

Experimental Performance of a Genetic Algorithm for Airborne Strategic Conflict Resolution

David A. Karr* Robert A. Vivona† David A. Roscoe‡

Stephen M. DePascale§

L-3 Communications, Billerica, MA 01821, USA

Maria Consiglio¶

NASA Langley Research Center, Hampton, VA, USA

The Autonomous Operations Planner, a research prototype flight-deck decision support tool to enable airborne self-separation, uses a pattern-based genetic algorithm to resolve predicted conflicts between the ownship and traffic aircraft. Conflicts are resolved by modifying the active route within the ownship's flight management system according to a predefined set of maneuver pattern templates. The performance of this pattern-based genetic algorithm was evaluated in the context of batch-mode Monte Carlo simulations running over 3600 flight hours of autonomous aircraft in en-route airspace under conditions ranging from typical current traffic densities to several times that level. Encountering over 8900 conflicts during two simulation experiments, the genetic algorithm was able to resolve all but three conflicts, while maintaining a required time of arrival constraint for most aircraft. Actual elapsed running time for the algorithm was consistent with conflict resolution in real time. The paper presents details of the genetic algorithm's design, along with mathematical models of the algorithm's performance and observations regarding the effectiveness of using complimentary maneuver patterns when multiple resolutions by the same aircraft were required.

I. Introduction

IN support of the National Aeronautics and Space Administration's (NASA's) concept of Distributed Air/Ground Traffic Management¹ and the NASA Airspace System Program research focus on functional allocation, researchers at NASA Langley Research Center have been investigating the extent to which aircraft separation responsibility can be delegated to the cockpit. In the NASA concept, properly-equipped aircraft perform autonomous flight by self-optimizing their four-dimensional trajectories while conforming to constraints such as required times of arrival (RTAs) generated by air-traffic service providers on the ground for the purpose of traffic flow management. Unlike concepts that allow autonomous flight only in segregated airspace, this concept allows autonomous aircraft to fly in airspace shared with aircraft managed by controllers or automation systems on the ground.

The cornerstone of NASA Langley's research into autonomous flight is the development of a prototype Airborne Separation Assistance System (ASAS) tool called the Autonomous Operations Planner (AOP).² The AOP's role in the cockpit is twofold: to support the flight crew in performing the new duties required for assumption of separation responsibility in the cockpit, and to support enhanced optimization of the aircraft's flight path given the flexibility enabled by autonomous flight. To successfully fulfill both of these roles, the AOP seamlessly integrates with other airborne avionics in charge of navigation, surveillance, and

*Principal Computer Scientist, member

†Chief Research Engineer, associate fellow

‡Senior Software Engineer, member

§Software Engineer, senior member

¶Senior Research Scientist, member

guidance, including the Flight Management System (FMS), the Mode Control Panel (MCP), the Automatic Dependent Surveillance-Broadcast (ADS-B) transceiver, and receivers for Flight Information System and Traffic Information System. AOP outputs are integrated into existing cockpit displays, including the Primary Flight Display (PFD), the Navigation Display (ND), and specialized AOP pages on the Multi-purpose Control and Display Unit (MCDU).

In order to enable delegated separation authority, the AOP supports airborne conflict management. Specifically, the AOP provides a variety of integrated airborne conflict detection and resolution (CD&R) capabilities. To support strategic planning (up to 20+ minutes look-ahead), the AOP provides intent-based CD&R advisories for evaluation and ultimate acceptance or rejection by the flight crew. Based on MCP and FMS data, the AOP automatically displays any conflict—defined as a predicted loss of separation (LOS) between two aircraft—on both the ND and the MCDU AOP pages. If this occurs when the aircraft’s auto-flight system is in fully engaged, FMS-based (strategic) Lateral Navigation (LNAV) and Vertical Navigation (VNAV) guidance modes, the AOP (upon request) displays a full four-dimensional resolution route on the ND that can be uploaded directly into the FMS for execution. If the aircraft is in an MCP-based (tactical) guidance mode (e.g., altitude hold and heading select), the AOP automatically displays MCP resolution advisories (e.g., altitude, vertical speed, and/or heading targets) on the PFD and ND when a LOS is predicted. The AOP also supports short-range tactical maneuvering (up to 5 minutes look-ahead) by providing state-based CD&R information and ultimately suppresses its CD&R output if Traffic Alert and Collision Avoidance System (TCAS) resolution advisories become available.

This paper focuses on the design and performance of the Pattern-Based Genetic Algorithm (PBGA), which provided strategic conflict resolution (CR) for AOP within batch-mode experiments performed at NASA Langley. PBGA creates conflict-free routes that meet time-based flow constraints (required times of arrival) and can be directly uploaded into the FMS.³ A primary focus of PBGA’s design has been to enable its use as a viable real-time CR algorithm. The paper begins with a high-level description of batch-mode simulation experiments performed by NASA that relied solely on the AOP strategic CR capability for separation assurance. Next, an overview of PBGA is provided with a description of additional enhancements developed to support the batch-mode experiments. The performance of PBGA during the batch mode simulations is then presented in terms of mathematical models of computation time and success rate as a function of traffic density. Finally, the paper makes observations regarding the complementary effectiveness of the different maneuver patterns when multiple resolutions were required for the same aircraft.

II. Batch-Mode Simulation Experiments

A. Overview

The Safety Performance of Airborne Separation (SPAS) simulation study is designed to investigate the effect of traffic demand on the safety performance of distributed air traffic control (ATC).^{4,5} In a distributed ATC environment, aircraft are responsible for traffic separation while maintaining adherence to traffic flow management constraints. The SPAS study is a series of batch-mode Monte Carlo simulation experiments designed to analyze and quantify the safety behavior of airborne self-separation. The experiments will also examine the implications of prediction errors and system uncertainties for ASAS system safety performance. To date, the first two simulation experiments have been completed.

The first experiment evaluated airborne self-separation behavior in a baseline scenario consisting of randomized routes in a generic high-density airspace in which all aircraft were constrained to the same flight level. Sustained traffic density was up to about 10 times the traffic density in a typical sector today.⁴ This scenario included limited sources of uncertainty (primarily modeling simplifications in the predicted trajectories) and was intended to develop an initial understanding of the safety performance of airborne self-separation and to establish a baseline for comparison with later experiments that add other sources of uncertainty.

