NASA/CR–2012-217580

Development of Complexity Science and Technology Tools for NextGen Airspace Research and Applications

Bruce J. Holmes, Bruce K. Sawhill, Jim Herriot, and Ken Seehart
NextGen AeroSciences, LLC, Williamsburg, Virginia

Dres Zellweger
Consultant, Rockville, Maryland

Rick Shay
DoubleBlack Aviation, Frisco, Colorado

June 2012

NASA STI Program . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA scientific and technical information (STI) program plays a key part in helping NASA maintain this important role.

The NASA STI program operates under the auspices of the Agency Chief Information Officer. It collects, organizes, provides for archiving, and disseminates NASA's STI. The NASA STI program provides access to the NASA Aeronautics and Space Database and its public interface, the NASA Technical Report Server, thus providing one of the largest collections of aeronautical and space science STI in the world. Results are published in both non-NASA channels and by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA Programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers, but having less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services also include organizing and publishing research results, distributing specialized research announcements and feeds, providing information desk and personal search support, and enabling data exchange services.

For more information about the NASA STI program, see the following:

- Access the NASA STI program home page at *http://www.sti.nasa.gov*

- E-mail your question to help@sti.nasa.gov

- Fax your question to the NASA STI Information Desk at 443-757-5803

- Phone the NASA STI Information Desk at 443-757-5802

- Write to:
  STI Information Desk
  NASA Center for AeroSpace Information
  7115 Standard Drive
  Hanover, MD 21076-1320

NASA/CR–2012-217580

Development of Complexity Science and Technology Tools for NextGen Airspace Research and Applications

Bruce J. Holmes, Bruce K. Sawhill, Jim Herriot, and Ken Seehart
NextGen AeroSciences, LLC, Williamsburg, Virginia

Dres Zellweger
Consultant, Rockville, Maryland

Rick Shay
DoubleBlack Aviation, Frisco, Colorado

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23681-2199

Prepared for Langley Research Center
under Contract NNL10AA29C

June 2012

Available from:

NASA Center for AeroSpace Information
7115 Standard Drive
Hanover, MD 21076-1320
443-757-5802

TABLE OF CONTENTS

ABSTRACT

The objective of this research by NextGen AeroSciences, LLC is twofold: 1) to deliver an initial "toolbox" of algorithms, agent-based structures, and method descriptions for introducing trajectory agency as a methodology for simulating and analyzing airspace states, including bulk properties of large numbers of heterogeneous 4D aircraft trajectories in a test airspace – while maintaining or increasing system safety; and 2) to use these tools in a test airspace to identify possible phase transition structure to predict when an airspace will approach the limits of its capacity. These 4D trajectories continuously replan their paths in the presence of noise and uncertainty while optimizing performance measures and performing conflict detection and resolution. In this approach, trajectories are represented as extended objects endowed with pseudopotential, maintaining time and fuel-efficient paths by bending just enough to accommodate separation while remaining inside of performance envelopes. This trajectory-centric approach differs from previous aircraft-centric distributed approaches to deconfliction.

The results of this project are the following: 1) we delivered a toolbox of algorithms, agent-based structures and method descriptions as pseudocode; and 2) we corroborated the existence of phase transition structure in simulation with the addition of "early warning" detected prior to "full" airspace. This research suggests that airspace "fullness" can be anticipated and remedied before the airspace becomes unsafe.

Keywords:

Traffic physics; phase state; phase transition; aircraft trajectory optimization; airspace capacity; optimal control; real-time optimization; air traffic conflict resolution; 4D trajectory; 5D trajectory; pseudopotential method; separation; deconfliction; Trajectory Based Operations (TBO); NextGen.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

3SAT	The Satisfiability Construct for all NP-hard problems
4DT	Four Dimensional Trajectories
5DT	Five Dimensional Trajectories (three position variables plus current time and future time variables)
ABM	Agent-Based Modeling
ANSP	Air Navigation Service Provider
AOC	Airline Operations Center
ATM	Air Traffic Management
ATOP	Advanced Technologies & Oceanic Procedures (FAA Ocean 21 Prog.)
ATSP	Air Transportation Service Provider
CUDA	Compute Unified Device Architecture
DARP	Dynamic Airspace Reroute Program
DCIT	Data Communications Implementation Team (FAA)
FANS	Future Air Navigation System
FMC	Flight Management Computer
JPDO	Joint Planning and Development Office
NAS	National Airspace System
NextAero	NextGen AeroSciences, LLC
NextGen	Next Generation Air Transportation System
PBC	Performance-Based Communication
PBN	Performance-Based Navigation
PBS	Performance-Based Surveillance
RBT	Reference Business Trajectory
RNP	Required Navigation Performance
RTP	Required Time Performance
RVSM	Reduced Vertical Separation Minimums
SAA	Sense and Avoid
SESAR	Single European Sky Advanced Research
TBO	Trajectory-Based Operations (of airspace)
UAS	Uncrewed Aerial Systems

ACKNOWLEDGEMENTS

The authors would like to thank Drs. John Cavolowski, Akbar Sultan, Parimal Kopardekar, and Leighton Quon of NASA for their encouragement to explore the possibilities in applying advancements from complexity science to airspace research and operations. We express special gratitude to Dr. Natalia Alexandrov of NASA Langley for her support, inspiration, and critique of this research. In addition, numerous diligent and dedicated NASA researchers provided counsel and guidance of great value in developing the path taken in this project. We greatly appreciate the project management support provided by several staff of NIA, especially Fred Brooks. The contributions of Connie K. Holmes in providing editorial counsel for the final report are gratefully acknowledged. This research was funded beginning in July 2010 under a NASA Airspace Program NRA contract, titled: "Development of Complexity Science and Technology Tools for NEXTGEN Airspace Research and Applications," with the National Institute for Aerospace, and NextGen AeroSciences, LLC of Virginia USA as the lead technical performing organization. The NASA contract number is NNL10AA29C.

1. INTRODUCTION

1.1 Objectives

The objectives of this research and results delivered are the following:

1. Deliver a "toolbox" of algorithms, agent-based structures and method descriptions for simulating and analyzing airspace states, while maintaining or increasing safety, involving large numbers of heterogeneous 4D aircraft trajectories in a test airspace;

2a. Use these tools in a test airspace to identify effective approaches for separation assurance for 4D aircraft trajectories in the test airspace; and

2b. Develop a traffic physics/phase transition description and algorithmic measures to predict when an airspace will approach the limits of its capacity.

In achieving the results described herein, we employed an approach to airspace research involving an up-leveling of the dynamics of aircraft and aircraft trajectories to fully *dynamical trajectories (employing continuous replanning)*, and managing the airspace in terms of its *bulk properties as generated by replanning*

Furthermore, we characterized entire airspaces as solvable (or not)–within the limits of available computational resources—while accounting for the physical constraints of aircraft using the airspace as well as short-lived constraints such as simulated weather events.

In the abstract, the airspace can be thought of as containing a *gas* of *trajectories*, continuously interacting and replanning as they interact with each other via a repulsive pseudopotential and incorporate internal (discrepancy between forecast and flown trajectories) and external (wind field and weather objects) influences. At low density the interaction is minimal and all the trajectories are easily separated from each other with each trajectory being very close to optimal. At higher density, the airspace may exhibit emergent collective properties analogous to road or pedestrian traffic exhibiting various kinds of crowding phenomena (traffic jams) as a result of cascading interactions between trajectory pairs.

This approach enables a better understanding of the overall properties of the airspace, by characterizing capacity limits, and forecasting congestion and other suboptimal behaviors before they occur. We examine the phase structure of the airspace using tools drawn from the science of traffic physics, in search of possible phase transition structure, as well as precursors to these phase transitions in space or time that could be used as part of a mitigation methodology.

1.2 5DT Dynamical Trajectories

At the most elementary physical level, the airspace consists of air, aircraft and obstacles (weather cells, closed airspace, etc.). In general, there are two overarching objectives involved in the management of the airspace. The first objective is always safety. Given the first, the second

objective is efficiency. Efficiency is measured in terms of resource usage by the system, most commonly fuel burn and time aloft associated with all of the aircraft in the airspace.
Safety is generally thought of in terms of aircraft position and heading. At a given time, no two aircraft can be too close nor on a path that will incur a separation violation. Furthermore, heading information gives predictive power and allows evasive action to take place so as to avoid predicted future conflicts, either mediated by ground control or between the aircraft themselves if they have the necessary technology to do so. When one seeks to measure the efficiency of airspace usage, it is apparent that the most important questions of time and fuel are associated with trajectories, *not* with aircraft positions and headings. If one is given a 4DT trajectory of an aircraft plus ancillary information such as initial weight, aircraft type, and wind field, it is possible to accurately predict time aloft and fuel burn.. Similarly, an optimization procedure can produce a 4DT trajectory given an objective function and initial conditions in the airspace. We desire a problem definition that can naturally address the issues of both safety and efficiency using the same formalism.

In order to accomplish this result, we introduce the notion of a collection of 4DT trajectories, continuously replanning so as to stay optimal and to adapt to new information as it arrives. We call this "5DT," as two time dimensions are required to specify a state of the airspace. For instance, at noontime the predicted positions and future trajectories of the aircraft at 2 pm and beyond will in general be different from the same positions and future trajectories as seen from 1 pm. One needs both a "from" time (when the prediction is being made) and a "to" time (the positions and trajectories at a future time). Our 4DT trajectories themselves change over time, and we keep track of future trajectory information because we seek to continuously maintain the entire airspace in a deconflicted state. This continuous deconfliction is made essential by the combination of an interacting system of trajectories combined with an evolving system of information. (Constraints, such as weather or unforeseen flight alterations, can emerge over time.)

Figure 1-1 shows an artist's rendition of a 5DT trajectory. Over time, the trajectory itself is deformed according to pseudopotential effects exerting pressure on the trajectory, thus changing its shape. The deformation might be to achieve minimum separation or to avoid weather. The actual deformation is constrained to occur in such a way as to minimize curvature and acceleration along the trajectory.

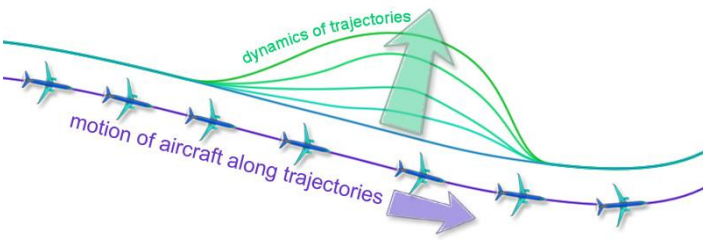

Figure 1-1. 5DT Trajectory.

1.3 Bulk Properties of the Airspace

In addition to endowing the airspace with dynamical trajectories, we also considered very large numbers of dynamical trajectories in the airspace and analyzed all of the dynamical trajectories *en masse* – more like an airspace filled with an interacting gas of dynamical trajectories rather than with individual aircraft. The aircraft interactions occurred in a randomized manner so as to generate results that were not specific to a particular configuration.

This inspiration from statistical physics and its modern traffic physics offshoot provided us with tools for investigating the bulk properties of the entire airspace. In particular, we have created the necessary simulation and modeling tools to examine the phase transition structure of the test airspace as described in this report. It should be noted that phase transitions have been observed in the past in super-dense airspace simulations using a number of different deconfliction techniques, but not in such a way as to generate predictive information that could be used to mitigate undesirable effects.

1.4 Computational Issues

In deconflicting a congested airspace, it is not enough for a solution to exist. A solution must both exist and be discoverable in time to use it. Hence, the amount of computation required to find a solution is as important as the existence of a solution. As we will demonstrate, the approach to phase transition of airspace capacity is manifested in two ways. First, a gradual increase in computational difficulty (computer iterations) can be measured as the phase transition is approached. This measure is one form of the predictive utility of our method for understanding airspace phase state and onset of phase transition. Second, the lead-time available between the present time and future time must be sufficient for mitigating actions to be taken to achieve a conflict-free solution. If the conflicted airspace cannot be deconflicted as that future time is approached, then this decreasing lead-time becomes a second form of predictive utility regarding phase transition. Conceptually, both of these computational issues, computational cycles required to find a solution, and the lead-time required to implement mitigating actions, will be sensitive to airspace scenarios (involving factors of weather, trajectory densities and geometries, airspace architectures and others). These two manifestations of phase transition form the basis for our method's predictive utility.

1.5 Visualizations of Trajectories and Airspace

As a heuristic aid to our research and a help to communicate our research more vividly, our simulation implementations show visualizations of airspaces with their associated trajectories. There are a few differences from previous research visualizations due to the 5DT nature of our work.

The portion of each trajectory in front of each aircraft is the projected future *at a given system time*. This future trajectory can move around continuously with replanning and does not necessarily return to a fixed preplanned trajectory. The notion of a "reference trajectory" is no longer applicable in a continuously replanned and dynamically deconflicted airspace. Examples

of trajectories are shown in the next subsection and throughout this report. Trajectories shown in red are portions of trajectories that are in conflict with other trajectories and are not yet deconfllicted. In general, red occurs both rarely and well into the future except in cases where we drive the simulation to failure.

1.6 Storms (Obstructions)

Our algorithms and implementations can also represent weather cells (storms) as dynamical obstructions in the airspace. Trajectories automatically separate from these storms – as well as other aircraft. Figure 1-2 shows an airspace with three storms and a number of aircraft avoiding each other and the storms. Most of the trajectories are deconflicted, but the computation is interrupted to show a few remaining conflicting trajectories indicated by the portions of trajectories drawn in red.

Storms are specifically designed to have unpredictable trajectories. A set of trajectories may be fully deconflicted at one point in the simulation. As a storm moves, however, new conflicts may suddenly arise – either directly from being too near the storm, or indirectly by the effects of aircraft moving away from storms, creating new conflicts with other nearby aircraft as shown below:

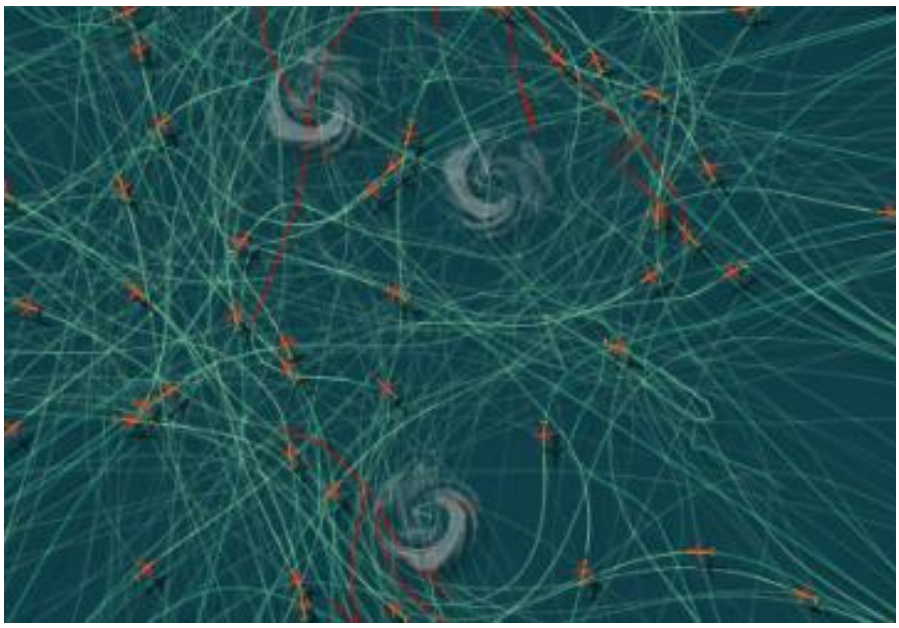

Figure 1-2. High resolution screenshot of conflicted trajectories

1.7 Fundamental Questions

Our experiments enabled the investigation of a suite of fundamental research questions concerning the properties of the test airspace. In this project, we are looking at three key questions:

- Does test airspace exhibit phase behaviors?
- Is it possible to predict phase behaviors in the test airspace?
- Is it plausible to control the test airspace phase state through management of bulk properties (of many trajectories simultaneously)?

In order to accomplish our objectives, we ran experiments in the form of computational simulations in a test airspace. Hence, we designed appropriate algorithms to perform simultaneous trajectory deconfliction and optimization (delivered in the form of pseudocode). We then implemented these algorithms on our own high-performance software platform, generated necessary data, and analyzed the data for phase structure.

We found that our experiments in test airspace not only exhibited phase behaviors (as expected and as shown using different methodologies in the past) but that phase behaviors in our experiments also had predictive "signatures" of phase behavior that could be utilized in a pre-emptive fashion to control the test airspace phase state through management of its bulk properties.

The following chapters provide a scientific background, followed by a description of our approach and the initial algorithm toolbox in the form of pseudocode is detailed in Chapter 4, and listed as an integrated corpus of code in Appendix A. Next we describe our experiments, results, recommendations and conclusions.

2. SCIENCE BACKGROUND

This chapter outlines the motivation and development of trajectory modeling and simulation algorithms based on Agent-Based Modeling (ABM) and traffic physics concepts. We seek to observe and understand collective phenomena arising from many "agents" representing aircraft trajectories that optimize their individual fitness functions in parallel

In addition, trajectory replanning comprises part of the dynamic trajectory management process. In this project, the continual replanning of trajectories incorporates objective functions for the separation and the business case considerations, as well as a pseudopotential "charged string" concept for trajectory separation coupled with trajectory elasticity. The outcome desired is to produce the algorithms that support the testing of concepts of the collective dynamics of large numbers of heterogeneous aircraft (thousands to tens of thousands) in the National Airspace System (NAS) undergoing continuous 4DT trajectory replanning in the presence of noise and uncertainty, while optimizing performance measures and the conflicting trajectories.

In addition to the benefits of powerful simulation capabilities, the nascent science of "traffic physics" [2] provides analytical rigor through insights derived from long-established principles of statistical physics dating back to the late 1800s. It does so by changing the question of trajectory planning from "What is the deconflicted solution to this particular traffic situation?" to "Is there a very high probability that a solution will exist and be computable in this class of traffic situations?" The usefulness of a statistical approach is in the understanding of traffic phenomena that are robust across a broad range of initial airspace configurations and trajectory interaction geometries. Insights based on a traffic physics approach have already proved useful in ground traffic analyses, particularly on European freeways [3].

2.1 Traffic Physics

The science of traffic physics is a new field emerging at the boundary of agent-based modeling and statistical physics. It addresses the statistical properties of large numbers of self-propelled objects acting on their own behalf. To date, the science has largely been applied to roadway vehicle dynamics because of the significant societal and financial import and because the problem is simplified by geometrical constraints. In addition, road traffic systems offer ready access to large amounts of data [3]. This research has applicability to other many-agent systems in addition to roadways [2, 4]. The utility of the science is the ability to define systemic measures that are independent of the particular behaviors of each agent in a traffic system and independent of details of the system itself (such as geometric characteristics), much as the pressure exerted by a gas on its container is independent of the details of motion of each individual molecule in the gas and independent of the shape of the container.

2.1.1 Phases and Phase Transitions

Physical systems consisting of many particles are often characterized in terms of *phase*, such as liquid, solid, or gaseous. The phase is a property of an entire system, rather than of any of its particular components. Systems of interacting agents in freeway traffic have been shown both theoretically and empirically to exhibit phases that correspond to free-flowing ("liquid") or jammed ("solid") traffic. Traffic also has phases that do not have analogues in common physical systems, such as backwards-flowing waves of stalled traffic mixed with moving traffic.

If a system has more than one phase, it will have boundaries between phases. Varying a control parameter (such as temperature moving water from ice to liquid) can generate a *phase transition*. In purely physical systems, control parameters are usually external, though in engineered or biological systems they can be internal and adaptive. The set of phenomena around phase transitions are called *critical phenomena*, and include the divergence of the correlation length, ergodicity breaking (not all possible states of the system reachable from a given configuration), and other phenomena. The divergence of the correlation length is of particular interest in traffic systems because it means that a perturbation in one part of a system can affect another part at a large distance, with implications for controlling methodologies.

Just as molecules obey certain laws (conservation of energy and momentum and the equipartition of energy), the traffic "molecules" (agents representing vehicles with drivers) obey simple laws implemented in a fully distributed fashion – attempting to get where they are going as quickly as possible (with an upper limit) and interacting with other vehicles, such as avoiding collisions and following at a safe distance. Even though systems of self-propelled entities do not obey the same conservation laws as traditional equilibrium statistical systems do, many of the traffic physics systems that have been recently proposed have mappings onto well-studied equilibrium systems.

2.1.2 Traffic Physics and Phase Behavior

An example of phase behavior is the highly simplified collective motion model of Vicsek et. al. [5], inspired by the computer graphics work of Reynolds [6]. Their model consists of a collection of entities all traveling at the same invariant speed in two dimensions but whose headings are allowed to vary. At each update cycle of the model, the directions of the particles are updated by the following rule: The direction is updated by taking the average of the directions of the neighboring particles in a radius r and adding a noise term. $v_i(t+1) = \langle v(t) \rangle_r + \theta_i$. The end result is a textbook phase transition as depicted below in Figure 2-1.

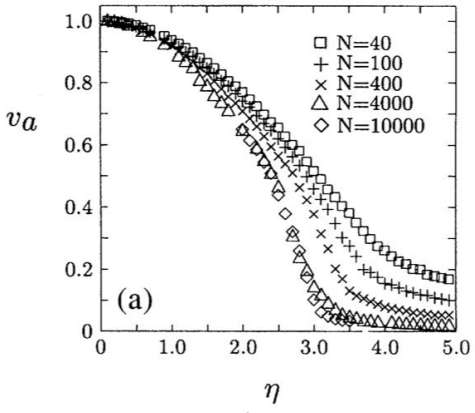

Figure 2-1. Phase Transitions and Noise.

The y-axis denotes average alignment of particles, the x-axis denotes noise.

At low noise values (η), the entire system tends to align. As noise increases, uncorrelated motion results. As the system size becomes larger (the multiple curves shown), the curves asymptote to a single curve, another classic indicator of phase transition behavior. If one approaches the phase boundary from the high-noise side (large values of η), then there is a

sudden emergence of preferred direction in the model; this is the phase transition boundary. As the system size approaches infinity, the onset of preferred direction becomes infinitely sharp.

A somewhat more realistic model than the previous one has been developed by Helbing and others [3] and corroborated with simulation and empirical data. In vehicle traffic, throughput (or capacity) of a roadway increases with density to a certain point after which a marked decrease is observed; hence, the emergence of a traffic jam. In this model the driving parameter is vehicle density per length of roadway, not noise. The two are related: The higher the density the greater the frequency of correcting behavior (speeding up, slowing down). Each incidence of correcting behavior is associated with uncertainty (noise). Instead of the noise being applied externally, it is endogenously generated by adaptive agent behavior. When density is low, overshoots and undershoots do not propagate very far because of the "slack" in the system.

