

Automated Conflict Resolution: A Simulation Evaluation Under High Demand Including Merging Arrivals

Todd Farley*

NASA Ames Research Center, Moffett Field, California 94035-1000

Michael Kupfer† and Heinz Erzberger‡

University of California at Santa Cruz, Moffett Field, California 94035-1000

Performance of an automated conflict resolution algorithm is assessed. Evaluations are based on fast-time simulations of 24 hours of nominal and heavily increased air traffic demand in the Cleveland Air Route Traffic Control Center airspace. Departure, en-route, and arrival traffic were represented, based on recorded Host data. The algorithm is designed to resolve detected conflicts—projected to be between two and twenty minutes from first loss of separation—by amending the lateral, vertical, or longitudinal trajectory of one aircraft such that the conflict is resolved, no new conflicts are created, and the aircraft is returned to its original flight plan. The results indicate that the conflict resolution algorithm is capable of resolving conflicts safely and efficiently at traffic levels significantly higher than today under nominal conditions. Safety and efficiency metrics are offered as benchmarks for comparison with alternative algorithms.

I. Introduction

CONVENTIONAL methods for assuring safe separation[§] of aircraft from proximate traffic are not scalable to accommodate the level of air traffic demand that is projected for the years 2020 and beyond.¹ To meet the demand forecasts, air traffic management (ATM) system designers in Europe and the US are developing plans for next-generation ATM systems that rely on a separation assurance capability that is more scalable than that of today's system. This future capability is expected to rely on advanced automation to detect conflicts and to resolve them.² The design standards for an *automated separation assurance* function call for safe and efficient separation of air traffic at two- to three-times recent levels to relieve the constraints that conventional, human-based separation assurance methods present as traffic levels increase. Accordingly, the research community is investigating conflict detection and conflict resolution algorithms that will meet the two- to three-fold traffic-increase design standard. This paper focuses specifically on the conflict resolution problem.

Many candidate algorithms for conflict resolution have been proposed. Kuchar and Yang cite 47 of them in a broad survey published in 2000.³ Many of the cited investigations remain active, and the literature has expanded to include approaches based on genetic algorithms (for example, Ref. 4 and 5) and the incorporation of arrival-time constraints (such as Ref. 6 and 7). Most proposed solutions have been exercised only through a set of scripted, controlled or simplified scenarios. Such exercises are a necessary step in the validation process. However, to validate a conflict resolution algorithm with respect to the "2x-to-3x" design standard, a candidate algorithm must be stressed by traffic volumes, densities, and complexities that are commensurate with today's busiest airspace as well as that of the envisioned future. Further, it is necessary to expose the algorithm to the full breadth and variety of conflict situations that occur in real-world operations now or in the foreseeable future. Farley & Erzberger conducted a simulation-based evaluation of a conflict resolution algorithm under one, two and three times present-day demand levels for Cleveland Center.⁸ However, that study did not include results for arrival merging operations. Arrival merging operations pose greater challenges for conflict resolution, because the arrival fix presents a routing

* Research Engineer, Aviation Systems Division, M/S 210-6, *todd.farley@nasa.gov*, AIAA Associate Fellow.

† Research Engineer, University-Affiliated Research Center, M/S 210-6.

‡ Adjunct Professor, Department of Electrical Engineering, P.O. Box 7, AIAA Fellow.

§ Two aircraft are said to have *safe separation* if they are separated vertically by at least 1000 ft or they are separated horizontally by at least 5 nmi.

constraint that obligates the resolution trajectories for both aircraft to cross a common point in close succession in a desired sequence.

In this study, performance of a candidate conflict resolution algorithm is assessed for conflicts detected in any phase of flight—including arrival merging operations—in a simulation environment designed to represent the complexity, variety and volume of current and future air traffic operations. The study probes a single candidate algorithm: the conflict resolution algorithm developed as part of the Automated Airspace Concept.^{9,10} Its results are intended to provide objective benchmarks by which ATM researchers may evaluate competing algorithms or approaches to the conflict-resolution problem. It is hoped that this work may stimulate the introduction of additional benchmarks generic enough to facilitate performance comparisons between alternative approaches. In so doing, it will help determine the potential of the present algorithm for further development.

