Trajectory Specification for Automation of Terminal Air Traffic Control

Russell A. Paielli[*]
NASA Ames Research Center, Moffett Field, California, 94035

"Trajectory specification" is the explicit bounding and control of aircraft trajectories such that the position at each 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. A standard language will be developed to represent these specifications and to communicate them by datalink. 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. As a more proactive form of ATC, it can also maximize airspace capacity and reduce the reliance on tactical backup systems during normal operation. It applies to both enroute airspace and the terminal area around airports, but this paper focuses on arrival spacing in the terminal area and presents ATC algorithms and software for achieving a specified delay of runway arrival time.

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. Likewise, an automated system can deliver arriving flights to a runway properly spaced more consistently than a controller. The problem is that an automated ATC system that works for all possible traffic scenarios 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 far-term enhancement of the Advanced Airspace Concept (AAC) being developed by NASA for automating ATC in both enroute airspace[1,2] and the terminal areas around major airports.[3,4] The trajectory specification concept was first proposed nearly 10 years ago,[5] but advancements in computer technology and communication, navigation, and surveillance (CNS) make the concept more feasible today than it was then. The main idea is to explicitly bound and control assigned trajectories so that the position at each 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 each point in time as the aircraft advances along its route. It generalizes Required Navigation Performance (RNP)[6,7] to the longitudinal plane by adding vertical and along-route bounds to the cross-track bounds that are already used in RNP. A standard trajectory language will be developed to represent these specifications and to communicate them by datalink, but that language is outside the scope of this paper.

---

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

The possibility of system outages poses a challenge for ATC automation. Safety must be maintained at a high level even if the ATC system or the datalink goes down for an extended period of time (e.g., 10 minutes or more) while traffic density is too high for a human controller to safely take over and manage. The solution for AAC is to stop any new traffic from entering the affected airspace when its ATC system fails.[8] The current traffic will then exit the affected airspace (or land) within approximately 10 to 15 minutes on the normal assigned trajectories, which have been deconflicted. Because those trajectories are unbounded in the longitudinal plane, however, conflicts could still arise due to inaccuracies in the weight, thrust, or winds 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 divergence from the assigned trajectory in all three axes, trajectory specification goes a step further and guarantees 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, which could be arbitrarily far out). If chosen well, the tolerances should have a minimal effect on fuel efficiency. The trajectories could become less efficient during a system outage due to a lack of updates (e.g., requiring inefficient airspeed or climb rate due to wind modeling errors), but that is a negligible price to pay for guaranteed safe separation during a rate system outage. Trajectory specification therefore enhances safety and may ultimately be necessary for safe and reliable ATC automation (or partial automation).

As a fundamentally proactive rather than reactive approach to ATC, trajectory specification can also provide safety and capacity benefits during normal operation. Rather than simply relying on continuously updated prediction and tactical maneuvering when necessary to compensate for prediction errors, it facilitates more rigorous and precise strategic planning. Tactical ATC backup systems[10, 11] would still be maintained, but they should need to intervene less often. The airborne collision avoidance (ACAS) system would also still be maintained as a backup.

Joulia and Le Talle proposed a "4D contract,"[9] which defines a reference trajectory and elliptical tolerances called "bubbles." An inner bubble called the "freedom bubble" is an ellipse (centered on the reference position at each point in time) in which the flight is allowed to move freely, and an outer "safety bubble" is a bound for ensuring that a collision cannot occur. This concept is similar to the trajectory specification concept presented in this paper and earlier,[5] but a key difference is that the size of the bubbles is apparently 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.

The remainder of the paper is organized as follows. The next section describes the trajectory specification concept in more detail, including the ATC component and the airborne FMC component. The section after that explains the application of the concept to arrival spacing in the terminal area, followed by a section that presents a set of algorithms for achieving a specified delay of runway arrival time. Finally, conclusions are given.

II. Trajectory Specification Concept

Trajectory specification is essentially the construction of dynamic, virtual roadways in the sky using data standards, datalink, and software to specify the parameters of the roads. It is more precise, more continuous, and more flexible than the static published routes and altitude restrictions that are currently used to separate arrival streams from departure streams in terminal areas.

A specified trajectory 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 in a single tube as 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 volumes at any point in time.

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, where 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. In this case, 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 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 would be fixed at the time of assignment (or reassignment).