At a certain critical point, these perturbations ricochet throughout the system, generating a cascade of corrections and pushing the system into a radically different configuration (the "traffic jam" phase). The noise generated with each speed correction creates an equal or greater number of other speed corrections, and the system cannot stably return to the initial configuration. This dynamic generates a phase transition. Figure 2-2 shows the results:

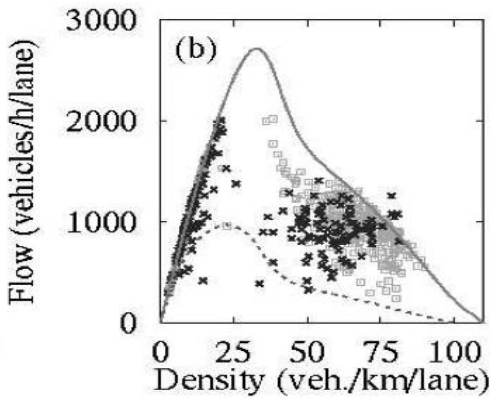

Figure 2-2. Freeway Traffic Phase Diagram [7].

Plot showing theory, simulation, and observation for freeway traffic. Dotted line represents theoretical prediction for pure truck traffic, solid line pure automobile traffic. Black crosses show simulation results for mixed traffic; grey boxes show actual freeway measurements.

2.1.3 Traffic Physics and Airspace

Extending the traffic physics paradigm to the airspace problem requires some modifications and extensions to the current models in the literature. For the most part, aircraft have *intent*, and this fact needs to be reflected in any realistic model of the airspace. The Helbing model discussed above effectively incorporates intent, as the particles are constrained to move in one dimension, and it is assumed they wouldn't be there and going in that direction unless they wished to do so..

The Vicsek model, though it has similarities to flight models, does not incorporate intent because there is no preferred direction of motion. Because of iterated directional corrections and the influence of noise, the initial direction of a particle may change by a large amount over time, and there is no notion of the initial (or any *a* priori) direction being "preferred" or "optimal," though the model spontaneously generates preferred direction under the right parameter settings. One of the principal challenges of our research has been to incorporate intent in a natural and computationally efficient way. We believe we have done this, with help from yet another subfield of physics (string theory) . Our approach is discussed in more detail in Section 3.

2.2 Bulk Properties of Airspace and Computation

2.2.1 Hard Problems

Kirkpatrick and Selman [10] have shown that all computationally NP-hard problems (such as the generalized deconfliction problem involving N aircraft [7, 8]) can be reduced to a construct known as **3SAT**, short for "satisfiability" [9], which displays a standard form:

$$(x_{11} \dot{\cup} x_{12} \dot{\cup} x_{13}) \dot{\cup} (x_{21} \dot{\cup} x_{22} \dot{\cup} x_{23}) \dot{\cup} (x_{31} \dot{\cup} x_{32} \dot{\cup} x_{33}) \dot{\cup} \ldots \quad (1)$$

where $\wedge$ represents Boolean "And," $\vee$ represents Boolean "Or," and the x_{ij} are Boolean variables taking on the values {True, False}, isomorphic to {1,0}. A statement is said to be "satisfied" if the truth value of the entire expression is "True."

An example of such a problem involving a mapping to trajectory deconfliction is the following: Imagine an "airspace" with only two aircraft, A and B, on a head-on collision course. Each aircraft has a set of actions available to it: *turn left, turn right, climb, descend, do nothing*. This set can be encoded in shorthand as an "alphabet" of behaviors: {*L,R,C,D,N*} A separation violation is only averted if both aircraft act symmetrically from the ownship point of view in the horizontal plane, or if one aircraft descends and the other does anything else but descend. Satisfiable airspace for this highly simplified two aircraft system is therefore:

$$\boldsymbol{S} = (A_L \wedge B_L) \vee (A_R \wedge B_R) \vee ((A_C \vee A_L \vee A_R \vee A_N) \wedge B_D) \vee (A_D \wedge (B_C \vee B_L \vee B_R \vee B_N))$$

One can see that twelve of the possible 25 collective behaviors (an alphabet of five squared because of two aircraft) result in a satisfied airspace. This simplified airspace resolution problem statement can then be transformed into a substantially longer expression in standard form above. Solving general 3SAT problems requires an amount of time exponential or greater in the problem size, which in the deconfliction problem would be the number of aircraft impinging on a particular volume of space and time. This solution procedure rapidly becomes impractical in full generality for congested airspace without incorporating information (such as intent or optimality measures) that constrain the space of possible solutions and order the deconfliction process.

2.2.2 Satisfiability

In addition to finding a good solution, we wish to characterize and exploit the entire *space* of solutions. This characterization is required by the 5DT nature of our trajectories: Not only must the current solution suffice, but it must be able to deform smoothly to nearby good solutions as the system evolves. The existence of other system solutions has no benefit unless they can be reached smoothly (without violating any constraints) from a given solution. In recent years, this

concept of a good solution space has been formalized, and like traffic physics, it maps onto the physics of phase transitions. Statistical ensembles of random 3SAT problems display a phase transition between soluble and insoluble that maps onto typical physical phase transitions [10].

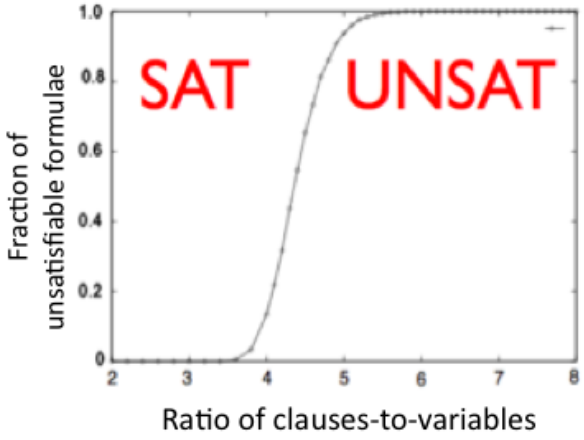

Figure 2-3. Satisfiability Phase Transition [10].

Figure 2-3 shows the probability of satisfying a randomly chosen 3SAT expression for different ratios of clauses to variables (c/v). As the ratio increases, many variables appear in multiple clauses, effecting constraints. Eventually most expressions are unsatisfiable when $c/v \geq 8\ ln2$.

The driving parameter for the phase transition in a logical system is the density of constraints – the *x*-axis in Figure 2-3. In an airspace environment, increasing density of aircraft means more constraints on the motion of individual aircraft. At high enough densities no viable solution can be found for the system as a whole. Naturally one would desire deconfliction problems to be certifiably on the satisfiable side of the Satisfiability Phase Transition graph, or failing that, have a prescribed mechanism for mitigating unsatisfiable configurations in advance of any separation violations.

2.2.3 Solution Spaces, Robustness, and Flexibility

The logical and physical realizations of phase transitions are formally connected in the work of Kirkpatrick and Selman [10], but can also be intuitively connected if one takes the agent's point of view: If a molecule (or car or aircraft) has no options as to where to go next in space, then the system freezes up, or the traffic jams, or the system is "full." Other attempts to address this question have used dynamic programming [1], vector field divergence [11], Lyapunov exponents [7], genetic algorithms [12], ray-tracing with diffraction [13], and others. Each one of these studies addresses the satisfiability question of airspace fullness from a different perspective, and like NP-hard problems in general, should in principle be unifiable under the rubric of satisfiability.

We chose our formalism not because it offers anything different from a computational complexity point of view, but because it unifies the concepts of safety and efficiency in a concise manner, and it is tailored to solving this particular NP-hard problem. The utilization of a trajectory-based formalism makes the problem easier, not harder, as it constrains and orders the solution search space by providing a smooth cost function, namely the flight time and fuel burn associated with a

given trajectory configuration. In addition, our approach lends itself to parallel computation, which makes the problem significantly easier from an engineering implementation perspective.

Early satisfiability research addressed the existence of a phase transition, but did not address the nature of solutions near the phase transition. Later research identified that it becomes increasingly difficult near the boundary to either find a solution or prove that one does not exist. This finding has practical implications for airspace science. Knowing that a solution probably exists is of no utility unless the solution can actually be found in time for the airspace participants to make use of the information.

The statistical nature of the phase transition identified in our experiments has a unique benefit that is not available using conventional deterministic conflict resolution methods. By localizing *regions of airspace-time* that are approaching a phase transition, based on the computational difficulty associated with finding a satisfactory solution, we were able to predict the existence of conflicts within those regions much further in advance of conflicts than may be possible using conventional deterministic methods. Though it may not be possible to predict a particular conflict using conventional deterministic methods more than approximately 20 minutes in advance of the predicted conflict (primarily due to uncertainties in the wind field), by localizing regions of airspace using our statistical methods, we were able to look farther ahead in our experiments. It is a separate claim to state that a region of airspace is likely to be rich in conflicts than to say two particular aircraft will have a conflict. Though the airspace region claim is less specific than a deterministic prediction, it still provides useful information, and our simulations have shown that this computational difficulty property is more persistent than individual conflicts. Therefore it holds the promise of long-range intervention with air traffic controls much further in advance of potential conflicts than the current 20-minute conflict detection capability using conventional deterministic methods. The computational difficulty property in an airspace region is shown below in Figure 2-4.

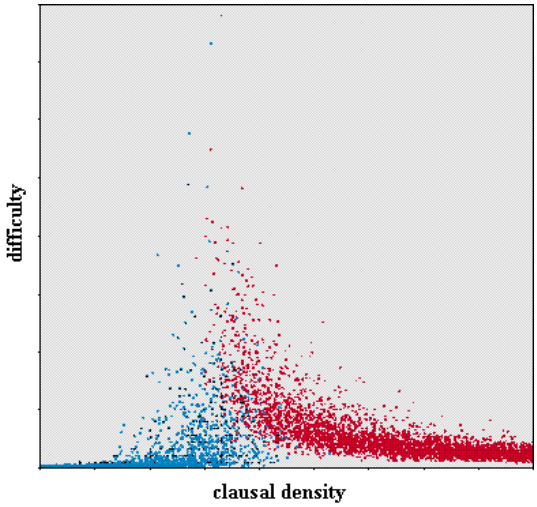

Figure 2-4. Snell Graph

The *x*-axis is proportional to the density of constraints on the system (c/v in Figure 2-3). The blue points indicate randomly generated satisfied expressions, the red dots indicate solutions shown to be unsatisfiable. The *y*-axis is proportional to the time it took to compute either case. The center of the phase transition (see Figure 2-3) is where the number of blue and red dots are equal.

Far to the left and far to the right of this point, it is easy either to find a solution or to show that it is impossible to find one. In an engineered system, not only does one want to avoid the phase transition, one wants to stay far enough away to be able to compute solutions readily.

Flexibility can be interpreted as the presence of many solutions to a problem, and flexibility has been interpreted in the aeronautical literature as the set of solutions reachable from a given airspace configuration in the presence of disturbance [1]. The detailed analysis of general SAT systems provides compelling insight and analytical rigor [8,14] for these abstract concepts. See Figure 2-5 below:

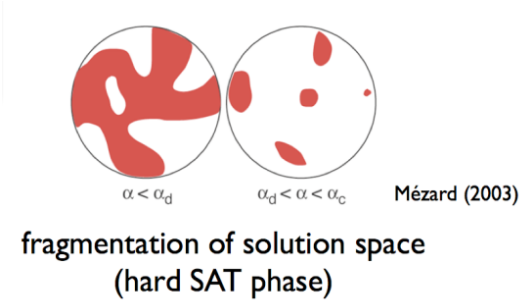

Figure 2-5. Solution Space of SAT Problems – Near Phase Transition [14]

> As the satisfiability phase transition is approached from the left, the nature of the solution space begins to change. Below a well-defined c/v value (less than and different from the phase transition value) the solution space shows flexibility in the form of many connected solutions. This flexibility disappears above that value, even though satisfiability remains.

In Figure 2-5, the solution space for a generalized 3SAT problem transforms from connected to disconnected as the phase transition boundary is approached. The near-phase-transition phenomenon is also appealing because it means that there is general evidence of an advance warning of the onset of a phase transition, something extremely useful in systems where humans might intervene to avoid undesirable dynamics, such as air traffic management systems.

To summarize the reasoning of this chapter: The generalized airspace deconfliction problem meets the definition of a computationally complex problem. Computationally complex problems of this class (NP-hard) have been shown to display phase transition behavior. Phase transition behavior has, in turn, been shown to display precursory evidence to the less constrained side of the phase transition graph (Figure 2-3). We wish to discover evidence of the approach to a phase transition by modeling and simulation and eventually use the evidence in our explanation of airspace dynamics.

3. APPROACH

3.1 Idealized Automated Airspace and Trajectories

In the interest of investigating general phase transition structure in airspaces, we have shed many of the variables in our test airspace which characterize actual real-world airspaces. In particular, we have idealized the airspace to a simple cylinder, focused exclusively on enroute trajectories, and simplified aircraft performance to specified limits on speeds and accelerations as set forth in detail in Sections 3.2, 3.3 and 5.1 below.

In addition, we have endowed our dynamical trajectories with agency, acting in concert to deform themselves according to separation and performance requirements, with no human intervention or guidance. For research purposes, our system as simulated is fully automated, thus enabling the simulation and analysis of multiple airspaces, driven exclusively from a well-defined set of parameters. We have used the guidance of previous research in traffic physics to create a model of airspace dynamics with detail sufficient to generate complex emergent behavior, but have avoided any additional model fidelity at this point.

3.2 Formal Problem Statement

We represent the continuous airspace replanning and deconfliction problem as follows:

Definitions:

1. 5DT Trajectory Definition

A **trajectory** $\mathbf{T}(\mathbb{x}(t,\tau);t,\tau), \mathbb{x} \in \mathbb{R}^3$ is a continuous one-dimensional curve of finite length embedded in five-dimensional space-time characterized by three spatial dimensions and two time dimensions $T:(\mathbb{R}^3 \otimes \mathbb{T} \otimes \mathbb{T}) \rightarrow \mathbb{R}$. Position along a trajectory is parametrized by t and the current state of all trajectories (see Def. 2) is parametrized by τ. Because of the extra time parameter associated with the current state of the system, these are known as "5DT" trajectories.

2. Airspace Definition

An **airspace** $\mathbb{A}$ is a set of $N(\tau)$ trajectories $\{\mathbf{T_i}(\mathbb{x}(t,\tau);t,\tau), i = 1, \ldots N(\tau), \mathbb{x} \in \mathbb{R}^3\}$ embedded in five-dimensional space-time $(\mathbb{R}^3 \otimes \mathbb{T} \otimes \mathbb{T})$ where t parametrizes position along each trajectory $\mathbf{T_i}$ and τ is system ("global") time. This time is also referred to as "Meta Time" in the code appendices.

3. 5DT Time Relations Definitions

$$t,\tau: t < \tau \textit{ is } \text{"past"}, t = \tau \text{ is "present"}, t > \tau \textit{ is } \text{"future"}$$

4. Aircraft Position Definition

$t_i = \tau$ defines nominal **position** of aircraft i along trajectory $\mathbf{T_i}(\mathbb{x}(t,\tau))$

5. Finite-Range Pseudopotential between Trajectory Elements $d\boldsymbol{T}_i$:

$$\phi(\mathrm{d}\mathbf{T}_1, \mathrm{d}\mathbf{T}_2, t, \tau) = \begin{Bmatrix} 0 \text{ if } \mathrm{D}(\mathrm{dT}_1, \mathrm{dT}_2) > \mathrm{d_c} \\ \mathrm{A}\left(\mathrm{d_c} - \mathrm{D}(\mathrm{d}\mathbf{T}_1, \mathrm{d}\mathbf{T}_2)\right)^{\alpha} \text{ otherwise} \end{Bmatrix},$$

Where D = distance between trajectory elements

Problem Statement:

Minimize total path length $\mathbb{L}$ of all trajectories for each τ: $[\tau_i, \tau_f]$

$$\mathbb{L}_{min}(\tau) = \mathrm{Min}\left\{\sum_{\mathrm{i}} \int \left\| \frac{\partial \mathbf{T}_{\mathrm{i}}(\mathbb{x}(\mathrm{t}, \tau))}{\partial \mathrm{t}} \right\| \mathrm{dt}\right\}$$

subject to the following constraints:

Constraints:

1. 4DT Fixed Endpoints of 5DT Trajectories (Endpoints and Flight Duration fixed):

$$\mathbf{T}_{\mathrm{i}}(\mathrm{t} = \tau_{\mathrm{init}}) = \{\mathbb{x}, \tau\}_{\mathrm{init}}, \mathbf{T}_{\mathrm{i}}(\mathrm{t} = \tau_{\mathrm{final}}) = \{\mathbb{x}, \tau\}_{\mathrm{final}}, \mathbb{x} \in \mathbb{R}^3$$

2. Continuous Deconflicted Airspace State Requirement

If $\left|\mathbf{T_i}(\mathrm{z}(\mathrm{t}, \tau)) - \mathbf{T}_{\mathrm{j}}(\mathrm{z}(\mathrm{t}, \tau))\right| < vsep$, $\left\|\overrightarrow{\mathbf{T}_{\mathrm{i}}}(\mathrm{x}(\mathrm{t}, \tau), \mathrm{y}(\mathrm{t}, \tau)) - \overrightarrow{\mathbf{T}_{\mathrm{j}}}(\mathrm{x}(\mathrm{t}, \tau), \mathrm{y}(\mathrm{t}, \tau))\right\| > hsep$ for all $\mathrm{i} \neq \mathrm{j}$ and all t, τ such that $\mathrm{t_i} = \mathrm{t_j}$. z is the vertical coordinate of trajectory coordinate $\mathbf{T}(\mathbb{x}(\mathrm{t}, \tau))$, x and y are the horizontal coordinates of $\mathbf{T}(\mathbb{x}(\mathrm{t}, \tau))$. The airspace exists in a deconflicted state as well as a planned deconflicted state at all system times τ. This separation specification is a statement of the typical separation criterion.

3. Bounded Speed and Acceleration along $\boldsymbol{T}_i$

$$\mathrm{vmin} < \left\| \frac{\partial \mathbf{T}_{\mathrm{i}}(\mathbb{x}(\mathrm{t}, \tau))}{\partial \mathrm{t}} \right\| < vmax \; for \; all \; t, i$$

$$\left\| \frac{\partial^2 \mathbf{T}_{\mathrm{i}}(\mathbb{x}(\mathrm{t}, \tau))}{\partial \mathrm{t}^2} \right\| < amax \; for \; all \; t, i$$

4. Constants: {vsep, hsep, vmin, vmax, amax, A, d_c, α} are all user specified constants

Assumptions:

1. Planning: The Evolution of Trajectories:

a. As global time τ increases, $N(\tau)$ changes as trajectories enter or leave the airspace system because of initiation or termination.
b. As τ increases, the parts of trajectories characterized by $\mathrm{t} < \tau$ become "past" and can no longer change.
c. The parts of trajectories characterized by $\mathrm{t} > \tau$ are "future" and are subject to continuous replanning until they become "past."

2. Acceleration

Acceleration bounds are only considered along the trajectory, perpendicular forces are not considered explicitly.

3. Test Airspace

a. The test airspace is a circular region of diameter 1000 km.
b. Only the cruise portion of flight is modeled and simulated.

Instantiation of Optimization Problem

1. Trajectories are approximated by a set of cubic splines T_i: $T_i \cong \{S_{i,j}(\mathbb{x}, \tau, t_j^{init}, t_j^{final}), j = 1, \ldots m)$ where each spline is defined over a time interval $[t_j^{init}, t_j^{final}]$ such that the union of the time intervals describes the entire trajectory and the intersection of the splines is a set of control points.
    a. Positions and velocities are matched at each intersection of splines, accelerations are discontinuous at intersections and functions of form $at + b$ otherwise.
    b. Positions and velocities are independent variables at each spline intersection point, accelerations are dependent variables.
2. Path integrals over the length of each trajectory are replaced by cost functions of the form

$$C = \sum_{i=1}^{N} \sum_{j=1}^{m-1} \left| a(S_{i,j}) - a(S_{i,j+1}) \right|$$

where the *a's* are accelerations along the trajectory as defined in **Constraints**.3. This approach minimizes a discrete form of the first derivative of acceleration, also known as "jerk." A cost function of this form is amenable to a local "smoothing" procedure that is simple and rapid to implement and is incorporated below in the conflictadapt procedure.

3. The pseudocode shown below is specific to the cubic spline instantiation of the trajectory deconfliction/optimization problem.

Pseudocode

```
procedure trajectory optimization/deconfliction()
begin
    initialize system time: τ ← τ_init
    initialize airspace 𝔸 with N(τ_init) trajectories
    repeat
        initialize trajectory time t ← τ
        repeat
            for all i, j: i > j
                if conflictdetect(T_i, T_j, τ) == False,
                    then next (i, j)
                else if conflictdetect(T_i, T_j, τ) == True
                    then conflictadapt(T_i, T_j, τ)
                    if conflictadapt(T_i, T_j, τ) == False
                        then
                            return adaptfailure( )
                            next (i,j)
                        else next (i,j)
                    end if
```

end if
end for
increment trajectory time $t \leftarrow t + \Delta t$
until ($t == t_{final}$)
increment system time $\tau \leftarrow \tau + \Delta\tau$
until ($\tau == \tau_{final}$)
end

procedure conflictdetect(T_i, T_j, τ)
begin
initialize current state of trajectories $T_i(t \leq \tau)$
compute time endpoint for trajectory pair $t_{max} = \mathrm{Min}(\tau_i^{final}, \tau_j^{final})$
initialize $t \leftarrow \tau$
repeat
if Distance($T_i(t), T_j(t)$) $\leq d_c$
return {distance, t}
end if
increment planned trajectory time $t \leftarrow t + \Delta t$
until ($t == t_{max}$)
end

procedure conflictadapt(T_i, T_j, τ)
begin
compute vector between desired and current closest spatial approach $\vec{w}_j^D((t,\tau))$
compute vector between desired and current velocity: $\vec{v}_j^D((t,\tau))$
initialize adjustmentcycle = 0;
initialize adjust() = FALSE
while (adjustmentcycle ≤ max || adjust() !=TRUE) **do**
begin
compute exponential damping factor

$$f = \frac{e^{-\mathrm{adjustmentcycle}}}{\sum_1^{\max} e^{-\mathrm{adjustmentcycle}}}$$

increment trajectory closest spatial approach by $f\vec{w}_j^D((t,\tau))$
increment velocity at closest approach by $f\vec{v}_j^D((t,\tau))$
adjust trajectory velocity and position with smoothing vector

$$\vec{s}(f\vec{w}_j^D((t,\tau)), f\vec{v}_j^D((t,\tau)))$$

if (accelconstraintsatisfy == TRUE &&
velocityconstraintsatisfy == TRUE &&
separationdistancesatisfy ==TRUE)
then adjust() = TRUE
end
if adjustmentcycle == max && adjust()==FALSE
return adaptfailure()
end

3.3 Details of our Approach

3.3.1 Trajectory Generation and Deconfliction

Typical optimal long-range vertical profiles for commercial jet transport aircraft consist of optimal ascent and descent segments connected by a long cruise-climb or step-climb segment. Optimal horizontal routes are not as easy to compute because the variations in the wind field lead to a non-convex nonlinear optimization problem with potentially many regions of local minima. As a result, approximate optimization solution approaches must often be considered even before the added complexity of deconfliction is factored in.

