# 4-D Perturbation Analysis of Conflict Scenarios

Larry A. Meyn[*]
*NASA Ames Research Center, Moffett Field, CA, 94035*

**A method is presented for perturbing air traffic scenarios and analyzing the resulting conflicts. The perturbations consist of a specified range of spatial and temporal modifications of the trajectories, and the analysis identifies all possible conflicts within the perturbation range. This method enables new scenarios to be generated for simulation testing of air traffic management tools and concepts. Some potential applications are presented, such as the analysis of the sensitivity of a scenario to temporal perturbations, the prediction of areas of high-density traffic, and the estimation of conflict probabilities for long-range (one-to-six hour) trajectory predictions. A study of discrete and interpolated conflict detection accuracy as a function of the surveillance-sampling period is also presented, showing that interpolated detection halves the number missed conflicts when compare to discrete detection and that accuracy degrades significantly when the sampling period exceeds twelve seconds.**

## I. Introduction

To effectively cope with projected increased traffic demands on the National Airspace System (NAS), new operational concepts and tools are being researched and developed. The development must ensure that aircraft are separated at or above safe separation minima. When aircraft are predicted to violate the prescribed separation minima, they are said to be in conflict. Concepts that resolve conflicts and maintain safe separation are called separation assurance (SA) concepts.

The effectiveness of SA concepts is evaluated through simulation using traffic scenarios that have conflicting flights. SA concepts need to be robust, so simulations need to be performed over a wide variety of scenarios containing diverse collections of conflicts. One can generate traffic scenarios containing conflicts from scratch,[1] but a more common practice is to generate scenarios by perturbing recorded scenarios derived from field observations or high-fidelity flight simulations.[2-4] This process shifts the original scenario trajectories in space and/or time to generate new scenarios with alternate sets of conflicts. These scenarios then drive simulations or exercise SA algorithms. Unfortunately, NAS simulations, especially ones involving live participants, can be extremely costly and time consuming. Therefore, efficiently identifying scenarios that have interesting or challenging sets of conflicts is desirable.

Manual perturbations can be used to achieve this, but they risk not adequately representing the full range of potential scenarios. A more conservative approach is to generate a large number of randomly perturbed scenarios and then to evaluate them to detect conflicts. As the number of random perturbations increases, more potential conflicts are identified, but it is an extremely

[*] Aerospace Engineer, Aerospace Operation Modeling Branch, MS 210-15, AIAA Associate Fellow.

time-consuming process that would be useful to streamline. One approach, developed by Paglione, et al.,[4] was to produce time-shifted trajectories using a genetic algorithm to selectively control the random generation of time perturbations. The scenarios produced in each generation cycle of the algorithm were analyzed to identify desirable scenarios, and these were used to guide the randomized production of future scenario generations. Compared to a pure random generation process, this method significantly reduces the number of perturbed scenarios that need to be generated and evaluated. However, this approach still entails the evaluation of a large number of scenarios, and it does not guarantee that all potential conflicts are identified in the process.

A new approach is presented in this paper in which the original scenario is evaluated for potential conflicts within a specified range of spatial and temporal perturbations. These conflicts can then be analyzed and used to rapidly generate new scenarios of interest. There is no need to perform conflict detection on a series of scenarios with perturbed trajectories. Some iteration and optimization may be necessary to produce scenarios with desirable combinations of conflicts, but the population of possible conflicts is known and all necessary conflict information can be pre-determined. This approach not only significantly shortens the process by which new conflict scenarios can be generated; it has other potential applications as well. Prototypes of three such applications are presented: an analysis of the sensitivity of scenario conflict count to time perturbations, identification of potentially congested air space, and estimation of conflict probabilities for large temporal uncertainties.

The next section provides some background material on conflict detection techniques. It is followed by a description of the scenario perturbation analysis methodology. Next is a section detailing some potential applications of the methodology; followed by conclusions. An appendix presenting the sensitivity of geometric conflict detection accuracy to trajectory time resolution is also provided.

## II. Background

Conflict detection can apply to uncertain predictions of flight trajectories or to theoretically exact trajectories. For the purposes of this paper, these two activities will be referred to as probabilistic and geometric conflict detection. The first step in probabilistic methods is often a geometric conflict detection using a set of predicted future trajectories. The focus of this paper is on finding conflicts in scenarios with pre-defined trajectories, which falls into the category of geometric conflict detection.

Geometric conflict detection uses aircraft trajectories to determine if pairs of aircraft are in violation of minimum horizontal and vertical separation distances at the same time. Several algorithms have been developed for probabilistic and deterministic (i.e., geometric) conflict detection[5-10] and for the closely related field of collision detection.[11-15] The primary difference between prior work and that presented here is the focus on finding all possible conflicts for a range of spatial and temporal perturbations. Previously developed geometric conflict detection algorithms generally lack the ability to specify a temporal proximity other than zero. An exception to this is the algorithm developed by Sridhar and Chatterji[6] that identifies groups of aircraft within a specified lateral proximity and then tests for vertical and temporal proximity. However, this was done to produce an extremely fast conflict detection algorithm, and the ability to perform time-shifted conflict detection was not mentioned as a feature.