The body of this report is organized as follows. The background section provides a brief overview of the conflict resolution algorithm evaluated. The experimental approach and procedure are presented to identify the underlying simplifications and assumptions. The results are presented, categorized according to safety- and efficiency-related metrics. Lastly, a summary of the study's findings is given, along with suggestions for future research.

II. Background

The conflict resolution algorithm is an extension of prior work by Erzberger (most recently, Ref. 11). Adopting the Kuchar & Yang taxonomy, the algorithm may be categorized as a “pairwise,” “optimized” approach, with resolution trajectories comprised of turn maneuvers, vertical maneuvers, or speed changes.^{**} Situations involving multiple potential conflicts are resolved sequentially in pairs. *Compound maneuvers*—those comprised of two or more basic maneuvers (e.g., a turn, a climb, or a speed change) in succession or in combination, are not considered. Similarly, *cooperative maneuvers*, which require both conflicting aircraft to maneuver simultaneously in coordinated fashion, are not considered. Optimization is based on a simple cost function, where induced delay is the driving term. The optimal resolution trajectory is selected from a pool of up to three acceptable resolution trajectories (one in each axis: lateral, vertical, and longitudinal) for each aircraft in conflict, where an “acceptable” resolution trajectory is one that (a) is projected to resolve the detected conflict(s), (b) is projected to induce no new conflicts, and (c) returns the aircraft to a point on its original flight plan, downstream of the detected conflict(s) within a given time horizon (typically 20 min). Developing and evaluating this pool of up to six candidate trajectories (3 for each aircraft) represents a significant difference from the version of the algorithm used in the prior work, where the first acceptable trajectory computed was executed. This design change is expected to result in more efficient resolution trajectories. The pool of acceptable resolution trajectories for each aircraft is generated by sequentially computing and evaluating candidate trajectories (*trial-plan trajectories*) in a structured progression until a trajectory is found that satisfies all of the resolution conditions cited above. The progression is tailored according to the dynamic and physical characteristics of the conflict; for example, a different progression is used for an overtake conflict than for a climb-cruise conflict.

Fault recovery and safety assurance are important aspects of algorithm design and system design for automated separation assurance. If no acceptable resolution trajectory is found for a projected conflict, the conflict resolution algorithm employs fault-recovery strategies to develop an acceptable resolution trajectory. The fault-recovery strategies are discussed in Ref. 8. That paper also includes two case studies which illustrate the operation of the algorithm. If fault recovery is unsuccessful, then the conflict must be delegated to a separate *conflict avoidance* algorithm. Conflict avoidance algorithms sacrifice some operational efficiency in favor of expediency in resolving imminent losses of separation. Specifically, they differ from conflict resolution algorithms by their expanded maneuvering envelope (e.g., maximum allowable bank angle, climb rate, etc.) and by their simpler, fail-safe architectures.¹² Along with *collision avoidance* systems such as TCAS (Traffic Alert and Collision Avoidance System), they represent an important safety net in an overall separation assurance architecture.

III. Approach and Procedure

To benchmark the conflict resolution algorithm, a simulation-based test plan was developed that would stress the conflict resolution algorithm with traffic densities ranging from today's busiest airspace to that of the envisioned

^{**} The framework categorizes conflict resolution algorithms along three axes: scope (i.e., pairwise vs. global); strategy (i.e., prescribed, optimized, force field, or manual); and available maneuvers (e.g., turns, vertical maneuvers, speed changes, or combined maneuvers). See Ref. 3 for definitions.

future. The traffic represented a breadth and variety of conflict situations characteristic of real-world operations today and for the foreseeable future.