A variety of different heuristics have been applied in the literature, including virtual wind fields that "blow the aircraft out of each other's way" [15], genetic algorithms acting on a discrete decision space [12, 16], dynamic programming [1, 17], path-planning by analogy with optics using refractive indices combined with pseudopotential methods to reduce the search space size [13], and others.

In order to generate *dynamic* optimization (continuous replanning) and deconfliction of thousands of trajectories and observe realistic emergent collective phenomena, a number of algorithmic accelerations were employed. We utilized scalable heuristics based on pseudopotential methods to achieve rapid systemic deconfliction. To incorporate intent and optimize path dependent measures such as time and fuel burn, we borrow a concept from theoretical particle physics, the notion of an ensemble of interacting extended objects ("strings"). We identify these extended objects with candidate 4D aircraft trajectories, depicted in Figure 3-1. Trajectories as represented by one-dimensional extended objects are endowed with a distributed pseudopotential so that they repel each other, an extension of traditional pseudopotential methods where point objects representing aircraft themselves repel each other, and the charge is sufficient such that required separation is maintained.

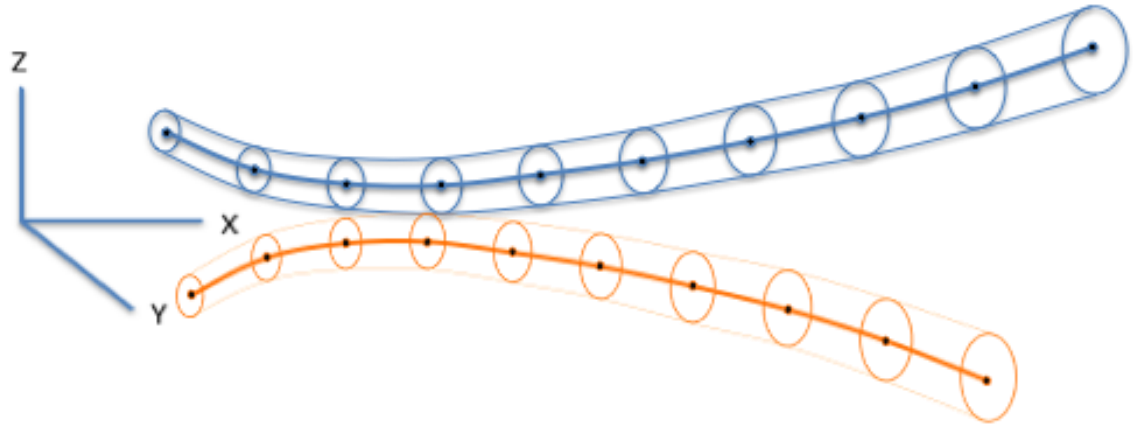

Fig. 3-1. Charged string concept.

Two aircraft trajectories endowed with repulsive pseudopotentials. The circles represent time slices in the predicted future. Separation is maintained by the pseudopotential deforming the strings, which distribute the deformation along their length so as to reduce curvature to acceptable levels.

3.3.2 Replanning and Agency

Approximately once per second in the simulation, all known trajectories are recalculated; thus, we assume there is implicit renegotiation. Therefore, all trajectories are "closed" in the sense of always having a current flight plan, and we do not need to make a distinction between tactical and strategic planning.

The agents in the simulation are the trajectories, not the individual aircraft. The cockpit avionics and pilot, the ANSP (ground based automation, controllers, flow control specialists), and the user operations control centers are not currently modeled. Because we do not model these elements of the system (and their interactions), we take no position about how and where trajectory negotiation, trajectory control, and aircraft separation occur. This position applies also to the specific structure and details of communication required to effect the dynamics of the model in a real system.

Our trajectory paths in the model are fully dynamical, meaning the paths are dynamically recomputed (replanned) throughout the flight at regular intervals, while continuously maintaining deconfliction. This concept is what we mean by "trajectory agency." Since the trajectory description subsumes the aircraft's position, additional information is incorporated compared to an aircraft-centric approach. The replanning interval, or "heartbeat" of the simulation, is a constant that is chosen to be small compared to the characteristic time scale of changing external conditions-in our case, moving weather and other external inputs (such as an airport being closed) and the consequent changing trajectories of other aircraft. This time relationship would correspond to a time interval of about a minute in the real world.

3.3.3 Assumptions and Caveats

No safety net such as TCAS (Traffic Collision Alerting System) is included. We assume that the simulation will handle all potential conflicts by proper trajectory planning and de-confliction in our test airspace. This assumption is a gross simplification for the purpose of our experiments.

No current airspace architectural components are included. Our simulation optimized all the trajectories for the entire test airspace without any administrative divisions.

In our simulations we used a rudimentary hierarchy of objective functions: First, trajectories must avoid each other. Given separation, trajectory curvature and acceleration along the trajectory are minimized. This generates a trajectory of a certain length. Since flight duration is specified in advance to be a constant, path length is related to average aircraft speed. A solution is acceptable if flight speed remains within a defined band after all of the above deformations..

3.3.4 Test Airspace

The following approach taken in the construction and operation of our experiments in the test airspace:

- Generic Airspace Cylinder: For purposes of developing the algorithms, applying them to the airspace phase state concept, exploring the parameters affecting phase state, and evaluating phase state control, a generic airspace cylinder of 1,000 km in diameter and 10,000 feet in height was employed. The test airspace was borrowed from previous

studies by NASA [Consiglio][18]. Weather: Weather objects have dynamical behavior, with uncertainty created by randomness in its path.

- Enroute trajectories: The climb and descent are computed for entry into the generic airspace cylinder, but not included in trajectory separation calculations.
- Dynamic replanning: Trajectories are dynamically replanned over their full length, from top-of-climb to top-of-descent.
- Proxy representation of aircraft performance and related trajectory control and maneuvering limitations: Maneuvering is allowed within "policy" limits derived from representative airline policies for maximum bank angles, accelerations, and pitch changes.

3.3.5 Data Collection

Most centrally to our research, our model kept track of the "amount" of computation required during the dynamical replanning process (per time instant and geographical location). We used the compute cycles required in the trajectory replanning process as an indicator of a possible phase transition as the airspace "heated up" as the density of the airspace utilization (trajectories) increased. This phenomenon has been observed in other super dense simulations of dynamic deconfliction [Erzberger][19]. Such data may aid in estimating the overall capacity of the airspace.

Another feature of this approach to modeling dynamical trajectories is the ability to generate data that are a measure of the correlations of nearby flight trajectories. If found, these data may indicate "flocking" [6] or other emergent activity in certain phases of the airspace phase space, similar to the emergent directional correlation discussed in the Vicsek model in Chapter 2 [Vicsek][5]. We believe this behavior should remain as simulations become more realistic and agents more complex.

4. TOOLBOX

4.1 Introduction

Our work on this project included preparation of an initial toolbox of algorithms, agent-based structures and method descriptions for introducing agency as a methodology for analyzing and managing the complexity of airspaces states while maintaining or increasing system safety. Appendix A contains the complete toolbox. Each of the algorithms is described conceptually and explicated in technical detail in the form of *pseudocode*. The pseudocode is intended to contain sufficient technical detail to enable implementation in a language of choice on the user's hardware platform. In this Chapter, we provide an overview of the central ideas contained in the toolbox algorithms, and leave the finer details to Appendix A.

4.1.1 Airspaces

The toolbox was prepared for the experiments we conducted on cylindrical enroute test airspaces. Aircraft enter and exit the test airspace at the perimeter of the cylinder in such a way that all their trajectories pass through the central region of the cylinder.

The test airspace provided the platform for generating trajectories that were separated and flyable, if possible. We characterized an airspace as "successful" if all trajectories were separated and flyable. If any of the trajectories violated minimum separation distances, or was not flyable, we characterized the airspace as a "failed" airspace. A flyable trajectory was defined as one where all the points along the trajectory lie within some specified range of speeds and accelerations of the aircraft. This definition is a proxy for the laws of physics, aircraft specifications, and airline policies.

4.1.2 Trajectories

Conceptually, dynamical trajectories are abstractions spanning both space and time. Hence, trajectories are 4DT, i.e. three dimensions of location and one time dimension. However, due to the exigencies of airspace, trajectories may need to be replanned dynamically after a flight has begun. In our algorithms, at every delta *t* time increment, all the trajectories were replanned (re-calculated) according to current conditions. Trajectories managed by these algorithms were seen to sometimes vary a large amount in space and time, depending on the density of other trajectories and external influences from simulated weather.

We considered every 4DT Trajectory itself a dynamical entity, replanned every delta *t*, which produced two types of time. There is the flight time embedded into every instance of a trajectory. Every trajectory also changes itself over time so, as indicated in the Introduction, we used an additional Meta Time variable (also called "system time"), which gave each trajectory 5 dimensions (3 dimensional location plus 2 time dimensions—Flight Time and Meta Time.

Intuitively, a single trajectory instance is like a hard strand of spaghetti lying still on a cold plate – whatever curve it has is statically fixed in place. A collection of dynamical (suite of changing) trajectories is like a soft strand of spaghetti curling, stretching, and moving away from other strands of spaghetti in a pot of boiling water. Over the course of its flight time, an aircraft might fly parts of many dynamically replanned trajectories. An actual flown flight path is, in effect,

pieced together from many instances of trajectories as the dynamical replanning process re-shapes the trajectory in Meta Time, responding to maintain separation or avoid weather.

The concept of 5DT is illustrated in Figure 4-1 where a trajectory itself is modified. The future of any particular trajectory has a Flight Time associated with it. In addition, trajectories are modified at some time *t* in Meta Time as well.

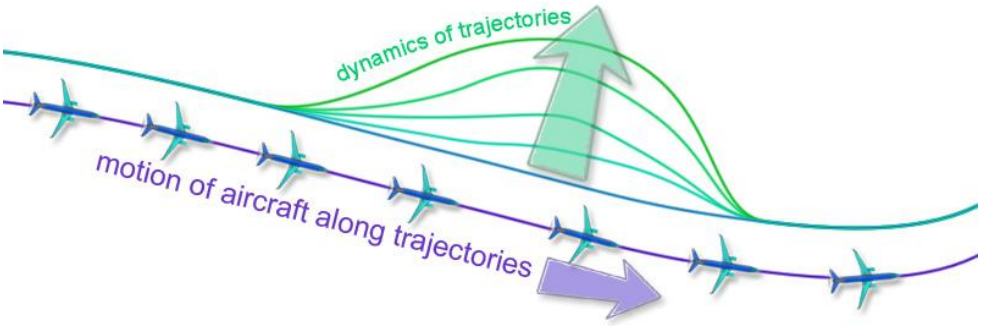

Figure 4-1. Trajectory dynamics.

4.2 Deforming Trajectories

Trajectories would remain unchanged if there were no pressures to change their paths. In a sparse airspace, initial trajectories can be quite stable with no need to change an optimal trajectory. However, in more dense airspaces, separation may force changes in paths, which typically involved lengthening the paths to go around some obstacle. On the other hand, economic pressures forced the path to be more evenly curved to save fuel or fly more smoothly. In addition, physical limits on velocity and acceleration tended to force the trajectory into more flyable shapes as well. The shortest possible path may not be flyable. In principle, our algorithms searched for the shortest flyable de-conflicted paths.

These practical requirements for trajectories can be conceptualized and implemented as a deforming influence generated by a pseudoforce (to implement separation) interacting with influences that maintain physically realistic ("flyable") trajectories thus simplifying the problem, as well as simplifying the algorithms used to deform the trajectories.

4.2.1 Three Influences on Trajectories

Our Toolbox of algorithms supports three types of influences that act to deform trajectories and implement the constraints:

- **Repulsion** – is a pseudoforce to maintain minimum separation.
- **Elasticity** – keeps trajectories gently curved to minimize distance, fuel consumption, etc.
- **Bounding** – keeps trajectory velocities and accelerations within physical and policy limits

These influences are described more formally as part of the pseudocode procedure conflictadapt(T_i, T_j, τ) in Section 3 and in more detail in the Appendices.

4.2.2 Repulsion

Rather than doing conflict detection and resolution per se, our trajectory strings or tubes were designed to repel each other in a manner that always maintains required separation. This method of separation was possible because entire trajectories were separated (throughout their entire length), as opposed to separating individual aircraft. In effect, no surprises were postponed into the future, unless new conditions arise, for example, changing weather conditions. Even then, entire trajectories were again immediately and fully separated through the operation of repulsion.

New conflicts may arise for a trajectory resulting from de-conflicting some other pair of trajectories. In addition, weather cells may move between one re-calculation cycle and another, generating new conflicts with the storm, reverberating to new conflicts between other previously deconflicted pairs of trajectories.

4.2.3 Elasticity

The application of a separation pseudoforce alone was insufficient to generate stable trajectories. Such paths were under-specified, causing instability of path locations, or "Brownian Motion," as paths remain unresolved. In these algorithms, we also applied an internal force of elasticity on each trajectory. This elasticity caused the trajectories to find shorter paths, conserving fuel, while still maintaining separation via the repulsive inter-trajectory force. This elasticity was implemented by a spline algorithm that minimized acceleration discontinuities along the trajectories.

4.2.4 Bounding

The third influence applied to implement a constraint, in addition to *repulsion* and *elasticity,* was *bounding*. This force was necessary to assure that the trajectories were flyable. In the simplified abstracted world of these algorithms and simulations, aircraft speeds were limited to a specified minimum-maximum range. Without this influence in the extreme case, aircraft could stop in the simulation and wait for other aircraft to pass by as a means of avoiding a conflict. Thus, bounding requires each aircraft to maintain a minimum speed in the test airspace.

Note that accelerations along each trajectory are limited as well, and this acceleration constraint is handled by the influence of *elasticity* on the trajectories as described in 4.2.3.

4.3 Visualizations

4.3.1 Successful Deconfliction and Resolution of Airspaces

We described above how the test airspaces were initialized, and then later described how to deform flyable trajectories to enforce minimum separation. Some airspaces were too congested to allow for successful deconfliction, while less congested airspaces had all of their trajectories successfully separated from each other.

The following Figures 4-2 and 4-3 illustrate the result of deforming the trajectories. In both of these cases, the deconfliction process was successful.

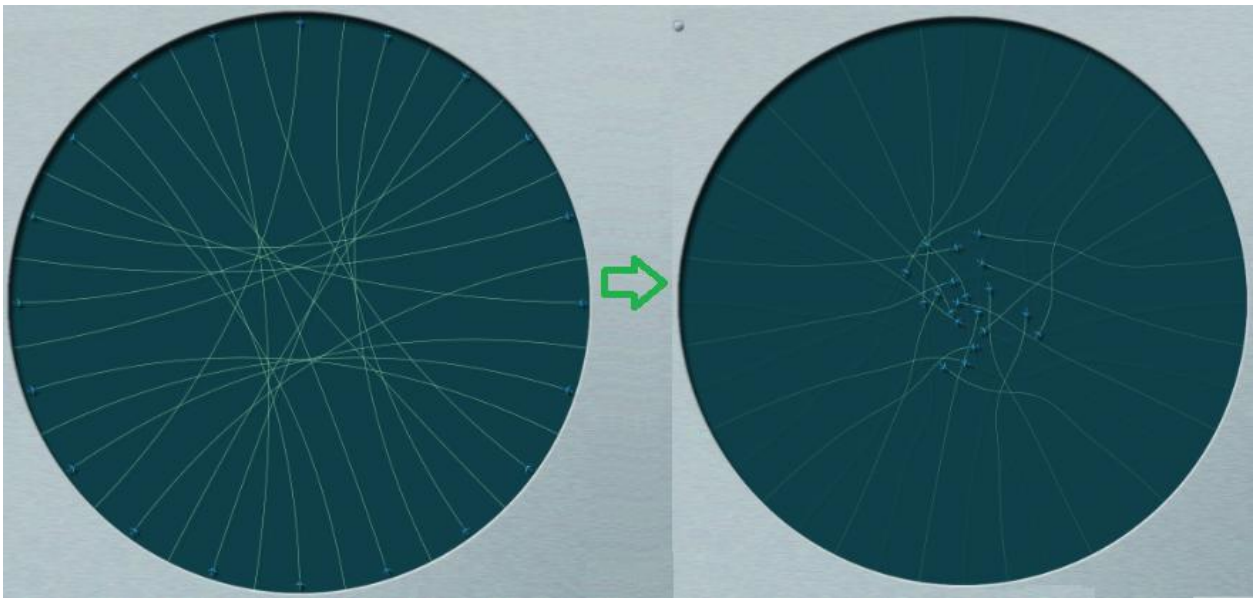

Figure 4-2. Deconflicting trajectories in an airspace – small number of trajectories.

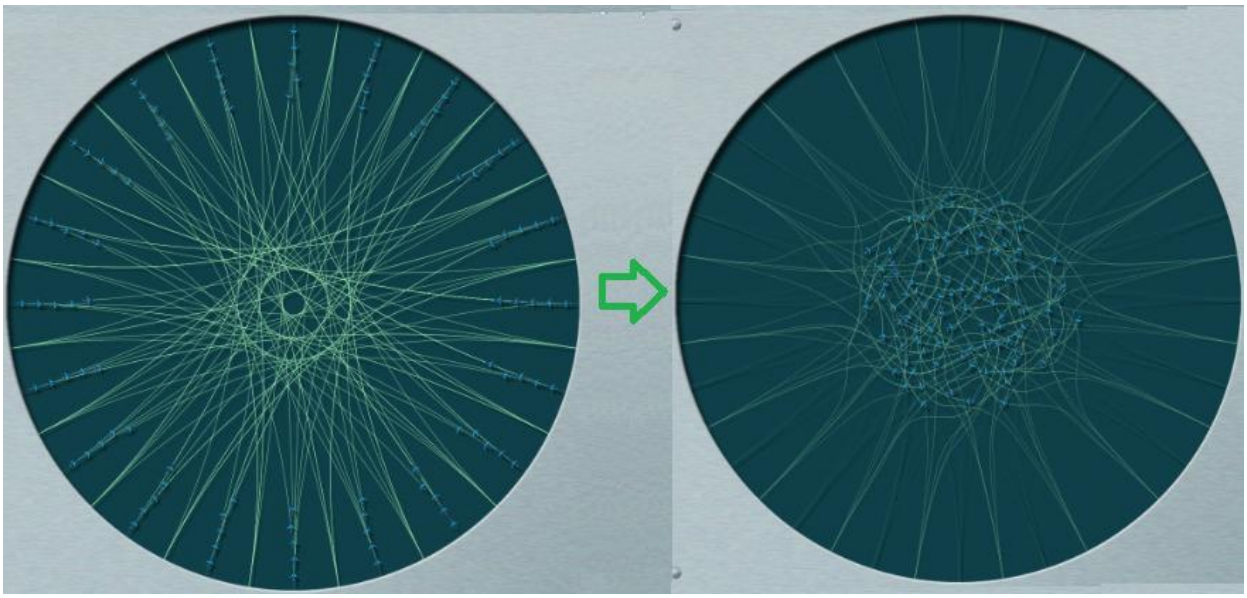

Figure 4-3. Deconflicting trajectories in an airspace – larger number of trajectories.

In the right hand screen shot in Figure 4-3, the detailed trajectories in the conjested center of the test airspace might be hard to see. Figure 4-4 shows the central portion of the airspace at a higher resolution, which makes the individual trajectories visible.

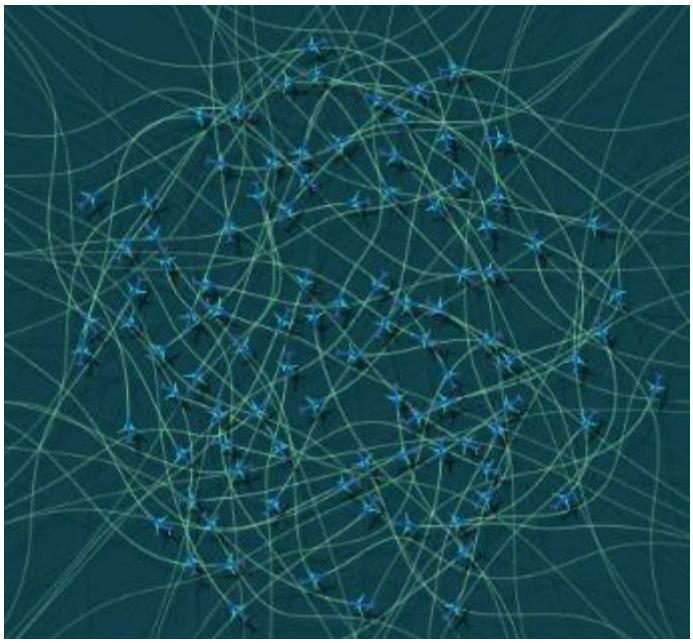

Figure 4-4. Close up of center of airspace shown in Figure 4.3 on right.

4.3.2 Failed Deconfliction and Resolution of Airspaces

The screen shots above show successful deconfliction and resolution of all the trajectories in these airspaces. In order to investigate the possible phase transitions at the limits of airspace capacity, it was necessary to generate conditions where airspaces could not be fully deconflicted.

When the density of the airspace became too great, resolving of some conflicts led to more new conflicts with other trajectories. Under these conditions, conflicts will persist in the test airspace, although not necessarily the same conflicts. Regardless of how many deformation cycles are executed in these conditions, the airspace will fail to converge to a solution.

Figure 4.5 shows two failed airspaces. Of the 200 trajectories (and aircraft) pictured here, in the left screen shot 55 of them are still in a state of conflict, shown by the portions of the trajectories drawn in red. In the screenshot on the right, there are 723 conflicts – more than one conflict per trajectory. The increase in conflicts was due to increasing the minimum separation from 4 nm to 5 nm.

