Trajectory Specification for Terminal Airspace: Conflict Detection and Resolution

Russell A. Paielli* and Heinz Erzberger[†]

NASA Ames Research Center, Moffett Field, California 94035

Trajectory Specification is a method of specifying aircraft trajectories with tolerances such that the position at any given time in flight is constrained to a precisely defined bounding space. The bounding space is defined by tolerances relative to a reference trajectory that specifies position as a function of time. The tolerances are dynamic and are based on the aircraft navigation capabilities and the traffic situation. 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. It can help to achieve the high level of safety and reliability needed for ATC automation, and it can reduce the reliance on tactical ATC backup systems during normal operation. This paper presents algorithms and software for detecting and resolving conflicts between specified trajectories in the terminal airspace serving a major airport. In a fast-time simulation of a full day of traffic in a major terminal airspace, all conflicts were resolved in near real time, demonstrating the computational feasibility and the preliminary operational feasibility of the concept.

I. Introduction

AIR traffic control is currently performed by human controllers using radar displays of traffic 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 air traffic control (ATC) system [1,2]. Controllers are remarkably reliable overall, but they are human and therefore they make mistakes. Over 1800 operational errors (breaches of minimum required separation officially attributed to controller error) occurred in one year in the United States, 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 project being developed by NASA for automating ATC in both en route airspace [4–6] and the terminal airspace around major airports [7,8]. The Trajectory Specification concept was first published in 2005 [9] and has been updated in more recent publications [10–12] (and issued a U.S. patent). A similar proposal by others [13] with a more airborne focus followed several years after the first publication on the concept.

The main idea of Trajectory Specification is to limit the allowed deviation from an assigned reference trajectory so that the aircraft position at any given time in flight is constrained to a precisely defined volume of airspace. By explicitly bounding deviation from the assigned trajectory in all three axes, Trajectory Specification can guarantee safe separation between flights for as long as they remain in conformance with the tolerances of their assigned trajectories, out to the conflict-free time horizon that was computed.

A previous paper presented results for Trajectory Specification applied to arrival spacing [11]. This paper presents fast-time simulation results for the full deconfliction process, including spacing and general conflict detection and resolution, for a full day of traffic in the terminal airspace around a major airport. The objective is to determine the computational feasibility and the preliminary

operational feasibility of the Trajectory Specification concept. Can deconflicted trajectories be found for all flights, how much flight delay results, and how long do the computations take?

For background, a description of the Trajectory Specification concept is repeated from earlier papers [12,14] in Sec. II. The ATC functions and algorithms are discussed in more detail in Sec. III. The application of the concept to conflict detection in terminal airspace is described in Sec. III, followed by the conflict resolution methods in Sec. IV. The fast-time simulation test methodology that was used to evaluate the feasibility of the concept is explained in Sec. V, followed by the results in Sec. VI and conclusions in Sec. VII.

II. Trajectory Specification Concept

A. Overview and Benefits

Trajectory Specification is essentially the construction of dynamic clearances in the form of virtual roadways 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. It is made possible by advances in communication, navigation, and surveillance. A Trajectory Specification language (TSL) has been developed to represent these specifications and to communicate them by digital air/ground datalink [14].

A route is the vertical projection of a trajectory onto the surface of the Earth. In the Trajectory Specification concept, a route consists of alternating straight (i.e., great circle) segments and circular turn arcs. Any point along the route can be specified by the distance along the route relative to an arbitrary reference point on the route (positive in the direction of flight), and that distance will be referred to as the along-track distance or position. A useful convention for terminal airspace is to define the runway threshold as the zero reference point, so that departure trajectories start at, and arrival trajectories end at, a zero along-track distance. Then, the along-track position indicates the distance flown from takeoff or the distance yet to be flown to landing.

Figure 1 shows an example of a plan view of trajectory bounds at an instant in time. The lane width is twice the cross-track (lateral) tolerance. The along-track bounds at a given time are vertical rectangles normal to the route direction (which appear as line segments in the plan view) and are defined by the along-track tolerances relative to the reference position at that time. The along-track bounds combine with the cross-track bounds to form a bounding area in the plan view that is a rectangle in the straight segments, an annular area in the turns, or a combination of the two, as shown in the figure.

*Aerospace Engineer, Flight Trajectory Dynamics and Controls Branch; Russ.Paielli@nasa.gov. Associate Fellow AIAA.

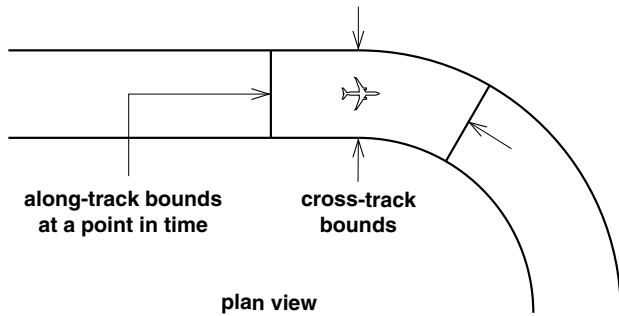†Senior Scientist, Air Traffic Management. Fellow AIAA.

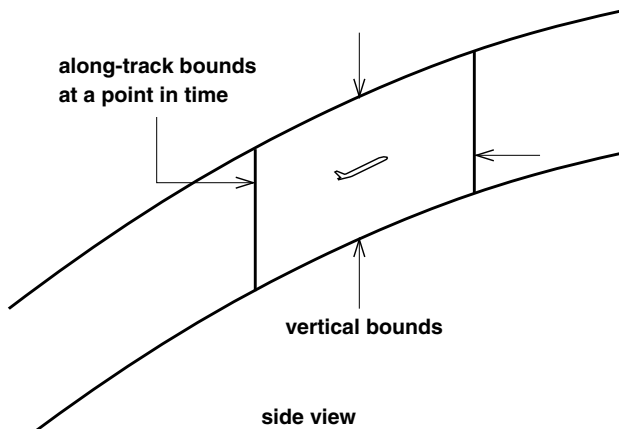**Fig. 1 Trajectory bounds in the horizontal plane.**

Fig. 2 Trajectory bounds in the longitudinal plane.

Figure 2 shows an example of a side view of trajectory bounds at an instant in time during a climb. The along-track bounds combine with the vertical bounds to form a shape with straight vertical sides and curved top and bottom in the longitudinal plane. In level flight, the top and bottom would also be straight. The vertical tolerances in level flight could be ± 200 ft but, in climb or descent, they could be larger, on the order of ± 1000 ft or more, depending on the traffic situation, because altitude is more difficult to predict and control in climb or descent. The tolerances can vary as a function of along-track distance, but the function itself will be fixed at the time of assignment (or reassignment).

The bounding volume at any given time in flight is a segment or "slice" of a stationary (Earth-fixed) bounding tube through which the aircraft is required to fly. Each tube is dynamically constructed for one flight. The vertical cross sections of the tube normal to the route direction are vertical rectangles, and the 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 bounding tube goes over or under another 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 such bounding tubes intersect or are not spatially separated at all times by the minimum separation standards, then separation must be guaranteed temporally within the tubes.

Trajectory Specification generalizes required navigation performance (RNP) [15,16] to the longitudinal plane by adding vertical and along-track tolerances to the cross-track tolerances that are already part of RNP. Dynamic RNP [17] allows routes to be created and assigned dynamically, and it allows discrete altitude constraints and a required time of arrival, but it does not specify a continuously bounded trajectory. Because it allows for a range of possible positions rather than a single position at each time instant, Trajectory Specification is a special application of *differential inclusion*, and the bounding tubes introduced in the following are a special case of *trajectory tubes* [18].