The second experiment evaluated the potential impact of operator (flight crew) delay or inaction when responding to airborne self-separation systems.⁵ This experiment modified the simulated pilot behavior from the first experiment by delaying resolution actions by as much as 240 seconds when responding to detected conflicts. Additionally, a percentage of pilots, selected at random, completely ignored detected conflicts and therefore performed no conflict resolution actions. This experiment’s scenarios were similar to those used in the baseline experiment, but the maximum sustained average traffic density was increased to approximately 12 times the density in a typical sector today.

In the scenarios selected for both experiments, each autonomous aircraft's FMS active route was a straight path, generated at random near the boundary of a circle of 160 nmi radius in such a way that the route passed through a circular evaluation area of 80 nmi radius concentric with the larger circle. The evaluation area represented a generic high-density en route sector. A required time of arrival (RTA) time constraint was added to each aircraft's route downstream of the evaluation area. All flight paths were restricted to a single altitude within the evaluation area. By design, the randomly generated routes exhibited a high rate of conflicts whose lateral encounter angles varied over nearly the full range of possible values. For this study, the autoflight mode remained fully coupled to the FMS for both lateral and vertical navigation; in other words, only strategic maneuvering was allowed. In an operational setting, a tactical system would normally be available as a backup to the strategic system.

B. Simulation Environment

The SPAS simulation runs were conducted in the Air Traffic Operations Laboratory (ATOL) of the NASA Langley Research Center. ATOL is a distributed simulation platform consisting of a network of workstation-class computers with a High Level Architecture communication infrastructure. Each aircraft simulator is a real-time, medium fidelity, 6-degree-of-freedom aircraft simulation running on its own computer. The aircraft simulator includes separate software emulations of an FMS, AOP, an ADS-B datalink, and other avionics. The baseline study simulated between 14 and 72 independent aircraft simultaneously in batch mode. The second experiment increased the maximum number of aircraft to 88. When an aircraft passed beyond the evaluation area, it was replaced by a new aircraft with another randomly-generated route to maintain average traffic density. Each aircraft was "flown" by pilot model (PM) software that performed basic pilot conflict management actions. The PM was composed of a sensory input model, a rule-based decision model, and an actuator response model, enabling a range of human "personalities" to be configured to study the impact of variation in pilot behavior.

To support both experiments, AOP strategic CR was configured for lateral maneuvers only. AOP's intent-based conflict detection (CD) look-ahead was set to 10 minutes for conflict alerts and 20 minutes for conflict resolution. The experiment used the standard 5 nmi minimum separation requirement, but AOP's internal minimum separation requirement was set to 5.1 nmi and increased by 0.25 nmi during resolution as an additional buffer. AOP's capability to define trajectory prediction uncertainty bounds⁶ to avoid missed alerts that can result from inaccuracies in predicted trajectories was not utilized during these runs. Priority Rules were used within AOP to limit the number of simultaneous resolution attempts between two aircraft in conflict; the rules identified one aircraft as having priority, requiring the other aircraft to maneuver. When the time remaining until predicted first LOS was less than a predetermined threshold,⁹ either aircraft was allowed to resolve the conflict. All tactical maneuvering support within AOP was disabled for these runs, including tactical conflict detection and resolution.

AOP's strategic resolutions were required to take all known trajectory constraints into account. The main constraints for both experiments were the RTA constraints. Any AOP-generated resolution accepted by the pilot model was uploaded directly into the FMS and executed as an active route change. Upon execution, the new aircraft trajectory was broadcast to all neighboring aircraft in the form of trajectory change points using trajectory change reports within simulated Mode S ADS-B. Both experiments assumed perfect ADS-B transmission (no dropouts), and each cycle of trajectory change reports contained enough points to define at least the first 20 minutes of the aircraft's future trajectory.

III. Pattern-Based Genetic Algorithm Design

A. Overview

PBGA selects and optimizes a resolution maneuver from a set of pre-defined maneuver patterns.³ Each maneuver pattern is a template designed to execute a different type of user-acceptable path modification. For example, one pattern defines a lateral offset to be added to the FMS active route where another defines a short-cut path from one active route leg to another downstream leg. The creation of a resolution route using a pattern is then just a matter of "positioning and sizing" the pattern (i.e., determining where along the active route to start the maneuver and defining values of its geometric parameters) to avoid the conflict. PBGA

⁹The threshold was five minutes for the first experiment, and seven minutes for the second experiment.

avoids unexpected and undesired interactions between the geometric requirements of different patterns by applying only one pattern at a time,^b so each pattern's geometry can be designed without regard to the geometric requirements of the other patterns.

To determine which maneuver patterns are viable (i.e., applicable) for a given conflict scenario, PBGA defines an independent set of viability constraints for each pattern. PBGA checks these constraints against the current active route, the location of predicted LOS, and any other relevant information. Viability constraints are typically based either on geometric considerations (e.g., whether an active route leg is long enough to perform a minimum lateral offset maneuver) or on procedural limitations (e.g., the new route is not allowed to bypass an altitude, time or speed constrained waypoint), but in practice can be based on any relevant criteria.

From the set of viable maneuver patterns, PBGA creates one or more parallel populations. Each population is a collection of chromosomes (routes based on the maneuver patterns) that reproduce, mutate, and compete against each other over the course of several generations. At the end of this process, the maneuver whose fitness (a numeric value) is the smallest^c in the population is its resolution advisory, provided that this fitness is below a pre-defined threshold; otherwise the population "fails to converge" and produces no advisory.

To support PBGA's convergence to an optimum resolution maneuver, each maneuver pattern provides its own definition of an optimal maneuver. A maneuver pattern can optimize a global characteristic such as minimum path or minimum fuel usage, but patterns typically attempt to optimize some aspect of the maneuver's geometry. For example, a lateral offset maneuver can optimize the lateral offset distance from the original route, the distance traveled while on the offset path, or some combination of the two, by making its fitness a function of these parameters when the maneuver is conflict-free.

The current AOP concept for strategic CR is to provide a single lateral and a single vertical resolution maneuver for display to the flight crew. Since the SPAS experiments were limited to a single altitude, only a single lateral solution was desired and hence, only a single population was required. By placing all viable patterns of the same type (lateral or vertical) in the same population, these patterns are forced to compete with each other with only the most optimum resolution (in this case, the resolution with the smallest fitness value) returned.

B. Selection and Design of Lateral Maneuver Patterns

Three lateral maneuver patterns are currently defined and implemented in PBGA.^d For the SPAS experiments, a decision was made to let all three of PBGA's lateral maneuver patterns compete against one another for all conflicts during the experiment. Because the experiment used computer-based pilot models in a batch simulation environment, the experiment imposed no restrictions on acceptable lateral maneuvering due to pilot preference. As long as the new resolution route successfully resolved the conflict, any of the lateral maneuvers was acceptable. This decision was intended to allow the greatest possible flexibility in resolving conflicts.