Although additional processing resolved some of these conflicts, new ones appeared, keeping the airspace in a continued roiling unresolved state.

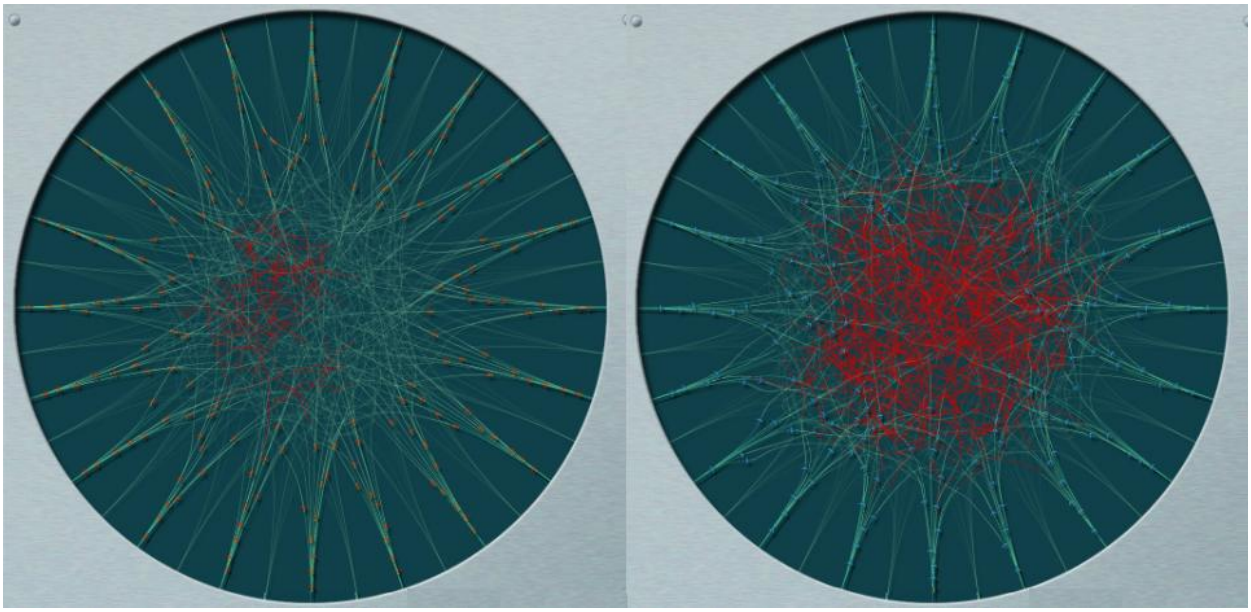

Figure 4-5. Screenshots of two failed airspaces.

At the resolution of the entire airspace, it can be difficult to see the fine structure of the trajectory conflicts. Figure 4-6 shows the central portion of both of the screenshots in Figure 4-5 at a higher resolution.

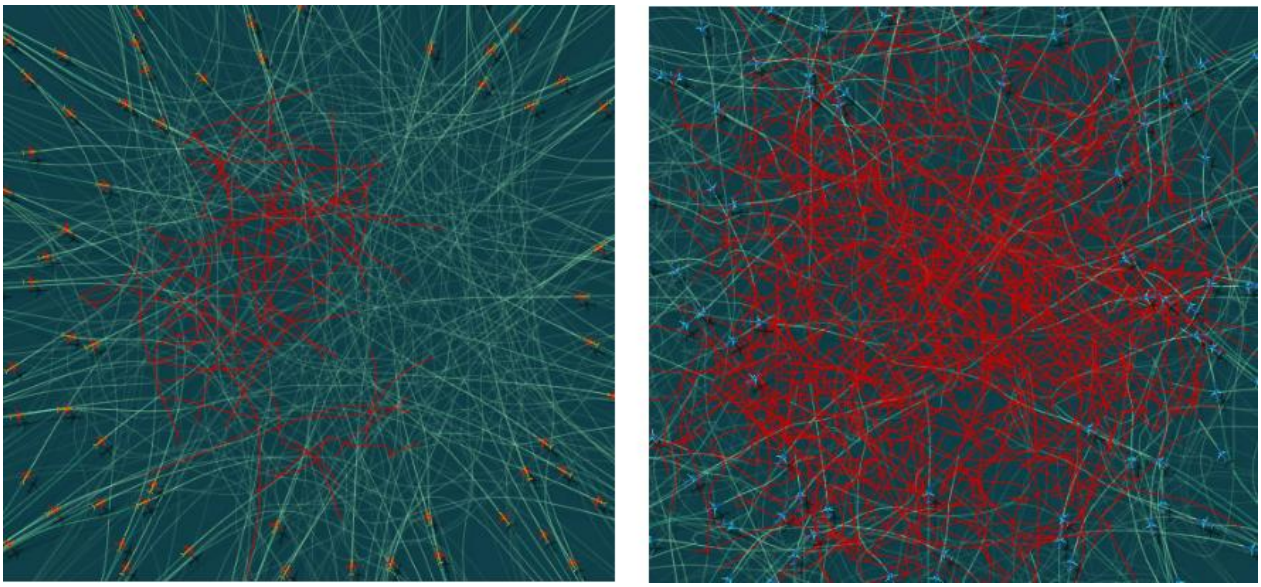

Figure 4-6. Higher resolution of screenshots of same failed airspaces.

5. EXPERIMENTS

This chapter outlines the experimental set up used to generate the data for analyses. Table 5-1 summarizes the experiments that were conducted.

Table 5-1. Experimental Run Sets

	Separation		
Dimensionality	5 nm	4 nm	3 nm
2D	X	X	X
3D	X	X	X

For each of the six experiments, we generated 1755 instances of airspaces at varying trajectory densities by altering the rate of incoming aircraft into the test airspace and the radius of the test airspace. In all 10,530 test airspaces were generated, each with a time series of the progress of dynamical trajectories up to 1000 cycles of deconfliction.

5.1 Experimental Parameters

The following four sections enumerate the parameters used in the algorithms described in Chapter 4. The values shown were used in the experiments conducted in our investigation of phase transition structure.

5.1.1 Aircraft and Airspace Parameters

1. (aircraft type) Boeing 757
2. vc = 530mph = cruising speed at cruising altitude
3. vmin,vmax = 450mph, 550mph = speed range at cruising altitude
4. zc = 30,000 feet = cruising altitude
5. zmax = 42,000 feet = ceiling
6. storm.rsep = 20nm = storm/aircraft separation
7. n_endpoints = 24 = number of entry points in test airspace
8. n_exit_spread = 8 = number of exit points to choose from for each entry point
9. n_waves = 16 = number of waves of aircraft entering test space

5.1.2 Data Structure Parameters

1. fleet_path_width = 64 = number of control points per trajectory in simulation
2. sim_interval = 30.0 = simulation heartbeat
3. node_interval = 180.0 seconds = time between control points
4. time_scale = 60.0 = visual simulation time compression factor (sim seconds per real second)

5.1.3 Trajectory Meta-Forces Parameters

1. conflict_buffer_zone = 6.0 km = width of zone outside of the conflict zone where repulsion is active and decreasing with distance
2. repulsive_force = 0.5 = strength of force that increases separation at closest approach
3. elastic_force = 8.0 = strength of force that smooths out trajectories

4. speed_limit_force = 2.9 = strength of force that moves speed toward cruising speed
5. altitude_force = 0.55 = strength of force that moves aircraft toward cruising altitude
6. momentum_decay = 0.8 = proportion of momentum that persists to the next cycle
7. storm_randomness = 0.8 = strength of randomizing force that blows storms around

5.1.4 Experiment Set-Up Variables

1. radius = [500..150] km = entry point radius (variable, controls density)
2. wave_interval = [400..50] seconds = time between waves (variable, controls density)
3. wave_variance = [100..12.5] seconds = max random time to delay individual aircraft entry
4. rsep = 3nm, 4nm, 5nm = aircraft separation (multiple experiments)

5.2 Running Experiments

5.2.1 Visualization

With 10,530 test airspaces simulated and with the potential of over 10 million individual states of the airspaces, visualizations were essential to recognize the phenomena generated in our simulations. The following depictions indicate that, as the density increases for a given airspace configuration, the number of un-resolvable conflicts also eventually increases,very nonlinearly in the manner of a phase transition.

Figure 5-1 shows the two co-planar airspaces with the same 5 nm minimum separation, but differing in the density of aircraft. The airspace on the left has 100 trajectories and zero conflicts. The airspace on the right has 200 aircraft with 723 unresolved conflicts. These airspaces are on opposite sides of the phase transition of satisfiability of the airspace.

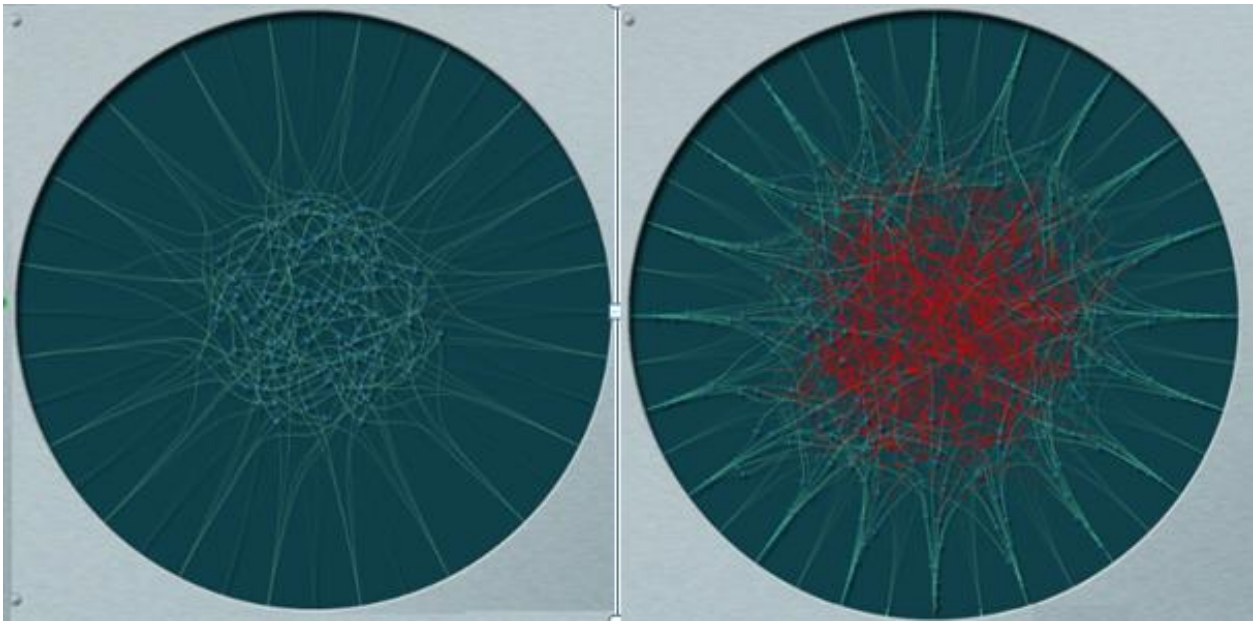

Figure 5-1. Both sides of a phase transition – 5 nm separation, varying densities.

For our experiment depicted in Figure 5-1, we varied the density of the test airspaces gradually to capture the region of density where a phase transition is located under these conditions.

What would happen, however, if we changed the minimum separation distance from 5 nm to a different value? Is the phase transition type behavior general? Figure 5-2 shows an airspace similar to the right screenshot in Figure 5-1, same density of 200 aircraft, except the separation minimum is 4 nm (instead of 5 nm). As we can see, there are only a few unresolved conflicts in this example. Hence, we have captured an instance of an airspace nearing a phase transition.

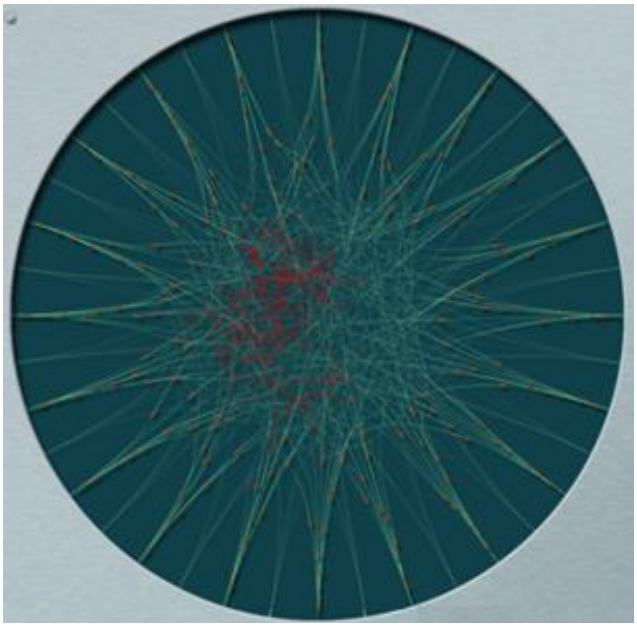

Figure 5-2. Depiction of dynamical trajectories nearing a phase transition – 4 nm separation,200 trajectories.

Quantitative Analysis is discussed in Chapter 6, Results.

5.3 Data

5.3.1 Data Format and Access

The spreadsheet files are structured with one row per run (i.e. instance of an airspace). Each row contains the radius of the airspace, the interval for sending aircraft into the airspace in successive waves, and two types of density: initial and final density. The final density was used in the data analysis in Chapter 6. As was mentioned earlier, the test airspace would in principle always deconflict by "thermal" expansion, given enough time. We truncated the computation time and determined whether or not the dynamical trajectories in the test airspace were deconflicted by that final time, and we recorded the density at that moment.

5.3.2 Density Measurements

Density is a metric used in our phase transition analysis. However, since the test airspace is non-uniform in its loci of trajectory interactions, we needed a more sophisticated method of determining overall density, other than calculating the number of aircraft per unit of test airspace.

For the density computation, we use a Gaussian integral applied to the distance from each aircraft to the measurement point. This formula provided the probability density of finding an aircraft at the specified point if the aircraft positions are considered to have an uncertainty specified by a spread parameter. Alternatively, this approach measured the density of aircraft weighted more heavily near the measurement point, which provided a smooth, well-behaved density measure without discontinuities. Because we were measuring aircraft per unit area, the vertical axis was omitted for this computation. For the purpose of density measurements for our experiments, we measured density at the center of the test jig with a spread of 160 km, which defined a "reaction zone." Density units were measured in aircraft per 10,000 km.

6. RESULTS

Test cases were selected to evaluate the principal tenets of the project regarding the existence, predictability, and manageability of phase behaviors in the airspace. We ran a variety of test cases to test the following hypotheses:

1. Is there a well-defined phase transition in solvability for the test airspace?
2. If so, how does its location depend on a) separation requirements? b) dimensionality of spatial maneuvering permitted? (2D, in the *x-y* plane; or 3D, in *x, y, and z*)
3. Can we efficiently predict whether or not an airspace configuration will be solvable (maintainable in a deconflicted state), resulting in no conflicts in a meaningfully short period of time (to allow for real-time decision-making)?

The test configuration used a fixed number of aircraft and adjusted the interaction density by introducing aircraft into the test airspace in "waves," all of which follow 18 radials toward a central intersection point. We observed that decreased spacing of waves (limited by separation requirements) increased interaction density.

6.1 Evidence of Phase Transitions

We ran a matrix of six experimental configurations, each varying the density over 1755 values. In the first three runs, we constrained motion to occur at a constant altitude (the "2D" cases) and set the required horizontal separation to be three different values: {3nm, 4nm, and 5nm}. In the second three cases, we permitted vertical maneuvering and varied the same horizontal separation requirements while limiting the vertical maneuvering to no greater than 1000 ft. of required vertical separation.

We expected the results to differ between the 2D and 3D cases in a qualitative manner, because aircraft maneuver differently in horizontal and vertical planes. Mixing these two different behaviors was likely to complicate the results away from pure phase transition behavior by mixing two fundamentally different sets of dynamics in the horizontal and vertical planes. For instance, much larger maneuver-induced accelerations were permitted on the horizontal axes than the vertical axis, as the horizontal accelerations combined with the gravitational acceleration to produce a vector sum, the magnitude of which was constrained as a matter of policy. Vertical accelerations add directly to gravity and are therefore more constrained. Nonetheless, allowing 3D conflict detection and trajectory replanning should have had the effect of a relaxation of constraints and push the phase transition boundary to the right on the satisfiability graph.

The 2D results are plotted below (Figure 6-1) for three different values of the separation parameter. The 3nm and 5nm runs were very close to the same shape (proportions of width to height were preserved on a log-linear plot), though the 4nm run was wider. It was clear that the reduction of the separation requirement pushed the phase transition boundary towards higher density.

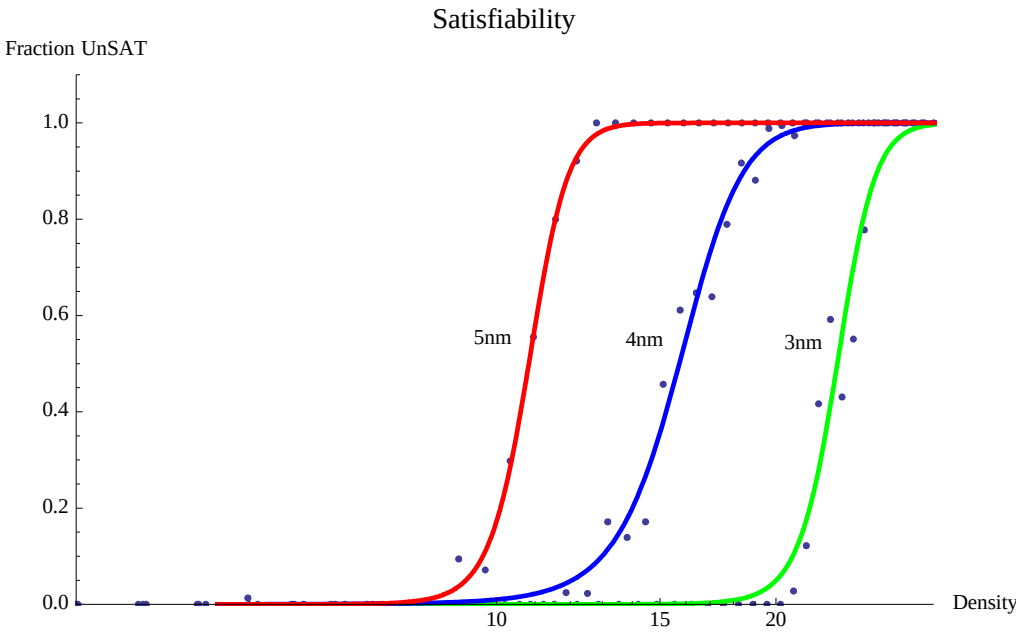

Figure 6-1. Satisfiability Graph for three cases of trajectory separation, 3 nm, 4 nm, 5 nm. The ordinate is the ratio of unsatisfied to satisfied computational solutions for the condition of no conflicting trajectories; the abscissa is the planar density of aircraft in Aircraft per 10^4 km^2 in the test airspace.

The centroids and widths of the three cases are as follows:

2D Dynamics	5 nm separation	4 nm separation	3 nm separation
Sigmoid centroid *c*	10.83 +/- .035	15.74 +/- 0.166	23.292 +/- 0.245
Sigmoid width *w*	0.532 +/- .031	1.250 +/- 0.146	1.113 +/- 0.216
Relative width *w/c*	0.0491	0.0794	0.0478

6.1.1 Phase Transition Dependencies

The standard form of a phase transition (a sigmoid curve) fit the data acceptably well. We fit the data with a sigmoid function of the form $1 - 1/(1 + e^{(x-c)/w})$ where *x* is the final density of the aircraft at the point of solution (or timeout), *c* is the centroid of the sigmoid, and *w* is the width. The position of the centroid of the phase transition forms ratios close to $1/5^2 : 1/4^2 : 1/3^2$, as might be expected in the coplanar case based on "packing" separation requirement volumes. The measured critical densities were much lower than would have been obtained by simply packing such circles, as the curvature of aircraft trajectories was constrained to be operationally realistic, requiring extra space to effect separation. Stacking would allow densities over ten times greater than what we found. This density could realistically occur only in the extremely unlikely case of all of the aircraft flying in formation in the same direction at the same speed, the opposite extreme from our test configuration of converging from all directions into a central interaction region.

We found that the width or sharpness of a standard phase transition curve was a function of the size of the problem (the number of particles in the system). As the number of particles approached infinity, a classic phase transition became infinitely sharp and occurred at a fixed value, rather than a distribution of values. In our experimental setup, the number of aircraft was

held constant, but their spacing was manipulated, thereby controlling the density at the center of the test airspace volume over a certain time period. Our experimental setup was not a homogeneous setup in particle or interaction distribution;. Instead, there was a preferred interaction point (the center of the test volume) which may affect finite-size scaling arguments and the interpretation of particle number. We do not know whether or not that was the reason for the anomalous behavior of the 4 nm separation data as seen above, as the relative widths of the 3 nm and 5 nm cases are very close to each other, but the 4 nm width was substantially larger.

We also performed experiments in a non co-planar configuration (allowing aircraft to maneuver in *x*, *y*, and limited *z*), releasing the constraint that was implemented above for the co-planar case as described in the Experiments section. As expected, the phase transition critical density moved towards higher values for the same values of separation, reflecting the increased degrees of freedom. The data are summarized below.

3D Dynamics	5nm separation	4 nm separation	3 nm separation
Sigmoid centroid *c*	21.96 +/- 0.069	29.40 +/- 0.40	34.74 +/- 0.627
Sigmoid width *w*	0.862 +/- 0.061	0.440 +/- 0.035	1.164 +/- 0.417
Relative width *w/c*	0.039	0.015	0.034

6.2 Dynamics of Solution Finding

A time series was created for each run at each density, showing the number of unresolved conflicts remaining. The time series terminated when zero conflicts remained or was terminated artificially at 1000 iterations and defined as "unsatisfiable" at that point. We created an analogue of the Boolean satisfiability graph [10] by binning the data into 40 density slices (producing a good balance between horizontal resolution and statistics per bin) and plotting the fraction of cases satisfied versus the central density value of each slice.