Among the potential benefits of Trajectory Specification is the mitigation of risks that arise in the case of a system outage. Safety must be maintained 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 the traffic. One alternative is to stop any new traffic from entering the affected airspace when its ATC system or datalink fails [19]. The traffic will then exit the affected airspace (or land as planned) within approximately 10–15 min based on the nominally deconflicted trajectories that were previously assigned. Because the deviation from those trajectories is not explicitly bounded, however, conflicts could still arise due to inaccuracies in the winds, weight, or thrust levels that were used to predict the trajectories.

By explicitly bounding deviation from the assigned trajectory in all three axes, Trajectory Specification can guarantee safe separation between flights for as long as they remain in conformance with the tolerances of their assigned trajectories, out to the conflict-free time horizon that was computed. That conflict-free time horizon will normally be on the order of 15–30 min, 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 flight management system (FMS) on board each aircraft to keep the flights safely spaced and separated.

The trajectory will be updated as necessary during the en route phase of flight, and the time between updates will also depend mainly on wind modeling accuracy, but will perhaps be on the order of 10–20 min on average. Many of the updates should simply be time shifts to recenter the bounding volume around the current position of the flight. The trajectory will also be updated for arrivals shortly before they enter the terminal airspace and for departures shortly before takeoff, and it will normally remain unchanged for the entire 10–15 min that a typical flight spends in the terminal airspace.

As a fundamentally proactive rather than reactive approach to ATC, Trajectory Specification can also provide safety benefits during normal operation. Rather than simply relying on continuous conflict detection and tactical maneuvering when necessary to correct for prediction errors, it facilitates more rigorous, precise, and predictable strategic planning. Tactical backup systems [20,21] will still be needed, but they should have to intervene less often. The airborne collision avoidance system will also still be maintained as an emergency backup. Trajectory Specification could also facilitate closely spaced parallel approaches or, eventually, formation landing of more than two flights in closely spaced formations to increase runway landing rates.

B. Operational Concept

Trajectory Specification is an extension of trajectory prediction, which will normally be done by the FMS on board the aircraft. The FMS takes the current flight state, the flight intent, and wind data as inputs and computes a trajectory prediction based on an aircraft performance model. Alternatively, an advanced electronic flight bag (EFB) could potentially be used for this purpose. The intent information includes the route, airspeed, cruise altitude, and possibly other parameters. The FMS can also (optionally) record and take into account the currently assigned trajectories of other flights in the vicinity to avoid conflicts while computing its own trajectory request. Taking other trajectories into account will greatly increase the probability that the requested trajectory will be free of conflicts and approved by ATC without modification. The FMS could use some of the same software components that implement the conflict detection and resolution to be discussed later.

The FMS then downlinks the predicted trajectory to ATC as a request using the Trajectory Specification language [14] mentioned earlier. ATC takes the predicted trajectory as an input and adds default tolerances, which can be constant or a piecewise linear function of along-track position. It then checks the trajectory for conflicts with the current trajectory assignments of other flights and modifies it to resolve conflicts, if necessary, and then uplinks it back to the FMS as the assigned trajectory, again using the TSL.

If a conflict-free trajectory cannot be found, the flight will be delayed until one can be found (by delaying takeoff time for a departure or by putting an arrival into a holding pattern near the terminal airspace boundary). The pilot (or the FMS, if programmed to do so) can request a new or updated trajectory at any time, and the ATC system should approve it if there are no conflicts or constraint violations. The ATC system will generate a new or updated trajectory, or prompt the FMS to generate one, whenever necessary to resolve a conflict.

The basic operational concept can be summarized as follows:

1) The FMS records trajectory assignments from ATC for other flights (optional).

2) The pilot enters route and intent data into the FMS.

3) The FMS computes a deconflicted trajectory prediction.

4) The FMS downlinks trajectory prediction to ATC as a request.

5) ATC assigns tolerances, and it checks for conflicts and constraint violations.

6) ATC modifies the trajectory to resolve conflicts and violations if necessary.

7) ATC uplinks the assigned trajectory with tolerances to the FMS.

8) The FMS flies the assigned trajectory to specified tolerances.

A possible variation of this operational concept substitutes the airline operational center (AOC) or an electronic flight bag for the FMS in all but the last two items. If the AOC is used, the dispatcher or an automated agent would replace the pilot in the second item to enter the desired flight parameters into the AOC trajectory predictor. Using the AOC to generate the trajectory requests would offload the FMS and allow the requests to be communicated to ATC over landlines for improved data transfer reliability and reduced radio frequency usage.

Note that the conflict checks by ground-based ATC are essential for several reasons even if the FMS (or AOC or EFB) is programmed to receive, and avoid conflicts with, all trajectory assignments for other flights. Firstly, the FMS could miss a trajectory assignment to another flight if the assigning ATC system is out of radio range at the time of the assignment or the signal is not accurately received for any reason. Secondly, an assignment to another flight could occur while the FMS (or AOC or EFB) is computing its own trajectory request, resulting in a potentially dangerous race condition. As an additional benefit, the conflict checks on the ground serve as a backup in case of an error in the FMS conflict detection software.

Trajectory tolerances will depend on the aircraft navigation capabilities and the traffic situation. The navigation capabilities determine the lower limit of feasible tolerances, and the traffic situation determines the upper limit. The FMS will know its own minimum tolerances but has no inherent incentive to keep its requested tolerances to a minimum, which is why the tolerances should be provided by ATC from an established database of minimum feasible tolerances for each aircraft equipage class. Alternatively, the aircraft could be required to use its minimum approved tolerances in its trajectory requests. The ATC system can then increase the tolerances if the traffic scenario allows it, but dynamic tolerances are a complex topic that cannot be covered in this paper.

Trajectories in terminal airspace will normally be assigned shortly before entry into the airspace, perhaps 1 or 2 min before entry for arrivals, and perhaps 30 s before the start of takeoff roll for departures. Once assigned, trajectories should normally not be modified for the entire 10–15 min that is typically spent in terminal airspace. In en route airspace, periodic time shifts or other updates can adjust for the accumulated effects of wind errors, and trajectories can be modified as necessary to accommodate weather changes or to resolve any conflicts that arise. Trajectory modifications during flight should allow a lead time of approximately 1–2 min for pilot review and acceptance before the new trajectory actually diverges from the old. That lead time can be reduced if the pilot initiates the change request or if the pilot function is automated in the future.

The concepts and algorithms presented in this paper could be used in the near term for trajectory prediction error budgeting in conflict detection and resolution, but the full Trajectory Specification concept requires both a ground-based ATC component and an airborne FMS component. The focus of this paper is the ATC component, for which prototype software has been developed using functional programming methods in the Scala programming language. The resulting software is intended to form the basis for a sound software architecture as well as a starting point for an actual operational implementation. The airborne component is outside the scope of this paper, but the following brief overview puts it into perspective.

The function of the airborne component is to interpret the Trajectory Specifications and keep the flight in conformance with its assigned trajectory to within the specified tolerances. Due to safety criticality and extreme certification requirements, the airborne component will require a major development effort, but the basic ideas are not complicated. The existing flight-control feedback loop could have a low-bandwidth outer loop wrapped around it to monitor for proximity to the bounds and make adjustments as necessary. Vertical speed would be adjusted to maintain vertical conformance, and airspeed would be adjusted to maintain along-track conformance. The pilot can already make these kinds of adjustments through the mode control panel, but they will probably have to be automated to make the concept feasible.