The ability to detect conflicts over a range of spatial and temporal perturbations was the primary design goal for the method presented here. Secondary goals were detection accuracy and reasonable computation speed. Both accuracy and computational speed are functions of the time step used in discretizing the trajectories. Although 3-D trajectory positions vary continuously with time, the available data is usually restricted to periodic samples due to instrumentation limits for observed trajectories or due to computational resource limits for trajectories generated by simulation. Current radar surveillance in the NAS has sampling rates as low as once per 12 seconds. Simulations generally use 5 to 10 second sampling rates for conflict detection.

However, using time discretized data can lead to detection errors. This loss of detection accuracy is due to conflicts that occur between time steps. An example of how this can happen is shown in Fig. 1. At time 0, the locations of aircraft A and B are shown at points $A_0$ and $B_0$. At time 1, their locations are shown at points $A_1$ and $B_1$. In this situation it is possible for the two aircraft to not be in conflict at times 0 and 1, but be in conflict at some intermediate time, time 0.5, represented by, points $A_{0.5}$ and $B_{0.5}$. Two ways to reduce the possibility of missed conflicts are to decrease the size of the time steps or to interpolate the aircraft positions between time steps. Appendix A presents a comparison of conflict detection accuracy between discrete detection and detection using linearly interpolated aircraft positions. In order maximize accuracy for any given time step, the interpolation method was chosen for this work.

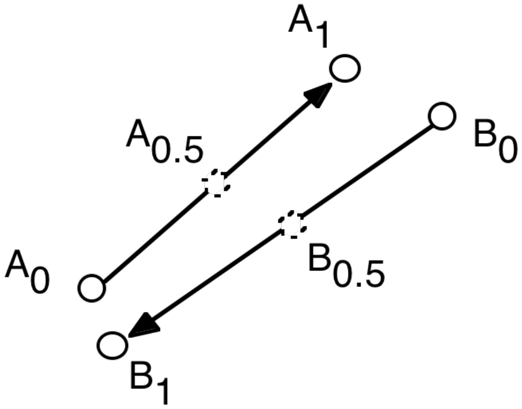

**Figure 1. Conflicts occurring between time steps.**

## III. Conflict Detection Methodology

The conflict detection method presented here is based on determining the proximity of bounding boxes.[12-14] For this application, a bounding box contains a segment of a trajectory between two time values. The finest resolution for bounding boxes consists of the segment between two time steps. At the coarsest resolution are bounding boxes that contain entire trajectories. Using bounding boxes at the coarsest and at some intermediate resolutions can speed up the proximity detection process. Each bounding box has a minimum and a maximum value for each dimension. These are $lat_{min}$ and $lat_{max}$ for latitude. Similar terms are defined for longitude, altitude and time, which are abbreviated here as *lon*, *alt*, and *t*, respectively. Perturbations ranges are specified for the horizontal plane and for the vertical and temporal dimensions. These are represented by the terms $\Delta h$, $\Delta v$ and $\Delta t$ respectively. The proximity values, *P*, for each dimension are defined by eqns. 1 through 4. The terms $h_{sep}$ and $v_{sep}$ are the horizontal and vertical separation minima.

$$P_{lat} = \left(\Delta h + h_{sep}\right)\frac{\deg}{60nmi} \qquad 1)$$

$$P_{lon} = \frac{P_{lat}}{\cos\left(lat_{\max}\right)} \qquad 2)$$

$$P_{alt} = \Delta v + v_{sep} \tag{3}$$

$$P_t = \Delta t \tag{4}$$

Equation 2 shows that $P_{lon}$ is adjusted to account for the changing ratio between degrees longitude and nautical miles that is a function of latitude.

To determine if two trajectories, *A* and *B*, are potentially within the specified proximity, each bounding box in *A* is tested for proximity with each bounding box in *B*. To determine if two bounding boxes, $A_i$ and $B_j$, are within the specified proximity values, $P_{dim}$, the logical expression shown in Eqn. 5 is applied for each dimension, *D*, where *D* can be *lat*, *lon*, *alt* or *t*. In this equation, $D_{max}(A_i)$ would be the maximum latitude of box $A_i$ if *D* were *lat*.

$$\left(D_{\max}\left(A_i\right) + P_D \geq D_{\min}\left(B_j\right)\right) \& \left(D_{\max}\left(B_j\right) + P_D \geq D_{\min}\left(A_i\right)\right) \tag{5}$$

If this equation evaluates as false for any dimension, then the two bounding boxes, $A_i$ and $B_j$, are not within the specified proximity. As these tests are applied serially, a false value at any evaluation step obviates the need to apply the test for any remaining dimensions.

A two dimensional example of this is presented in Fig. 2. Bounding boxes are determined for trajectory segments in trajectories *A* and *B* for the dimensions *x* and *y*, where *x* and *y* are chosen from the dimensions latitude, longitude, altitude and time. Then for each bounding box in *A*, the proximal boxes in *B* are determined. In this example, boxes $B_1$, $B_2$, $B_3$ and $B_4$ are all within $P_x$ of $A_7$. In the *y* dimension, only boxes $B_1$, $B_2$ and $B_3$ are with $P_y$ of $A_7$. Since boxes must be proximal in all dimensions, only $B_1$, $B_2$ and $B_3$ are considered within the specified proximity.

