from the following equation:

$$dt_f = -K_{ts}\delta s - K_{t\psi}d\psi \quad (\text{A1.94})$$

where the feedback gains are defined as

$$K_{ts} \equiv \frac{m^T}{\alpha} - \frac{q^T}{\alpha} \bar{Q}^{-1} \bar{R}^T \quad (\text{A1.95})$$

$$K_{t\psi} \equiv \frac{q^T}{\alpha} \bar{Q}^{-1} \quad (\text{A1.96})$$

where \bar{S} , \bar{R} , and \bar{Q} are defined as

$$\bar{S} = S - \frac{mm^T}{\alpha} \quad (\text{A1.97})$$

$$\bar{R} = R - \frac{mq^T}{\alpha} \quad (\text{A1.98})$$

$$\bar{Q} = Q - \frac{qq^T}{\alpha} \quad (\text{A1.99})$$

The terms S , R , Q , m , q , and α are computed by integrating the following set of nonlinear differential matrix equations, with specified initial conditions, from t_f backwards to t_0 :

$$\begin{aligned} \dot{S} &= -SA - A^T S + SBS - C \\ S(t_f) &= \left(\frac{\partial^2 \Phi}{\partial s^2} \right)_{t=t_f} = \frac{\partial^2}{\partial s^2} (t_f + v^T \Psi(t_f)) = 0^{n \times n} \end{aligned} \quad (\text{A1.100})$$

$$\begin{aligned} \dot{R} &= -(A^T - SB)R \\ R(t_f) &= \left(\frac{\partial \Psi}{\partial s} \right)^T_{t=t_f} = \begin{bmatrix} 1 & 0 & 0^{1 \times N_w} & 0^{1 \times N_w} \\ 0 & 1 & 0^{1 \times N_w} & 0^{1 \times N_w} \end{bmatrix}^T \end{aligned} \quad (\text{A1.101})$$

$$\dot{Q} = R^T B R, \quad Q(t_f) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (\text{A1.102})$$

$$\begin{aligned} \dot{m} &= -(A^T - SB)m, \quad m(t_f) = \left(\frac{\partial \Omega}{\partial s} \right)^T_{t=t_f} \\ \left(\frac{\partial \Omega}{\partial s} \right)^T_{t=t_f} &= H_s(t_f) = f_s \cdot \lambda|_{t=t_f} = \begin{bmatrix} 0 \\ (N_w - 1)(V_{b2} - V_{b1}) \\ 0^{N_w \times 1} \\ 0^{N_w \times 1} \end{bmatrix} \end{aligned} \quad (\text{A1.103})$$

$$\dot{q} = R^T B m, \quad q(t_f) = \left(\frac{d\psi}{dt} \right)_{t=t_f} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (\text{A1.104})$$

$$\dot{\alpha} = m^T B m, \quad \alpha(t_f) = \left(\frac{d\Omega}{dt} \right)_{t=t_f} = 1 \quad (\text{A1.105})$$

The variables A , B , and C are defined as

$$A \equiv f_s - f_\theta H_{\theta\theta}^{-1} H_{\theta s} \quad (\text{A1.106})$$

$$B \equiv f_\theta H_{\theta\theta}^{-1} f_\theta^T \quad (\text{A1.107})$$

$$C \equiv H_{ss} - H_{s\theta} H_{\theta\theta}^{-1} H_{\theta s} \quad (\text{A1.108})$$

where f_s and f_θ are determined by equation (A1.76) and equation (A1.85) respectively.

The various derivatives of H are given by

$$H_{\theta\theta} = \lambda^T \cdot f_{\theta\theta} = -\cos\theta = 1 \quad (\text{A1.109})$$

$$H_{\theta s} = 0^{1 \times n} \quad (\text{A1.110})$$

$$H_{s\theta} = H_{\theta s}^T \quad (\text{A1.111})$$

$$H_{ss} = f_{ss}^T \cdot \lambda \quad (\text{A1.112})$$

where f_{ss} is a 3-dimensional matrix defined as

$$f_{ss} \equiv \left\langle \left[\frac{\partial}{\partial s} \left(\frac{\partial f_1}{\partial s} \right) \right] \left[\frac{\partial}{\partial s} \left(\frac{\partial f_2}{\partial s} \right) \right] \cdots \left[\frac{\partial}{\partial s} \left(\frac{\partial f_n}{\partial s} \right) \right] \right\rangle \in R^{n \times n | n} \quad (\text{A1.113})$$

and the multiplication of f_{ss}^T with λ is defined as

$$f_{ss}^T \cdot \lambda \equiv \sum_{i=1}^n \left(\left[\frac{\partial}{\partial s} \left(\frac{\partial f_i^T}{\partial s} \right) \right] \cdot \lambda_i \right) \in R^{n \times n} \quad (\text{A1.114})$$

Note that f_{ss} is a piecewise varying function of x because of the piecewise varying wind functions.

