



Trajectory Specification for Terminal Air Traffic: Pairwise Conflict Detection and Resolution

Russell A. Paielli* and Heinz Erzberger†

NASA Ames Research Center, Moffett Field, California, 94035

Trajectory Specification is the explicit bounding and control of aircraft trajectories such that the position at any point in time is constrained to a precisely defined volume of space. The bounding space is defined by cross-track, along-track, and vertical tolerances relative to a reference trajectory that specifies position as a function of time. The tolerances are dynamic and will be based on the aircraft navigation capabilities and the current traffic situation. Assuming conformance, Trajectory Specification can guarantee safe separation for an arbitrary period of time even in the event of an air traffic control (ATC) system or datalink failure; hence it can help to achieve the high level of safety and reliability needed for ATC automation. It can also reduce the reliance on tactical backup systems during normal operation. This paper applies it to the terminal area around a major airport and presents algorithms and software for detecting and resolving conflicts. A representative set of pairwise conflicts was generated, and a fast-time simulation was run on them. All conflicts were successfully resolved in real time, demonstrating the computational feasibility of the concept.

I. Introduction

Air traffic control is currently performed by human controllers using radar displays and voice communication with pilots. The number of flights that a controller can reliably manage at one time, however, is substantially less than the number that could safely fly in the airspace with an automated ATC system.^{1,2} Controllers are remarkably reliable overall, but they are human and therefore make mistakes. Over 1,800 operational errors (breaches of minimum required separation officially attributed to controller error) occurred in one recent year in the US, including 55 serious cases in which “a collision was barely avoided.”³ Automation can reduce human error, but an autonomous ATC system that works for all possible traffic situations and conditions is difficult to design and implement and is even more difficult to verify and validate to the required level of reliability and integrity.

Trajectory Specification is a proposed far-term enhancement of the Advanced Airspace Concept (AAC) being developed by NASA for automating ATC in both enroute airspace^{4,5} and the terminal areas around major airports.^{6,7} The Trajectory Specification concept was first published in 2005⁸ and has since been issued a US patent (US7650232 B1). The main idea is to explicitly bound and control assigned trajectories, which are unbounded in AAC, so that the position at any point in time is explicitly constrained to a precisely defined volume of space. The bounding space is defined by cross-track, vertical, and along-route tolerances around a reference position at any point in time as the flight advances along its route.

*Aerospace Engineer, Code AFT, MS 210-10, Russ.Paielli@nasa.gov, AIAA Associate Fellow

†Senior Scientist, Universities Space Research Assoc., and Adjunct Prof., Dept. of Aero. & Astro., Stanford U.

Trajectory Specification generalizes Required Navigation Performance (RNP)^{9,10} to the longitudinal (vertical and along-track) plane by adding vertical and along-track bounds to the cross-track bounds that are already part of RNP. A Trajectory Specification Language (TSL) based on XML has been developed to represent these specifications and to communicate them by air/ground datalink, and it will be published in a separate paper.¹¹ Another recent paper on Trajectory Specification¹² focused on arrival spacing, and this paper continues the development of the concept by applying it to general conflict detection and resolution in the terminal area.

The possibility of system outages poses a challenge for any ATC automation concept, including AAC. Safety must be maintained at a high level even if the ATC system or the datalink goes down for an extended period of time while traffic density is too high for a human controller to safely take over and manage. One alternative is to stop any new traffic from entering the affected airspace when its ATC system fails.¹³ The current traffic will then exit the affected airspace (or land as planned) within approximately 10 to 15 minutes based on the normal deconflicted and assigned trajectories. Because those trajectories are unbounded, however, conflicts could still arise due to inaccuracies in the winds, weight, or thrust levels that were used to predict the trajectories. The problem could be mitigated by adding an extra separation buffer to the assigned trajectories, but that would diminish airspace capacity during normal operation.

By explicitly bounding deviation from the assigned trajectory in all three axes, Trajectory Specification goes a step further and can guarantee safe separation between equipped flights for as long as they remain in conformance with their assigned trajectories, out to the conflict-free time horizon that was computed. That conflict-free time horizon would normally be on the order of 20 to 30 minutes or more, depending mainly on the current wind modeling accuracy. If the ATC system or datalink goes down, the previously deconflicted and assigned trajectories will remain active in the onboard Flight Management Systems (FMSs) to keep the flights safely spaced and separated.

The Trajectory Specification concept could be adapted for use with current FMSs, but the full concept requires a new generation of FMSs. Other similar concepts have been proposed, but most of them are intended for use with existing FMSs. For example, a “4D” trajectory datalink was proposed by Jackson et al.¹⁴ It allows for altitude bounds at several discrete points, but it does not allow continuous vertical bounds, and it allows a required time of arrival at only one discrete point in the trajectory. While this and other related work¹⁵ constitute important advancements for the near term (5 to 10 years), they do not explicitly bound the position at each point in time and hence cannot guarantee safe separation for an extended period of time in an automated or autonomous system.

Jouliia and Le Talle proposed a “4D contract,”¹⁶ which defines a reference trajectory and elliptical tolerances called “bubbles.” This concept is similar to the Trajectory Specification concept presented in this paper and the original paper,⁸ but a key difference is that the size of the bubbles is fixed, which is overly constraining in light traffic. The dynamic tolerances proposed here, on the other hand, are more flexible and need not constrain the trajectory any more than necessary for the current traffic situation.

Conflict resolution is complicated by the fact that secondary conflicts can arise in the process of resolving a conflict. For testing purposes, however, it is useful to start with the simplified problem of “pairwise” resolution against only one other trajectory. This paper addresses pairwise conflict resolution, and a future paper will address general conflict resolution accounting for all traffic. The main objective is to demonstrate the computational feasibility of pairwise conflict detection and resolution based on the Trajectory Specification concept.

The remainder of the paper is organized as follows. Section II describes the Trajectory Specification concept in more detail, including the ground-based ATC component and the airborne FMS component. Section III explains the application of the concept to conflict detection and resolution in the terminal area, commonly called the TRACON (Terminal Radar Approach Control). After

that, Section IV explains the fast-time simulation tests of pairwise encounters that were used to evaluate the feasibility of the concept, followed by the results and conclusions.

II. Trajectory Specification Concept

Trajectory Specification is essentially the construction of dynamic, virtual roadways, corridors, or tubes in the sky using data standards, an air/ground datalink, and software to specify the parameters. It is more precise, more continuous, more dynamic, and more flexible than the static published routes and discrete altitude restrictions that are currently used to organize traffic and separate arrival streams from departure streams in terminal airspace.

Figures 1 and 2 show examples of a plan view and a side view of trajectory bounds. As the plan view shows, the route in the horizontal plane resembles a “freeway lane in the sky” consisting of straight segments and circular arcs. The lane width is twice the cross-track tolerance and would be consistent with current RNP standards. The along-track bounds at a point in time combine with the cross-track bounds to form a rectangle in the straight segments or a “rounded rectangle” in the turns, as shown. The side view shows the altitude and along-track bounds in the longitudinal plane for a climb. The along-track bounds combine with the vertical bounds to form a shape with vertical sides and curved top and bottom in the longitudinal plane. The vertical tolerances in level flight could be ± 100 or ± 200 ft, but in climb or descent they could be on the order of ± 2000 ft or more, depending on the traffic situation. The tolerances can vary as a function of time or distance, but the function itself will be fixed at the time of assignment (or reassignment).