A typical time series for a solution is illustrated in Figure 6-2, as follows:

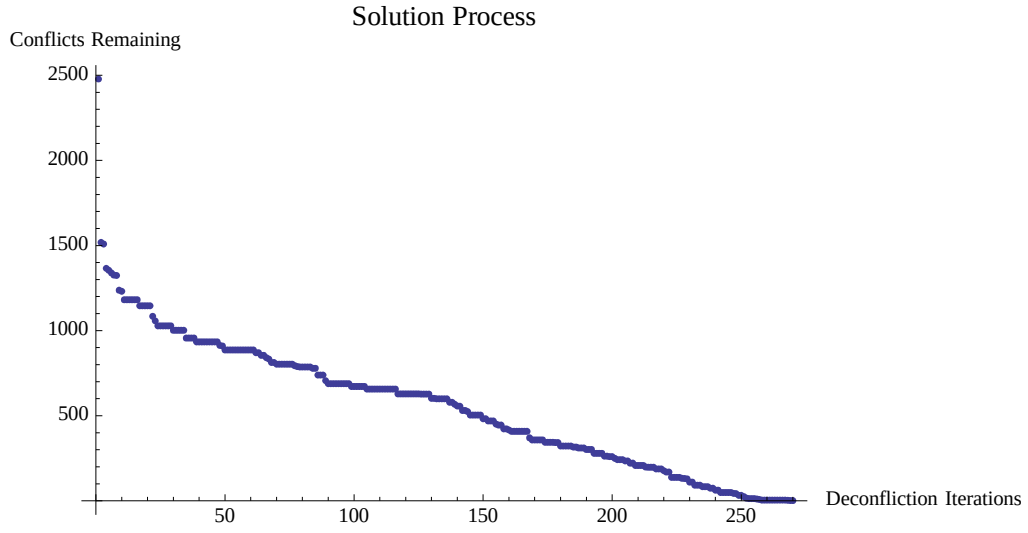

Figure 6-2. Deconfliction management.

The algorithm resolved conflicts as discussed in Chapter 3 on Approach. When the number of conflicts was plotted as a function of time, they tended to display one or both of two patterns. The first pattern was a rapid (usually exponential) decay to a smaller number of conflicts. When the system was far from the phase transition, the exponential behavior could persist all the way to the systemic deconfliction. The second class of behavior was a much slower decay that was reminiscent of thermal expansion: the conflicts that could be quickly removed were removed, but the remaining conflicts required expanding the envelope of the interaction region of the test airspace. That required time, as the algorithm had to successively move all of the trajectories away from the center of the interaction region to create more space, which allowed a resolution. In principle, without any time or space constraints and removing the requirement of fixed flight duration that we used in our simulations, the algorithm could always find a solution by expanding the space and decreasing the interaction density to a level low enough to find a deconflicted solution. We truncated this process at an arbitrary cutoff of 1000 iterations.

The effects of the algorithm can be seen by creating an iterative plot showing the mapping of initial density onto final density for a sample dataset parameterized by 2D maneuvering, 4nm separation, 1755 data points.

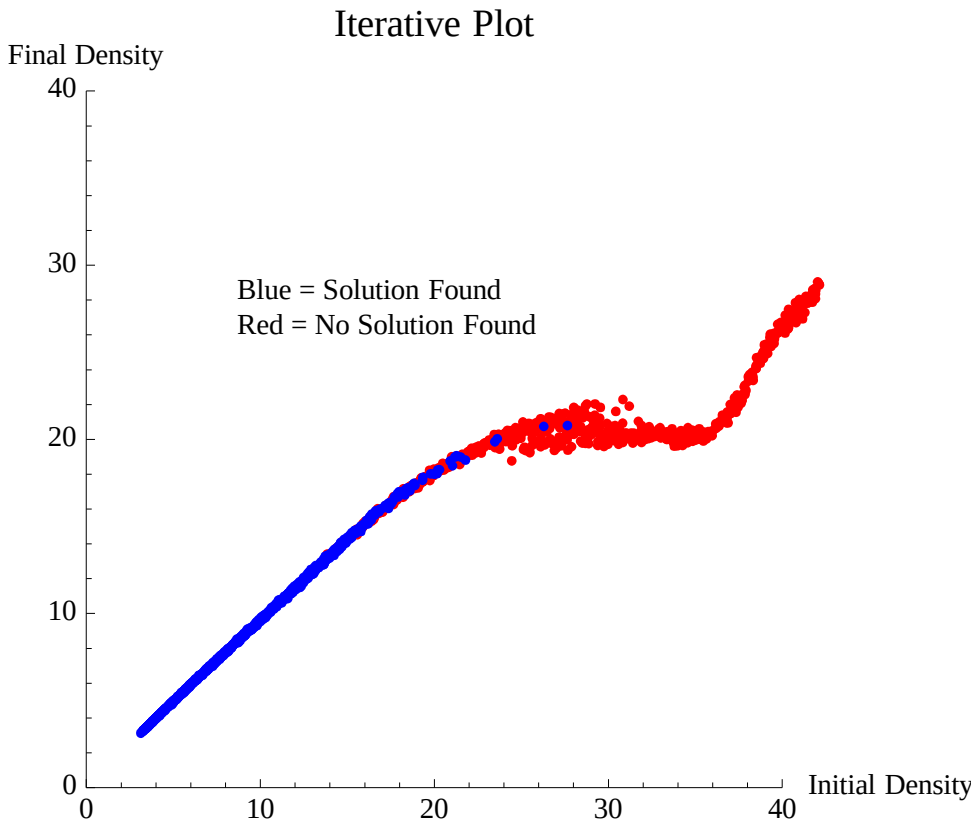

Figure 6-3. Phase transition precursors (planar maneuvering, 4 nm separation minima).

Our experiments revealed that at first the initial density and final density were the same, producing a diagonal line. As the density increased, the interaction volume began to expand, forcing the line to level off. The centroid of the phase transition in the plot below occurred at a final density of 15.738. As the phase transition was approached, the algorithm attempted more and more expansion of the space but was overwhelmed and could deconflict the trajectories in the allotted computation time. In principle, the phase transition could also be moved to the right by allowing more compute cycles, but this process would involve diminishing returns and eventually halt as the remaining optimization headroom was squeezed out of the system.

In practice, this freedom to find a solution by arbitrarily allowing the expansion of the interaction volume would not be possible. Airspace and fuel are finite, and acceptable parameters would be established for the average amount of extra time, fuel, and distance required for conflict-free operation. Our proxy for operational constraints in our problem constrained the number of solution cycles, which had the effect of constraining the size of the interaction region and the amount of extra distance traveled by the aircraft as deconfliction was performed.

6.3 Satisfiability of Airspace

If we wanted to construct the equivalent of a Snell plot, we must account for the fact that the problem statement was different from the statement in the original case. In a Boolean KSAT problem, the computation can terminate based on two events: (1) A solution is found, satisfying the entire expression; or (2) a logical construct of the form $A \wedge \neg A$ is found as part of the process of simplifying the original expression, signifying categorical falsehood. Either one of these events "stops the clock" and identifies the expression as definitely satisfiable or definitely non-satisfiable. Even though highly simplified airspace interactions could permit a formal proof of solution existence or non-existence, they have not been extended to situations involving large numbers of aircraft, replanning because of weather and other sources of uncertainty, and variable speed and altitude.

We propose a modified version of the Boolean KSAT solution methodology in which we ask, "What is the value of the solution iteration number at which we can either a) solve the current airspace configuration; and b) say with a given degree of certainty that a solution will *not be found* within our maximum allowed number of iterations?"

We introduced the concept of "solution hulls" to frame this problem. Consider the set of all solution paths (as shown in Figure 6-4) that result in a successful resolution before 1000 compute cycles. For each value time step (compute cycle number), we computed the highest fraction of conflicts remaining. If we connected all of these points, we had a "hull" composed of extrema values. All successful computations were contained below this hull. When we tested the converse for solution paths that failed to reach a solution, using the lowest fraction of conflicts remaining to create the hull, only failed solutions were above this hull. Because we constrained the computation to terminate at 1000 iterations, the blue and green lines were forced to meet as shown. They also met in the first iteration, as all solution proportions were set to 1 as an initial value—this was off the left side of the plot. When we plotted the two hulls, we obtained the following Phase Graph:

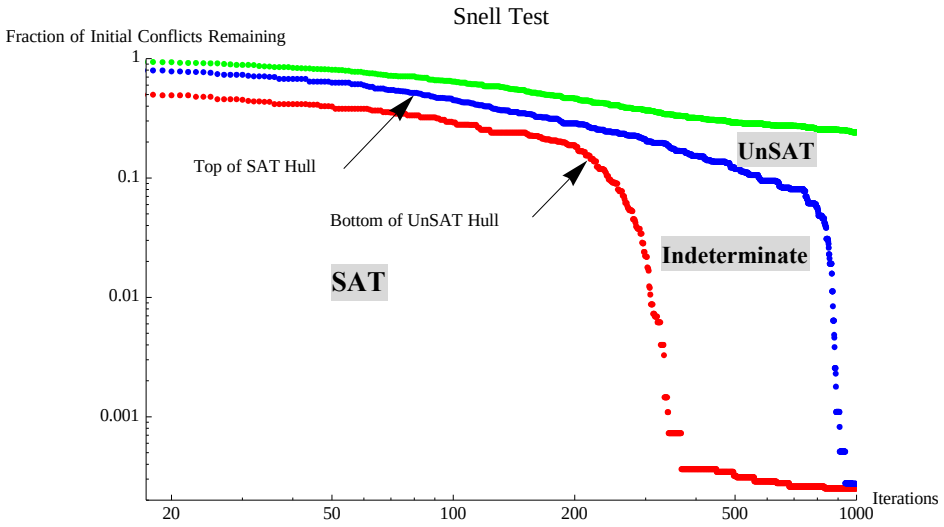

Figure 6-4. Phase region graph.

The space above the blue line was outside of the region circumscribed by the set of the extrema of the union of all of the satisfiable solutions, and was therefore definitely unsatisfiable. The space below the red line was outside of the region circumscribed by the set of the extrema of the union of all of the unsatisfiable solutions, and was therefore definitely satisfiable. An arbitrary solution process between the two lines would remain in an unknown state until it crossed either of the two lines. If it crossed the red line, it would have been satisfiable in the time allotted and could have been computed to the end. If the solution process crossed the blue line, it should have been abandoned at that point. Applying this methodology to a sample dataset (2D maneuvering, 3nm separation, 1755 data points) generated the following analogue of a Snell Plot.

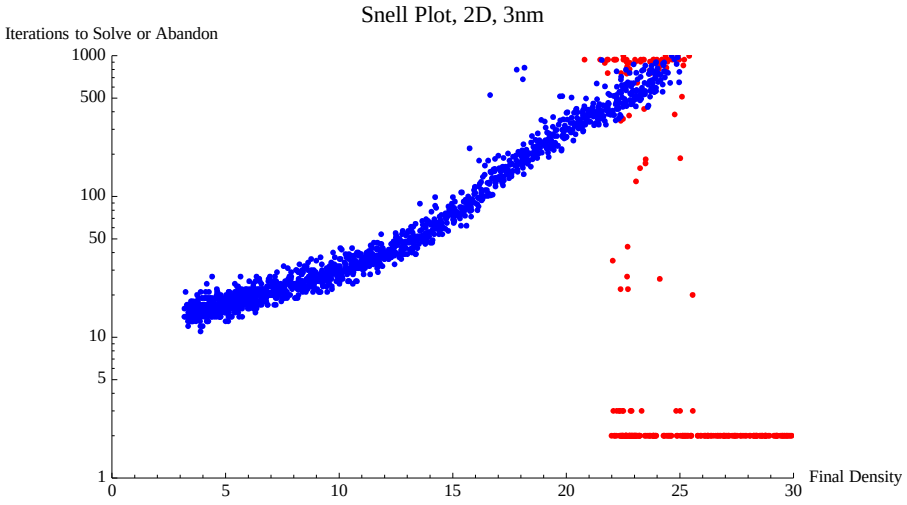

Figure 6-5. Snell-like graph.

6.4 Discussion

There are two types of prediction associated with phase transitions. The first occurs in the relation between density and satisfiability (ability to find a deconflicted solution) of the airspace. The approach to this phase transition has a clear signature in terms of computational complexity as demonstrated by our experiments and analysis. This signature has strategic utility in that an authority in charge of managing airspace would not permit traffic flow on an ongoing basis that was known to produce problematic densities under a wide variety of conditions.

The second type of prediction occurs in the predicted future of trajectories in the simulated airspace. This form of prediction is manifest as the red lines (portions of trajectories that remain in projected conflict) as their projected conflict time is approached. We did not quantify this phenomenon of persistent planned conflict, but it is evident that the presence of such conflicts is an important signal that could have utility in airspace management. Since the environment is changing (due to updated information about other aircraft and weather), a tradeoff might well exist between the "look-ahead" time and the probability of taking mitigating action that turned out to be unnecessary and/or expensive. This tradeoff is similar to the notion of "bounded rationality" in economics—In an uncertain environment with limited knowledge, it does not pay to plan arbitrarily far ahead.

We believe that the results of our experiments described in this report provided evidence of the presence of phase transitions and the possibility of influencing phase transitions by modifying the degrees of freedom for maneuvering by either increasing the dimensionality allowed for deconfliction (allowing vertical maneuvers) or decreasing the separation standard. The details of how the phase transitions would be moved in response to a system's dynamics remain for future research.

6.5 Comments on Uncertainty

In our ABM, the negotiated set of trajectories at any point in time was based on the best available knowledge of all parameters affecting the difference between the original desired trajectory and the current trajectory parameters. Rather than building "uncertainty" in the current ATM sense into the model, we introduced changes into the system that reflected changes in the knowledge that were previously available. The effects of these changes were accounted for in the replanning and, once a new plan was selected, a new set of negotiated 4D trajectories was established. In the initial simulation the only changes we introduced were new paths for the storm cells

The most significant sources of uncertainty include the following:

- Convective weather
- Wind field predictions
- Airport capacity dynamics (as affected, for example, by wind-field changes and the resulting airport configuration)

We observed that maintenance of a system of conflict-free trajectories could be managed by managing the bulk properties (airspeed, direction, altitude, for example) of the sets of dynamical

trajectories in the test airspace, so that a "safe" time/distance was maintained away from the phase boundary. Bulk property control in our experiments meant the maintenance of conflict-free trajectories by keeping a "safe" distance between the current state of the system and a phase transition. "Safe" in this context meant maintaining separation assurance, with conflict-free trajectories, throughout the test airspace.

The presence of a continual flow of updated information implies that we must quantify our ignorance and allow for it. We must define a safe distance from the phase boundary in the context of experience dictated by the best response to weather and other airspace disturbances. It is possible to be ready for something, even if one does not know what it is.

7. CONCLUSIONS AND RECOMMENDATIONS

We set out to create an initial toolbox of algorithms, agent-based structures, and method descriptions for simulating and analyzing a test airspace, while maintaining or increasing system safety. The following lines of investigation were followed in this effort: (1) development of agent structures, including effective approaches for separation assurance for 4D trajectories; and (2) developing a traffic physics/phase transition description and algorithmic measures from the standpoints of satisfiability and computability within representative scenarios. Based on our experiments using an initial toolbox of algorithms, agent-based structures, and methods for introducing agency as described herein for analyzing and managing the complexity of test airspace states, we concluded the following:

- Airspace exhibits phase behaviors, in response to increasing aircraft operations. This behavior reproduces earlier work at NASA and elsewhere, as expected. Phase transition behavior has been seen in systems modeled with either distributed and centralized deconfliction, in addition to our trajectory-centric modeling. This behavior is as expected, as phase transitions are driven by the physics of flight, not by the computational framework used to describe flight.

- The phase boundary between smoothly flowing traffic and "full" airspace has a gradual onset as measured by computational complexity and can therefore be detected and measured.

- Using a trajectory-centric instead of an aircraft-centric approach allows for the prediction in space and time of phase transition type behavior (regions of unsatisfiable airspace). This occurs as a by-product of combined deconfliction and optimization of trajectories. Sufficient advance warning means that an airspace authority could use this information to implement measures that would allow *efficient* implementation of separation assurance well in advance of tactical considerations.

- The location of the phase boundary in terms of aircraft densities can be changed using bulk properties such as separation standards, vertical maneuvering, and other control parameters such as speed.

- Our model has a self-organizing characteristic to it that could be used as a design principle. Dense trajectories created an outward pressure that expanded the interaction region until density was low enough to achieve deconfliction. This expansion was limited by the combined constraints of fixed flight duration and a viable aircraft speed band. Refining our deconfliction algorithm to include local constraint relaxation may produce a viable and efficient way to manage conflicted portions of airspace.

The initial toolbox is suitable for the analyses of test airspace described in this report. We propose comprehensive enhancements, as described in items 1-5 below, and expansion to include terminal and surface operations, thereby allowing full gate-to-gate ABM simulations. All of the items below are of equal importance:

(1) Aircraft Physics: Add econometric models, fuel burn, maneuverability, BADA performance data for all aircraft of interest to NextGen concept evaluation.
(2) Weather: Add current and predicted weather, wind fields and policies for separating aircraft from such weather phenomena. Provide forecasting model data feeds from Consolidated Storm Prediction for Aviation (CoSPA) or other probabilistic-based to incorporate uncertainties in weather into the analysis.
(3) Exclusionary Airspace Dynamics: Implement the ability to activate and deactivate special use airspace and Temporary Flight Restrictions (TFR).
(4) Separation standards: Expand simulations to model a variety (including dynamic) of separation standards.
(5) Objective functions: Expand the simulation to model individual objective functions for each aircraft that can be updated as the flight progresses with actual flight data.
(6) Terminal Airspace with Airport Surface: Implement sufficient structure in the terminal airspace and airport surface details to support dynamic trajectory optimization and management throughout gate-to-gate trajectories.
(7) Model Validity: Conduct comparative assessments of the model performance (with the items above incorporated) against physics-and economics-based data (data from flight or other models that have been verified and produce data appropriate for such assessments).

7.1 Research Topics

We recommend that an enhanced toolset be used to advance the prediction and management of dynamic airspace phase states. We also recommend the application of the enhanced toolset to study the Trajectory Based Operations (TBO) concept proposed below. The latter is described in order of priority with research on terminal and surface issues lower in priority, since we expect enroute enhancements to be added first under NextGen.

7.2 Airspace Phase State Research

- **More Detailed Phase Diagrams:** Apply the enhanced toolset to a more complete understanding the phase behavior of the current airspace architecture. Current architectural features and procedures in the airspace represent additional constraints for 4DT operations. We recommend implementing these features and procedures in order to evaluate the use of our approach to phase behavior prediction. These studies should include evaluation of human factors considerations associated with the work by E. Hollnagel and others on the phases of ATC workloads described as Strategic, Tactical, Opportunistic, and Scramble.

- **Bulk Property Controls:** Evaluate the effectiveness of managing predicted phase boundary "closeness" using separation standards, PBN conformance requirements, speed changes, passing procedures, vertical maneuvering, and control structures (centralized versus distributed). Analyze and evaluate the influence of various "look-ahead" times on the efficiency of airspace operations. Is it possible to look too far ahead?

- **"Playbooks":** Look at "a day in the life" of the US Airspace (for example) under a wide variety of weather days, using recorded historical weather data. If it is possible to

characterize these types of weather days, then is it possible to bias trajectory planning algorithms in advance of the flown day to reduce inefficiencies associated with replanning and deconfliction? This prospect falls under the rubric of "If I knew then what I know now" analysis.

- **Probabilistic Wx:** Analyze the influence of incorporating probabilistic weather data and if that can produce efficiencies equal or better than "Playbooks", above.

- **Mathematics and Algorithms for Computational Acceleration:** Explore approaches to accelerating computational convergence to the desired airspace phase state, including, for example, ellipsoidal versus cylinder-shaped separation zones, Differential Evolution, and additional heuristics that increase computational performance.

7.3 NextGen TBO Research

- **TBO planning:** We will have a flight plan, as currently defined for each aircraft; however, in a 4DT system, some or all of this flight plan will be translated into a 4D trajectory that becomes a contract between the ASNP and the aircraft with concurrence from the ATSP (AOC). This concept provokes several questions listed below in descending order of priority:

  - How brittle is TBO with end-to-end contracts when it must replan because of major perturbations or significant changes in aircraft objective functions? Currently, Traffic Flow Management (TFM) provides only a small reservoir of aircraft in the terminal area to make maximum use of arrival runways. Are these reservoirs necessary? If so, how large should they be? These questions must be answered to predict how far into the future, based on uncertainties, a 4DT system can be managed.

  - How frequently should trajectories be re-negotiated under different conditions? Perhaps, one should generally leave options open as long as possible. There may be situations where downstream effects (hot spots, known perturbations, etc.) should be planned for as early as possible. The accuracy of our trajectory prediction will have an impact on the answers to all of these questions. For example, if weather and winds are highly predictable, trajectory re-negotiation frequency can be very low. Also, for example, what pilot-controller-dispatcher workloads must be accounted for in re-negotiation?

  - What is the sensitivity of the system performance the aircraft conformance to its 4D contract? How far do we allow the aircraft to depart from its 4D trajectory before we declare it "out of conformance" and renegotiate the trajectory?

- **Game Theoretics and Economics:** What system behaviors emerge under different policies for resolving competing objective functions? This question is approachable through the implementation of game theory principals in the trajectory negotiation/replanning process. When scarce resources (primarily arrival runway availability, for example) must be allocated,

not all aircrafts' objective functions can be met. A number of different policies have been proposed to resolve such "conflicts."

- **UAS and the NAS:** Integration of Uncrewed Aircraft Systems (UAS) in the National Airspace System (NAS) presents a set of challenges for which high-fidelity modeling, simulation, and experimentation can support technology strategies, policy analysis, and certification. Sense and avoid (SAA) capability is necessary to provide tactical separation assurance and collision avoidance. The prospect of TBO offers future potential strategic separation for certain types of UAS aircraft and airspace. Additionally, the potential numbers of UAS, the wide variety of performance characteristics, the various approaches to vehicle control, and different operational characteristics will introduce new challenges to the NAS. We recommend applying our toolset to agent-based trajectory models for UAS operational concepts in the near-term and the NextGen NAS.

- **Workloads and Communications Analyses:** By postulating alternatives for specific roles of the ATSP, ANSP, and Flight Deck (where decisions are made, what dispatchers', pilots', and controllers' actions are required, what data has to be exchanged for negotiation, etc.), one can use these data to analyze workload and data communication loads. Research on the allocation of roles and responsibilities for these functions would logically be conducted in laboratories where Human-in-the-Loop (HITL) exploration could be conducted.