Tables 1, 2, and 3 list the parameters that define the geometry for each pattern. Each parameter is a gene, that is, a degree of freedom manipulated by the genetic algorithm to find an optimal, conflict-free resolution path. The tables provide each parameter's type (discrete if the parameter is restricted to a finite set of values, continuous if the parameter can vary continuously over a range^e) and its description. A more detailed description of these patterns can be found in an earlier publication.³

Figures 1, 2, and 3 present actual screen captures of the enhanced cockpit ND in ATOL showing the patterns as used by the SPAS simulation system. The magenta line in each screen shot represents the original (conflicted) FMS active route, the white or yellow "dogbone" on this route represents the predicted conflict (first LOS to last LOS), and the blue line represents the AOP proposed strategic CR route. The white chevrons represent traffic aircraft positions (all at the same altitude as the ownship aircraft, as indicated by the symbol "00" at each aircraft's position) and the green dashed circle represents the SPAS evaluation area.

^bThe simultaneous application of multiple patterns would be achieved by designing a new "combined" pattern.

^cIn contrast, some genetic algorithms define the "best" fitness as the value that is numerically largest. In any GA, it is an arbitrary design choice whether the largest or smallest fitness is considered optimal, provided that this criterion is consistently applied throughout the algorithm.

^dA fourth pattern was previously considered³ but was not implemented.

^eA continuous parameter is considered to vary continuously from its minimum to its maximum value even if the results sometimes "snap" to values determined by route geometry.



Figure 1. Lateral Offset pattern example.

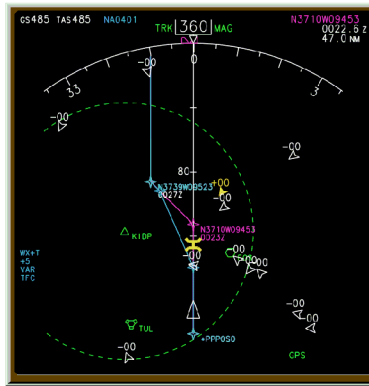


Figure 2. Direct Intercept pattern example.



Figure 3. Path Stretch pattern example.

Table 1. Lateral offset pattern parameters

Name	Type	Description
maneuver leg	discrete	Route leg on which the offset is applied
offset direction	discrete	Direction of the offset from the original route
start distance	continuous	Distance from the start of the maneuver leg at which the offset maneuver initiates
offset distance	continuous	Perpendicular distance from the original route to the offset path
offset length	continuous	Distance traveled along the offset path

Table 2. Direct intercept path pattern parameters

Name	Type	Description
maneuver leg	discrete	Route leg from which the new route leaves
intercept leg	discrete	Route leg to which the new route returns
turnout direction	discrete	Direction in which the new route leaves the original route
start distance	continuous	Distance from the start of the maneuver leg at which the turnout initiates
turnout angle	continuous	Angle at which the new route leaves the original route

Table 3. Path stretch pattern parameters

Name	Type	Description
maneuver leg	discrete	Route leg from which the new route leaves
capture waypoint	discrete	Waypoint to which the new route returns
turnout direction	discrete	Direction in which the new route leaves the original route
start distance	continuous	Distance from start of the maneuver leg at which turnout initiates
turnout angle	continuous	Angle at which the new route leaves the original route
stretch distance	continuous	Distance from turnout at which the new route turns toward the capture waypoint

All maneuver patterns were restricted to begin no closer than 15 nmi from the aircraft position at the time the resolution was generated. As can be seen from the figures, each of the three patterns uses a different geometrical approach to solve the conflict. Though each pattern is individually relatively simple and potentially restrictive in its ability to solve a wide range of problems, the three patterns, as a set, are highly complementary and provide a very flexible approach to resolving conflicts. For example, the Direct Intercept pattern is not very effective when the original route is long and straight, nor is the Path Stretch Pattern if there are no capture waypoints along this straight path, but this case is where the Lateral Offset Pattern has the most flexibility. On the other hand, if the original route has a large number of short legs between turn waypoints, the Lateral Offset pattern becomes significantly less effective, but the Direct Intercept and Path Stretch patterns are now very flexible.

Additionally, in airspace with very dense traffic, it is expected that an aircraft will perform several resolutions in succession as new conflicts arise. If only the Lateral Offset pattern were available, strategic CR could lose effectiveness as each resolution “segments” the route by replacing a long straight leg with several shorter legs between turns. If only Direct Intercept pattern were available, it would “straighten” any route until the pattern lost effectiveness. With both patterns available, the very thing that reduces the effectiveness of the Lateral Offset pattern increases the effectiveness of the Direct Intercept pattern: as one pattern adds legs to the route, the other removes them, and vice versa. The Path Stretch pattern complements the others both by adding legs (if leaving and returning to the same leg) and by removing legs (if capturing a waypoint several legs downstream from the maneuver leg). The end result is a dynamic and effective resolution capability.

C. Handling of Unmet RTA Constraints

The goal of the AOP strategic CR algorithm is to return a new route that is conflict-free with respect to all hazards (including traffic aircraft and airspace hazards) and meets all constraints (e.g., an RTA time constraint at a defined waypoint) imposed on the aircraft. In earlier versions of PBGA, a missed RTA constraint was treated similarly to a conflict. Specifically, either event would cause the fitness of a chromosome to be larger (that is, worse) than the threshold value for convergence; that is, a chromosome that missed an RTA could never be used in a resolution advisory, even if it was conflict-free.

There are occasional situations, however, in which there clearly is no conflict-free route meeting all constraints. For example, if an aircraft’s active route meets an RTA by flying straight to the constraint waypoint at the aircraft’s maximum speed, any maneuver whatsoever will necessarily fail to meet the RTA. It was decided for the SPAS experiment that in the case where no conflict-free route was found that also maintained the RTA constraint, the conflict-free route that most closely met the RTA constraint (that is, a route that minimized delay or early arrival) was to be considered optimal and should be returned as a resolution advisory.