Ideally, all aircraft will eventually be equipped for the concept, but that will take a long time, and some unequipped aircraft may always be allowed. The basic concept can still be applied to unequipped aircraft, but it would be used for trajectory error budgeting, and the tolerances would then be based on the expected prediction error to some level of probability. Tactical monitoring will then be needed (by a human controller or an automated agent). But tactical monitoring will be necessary anyway, even for equipped aircraft, to verify that they are maintaining conformance, particularly near close encounters. The tactical monitor will issue a tactical maneuver (e.g., an altitude hold or a heading vector) for both equipped and unequipped flights when necessary. A tactical maneuver will take a flight off of its assigned trajectory, and a protocol must be established to get the flight back on another assigned trajectory, but that problem is outside the scope of this paper.

C. ATC Functions

A Trajectory Specification includes a two-dimensional route, which is specified as a sequence of waypoints and a turn radius associated with each waypoint. The Trajectory Specification algorithm takes those route parameters 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 radius-to-fix turn leg type in the ARINC 424-20 navigation standard) [22]. If two successive waypoints are too close together to accommodate the specified turn radius, the route will be rejected as geometrically invalid. The route representation constitutes a curvilinear coordinate system in which the coordinates are along-track distance and cross-track position, which can be converted to geodetic or locally level Cartesian coordinates and vice versa.

The use of circular turn arcs simplifies the necessary coordinate transformations but is not absolutely required for the concept. Constant turn radii could require excessive bank angles for slow aircraft in large turns in high winds. In that case, the turn could be divided into smaller segments of different turn radii by simply placing multiple waypoints (with different associated radii) along the turn (although care must be taken to ensure that the resulting route is geometrically valid). Turns could also perhaps be represented in some other more general mathematical way, but that would be more complicated and will not be discussed in this paper.

The cross-track tolerance will typically be constant for long distances as it is for RNP, but it could change discontinuously at the entry into, or the exit from, the terminal airspace, for example. The vertical and along-track tolerances can be specified as a constant or as a linear or piecewise-linear function of distance along the route. This feature allows for tighter tolerances near encounters with other flights or, to put it another way, looser tolerances away from other traffic. It also allows for vertical tolerances to increase during climb as the cumulative effect of uncertainty naturally increases. The vertical and along-track tolerances need not be symmetric about the reference position. The along-track tolerance will typically decrease during descent in heavy arrival traffic as the arrival stream compresses on final approach and the normal spacing between flights decreases with
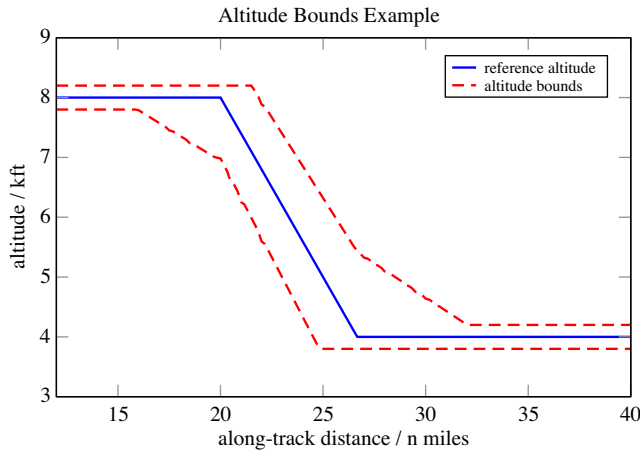
Fig. 3 Simplified example of altitude bounds as a function of distance along route.

time as they get closer to the runway. The vertical and along-track tolerances should never decrease discontinuously or at a higher rate than the aircraft can follow without causing passenger discomfort.

The altitude tolerance during level flight will be ±200 ft, but during climb or descent, it will typically be larger to accommodate larger uncertainties. Figure 3 shows an example of a side view of the stationary, rectangular tube in which an aircraft is required to fly. The Trajectory Specification software automatically detects the level segments and applies the default tolerance of ±200 ft in those segments. It also applies the tapered transitions between the level and nonlevel segments and cuts off the altitude overshoots that would otherwise precede or follow a level segment. The tapered transitions are shown at a slope of 2.5 deg but could be varied slightly. The smaller the taper angle is, the more airspace is reserved. The taper angle should be slightly less than the climb or descent angle to allow enough room for a normal leveloff or start of climb or descent.

Note that the end of climb and the start of descent are clearly bounded. Lack of such bounds causes significant uncertainty for automated conflict detection, diminishing airspace capacity [23]. Discretionary descents in particular (in which the pilot is given discretion as to when to start descent) have caused problems for automated conflict detection, but they can be accurately represented, if necessary, by using larger altitude tolerances.

III. Conflict Detection

Determining the minimum separation between two trajectories with tolerances requires much more computation than determining separation without tolerances. Without tolerances, the predicted separation at a given time is a simple calculation of horizontal and vertical separation between two points. If the horizontal separation meets or exceeds the horizontal separation standard of 3 n miles, or if the vertical separation meets or exceeds the vertical separation standard of 1000 ft, separation is sufficient at that time. With nonzero tolerances, on the other hand, separation is guaranteed to be sufficient at a given time only if every point in the bounding volume for one flight is sufficiently separated with every point in the bounding volume of the other flight. That requires a lot of computation, but a few basic optimization methods can greatly improve computational efficiency.

Alligier et al. [24,25] developed algorithms for efficiently detecting conflicts between bounding volumes. However, their focus was on tactical maneuvers with uncertain pilot response time, and they make assumptions about the shape of the bounding volumes that do not apply to the Trajectory Specification concept. The horizontal projections of their bounding volumes are assumed to be convex, which is inappropriate for turns. Their algorithm also assumes a single altitude range (minimum and maximum) for each bounding volume, which would be inappropriate for climb and descent.

The reference trajectory can be specified with points that are irregularly spaced in time (because closer spacing is appropriate for turns and altitude transients, whereas larger spacing is appropriate for steady-state flight). However, finding position, altitude, altitude range (upper and lower), or other flight variables as a function of time can be done much faster with array indexing if the points are uniformly spaced in time. One method of improving computational efficiency, therefore, is to convert all trajectory variables that need to be accessed as a function of time (or along-track distance) into arrays of uniformly spaced points for fast (constant time) lookup by array indexing. The time increment used for this paper was 3 s. Linear interpolation between adjacent points is also used for better accuracy.

An accurate computation of the minimum separation between bounding spaces is needed only if it is less than or near the minimum required separation. A key to efficient computation, therefore, is to avoid unnecessary computation when the separation is significantly larger than the minimum required, either horizontally or vertically. The following paragraphs explain two kinds of coarse calculations or checks that are used to minimize unnecessary computation: one pertains to spatial separation at a given time, and the other pertains to the time step to the next required spatial separation check. Algorithm 1 shows the simplified high-level pseudocode of the conflict detection algorithm for one trajectory pair and is explained in the following paragraphs.