- **Dynamic Separation Standards:** Investigate how sensitive airspace phase state (performance of a 4DT system) is in the presence of alternative, perhaps even dynamic, separation standards for trajectories. These separation standards could be varied based on the complexity of trajectory interactions: for example, whether a group of trajectories are aligned in a common direction or whether they are intersecting or converging at fast rates. The separation standards variables in the model that can be proposed for future research are based on the following considerations:

  Enroute Performance-Based Navigation (PBN) with
  - Lateral: RNP = 0.X nm (95% containment; 2 X RNP yields "safe containment")
    - Proposal: Consider X=0.3 for en-route RNP in TBO; the value of X can also be a variable for future modeling
  - Vertical:
    - Climbing/Descending: Assume a fixed uncertainty in vertical profile flight management for future aircraft
    - Level flight: Existing RVSM altimetry 1000 ft above FL 290
  - Longitudinal: The concept of using time for separation sets the longitudinal performance requirement
    - Proposal: Consider using Required Time Performance (RTP) in en-route airspace of + 2 minutes; make RTP a variable in the modeling.

- **Mixed Equipage and Best-Equipped-Best-Served TBO:** We recommend exploring TBO in environments of mixed aircraft performance and equipage for current and future aircraft types and capabilities, particularly in the terminal area. An enhanced toolset could be used to implement variants of PBN, PBC, and PBS (or Total System Performance) variables for

studies of emergent behaviors in TBO airspace, including phase boundary behaviors. Such experiments could help with policy evaluations related to Best-Equipped-Best-Served operational costs and benefits.

- **Environmental Constraints:** Airport noise and enroute contrails present environmental constraints that can be incorporated into the 5DT management and optimization system. In the case of noise, time-of-day-sensitive approach and departure paths would become part of the constraint system for trajectories in the terminal airspace. In the case of contrails, the altitude at which contrails form varies as a function of meteorological conditions. The effects of contrails on global heat balance varies between day and night. We recommend exploring the impact of TBO concepts for dealing with these environmental issues.

REFERENCES

[1] H. Idris, D. Delahaye, and D. Wing, "Distributed trajectory flexibility preservation for traffic complexity mitigation", (June 29-July 2, 2009), 8th USA/Europe Air Traffic Seminar (ATM2009), Napa, CA.

[2] D. Helbing, "Traffic and related self-driven many-particle systems*",* Reviews of Modern Physics, Vol. 73, 2001, pp.1067-1141.

[3] D. Helbing, *et al*., "Micro-and macro-simulation of freeway traffic", Mathematical and Computer Modeling, Vol 35, 2002, pp. 517-47.

[4] T. Nagatani, "The physics of traffic jams", Reports on Progress in Physics, Vol. 65, 2002, p.1331.

[5] T. Vicsek, A. Czirok, E. Ben Jacob, I. Cohen, and O. Schochet, "Novel type of phase transitions in a system of self-driven particles", Physical Review Letters, Vol. 75(1995), pp.1226–1229.

[6] C. Reynolds, "Flocks, birds, and schools: a distributed behavioral model", Computer Graphics, Vol. 21(1987), pp.25–34.

[7] D. Delahaye, S. Puechmorel, R.J. Hansman, and J.M. Histon, "Air traffic complexity based on non linear dynamical systems", Proceedings of the 5th USA/Europe Air Traffic Management R&D Seminar, Budapest, Hungary, 23-27 June 2003, 2003.

[8] M. Mézard, M. Palassini M, and O. Rivoire, "Landscape of solutions in constraint satisfaction problems", Physical Review, Vol. 95, 2003, pp. 200-202.

[9] C. H. Papadimitriou, "Computational complexity", 1994, Addison publishing.

[10] S. Kirkpatrick and B. Selman, "Critical behavior in the satisfiability of random Boolean expressions", Science Vol. 264, 1994, pp. 1297-1301.

[11] J. Clarke, S. Solak, Y. Chang, L. Ren, and A. Vela, "Air traffic flow management in the presence of uncertainty," 8th USA/Europe Air Traffic Management Research and Development Seminar (ATM2009), 2009.

[12] N. Durand, J.-M. Alliot, et al, "Optimal resolution of en route conflicts," Air Traffic Control Quarterly, Vol. 3, No. 3, 1995, pp. 139-161.

[13] N. Dougui, D. Delahaye, S. Puechmorel, and M. Mongeau, "A new method for generating optimal conflict free 4D trajectory", Proceedings of the 4th International Conference on Research in Air Transportation, Budapest, Hungary.

[14] M. Mézard, G. Parisi, and R. Zecchina, "Analytic and algorithmic solution of random satisfiability problems", Science, Vol. 297, No. 5582, 2002, pp. 812-815.

[15] M. Jardin, "Real-time conflict free trajectory optimization," 5th USA/Europe ATM 2003 R&D Seminar, Budapest, Hungary, June 23-27, 2005.

[16] X. Hu, S.-F. Wu and J. Jiang, "GA based on-line real-time optimization of commercial aircraft's flight path for a free flight strategy," AIAA Guidance, Navigation, and Control Conference, Montreal, Canada, August 6-9, 2001.

[17] D.J. Bertsimas and S. Stock, "The air traffic flow management problem with enroute capacities", Operations Research Vol. 46 (1998), pp. 406–422.

[18] Maria C. Consiglio, Sherwood T. Hoadley, David J. Wing, and Brian T. Baxley: Safety Performance of Airborne Separation: Preliminary Baseline Testing. AIAA Report 2007-7739.

[19] H. Erzberger, R. A. Paielli, D. R. Isaacson, and M. M. Eshow, "Conflict detection and resolution in the presence of prediction error," in 1st USA/Eur. Air Traffic Management Res. Development Seminar, Saclay, France, June 1997.

Supplemental References:

[a] Hicks, C. B. (1956). We're Running Out of Airspace. Popular Mechanics. Chicago, IL, USA, H. H. Widnsor, Jr. 105: 8.

[b] JPDO Global Harmonization Working Group, "A comparative assessment of the NextGen and SESAR operational concepts", Report No: 08-001, May 2008, Washington DC [available at www.jpdo.gov].

[c] JPDO TBO Study Team, "Trajectory-Based operations (TBO); Operational scenarios for 2025", Version 1.9.1, September 2010, Washington DC [available at www.jpdo.gov].

[d] S. M. Green, K. D. Bilimoria, and M. G. Ballin, "Distributed air/ground traffic management for en route flight operations." Air Traffic Control Quarterly, Vol. 9, No. 4, 2001, pp. 259–285.

[e] H. Idris, R. Vivona, S. Penny, J. Krozel, and K. Bilimoria, "Operational concept for collaborative traffic flow management based on field observations," Proceedings of the 5th AIAA 5th Aviation Technology, Integration and Operations (ATIO) Conference, AIAA-2005-7434, 2005.

[f] Y. Shang, M. P. J. Fromherz, T. Hogg, and W. B. Jackson, "Complexity of continuous, 3-SAT-like constraint satisfaction problems", In IJCAI-01 Workshop on Stochastic Search Algorithms, 2001, Seattle, WA.

[j] J. Reif and M. Sharir, "Motion planning in the presence of moving obstacles," Journal of the Association for Computing Machinery, Vol. 41, No. 4, July 1994, pp. 764-790.

[k] H. Erzberger, H. and H. Lee, "Constrained optimum trajectories with specified range," AIAA Journal of Guidance and Control, Vol. 3, Jan-Feb, 1980, pp. 78-85.

[l] S. Lidén, "Practical considerations in optimal flight management computations," AIAA Journal of Guidance, Vol. 9, No. 4, Jul-Aug, 1986, pp. 427-432.

[m] R. Hagedorn, "Statistical Mechanics of Strong Interactions at High Energies", Nuovo Cimento Suppl. Vol. 3(1965) pp. 147-187.

[n] R. Storn R and K. Price, "Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces.", J. Global Optimization Vol. 11(1997), pp.341–359.

[o] J. Kennedy J and R.C. Eberhart, “Swarm Intelligence”, Morgan Kaufmann Publishers, 2001.

[p] R. Stone and H. Xin (5 November 2010). "Supercomputer Leaves Competition - And Users - in the Dust", Science Vol. 330 (2010), p. 746.

[q] Darr S., Morello S., Shay R., Jacobsen R.: “A Consideration of Constraints on Aircraft Departure Operations” NASA/CR-2009-215763, June 2009.

[r] David Wing, Bryan Barmore, “NASA Research Results for 4D-ASAS Applications”, Presented at the ASAS Thematic Network 2 Third Workshop, Glasgow, Scotland, 11-13 September 2006.

[s] Liu, Y.-Y. S., Jean-Jacques; and Barabasi, Albert-Laszlo (2011). "Controllability of complex networks." Nature 473: 7.

APPENDIX A. ALGORITHMS, TECHNICAL DESCRIPTIONS AND PSEUDOCODE

A.1 Airspaces, Pseudocode, and Visualizations

A.1.1 Introduction

The following algorithms are intended to enable simulations of the bulk properties of large numbers of enroute dynamical trajectories (and associated aircraft) in arbitrary airspaces.

A.1.2 Successful and Failed Airspaces

The airspace is a platform for generating trajectories that are separated and flyable – if possible. We call an airspace "successful" if all trajectories are separated and flyable. If there is even one trajectory that violates minimum separation distances, or is not flyable, we call this a "failed" airspace. A flyable trajectory is defined as one where all the points along the trajectory lie within some specified range of speeds and accelerations of the aircraft. This definition is a proxy for the laws of physics, aircraft specifications, and airline policies.

In our experiments, we will be looking for the conditions where airspaces transition from successful to failed – as well as the mathematical character of this phase transition boundary.

A.1.3 Pseudocode

One of the objectives of this project is to provide a toolbox of these airspace algorithms. The algorithms will be described conceptually, as well as explicated in technical detail in the form of *pseudocode*. The pseudocode is intended to contain adequate technical detail to enable implementation in a language of choice on a hardware platform of choice.

The pseudocode is organized by six distinct tasks carried out by these algorithms. These tasks are described separately, embedded in conceptual descriptions and commentary.

A.1.4 Organization of Pseudocode

This section describes a higher-resolution version of the simulation pseudocode presented in Section 3. These algorithmic tasks are organized into two main high-level tasks, with task number two containing, among other steps, three important separately defined sub-tasks – five tasks in all, described in five separate sections of this chapter:

1. Initialize the test airspace with an initialization script (see section A.2.6)
2. Perform re-calculation cycles on trajectories (see section A.4.8), where three important sub-tasks:
   a. Apply repulsion/separation force to closest approach of conflicting trajectories (see section A.6.6)
   b. Apply elasticity/smoothing influence to all Control Points on all trajectories (see section A.7.2)
   c. Apply bounding/limits influence to all Control Points on all trajectories (see section A.8.2)

In addition to the toolbox of algorithms for performing experiments on test airspaces, the report describes the process of analyzing the output data from the experimental runs and present results as well. Although there is no pseudocode for this analysis step, we include it here as an "honorary" task 3:

3. Perform post-run data analysis, successful/failed airspace, phase transition structure, etc. (no pseudo code per se, but see chapters 5 and 6 for methodology)

Detailed pseudocode and commentary for each of these algorithms are provided in the appropriate sections below – and co-located in one consolidated listing in Appendix A.

A.2 Initializing the Airspace

A.2.1 Cylindrical Test Airspace

The enroute airspace used for the simulations in this report is a cylinder up to 1000 kilometers in diameter (although the algorithms here have no such limit). Aircraft trajectories enter and exit at precisely designated points on the perimeter of the cylinder, at the designated cruise altitude for the particular aircraft.

Since the intent is to model large numbers of interactions between trajectories, the entry and exit points are positioned randomly across the cylinder from each other, thus forcing all trajectories to attempt to cross through the middle area of the cylinder where they can participate in multiple separation encounters.

A.2.2 Entry and Exit Points of Trajectories

The entry points for the trajectories on the cylinder are positioned at regular intervals around the full perimeter of the cylinder. For example, for an initial parameter setting of 20 entry points ("pigeon holes"), these points are distributed regularly around the cylinder 18 degrees apart. The same number of exit points is similarly regularly distributed, except offset midway between entry points (in this example 9 degrees away), thus avoiding head-on conflicts of comings and goings of aircraft at the edges of the airspace.

Each trajectory is assigned entry and exit points on the perimeter of the cylinder. There is an initial control parameter designating the range of possible exit points that are assignable for exiting. For example, with parameters set at 20 total entry points and a range of six possible trajectory exit points, any of the six farthest (most opposite) exit points could be assigned for a trajectory's exit point. Mating between any particular entry point and allowable candidate exit points is chosen at random, with an ordinary shuffling algorithm used to enforce exactly one trajectory per exit point.

A.2.3 Image of Simulated Airspace

The Figure A-1 below shows two cylindrical airspaces with (different numbers of) trajectories and aircraft entering and exiting at the perimeter. Once constructed, the airspace is rich with curved trajectories resulting from the geometry of the entry and exit points. These initial trajectories have not yet been deconflicted, i.e. deformed to enforce minimum separation.

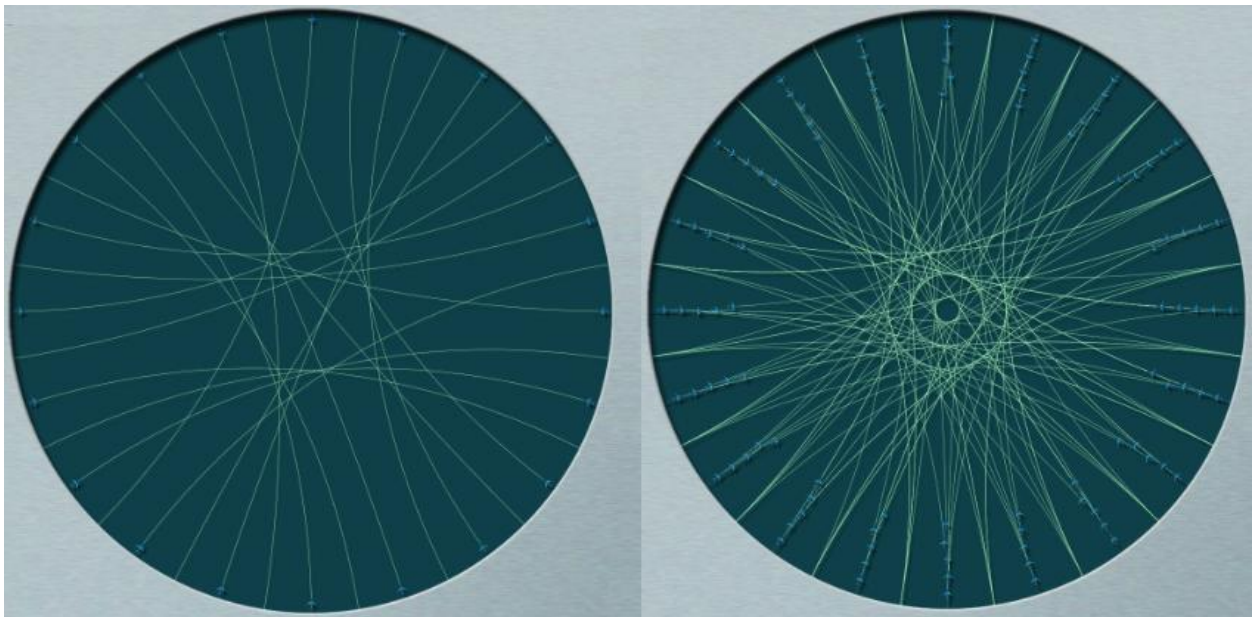

Figure A-1. Top view of examples of a two initialized cylindrical airspaces.

A.2.4 Pseudocode – Assumptions, Abstractions, Classes, Parameters, Visualizations

As mentioned above, we are delivering the algorithms in the form of pseudocode – with the intent that software engineers can generate actual operational code in their language of choice for particular custom implementations. The code below assumes the programmer has already created the necessary object-oriented classes to represent the central abstractions of this genre of simulation, namely a cylindrical test airspace, aircraft, and dynamical trajectories.

As we will see below, these trajectories are represented using Control Points linked together by cubic splines. Other abstractions are also described below including Target Points, and their associated physics-like "forces," momentum, etc.

These classes need to be endowed with appropriate state as well as exogenous tuning parameters. Details on parameters and their particular values for our experiments are detailed below.

Although visualizations are immensely valuable in understanding the complex dynamics of these algorithms, pseudocode here is confined to the calculation algorithms only. No pseudocode for the visualizations is provided here. In fact, there are many possible visualizations one could imagine and desire for this type of research.

A.2.5 Pseudocode: 'Flying' Aircraft

The algorithms delivered in the toolbox perform the dual function of 1) "flying" aircraft within any particular trajectory, and 2) every delta t of Flight Time, dynamically changing the trajectories themselves.

The primary clock of simulations (using these algorithms) is in Flight Time (seconds). Flight Time moves forward (incrementally increases in value) as the simulation proceeds. To "fly" an aircraft (forward), the location and velocity of an aircraft "flying" a trajectory are calculated by sampling the (appropriate cubic spline of the) trajectory at time Flight Time. These values determine the current location, speed, and heading of aircraft displayed in visualizations.

Most importantly, every delta t of Meta Time, the trajectories themselves are re-calculated (replanned) according to current conditions. Naturally, only the future can be replanned. The past is, by definition, frozen to whatever path the aircraft actually flew.

A.2.6 Pseudocode – Initializing the Test Airspace

The following listing is the 1st of 5 fragments of pseudocode, corresponding to task 1 outlined in section A.1.4 above as "1. Initialize the test airspace with an initialization script".

1. Initialize parameters from an initialization script, including setting the following variables:
   a. Size of test airspace cylinder
   b. Number *N* of entry and exit point 'pigeon holes' for aircraft on perimeter of cylinder
   c. Range *R* of exit points on far side of cylinder from entry points
   d. Value of *DeltaFlightT*, delta time of Control Point time spacing defining and controlling trajectory shapes
   e. Value of *DeltaMetaT*, delta time for replanning (re-calculating) dynamical trajectories
   f. Specific profiles of *N* aircraft chosen to participate in this airspace simulation
2. Divide the perimeter of the cylinder into *N* equal parts
3. Choose *N* aircraft with designated default cruise altitude and speed
4. Construct *N* entry points at degree positions: $i * 360/N$
5. Construct *N* exit point at degree positions: $i * 360/N + 360/N/2$
6. For each *i-th* trajectory of these *N* trajectories.
   a. Choose an aircraft
   b. Set the *i-th* entry point at position $i * 360/N$ in degrees around the cylinder's perimeter
   c. Set the velocity of the entry point to direction toward center at default aircraft speed
   d. Set the *i-th* exit point at some position around the cylinder's perimeter, randomly chosen from the R exit points farthest from the entry point.
   e. Set the velocity of the exit point to direction from the center at default aircraft speed
   f. Construct the cubic spline from the entry point to the exit point
   g. Construct a set of Control Points, one per time *t* (see Control Point *DeltaFlightT* on line 1.d above)
   h. Sample the cubic spline constructed on line 6.f above at each time *t*, and fill in values of Control Points
7. Initial trajectories are now constructed for each wave of aircraft entering test airspace
8. Update the positions of the aircraft on the trajectories associated with each aircraft according to the time values of the points on the cubic splines. Visually this is the process of 'flying' the aircraft to an updated location and heading. Refer to section above "Pseudocode: 'Flying' Aircraft" for more detail on this process.

9. At a designated rate, construct new waves of a set of *N* trajectories, and 'fly' them similarly

This pseudocode continues as line 10 in section A.4.8 below.

A.2.7 Controlling Airspace Density

Our experiments require recording and analyzing specified metrics of the behavior of the airspace as a function of aircraft density and separation minima. Density is controlled by the rate at which aircraft are introduced into the airspace and the radius of the airspace. Rate of aircraft entering the airspace is controlled by the number of entry points in the cylinder and the rate at which waves of aircraft enter through the entry points.

Refer to chapter 5 for additional detail on simulation parameters, including controlling the geometry of the airspace and the flow of aircraft into the airspace affecting density.

A.2.8 Heterogeneous Aircraft

Although these algorithms focus on trajectories, each trajectory has a unique aircraft flying the trajectory for the duration of its flight. These algorithms support multiple heterogeneous aircraft types, with varied flight characteristics, including default cruise altitude, speed, etc. We recommend storing these aircraft profiles in a small (changeable) database in spreadsheet form, as we do in our implementations.

A.3 Trajectories

A.3.1 Initial Trajectories

All trajectories in this simulated airspace are enroute only, beginning at top of climb, and ending at top of descent. By convention, the altitude of the endpoints of every trajectory is the default cruise altitude of the particular aircraft flying the trajectory.

The velocities of trajectories at the entry and exit points at the edge of the cylinder have direction towards the center of the cylinder (i.e. perpendicular to the tangent of the cylinder), and a magnitude equivalent to the default cruise speed of the associated aircraft.

The initial trajectory path is the cubic spline connecting the entry and exit points of the cylinder. Since entry and exit points are slightly offset from one another, trajectory paths will never be a simple straight-line diameter across the cylinder. Rather they will be curved, following the shape of the cubic spline.

A.3.2 Note on Cubic Splines

Cubic splines are used extensively in representing trajectories here, as well as in all of the calculations of forces applied to trajectories to move and modify them.

A natural way of representing curves is with polynomials, which have the convenient property that they are easily differentiable for ease of inter-calculating locations, velocities, and accelerations. In addition, polynomials are computationally efficient.

For trajectories, we want to specify both three-dimensional location and velocity of both end points of each spline into our polynomials. Hence we need a third degree (cubic) polynomial in each spatial dimension to accomplish this, because a 3rd degree-polynomial has four free variables which are uniquely specified by four constraints of position and velocity at both endpoints. Once defined, any point along a cubic spline can be quickly sampled for location, velocity, and acceleration.

Many readers of this document may be familiar with control points for cubic splines used in graphics applications. These control points are somewhat different from the ones we use, in that graphics applications typically use *four* control points to define each segment.

We use cubic *Hermite* splines, which are defined by *two* control points with velocity as well as position, and all control points are on the trajectory. A control point is simply the position and velocity of the desired trajectory, sampled at a specified time. This difference is due to our interest in time and velocity, which is not shared by graphics applications.

A.3.3 Control Points – Representing Complex Trajectory Path Shapes

Although trajectories are initialized as simple cubic splines connecting entry and exit points on the perimeter of the cylinder, as trajectories need to deform to maintain separation from other trajectories, they will need to take on more complex shapes.

In order to represent arbitrary complex curved paths through the airspace, trajectories are endowed with "Control Points," spaced regularly in time, one Control Point every delta *t* (DeltaFlightTime) along the entire trajectory path. Control Points are connected together with cubic splines.

Hence, trajectories are actually a set of many cubic splines, connected together via Control Points. Although the initial trajectory is calculated as a single cubic spline connecting the entry and exit points of the cylinder in a single graceful curve, in fact, this single spline is sampled at each time *t* of each of the Control Points of the trajectory, and the full cubic spline trajectory is re-represented as a set of cubic splines. Once represented in this compound spline fashion, it is still the same curve, but with much more flexibility to be deformed as forces are applied to it later in the process.