To handle this situation, a new step was added to the fitness calculation. When a conflict-free route failed to meet an RTA constraint, the chromosome’s fitness was defined by adding a constant value, the “base RTA fitness,” to a conflict penalty that was an increasing function of the magnitude of the difference between the RTA and the ETA at the constraint point. The base RTA fitness and the scale of the conflict-penalty function were chosen so that the resulting fitness value would always be less than the “base conflict fitness” value (which was the best-case fitness value for a conflicted route) but greater than the worst-case fitness score of any conflict-free route that met the RTA (Fig. 4). PBGA defined a successful resolution as any route whose fitness value was below the base conflict fitness value; this allowed it to return a conflict-free route with an unmet RTA as a resolution advisory, but only if it found no conflict-free solution that met the RTA constraint.

D. Selection of Genetic Algorithm Parameters

In principle, a genetic algorithm can be “tuned” by setting parameters such as population size, number of generations, and mutation rate. Though no specific analysis was performed to determine the most effective values for these parameters, values selected during earlier work on PBGA were empirically shown to have acceptable performance through months of system testing that preceded the SPAS experiment. These parameter values, which were not changed during any runs in the SPAS experiment, are presented in Table 4. For more information on the use of these parameters, see the previously published description of PBGA.³

Table 4. AOP strategic CR genetic algorithm parameter settings

Parameter	Value	Description
population size	20	Number of chromosomes evaluated in each generation.
number of generations	20	Number of times the population was “evolved” through mating and mutation to find the optimal solution.
number of survivors	10	In each generation, the number of chromosomes with the best fitness scores that “survived” into the next generation. Non-surviving chromosomes were replaced by mating the survivors to produce new chromosomes.
number of elites	2	In each generation, the number of chromosomes with the highest fitness scores that were not allowed to mutate. All other chromosomes had a positive probability of mutating, whether they were survivors of the previous generation or newly produced by mating.
mutation rate	20%	In each generation, the percentage of genes (maneuver pattern parameters) from non-elite chromosomes that were mutated. Selection of the specific parameters to mutate was random.

E. Strategic CR Computational Load Design

The objective of the SPAS experiment is to investigate the effect of traffic demand on the safety performance of distributed ATC. To support this investigation, the experiment used scenarios of varying levels of traffic density, from current day levels to several times current day levels. This increased level of traffic increased the computational load on the entire AOP system, not just its strategic CR functionality

The previous description of the algorithm³ details several of the design approaches implemented within PBGA to increase computational efficiency and expedite the identification of an acceptable resolution route. One of these approaches is the dynamic reassessment of the acceptable ranges of pattern parameters. For each maneuver pattern, the parameters are quantified in the order presented in Tables 1, 2, and 3 on page 5. As each parameter is quantified, geometric constraints are imposed on the remaining parameters to eliminate values that, in the context of the values already chosen, would prevent the creation of a candidate path. This approach avoids creating a large number of invalid chromosomes in each generation of a population, as would occur if the parameters were checked for consistency only after the population was filled; this serves to increase the computational efficiency of the algorithm.^f Another approach detailed in the same reference³ is the separation of the conflicted fitness function from the pattern optimization fitness functions. By focusing solely on deconfliction until a conflict-free solution is obtained, the identification of a conflict-free route is expedited, though the achievement of an optimal conflict-free route may be delayed. Since obtaining a conflict-free route is the primary objective, with optimization a distant second, this approach also increases the success rate of PBGA. These approaches were all used without modification for the SPAS experiment.

The increased SPAS traffic levels, however, required a redesign of the way the load of the computer’s central processing unit (CPU) is balanced between the strategic CR functionality and the rest of AOP’s functionality. After PBGA is initiated, it can use 100% of the CPU resources until it completes. Since the algorithm can take several seconds to finish, this has a negative impact on other AOP functionality that may

^fThe approach of checking parameters for consistency after all values are set is valid when a genetic algorithm does not have AOP’s time constraints.

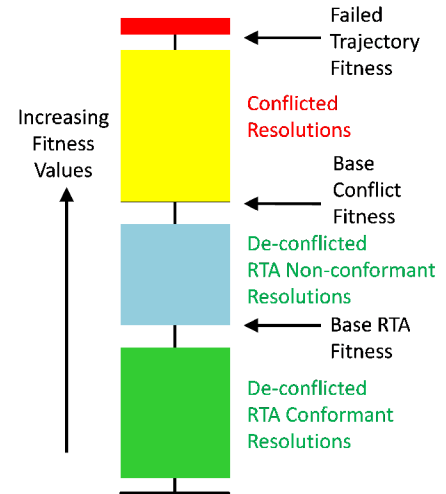


Figure 4. Fitness function approach with unmet RTA.

Table 5. Summary of SPAS simulation runs.

(a) Baseline experiment						
Sustained mean density*	Standard deviation	Peak density*	Simulation hours	Flights	Sum of flight hours	Sum of conflicts
3.45	0.59	4.97	36	881	237.27	195
6.11	0.83	8.29	36	1527	418.6	550
8.61	0.97	11.44	36	2195	544.57	1018
11.64	1.23	15.34	36	3000	797.17	1788
15.24	1.49	19.31	12	1302	347.45	963
17.18	1.54	21.39	12	1560	399.08	1256
Totals			168	10465	2744.14	5770

(b) Pilot delay experiment					
Sustained mean density*	Mean pilot delay	Sum of flight hours	Sum of conflicts	Total LOS	
11.2	3.5	240.73	583	0	
16.3	3.5	90.71	316	1	
21.4	3.5	572.76	2307	2	
Totals		904.2	3206	3	

* Units of density are aircraft per 10,000 nmi².

need to update on a one-second periodic cycle (e.g., processing of new ADS-B data, creation of new ownship trajectory predictions, and conflict detection). Since situational awareness of changes to the aircraft’s current conflict status is more important than finding a strategic solution to a previously detected conflict, a design change forced PBGA’s thread to yield the processor at strategic times within the algorithm. Though this design increased the total elapsed time of PBGA, it provided the proper balance of computational load to support the entire range of AOP functionality required for SPAS.

IV. Safety Performance of Airborne Separation Experiment Results

The baseline study consisted of six sets of simulation runs, in which each run performed six hours of continuous simulation at a sustained traffic density level and each set consisted of six independent runs all with the same parameters (except for the two highest density sets, which had two runs each). Table 5(a) (reproduced from Consiglio et al. (2007)⁴) presents the key statistics for the six run sets.

The mean density numbers were selected to represent approximate multiples of current day traffic demand levels. In the source of these data,⁴ 1X traffic levels for current-day operations were estimated at densities of 1.8 per 10,000 nmi² for a median-density en-route sector^g and 8.45 per 10,000 nmi² for a high-density en-route sector. The traffic levels simulated therefore represent a range of density from 2X to nearly 10X current median en-route sector traffic density.