A specified trajectory as defined here is a stationary (earth-fixed) tube through which the aircraft is required to fly, where the vertical cross-sections are vertical rectangles, and position along the tube is temporally constrained. (These tubes should not be confused with another tube concept that allows many flights to fly in parallel in a single tube like cars on a freeway.) If one such tube goes over or under another tube with sufficient vertical separation, then separation is guaranteed as traffic on a freeway is guaranteed to be separated from traffic on a road that goes over or under the freeway. If two tubes intersect or are separated by less than the minimum allowed separation between flights, then separation must be guaranteed temporally by maintaining the minimum required separation between the bounding spaces or volumes at any point in time.

Trajectory Specification is an extension of trajectory prediction, with tolerances added as shown in Figure 3. Trajectory prediction should normally be done by the FMS, which takes the current state, the flight intent, and wind data as inputs and computes a trajectory prediction based on an aircraft performance model. The FMS then downlinks the predicted trajectory to ATC as a request. ATC takes the predicted trajectory as an input along with any relevant constraints (such as scheduled arrival time) and adds tolerances to produce a Trajectory Specification that constrains the aircraft position to a precisely defined volume of space at any point in time. It then checks the trajectory for conflicts and modifies it (either the reference trajectory, the tolerances, or both) to resolve any conflicts, then uplinks it back to the FMS as the assigned trajectory. The pilot or the FMS can request a new or updated trajectory at any time, and the ATC system should approve it if there are no conflicts. The ATC system will generate a new or updated trajectory and uplink it whenever necessary to resolve a conflict.

A more detailed explanation of the operational concept was presented in the previous paper¹² and will not be repeated here.

The full Trajectory Specification concept requires both a ground-based ATC component and an airborne FMS component. The function of the FMS component is to keep each flight in conformance with its assigned trajectory to within the specified tolerances. The FMS component will require a major development effort that is outside the scope of this paper. The focus of this paper is the ATC component, for which prototype software has been developed using functional programming

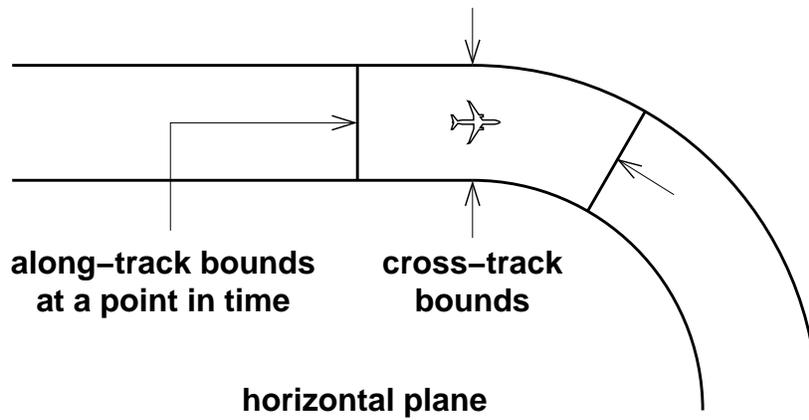


Figure 1. Trajectory bounds in the horizontal plane

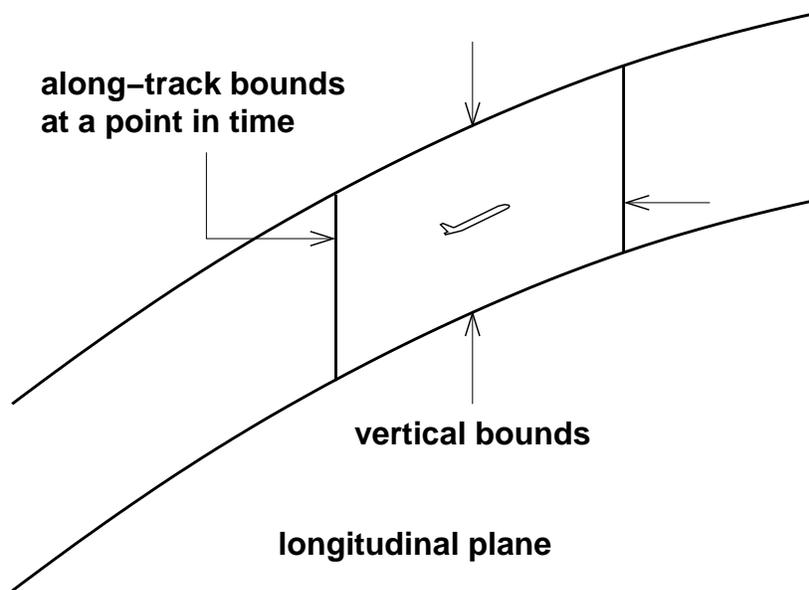


Figure 2. Trajectory bounds in the longitudinal plane

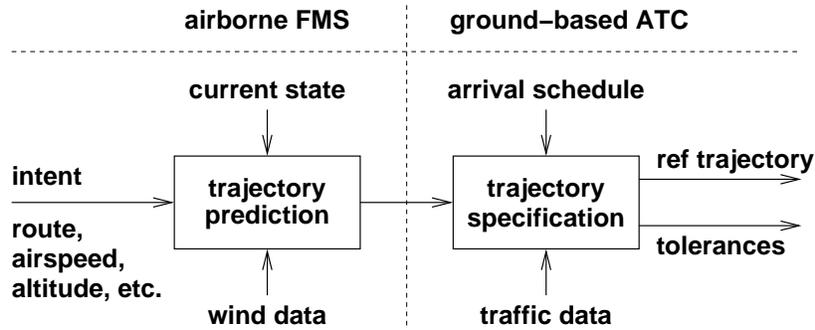


Figure 3. Block diagram of trajectory prediction and specification

with the Scala programming language. The resulting software is not just a “throw-away” prototype but is intended to form the basis for a sound software architecture and design as well as a starting point for an actual operational implementation.

III. ATC Functions and Algorithms

A route is specified as a sequence of waypoints and a turn radius associated with each waypoint. The Trajectory Specification algorithm takes the given waypoints and constructs a detailed route representation consisting of alternating straight and turn segments. All turns are tangent-arc or “flyby” turns of constant radius (similar to the RF turn leg type in the ARINC 424 navigation standard).¹⁹ If waypoints are too close together to accommodate the specified turn radius, the route should be rejected as geometrically invalid. The route representation constitutes a curvilinear route-based coordinate system comprised of Cartesian coordinates for the straight segments and polar coordinates for the turn segments. The 2D coordinates in the locally level frame can then be converted to route-based (along-track and cross-track) coordinates and vice versa.

The altitude tolerance during level flight could be ± 100 or ± 200 ft, but during climb or descent (as shown in Fig. 2) it would typically be much larger, on the order of 1000 to 2000 ft or even more. Altitude tolerances in climb and descent can be specified as a constant or as a piecewise linear function of distance along the route. The upper altitude tolerance need not be the same as the lower tolerance. The altitude tolerance would typically increase during a climb, but in descent it could decrease as the flight nears final approach. Standard landing systems such as ILS (Instrument Landing System) supersede the assigned trajectory on final approach.

Tolerances should never increase or decrease discontinuously or at a higher rate than the aircraft can follow (without causing passenger discomfort). In particular, discontinuities must be avoided at transitions from level to non-level flight and vice versa. Figure 4 shows an example of a simplified reference altitude profile (generated for testing) and the resulting altitude bounds as a function of distance along the route. While the reference altitude profile can be plotted as a function of time or distance, note that the altitude bounds cannot properly be plotted as a function of time because the altitude bounds at a particular time also depend on the along-track deviation from the reference trajectory at that time.

Figure 4 is a side view of the stationary, rectangular tube in which the flight is constrained to fly. Note the tapered transitions between the level and non-level segments and the cutoff of altitude overshoots at leveloff. The tapered transitions are at a slope of 2.5 deg but could be varied slightly.