The algorithm was evaluated in a fast-time air traffic simulation environment. The Cleveland Air Route Traffic Control Center (ARTCC) airspace—the airspace with the most daily operations in the U.S. National Airspace System (NAS)—was chosen for the experiment. Cleveland Center airspace and flight operations were simulated using the Airspace Concept Evaluation System (ACES).¹³ The simulation included all types of Cleveland Center air traffic: departures, overflights, and arrivals; air carrier, business, and general aviation. Flight operations over a clear-weather, 24-hour period were simulated based on Cleveland Center’s radar and flight plan data, as recorded on April 21, 2005. These data were provided by the FAA. Extrapolated demand sets representing 50% and 100% increases above the nominal demand level were generated according to the methodology established by Paglione.¹⁴

The conflict resolution algorithm was evaluated in three separate simulations of increasing traffic volume. Each simulation’s demand set (i.e., nominal traffic (“1x”), a 50% increase in traffic (“1.5x”), and double the traffic (“2x”)) provided 24 continuous hours of simulated air traffic transitioning through Cleveland Center airspace. A single instantiation of the resolution algorithm was responsible for resolving all conflicts across the Center airspace; there was no subdivision of the conflicts or traffic into sectors, and there was no human support. The simulation was executed in fast time, requiring between four and twelve hours to execute each 24-hr scenario, depending on the traffic volume. Dependent variables included the number of conflicts detected, the number of conflicts resolved, and the delay imposed with each resolution trajectory.

The conflict resolution process was set to activate 8 min prior to each predicted loss of separation for most conflict categories; for conflicts between merging arrivals—where early action can significantly improve the likelihood of finding a resolution trajectory—it was 12 min. Sensitivity studies are planned to better understand the tradeoffs associated with the selection of this parameter. For conflicts detected with fewer than 8 (or 12) min to predicted loss of separation, the conflict resolution process was initiated upon detection. The relevant data for each detected traffic conflict within this *resolution-initiation horizon* were passed to the conflict resolution algorithm. Each resolution trajectory computed by the algorithm was transmitted to ACES, which implemented the resolution trajectory for the appropriate aircraft.

Some assumptions and simplifications were made in the modeling and execution of the experiment. In the ACES simulation environment, aircraft trajectories were entirely deterministic. Aircraft executed their planned trajectories and conflict resolution trajectories perfectly. As such, trajectory conflicts could be predicted with perfect accuracy over any time horizon, and resolution trajectories could be assured to be conflict-free. To offset the absence of uncertainty in trajectory modeling, a buffer of 2 nmi was added to the nominal (5 nmi) horizontal minimum separation parameter for the purpose of computing resolution trajectories. For example, resolution trajectories were required to achieve minimum horizontal separation of 7 nmi (i.e., 5 nmi standard separation plus the 2 nmi buffer). Execution of the resolution trajectories also was simplified by not modeling the negotiation of trajectories between aircraft operators and/or the air navigation service provider (ANSP), data-link transmission delays, nor pilot-action delays. Once a resolution trajectory was determined by the automation, it was executed immediately and precisely. Lastly, arrival traffic management initiatives such as miles-in-trail restrictions, time-based metering, or ground-delay programs were not modeled. Accordingly, aircraft were permitted to proceed to their destinations unimpeded, even if arrival demand was projected to exceed capacity. The conflict resolution algorithm was the only means for organizing arrival traffic flows. During peak arrival rush periods, this presented a stern challenge to the algorithm.

Detailed trajectory data and internal algorithmic parameters were recorded for each detected conflict and each trial-plan resolution trajectory considered (i.e., rejected or accepted) by the resolution algorithm. These data were archived in a relational database for post-processing and analysis, the results of which are presented in the next section.