Trajectory specification is an extension of trajectory prediction, with tolerances added as shown in Figure 3. Trajectory prediction would normally be done by the FMC, 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 FMC 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 each point in time. It then checks the trajectory for conflicts and modifies it (either the reference trajectory, the tolerances, or both) to resolve the conflicts, then uplinks it as the assigned trajectory. (The pilot could then be allowed to modify the assigned trajectory and send a new request, but that would be a refinement of the basic concept proposed here.)

A standard trajectory language (possibly based on XML) will be needed to represent and communicate these specifications between aircraft and ATC systems, but that language is outside the scope of this paper. Controller-Pilot Data Link Communication (CPDLC)[12] could possibly be used for a simplified form of trajectory specification before the full concept can be fielded, but it cannot specify a continuous trajectory with continuous dynamic tolerances. The trajectory language will be used to downlink trajectory requests and to uplink trajectory assignments, and the FMC will be programmed to understand the language and to keep the flight within the allotted tolerances of the assigned reference trajectory. Periodic updates can adjust for the cumulative effects of wind modeling error when necessary (provided that the update causes no conflict and violates no time constraint).

Trajectory Specification would be used operationally as follows in the terminal area. Basic flight intent information is input to the FMC, including the route waypoints, the altitude at entry into or exit from the terminal area, and possibly an assigned arrival time at the runway. This intent information could be entered into the FMC by the pilot, or it could be uplinked from ATC to the FMC. The FMC then computes an efficient trajectory prediction that meets all airspace constraints and downlinks it to ATC as a request. The ATC system receives the trajectory request, assigns tolerances based on the aircraft navigational capabilities and the current traffic situation, and checks for spacing and separation conflicts. If no conflicts are found, ATC uplinks the resulting trajectory specification as an assignment; otherwise ATC modifies the trajectory and/or its tolerances (or possibly modifies the assigned trajectory of another flight) to resolve conflicts before uplinking it.

For departures, this process could start before take-off and be used to determine a conflict-