Note also that the start of descent is clearly bounded. Lack of such bounds is well known to cause significant problems for automated conflict detection, significantly diminishing airspace capacity.²⁰ Discretionary descents in particular (in which the pilot is given discretion as to when to start descent) have also caused problems for automated conflict detection, but they can be accurately represented if necessary by using larger altitude tolerances.

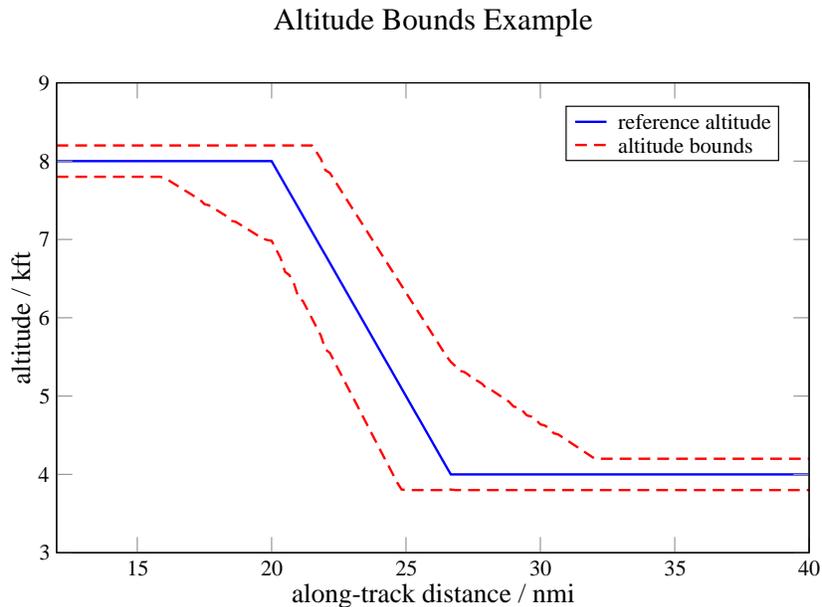


Figure 4. Simplified example of altitude bounds as a function of distance along route

In the horizontal plane, the cross-track tolerance would typically be constant for long distances, but the cross-track and along-track tolerances, like the altitude tolerance, can be specified as a constant or as a linear or piecewise linear function of distance along the route. The along-track tolerances need not be symmetric (front and back) about the reference position. The along-track tolerance would typically increase in enroute airspace, but in heavy arrival traffic it would more likely decrease in descent as the arrival stream converges on final approach and the normal spacing between flights decreases with time as they get closer to landing.

The horizontal bounds form a rectangle in the straight segments, but in turning segments the sides of the rectangle are rounded into arcs as shown in Fig. 1. For purposes of computing the separation between bounding spaces, the circular sides can be approximated as piecewise linear (in angular sections of 30 deg or less). This method approximates the horizontal bounding area as a polygon that fully encompass the turn segments in such a way that the approximated separation may be slightly less than the actual separation but cannot be greater. This guarantee is ensured by using an inscribed polyline for the inner arc and a circumscribed polyline for the outer arc. The accuracy of the approximation is more than adequate for practical purposes.

A. Conflict Detection

Determining the minimum separation between two trajectories with tolerances is much more computationally intensive than determining separation without tolerances. Without tolerances, the predicted separation at a point in time is a simple calculation of horizontal and vertical separation

between two points. With nonzero tolerances, on the other hand, the separation at a point in time is a calculation of separation between bounding volumes at that point in time. Nevertheless, the speed of modern computers combined with a few basic optimization methods allows these calculations to be done in real time as will be shown.

The separation calculations start with a coarse check to avoid unnecessary computation. Given a pair of trajectories, the checks start at the later of the two start times. The reference position of each trajectory at that start time is determined, and the horizontal separation of the reference trajectories is calculated (neglecting tolerances for now). The cross-track and along-track tolerances of each trajectory are then added together and subtracted from that separation to get a lower bound on the horizontal separation between bounding areas. This method is based on the fact that no point in the horizontal bounding area can be further from the reference position than the sum of the cross-track tolerance and the (larger of the front and back) along-track tolerance at that point in time. If the resulting lower bound exceeds the horizontal separation standard by some threshold (e.g., 2 nmi), the horizontal separation need not be calculated.

The next method for avoiding unnecessary computation is to skip ahead in time when the separations are sufficiently large. A maximum speed is found for each flight by looking up the final speed for departures (at or near exit from the terminal area) or the initial speed for arrivals (at or near entrance to the terminal area). Regardless of the relative headings, the sum of those speeds constitutes an upper bound on the closing rate between the two flights. The lower bound on excess separation that was discussed in the preceding paragraph is then divided by this upper bound on closing speed to determine a lower bound on the time at which the horizontal separation could go below the standard. If that time is more than a few seconds ahead, the algorithm skips ahead to that time and repeats the coarse check outlined in the preceding paragraph.

When the upper bound on time to loss of horizontal separation is less than a few seconds, the trajectories are stepped in increments of one second, and a more accurate separation calculation is performed at each time step. The problem is to determine whether any point in the bounding volume for one flight violates separation standards with any point in the bounding volume for the other flight (at the same time). This calculation is not simply a calculation of separation between volumes in space but is further complicated for two reasons. First, the horizontal separation standard of 3 nmi is approximately 18 times larger than the vertical separation standard of 1,000 ft. Secondly, the current separation rules are such that either the horizontal or the vertical separation standard must be met, but if neither one is met, a combination of the two (e.g., the root-sum-square or “hypotenuse” of the two separation ratios) cannot satisfy the requirement.

The polygon approximation discussed above can be used to determine the horizontal separation of the bounding areas, but if the resulting separation is less than the standard 3 nmi, no such analytical method is known for the combined horizontal and vertical separation calculations. The following numerical gridding method was therefore developed.

The horizontal bounding area is gridded, as shown in Figure 5, in increments of 0.5 nmi or less. The grid points are selected in such a way that all four corners and sides are covered. The position and altitude range at each grid point are then calculated and stored. For every possible pair of grid points from one flight to the other, the horizontal and vertical separations are calculated, where the latter is the separation between the altitude ranges at that point (zero if the ranges overlap). A scalar metric called the “separation ratio” is then computed for each possible pair of grid points, as defined below, and the minimum value is recorded.

The separation ratio for a pair of grid points is the maximum of the vertical separation ratio and the horizontal separation ratio. The horizontal separation ratio is the horizontal separation of the two grid points divided by the separation standard of 3 nmi. The vertical separation ratio is the vertical separation divided by the separation standard of 1,000 ft. (If the vertical separation ratio is 1.0 or more while both flights are fully established in level flight, it is arbitrarily increased to 2.)

The separation ratio for the time step is the minimum of the separation ratio for each possible pair of grid points. The separation ratio for the pair of trajectories is the minimum of the separation ratios for each time step. A separation ratio of 1.0 or greater for the entire overlapping time range of the two trajectories means that the minimum required separation is guaranteed as long as the flights are in conformance.