The mean number of traffic conflicts generated within the test region increased from 0.82 conflicts per flight hour at a density of 3.45 aircraft per 10,000 nmi² to 3.15 conflicts per flight hour at a density of 17.18 aircraft per 10,000 nmi². There were only three cases of LOS, with points of closest approach at 4.986, 4.989, and 4.999 nmi respectively. These three cases did not involve a failure of conflict resolution, but were the result of missed alerts due to an approximation used in turn modeling within the trajectory predictor. These missed alerts would have been prevented by trajectory prediction uncertainty bounds if the bounds had been utilized during these runs.

^gReference median-density sector ZOA31 (Cleveland ARTCC) for Flight Level 310 on 19 Feb 2004.

For the pilot delay experiment, Table 5(b) (reproduced from Consiglio et al. (2008)⁵) illustrates the results of an additional 904.2 flight hours and 3206 conflicts. Using the same estimate of 1X sector density as the baseline experiment, these runs represent 5X to 12X current-day median en-route sector traffic density. These runs all used a pilot delay value of 3.5 seconds. All three LOS events were caused by failures of PBGA to return a conflict-free solution for either aircraft in the conflict pair. Each of these events involved a highly complex multiple-aircraft situation. In the two-LOS case at density 21.4, a four-aircraft conflict situation caused one aircraft to lose separation with two of the other aircraft.

V. Pattern-Based Genetic Algorithm Performance

This section presents an analysis of the performance of PBGA according to various metrics applied to data obtained from AOP during the SPAS baseline study. These data include an observation for every CR attempt (call to PBGA) that occurred during any simulation, including CR attempts that were later aborted or canceled by the pilot model or by external events such as an update of ownship intent. The dataset also included an observation each time PBGA was not called because the time to loss of separation was less than one minute. While all these events contributed to the outcomes of the SPAS experiment, they were excluded from the following analyses, which are intended to represent the performance of PBGA when it is allowed to run to completion. In this study, the remaining set of data is called the SPAS dataset.

A. Elapsed Time of Conflict Resolution

The elapsed time of conflict resolution, T_{CR} , is the amount of simulated time that passed between the instant when AOP issued a request to execute PBGA and the instant when AOP received the result of this computation from PBGA. Since all simulation occurred in real time, this measurement is a good approximation of the real (wall-clock) elapsed time.

This study used the SPAS dataset to estimate T_{CR} . The start and end times of PBGA were recorded with a precision of one second, so the elapsed time of each CR attempt was measured in whole seconds. Since the objective was to estimate T_{CR} for an arbitrarily selected conflict, if several CR attempts were made on the same unique conflict (that is, with similar input each time, except that aircraft positions were advanced a few seconds), the first value of T_{CR} in the sequence was selected as the most representative value and the others were discarded. This resulted in a set of 6944 observed values of T_{CR} , including 6699 cases in which PBGA returned a conflict-free route and 245 cases in which CR failed to converge.

1. Hypothetical Model of Performance

Informal observations of processor load during simulations indicate that the bulk of AOP processing during execution of PBGA is spent on conflict probing and evaluation of the conflicted fitness function.^h The conflict probe, performed once on each chromosome that is evaluated, consists of independent calculations comparing the trajectory of the ownship with the trajectory of each of the traffic aircraft considered by the probe.ⁱ Let N_{AC} be the number of traffic aircraft to be considered. Then the conflicted fitness function, executed once on each conflicted chromosome, consists mainly of N_{AC} calculations to find the region of airspace (if any) in which each traffic aircraft is likely to conflict with possible ownship trajectories. The expected total number of chromosomes evaluated during PBGA, however, can be considered a constant.^j All processing outside of PBGA is periodic and can be assumed to increase elapsed time by a constant factor not much greater than unity. These facts suggest that the principal trend will be for the total elapsed time of conflict resolution, T_{CR} , to be a linear function of N_{AC} .

As N_{AC} increases, however, a conflict-free route may be harder to find. This suggests that a higher percentage of chromosomes may have conflicts and may execute the conflicted fitness function, causing T_{CR} to grow at more than a linear rate. It is reasonable to model this accelerated growth rate by a term that is quadratic in N_{AC} . A hypothetical predictor of elapsed time, \hat{T}_{CR} , then has the general form:

$$\hat{T}_{CR} = \beta_0 + \beta_1 N_{AC} + \beta_2 N_{AC}^2. \quad (1)$$

^hThe principal contributors to PBGA processing time were identified by the Rational Quantify performance profiler.

ⁱIn the experiment, the conflict probe considered all traffic currently being simulated at the time CR was attempted. Conflicts detected outside the evaluation region were ignored during later stages of computation.

^jThe only variation in the number of chromosomes evaluated was due to the random selection of chromosomes to mutate.

Because the input values are limited by the traffic densities that might occur in en-route airspace within the next few decades, this predictor is not intended to address the asymptotic behavior of PBGA. While it is a reasonable guess that any trends in the existing data will continue to some extent, caution is needed when applying any results of this study much beyond the range studied in SPAS.

2. Initial Exploration of the Data

The observed values of N_{AC} ranged from a minimum of 3 to a maximum of 70 among all 6944 unique CR requests in the SPAS dataset. The number of traffic in CR requests was clustered around six distinct modal values, as might be expected since the simulations were run on scenarios of six different sizes. The maximum frequency of any value of N_{AC} was 404 CR requests for $N_{AC} = 41$. Fig. 5 shows the combined distribution of N_{AC} and T_{CR} among all CR requests.

Judging from Fig. 5, the mean value of T_{CR} for any given value of N_{AC} appears to lie along a line connecting the six peak frequencies of observation. Assuming that the remaining variation in T_{CR} fell in a normal (that is, Gaussian) probability distribution whose variance was independent of N_{AC} , one could then perform a linear regression by the method of least squares to estimate the parameters of (1), treating N_{AC} and N_{AC}^2 as independent predictors of the observed variable T_{CR} . It is clear from Fig. 5, however, that the variance of T_{CR} is much greater for larger values of N_{AC} than for smaller values. Moreover, T_{CR} does not appear to have a normal probability distribution. For example, consider the values of T_{CR} observed at $N_{AC} = 41$. Fig. 6 compares the observed values against a “best fit” normal distribution determined by the `normfit` function in MATLAB version 7.7.