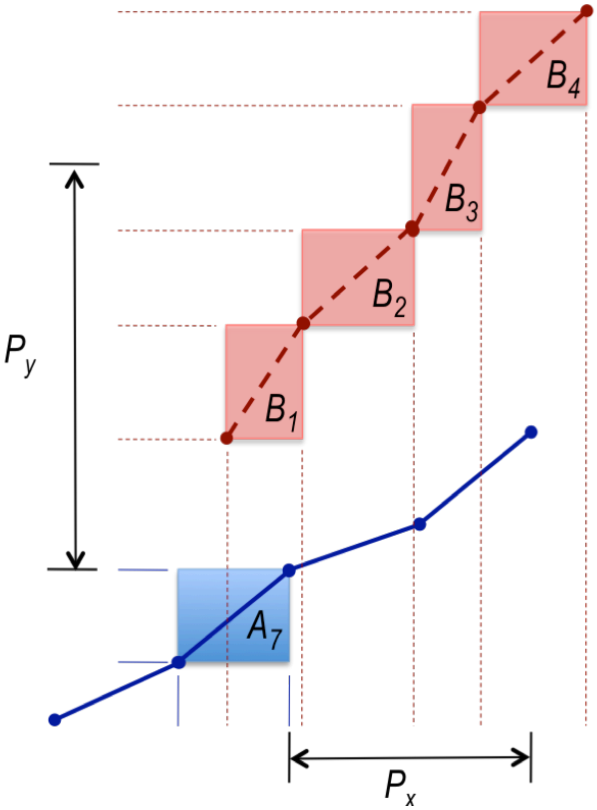

**Figure 2. Schematic of bounding box tests for trajectories A and B in two dimensions.**

To identify all potential conflicts in a scenario within the specified proximity ranges, all pairs of trajectories must be evaluated to determine if they are within proximity values as defined previously. There are *n*(*n*-1) pairs for *n* trajectories and each trajectory can have one or more bounding boxes depending on the time resolution chosen. For each trajectory pair, all the bounding boxes for the first trajectory must be tested for proximity against all the bounding boxes for the second trajectory. The number of proximity tests increases as resolution time step decreases. In an effort to reduce the computational resources required, each trajectory is processed to generate bounding boxes at three resolutions, a single bounding box for the entire trajectory, bounding boxes for 5-minute trajectory segments, and bounding boxes for each time step. Trajectory pairs are first tested for proximity using the full-trajectory bounding boxes. If this proximity test fails, then no further proximity tests are required. If this proximity test passes, then the 5-minute bounding boxes are tested for proximity. Again, if this test fails, no further proximity tests are required. If the 5-minute test indicates possible proximity, then the proximity test is run using bounding boxes for each trajectory time step.

This series of bounding box tests down-selects the possible trajectory segment pairs that are in potential conflict within the specified perturbation range. Up to this point in the process, time has been treated the same as any other dimension. However, to determine if two trajectory segments are within the specified radial proximity, $\Delta h + h_{sep}$, it is assumed that the starting and ending times of segment pairs are synchronized. This enables the time-varying distance between points on the two trajectory segments to be easily expressed as a function of time, so that radial proximities in between the starting and ending points can be easily determined. However, this effectively limits the possible time perturbations to multiples of the trajectory discretization time step, with a maximum value of the specified maximum time perturbation.

This methodology was implemented in the Python[16] programming language and run on a 3 GHz desktop computer with eight processing cores. Processing times vary depending on the set of trajectories and the perturbation range selected, but a processing time on the order of three minutes is typical for a scenario with over 900 trajectories. Further optimizations could be made to reduce the processing time, but it was not deemed necessary for the purposes of this research.

## IV. Applications

The target application for the conflict scenario analysis tool is the development of perturbed scenarios with a range of conflict sets for evaluating air traffic management (ATM) concepts and tools. However, the fact that all possible conflicts are identified for a range of spatial and temporal perturbations leads to some possible new applications. Some early prototypes of these applications are also presented here. These include the analysis of how temporal perturbations affect the conflict count, the prediction of potentially congested air space, and the estimation of conflict probabilities for far-term (1-6 hour) trajectory predictions with large temporal uncertainties.

All of the results presented here were developed using a sample scenario generated by the Airspace Concept Evaluation System (ACES.)[17] Although ACES integrates trajectories using half-second time steps, the scenario trajectory data was only recorded at 10-second time intervals for this study. The scenario represents current day traffic levels for flights departing nationwide over a three-hour period. Only flights that pass through the Cleveland Air Route Traffic Control Center were included in the scenario. Figure 3 shows the boundaries of Cleveland Center, along with some of the airports within the center. Chicago is nearby, to the West of the center. Washington D.C. and Baltimore are to the Southeast. Philadelphia, New York and Boston are East of the center. In addition to flights departing from and arriving at airports within the center, a significant

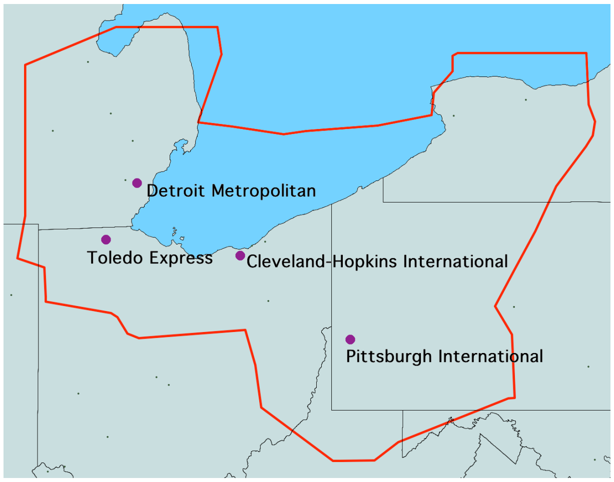

**Figure 3. Cleveland Air Route Traffic Control Center.**

amount of traffic passes through the center en route between the major airports serving the cities mentioned above.