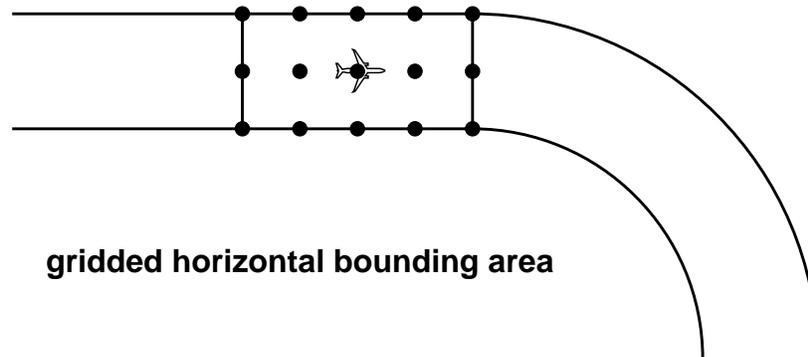


Figure 5. Example of gridded horizontal bounding area

The numerical gridding method can result in a large computational load. For example, if the bounding areas for each of two flights at some point in time each have 15 grid points as shown in Figure 5, the number of pairs is $15 \times 15 = 225$. The positions and altitude ranges are computed only once for each grid point, so the number of position and altitude range computations is 30, but the number of horizontal and vertical separation calculations is 225, which is a lot of computation for one point in time for one pair of trajectories. That computation then needs to be repeated for each time step that the coarse separation checks do not eliminate. That then all needs to be repeated for each pair of trajectories to be checked for conflict. On top of all that, the conflict resolution methods to be discussed later generate many candidate maneuver trajectories that must also be checked for conflict.

The number of grid points for each bounding area is inversely proportional to the square of the grid step size; hence the computational load for each encounter pair is inversely proportional to the fourth power of the grid step size. After some experimentation, a step size of 0.5 nmi was determined to be a good compromise between accuracy and computational load. Using the coarse checks and other optimization methods explained above, along with parallel processing in Scala, the separation computations can be done faster than real time on average, as will be discussed later in the paper. However, the computations can take too long for real time in some difficult cases of conflict resolution with many candidate resolution maneuvers. Many other opportunities for optimization remain, which can be pursued later as necessary. Note also that, with parallel processing, the computation speed should improve as the number of available processor cores increases in the future.

In the terminal area, violations of minimum required separation can be excused for several possible reasons.²¹ For example, if two arrivals are established on final approach on separate, parallel independent runways, the separation requirements do not apply. The same is true of two departures on their initial takeoff leg on different independent runways. Nor do separation requirements apply to an arrival on final approach and a departure taking off on a different parallel runway. A phenomenon called “diverging course” is also considered a valid excuse for a violation

of normal separation requirements. Diverging course occurs when the intersection of the projected courses is behind one or both flights (and the relative course angle is less than 135 deg). Diverging course typically occurs when an arrival turns onto final approach and the separation from the arrival behind it going to the same runway (and still on its base leg) briefly drops below 3 nmi.

The reasons for excusing a separation violation may not apply for the entire horizontal bounding area at a particular point in time. If a bounding area straddles the turn to final approach, for example, part of it may be on final approach and part may not be. Therefore the checks for excused loss of separation must be done for each pair of grid points in the horizontal bounding areas. To avoid inefficient re-computation, various states are stored with each grid point (in addition to the position and altitude range). These states include flags to indicate whether it is on final approach or initial takeoff.

B. Conflict Resolution

The two main requirements for terminal air traffic are arrival spacing and general separation. The spacing required between arrivals to the same runway is usually 3 nmi but can be more, depending on the weight classes of the leading and trailing aircraft. The minimum separation standard in terminal areas is 3 nmi horizontally or 1,000 ft vertically. Arrival spacing, which was discussed in a previous paper,¹² is basically one-dimensional, but general separation is three-dimensional and is therefore more complex. The approach taken here is to first solve the arrival spacing problem because that usually also solves the general separation problem for arrivals in the same arrival stream. Previous studies⁶ indicate that approximately 80-90% of separation conflicts are resolved as a by-product of arrival spacing.

The conflicts that remain after arrival spacing is achieved can be resolved by several different types of maneuvers as will be explained shortly. The resolution maneuvers that will be used in this study are a variation of the maneuvers used in previous work at NASA^{6,7} and are not original in this paper. Although some significant refinements were made to those maneuvers, the main contribution of this work is their implementation within the framework of Trajectory Specification (i.e., nonzero tolerances). As before, the resulting maneuver algorithms are intended to be an example of how conflicts can be systematically resolved, but they are not the only reasonable way. For an actual operational system, an extended refinement process will be required well beyond what can be done in this basic feasibility study.

In the Trajectory Specification concept as applied to the terminal area, conflict resolution would normally be done only once for each flight. For departures, conflict resolution should be done just before takeoff; for arrivals, it should be done shortly before entry into the TRACON airspace. At that point, the flight will be assigned a trajectory through the TRACON that has no conflict with any previously assigned trajectory. Any change to the assigned trajectory after that point, particularly for arrivals, should be relatively rare and in response to nonconformance of some sort. Because departures are less constrained than arrivals, a second maneuver after takeoff could be allowed for a conflict with a new arrival.

The approach to conflict resolution used here is a “scattershot” approach with many candidate maneuvers generated and tested. Several different types of maneuvers are used, including speed reductions, temporary altitude holds, reroutes, takeoff delays, and other maneuver types. For each maneuver type, a list of candidate maneuvers is generated and sorted, if necessary, according to some cost metric. Temporary altitudes are sorted by deviation from the original assigned altitude, for example, and reroutes are sorted by the resulting delay. The list is then iterated through, and the first candidate that realizes a target separation ratio is selected. A separation ratio of 1 or greater resolves the conflict, but a target value of 1.1 was used to add an extra safety buffer.

For efficiency, each maneuver is first checked against the original conflicting flight or flights only, then it is checked against all other traffic if no conflict is found with the original conflicting flight or flights. If no candidate maneuver resolves the conflict, the one with the longest time to loss of separation is stored, and the next maneuver type is tried. If none of the maneuver types resolve the conflict, the candidate with the longest time to loss of separation is selected and a second simultaneous maneuver is tried using the first maneuver as the starting trajectory. The same procedure is repeated for the second maneuver, except that this time the maneuver with maximum separation ratio (rather than the longest time to loss of separation) is stored until a candidate is found that meets or exceeds the target separation ratio. If the conflict is still not resolved after the second maneuver, the resolution is deferred (by putting an arrival into a holding pattern or delaying takeoff of a departure).

The conflict resolution algorithm will always be subject to refinement and modification, and the methods used for arrivals and departures are slightly different, but in general the following maneuver types are tried in the order shown:

- temporary altitude hold
- speed reduction
- reroute
- takeoff delay (for departures)
- other

The “other” category includes various experimental methods that are still being developed for conflicts that are not successfully resolved using the main methods. One such method is base leg extension, in which the length of the base leg is increased (not to be confused with a “trombone” maneuver, in which the base leg is shifted by extending the downwind leg). The following subsections explain each major maneuver type, and examples are shown later in the Results section.

1. *Temporary Altitude*

The first maneuver method tried for both arrivals and departures is a temporary altitude hold during climb or descent. This maneuver type is commonly used by controllers today. The flight is simply held at a temporary altitude, while on its way to a another altitude, until the conflicting flight passes over or under. The candidate altitudes start at the altitude of arrival into, or departure from, the terminal area and step down in increments of 500 ft, with an arbitrary lower limit of 4,000 ft for this study. For each candidate altitude, the hold time is varied in steps of 2 minutes up to 8 minutes. The increments can be made smaller at the expense of more computation time. The first altitude and hold time that realizes the target separation ratio is selected if one is found.

Another maneuver type called a “new level altitude” also involves a change in altitude and is similar to a temporary altitude hold. It is a change in the altitude of the level altitude segment at entry into the TRACON for an arrival or exit from the TRACON for a departure. Usually, but not always, the change is to a higher altitude.