Aelow Figure A-2 shows a single arced cubic spline represented as nine shorter (almost linear) cubic splines, connecting 10 Control Points. (The yellow Control Point marks the beginning Control Node at the entry to the enroute airspace.)

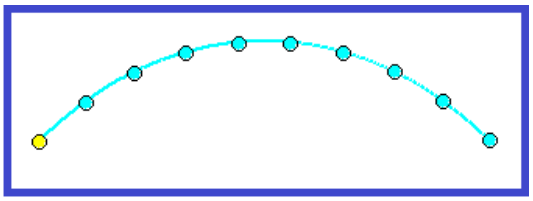

Figure A-2. A trajectory represented by a set of Control Points connected by cubic splines.

A.3.4 Implementation Note

Although in principle trajectories may have an arbitrary number of control points, in our implementations of these algorithms we limit the number of Control Points to 64 per trajectory. For example, with a 1000 km diameter test airspace cylinder, we can have about one Control Point per minute of Flight Time.

A.3.5 4DT Trajectories

Conceptually trajectories are abstractions embedded in both space and time. Hence trajectories are four dimensional entities – one temporal and three spatial dimensions.

However, due to the exigencies of airspace, trajectories may need to be replanned dynamically. In our algorithms, at every delta *t* time increment, all the trajectories are replanned (re-calculated) according to current conditions. The calculation may or may not actually result in changed paths. If needed, trajectories will be re-shaped by altering one or more Control Points on the trajectories. Trajectories managed by these algorithms described here are quite dynamical.

A.3.6 5DT with Replanning

Every 4DT Trajectory is itself a dynamical entity, replanned every delta *t*. Hence, there are two types of time. There is the Flight Time embedded into every instance of a trajectory, but a trajectory itself changes over time. Thus, there is an additional Meta Time as these 4DT trajectories themselves dynamically change over time.

In this sense, dynamical trajectories are abstractions spanning space and *two* types of time. Hence these dynamical (suites of altered) trajectories are conceptually five dimensional entities – two temporal and three spatial dimensions.

Intuitively, a single trajectory instance is like a hard strand of spaghetti lying still on a cold plate – whatever curve it has is statically fixed in place. A collection of dynamical (suite of changing) trajectories is like a soft strand of spaghetti curling, stretching, and moving away from other strands of spaghetti in a pot of boiling water.

Over the course of its Flight Time an aircraft might fly parts of many dynamically replanned trajectories. An actual flown flight path is, in effect, pieced together from many instances of trajectories as the dynamical replanning process re-shapes the trajectory in Meta Time, responding to separation, etc. issues ‘du jour’.

The concept of 5DT is illustrated in Figure A-3 where a trajectory itself is modified. The future of any particular trajectory has a Flight Time associated with it. In addition, trajectories are modified at some time *t* in Meta Time as well. Hence, we simulate dynamical trajectories.

In Figure A-3 an original trajectory (blue), possibly modified to detour around some obstacle at some time *t* in Meta Time, thus generating modified trajectories. Each trajectory and its

associated Control Points have time variables in Flight Time. In addition, these trajectory modifications occurred at some different flavor of time *t* in Meta Time.

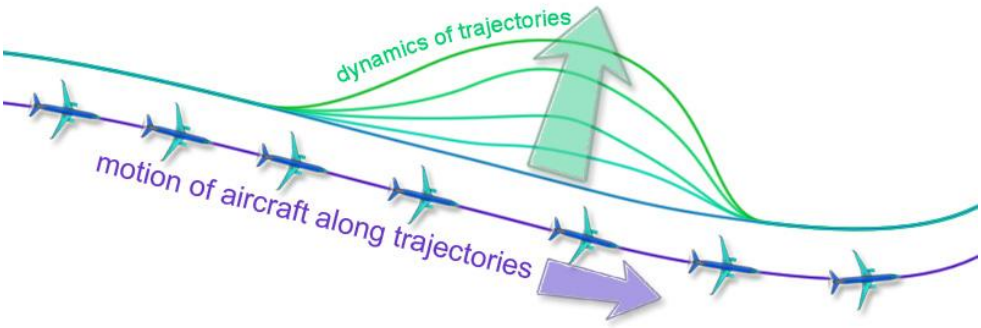

Figure A-3. Trajectory dynamics.

A.3.7 Control Points

As described above, Control Points are used to represent and define the path of a trajectory. A trajectory consists of one Control Point for each delta *t* of its path. Control Points are connected together by cubic splines.

Control Points are represented by 7 double-precision values:

- Time, in seconds, in Flight Time – constant
- 3 *x*-*y*-*z* spatial coordinates, in kilometers
- 3 *x*-*y*-*z* velocities, in km/sec

When a trajectory is altered (changed to a different trajectory), the values of one or more Control Points are changed. In particular, a Control Point can be changed by revising the values of the spatial and/or velocities. Note that the Flight Time associated with the Control Point is immutable, i.e. is a constant.

A.4 Deforming Trajectories

A.4.1 Target Goals

The values of Control Points are informed by applying iterated influences to the trajectories, producing Target Points for moving Control Points.

In the algorithms for applying specific influences detailed below, all of the influences calculate some Target Point goal (an influence vector)– regardless of how each influence makes its specific calculation. The *lingua franca* for all influences is to calculate one or two Target Points per application of the influences, which then directs the universal deformation machinery, described below. This process simplifies and reduces the process of generating deformations to only calculating Target Points.

Once a Target Point is calculated, it is handed off to the general dynamical machinery for actual movement of the Control Points (change their positions and velocities) according to multiple forces acting simultaneously on each Control Point.

A.4.2 Moving *Toward* Target Points

Rather, than wholesale moving Control Points to these Target Points, the Control Points are instead moved *toward* the target goals incrementally. More precisely, these influences act to change the acceleration of a Control Point in some specified direction, causing it to eventually arrive there (or even beyond) unless, of course, it is pulled in other directions by other influences.

As we will see below, the actual effect of many of these influences acting in concert is to generate a constellation of effects on Control Points (more precisely accelerations on Control Points in Meta Time) toward various Target Points, which are summed and applied in aggregate to each Control Point. Hence, the Control Points move in carefully coordinated ways, deforming the trajectories toward the macro goals of separation and efficient flyable flight paths.

A.4.3 Magnitude of Influences

Once a Target Point is identified by applying an influence, the effect of the influence is calculated as the difference between the current location of the point and the location of the Target Point. Differences are calculated in all six spatial dimensions of the Control Point – *x y z* position and *x y z* velocity. As we will see later these differences multiplied a constant are added to the Momentum Buffer.

The effect here is to implement a dynamic similar to Hooke's Law (F = -kx), where the farther away from the goal, the larger the influence (and acceleration) towards the goal.

In our algorithms, in the case of separation, we depart slightly from Hooke's Law by applying a sigmoid function to the otherwise linear response. The sigmoid function is applied, centered at minimum separation. Therefore, repulsion is applied up to the safety margin, but it is significantly stronger below minimum separation.

This relationship is important so that even a single separation violation is given increased importance (and acceleration in Meta Time). The result is a quick resolution of airspaces, which if solvable, converge to zero conflicts quickly.

Figure A.4 shows a Control Point being moved according to current influences. Note that both location and velocity can be affected. Because of the trajectory representation as a series of cubic splines, these can be independently influenced.

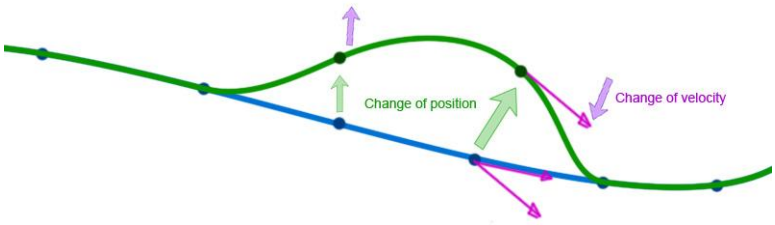

Figure A-4. Forces acting on location and/or velocity of trajectory Control Points.

A.4.4 Re-calculation cycles

The primary rhythm of the dynamical airspace described here is to generate dynamically changing trajectories, one cycle every delta *t* in Meta Time. There can be arbitrarily re-calculation event along a trajectory.

From the point of view of an aircraft, limited only by available computation cycles, there can be one trajectory re-calculation (replanning) cycle carried out every few seconds of Flight Time. Hence, in practice, this process of many re-calculations per aircraft enroute flight is pretty close to *continuous* replanning of the aircraft's trajectory *while* it is flying.

The system attempts to carefully deform the trajectories such that separation is enforced, and the paths are always flyable (i.e. velocity and acceleration limits are maintained).

A.4.5 Deformation (sub-) cycles

A secondary rhythm occurs within each re-calculation cycle. Many steps (sub-cycles) are required to properly deform the current trajectory so as to respond to current pressures and urgencies (e.g. separation exigencies).

In each deformation cycle, the trajectories are gradually, incrementally changed a small amount at a time. All the deformation cycles taken together within a single larger re-calculation cycle may have a very large impact on trajectories, depending on the pressures at that moment in the aircrafts' journeys. These "pressures" are the influences of repulsion, elasticity being applied to the trajectories.

Before a re-calculation cycle, a trajectory has some set of Control Point values. After the re-calculation cycle, the Control Points may have new values (and in effect be a new trajectory). At this level of detail, the seven values described above are necessary and sufficient for representing Control Nodes.

However, during the re-calculation process itself, an additional state is required to coordinate the gradual deformation of the trajectories over many deformation cycles.

A.4.6 Momentum Buffer

The additional state needed to coordinate deformation is stored in the Momentum Buffer. Momentum, as implemented here, enables continually maintaining near-optimal trajectories over the course of entire flights. The purpose of deformation cycles is to iteratively calculate the underlying dynamics required to 'glide' or translate the trajectories into new positions in the airspace, like the wiggling strands of spaghetti described above. This dynamic movement requires that the successive deformation cycles be tied together into one (apparently) continuous movement, guided by local pressures. This dynamical "gliding" process is analogous to momentum (with friction) in physics.

To link deformation cycles together to accomplish (apparently) continuous movement of trajectories, additional state is needed to augment the state already contained in the Control Points. This additional state is captured in the Momentum Buffer, which stores the current state of dynamic movement of each Control Point. Using the principle of inertia, if a Control Point is moving in a given direction, the Momentum Buffer will enable it to keep it moving in that way, modulo friction. This has the effect of smoothing adaptive response of trajectories to each other and the environment and reducing numerical instabilities by introducing exponential damping to trajectory corrections.

For every Control Point, there is exactly one Momentum Buffer. It has the same structure as a Control Point with the exception of no need to repeat Flight Time (which is a constant in a Control Point). A Momentum Buffer has the following structure:

- 3 *x*-*y*-*z* spatial coordinates in kilometers
- 3 *x*-*y*-*z* velocities in km/sec (seconds in Flight Time)

As stated above, the purpose of the Momentum Buffer is to provide inertia to the trajectory Control Points during the deformation process, so influences on trajectories continue to have their effect over subsequent deformation cycles.

For example, if part of a trajectory is being repelled by another entity (another trajectory, weather cell, etc.), the trajectory receives an initial push (acceleration in Meta Time) from the force of repulsion. With momentum machinery built in to this process, the initial push continues to push on the trajectory, even after that deformation cycle, into subsequent deformation cycles. Visually, this has the effect of trajectories gracefully gliding away from each other.

A.4.7 Momentum and Friction

In addition to momentum, there is also a notion of friction. Momentum is attenuated every deformation cycle, thus gradually reducing the effect of previous accelerations applied to Control Points. Hence, trajectories glide to a stop in the absence of applications of new forces.

Algorithmically, each Momentum Buffer accumulates the effects of the multiple forces acting on a Control Point, when they are then added to the values of the Control Point at the end of each deformation cycle. The Momentum Buffer retains its values across deformation cycles although they are attenuated every cycle, resulting in an exponential decay of the original influence.

A.4.8 Pseudocode for Re-calculations of Trajectories

The following listing is the 2nd of 5 fragments of pseudocode, corresponding to task 2 outlined in section A.1.4 above as "2. Perform re-calculation cycles on trajectories".

The following is the high-level pseudocode for performing re-calculation cycles, continuing from line 9 of the pseudocode in section A.2.6 above.

10. Run the trajectory initialization script
11. Initialize all the Momentum Buffers to zero
12. Repeat the following until the end of the simulation
    a. Repeat until deconflicted or maximum re-calculation cycles exceeded
        i. If maximum iterations exceeded:
            1. Note separation failure
            2. Either continue, or exit depending on preferences
        ii. Collect enumeration of all pairs of conflicting trajectories
        iii. Apply Influences to Momentum Buffers (generating Target Points)
            1. * Apply repulsion/separation force to closest approach of conflicting trajectories
            2. * Apply elasticity/smoothing influence to all Control Points on all trajectories
            3. * Apply bounding/limits influence to all Control Points on all trajectories
        iv. Add the effect of each force to its corresponding Momentum Buffer
        v. Apply Momentum to trajectories (according to target points)
            1. Add each Momentum Buffer to its corresponding Control Point, component by component
            2. Control points will have moved (changed location and/or velocity) some (small) amount where the Momentum Buffers were non-zero
        vi. Attenuate Momentum Buffers (analogous to applying friction)
    b. Fly aircraft forward one simulation time step *(Note: this is not one Control Point)* by adding delta-t to the time value of aircraft, and sampling each aircraft's trajectory at this new time. (See section above on "Pseudocode: Flying Aircraft")
    c. Record measurements (density, number of conflicts, etc.)
    d. Update visualization
13. The simulation of this airspace is complete with data collected for later analysis
14. ** Perform post-run data analysis, successful/failed airspace, phase transition structure, etc.

Note that asterisk * starred lines refer to algorithms described in more detail in sections below. The last line14 marked with a double asterisk ** refers to the process described in chapters 5 and 6 on analyzing data generated by airspace experiments. There is no pseudocode associated with line 14 above.

In the section above on the organization of pseudocode, we described the six tasks carried out by our algorithms. So far we have provided the pseudocode for two main high-level processes of the airspace simulator:

1. Initialize the test airspace
2. Perform re-calculation cycles (immediately above)

At a high level, this pseudocode is a complete description of the simulation algorithms provided in the Toolbox. However, there is still additional pseudocode needed to explicate the deeper details of the simulation processes. The starred lines in 12.a.iii above indicate the detail yet to come. In particular, we will now describe and present the pseudocode applying these influences to the trajectories to deform them appropriately. These (sub-) tasks are the following:

a. Apply repulsion/separation force to closest approach of conflicting trajectories
b. Apply elasticity/smoothing influence to all Control Points on all trajectories
c. Apply bounding/limits influence to all Control Points on all trajectories

A.5 Deforming Influences

A.5.1 Three Influences

Trajectories would remain unchanged if there were no pressures to change their paths. In a sparse airspace, initial trajectories can be quite stable with no need to change already optimal trajectory paths.

However, in more dense airspaces, separation may force changes in paths, typically lengthening the paths to go around some obstacle. On the other hand, economic pressures will tend to force the path to be more evenly curved to save fuel, fly more smoothly, etc. In addition, physical limits on velocity and acceleration will tend to force the path into more flyable shapes as well. The shortest possible path may not be flyable. In principle, our algorithms search for shortest flyable de-conflicted paths (modulo issues around local minima, etc.).

These practical requirements for trajectories can be conceptualized and implemented as three additive vector influences on trajectories thus simplifying the problem, as well as simplifying the algorithms used to deform the trajectories.

Our Toolbox of algorithms supports these three types of influences that act to deform trajectories:

- **Repulsion** – maintains minimum separation of nearby trajectories, generated by pseudopotential
- **Elasticity** – keeps trajectories gently curved to minimize distance, fuel consumption, etc.
- **Bounding** – keeps trajectory velocities and accelerations within physical and policy limits

A.5.2 Application of Influences

For every deformation cycle, the three influences above are applied to some or all of the Control Points, depending on the type of force:

- **Repulsion** – only on closest approach of pairs of conflicting trajectories

- **Elasticity** – on every Control Point
- **Bounding** – on every Control Point

A.5.3 Deforming Trajectories

As described above, the result of applying an influence is not to move a Control Point per se. The effect of an influence is simply to contribute effects (more precisely accelerations in Meta Time) to Control Points, implemented in our algorithms as adding values to the Momentum Buffers.

A.5.4 Repulsion – Separation

Maintaining minimum (safe) separation between trajectories is arguably the most important constraint of the trajectory replanning process. Rather than doing conflict detection and resolution per se, the innate character of our trajectory strings or tubes is that they repel each other in such a way as to be always in a state of separation.

This method of separation is possible because entire trajectories are separated (throughout their entire length), as opposed to separating aircraft *per se*. In effect, there are no surprises postponed into the future except when new conditions arise, for example, changing weather conditions. Even then, entire trajectories are once again immediately and fully separated through the operation of repulsion. We will present the details of the repulsion algorithm further below.

The most complex influence to apply is repulsion, because it is only applied conditionally – that is, only when conflicts are detected among pairs of trajectories. The process is additionally complex because conflicts themselves must be detected dynamically for each deformation cycle.

New conflicts may arise for a trajectory resulting from de-conflicting some other pair of trajectories. In addition, weather cells may move between one re-calculation cycle and another, generating new conflicts with the storm, reverberating to new conflicts between other previously deconflicted pairs of trajectories.

A.5.5 Conflict Detection

At the beginning of each deformation cycle, the repulsion algorithm requires an enumeration of the set of all pairs of trajectories that are currently in conflict – and if conflicting, the algorithm needs to know the precise points of closest approach for each trajectory.

The simplest algorithm for this is to exhaustively search all possible pairs of trajectories, for those for which the closest approach is less than the minimum allowed separation. There is no simple analytic expression for the closest approach of two cubic splines. However, a numerical approximation is fast and practical. Our algorithms sample the cubic splines at a granularity of 32 samples between each pair of Control Points. In practice, this works quite well.

A.5.6 Scaling and Optimization of Conflict Detection

The simple exhaustive algorithm for conflict detection described above scales as the square of the number of trajectories. Hence, for large numbers of trajectories, optimizing the conflict detection algorithm becomes a priority.

There are a number of candidate optimization algorithms. The most straightforward approach is to "tile" the 4DT space, and annotate the tiles with all the control points that fall within corresponding tile areas. Since control nodes tend to move slowly, so the content of the tiles is fairly stable, this approach is quite efficient, scaling linearly with the number of trajectories

However, since this is only an optimization and there are multiple candidates to accomplish this, and since this optimization is not focal to the central set of Toolbox algorithms, we have omitted details of such algorithms here.

A.5.7 Elasticity

Applying a repulsive influence for maintaining separation is a powerful technique. However, this influence alone is insufficient for generating stable trajectories. Such paths are under-specified causing instability of path locations, or "Brownian Motion" as paths remain restless.

In these algorithms, we also apply an internal influence of elasticity on each trajectory. This influence causes the trajectories to follow ever more flyable, relatively shorter curved paths, conserving fuel, while still maintaining separation via the repulsive inter-trajectory force.

Elasticity can be thought of the tendency for short sections of a trajectory to imitate the natural curve of longer sections of the trajectory. With the removal of obstacles, elasticity will return the trajectory to its initial cubic spline connecting the entry and exit points in the space. However, since obstacles are endemic to a crowded airspace, the force of elasticity will do its best under whatever circumstances and separation issues the trajectory finds itself within in any particular moment.

A beneficial emergent property associated with elasticity is that all of the applied influences propagate throughout the airspace. "Pressure" from highly conflicted regions of the airspace cause outward expansion, thus reducing local density. Without elasticity, this emergent property of "pressure" is negligible.

As we will see below, elasticity is applied by using the same cubic spline mathematical machinery that we use to generate trajectory paths from Control Nodes. The effect of this algorithm is to reduce accelerations along the trajectories. Reducing accelerations has the bonus of making trajectories more flyable.

A.5.8 Bounding

The third influence in addition to *repulsion* and *elasticity* is *bounding*. This effect is necessary to assure that the trajectories are flyable. In the simplified abstracted world of these algorithms and simulations, this means that speeds are limited to a specified minimum-maximum range. Without this influence, in the extreme case, flight conditions could violate speeds associated with drag divergence Mach numbers or stall.

Note that accelerations (in Flight Time) must be limited as well, but this is handled by the influence of *elasticity* on the trajectories.

A.6 Repulsion / Separation Algorithm

The purpose of applying the repulsion influence to a trajectory is to generate Target Points that can be turned into changes on Control Points as described in commentary and pseudocode above.

In this section, we will describe how separation encounters generate associated Target Points.

A.6.1 Minimum Separation Plus a Margin

In our algorithms, we use the customary notion of minimum separation (e.g. 5 nm). In addition, we add the notion of a "margin" of separation (e.g. 2 nm). When a conflict is found, our algorithms use a separation goal of minimum separation plus an extra margin (e.g. 5+2 = 7 nm). This policy enforces extra safety while guarding against some potential oscillations at the boundary of the separation minimum.

Therefore, the Target Point is constructed based on this more aggressive separation distance, including the margin.

A.6.2 The Null Separation (non-) Problem

For thoroughness, we will begin with two trajectories that *are* adequately separated. Figure A-5 shows this situation. The two trajectories are just at the minimum desired distance apart including the green margin. The trajectories are in blue with control nodes marked as points. Separation minimum (e.g. 5 nm) is displayed in red, with the extra margin displayed in green. In this case, there is no separation issue, so no repulsive force needs to be applied..

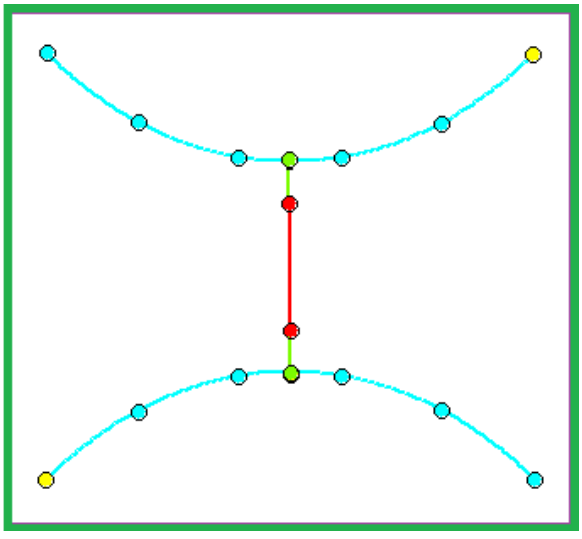

Figure A-5. Two adequately separated trajectories

A.6.3 Separation Conflict

However, in Figure A-6, we do have a separation conflict. The trajectories are too close to each other, indicated by the red line segment, which is longer than the shortest distance between the two trajectories (at the same time *t*). The trajectories are in blue with control nodes marked as points. Separation minimum (e.g. 5 miles) is displayed in red, with the extra margin displayed in green. In this case, the two trajectories are too close in space-time, so separation will be attempted by applying a repulsive influence to both trajectories.