### A. Generation of Test Scenarios

Thorough evaluation of new SA concepts requires their evaluation with a wide variety of conflicts and with sets of conflicts that represent possible off-nominal or "worst-case" scenarios. An example of this is the generation of an off-nominal scenario with a large number of conflicts and groups of conflicts that are in close proximity.

To produce this type of scenario, the original ACES scenario, which contained 723 conflicts for $h_{sep} = 5$ nmi and $v_{sep} = 1000$ ft, was analyzed for a time perturbation range, $\Delta t$, of 5 minutes. This analysis identified 1784 possible conflicts for relative time offsets between trajectory pairs within this perturbation range. Using this information, the flight departure times in the original scenario were adjusted to increase the number conflicts by means of an optimization routine that perturbed departure times to generate relative time offsets identified in the set of all possible conflicts. This procedure produced a new scenario with over 1300 conflicts.

When this scenario was used in an ACES simulation that incorporated an SA algorithm, there was little difference in the conflict resolution statistics when compared to the statistical results of the original scenario. The primary reason for this is that the procedure for simulating SA concepts within ACES is to ignore flights that are already violating the separation minima when they enter the region of SA authority. This is necessary because the SA models are designed to prevent violations from occurring and flights that are already in violation should be a rare occurrence that is handled by other procedures. When these flights were filtered out, both scenarios had roughly the same number of conflicts.

A method of altering scenario trajectories to remove violations prior to entering the SA region is currently being pursued. This should allow most of the potential conflicts within a scenario to be considered by the SA model. The ability to create new scenarios based on metrics other than conflict count is also under development.

### B. Determining Scenario Sensitivity to Temporal Perturbations

The original ACES scenario described above was one of the test scenarios used to test an SA concept in ACES. When flight departure times were randomly perturbed for some early tests, the number of conflicts always decreased. As the only differences in the scenarios were the trajectory start times, the result was perplexing. In an attempt to understand this, a conflict perturbation analysis was run for time perturbations of up to 15 minutes ($h_{sep} = 5$ nmi, $v_{sep} = 1000$ ft, $\Delta t = 15$ min and $\Delta h = \Delta v = 0$,) and then the total number of conflicts was determined for different relative time shifts between trajectories. The results are presented in Fig. 4. A 5-minute period in the conflict count is clearly evident. This is believed to be due to the scheduled flight departure times in the original scenario data having pronounced peaks at 5-minute intervals as shown in Fig. 5. To test this hypothesis, the flight departure times were randomized using a Gaussian distribution with a

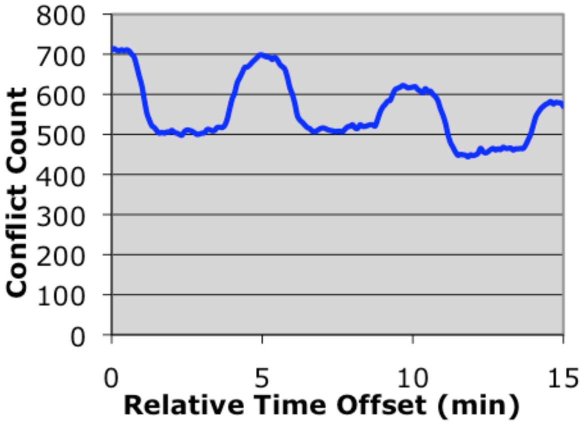

**Figure 4. Conflict count vs. relative time offset for original scenario.**

standard deviation of 2 minutes. The corresponding conflict count plot is shown in Fig. 6.

The periodic structure shown in Fig. 4 would not be expected to occur in real world trajectory data, as there are many mechanisms that would tend to randomize trajectory times at the airport gate, on the airport surface, and in the air. In fact, the ACES SA simulation run would probably not have had the same periodic structure had the air traffic control models for airports been enabled. (They were disabled to focus strictly on the SA model behavior.) However, the existence of such structure in scenarios should be identified, so that researchers can avoid situations where that structure might influence results. An example would be the comparison between two concepts whose primary difference is the accuracy by which flights adhere to predicted trajectories. When using a scenario with the time structure in Fig. 4, flights with more accuracy would be expected to have more potential conflicts. For such a comparison, scenarios without a periodic time structure should be used.