2. *Reroute*

The arrival reroute algorithm starts by generating permutations of three parameters: the turn angle away from the initial arrival leg, the length of the leg in that direction, and the length of the final approach leg. The turn angle away from the initial arrival leg is varied from 0 to 60 deg in each direction in increments of 20 deg. The length of the leg following that turn is varied from

6 to 16 nmi in increments of 2 nmi. The length of the final leg is varied from 8 to 24 nmi in increments of 8 nmi. The increments can be made smaller at the expense of more computation time. Geometrically invalid routes are eliminated (i.e., routes that have turns too large and legs too short to accommodate the specified turn radius, a non-trivial determination). Candidates that go outside of the TRACON boundary are also eliminated. A base leg is added, if necessary, to limit the magnitude of the final turn to 90 deg.

Figure 6 shows an example of the reroute candidates for an arrival. The candidate trajectories are constructed and sorted by the resulting arrival delay. The algorithm then iterates through the candidates and selects the first one that successfully realizes the target separation ratio if one is found. If no candidate meets or exceeds the target separation ratio, the candidate with the largest separation ratio is stored.

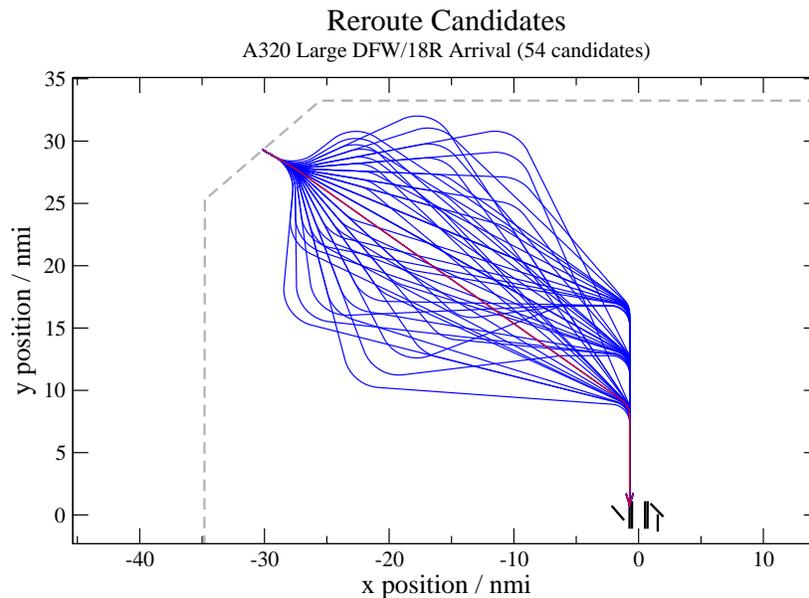


Figure 6. Example of reroute candidates for an arrival

The departure reroute algorithm is similar to the arrival reroute algorithm. It also starts by generating permutations of three parameters: the length of the initial takeoff leg, the turn angle away from the initial takeoff leg, and the length of the next leg before turning back to the departure fix (near the TRACON boundary). The length of the initial takeoff leg is varied from 8 to 24 nmi in increments of 8 nmi. The turn away from the initial takeoff leg is varied from 0 to 60 deg in each direction relative to the original turn angle, in increments of 20 deg, with a turn magnitude limit of 90 deg. As before, the increments can be made smaller at the expense of more computation time. The length of the leg following the turn is varied from 6 to 16 nmi in increments of 2 nmi. As before, geometrically invalid routes and candidates that go outside of the TRACON boundary are eliminated.

Also as before, the candidate trajectories are all constructed and sorted by the resulting delay. The algorithm then iterates through the candidates and selects the first one that successfully realizes the target separation ratio if one is found. If no candidate meets or exceeds the target separation ratio, the candidate with the largest resulting separation ratio is stored.

3. Speed Reduction

Speed reduction is useful for resolving many kinds of conflicts, but it is particularly good for arrivals on independent parallel runways that are in conflict during, or just before, their turns to final approach. The speed reduction applies to the segment of constant CAS (Calibrated Airspeed), typically a few minutes long, which is the initial few minutes in the TRACON for arrivals or the final few minutes for departures. The CAS is reduced in steps of 5 kn to a maximum reduction of 40 kn for arrivals or 20 kn for departures. The speed is held at the reduced value until it drops lower as usual for arrival or until a departure exits the TRACON. If the CAS was already reduced for arrival spacing, it is not reduced to more than 40 kn less than the original CAS.

4. Takeoff Delay

Another maneuver that is available for departures is a delay in the start of the takeoff roll. A takeoff delay is simply a time shift of the entire departure trajectory. In this study, takeoff delays are attempted in steps of 15 sec out to 4 minutes. In terms of fuel efficiency, a delay on the ground costs virtually nothing and is preferable to a delay in the air. However, if a runway departure queue follows behind the delayed departure, the delay will propagate back through the queue. The desirability of this maneuver therefore depends on the state of the departure queue behind the departure to be delayed. If no queue or a short queue follows the departure, a takeoff delay may be the preferred maneuver if it can resolve the conflict. At some airports, departure runways have multiple access ramps for takeoff, in which case the opportunity may exist to reorder the departure sequence to avoid propagating the delay to the departure queue.

IV. Simulation Test Methods

A set of fast-time simulations were run to test the Trajectory Specification concept at the D10 TRACON, which serves the Dallas/Fort Worth (DFW) airport and the older and smaller Dallas Love Field (DAL) airport. Other, smaller airports within this TRACON are not considered in this study. The objective was to determine the computational feasibility of resolving conflicts in the terminal area. Traditional time-stepped simulation was not necessary because, assuming conformance, the Trajectory Specification concept guarantees separation once a deconflicted trajectory is found. Time-stepping is part of the conflict detection procedure for each trajectory as explained earlier and need not be repeated in simulation.

The trajectories used in these simulations were produced with the Kinematic Trajectory Generator (KTG),²² which is a plug-in to the Airspace Concept Evaluation System (ACES),²³ a fast-time simulation program developed for NASA. KTG uses aircraft performance data from the Base of Aircraft Data (BADA)²⁴ developed by Eurocontrol.

The tests to be based on 903 departures and 1001 arrivals, for a total of 1904 flights to and from the DFW and DAL airports. The trajectories are based on actual filed flight plans flown in the D10 TRACON and were produced by running ACES and KTG *without conflict resolution* and storing the resulting trajectory data for a simulation period of 18 hours, from 6:00 am to midnight based on 4/25/2012. These trajectories follow nominal routes that aircraft would fly with no conflicts to resolve. They were used in this study to represent typical downlinked trajectory requests. A key part of the Trajectory Specification concept is to accept such requests, check for conflicts, and modify the trajectories to resolve conflicts when necessary.

The default trajectory tolerances were set as follows. The cross-track tolerance was set to 0.6 nmi, which corresponds to RNP 0.3 (the RNP cross-track bound is twice the nominal RNP value in nmi). The altitude tolerances for arrivals start at ± 500 ft and increase to ± 800 near the turn to final (at which point the glideslope supersedes it). For departures, the altitude tolerances start

at ± 500 ft at takeoff and increase to ± 1000 ft at TRACON exit. The along-track tolerances for arrivals start at ± 1.0 nmi at TRACON entry and decrease to ± 0.2 nmi near the turn to final, corresponding to approximately ± 5 sec in terms of arrival time. For departures, the along-track tolerances start at ± 0.5 nmi and increase to ± 1.0 nmi at TRACON exit. These tolerance values are somewhat arbitrary but “realistic” enough for this preliminary feasibility study. They will eventually be refined based on further analysis and testing. The tolerances could be increased for individual flights if doing so does not result in a conflict, but that possibility was not pursued in this study.