Fig. 6 illustrates at least three problems. First, there are outlier observations at both ends of the range. Second, the remainder of the distribution (excluding the outliers) is skewed, with a larger tail on the right than on the left.^k Third, since the elapsed times in this experiment were measured in whole seconds, all the observed values fell in a discrete distribution with only 11 distinct values. It should be noted, however, that except for a small number of outliers on the left (about 1% of the data in this case), no value differs by more than two or three seconds from the values predicted by a normal distribution. Our concern therefore is with the validity of applying the method of least squares in this case, not with the general notion of fitting a model to these data.

3. Estimation of Parameters

Rather than linear regression by the method of ordinary least squares, estimation of \hat{T}_{CR} used a robust method of linear regression, specifically the `robustfit` function of MATLAB version 7.7 with the default settings. Conveniently expressed in terms of $N_{AC}/100$, the result is

$$\hat{T}_{CR} = 2.381 + 11.698 \cdot \frac{N_{AC}}{100} + 7.315 \cdot \left(\frac{N_{AC}^2}{100} \right). \quad (2)$$

This estimate of T_{CR} ranges from 2.74 to 14.15 seconds over the 3 to 70 traffic aircraft observed in CR requests in SPAS.

^kThe causes of these two problems are identified in Subsection 4 on the next page.

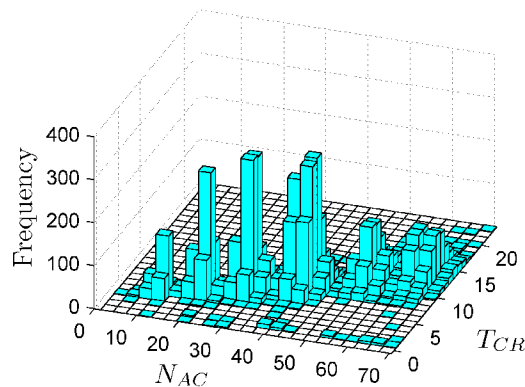


Figure 5. Unique CR requests for N_{AC} traffic aircraft completed in T_{CR} seconds elapsed time.

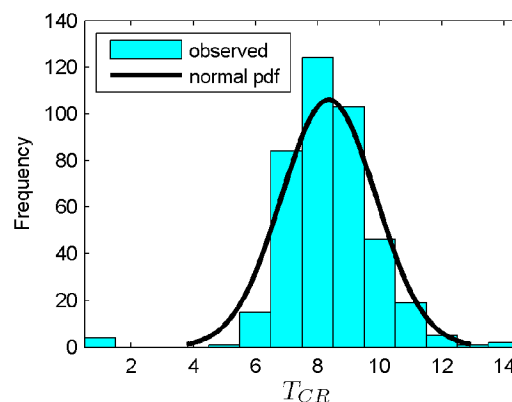


Figure 6. Elapsed CR time in seconds for 41 traffic aircraft.

Fig. 7 shows this model superimposed on the range of T_{CR} at each N_{AC} for which there were at least two observations. The mean value of T_{CR} follows the model closely; of course, the means of very small samples tended to deviate further from the estimated value. The extreme values plotted at each N_{AC} are the minimum and maximum values of T_{CR} observed over all first CR attempts with N_{AC} traffic aircraft; the distance of each of these values from the mean increases not only with the variance of T_{CR} but also with the size of the sample, so it is not surprising to see large variations in the range of these values. The 16th and 84th percentiles are arguably a better representation of the variability of the data; these percentiles were selected because they are the percentile ranks at one standard deviation below and above the mean of a normal distribution. A linear model (not shown here), computed by similar methods, did not fit the data well.

To estimate the standard deviation of T_{CR} , which is clearly increasing with N_{AC} , the sample standard deviation of T_{CR} for the set of all CR attempts on N_{AC} traffic aircraft,

$$\sqrt{\frac{\sum_{i=1}^N (T_i - \hat{\mu})^2}{(N-1)}}, \quad (3)$$

was computed at each value of N_{AC} for which $N \geq 2$, where N was the number of CR attempts, T_i the elapsed time of the i th CR attempt, and $\hat{\mu}$ the observed mean elapsed time at that value of N_{AC} . Since N actually ranged from 2 to 404, the expected error of this estimator was small for some values of N_{AC} but large for others. Using robust linear regression, a quadratic function of N_{AC} appeared to fit the data better than a linear function. The estimated standard deviation of T_{CR} was

$$\hat{\sigma} = 0.349 + 1.700 \cdot \frac{N_{AC}}{100} + 1.756 \cdot \left(\frac{N_{AC}^2}{100}\right). \quad (4)$$

Fig. 8 compares this model to the sample standard deviation of the subset of CR attempts at each value of N_{AC} . Not surprisingly, the standard deviations of small samples often had large deviations from the prediction, but larger samples tended to behave as predicted.

Other methods to model the mean and standard deviation of T_{CR} , including ordinary least-squares linear regression and least-squares linear regression scaled for more uniform variance, produced models that agree with the robust model to within a fraction of a second over the applicable range.

4. Search for Alternative Models

There may be other factors that influence the elapsed time of CR. For example, when all patterns were found non-viable, PBGA completed unusually quickly, whereas in other cases of failure to converge and cases when the RTA was not met, PBGA often completed unusually slowly.

Failure to converge and unmet RTA seem to account for most of the outliers and some of the skew of the distribution of T_{CR} . The remaining variability in the data was investigated within the subset of the SPAS dataset consisting only of CR attempts that returned a conflict-free route that met the ownship's RTA. This subset also excluded eight CR attempts for which some information was missing. The remaining data comprised 6503 observations in which the errors in T_{CR} can reasonably be assumed to be normally distributed, although not with uniform variance. A multivariate linear regression by the method of least squares on 11 predictor variables selected from among the possible candidates in the SPAS dataset (eliminating variables that were obviously redundant) resulted in estimated regression coefficients for these variables, as well as bounds of the 95% confidence intervals of the coefficients.

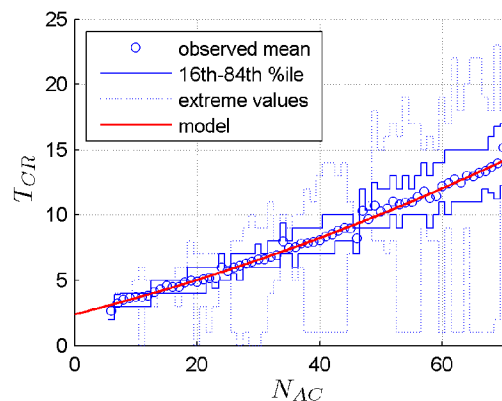


Figure 7. Elapsed CR time (seconds) vs. number of traffic aircraft.