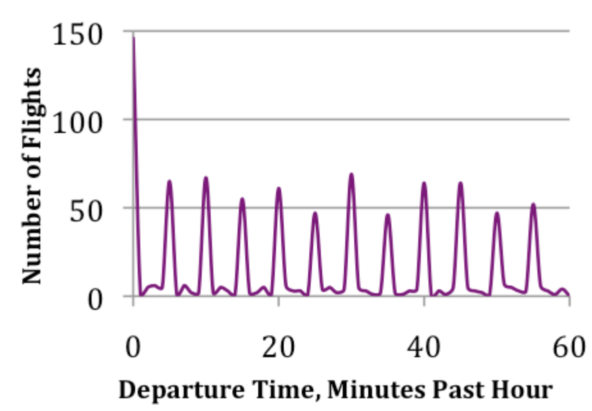

**Figure 5. Scheduled flight departure times.**

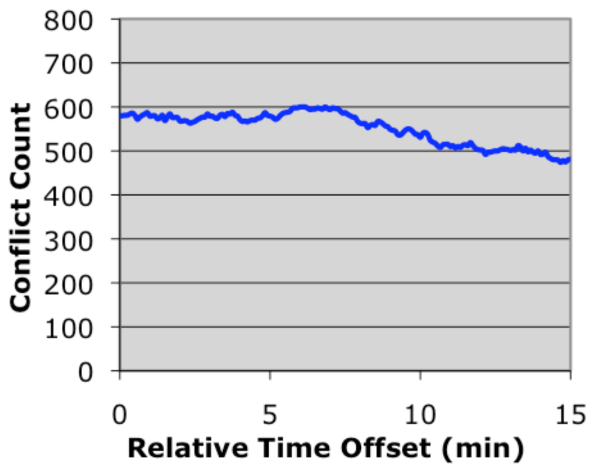

**Figure 6. Conflict count vs. relative time offset for scenario with randomized departure times.**

### C. Prediction of High-Traffic-Density Areas

One of the concepts being proposed for future NAS operations is Dynamic Airspace Configuration (DAC).[18] The main idea is to generate new airspace control volumes in response to changes in air traffic demand, aircraft equipage and weather. One of the factors to consider in designing new airspace regions is where and when concentrations of aircraft are likely to occur. One approach is to estimate these traffic concentrations using flight trajectories derived from historical data.[19] This may work well when expected flight patterns are similar to historical patterns, but if the expected flight patterns are significantly different, then trajectory predictions will have to be used. DAC operations are likely to be working with predictions that may be hours in the future, so the temporal uncertainties in the flight trajectories can be several minutes for airborne flights, or significantly larger for flights that have not yet departed.[20,21]

By using a conflict perturbation analysis with perturbations sized according to the estimated trajectory uncertainties, it would be possible to identify potential concentrations of flights for use in DAC airspace designs. Figure 7 shows an example of a set of flight trajectory segments that have overlapping proximities when the horizontal range is set to 10 nm, the vertical range is 1500 ft and the temporal range is 15 min. The set of flight segments shown is the largest of several distinct sets found in the scenario data. Once a grouping of this type is identified it could be further partitioned using origin/destination information, aircraft equipage and other factors that might be used in DAC airspace design.

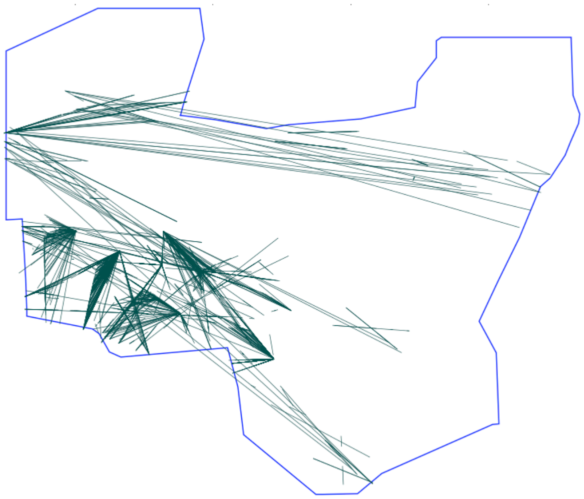

**Figure 7. Sample group of proximal flight trajectory segments.**

### D. Estimating Far-Term Conflict Probabilities

In the previously described application, perturbation ranges equal to two standard deviations of the trajectory uncertainty in each dimension were used to map out potential traffic concentrations. A closely related application is to utilize the trajectory uncertainty estimates to calculate conflict probabilities for such far-term predictions.

The approach presented here was to calculate the uncertain location of 5 nm diameter disks representing the separation minima about each aircraft. The regions in space and time that these disks occupy are initially given a probability of one. These probability values are then convolved by probability distributions representing the lateral and temporal location uncertainty for each aircraft to produce probabilistic conflict regions. In the example shown in Fig. 8, the lateral and temporal uncertainties are Gaussian with standard deviations of 2 nm and 1 min, respectively. This image is for a 10 second period for a 160 nm x 130 nm region of airspace, where each pixel represents an area that is 0.1 nmi by 0.1 nmi. For this example, altitude variation was ignored, due to the difficulty of presenting an additional dimension. The colors are "hotter" for high probabilities that two or more conflict regions will be overlap and "cooler" when only one conflict region is probable.

The uncertainty values used here were chosen for example purposes only. For actual practice, the uncertainties would vary for each flight and would change with the time horizon of the trajectory prediction. This type of far-term conflict probability could contribute to the computation of congestion or complexity metrics used for DAC or traffic flow management.