IV. Results

This exploratory experiment sought to benchmark the performance of the candidate algorithm in unscripted, large-scale, high-volume air traffic scenarios based on real data. Table 1 summarizes the total number of flight operations and detected conflicts for each simulation run (i.e., “1x,” “1.5x,” and “2x”). While total flights (i.e., the independent variable) were increased linearly for each test case in this experiment, the number of conflicts produced increased geometrically, as expected. The “Total conflicts” result from each simulation run is indicative of the magnitude of the problem posed to the conflict resolution algorithm. Total conflicts are partitioned into “merging

arrival conflicts” and “all other conflicts.” Merging arrival conflicts are defined as projected conflicts between two aircraft on arrival to the same arrival fix and within 20 minutes flying time of that fix. Conflicts between any other pair of aircraft (e.g., cruise–cruise, departure–arrival, or two arrivals to *different* arrival fixes) are tallied in the last row of Table 1. Although merging arrivals (line c of Table 1) comprise a small minority of the traffic in Cleveland Center, conflicts between such aircraft comprised nearly half of all conflicts. This was expected, since arrival-fix crossing restrictions compel these aircraft to fly merging trajectories. Since the prior work did not include results for merging arrivals,⁸ the present study represents a more complete investigation for data sets up to twice the nominal demand level. The remainder of this section presents results which benchmark the performance of the algorithm in resolving both sets of conflicts and draws attention to performance differences in resolving arrival-merge conflicts as compared to conflicts in other phases of flight.

Table 1. Summary of flight operations and detected conflicts.

	1x	1.5x	2x
a. Total flights (24 hrs)	7493	11,789	17,059
b. Total conflicts	1430	3145	6536
c. – merging arrival conflicts	662	1177	2156
d. – all other conflicts	768	1968	4380

The results that follow are categorized into a safety-related benchmark and an efficiency-related benchmark. The safety-related benchmark captures the percentage of detected conflicts that were successfully resolved in each test case. The efficiency-related benchmark uses delay per resolved conflict to characterize the operational efficiency of the resolution maneuvers prescribed by the algorithm. The safety-related results are presented first, followed by the efficiency-related results.

A. Safety-Related Results

Safety is the paramount consideration for any conflict resolution algorithm. In this study, safety was measured in binary fashion based on whether or not the conflict resolution algorithm solved the detected conflict with a successful resolution trajectory. The results are presented in Table 2.

Table 2. Conflict resolution results.

	1x	1.5x	2x
a. Total conflicts	1430	3145	6536
b. Resolved by conflict resolution algorithm	1428	3143	6515
c. Not resolved by conflict resolution algorithm; delegated to conflict avoidance algorithm	2	2	21
d. Success rate for conflict resolution algorithm	99.86%	99.90%	99.68%

As described in Section II, the conflict resolution algorithm provides the first layer—the most strategic layer—of separation assurance in a multi-layer system (conflict avoidance and collision avoidance being the other two layers). Line d of Table 2 shows that more than 99% of the conflicts were resolved at this first layer. While these numbers are strong in an absolute sense, safety-critical systems are typically required to meet success rates several orders of magnitude closer to unity (perfection). Without considering the performance of the other layers of the system and without an established performance target for the overall system, it is not possible to assess the adequacy of the results above. These are matters for future research.

Tables 3 and 4 categorize the results from Table 2 by conflict type: merging arrival conflicts versus all other conflicts. As expected, the conflict resolution algorithm performed better for the latter set of conflicts (Table 4), as those conflicts impose fewer constraints on the resolution algorithm. More than 99.9% of those conflicts were successfully resolved by the conflict resolution algorithm at each demand level tested. These results are consistent with those in Ref. 8. They contrast, however, with the merging arrival results shown in Table 3, where more failures occurred despite there being fewer conflicts to resolve. The success rates still exceed 99% for merging arrival operations.