From these equations, it is straightforward (but perhaps tedious) to use numerical integration routines (e.g. MATLAB function `ode45()` [68]) to determine the time histories of the \bar{S} , \bar{R} , and \bar{Q} matrices to solve for the neighboring optimal feedback gains.

One final simplification is that the feedback gains associated with perturbations in desired final state, $d\psi$, are not required. Optimal solutions for different final positions are achieved through the normalization and rotation to different coordinate systems so that the $d\psi$ gains aren't needed.

The feedback gains are then tabulated as a function of longitudinal distance-to-go. The values must be spaced closely enough that when a particular interpolation method (e.g. linear interpolation) is used to look up the gain values, the error is within tolerable limits.

The number of grid points, N_w , is only limited by the amount of computer memory available to integrate the backward sweep equations and by numerical precision. The S matrix in the matrix Riccati equation, equation (A1.100), is $S \in R^{n \times n}$. If just 10 grid points are used, then $n = 2 + 2N_w = 22$. Since some of the sweep equations are coupled, many more equations must be integrated concurrently. Some reduction in effort can be made by taking advantage of symmetry. More simplification can be achieved by noting that many elements of these matrices are identically zero for this specific problem, but software for computing solutions to general problems cannot assume that any of the elements are zero.

Appendix 2

The Rapid Update Cycle

*The Rapid Update Cycle (RUC) is an operational atmospheric prediction system comprised primarily of a numerical forecast model and an analysis system to initialize that model. MAPS [Mesoscale Analysis and Prediction System] is the research counterpart to the RUC. The RUC has been developed to serve users needing short-range weather forecasts, including those in the US aviation community.**

The RUC is essentially a computational fluid dynamics (CFD) model that provides quasi-real-time atmospheric forecast data on a grid spanning the continental United States from the ground level up to approximately 50,000 ft above mean sea level. The basic data include horizontal wind-vector component magnitudes, temperature, and geopotential height as a function of pressure, but often hundreds of additional atmospheric data parameters are also provided. Forecasts are provided from the analysis initiation time (called an *analysis* file) through a 12-hr forecast, in 1-hr increments. The data are available in several different horizontal grid definitions of varying precision, and several different vertical coordinate definitions. Typically, the RUC analysis is run every hour, but the cycle takes a finite amount of time to run so that the results are not available for some minutes or hours after the analysis time. For this reason, real-time users of the RUC must use one of the forecast data files, or interpolate between forecast files, to obtain current predicted atmospheric conditions.

* Introductory statement from the National Oceanic and Atmospheric Administration (NOAA) Forecast Systems Laboratory (FSL) RUC/MAPS Information website:
<http://maps.fsl.noaa.gov>

The RUC data are provided in a format called Gridded Binary (GRIB), which is a compressed format for storing large gridded data sets. The GRIB is the World Meteorological Organization's gridded data standard and is well-supported by several organizations on various Internet sites [80]. Several sites include software source code and utilities for reading and processing GRIB data files (e.g. WGRIB, found at: <http://wesley.wvb.noaa.gov/wgrib.html>). Working with some of these software products can be exasperating, but with patience, decoding and processing GRIB RUC files becomes routine.

The National Centers for Environmental Prediction (NCEP), a part of NOAA, run the operational RUC and currently (as of October 2002) provide the data at the following http and ftp server sites:

<http://www.nco.ncep.noaa.gov/pmb/products/ruc2/>

<ftp://ftpprd.ncep.noaa.gov/pub/data/nccf/com/ruc/prod>

An excellent website to visit for information about the RUC is

<http://ruc.fsl.noaa.gov>

References

- [1] Bilimoria, K. D., and Lee, H. Q., "Properties of Air Traffic Conflicts for Free and Structured Routing," AIAA 2001-4051, AIAA Guidance, Navigation, and Control Conference, Montreal, Canada, Aug. 6-9, 2001.
- [2] Bertsimias, D., Tsitsiklis, J. N., *Introduction to Linear Optimization*, Chapt. 11, Athena Scientific, Belmont, MA, 1997.
- [3] Williams, C. P., Clearwater, S. H., *Explorations in Quantum Computing*, Springer-Verlag, New York, NY, 1998.
- [4] Nielsen, M. A., Chuang, I. L., *Quantum Computation and Quantum Information*, Cambridge University Press, Cambridge, UK, 2000.
- [5] Reif, J., Sharir, M., "Motion Planning in the Presence of Moving Obstacles," *Journal of the Association for Computing Machinery*, Vol. 41, No. 4, July 1994, pp. 764-790.
- [6] Dean, G., Fron, X., Miller, W., Nicolaon, J-P., "ARC 2000: An Investigation into the Feasibility of Automatic Conflict Detection and Resolution," *Air Traffic Control Quarterly*, Vol. 3, No. 4, 1995, pp. 229-259.
- [7] Hu, J., Prandini, M., Nilim, A., and Sastry, S., "Optimal Coordinated Maneuvers for Three Dimensional Aircraft Conflict Resolution," *Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 5, Sept.-Oct. 2002, pp. 888-900. (also presented as AIAA 2001-4294, AIAA Guidance, Navigation, and Control Conference, Montreal, Canada, Aug. 6-9, 2001.)
- [8] Mondoloni, S., Conway, S., "An Airborne Conflict Resolution Approach Using a Genetic Algorithm," AIAA 2001-4054, AIAA Guidance, Navigation, and Control Conference, Montreal, Canada, Aug. 6-9, 2001.
- [9] Hu, X., Wu, S-F., and Jiang, J., "GA Based On-line Real-time Optimization of Commercial Aircraft's Flight Path for a Free Flight Strategy," AIAA 2001-4234, AIAA Guidance, Navigation, and Control Conference, Montreal, Canada, Aug. 6-9, 2001.
- [10] Sridhar, B., and Chatterji, G. B., "Computationally Efficient Conflict Detection Methods for Air Traffic Management," 1997 American Control Conference, Albuquerque, NM, June 4-6, 1997.
- [11] Menon, P. K., Sweriduk, G. D., Sridhar, B., "Optimal Strategies for Free-Flight Air Traffic Conflict Resolution," *Journal of Guidance, Control, and Dynamics*, Vol. 22, No. 2, March-April 1999, pp. 202-211.
- [12] Paielli, R.A.: "A Linear Altitude Rule for Safer and More Efficient Enroute Air Traffic," *Air Traffic Control Quarterly*, vol. 8, no. 3, Fall 2000.
- [13] Sietzen, F. Jr., "Emptying the Skies," *Aerospace America*, Vol. 41, No. 1, Jan. 2003.