Maneuvers for conflict resolution are simulated in this study by directly manipulating the original trajectory data. Route changes are implemented by adding, removing, and/or moving waypoints. Speed and altitude changes are implemented by modifying the longitudinal profile. The approach taken in this study is based on the assumption of decoupled lateral and longitudinal dynamics, which is normally a reasonable assumption for commercial passenger airplanes. The original longitudinal profile is simply superimposed onto any new route, starting from the runway and going back, and a level, constant-CAS section is added to fill any gap that is left due to a longer route. The altitude as a function of time and distance are therefore identical for the original and the modified trajectory from the runway to back as far as the length of the original trajectory. The same is true for the along-track distance as a function of time. This approach could be refined if necessary to account for some coupling between the lateral and longitudinal dynamics (e.g., a change in speed or climb rate during a turn).

Using the route modification methods discussed earlier, all DFW arrival trajectories used for this paper were first made to go direct to final approach, with an appropriate base leg added if necessary to prevent the magnitude of the final turn from exceeding 90 deg. Some DAL arrivals were not sent direct to final because they would come too close to the DFW airport and possibly interfere with operations there.

As explained earlier, Conflict resolution is complicated by the fact that secondary conflicts can arise in the process of resolving a conflict. For testing purposes, however, it is useful to start with the simplified problem of “pairwise” resolution against only one other trajectory. Jumping to the general traffic problem before solving the pairwise problem can make it more difficult to understand why a resolution algorithm is failing in certain cases. This paper addresses pairwise conflict resolution, and a future paper will address general conflict resolution accounting for all traffic.

A pairwise conflict resolution testing method was developed that uses time shifting to generate almost any kind of pairwise encounter that can possibly result from the stored trajectories. To keep the computation manageable, the method first finds all unique routes from a full day of simulated traffic at the D10 TRACON, then it selects one trajectory as a representative for each unique route, based on the assumption that trajectories on the same route are likely to be similar. The testing procedure then takes every possible pair of those representative trajectories and time-shifts one relative to the other in increments of 30 sec to find nearly all possible conflicts that could occur between those two trajectories for all possible arrival and departure times. Because arrival and departure times can vary arbitrarily, this approach provides a more comprehensive test than simply using the given trajectories without time shifting.

Conflicts between pairs of arrivals to the same runway and pairs of departures from the same runway are discarded because those are spacing conflicts rather than separation conflicts, and the resolution methods are different. Conflicts between two arrivals that occur within 2 minutes of entry into the TRACON are also discarded because the enroute system should not feed such imminent conflicts to the terminal area.

V. Results

Figure 7 shows 52 unique routes in the D10 TRACON, of which 30 are departure routes, and 22 are arrival routes. This TRACON is roughly a square (with clipped corners), and the arrivals enter at the corners of the square while the departures exit through the sides. The DFW arrival routes all go direct to final approach as explained earlier. The number of possible pairs is $52 \times 51/2 = 1326$. For each of those route pairs, the time-shifting method explained above was used to find (nearly) all possible conflicts between that pair trajectories for all possible arrival or departure times. A total of 1325 pairwise conflicts resulted. For each conflict, the flight with the later assignment time was maneuvered to resolve the conflict. The assignment time is defined here as the takeoff time for departures, or 15 minutes before the landing time for arrivals. Every conflict was resolved successfully, and the entire run took approximately 16 minutes, which is slightly less than 1 sec per pairwise conflict (on an Intel processor with 32 cores).

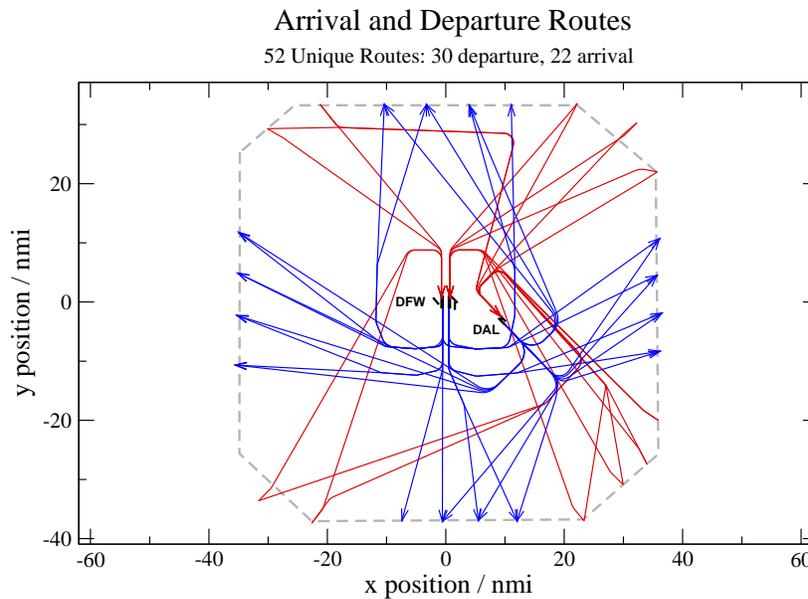


Figure 7. Unique routes for simulated traffic in the D10 TRACON

Table 1 shows counts of the types of resolution maneuvers that were used to resolve the pairwise conflicts. The top row shows the type of the maneuvered flight (arrival or departure), and the second row shows the type of the other (non-maneuvered) flight in conflict. The last column shows the total count for each maneuver type. The counts show that temporary altitudes were the most common maneuver type at 849 total cases. New level altitudes (in which the level altitude segment at TRACON exit for departures or TRACON entry for arrivals is changed to a different altitude) account for another 91 resolution cases. The next most common types of maneuver are reroutes and speed decreases, which were used about equally. In five cases, a combination of two maneuvers were required to resolve the conflict.

Plots were generated for each resolution maneuver, including planview and altitude plots, and examples will be presented shortly. A preliminary review of those plots shows that the maneuvers appear to be acceptable, but a more thorough evaluation can eventually be done by air traffic controllers and other subject matter experts to check whether the maneuvers are appropriate, and

Table 1. Pairwise resolution maneuver type counts

maneuvered:	arr	arr	dep	dep	
other:	arr	dep	dep	arr	sum
temp altitude	11	245	311	282	849
speed decrease	173	0	2	0	164
reroute	55	3	27	79	173
new level alt	5	22	53	11	91
takeoff delay	0	0	40	6	46

that evaluation can be used to refine the resolution methods and algorithms if necessary. However, this more detailed evaluation is outside the scope of this paper.

Figure 8 shows the planview of a temporary altitude hold for a DFW departure in conflict with a DAL arrival. The departure takes off southbound from DFW and turns left. The arrival heads northwest on an extended downwind leg to DAL. The dashed green ovals represent the buffered bounding areas at the original time of the minimum separation ratio of 0.666. By definition, horizontal separation is the minimum required if and when the buffered bounding areas first come into contact. The temporary altitude for the departure was 6,500 ft for 7.0 minutes, which yielded a minimum separation ratio of 1.178. The solid red rectangles represent the horizontal bounding areas of each flight at the point of minimum separation ratio of the resolved trajectories, and the solid green ovals represent the corresponding buffered bounding areas. (Note that the solid green ovals represent a different time than the dashed green ovals.)

Figure 9 shows the corresponding altitude profiles, where the dashed line represents the original trajectory, and the solid lines represent the resolved trajectories. The solid vertical red line represents the time at which the horizontal separation (of bounding areas) drops below 3 nmi, and the dashed green line represents the time at which it goes back above 3 nmi.

Figure 10 shows an example of a departure reroute. A DFW departure takes off to the south and then turns eastward, conflicting with a DAL arrival heading northwest on a long downwind leg. The dashed line represents the original departure trajectory, and the dashed green ovals represent the buffered bounding areas at the original time of the minimum separation ratio of 0.657. As shown in the subtitle of the figure, the shortest resolution maneuver found to meet or exceed the target separation ratio of 1.1 had an initial takeoff leg of 12 nmi followed by a turn of 90 deg left to a leg of 16 nmi then a turn toward the original departure fix. As before, the solid red rectangles represent the horizontal bounding areas of each flight at the point of minimum separation ratio after resolution, and the solid green ovals represent the corresponding buffered bounding areas.