Given a pair of trajectories, the separation calculations apply to the overlapping time range of the trajectory pair. As shown in lines 4 and 5 of Algorithm 1, that time range extends from the later of the two

---

Algorithm 1   Simplified conflict detection algorithm for one trajectory pair

```
 1:  HSS = 3 n miles // horizontal separation standard
 2:  VSS = 1000 ft // vertical separation standard

 3:  LOS = false // flag to indicate loss of separation
 4:  time = (later of trajectory start times)
 5:  endTime = (earlier of trajectory end times)
 6:  dt = 1 s // fine time step

 7:  while time <= endTime:

 8:      horizSepLowerBound = (calculate horizontal separation lower bound at time)

 9:      if horizSepLowerBound > HSS: time += (calculate "safe" time step)

10:      else: // more accurate and time-consuming calculations needed

11:          horizSep = (calculate horizontal separation at time)

12:          if horizSep < HSS:
13:              vertSepLowerBound = (calculate vertical separation lower bound)
14:              if (vertSepLowerBound < VSS:
15:                  sepRatio = (calculate separation ratio at time)
16:                  if sepRatio < 1.0: LOS = true; time = endTime

17:          time += dt
```

start times to the earlier of the two end times. As indicated by line 8 of Algorithm 1, the first step is a fast calculation of a lower bound on the horizontal separation between bounding areas, which is done as follows:

The reference position of each trajectory at that start time is determined, and the pointwise horizontal separation of the reference trajectories at that time is calculated (neglecting tolerances for now). The along-track and cross-track tolerances will make the horizontal separation of the bounding spaces less than that pointwise separation. An upper bound is determined for the effect of the tolerances by adding together the cross-track and along-track tolerances of each trajectory at that time, and the result is subtracted from the pointwise separation to obtain a lower bound on the horizontal separation between bounding areas as mentioned previously. 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 and along-track tolerances at that time. If the resulting lower bound exceeds the horizontal separation standard of 3 n miles plus some arbitrary buffer (e.g., 2 n miles), sufficient separation is guaranteed at that time, and the exact predicted separation of the bounding areas need not be calculated. The "extra" separation beyond the minimum needed is recorded for use in the next step.

The next method for avoiding unnecessary computation is to skip ahead in time when a quickly calculated lower bound on separation is sufficiently large to guarantee the minimum required separation. This step is shown in line 9 of Algorithm 1 and is done as follows. An estimate of maximum ground speed is determined for each flight, and the maximum speed for each of the two flights are added to obtain an upper bound on the closing speed, regardless of the relative headings. The extra separation at that time, which was calculated in the preceding step as mentioned previously, is then divided by this upper bound on closing speed to determine a lower bound on the time at which the horizontal separation could get near or 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 smaller increments of 1 s, 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 the separation between the bounding volumes because the minimum vertical separation can occur at a different point than the minimum horizontal separation.

Recall that the horizontal bounding area is a rectangle in the straight segments and is an annular section in the turns, as was shown in Fig. 1. Standard geometric calculations can be used to compute the separation between the horizontal bounding areas at a given time. This step is shown in line 11 of Algorithm 1. If the resulting separation is less than or near the standard 3 n miles, no analytical method is known for the combined horizontal and vertical separation calculations. The following numerical method was therefore developed.

In an earlier paper [12], the horizontal bounding area was discretized into a grid of regularly spaced points, but since then, a refinement has been found that improves both accuracy and computational efficiency. Because the reference altitude and altitude range (lower and upper bounds) are independent of cross-track position, the horizontal bounding areas are discretized into line segments in the cross-track direction as shown in Fig. 4. The spacing between the line segments will be discussed later. The line segments are selected in such a way that the front and back along-track bounds are both covered by the selected line segments as shown in the figure (i.e., a smaller increment is used if necessary to cover the ends of the bounding area). The horizontal positions of the end points and the allowed altitude range at those points are then calculated and stored for each line segment.

A coarse vertical separation check is then done as follows. The overall minimum and maximum altitudes are determined for
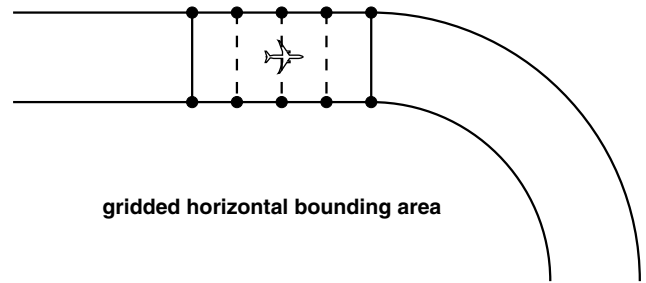

gridded horizontal bounding area

Fig. 4   Example of discretized horizontal bounding area.

each bounding volume by finding the minimum and maximum of the altitude range for each of its line segments. The vertical separation based on these overall altitude ranges constitutes a lower bound on the vertical separation, and if it greater than or equal to the standard 1000 ft, the vertical separation is guaranteed to be sufficient. Otherwise, the following numerical method is used, which is represented on line 13 of Algorithm 1.

For every possible pair of line segments from one flight to the other (at a given time), the horizontal and vertical separations are calculated. The vertical separation is the separation between the altitude ranges at that point in space (which is zero if the altitude ranges overlap). A scalar metric called the "separation ratio" is then computed for each possible pair of line segments, as defined in the following, and the minimum value is recorded. (Determining the separation of two finite line segments in the plane is an interesting problem in itself but will not be discussed here.)

The separation ratio is a basic concept that combines vertical and horizontal separations into a single scalar metric. The separation ratio of a pair of points (or line segments) is the maximum of the horizontal and vertical separation ratios, which are defined as follows. The horizontal separation ratio is the horizontal separation divided by the standard 3 n miles. The vertical separation ratio is the vertical separation divided by the standard 1000 ft. If the vertical separation ratio is 1.0 or more while both flights are fully established in level flight, it is arbitrarily increased (e.g., to 2.0) to account for the fact that altitude is more stable and predictable in level flight. This convention is useful for distinguishing between higher-risk and lower-risk close encounters, and it is useful when the target separation ratio is increased slightly above 1.0 as an added safety buffer (a target value of 1.1 is used later in the paper).

The separation ratio at a time step is the minimum of the separation ratio for each possible pair of points or line segments at that time. The separation ratio for a trajectory pair is the minimum of the separation ratios at each time step of the trajectories. 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 both flights are in conformance with their assigned trajectories.

The numerical method explained previously can result in a large computational load. For example, if the bounding areas for each of two flights at a given time each have five cross-track line segments as shown in Fig. 4, the number of pairs is $5 \times 5 = 25$. The positions and altitude ranges are computed only once for each line segment, and so the number of position and altitude range computations is 10, but the number of horizontal and vertical separation calculations is 25, which is a significant amount of computation for one instant in time for one pair of trajectories. That computation then needs to be repeated for each time step that the coarse separation checks discussed earlier 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 line segments for each bounding area is inversely proportional to the discretized spacing between the line segments; hence, the computational load for each encounter pair is inversely proportional to the square of that spacing. After some experimentation, a spacing of 0.25 n miles was determined to be a good compromise between accuracy and computational load. Using the coarse checks explained previously, along with parallel

processing in Scala, the separation computations can be done much faster than real time on average, as will be discussed later in the paper. However, the computations can take too long for real time for some difficult conflicts with many candidate resolution maneuvers, as will also be discussed later in the paper. 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 terminal airspace, nominal separation standards do not apply in certain situations [26]. For example, if two arrivals are established on final approach on separate, parallel, independent runways, nominal separation standards do not apply. The same is true of two departures on their initial takeoff leg on different parallel independent runways. And, nominal separation standards do not apply to an arrival on final approach and a departure taking off on a different parallel runway. According to official rules [26], a phenomenon called "diverging course" also excuses a violation of nominal separation standards. 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).