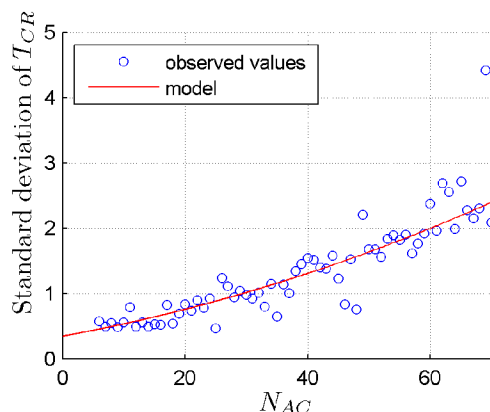


Figure 8. Standard deviation of elapsed CR time (seconds) vs. number of traffic aircraft.

Table 6. Coefficients of reduced multivariate regression to predict elapsed time of CR.

Term in regression equation	Regression coefficient		
	Estimated	Lower bound	Upper bound
Constant	-4.412	-4.712	-4.113
Traffic aircraft	0.178	0.176	0.180
Total waypoints	0.506	0.480	0.533
Fitness	0.015	0.014	0.016

This regression was iterated; after each iteration, the variables corresponding to coefficients that were statistically insignificant (that is, whose 95% confidence intervals included the value zero) were eliminated. If there were no such variables, the variable that explained the least amount of variation in T_{CR} was eliminated. (The product of each variable’s regression coefficient and the difference between its least and greatest observed values was deemed to be the amount of variation explained, that is, the estimated number of seconds this variable might add or subtract from T_{CR} .) In the fifth regression there were only three variables: number of traffic aircraft (N_{AC}), total number of waypoints in the resolution route (ranging from 7 to 15), and fitness of the winning chromosome (ranging from 8.0 to 255.7). The coefficients of this regression (including the estimated constant term in (1)), and the bounds of their 95% confidence intervals, are shown in Table 6. Although the regression coefficient of total number of waypoints was greater than that of the other two variables, N_{AC} had a much larger range of values (from 3 to 70) and as a result explained approximately three times as much of the variation in T_{CR} as either of the other two variables.

The number of waypoints in the conflicted route (prior to PBGA) was not considered in this regression, since it was known for only 1622 CR attempts. When the regression was restricted to these CR attempts and this variable was included along with the three variables in Table 6, it explained by far the least amount of variation in T_{CR} . (In fact the coefficient of N_{AC} increased slightly.) Even when the only other variable was N_{AC} , the coefficient of active waypoints was only 0.202. It does not appear worthwhile to use this much smaller dataset so that this variable could be included in other calculations. On the other hand, total waypoints and fitness of the winning chromosome are not known until after PBGA finishes. Number of traffic aircraft is therefore the only really useful predictor of elapsed time based on this dataset.

B. Probability of Failure to Converge

The probability of failure to converge is defined as $p_{FC} = N_{FC}/(N_R + N_{FC})$, where N_{FC} is the number of CR attempts that failed to converge and N_R is the number that returned a conflict-free route.

In order for p_{FC} to represent the probability of solving an arbitrarily chosen conflict, only the first CR attempt for each unique conflict was considered in this calculation. This is a conservative method, and it is representative of the outcome of an arbitrary CR attempt.

Since CR is expected to become “harder” as traffic density increases, one can hypothesize that p_{FC} increases with the number of traffic aircraft N_{AC} . Fig. 9 shows the values of p_{FC} computed by aggregating CR attempts at each value of N_{AC} for which there were at least 64 observations. (Due to the low failure rates, groups of much fewer than 64 observations cannot estimate the probabilities with sufficient precision.)

There is clearly a trend toward larger values of p_{FC} for larger values of N_{AC} , as one might expect (since CR is “harder” in denser airspaces). Moreover, there appears to be an acceleration of the trend at larger values of N_{AC} . This suggests a model that is quadratic in N_{AC} . Fitting this model to the data in Fig. 9 using linear regression by the method of least squares, the resulting predicted relationship is

$$100 \cdot \hat{p}_{FC} = 1.01 - 2.02 \cdot \frac{N_{AC}}{100} + 16.97 \cdot \left(\frac{N_{AC}^2}{100} \right). \quad (5)$$

Despite the large number of first CR attempts observed, none of the coefficients of (5) is statistically significant at the 95% level. Nevertheless, this prediction matches the observed results reasonably well when the data are subdivided into three clusters of more than 2000 observations each, as shown in Table 7. A robust method yielded similar regression coefficients, but did not result in as close a fit to the data in this table. In any case, the probability of failure to converge is clearly low even when CR is not re-tried.

Table 7. Predicted and actual failure rates.

Range of N_{AC}	Number of observations	Number of failures	
		<i>Predicted</i>	<i>Actual</i>
3 to 34	2334	37.26	32
35 to 46	2091	63.17	65
47 to 70	2519	148.08	148
Totals	6944	248.51	245

C. Evolution of Winning Patterns

In 1597 cases within the SPAS dataset, the same flight executed two or more conflict resolutions in succession. Due to the complementary features of the three available patterns, one might expect to find that the relative probabilities of different winning patterns would change as a flight executed a second, third, or fourth resolution. In fact, this was observed. Since the flights generally started with similar routes (primarily long, straight paths), it is reasonable to aggregate all first resolutions, all second resolutions, and so forth. Since only two flights executed more than four resolutions each during the experiment, the data illustrate only a limited progression, as shown in Table 8. Moreover, the relative frequencies of winning patterns in the fourth resolution should be considered uncertain due to the small number of cases. Nevertheless, the data appear to confirm the expected trend toward more frequent application of the Direct Intercept pattern after the Offset and Path Stretch patterns have added turn waypoints to the route.

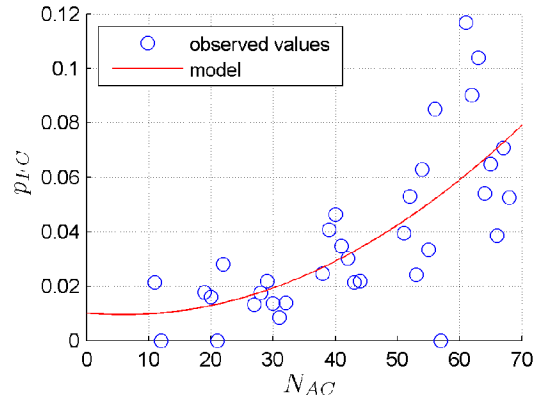


Figure 9. Probability of failure vs. number of traffic aircraft.