Figure 11 shows an example of a conflict that was resolved by speed reduction. One arrival is coming from the southwest to land at DAL, and the other is coming from the southeast to land at DFW. The DAL arrival follows an indirect route to avoid interfering with DFW operations. The dashed green ovals represent the buffered bounding areas at the original time of the minimum separation ratio of 0.899. (The dashed oval for the DAL arrival is difficult to see because it is almost exactly the same as the solid green oval for the same flight.) The maneuver slowed the DAL arrival by 15 kn, which resulted in a delay of 48 sec. This delay allowed the DFW arrival to pass ahead, increasing the separation ratio to 1.162 and resolving the conflict. The solid red rectangles represent the horizontal bounding areas of each flight at the point of minimum separation ratio of the resolved trajectories, and the solid green ovals represent the corresponding buffered bounding areas.

An important metric for conflict resolution maneuvers is the resulting delay. For the pairwise

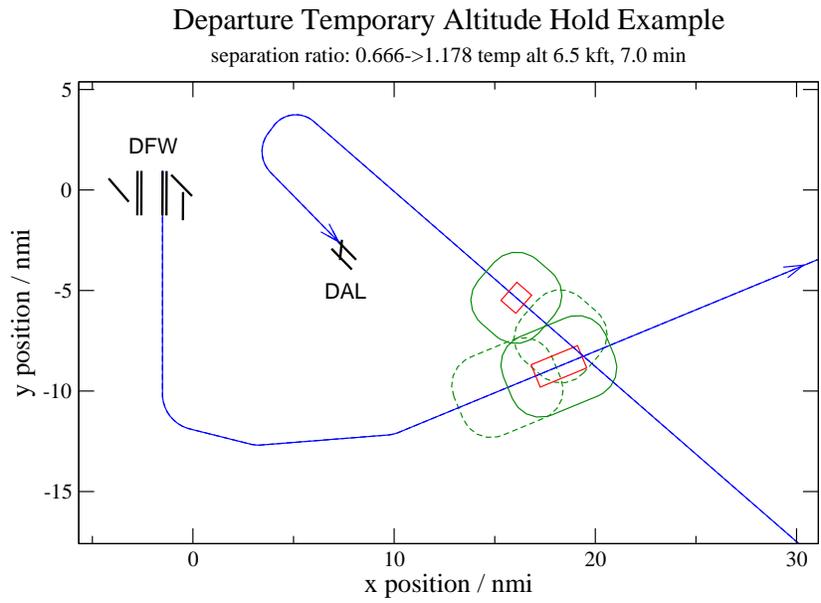


Figure 8. Example of conflict resolution by temporary altitude hold (planview)

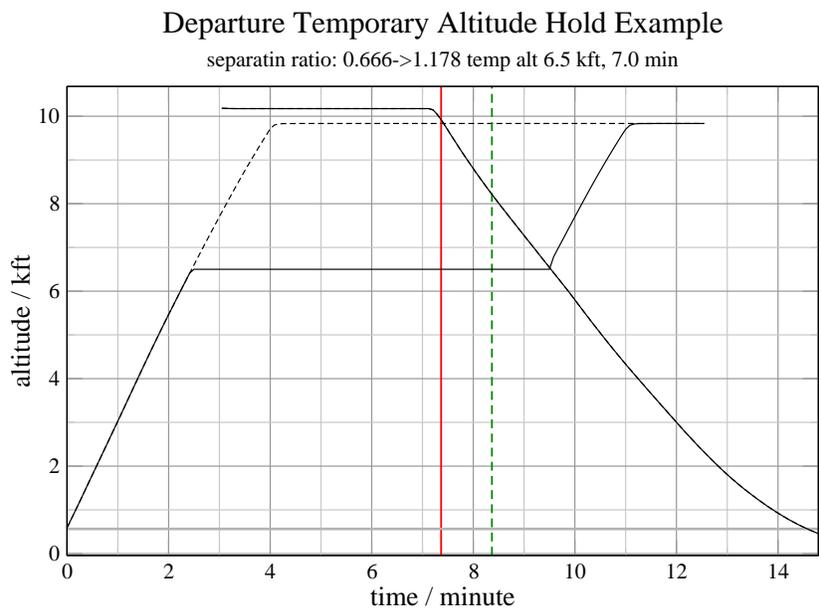


Figure 9. Example of conflict resolution by temporary altitude hold (altitude profile)

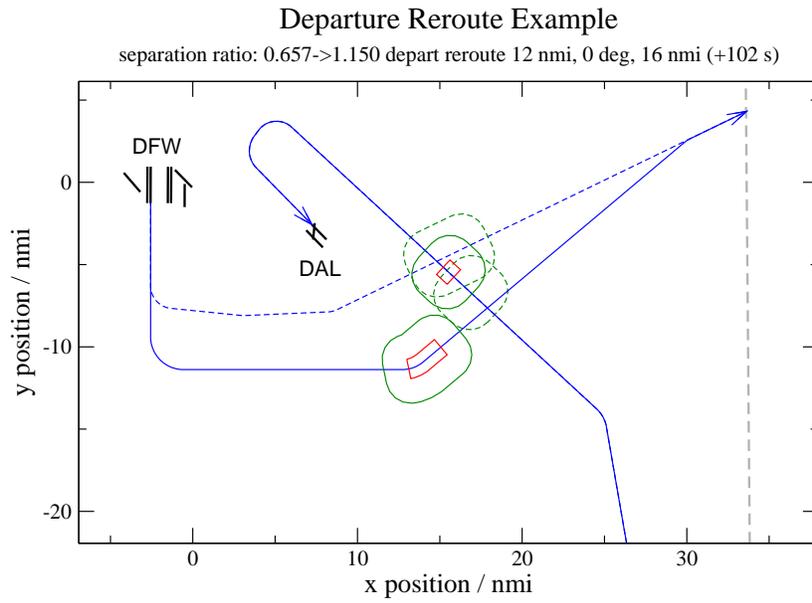


Figure 10. Example of conflict resolution by reroute

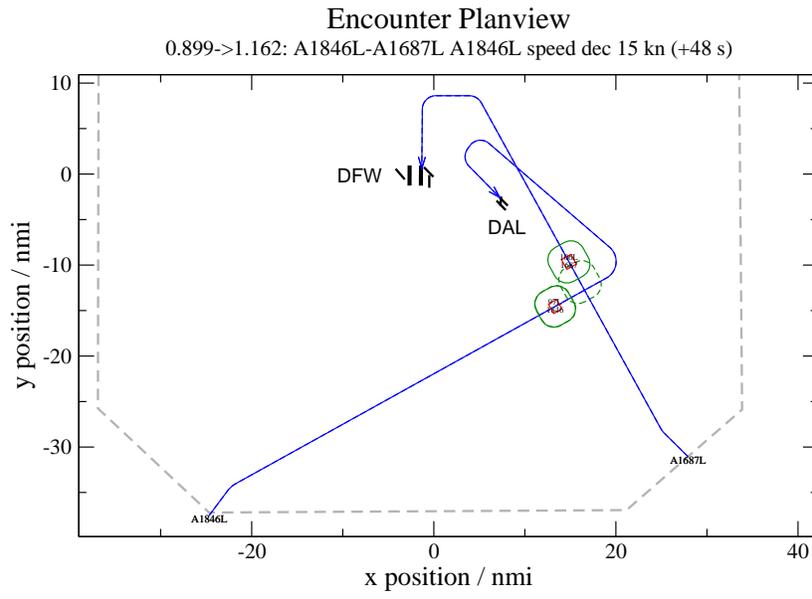


Figure 11. Example of conflict resolution by speed reduction

resolution tests, the mean delay for arrivals was 31 sec and the mean delay for departures was 22 sec. These delays are for conflict resolution only and do not account for the (wake-vortex) spacing delays that would precede these maneuvers if necessary. Moreover, these delays are not representative of the delays that would result in an operational system because they apply to pairwise conflicts only and do not account for other traffic in the area or available landing slots. Note also that they are for flights in conflict and do not include the zero delay for flights in normal operations that have no conflict to be resolved. Nevertheless, these values can be used to compare the effect of varying resolution maneuvers and strategies.