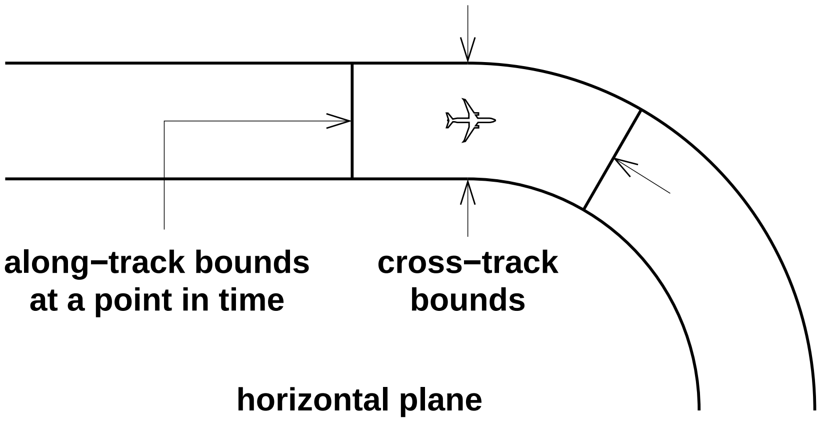

Figure 1. Trajectory bounds in the horizontal plane

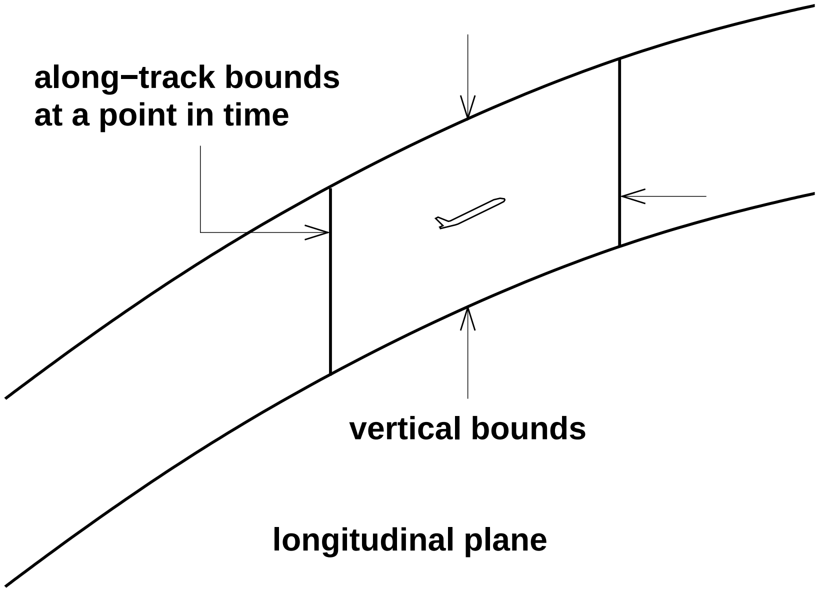

Figure 2. Trajectory bounds in the longitudinal plane

free takeoff time. Also, in a more advanced version of the concept, the FMC could receive the trajectory assignments for other flights and avoid conflicts with them when generating its own trajectory requests. However, those trajectories would still be double-checked for conflicts by ATC in case the information available to the FMC pertaining to other flights is missing, incomplete, superseded, or erroneous, and also in case the FMC software does not function correctly.

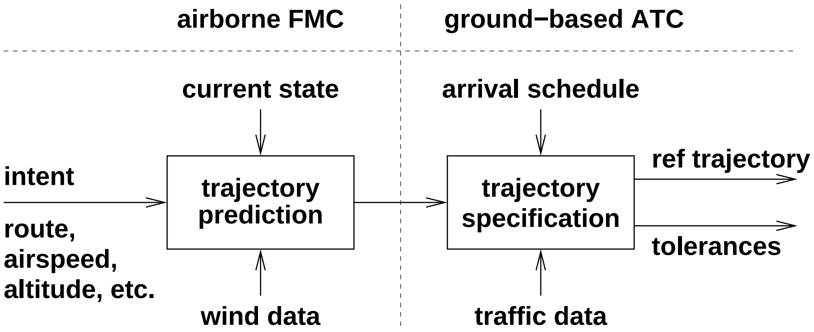

Figure 3. Block diagram of trajectory prediction and specification

In order to guarantee separation, conformance to the assigned trajectories must be guaranteed. If a descending flight is to pass under a climbing flight, for example, the climbing flight must be able to climb at a sufficient rate to stay above its lower altitude bound. If the wind data or the air density that were used to predict the trajectory were grossly in error, or if engine performance is degraded due to a mechanical problem, the climbing flight could drop below its lower altitude bound, and separation might then no longer be maintained. The probabilities of such events should be taken into consideration in determining appropriate vertical bounds, but such considerations are outside the scope of this paper.

A trajectory request consists of a route and a trajectory. The route is specified as a series of waypoints, where each waypoint is a position on the surface of the earth. An explicit turn radius is also associated with each waypoint except the first and last. The trajectory is specified as a series of points of sufficient density to accurately define the trajectory, where each point consists of a time and a position (including altitude). The trajectory must be consistent with the route, of course, meaning that the cross-track errors are zero or below some reasonable threshold.

The input series of points that specify the trajectory need not be equally spaced in time, but they will be converted to equally spaced as discussed later. The waypoints and positions can be specified in geodetic coordinates or, in the terminal area, locally level coordinates. If they are given in geodetic coordinates (latitude and longitude) they will be converted to locally level Cartesian coordinates, which are mathematically more convenient and more efficient to work with for conflict detection and resolution. The locally level coordinate system could be based on any of several standard map projections, such as stereographic or gnomonic (gnomonic was used in this study). Terminal areas are usually small enough that the inevitable distortion due to the map projection is negligible.

Trajectory tolerances would depend on the aircraft navigation capabilities and the current traffic situation. The navigation capabilities would determine the lower limit of feasible tolerances, and the current traffic situation would determine the upper limit. In general, longitudinal tolerances would be made as large as reasonably possible while guaranteeing safe separation. The tolerances could perhaps even be completely disabled when they are unnecessary, although arbitrarily large tolerances would be practically equivalent. The thrust and airspeed adjustments that are necessary to maintain conformance should be relatively small except in rare cases when the wind model used to generate the reference trajectory was grossly in error. Periodic updates can adjust for the accumulated effects of wind errors. If the tailwind is stronger than predicted, for example, the along-track bounds can be shifted ahead periodically to re-center the flight if the shift causes no conflict and violates no time constraint.