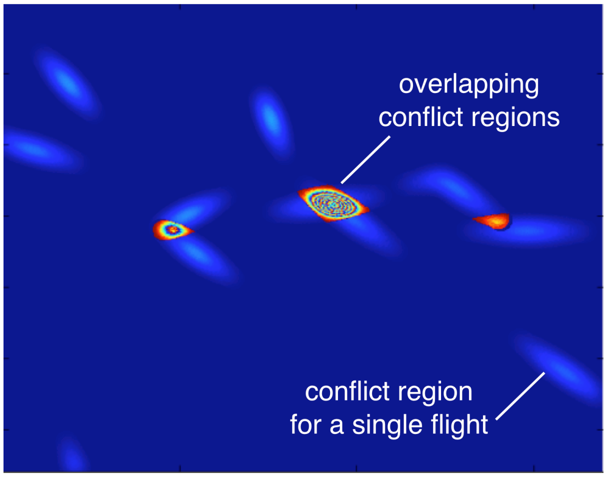

**Figure 8. Image of far-term, probabilistic conflict region locations for several flights.**

## V. Conclusions

A method has been presented to analyze flight trajectory scenarios to determine all of the conflicts that could occur if the trajectories were spatially and temporally shifted within a specified perturbation range. Although the impetus of this work was to enable rapid generation of off-nominal scenarios for evaluating separation assurance concepts, the fact that all possible conflicts within a specified perturbation range are identified, leads to potential new applications. Prototypes of three such applications were presented: the analysis of scenario sensitivity to temporal perturbations, the prediction of high traffic density areas, and the estimation of conflict probabilities for far-term trajectory predictions. The temporal sensitivity analysis showed that strong periodicity in airline departure schedules can lead increased conflict rates in simulations that don't include mechanisms that distribute departures. The predictions of high traffic density areas and far-term conflict probabilities are currently being evaluated for possible use in new air traffic management tools. The 4-D perturbation analysis has been shown to have several potential applications and should be a useful new tool for air traffic research and development.

## Appendix
### The Effect of Trajectory Sampling Rate on Conflict Detection Accuracy

It is generally impractical to acquire or generate trajectory data that is continuous in time. However using sampled trajectory data could lead to detection errors. In an effort to determine how the trajectory sampling rate would effect detection accuracy, a sample set of trajectories, generated by an ACES simulation, with 933 flights were processed by two conflict detection algorithms for trajectory sampling periods ranging from 1 to 60 sec. Although sampling periods greater than 12 seconds are not generally used, the larger periods were included to provide a more complete picture of how accuracy degrades with increasing sampling periods. Both algorithms were simple geometric conflict detection algorithms, but one only calculated flight separation distances at the discrete time steps of the sampled data and the other calculated flight separation distances for linearly interpolated flight positions between time steps. These are henceforth referred to as the discrete and interpolated algorithms.

For the purposes of this study, the conflicts that were detected using the interpolated algorithm at a 1 sec sample rate were considered the "truth" set. One metric for accuracy is the number of missed conflicts, which applies to both algorithms. Another metric, for the interpolated algorithm only, is the number of false conflicts, because it assumes flight positions vary linearly between trajectory samples.

Table A-1 presents the number of missed conflicts and false conflicts for a range of sample periods as a percentage of the number of conflicts in the truth set. The truth set for this study contained 723 conflicts for separation minima defined as 5 nm in the horizontal plane and 1000 ft vertically. The table shows that the discrete algorithm missed a few conflicts even at the 1 sec sample period. No results are shown for the interpolated algorithm for this sample rate since it serves as the reference. At a sample period of 2 sec, the interpolated algorithm has no missed or false conflicts, which lends some confidence in the chosen truth set. For sample periods of 3 sec and greater, the discrete algorithm has about twice a many missed conflicts as the interpolated algorithm. The interpolated algorithm doesn't have any false conflicts until a sample period of 10 sec and the rate of false conflicts doesn't exceed 0.5% until a sample period of 30 sec.

**Table A-1. Missed and false conflicts for a range of sampling periods.**

| Sample Period (sec) | Missed Conflicts Discrete | Missed Conflicts Interpolated | False Conflicts Interpolated |
|---|---|---|---|
| 1 | 0.40% | -- | -- |
| 2 | 0.70% | 0.00% | 0.00% |
| 3 | 2.10% | 0.80% | 0.00% |
| 5 | 2.40% | 1.40% | 0.00% |
| 10 | 4.00% | 1.80% | 0.10% |
| 12 | 5.00% | 2.10% | 0.10% |
| 15 | 5.00% | 2.20% | 0.30% |
| 20 | 7.30% | 3.60% | 0.30% |
| 30 | 10.50% | 4.80% | 0.60% |
| 60 | 24.80% | 11.50% | 1.10% |

A missed conflict may not be significant if the violation is of short duration and the separation distances don't violate the separation minima by large amounts. To gage the relative severity of a missed conflict, a simple severity metric, *S*, is defined by Eqn. A-1, where *h* is the radial distance in the horizontal plane and *v* is the vertical distance between the two aircraft. The variables $h_{sep}$ and $v_{sep}$ are the required horizontal and vertical separation minima. When both *h* and *v* are less than $h_{sep}$ and $v_{sep}$ respectively, the flights are in conflict. If either value equals or exceeds its minimum separation requirement then there is no conflict and *S* is 0.

$$S = \min\left(\frac{h_{sep} - h}{h_{sep}}, \frac{v_{sep} - v}{v_{sep}}\right) \quad for\ h < h_{sep}, v < v_{sep} \qquad \text{A-1)}$$