VI. Conclusions

This paper updates the Trajectory Specification concept and applies it to the terminal area around a major airport. The main idea is that aircraft trajectories are explicitly bounded to a precisely defined volume of space at any point in time. It is a generalization of RNP to the longitudinal plane, adding vertical and along-route bounds to the cross-track bounds that are already used in RNP. The tolerances around the reference position are dynamic and will be based on the aircraft navigation capabilities and the traffic situation. Because it can guarantee safe separation for an arbitrary period of time even in the event of an ATC system or datalink failure, Trajectory Specification could be a key to achieving the high level of safety and reliability needed for ATC automation.

The main objective of the paper was to demonstrate the computational feasibility of pairwise conflict detection and resolution based on the Trajectory Specification concept. It focused on the terminal area, using the D10 TRACON serving DFW and DAL airports as a representative airspace. Realistic trajectories from a fast-time simulation were used to represent downlinked arrival trajectory requests, and algorithms were developed to modify trajectories to simulate conflict resolution maneuvers. The maneuver types included speed reductions, temporary altitude holds, reroutes, and takeoff delays. Numerical testing showed that all conflicts were resolved in real time on average, demonstrating the computational feasibility of the concept for pairwise conflicts. A future paper will test the concept in general traffic conditions rather than just pairwise conflicts.

References

- ¹Mercer, J.; Homola, J.R.; Cabrall, C.D.; Martin, L.H.; Morey, S.E.; Gomez, A.N.; Prevot, T.: "Human-Automation Cooperation for Separation Assurance in Future NextGen Environments," *Proceedings of the International Conf. on Human-Computer Interaction in Aerospace (HCI-Aero 2014)*, Santa Clara, CA, 2014.
- ²Prevot, T.; Homola, J.R.; Martin, L.H.; Mercer, J.; Cabrall, C.D.: "Toward Automated Air Traffic Control — Investigating a Fundamental Paradigm Shift in Human/Systems Interaction," *International J. of Human-Computer Interaction*, 28:2, 77-98, Special Issue on NextGen, 2012.
- ³Guzzetti, J.B.: "FAAs Progress and Challenges in Advancing Safety Oversight Initiatives," US Dept. of Transportation, April 16, 2013.
- ⁴Erzberger, H.; Paielli, R.A.: "Concept for Next Generation Air Traffic Control System," *Air Traffic Control Quarterly*, Vol. 10(4)(2002), pp 355-378.
- ⁵Erzberger, H.; Lauderdale, T.A.; Chu, Y.C.: "Automated Conflict Resolution, Arrival Management, and Weather Avoidance for Air Traffic Management," *J. Aerospace Engineering*, 2011 (full ref)?
- ⁶Nikoleris, T.; Erzberger, H.; Paielli, R.A.; Chu, Y.C.: "Autonomous System for Air Traffic Control in Terminal Airspace," *AIAA Aviation Technology, Integration, and Operations (ATIO) Conf.*, Atlanta GA, 16-20 June 2014.
- ⁷Erzberger, H.; Nikoleris, T.; Paielli, R.A.; Chu, Y.C.: "Algorithms for Control of Arrival and Departure Traffic in Terminal Airspace," *Journal of Aerospace Engineering*, DOI: 10.1177/0954410016629499, Feb 2016.
- ⁸Paielli, R.A.: "Trajectory Specification for High-Capacity Air Traffic Control," *AIAA Journal of*

Aerospace Computation, Information, and Communication, vol. 2, no. 9, Sept 2005.

⁹RTCA DO-283A: “Minimum Operational Performance Standards for Required Navigation Performance for Area Navigation,” Oct 2003.

¹⁰RTCA DO-236C: “Minimum Aviation System Performance Standards: Required Navigation Performance for Area Navigation,” June 2013.

¹¹Paielli, R.A.: “Trajectory Specification Language for Air Traffic,” to be published as a NASA Technical Memorandum (TM).

¹²Paielli, R.A.: “Trajectory Specification for Terminal Air Traffic Control: Arrival Spacing,” *AIAA Journal of Aerospace Information Systems*, vol. 13, no. 10, Oct 2016.

¹³Andrews, J.W.; Erzberger, H.; Welch, J.D.: “Safety Analysis for Advanced Separation Concepts,” *Air Traffic Control Quarterly*, vol. 14, no. 1, 2006.

¹⁴Jackson, M.R.C; Gonda, J.; Mead, R.; Saccone, G.: “The 4D trajectory data link (4DTRAD) service - Closing the loop for air traffic control,” *Integrated Communications, Navigation and Surveillance Conf.* (ICNS), Arlington, VA, 13-15 May 2009.

¹⁵Mutuel, L.H.; Neri, P.; Paricaud, E.: “Initial 4D Trajectory Management Concept Evaluation,” *Tenth USA/Europe Air Traffic Management Research and Development Seminar (ATM2013)*

¹⁶Joulia, A.; Le Talle, C.: “Aircraft 4D contract based operation: the 4DCo-GC project,” *11th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference*, AIAA 2011-6859, VA, 20-22 Sep 2011.

¹⁷Paielli, R.A.: “Evaluation of Tactical Conflict Resolution Algorithms for Enroute Airspace,” *AIAA Journal of Aircraft*, vol. 48, no. 1, Jan-Feb 2011.

¹⁸Tang, H.; Robinson, J.E.; Denery, D.G.: “Tactical Conflict Detection in Terminal Airspace,” *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 34, No. 2, 2011, pp 403-413.

¹⁹Aeronautical Radio, Inc.: “Navigation System Database,” ARINC Specification 424-20, Dec 2011.

²⁰Cone, A.C.; Bowe, A.R.; Lauderdale, T.A.: “Robust Conflict Detection and Resolution around Top of Descent,” *12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conf.*, Indianapolis, IN, 17-19 Sept. 2012.

²¹“Air Traffic Control,” FAA Order JO 7110.65V, Federal Aviation Administration, Dept. of Transportation, Washington, DC., 2014.

²²Zhang, Y.; Satapathy, G.; Manikonda, V.; Nigam, N.: “KTG: A Fast-time Kinematic Trajectory Generator for Modeling and Simulation of ATM Automation Concepts and NAS-wide System Level Analysis,” *AIAA Modeling and Simulation Technologies Conf.*, Toronto, Ontario Canada, 2-5 Aug 2010.

²³George, S.; Satapathy, G.; Manikonda, V.; Palopo, K.; Meyn, L.; Lauderdale, T.A.; Downs, M.; Refai, M.; and Dupee, R.: “Build 8 of the Airspace Concept Evaluation System,” *AIAA Modeling and Simulation Technologies Conf.*, AIAA-2011-6373, Portland, OR, 2011.

²⁴Eurocontrol Experimental Centre, “Base of Aircraft Data (BADA) Aircraft Performance Modelling Report,” EEC Technical/Scientific Report No. 2009-009, 2009.