- [14] Erzberger, H., Pecsvaradi, T., "4-D Guidance System Design with Application to STOL Air Traffic Control," A72-38252, 13th Joint Automatic Control Conference, Stanford, CA, Aug. 16-18, 1972, pp. 445-454.
- [15] Erwin, R. L., "'Strategic' Time-Based ATC," *Astronautics & Aeronautics*, V. 16, No. 11, Nov., 1978, pp. 56-61.
- [16] Benoît, A., Swierstra, S., "A Cost-Efficient Control Procedure for the Benefit of All Airspace Users," ACARD-CP-360, Guidance & Control Panel, 39th Symposium, Izmir, Turkey, 16-19 Oct., 1984.
- [17] Erzberger, H., Tobias, L., "A Time-Based Concept for Terminal-Area Traffic Management," AGARD-CP-410, Brussels, Belgium, 1986., pp. (52-1)-(52-14).
- [18] Credeur, L., and Capron, W. R., "A Description and Evaluation of 'TIMER' -- A Time-Based Terminal Flow-Control Concept," AGARD-AG-301, Vol. 2, May 1990, pp. 27-1 - 27-42.
- [19] Wilson, I. A. B., "PHARE: Definition and use of Tubes," DOC 96-70-18, Eurocontrol, Rue de la Fusée 96, B-1130, Brussels, Belgium, 15 July, 1996 [Available at <http://www.eurocontrol.int/phare/>].
- [20] Post, W., "PHARE Demonstration 3: A Contribution to the Future of Air Traffic Management," 3rd USA/Europe Air Traffic Management Research and Development Seminar, Napoli, Italy, 13-16 June 2000.
- [21] RTCA, Inc., "Free Flight Implementation," Final Report RTCA Task Force 3, RTCA, Inc., Washington, DC, Oct. 1995.
- [22] Tomlin, C., Pappas, G., et al, "A Next Generation Architecture for Air Traffic Management," Proceedings of the Automation for Robotics and Control Workshop, San Diego, CA, Dec. 1997.
- [23] Ghosh, R., and Tomlin, C., "Maneuver Design for Multiple Aircraft Conflict Resolution," Proceedings of the American Control Conference, Chicago, IL, June, 2000, pp. 672-676.
- [24] Erzberger, H., Paielli, R., "Concept for Next Generation Air Traffic Control System," *Air Traffic Control Quarterly*, Vol. 10, No. 4, 2002.
- [25] Erzberger, H., Barman, J. F., and McLean, J. D., "Optimum Flight Profiles for Short Haul Missions," AIAA75-1124, AIAA Guidance and Control Conference, Boston, MA, Aug. 20-22, 1975.
- [26] Erzberger, H., Lee, H., "Constrained Optimum Trajectories with Specified Range," *Journal of Guidance and Control*, Vol. 3, Jan-Feb, 1980, pp. 78-85.
- [27] Sorenson, J. A., Waters, M. H., "Airborne Method to Minimize Fuel with Fixed Time-of-Arrival Constraints," *AIAA Journal of Guidance and Control*, Vol. 4, No. 3, 1981.
- [28] Burrows, J. W., "Fuel-Optimal Aircraft Trajectories with Fixed Arrival Times," *Journal of Guidance*, Vol. 6, No. 1, 1983.
- [29] Chakravarty, A., Vagners, J., "4-D Aircraft Flight Path Management in Real Time," 1983 American Control Conference, San Francisco, CA, June 1983, pp. 794-795.
- [30] Chakravarty, A., "Four-Dimensional Fuel-Optimal Guidance in the Presence of Winds," *Journal of Guidance*, Vol. 8, No. 1, 1985.
- [31] Lidén, S., "Practical Considerations in Optimal Flight Management Computations," *Journal of Guidance*, Vol. 9, No. 4, Jul-Aug, 1986, pp. 427-432.
- [32] Lidén, S., "Optimum Cruise Profiles in the Presence of Winds," 1992 IEEE/AIAA 11th Digital Avionics Systems Conference, pp. 254-261.
- [33] Lidén, S., "Optimum 4-D Guidance for Long Flights," 1992 IEEE/AIAA 11th Digital Avionics Systems Conference, pp. 262-267.
- [34] Hagelauer, P., Mora-Camino, F., "A Soft Dynamic Programming Approach for On-line Aircraft 4D-Trajectory Optimization," *European Journal of Operational Research*, No. 107, 1998, pp. 87-95.