Figure A-1 shows plots of conflict severity and duration for conflicts missed by the discrete and interpolated algorithms for trajectory time steps of 5, 10, 12 and 15 sec. Any conflict missed by the interpolated algorithm is also missed by the discrete algorithm, which is why the data points of the interpolated algorithm always overlay a data point for the discrete algorithm. As can be seen, severity and duration increase for both algorithms as the time step size increases. The severity values for the discrete algorithm seem to increase significantly when going from a time step of 12 sec to one of 15 sec. This does not occur for the interpolated algorithm. These results suggest that a 12 sec step size is the maximum that should be consider for discrete conflict detection. Although not presented here, both algorithms miss a conflict with a severity of 0.65 and duration of 17 sec when the time step is 20 sec, which would clearly be unacceptable accuracy.

Overall, interpolated detection is clearly superior to discrete detection in terms of the number of missed conflicts. The occurrence of false conflicts for interpolated detection is only 0.1% for sampling periods of 12 sec or less. To avoid the occurrence of false conflicts, the sampling period should be 5 sec or less. It is recommended that the sampling period for both detection methods be 12 sec or less and that an interpolated detection algorithm be used when feasible.

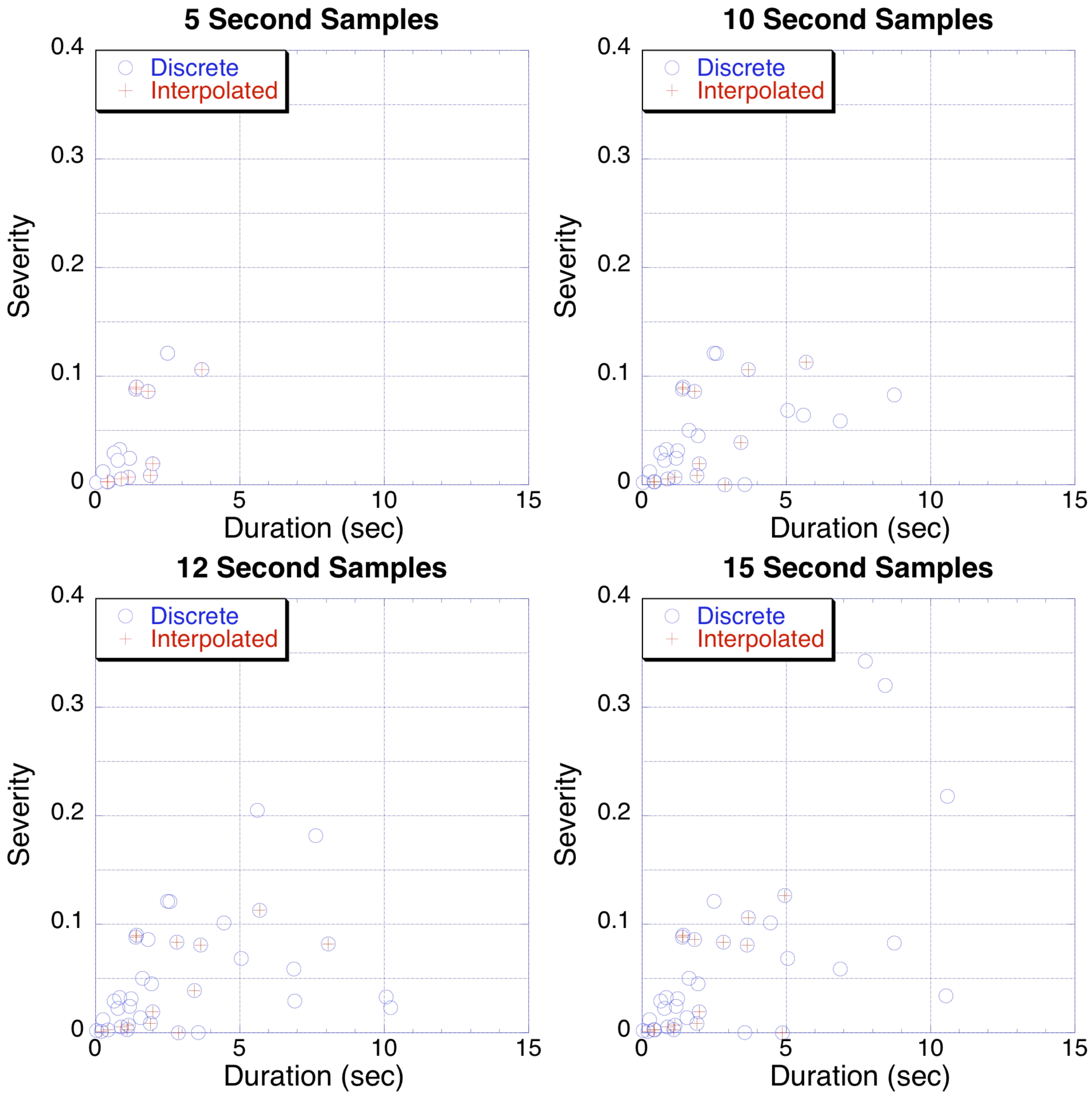

**Figure A-1. Severity and duration of missed conflicts for different trajectory sampling periods.**

## References

[1] Shan, Y., McKay, R., Essam, D., and Abbass, H., "Forming an Intelligent Ensemble of Conflict Detection Algorithms in Free Flight," The Artificial Life and Adaptive Robotics Laboratory, ALAR Technical Report TR-ALAR-200510014, University of New South Wales, Canberra, Australia, 2005.