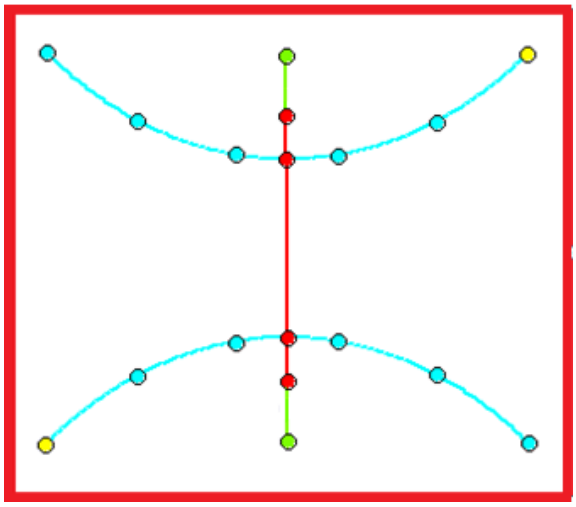

Figure A-6. Two trajectories in conflict, i.e. not adequately separated.

In an attempt to resolve this conflict, a repulsive influence will be generated on both trajectories (or just one aircraft trajectory if the other is a weather cell, etc.). Since the point of closest approach (and greatest conflict) is between Control Points, that point on each trajectory cannot be directly moved. Instead, Target Points are calculated for adjacent Controls Points on each side of the conflict.

The diagram in Figure A-7 shows the algorithm for calculating the Target Point *B* for current point *b*, and likewise, the Target Point *C* for current point *c*. Target Points *B* and *C* are calculated by sampling the cubic spline *a*-*P* at time *b*, and cubic spline *P*-*d* at time *c*.

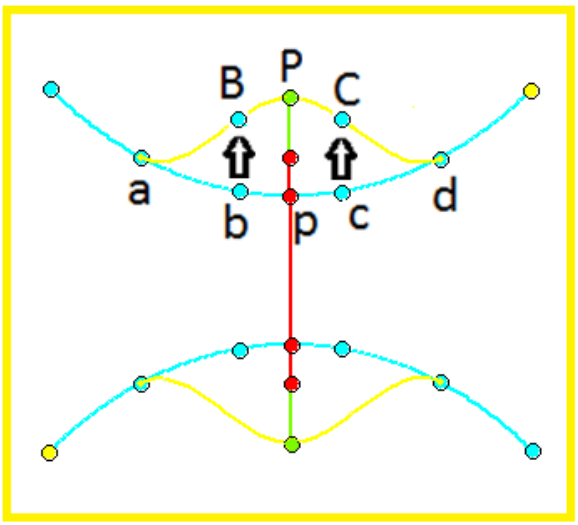

Figure A-7. Deconfliction generating Target Points.

Once Target Points *B* and *C* are calculated, the process of moving Control Points is handed off to the higher-level deformation algorithms described above.

Figure A-8 provides another look at the process of at the generating Target Points from deconflicting two trajectories. This figure uses P and P' notation, but otherwise is similar. The trajectories are suggestive of a wider range of shapes than figure A-7. Otherwise, Figures A-7 and A-8 describe similar dynamics.

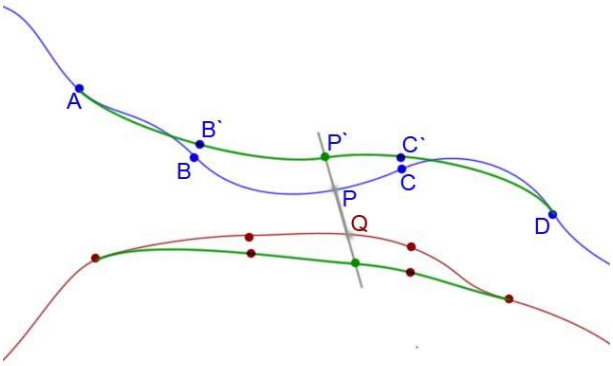

Figure A-8. Spline-based trajectory physics.

A.6.4 Postscript on Smoothing

Note that repulsion alone will tend to result in separated trajectories, yet with unseemly bumps. However, the elastic influence will tend to smooth out any isolated bumps in trajectories, yielding smoother (and generally shorter) overall paths. Figure A-9 shows the results of multiple repulsion and elastic iterations and the resulting separated and smooth trajectories. After a few

repulsion and elastic iterations of deformation, the trajectories in figure A-9 are separated, including extra green margins, and smoothed as well.

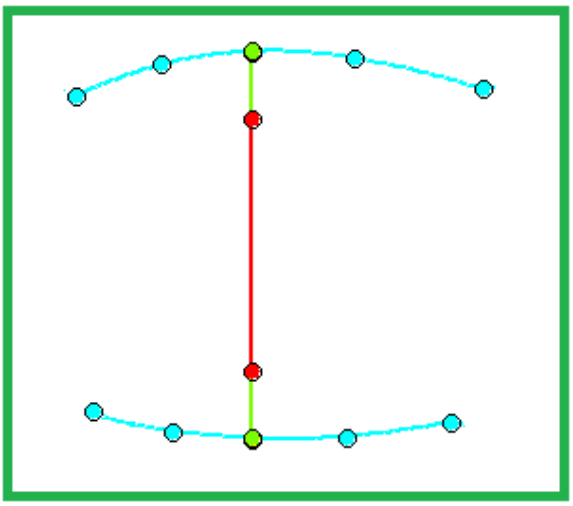

Figure A-9. Successful deconfliction and resolution.

A.6.5 Another Example

The examples of trajectory conflicts above were visually compelling. However, since the time dimension of the trajectories is not obvious, the point of closest approach at same time may not be where the trajectories appear to cross each other.

Figure A-10 shows this situation. This diagram is similar to Figure A-7, except that the trajectories appear to intersect. In fact the closest approach at the same time is where the red vertical line is shown. Nevertheless, the process of determining the Target Points is the same as before.

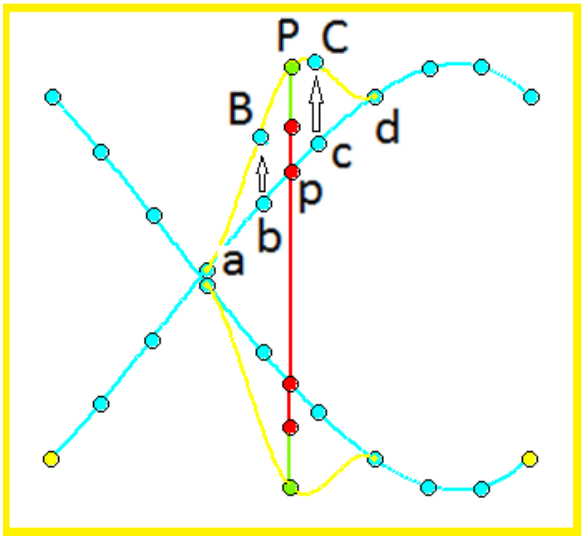

Figure A-10. Two conflicting trajectories in space-time.

A.6.6 Pseudocode for Repulsion / Separation

The following listing is the third of five fragments of pseudocode, corresponding to task 2a outlined in Section A.1.4 above as "2a. Apply repulsion/separation force to closest approach of conflicting trajectories".

The following is the pseudocode for generating Target Points to implement repulsion/separation operations, expanding and filling in the details of line 12.a.iii.1 in section A.4.8 above.

1. Begin with a pair of trajectories (or trajectory and a storm cell) that violate separation minima.
2. For each of the two trajectories (or one trajectory if the other element is a storm cell, etc.)
3. Find the point *p* of closest approach with the other trajectory (or storm cell)
4. Draw the line segment connecting the two points of closest approach of these two trajectories
5. Extend the line segment symmetrically to a distance of separation minimum plus margin
6. Point *P* is as the far end of this line segment in the direction away from the other trajectory
7. Point *b* is the nearest Control Point to point *p* in the downward time direction
8. Point *a* is the Control Point which precedes point *b*
9. Point *c* is the nearest Control Point to point *p* in the upward time direction
10. Point *d* is the Control Point which succeeds point *c*
11. Calculate the cubic splines *a*-*P* and *P*-*d*
12. Calculate point *B* by sampling *a*-*P* at time *b* (i.e. at the time corresponding to point *b*)
13. Calculate point *C* by sampling *P*-*d* at time *c*
14. Point *B* is a new Target Point for point *b*
15. Point *C* is a new Target Point for point *c*
16. Hand these two points off to the pseudocode for the high-level re-calculation algorithm above

This pseudocode continues as line 17 in section A.7.2 below.

A.7 Elasticity / Smoothing Algorithm

Implementing the influence of elasticity is much simpler than implementing repulsion. Elasticity acts on trajectories internally. In addition, this influence only acts on Control Points, and only uses neighboring Control Points for the calculation. As with all influences in these algorithms, this influence produces a Target Point.

A.7.1 Reducing Accelerations

Elasticity is accomplished by reducing accelerations at Control Points, which has the effect of smoothing trajectories. The process of reducing accelerations makes use of the theorem that maximum accelerations on a cubic spline occur at their end points. Therefore, any point sampled on a cubic spline will have an acceleration less than or equal to the accelerations at the end points.

For a Control Point b with an excessive accelerations, consider the Control Points *a* and *c* adjacent to *b*. Construct the cubic spline *a*-*c*. Then generate point *B* by sampling *a*-*c* at time *b*. Figure A-11 shows the process of applying the influence of elasticity to Control Point *b* on a trajectory. Construct the cubic spline *a*-*c*. Then generate point *B* by sampling *a*-*c* at time *b*.

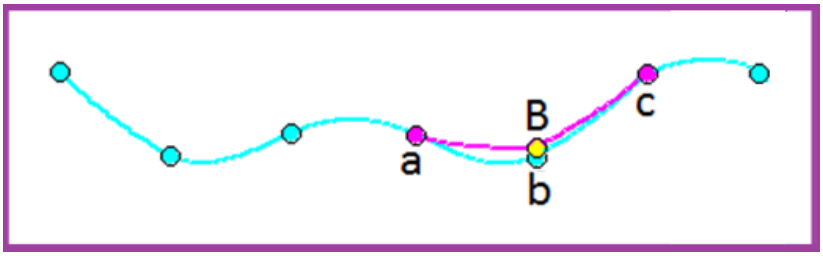

Figure A-11. Applying the influence of elasticity to Control Point.

Point *B* is a Target Point for Control Point *b*, which can be used to guide deformation of the trajectory towards point *B*, as described above in the high-level re-calculation algorithms.

A.7.2 Pseudocode for Elasticity / Smoothing

The following listing is the fourth of five fragments of pseudocode, corresponding to task 2b outlined in section A.1.4 above as "2b. Apply elasticity/smoothing force to all Control Points on all trajectories".

The following is the pseudocode for generating Target Points to implement elasticity/smoothing operations, expanding and filling in the details of line 12.a.iii.2 in section A.4.8, and continuing from line 16 of section A.6.6 above.

17. Begin with a Control Point *b* on a trajectory
18. Control Point *a* immediately precedes point *b*
19. Control Point *c* immediately succeeds point *b*
20. Construct cubic spline *a*-*c*
21. Calculate point *B* by sampling *a*-*c* at time *b* (i.e. at the time corresponding to point *b*)
22. Point *B* is a new Target Point for Control Point *b*
23. Hand point *B* off to the pseudocode for the high-level re-calculation algorithm above

This pseudocode continues as line 24 in section A.8.2 below.

A.8 Bounding / Limits Algorithm

A.8.1 Limiting Speed

There are three influences which act on trajectories in our algorithms: repulsion, elasticity, and bounding. The first two, repulsion and elasticity, deform the trajectories away from obstacles while maintaining smooth paths.

However, without bounding aircraft speed within specified limits, the repulsion and elasticity algorithms might bring an aircraft to a full stop in the sky to wait out a conflict, or speed up excessively. Without limits on speed, solving a congested airspace will always succeed simply by expanding the trajectory snarl like inflating a balloon. In this fashion, some trajectories would go far out of their way to avoid conflicts, yet still arrive on time, by flying excessively fast to do so.

The bounding influence acts on all trajectory Control Points to revise their trajectories towards the default cruising speed for the specific aircraft.

Note that possible excessive accelerations of aircraft do not need to be handled by the Bounding/Limits algorithm. Accelerations are addressed by the Elasticity/Smoothing algorithm above.

The Bounding/Limits algorithm is quite simple. For any Control Point, the default cruise speed for the aircraft (flying the trajectory) is the de facto Target Point.

A.8.2 Pseudocode for Bounding / Limits

The following listing is the fifth of five fragments of pseudocode, corresponding to task 2c outlined in section A.1.4 above as "2c. Apply bounding/limits force to all Control Points on all trajectories"

The following is the pseudocode for generating Target Points to implement Bounding/Limits operations, expanding and filling in the details of line 12.a.iii.3 in section A.4.8, and continuing from line 23 of section A.7.2 above.

24. Begin with a Control Point *p* on a trajectory
25. Construct point *P* with same values as *p*

26. Change the velocity so its new magnitude is the default speed for the trajectory's aircraft
27. Point *P* is a new Target Point for Control Point *p*
28. Hand point *B* off to the pseudocode for the high-level re-calculation algorithm above

This pseudocode continues as line 12.a.iv in section A.4.8 above.

Note that point *p* and point *P* have identical position – only the velocity may be different. The positions of the Control Point and the Target Point are same. Hence, the Bounding/Limits operation is harder to visualize.

A.9 Consolidated Pseudocode – Algorithm Toolbox

The text above describes the toolbox algorithms with pseudocode in 5 distinct tasks embedded in commentary. For those researchers who wish to reproduce this code (and the associated results), below is the same pseudocode, except stitched together into one consolidated corpus of pseudocode.

The six tasks, the first five of which are described in distinct sections above, are the following:

1. Initialize the test airspace with an initialization script
2. Perform re-calculation cycles on trajectories, where the 3 important sub-tasks are..
   a. Apply repulsion/separation force to closest approach of conflicting trajectories
   b. Apply elasticity/smoothing force to all Control Points on all trajectories
   c. Apply bounding/limits force to all Control Points on all trajectories
3. Perform post-run data analysis, successful/failed airspace, phase transition structure, etc.

Detailed pseudocode as one integrated algorithm, annotated according to the above task structure follows:

1. Initialize the test airspace with an initialization script (see section A.2.6)

1. Initialize parameters from an initialization script, including setting the following variables:
   a. Size of test airspace cylinder
   b. Number *N* of entry and exit point 'pigeon holes' for aircraft on perimeter of cylinder
   c. Range *R* of exit points on far side of cylinder from entry points
   d. Value of *DeltaFlightT*, delta time of Control Point time spacing defining and controlling trajectory shapes
   e. Value of *DeltaMetaT*, delta time for replanning (re-calculating) dynamical trajectories
   f. Specific profiles of *N* aircraft chosen to participate in this airspace simulation
2. Divide the perimeter of the cylinder into *N* equal parts
3. Choose *N* aircraft with designated default cruise altitude and speed
4. Construct *N* entry points at degree positions: $i * 360/N$
5. Construct *N* exit point at degree positions: $i * 360/N + 360/N/2$
6. For each *i-th* trajectory of these *N* trajectories.
   a. Choose an aircraft

b. Set the *i-th* entry point at position *i* * 360/*N* in degrees around the cylinder's perimeter
c. Set the velocity of the entry point to direction toward center at default aircraft speed
d. Set the *i-th* exit point at some position around the cylinder's perimeter, randomly chosen from the R exit points farthest from the entry point.
e. Set the velocity of the exit point to direction from the center at default aircraft speed
f. Construct the cubic spline from the entry point to the exit point
g. Construct a set of Control Points, one per time *t* (see Control Point delta *t* on line 4 above)
h. Sample the cubic spline on line 15 above at each time *t*, and fill in values of Control Points

7. Initial trajectories are now constructed for each wave of aircraft entering test airspace
8. Update the positions of the aircraft on the trajectories associated with each aircraft according to the time values of the points on the cubic splines. Visually this is the process of "flying" the aircraft to an updated location and heading. Refer to section above "Pseudocode: "Flying' Aircraft" in chapter 4 for more detail on this process.
9. At a designated rate, construct new waves of a set of *N* trajectories, and fly them similarly

2. Perform re-calculation cycles on trajectories (see section A.4.8)

10. Run the trajectory initialization script
11. Initialize all the Momentum Buffers to zero
12. Repeat the following until the end of the simulation
    g. Repeat until deconflicted or maximum re-calculation cycles exceeded
        i. If maximum iterations exceeded:
            1. Note separation failure
            2. Either continue, or exit depending on preferences
        ii. Collect enumeration of all pairs of conflicting trajectories
        iii. Apply Forces to Momentum Buffers (generating Target Points)

2.a. Apply repulsion/separation force to closest approach of conflicting trajectories (see section A.6.6)

1. Begin with a pair of trajectories (or trajectory and a storm cell) that violate separation minima.
2. For each of the two trajectories (or one trajectory if the other element is a storm cell, etc.)
3. Find the point p of closest approach with the other trajectory (or storm cell)
4. Draw the line segment connecting the two points of closest approach of these two trajectories
5. Extend the line segment symmetrically to a distance of separation minimum plus margin
6. Point P is as the far end of this line segment in the direction away from the other trajectory
7. Point b is the nearest Control Point to point p in the downward time direction
8. Point a is the Control Point which precedes point b
9. Point c is the nearest Control Point to point p in the upward time direction
10. Point d is the Control Point which succeeds point c

11. Calculate the cubic splines a-P and P-d
12. Calculate point B by sampling a-P at time b (i.e. at the time corresponding to point b)
13. Calculate point C by sampling P-d at time c
14. Point B is a new Target Point for point b
15. Point C is a new Target Point for point c
16. Hand these two points off to the pseudocode for the high-level re-calculation algorithm above

2.b. Apply elasticity/smoothing force to all Control Points on all trajectories (see section A.7.2)

17. Begin with a Control Point b on a trajectory
18. Control Point a immediately precedes point b
19. Control Point c immediately succeeds point b
20. Construct cubic spline a-c
21. Calculate point B by sampling a-c at time b (i.e. at the time corresponding to point b)
22. Point B is a new Target Point for Control Point b
23. Hand point B off to the pseudocode for the high-level re-calculation algorithm above

2.c. Apply bounding/limits influence to all Control Points on all trajectories (see section A.8.2)

24. Begin with a Control Point p on a trajectory
25. Construct point P with same values as p
26. Change the velocity so its new magnitude is the default speed for the trajectory's aircraft
27. Point P is a new Target Point for Control Point p
28. Hand point B off to the pseudocode for the high-level re-calculation algorithm above

2. (Continued) Perform re-calculation cycles on trajectories (see section A.4.8)

iv. Add the effect of each influence to its corresponding Momentum Buffer
v. Apply Momentum to trajectories (according to target points)
    1. Add each Momentum Buffer to its corresponding Control Point, component by component
    2. Control points will have moved (changed location and/or velocity) some (small) amount where the Momentum Buffers were non-zero
vi. Attenuate Momentum Buffers (analogous to applying friction)

h. Fly aircraft forward one simulation time step *(note: this is not one Control Point)* by adding delta-t to the Flight Time value of aircraft, and sampling each aircraft's trajectory at this new Flight Time. (See section in chapter 4 on "Pseudocode: Flying Aircraft")
i. Record measurements (density, number of conflicts, etc.)
j. Update visualization

13. The simulation of this airspace is complete, with data collected for later analysis

3. Perform post-run data analysis, successful/failed airspace, phase transition structure, etc. (No pseudocode for line 14. See chapters 5 and 6 for details on the data analysis process.)

14. Perform post-run data analysis, successful/failed airspace, phase transition structure, etc.

END

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
01- 06 - 2012	Contractor Report	

4. TITLE AND SUBTITLE

Development of Complexity Science and Technology Tools for NextGen Airspace Research and Applications

5a. CONTRACT NUMBER

NNL10AA29C

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Holmes, Bruce, J.; Sawhill, Bruce K.; Herriot, Jim; Seehart, Ken; Zellweger, Dres; Shay, Rick

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

305295.02.07.07.01

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

NASA Langley Research Center
Hampton, Virginia 23681-2199

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

National Aeronautics and Space Administration
Washington, DC 20546-0001

10. SPONSOR/MONITOR'S ACRONYM(S)

NASA

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

NASA/CR-2012-217580

12. DISTRIBUTION/AVAILABILITY STATEMENT

Unclassified - Unlimited
Subject Category 03
Availability: NASA CASI (443) 757-5802

13. SUPPLEMENTARY NOTES

This report was prepared by NextGen AeroSciences, LLC, under NASA contract NNL10AA29C with National Institute of Aerospace, Hampton, Virginia (NIA Subcontract Number X10-7082-NGA; NIA Project Manager: Fred Brooks)
Langley Technical Monitor: Natalia Alexandrov

14. ABSTRACT

The objective of this research by NextGen AeroSciences, LLC is twofold: 1) to deliver an initial "toolbox" of algorithms, agent-based structures, and method descriptions for introducing trajectory agency as a methodology for simulating and analyzing airspace states, including bulk properties of large numbers of heterogeneous 4D aircraft trajectories in a test airspace – while maintaining or increasing system safety; and 2) to use these tools in a test airspace to identify possible phase transition structure to predict when an airspace will approach the limits of its capacity. These 4D trajectories continuously replan their paths in the presence of noise and uncertainty while optimizing performance measures and performing conflict detection and resolution. In this approach, trajectories are represented as extended objects endowed with pseudopotential, maintaining time and fuel-efficient paths by bending just enough to accommodate separation while remaining inside of performance envelopes. This trajectory-centric approach differs from previous aircraft-centric distributed approaches to deconfliction.

15. SUBJECT TERMS

Air traffic conflict resolution; Aircraft trajectory optimization; Airspace; Deconfliction; NextGen; Phase state; Phase transition; Pseudopotential method; Traffic physics; Trajectories

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			STI Help Desk (email: help@sti.nasa.gov)
					19b. TELEPHONE NUMBER *(Include area code)*
U	U	U	UU	83	(443) 757-5802