- [35] Bijlsma, S. J., "A Computational Method for the Solution of Optimal Control Problems in Ship Routing," *Navigation Journal of the Institute of Navigation*, Vol. 48, No. 3, Fall 2001, pp. 145-154.
- [36] Zermelo, E., "Über das Navigationsproblem bei ruhender oder veränderlicher Windverteilung ['On the Navigation Problem in Calm or Variable Wind Distributions']," *Zeitschrift für angewandte Mathematik und Mechanik*, Vol. 11, 1931, pp. 114-124.
- [37] Zermelo, E., "Über die Navigation in der Luft als Problem der Variationsrechnung ['On the Navigation in the Air as a Problem of the Calculus of Variations']," *Jahresbericht der Deutschen Mathematiker-Vereinigung*, Vol. 39, 2nd section, 1930, pp. 44-48.
- [38] Erzberger, H., Lee, H. Q., "Optimum Guidance Techniques for Aircraft," *Journal of Aircraft*, Vol. 8, No. 2, 1971, pp. 95-101.
- [39] Bryson, A. E. Jr., Ho, Y. C., *Applied Optimal Control: Optimization, Estimation, and Control*, rev., Hemisphere, New York, 1975.
- [40] Jardin, M. R., Bryson, A. E. Jr., "Neighboring Optimal Aircraft Guidance in Winds," *Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 4, Jul-Aug 2001, pp. 710-715.
- [41] Jeppesen PC Development Group, "Autorouting Algorithm Analysis: FliteStar vs. FliteSoft and Destination Direct," Technical White Paper, Jeppesen Dataplan, Los Gatos, CA, July 2000 (available at <http://www.jeppesen.com>).
- [42] McVey, C. B., Clements, D. P., et al, "Worldwide Aeronautical Route Planner," *Proceedings of the National Conference on Artificial Intelligence*, 1999, pp. 916-917.
- [43] Reinkins, L., personal communication, April 23, 2002.
- [44] Bilimoria, K., Sridhar, B., et al., "FACET: Future ATM Concepts Evaluation Tool," *Air Traffic Control Quarterly*, Vol. 9, No. 1, 2001.
- [45] Durand, N., Alliot, J-M., and Chansou, O., "Optimal Resolution of En Route Conflicts," *Air Traffic Control Quarterly*, Vol. 3, No. 3, 1995, pp. 139-161.
- [46] Paielli, R. A., and Erzberger, H., "Conflict Probability Estimation for Free Flight," NASA TM-10411, Oct. 1996.
- [47] Foudriat, E., "Aircraft 4-D Constant Velocity Control System," *Journal of Aircraft*, Vol. 11, No. 6, June 1974, pp. 326-333.
- [48] Nagarajan, N., "Four-Dimensional Guidance Problem with Control Delays," *Journal of Aircraft*, Vol. 13, No. 8, August 1976, pp. 559-564.
- [49] Knox, C. E., "Experimental Determination of the Navigation Error of the 4-D Navigation, Guidance, and Control Systems on the NASA B-737 Airplane," AGARD-CP-240, Dayton, Ohio, 17-20, Oct. 1977, pp. (25-1)-(25-14).
- [50] Menga, G., Erzberger, H., "Time-Controlled Descent Guidance in Uncertain Winds," *Journal of Guidance and Control*, Vol. 1, No. 2, March-April, 1978, pp. 123-129.
- [51] Erzberger, H., McLean, J. D., "Fuel-Conservative Guidance System for Powered-Lift Aircraft," A79-45363, AIAA Guidance and Control Conference, Boulder, CO, 6-8 Aug., 1979, pp. 114-128.
- [52] Moor, D. A., "Time Flies: An In-Service Evaluation of a 4-D Flight Management System," AIAA84-2607, AIAA/IEEE 6th Digital Avionics Systems Conference, Baltimore, MD, 1984, pp. 54-59.
- [53] Benoit, A., Swierstra, S., "Ground-Based 4-D Guidance of Flights in Strong Winds," *Journal of Navigation*, Vol. 43, No. 2, May 1990, pp. 176-186.
- [54] Erwin, R. L., Izumi, K. H., "Use of 4D RNAV in Time-Based En Route Arrival Metering," AGARD-AG-301, Vol. 2, May 1990, pp. (28-1)-(28-19).
- [55] Williams, D. H., Green, S. M., "Airborne Four-Dimensional Flight Management in a Time-Based Air Traffic Control Environment," NASA TM 4249, March 1991.