[2] Bilimoria K., "Methodology for the Performance Evaluation of a Conflict Probe," *Journal of Guidance, Control, and Dynamics*, Vol. 24 No. 3, 2001, pp. 444,451.

[3] Loureiro, T., Blin, K., Hoffman, E., and Zeghal, K. "Development Of A Tool For Comparing Conflict Detection Algorithms For Air Traffic Management," *AIAA Guidance, Navigation, and Control Conference*, Montreal, Canada, Aug. 6-9, 2001.

[4] Paglione, M., Oaks, R., and Bilimoria, K., "Methodology for Generating Conflict Scenarios by Time Shifting Recorded Traffic Data," *AIAA's 3rd Annual Aviation Technology, Integration, and Operations Forum*, Denver, Colorado, Nov. 17-19, 2003.

[5] Paielli, R. and Erzberger, H., "Conflict Probability for Free Flight," *Journal of Guidance, Control, and Dynamics*, Vol. 20 No. 3, 1997, pp. 588,596.

[6] Sridhar, B. and Chatterji, G., "Computationally Efficient Conflict Detection Methods For Air Traffic Management," *Proceedings American Control Conference*, Albuquerque, NM, June 3-5, 1997.

[7] Krozel, J. and Peters, M., "Strategic Conflict Detection and Resolution for Free Flight," *Proceedings of the 36th IEEE Conference on Decision and Control*, Vol. 2, Institute of Electrical and Electronics Engineers, New York, Dec. 1997, pp. 1822,1828.

[8] Bilimoria, K., Lee, H., Mao, Z., and Feron, E., "Comparison Of Centralized And Decentralized Conflict Resolution Strategies For Multiple-Aircraft Problems," *AIAA Guidance, Navigation, and Control Conference*, Denver, CO, Aug. 14-17, 2000.

[9] Neogi, N., "Using Intent Information To Investigate The Relationship Between Missed Detections And False Alarms In Conflict Detection Verification," *Digital Avionics Systems Conference, 2003. DASC '03*, Institute of Electrical and Electronics Engineers, New York, Oct. 12-16, 2003.

[10] Paielli, R. and Erzberger, H., "Tactical Conflict Detection Methods for Reducing Operational Errors," *Air Traffic Control Quarterly*, Vol.13, No. 1, 2005.

[11] Hubbard, P., "Collision Detection for Interactive Graphics Applications," *IEEE Transactions on Visualization and Computer Graphics*, Vol.1, No. 3, Sept. 1995, pp. 218,230.

[12] Manocha, D., Cohen, J., and Gottschalk, S., "Collision Detection: Algorithms and Application," *Algorithms for Robot Motion and Manipulation*, A. K. Peters, Wellesley, MA, 1997, pp. 129,142.

[13] Suri, S., Hubbard, P., and Hughes, J., "Analyzing Bounding Boxes for Object Intersection," *ACM Transactions on Graphics*, Vol.18, No. 3, 1999, pp. 257,277.

[14] Zomorodian, A. and Edelsbrunner, H., "Fast Software For Box Intersections," *International Journal of Computational Geometry and Applications*, Vol.12, No. 1-2, 2002, pp. 143,172.

[15] Avneesh, S., Govindaraju, N., Gayle, R., Kabul, I., and Manocha, D., "Fast Proximity Computation Among Deformable Models Using Discrete Voronoi Diagrams," *ACM Transactions on Graphics*, Vol.25, No. 3, 2006, pp. 1144,1153.

[16] Langtangen, H., *Python Scripting for Computational Science*, Third Ed., Springer, New York, 2008.

[17] Meyn, L., Windhorst, R., Roth, K., Van Drei, D., Kubat, G., Manikonda, V., Roney, S., Hunter, G., Huang, A., and Couluris, G., "Build 4 of the Airspace Concept Evaluation System," *AIAA Modeling and Simulation Technologies Conference*, Keystone, CO, Aug. 21-24, 2006.

[18] Kopardekar, P., Bilimoria, K., and Sridhar, B., "Initial Concepts for Dynamic Airspace Configuration," *7th AIAA Aviation Technology, Integration and Operations Conference*, Belfast, Northern Ireland, Sep. 18-20, 2007.

[19] Bilimoria, K., and Jastrazebski, M., "Properties of Aircraft Clusters in the National Airspace System," *6th AIAA Aviation Technology, Integration, and Operations Conference*, Wichita, KS, Sep. 2006.

[20] Mueller, T., Sorensen, J., and Couluris, G., "Strategic Aircraft Trajectory Prediction Uncertainty and Statistical Sector Traffic Load Modeling," *AIAA Guidance, Navigation, and Control Conference*, Monterey, CA, Aug. 5-8, 2002.

[21] Lindsay, K., Greenbaum, D., and Wanke, C., "Predeparture Uncertainty and Prediction Performance in Collaborative Routing Coordination Tools," *Journal of Guidance, Control, and Dynamics*, Vol. 28 No. 6, 2005, pp. 1178,1186.