Nominal separation standards may apply for only a portion of the along-track bounding range at a given time. If the along-track range encompasses the turn to final approach, for example, part of the bounding area may be on final approach and part may not be. Therefore, the separation requirements must be determined for each pair of line segments in the horizontal bounding areas. To avoid inefficient recomputation, various states are stored with each line segment (in addition to the position and altitude range). These states include flags to indicate whether it is on final approach or initial takeoff.

IV.   Conflict Resolution

The main requirements for terminal air traffic control are to establish and maintain spacing and separation standards. The spacing required between arrivals or departures to a common runway is usually 3 n miles but can be up to 7 n miles, depending on the weight classes of the leading and trailing aircraft. The minimum separation standard in terminal airspace is 3 n miles horizontally or 1000 ft vertically. Arrival spacing, which was addressed in a previous paper [11], is a one-dimensional delay problem, 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 also solves most of the general separation conflicts for arrivals in the same arrival stream to a common runway. Previous studies [7] indicate that approximately 80–90% of conflicts are resolved as a result 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 based on the maneuvers used in previous work at NASA [7,8] and are not original in this paper. Although some 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). The proposed maneuvers are intended as examples of how conflicts can be systematically resolved, but they are not the only reasonable way. For an actual operational system, an extended development and refinement process will be required well beyond what can be done in this basic feasibility study.

In the Trajectory Specification concept as applied to terminal airspace, spacing and conflict resolution should normally be done only once for each flight. For departures, it should be done shortly before takeoff; for arrivals, it should be done shortly before entry into the terminal airspace. At that point, the flight will be assigned a trajectory that is typically 10–15 min in duration through the terminal airspace and 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. Because departures are less constrained than arrivals, a second

maneuver after takeoff could be allowed for a conflict with a new arrival, but that capability was not implemented in this study.

The heuristic approach used here for conflict resolution involves generating and testing many candidate maneuvers. A more analytical approach may be possible but would be very difficult given the complexity of the problem. Several different types of maneuvers are used, including temporary altitude holds, speed reductions, reroutes, takeoff delays, and other maneuver types. For each maneuver type, a list of candidate maneuvers is generated and sorted by the resulting delay. The list is then iterated through, and the first candidate that realizes a target separation ratio (1.1) with a delay less than a threshold (30 s) is selected. A separation ratio of one or greater indicates a successful resolution, but a target value of 1.1 was used to add an extra safety buffer. If no maneuver is found with a separation ratio above the target value and a delay below the given threshold, but one or more maneuvers are found that resolve the conflict, the one with the smallest delay is selected.

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

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: 1) temporary altitude hold, 2) speed reduction, 3) reroute, 4) takeoff delay (for departures), and 5) other.

The order of the maneuver types shown previously is the usual order of preference for efficiency (in terms of time or fuel consumption). An altitude hold is usually, but not always, more efficient than the other maneuver types, for example. The efficiency of an altitude hold is not simple to model, however, and simplified heuristic models are used in this study (based on hold time and deviation from desired altitude). More advanced maneuver efficiency models can be added in future work if necessary.

A.   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 cleared altitude, until the conflicting flight passes over or under. The candidate altitudes start at the altitude of arrival into, or departure from, the terminal airspace and step down in increments of 500 ft, with an arbitrary lower limit of 4000 ft for this study. For each candidate altitude, the hold time is varied in steps of 2 min up to 8 min. 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.

B.   Reroute

The arrival reroute algorithm starts by generating permutations of three parameters: the turn angle away from the initial arrival direction into the terminal airspace, 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 n miles in increments of 2 n miles. The length of

the final approach leg is varied from 8 to 24 n miles in increments of 8 n miles. 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). Candidates that go outside of the terminal airspace boundary are also eliminated. A base leg is added, if necessary, to limit the magnitude of the final turn to 90 deg.

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 terminal airspace boundary). The length of the initial takeoff leg is varied from 8 to 24 n miles in increments of 8 n miles. 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 n miles in increments of 2 n miles. As before, geometrically invalid routes and candidates that go outside of the terminal airspace boundary are eliminated.

For both arrivals and departures, the candidate trajectories are 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 that realizes the largest time to loss of separation or the largest separation ratio is stored, depending on whether it is the first or second attempt at resolution.

C. Speed Reduction

Speed reduction was also used in this study. Although speed increases are also possible, they tend to be more limited in range and were not considered in this study. The speed reduction applies to the segment of constant calibrated airspeed (CAS), typically a few minutes long, which is the initial few minutes in the terminal airspace for arrivals or the final few minutes for departures. The CAS is reduced in steps of 5 kt to a minimum of 1.3 times the stall speed. The speed is held at the reduced value until it drops lower as usual as an arrival gets closer to landing or until a departure exits the terminal airspace.

D. Takeoff Delay

Another maneuver that is available for departures is a takeoff delay, which is a delay in the brake release at the start of the takeoff roll. Takeoff delays are implemented in this study as simple time shifts of the entire departure trajectory in increments of 15 s out to a maximum of 4 min. 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 back through the departure queue. The state of the takeoff queues was not considered in this study in selecting the maneuver type, but the propagation of delay back through the queue was accounted for in the delay statistics to be presented later.

E. Other Maneuver Types

The "other" category includes various experimental maneuver types that are still being developed and tested for conflicts that are difficult to resolve with other maneuver types. One such method is base-leg extension, in which the length of the base leg is increased. The base leg, if there is one, immediately precedes the final leg and is perpendicular to it. (This maneuver should not to be confused with a "trombone" maneuver, in which the base leg is shifted away

from the runway.) Examples of resolution maneuvers are shown later in the Results section (Sec. VI).

V.   Simulation Test Methodology

A fast-time simulation was developed to evaluate the Trajectory Specification concept applied to the terminal airspace controlled by the D10 Terminal Radar Approach Control (TRACON) facility, which serves the Dallas/Fort Worth (DFW) and Dallas Love Field (DAL) airports. (Other smaller airports within this TRACON are not considered in this study.) The main objective was to test the computational feasibility and preliminary operational feasibility of the Trajectory Specification concept applied to the detection and resolution of conflicts in terminal airspace. Arrival spacing based on speed changes and path stretch maneuvers was evaluated in a previous paper [11].

Conflict resolution is complicated by the fact that secondary conflicts can arise in the process of resolving a conflict. A previous paper [12] addressed the simplified problem of "pairwise" resolution against only one other trajectory, and this paper addresses general conflict resolution accounting for all traffic. The tests to be discussed in this paper exercise the entire deconfliction process as it would be used in an operational system, including arrival spacing and general conflict resolution for both arrivals and departures. The interaction of traffic from two different airports in fairly close proximity greatly complicates the problem of finding conflict-free trajectories.

The trajectories used in these simulations were produced with the Kinematic Trajectory Generator (KTG) [27], which is part of the Airspace Concept Evaluation System [28], a widely used fast-time simulation program developed for NASA. The KTG uses aircraft performance data from the Base of Aircraft Data [29] from EUROCONTROL. Winds were not modeled in this study (but the effects of wind modeling errors are implicitly accounted for by the trajectory tolerances). A traditional time-stepped simulation was not necessary because, assuming conformance, the Trajectory Specification concept guarantees separation once a deconflicted trajectory is found and assigned. Note, however, that an independent test for conflicts between the resulting assigned trajectories was done to verify the validity of the results.