The resolution algorithm failed to produce an acceptable resolution trajectory for a total of 25 conflicts across the three different traffic levels. In assessing these conflicts, two observations can be made. First, 21 of those cases were merging arrival conflicts (ref. line c of Table 3), which underscores the importance and relative difficulty of this class of conflict. Second, in this more challenging category of conflict, the algorithm’s performance was stable to traffic increases up to 50% above today’s levels. Beyond that, its performance was lower. Additional evaluations at traffic levels from 1x to 3x are needed to better understand the algorithm’s sensitivity to increasing levels of demand.

Table 3. Conflict resolution results for merging arrivals.

	1x	1.5x	2x
a. Conflicts between merging arrivals	662	1177	2156
b. Resolved by conflict resolution algorithm	660	1176	2138
c. Not resolved by conflict resolution algorithm; delegated to conflict avoidance algorithm	2	1	18
d. Success rate for conflict resolution algorithm	99.70%	99.92%	99.17%

Table 4. Conflict resolution results for all other operations.

	1x	1.5x	2x
a. All other conflicts (i.e., not merging arrivals)	768	1968	4380
b. Resolved by conflict resolution algorithm	768	1967	4377
c. Not resolved by conflict resolution algorithm; delegated to conflict avoidance algorithm	0	1	3
d. Success rate for conflict resolution algorithm	100%	99.95%	99.93%

B. Efficiency-Related Results

Second in importance to safety is the operational efficiency of the resolution trajectories produced by the algorithm. Successful resolution trajectories that impose less delay or consume less fuel are preferable to those which impose or consume more.

The histogram in Figure 1 provides insight into the delay imposed by the algorithm. The histogram includes the delay data from each of the three simulation runs. The histogram is partitioned into 30-second bins (i.e., -15 sec to +15 sec of delay, +15 sec to +45 sec of delay, etc.). Negative minutes of delay indicate that the resolution trajectory realized a time savings relative to the original flight plan. The delay distributions are relatively insensitive to demand. The mean delay at each demand level is less than 35 sec as shown in Figure 2, and it appears to be stable with respect to projected increases in demand. (The slight decrease observed in average delay as demand increases is not a statistically significant result, based on the Kolmogorov-Smirnov test at $p < 0.05$.) Twenty-one percent of conflicts were resolved with a time savings (i.e., negative delay); 8% of conflicts required a delay of 2 min or more. These results, too, were insensitive to the level of demand.

As expected, these metrics were not as good for conflicts between merging arrivals. Figure 3 presents the histogram of delay imposed by the algorithm for merging arrivals. While the proportion of conflicts resolved with a time savings is marginally higher than for the general population, so is the proportion of conflicts with resolution delays of 1 min or higher. Figure 4 contrasts the mean delay for merging arrivals versus all other operations at each demand level. Resolution trajectories produced by the algorithm for arrivals bound for the same arrival fix imposed roughly three times the delay of resolution trajectories for other types of conflict.

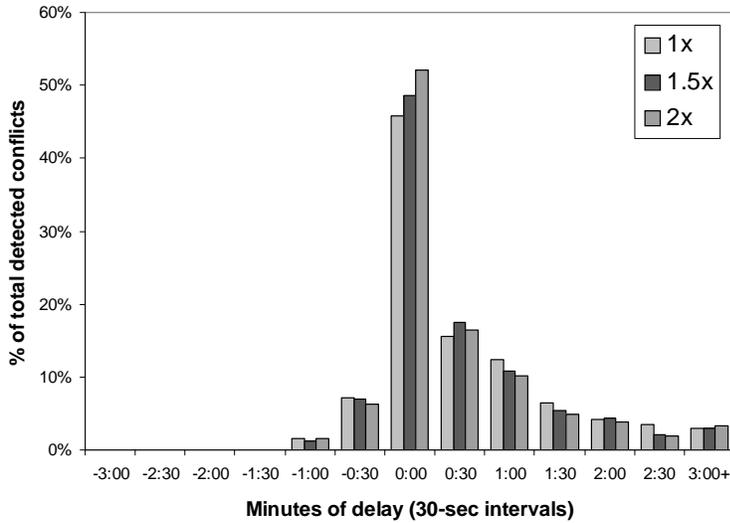


Figure 1. Histogram of delay assessed by the conflict resolution algorithm for all detected conflicts.