VI. Ongoing Work

The SPAS dataset is likely to yield a very conservative estimate of the speed of PBGA. According to informal observations on comparable (but more recently acquired) equipment, PBGA elapsed time has since improved significantly due to processor speed alone. These observations also indicate that the elapsed time is sensitive to the compiler options under which the software was built. SPAS used a “debug” build of AOP (one that was linked with debug information), which made the execution of AOP consistent with debug

Table 8. Relative frequencies of patterns in sequential resolutions by the same flight.

CR number	Number of cases	Relative frequency (percent)					
		<i>Viable patterns</i>			<i>Winning Patterns</i>		
		<i>Direct Intercept</i>	<i>Offset</i>	<i>Path Stretch</i>	<i>Direct Intercept</i>	<i>Offset</i>	<i>Path Stretch</i>
1	4955	20.20	98.00	99.84	0.02	67.23	32.75
2	1597	78.40	74.39	94.24	16.47	37.82	45.71
3	260	89.62	75.00	91.15	26.92	33.85	39.23
4	37	78.38	78.38	91.89	29.73	37.84	32.43

runs performed by developers, but slower than an optimized build. In a small number of controlled test runs comparing a “debug” build of AOP with one that was built in “release” mode, the elapsed time for an example with 11 aircraft was reduced by 51 percent, and reductions in elapsed time for 48 to 51 aircraft were in the range of 61 to 65 percent. The “release” build of AOP has the same interactions with other software subsystems as the “debug” build (except that its responses are more prompt), and produces all of the same output data files and log files for postprocessing, so no useful information is lost; the added information in the “debug” build is currently used only when a software developer has opened a run of AOP in a debugger window, not in any environment in which AOP can produce useful results. Future experiments involving AOP are likely to see significant improvements in PBGA performance due simply to the use of compiler optimizations in the executable code. In particular, large-scale experiments currently under development will use copies of AOP built in the “release” mode. We hypothesize that these experiments will demonstrate a substantial decrease in the mean elapsed time of PBGA relative to comparable scenarios observed in the SPAS experiment.

In the new experiments mentioned above, aircraft will climb and descend as well as fly level, and PBGA will be permitted to propose both lateral and vertical maneuvers. This will require AOP to track a larger number of aircraft at any given airspace density, since it must consider aircraft at multiple flight levels. On the other hand, the average cost per traffic aircraft is likely to decrease, since it typically takes far fewer operations to rule out conflicts with an aircraft at a different level than it does for an aircraft at the same flight level as the ownship. At the same time, the vertical degree of freedom may enable resolution of some conflicts that could not be resolved by lateral patterns.

At this point no formal studies have been performed to determine the most efficient patterns and parameters for PBGA. The algorithm often finds conflict-free routes almost immediately and spends the rest of its time optimizing these routes. On the other hand, the patterns in use were purposely limited. For example, on straight portions of a route, the existing patterns permit significant departure from this path on only one side. This prevents lateral resolution in some cases where a pattern such as an S turn might be conflict-free. It is possible that a different choice of patterns and GA parameters would make CR possible in less elapsed time with a lower rate of failure on each attempt.

While the number of aircraft in the SPAS experiments was limited by the size of the evaluation area at any given traffic density, it is possible that future applications of AOP may receive information about all traffic aircraft in a much larger airspace. Fortunately, AOP uses less processing to probe distant trajectories than to probe nearer ones. It may be necessary, however, to employ additional known techniques that can efficiently eliminate many aircraft from consideration before PBGA is started.

VII. Concluding Remarks

The pattern-based genetic algorithm that is the foundation of the AOP strategic CR capability successfully resolved all conflicts experienced during the SPAS baseline experiment and all but three highly complex conflicts in the SPAS pilot delay experiment. Though the number of error sources was kept to a minimum in these runs, the fact that AOP’s strategic CR capability was able to resolve these conflicts without the need for either vertical or tactical maneuvering, even when traffic densities were increased to 10X–12X current day traffic levels, is a significant result. PBGA’s current performance is adequate for follow-on batch-mode experiments that will explore and identify specific error sources (such as wind errors) and operational conditions under which vertical and tactical maneuvering become necessary. In addition, even without expected improvements this is a promising baseline for future human-in-the-loop experiments.

Acknowledgment

This effort was conducted for NASA’s Langley Research Center under subcontract to Raytheon Corporation. The authors would like to thank Sherwood Hoadley of NASA Langley for her help in collecting the analysis data and David Wing of NASA Langley for his many inputs and feedback on AOP functionality. The authors would also like to thank the many Raytheon Team staff members who developed the other components of NASA Langley’s ATOL simulation capability, without which this paper would not have been possible.

References

- ¹Green, S. M., Bilimoria, K., and Ballin, M. G., "Distributed air/ground traffic management for en route operations," *Air Traffic Control Quarterly*, Vol. 9, No. 4, 2001.
- ²Ballin, M., Sharma, V., Vivona, R., Johnson, E., and Ramiscal, E., "A flight deck decision support tool for autonomous airborne operations," *Proceedings of the AIAA Conference on Guidance, Navigation, and Control*, AIAA, Monterey, CA, USA, 2002.
- ³Vivona, R., Karr, D., and Roscoe, D., "Pattern-based genetic algorithm for airborne conflict resolution," *AIAA Guidance, Navigation and Control Conference*, AIAA, Keystone, CO, USA, 2006.
- ⁴Consiglio, M., Hoadley, S., Wing, D., and Baxley, B., "Safety performance of airborne separation: preliminary baseline testing," *Seventh AIAA Aviation Technology, Integration and Operations (ATIO) Conference*, AIAA, Hilton Head, SC, USA, 2007.
- ⁵Consiglio, M., Hoadley, S., Wing, D., Baxley, B., and Allen, D., "Impact of pilot delay and non-responsiveness on the safety performance of airborne separation," *Eighth AIAA Aviation Technology, Integration and Operations (ATIO) Conference*, AIAA, Honolulu, HI, USA, 2008.
- ⁶Karr, D. and Vivona, R., "Conflict detection using variable four-dimensional uncertainty bounds to control missed alerts," *AIAA Guidance, Navigation and Control Conference*, AIAA, Keystone, CO, USA, 2006.