The determination of appropriate trajectory tolerances is a major topic in itself. In general, larger tolerances tend to reduce airspace capacity and increase computation time. For purposes of this paper, however, the default tolerances shown in Table 1 and explained in the following were used. The cross-track tolerance was set to a constant value of ±0.6 n miles for both arrivals and departures, which corresponds to a RNP of 0.3 (the RNP cross-track bound is twice the nominal RNP value in units of nautical miles). The other default tolerances were set as follows.

For arrivals, the altitude tolerances are set to a constant value of ±200 ft all the way to final approach, where the glideslope supersedes it. The along-track tolerances for arrivals start at ±0.5 n miles at terminal airspace entry and decrease to ±0.2 n miles near the turn to final approach, corresponding to approximately ±5 s in terms of arrival time. This decreasing along-track tolerance for arrivals is necessary during arrival rushes to maximize arrival rates at the runway but, of course, it could be relaxed during nonrush periods.

For departures, the altitude tolerances start at ±250 ft at takeoff and increase to ±1000 ft at the terminal airspace exit. The along-track tolerances for departures start at ±0.4 n miles and increase to ±1.0 n miles at terminal airspace exit. The increasing tolerance values for departures account for the fact that the prediction error

Table 1   Default trajectory tolerances (±)

	Cross-track	Vertical		Along-track	
		Start	End	Start	End
Arrival	0.6 n miles	200 ft	200 ft	0.5 n miles	0.2 n miles
Departure	0.6 n miles	250 ft	1000 ft	0.4 n miles	1.0 n miles

typically increases with prediction lookahead time, and departures are usually unconstrained when they exit the terminal airspace.

These tolerance values are somewhat arbitrary but are considered "realistic" enough for this preliminary feasibility study. The ultimate selection of default tolerance values for each aircraft model is beyond the scope of this paper. 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. The tolerances could also be modified based on the evolving traffic situation as the flight progresses through the terminal airspace, but that possibility was not considered here either.

The trajectories used in this test are based on simulated flights in the D10 TRACON terminal airspace for a period of 18 h, from 0600 to 0000 hrs based on recorded flight plans on 25 April 2012. The total number of flights was 1943, including 929 departures and 1014 arrivals. The traffic pattern was "south flow," with DFW runways 17C, 18R, 17L, and 13R used for arrivals, and runways 17R and 18L used for departures. The usual arrival and departure fixes near the terminal airspace boundary were maintained, but routes inside the terminal airspace were modified as follows.

All DFW arrival trajectories were first modified to go direct to final approach, with an appropriate base leg added if necessary to prevent the turn to final approach from exceeding 90 deg. Most DAL arrivals were not sent direct to final because they would come too close to the DFW airport and possibly interfere with operations there (by blocking the arrival or departure path of a trajectory to be scheduled later). Departures were also sent directly to their exit fixes, with the equivalent of a base leg added when necessary to keep turn magnitudes from exceeding 120 deg. Departures were allowed to climb continuously to 14,000 ft and exit through the top of the terminal airspace rather than leveling out at the usual exit altitudes of 10,000 to 12,000 ft. Departures were also prespaced (before the simulation run) for the required wake-vortex spacing at takeoff so that the resulting delay statistics were not affected by delays due to departure overscheduling.

The trajectories were scheduled (and deconflicted, if necessary) in order of their trajectory assignment time, which is defined as follows. For departures, the assignment time is 30 s before the takeoff time. For arrivals, the assignment time is 2 min before entry into the terminal airspace. Arrivals were then scheduled and deconflicted in first-come/first-served order by assignment time to each runway except when a large enough gap was found in an arrival sequence and a conflict-free trajectory was found that could make use of the gap to insert an additional arrival into the sequence.

Starting with the first trajectory in the sorted list, each trajectory was spaced as required behind the previous arrival or departure on the same runway (as discussed in the earlier paper [11]). After spacing, each flight was checked for conflicts with all previously assigned trajectories. If the trajectory had no conflict or a successful resolution was found, the resulting deconflicted trajectory was assigned and added to the list of assigned trajectories. If no successful resolution was found, the flight was deferred for later resolution (to simulate a takeoff delay longer than 4 min for departures or a holding pattern of 3 min for arrivals). The objective was to resolve all conflicts with as few deferrals as possible and with acceptable delays.

VI. Results

For reference, a baseline run was executed with the trajectory tolerances all set to zero, which represents perfect trajectory prediction. As expected, this idealized run was much less challenging than a run with realistic tolerances, both computationally and in terms of conflict resolution. In this zero-tolerance run, all conflicts were resolved with no deferred resolutions. The mean arrival delay was 27 s, and the mean departure delay was 6 s. The largest arrival delay was 9 min and 15 s, and the largest departure delay was 4 min and 40 s. This zero-tolerance run took 9.1 min to complete, which is 113 times faster than real time on average for the simulated time span of 18 h.

The main test run was then executed with the trajectory tolerances set to the default values shown in Table 1. All conflicts were

**Table 2   Resolution maneuver type counts (Arr: arrival; Dep: departure)**

	Maneuvered/other				
	Arr/Arr	Arr/Dep	Dep/Arr	Dep/Dep	Sum
Temporary altitude	50	34	58	125	267
Reroute	185	13	9	13	220
Speed decrease	133	2	2	1	138
Takeoff delay	0	0	4	93	97
New level alt	5	5	0	18	28
Base-leg extension	7	1	0	0	8

successfully resolved, but four flights had to be deferred: two DAL departures and two DAL arrivals. An arrival is deferred by putting it into a holding pattern near the terminal airspace boundary for 3 min. Holding patterns are undesirable, of course, but two holding patterns for the entire day are acceptable (especially considering that they were small DAL arrivals). Of all flights, 26.1% required no change to their original trajectory, including 24.7% of departures and 27.5% of arrivals. For seven flights, a combination of two maneuvers types (for the same flight) was required to resolve the conflict (e.g., a takeoff delay combined with a reroute, or a reroute combined with a speed decrease). Of the arrivals, 17.8% had runway changes to reduce delay, and 2.5% found conflict-free slots in existing runway streams.

The entire 18 h of simulated traffic took 45.3 min to process, which is approximately 23 times faster than real time on average (on an Intel Xeon processor with 32 cores at 2.6 GHz). The mean time to process a new trajectory request was 1.4 s, and the maximum time was 64 s. The processing time limit for real time has not yet been determined but is perhaps 5 to 10 s, and so more optimization and/or processing power is still needed for use in real time, but that should be achievable within a few years as processor core counts increase and the resolution algorithms are further developed and optimized.

The resolution maneuver type counts are shown in Table 2. The type before the slash represents the type of the maneuvered flight, and the type after the slash represents the type of the other (non-maneuvered) flight in conflict. For example, "Arr/Dep" means that the maneuvered flight was an arrival and the other flight was a departure. The last column shows the total count for each maneuver type. The most common type of maneuver was the temporary altitude hold, which is consistent with field experience. The next most common maneuver type was the reroute, which was used mostly for arrival/arrival conflicts. The next most common maneuver type, speed reduction, was also used mostly for arrival/arrival conflicts. Takeoff (brake release) delays were the next most common maneuver type. Plots were automatically generated for each resolution maneuver and examined to verify that the selected maneuvers appeared reasonable.