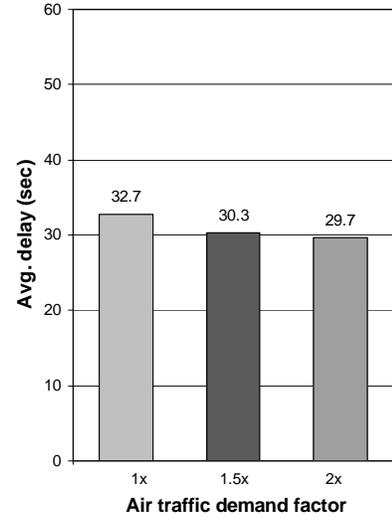


Figure 2. Average delay assessed by the conflict resolution algorithm.

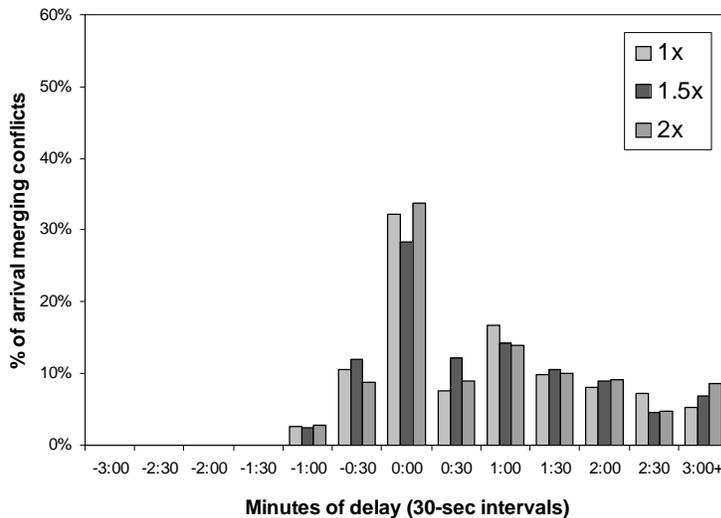


Figure 3. Histogram of delay assessed by the conflict resolution algorithm conflicts between merging arrivals.

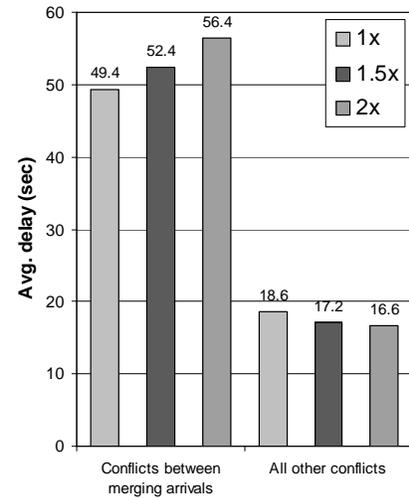


Figure 4. Average delay assessed by the conflict resolution algorithm, by type of conflict.

V. Discussion

These experimental results confirm the expected results: conflicts arising during arrival merging operations are more difficult to resolve, and their resolution trajectories are generally less efficient. These conflicts are more difficult to resolve because, due to the structure and operational conventions of the NAS, the solution criteria are more onerous than for other types of conflict. They require that resolution trajectories achieve safe separation while still ensuring that both aircraft overfly the same nearby point (2 to 20 min ahead) at the same altitude in relatively close succession, and they specify the order in which the aircraft should cross that point. These more restrictive criteria naturally constrain the pool of available resolution trajectories, making it less likely that a conforming trajectory will be found. Conflict resolution trajectories for merging arrivals are generally less efficient, because the algorithm favors lateral path-stretching maneuvers over vertical maneuvers and the former tend to impose more delay than the latter. Lateral maneuvers are favored based on a design decision not to amend an aircraft's vertical trajectory during the arrival phase of flight.