- [56] Kaminer, I., O'Shaughnessy, P., "Integration of Four-Dimensional Guidance with Total Energy Control System," *Journal of Guidance*, Vol. 14, No. 3, June 1991, pp. 564-573.
- [57] Adam, V., Kohrs, R., "On Board Planning of 4D-Trajectories," AGARD-CP-504, Amsterdam, The Netherlands, 22-24, Oct., 1991, pp. (16-1)-(16-12).
- [58] Stiharu-Alexe, I., O'Shea, J., "Four-Dimensional Guidance of Atmospheric Vehicles," *Journal of Guidance, Control, and Dynamics*, Vol. 19, No. 1, Jan.-Feb. 1996, pp. 113-122.
- [59] Jardin, M. R., "4D Air Traffic Control for Non-4D-Equipped Aircraft," AIAAF-97063, American Control Conference, Albuquerque, NM, June 4 - 6, 1997.
- [60] Jardin, M., Heraud, J., Bajikar, S., "Time-Based GPS Trajectory Guidance," Institute of Navigation National Technical Meeting, Anaheim, CA, Jan. 26-28, 2000.
- [61] Jardin, M., "Ideal Free-Flight through Multiple-Aircraft Neighboring Optimal Control," ACC00-AIAA0012, American Control Conference, Chicago, IL, June 28-30, 2000.
- [62] Cole, R. E., Richard, C., et al, "An Assessment of the 60 km Rapid Update Cycle (RUC) with Near Real-Time Aircraft Reports," Project Report NASA/A-1, MIT Lincoln Laboratory, 15 July, 1998 (available through the National Technical Information Service, Springfield, VA 22161).
- [63] Benjamin, S.G., J.M. Brown, K.J. Brundage, B.E. Schwartz, T.G. Smirnova, T.L. Smith, L.L. Morone, 1998: RUC-2 - The Rapid Update Cycle Version 2. NWS Technical Procedure Bulletin No. 448. NOAA/NWS, 18 pp. [National Weather Service, Office of Meteorology, 1325 East-West Highway, Silver Spring, MD 20910]
- [64] Bryson, A. E. Jr., *Dynamic Optimization*, Addison Wesley Longman, Inc., Menlo Park, CA, 1999.
- [65] Oussedik, S., Delahaye, D., and Schoenauer, M., "Air Traffic Management by Stochastic Optimization," 2nd USA/Europe Air Traffic Management Research and Development Seminar, Orlando, FL, Dec. 1-4, 1998.
- [66] Heymann, V. I., Ben-Asher, J. Z., "Aircraft Trajectory Optimization in the Horizontal Plane," *Journal of Guidance*, Vol. 20, No. 6, Nov.-Dec., 1997, pp. 1271-1274.
- [67] Jardin, M., Erzberger, H., "Atmospheric Data Acquisition and Interpolation for Enhanced Trajectory Prediction Accuracy in the Center-TRACON Automation System", AIAA96-0271, 34th Aerospace Sciences Meeting, Reno, NV, Jan. 15 - 18, 1996.
- [68] MATLAB Version 5.3, The Mathworks, Inc., Jan. 21, 1999.
- [69] Kuchar, J., and Yang, L. C., "Survey of Conflict Detection and Resolution Modeling Methods," AIAA Guidance, Navigation, and Control Conference, New Orleans, LA, 1997.
- [70] Tomlin, C., Pappas, G. J., Sastry, S., "Conflict Resolution for Air Traffic Management: A Study in Multiagent Hybrid Systems," *IEEE Transactions on Automatic Control*, Vol. 43, No. 4, April 1998, pp. 509-521.
- [71] Isaacson, D. R., and Erzberger, H., "Design of a Conflict Detection Algorithm for the Center/ TRACON Automation System," 16th Digital Avionics Systems Conference, Irvine, CA, Oct. 26-30, 1997.
- [72] Chiang, Y.-J., Klosowski, J. T., et al, "Geometric Algorithms for Conflict Detection/Resolution in Air Traffic Management," Proc. of the 36th IEEE Conference on Decision and Control, 1997.
- [73] Erzberger, H., Paielli, R., et al, "Conflict Detection and Resolution in the Presence of Prediction Error," 1st USA/Europe Air Traffic Management R&D Seminar, Saclay, France, 17-20 June, 1997.
- [74] Krozel, J., Peters, M., "Conflict Detection and Resolution for Free Flight," *Air Traffic Control Quarterly*, Vol 5. No. 3, 1997, pp. 181-212.
- [75] Bilimoria, K., "A Geometric Optimization Approach to Aircraft Conflict Resolution," AIAA 2000-4265, AIAA Guidance, Navigation, and Control Conference, Denver, Colorado, 14-17 Aug., 2000.

- [76] Eby, M. S., "A Self-Organizational Approach for Resolving Air Traffic Conflicts," *The Lincoln Laboratory Journal*, Vol. 7, No. 2, 1994, pp. 239-254.
- [77] Innocenti, M., Gelosi, P., et al, "Air Traffic Management Using Probability Function Fields," AIAA-99-4149, AIAA Guidance, Navigation, and Control Conference, August, 1999.
- [78] Frazzoli, E., Mao, Z.-H., Oh, J.-H., Feron, E., "Resolution of Conflicts Involving Many Aircraft via Semidefinite Programming," *Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 1, Jan.-Feb., 2001, pp. 79-86.
- [79] Schwartz, B., Benjamin, S., Green, S., Jardin, M., "Accuracy of RUC-1 and RUC-2 Wind and Aircraft Trajectory Forecasts by Comparison with ACARS Observations," *Weather and Forecasting*, June 14, 1999.
- [80] Dey, Clifford H., "The WMO Format for the Storage of Weather Product Information and the Exchange of Weather Product Messages in Gridded Binary Form as Used by NCEP Central Operations," Office Note 388, U.S. Dept. of Commerce, National Oceanic and Atmospheric Administration, National Weather Service, National Centers for Environmental Prediction, March 10, 1998 [available through many sites on the Internet].