Figure 5 shows the plan view of a temporary altitude hold for a DFW departure in conflict with a DAL departure. The DFW
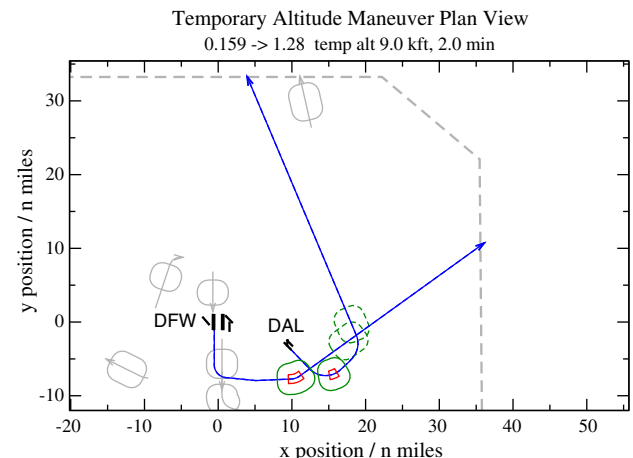
Fig. 5   Example of conflict resolution by temporary altitude (temp alt) hold (plan view).

departure takes off southbound and turns left. The DAL departure takes off heading southeast and then turns left nearly 180 deg to a northwest heading. The overlapping dashed green ovals represent the buffered bounding areas at the original (preresolution) time of the minimum separation ratio of 0.159. By definition, horizontal separation is the minimum required if and when the buffered bounding areas first come into contact (if the tolerances were zero, they would be circles with a diameter of 3 n miles). The temporary altitude for the DAL departure was 9000 ft for 2.0 min, which yielded a minimum separation ratio of 1.28. 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.) The gray ovals represent other traffic at the postresolution time of the minimum separation ratio. Figure 6 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 n miles, and the dashed green line represents the time at which it goes back above 3 n miles.

Figure 7 shows an example of a departure reroute. A DFW departure takes off to the south and then turns right, conflicting with a DFW arrival heading north 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 0.300. 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 8 n miles
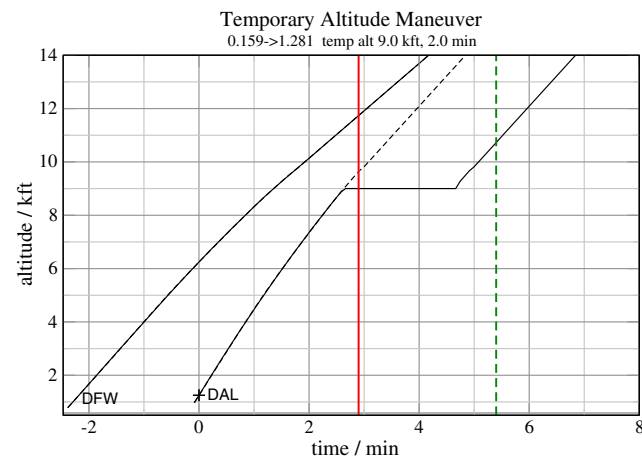

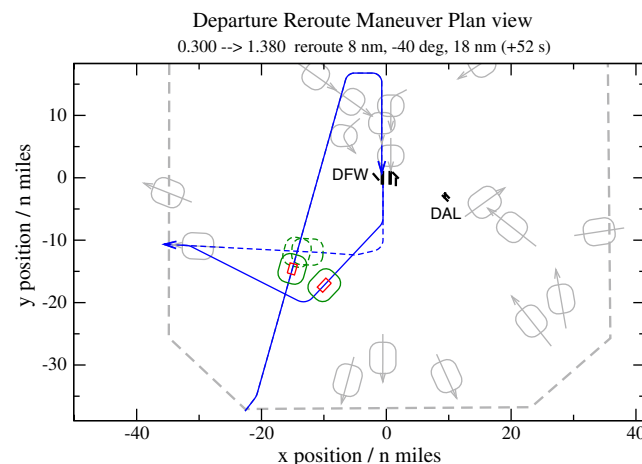**Fig. 6 Example of conflict resolution by temporary altitude hold (altitude profile).**

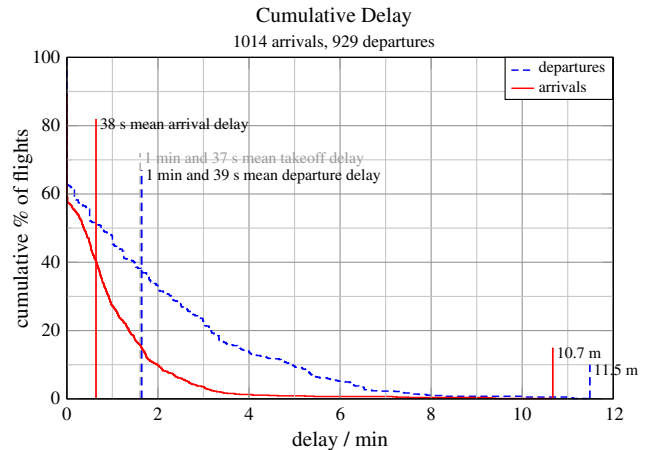**Fig. 7 Example of conflict resolution by reroute.**

Fig. 8 Cumulative delay plot.

followed by a turn of 40 deg right to a leg of 18 n miles, and 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. As before, the gray ovals represent other traffic at the postresolution time of minimum separation ratio.

An important performance metric for conflict resolution is the flight delay resulting from the maneuver. The delay results to be presented in this paper do not account for delays due to demand/capacity imbalances outside the terminal airspace modeled in this paper. Those externally induced delays are often much larger than the internally induced delays for conflict resolution. As mentioned earlier, in the zero-tolerance baseline run, the mean arrival delay was 27 s, and the mean departure delay was 6 s. For reference, another baseline run with wake-vortex spacing only (and no other conflict resolution) resulted in a mean departure delay of zero and a mean arrival delay of 12 s.

Figure 8 shows the cumulative delays for the run with default tolerances, including delays for spacing as well as any delays due to conflict resolution maneuvers. The vertical axis represents the percentage of flights that were delayed by more than the time on the horizontal axis. The mean arrival delay was 38 s, and the mean departure delay was 1 min and 39 s. (These mean delay values are averaged over all flights, whether delayed or not.) The delay for departures was virtually all takeoff delay (delay in brake release for start of takeoff roll). As shown on the plot, the maximum arrival delay was 10 min and 40 s, and the maximum departure delay was 11 min and 29 s.

The resulting delays depend on the trajectory error tolerances, with larger tolerances resulting in larger delays in general. The default tolerances used in this paper (shown in Table 1) resulted in reasonable delays, but those tolerances may be too tight to be acceptable to the aviation community (note that they apply only in the terminal airspace and would be larger for en route airspace). The resulting delays for a given level of tolerances could possibly be reduced with more advanced conflict resolution methods and algorithms. Also, future work can be done on allowing the tolerances to be increased as a function of the current and predicted traffic situation and density, which was not done in this study.

VII.   Conclusions

This paper updates the Trajectory Specification concept and applies it to the terminal airspace around a major airport. The main idea is that aircraft trajectories are explicitly bounded to a precisely defined volume of space at any given time in flight. It is a generalization of required navigation performance (RNP), 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 air traffic control (ATC) system or datalink failure, Trajectory Specification could be a key to achieving the high level of safety and reliability needed for ATC automation.