The analysis of arrival merging delays focused on the amount of delay imposed per conflict resolved. It did not include an accounting of aggregate delay per aircraft over the course of its arrival descent. Some arrival aircraft experienced multiple conflicts on their way to the runway. While some resolution delays were partially subsumed by subsequent resolution delays, there were also cases where aircraft incurred the full brunt of one conflict-resolution delay before being assessed another one, especially in the highly congested 1.5x and 2x scenarios. It is important to reiterate that no demand-management initiatives (e.g., time- or distance-based metering, ground delay programs, etc.) were employed in these simulations. In the absence of such, the conflict detection and resolution function was the only means of managing the arrival processes to Detroit, Cleveland, Pittsburgh, etc. By tallying the aggregate delay experienced by each aircraft, researchers may begin to discern the “congestion frontier” at which conflict detection and resolution should be supplemented with traffic management initiatives—or conversely, the frontier beneath which traffic management initiatives may prudently be removed.

For conflict cases arising from flight operations *other* than the arrival merge, the results of this study corroborate the results of the prior benchmarking effort⁸ and, in fact, raise the bar. As shown in Table 5, the algorithm’s performance on the safety-related metric was virtually identical, and its performance on the efficiency-related metric was improved. Conflict resolution delays were reduced significantly ($p < 0.001$, Kolmogorov-Smirnov test). The improved performance validates the design change discussed in Section IV, whereby the algorithm now develops a pool of acceptable resolution trajectories (i.e., one in each axis for each aircraft) and executes the best of them, based on a cost function. Broader input from stakeholders may be needed to assess whether or not the current cost function is the “right” one.

Table 5. Results from this study versus the prior work.⁸

	1x		2x	
	Prior study	Present study	Prior study	Present study
a. Success rate for conflict resolution algorithm	100%	100%	99.92%	99.93%
b. Average delay per resolution trajectory (sec)	20.6	18.6	22.1	16.6

VI. Summary

Automated separation assurance is believed to be a key enabler in expanding the capacity of the air transportation system. System designers have established target capacity goals as high as three-times that of today. While research in pursuit of automated conflict detection and resolution algorithms has been ample, results to date have not adequately probed the performance envelope with the volume and complexity of air traffic envisioned in the future air traffic system.

This study updates some initial benchmarks for the performance envelope of one candidate conflict resolution algorithm. In fast-time simulations of the United States’ busiest en-route airspace, the algorithm successfully resolved 99.9% of detected conflicts at the 1x and 1.5x demand levels, and 99.7% at the 2x demand level. It must be noted that these metrics were taken in simulations having no trajectory-modeling error; to compensate, resolution trajectories were required to achieve 7 nmi horizontal separation from proximate aircraft, a 2-nmi buffer relative to the legal separation standard.

The efficiency metrics were improved relative to the prior work, owing to a change in the conflict resolution algorithm. Average delay incurred per conflict was less than 20 sec for those that did not arise during arrival merging, and delays were not sensitive to the traffic levels tested. On the other hand, delays imposed during arrival merging were much higher, as expected, reflecting the design decision to favor lateral maneuvers over vertical maneuvers in the descent phase of flight.

VII. Future Work

The prior work probed the performance of the conflict resolution algorithm at traffic levels up to three times that of present-day operations but did not include results for merging arrivals. The present work included merging arrivals, but only at traffic levels up to twice the present-day level. This leaves a gap in coverage for a future simulation experiment that includes merging arrivals for a traffic demand level of three times that of present-day operations.

To provide a fair and complete set of performance benchmarks for the subject algorithm, the study should be revisited once appropriate trajectory uncertainty models have been incorporated. At that time, the study should include sensitivity studies for parameters such as the resolution-initiation time horizon, an accounting of aggregate delay per aircraft, and evaluations for additional traffic days and for airspaces other than Cleveland Center. Pending the results of those experiments, the validation process will be augmented with real-time simulation experiments.