This paper focused on terminal airspace, using the D10 Terminal Radar Approach Control serving Dallas/Fort Worth and Dallas Love Field airports as a representative airspace. Algorithms were developed to simulate conflict resolution maneuvers, and the maneuver types included temporary altitude hold, reroute, speed reduction, and takeoff delay. Numerical testing demonstrated the computational feasibility and the preliminary operational feasibility of the concept. A few conflicts still take longer than acceptable to compute a resolution maneuver for, but additional optimization and more processor cores should help in the future. Future work is required on selecting tolerances and ensuring that the tolerance and the resulting delays are acceptable to the aviation community.

References

[1] Mercer, J., Homola, J. R., Cabrall, C. D., Martin, L. H., Morey, S. E., Gomez, A. N., and Prevot, T., "Human-Automation Cooperation for Separation Assurance in Future NextGen Environments," *Proceedings of the International Conference on Human-Computer Interaction in Aerospace (HCI-Aero 2014)*, ACM Press, New York, 2014, pp. 1–8. doi:10.1145/2669592.2669644

[2] Prevot, T., Homola, J. R., Martin, L. H., Mercer, J., and Cabrall, C. D., "Toward Automated Air Traffic Control—Investigating a Fundamental Paradigm Shift in Human/Systems Interaction," *International Journal of Human-Computer Interaction*, Vol. 28, No. 2, 2012, pp. 77–98. doi:10.1080/10447318.2012.634756

[3] Guzzetti, J. B., "FAA's Progress and Challenges in Advancing Safety Oversight Initiatives," U.S. Dept. of Transportation, April 2013.

[4] Erzberger, H., "Automated Conflict Resolution for Air Traffic Control," *International Congress of Aeronautical Sciences (ICAS 2006)*, ICAS Paper 2006-8.2.1, Hamburg, Sept. 2006.

[5] Erzberger, H., and Paielli, R.A., "Concept for Next Generation Air Traffic Control System," *Air Traffic Control Quarterly*, Vol.10, No. 4, 2002, pp. 355–378. doi:10.2514/atcq.10.4.355

[6] Erzberger, H., Lauderdale, T. A., and Chu, Y. C., "Automated Conflict Resolution, Arrival Management, and Weather Avoidance for Air Traffic Management," *Journal of Aerospace Engineering*, Vol. 226, No. 8, 2011, pp. 930–949. doi:10.1177/0954410011417347

[7] Nikoleris, T., Erzberger, H., Paielli, R. A., and Chu, Y. C., "Autonomous System for Air Traffic Control in Terminal Airspace," *AIAA Aviation Technology, Integration, and Operations (ATIO) Forum*, AIAA Paper 2014-2861, June 2014. doi:10.2514/6.2014-2861

[8] Erzberger, H., Nikoleris, T., Paielli, R. A., and Chu, Y. C., "Algorithms for Control of Arrival and Departure Traffic in Terminal Airspace," *Journal of Aerospace Engineering*, Vol. 230, No. 9, Feb. 2016, pp. 1762–1779. doi:10.1177/0954410016629499

[9] Paielli, R. A., "Trajectory Specification for High-Capacity Air Traffic Control," *Journal of Aerospace Computing, Information, and Communication*, Vol. 2, No. 9, Sept. 2005, pp. 361–385. doi:10.2514/1.12335

[10] Paielli, R. A., "Trajectory Specification for Automation of Terminal Air Traffic Control," *AIAA Guidance, Navigation, and Control Conference*, AIAA Paper 2016-1868, Jan. 2016.

[11] Paielli, R. A., "Trajectory Specification for Terminal Air Traffic: Arrival Spacing," *Journal of Aerospace Information Systems*, Vol. 13, No. 10, Oct. 2016, pp. 381–392. doi:10.2514/1.I010402

[12] Paielli, R. A., "Trajectory Specification for Terminal Air Traffic: Pairwise Conflict Detection and Resolution," *AIAA Aviation Technology, Integration, and Operations Conference*, AIAA Paper 2017-4256, June 2017.

[13] Finkelsztein, D. M., Sturdy, J. L., Alaverdi, O., and Hochwarth, J. K., "4D Dynamic Required Navigation Performance, Final Report," NASA CR–2011-217051, Feb. 2011.

[14] Paielli, R. A., "Trajectory Specification Language for Air Traffic Control," *Journal of Advanced Transportation*, Vol. 2018, March 2018, Paper 7905140. doi:10.1155/2018/7905140

[15] "Minimum Operational Performance Standards for Required Navigation Performance for Area Navigation," RTCA STD DO-283A, Oct. 2003.

[16] "Minimum Aviation System Performance Standards: Required Navigation Performance for Area Navigation," RTCA STD DO-236C, Washington, D.C., June 2013.

[17] "Dynamic Required Navigation Performance: Preliminary Concept of Operations," Ver. 1.0, Federal Aviation Administration, RTCA Paper 069-14/PMC-1199, Washington, D.C., March 2014.

[18] Kurzhanski, A. B., and Varaiya, P., *Dynamics and Control of Trajectory Tubes: Theory and Computation*, Springer, New York, 2014.

[19] Andrews, J. W., Erzberger, H., and Welch, J. D., "Safety Analysis for Advanced Separation Concepts," *Air Traffic Control Quarterly*, Vol. 14, No. 1, 2006, pp. 5–24. doi:10.2514/atcq.14.1.5

[20] Paielli, R. A., "Evaluation of Tactical Conflict Resolution Algorithms for Enroute Airspace," *Journal of Aircraft*, Vol. 48, No. 1, Jan.–Feb. 2011, pp. 324–330. doi:10.2514/1.C031131

[21] Tang, H., Robinson, J. E., and Denery, D. G., "Tactical Conflict Detection in Terminal Airspace," *Journal of Guidance, Control, and Dynamics*, Vol. 34, No. 2, 2011, pp. 403–413. doi:10.2514/1.51898

[22] "Navigation System Database," Aeronautical Radio, Inc., ARINC Specification 424–20, Annapolis, MD, Dec. 2011.

[23] Cone, A. C., Bowe, A. R., and Lauderdale, T. A., "Robust Conflict Detection and Resolution Around Top of Descent," *12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference*, AIAA Paper 2012-5644, Sept. 2012. doi:10.2514/6.2012-5644

[24] Alligier, R., Durand, N., and Alligier, G., "Efficient Conflict Detection for Conflict Resolution," *ICRAT 2018, 8th International Conference on Research in Air Transportation*, Castelldefels, Spain, June 2018.

[25] Allignol, C., Barnier, N., Durand, N., Gondran, A., and Wang, R., "Large-Scale 3D En-Route Conflict Resolution," *12th USA/Europe Air Traffic Management Research and Development Seminar (ATM2017)*, 2017.

[26] "Air Traffic Control," Federal Aviation Administration, U.S. Dept. of Transportation, Order JO 7110.65V, Washington, D.C., 2014.

[27] Zhang, Y., Satapathy, G., Manikonda, V., and 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 Conference*, AIAA Paper 2010-8365, Aug. 2010. doi:10.2514/6.2010-8365

[28] George, 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 Conference*, AIAA Paper 2011-6373, 2011. doi:10.2514/6.2011-6373

[29] "Base of Aircraft Data (BADA) Aircraft Performance Modelling Report," EUROCONTROL Experimental Centre (EEC) Technical/Scientific Rept. 2009-009, Brtigny-sur-Orge, France, 2009.