The purpose of this paper was to establish a benchmark for automated conflict resolution in a complex, high-volume air traffic environment. It is hoped that others in the community will collaborate in this effort by documenting the performance of alternative algorithms against these same benchmarks, and by offering additional, perhaps more insightful, benchmarks for future studies here and at large. In this way, the community can begin to close in on a viable solution for automated separation assurance.

Acknowledgement

The authors wish to acknowledge Mr. Christopher Farrell, whose contributions to the data collection and reduction efforts were invaluable.

References

- ¹ FAA, *Terminal Area Forecast*, Washington, District of Columbia, 2006.
- ² FAA Joint Planning & Development Office, *Concept of Operations for the Next-Generation Air Transportation System*, Draft 3, Version 1.1a, Washington, District of Columbia, 2006.
- ³ Kuchar, J. K., Yang, L. C., *A Review of Conflict Detection and Resolution Modeling Methods*, 2000, IEEE Transactions on Intelligent Transportation Systems, Vol. 1, No. 4, December 2000, pp. 179-189.
- ⁴ Mondoloni, S., Conway, S., *Airborne Conflict Resolution for Flow-Restricted Transition Airspace*, AIAA-2001-4054, American Institute of Aeronautics & Astronautics, Reston, Virginia, 2003.
- ⁵ Vivona, R. A., Karr, D. A., Roscoe, D. A., *Pattern-Based Genetic Algorithm for Airborne Conflict Resolution*, AIAA-2006-6061, American Institute of Aeronautics & Astronautics, Reston, Virginia, 2005.
- ⁶ Bilimoria, K. D., Lee, H. Q., *Aircraft Conflict Resolution with an Arrival Time Constraint*, AIAA-2002-4444, American Institute of Aeronautics & Astronautics, Reston, Virginia, 2002.
- ⁷ Mondoloni, S., Ballin, M. G., *An Airborne Conflict Resolution Approach Using a Genetic Algorithm*, AIAA-2003-6725, American Institute of Aeronautics & Astronautics, Reston, Virginia, 2001.
- ⁸ Farley, T., Erzberger, H., *Fast-Time Simulation Evaluation of a Conflict Resolution Algorithm Under High Air Traffic Demand*, Proceedings of the 7th USA/Europe Air Traffic Management R&D Seminar, Barcelona, Spain, 2007.
- ⁹ Erzberger, H., *Transforming the NAS: The Next Generation Air Traffic Control System*, NASA TP-2004-212828, Moffett Field, California, 2004.
- ¹⁰ Erzberger, H., *The Automated Airspace Concept*, Proceedings of the 4th USA/Europe Air Traffic Management R&D Seminar, Santa Fe, New Mexico, 2001.
- ¹¹ Erzberger, H., *Automated Conflict Resolution for Air Traffic Control*, Proceedings of the 25th International Congress of the Aeronautical Sciences (ICAS), Germany, 2006.
- ¹² Paielli, R. A., *Algorithms for Tactical Conflict Resolution and Strategic Conflict Probability Reduction*, AIAA-2001-5243, American Institute of Aeronautics & Astronautics, Reston, Virginia, 2001.
- ¹³ Sweet, D. N., Manikonda, V., Aronson, J. S., Roth, K., Blake, M., *Fast-Time Simulation System for Analysis of Advanced Air Transportation Concepts*, AIAA-2002-4593, American Institute of Aeronautics & Astronautics, Reston, Virginia, 2002.
- ¹⁴ Paglione, M. M., Oaks, R. D., Bilimoria, K. D., *Methodology for Generating Conflict Scenarios by Time Shifting Recorded Traffic Data*, AIAA-2003-6724, American Institute of Aeronautics & Astronautics, Reston, Virginia, 2003.