The selection of appropriate tolerances for any given aircraft type and traffic situation is too large of a topic to be discussed in detail in this paper. Several issues are involved, including the probability of conformance for a given value of tolerance. That probability should be high, but exactly how high, and how it would be determined, are both open questions that are likely to require further research and analysis. The current lack of explicit tolerances leaves the effective tolerances ambiguous, however, and airspace capacity and safety are not likely to be maximized with ambiguous tolerances.

Before proceeding, it is worthwhile to forestall some possible misunderstandings. The trajectory specification concept does not mandate precise trajectories when they are not needed. It is simply a way to explicitly specify whatever level of trajectory precision is deemed appropriate for a particular aircraft model in a particular traffic situation. The along-route and altitude tolerances can be as large as the traffic situation will permit without a conflict. And if a particular aircraft is not equipped to conform to along-route and/or altitude tolerances, those tolerances can simply be disabled for that aircraft (and aircraft that *are* capable of conforming could perhaps be given preference in conflict resolution). How such unequipped aircraft would be managed by ATC is beyond the scope of this paper, but note that they would be no different than any aircraft in the Advanced Airspace Concept[2,3] without trajectory specification.

A possible objection to the trajectory specification concept is that it continuously constrains the trajectory throughout the entire route even though potential conflicts may be present over only a small portion of it. Whether this issue would really be a problem in practice is not yet known, but if it is, several alternatives are available to solve the problem. One alternative is to make the tolerances larger when no other traffic is around and tighter during encounters. Care should be taken, however, to avoid discontinuous or abrupt tightening of tolerances. Another alternative would be to stipulate that a breach of tolerances is not enforced if no conflict occurs as a result (and no other flight has to deviate from its assigned trajectory to avoid a conflict). Yet another possibility would be to explicitly specify the ranges of along-track distance in which conformance is mandatory.

The trajectory specification concept requires both an ATC component and an airborne FMC component, which are discussed in the next two subsections. Prototype software is currently being developed for the ATC component but is not yet being developed for the FMC component.

A. ATC Component

The first step of the trajectory specification algorithm is to construct 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[13]). If waypoints are too close together to accommodate a particular turn radius, the route should be rejected. 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 locally level coordinates can then be converted to route-based (along-track and cross-track) coordinates and vice versa.

The next part of the algorithm is to convert the trajectory data into a set of fast interpolation functions for several flight variables as a function of time and distance along route. For computationally efficient conflict detection and resolution, several flight variables must be computable in minimal time, including along-track bounds as a function of time and altitude bounds as a function of along-track distance. In the earlier paper,[5] polynomial approximation was proposed, but in retrospect that was a poor choice because real trajectories usually cannot be modeled well with polynomials (unless they are divided into several segments, which adds complexity). In this study, a relatively simple array indexing and interpolation function is constructed as follows.

As explained earlier, the requested trajectory is provided as a series of (time and position) points that are not necessarily equally spaced in time. Those points are first converted to route-based coordinates of along-track distance as a function of time. A series of equally spaced times are then generated to span the time interval from the beginning to the end of the reference trajectory, where the constant time step should be somewhere in the range from 2 to 5 seconds (5 seconds was used in this study). The input trajectory data is then "sampled" at those times by finding the input points with the closest bounding times and interpolating between them as necessary.

Once the interpolated points with equal time steps are computed, the relevant variable as a function of time or along-track distance is determined by a fast array lookup function. If the start time of the trajectory is t_0, and the time step is Δt, then the array index corresponding to time t is simply $(t - t_0)/\Delta t$. If that value is not an integer, the values of the array at the two closest bounding integer indices are interpolated. This procedure provides a fast (constant-time) lookup and interpolation of the relevant variables as a function of time or distance.

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 3000 ft. Altitude tolerances in climb and descent can be specified as a constant or as a linear or piecewise linear function of distance along the route. (More general nonlinear functions could also be used, but their usefulness would be unlikely to justify the increased complexity.) The upper altitude tolerance need not be the same as the lower tolerance. The altitude tolerance would typically increase during a climb or descent, but in descent it could also 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 non-level to 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 essentially a side view of the stationary, rectangular tube in which the flight is constrained to fly, as discussed earlier. Note the tapered transitions between the level and non-level segments and the cutoff of overshoots. The tapered transitions are at a specified slope in the range of approximately 2 to 3 deg (a slope of 2.5 deg is shown in the figure). 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.[14] 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 by using larger altitude tolerances.

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 normal spacing between flights decreases with time as they get nearer to final approach.

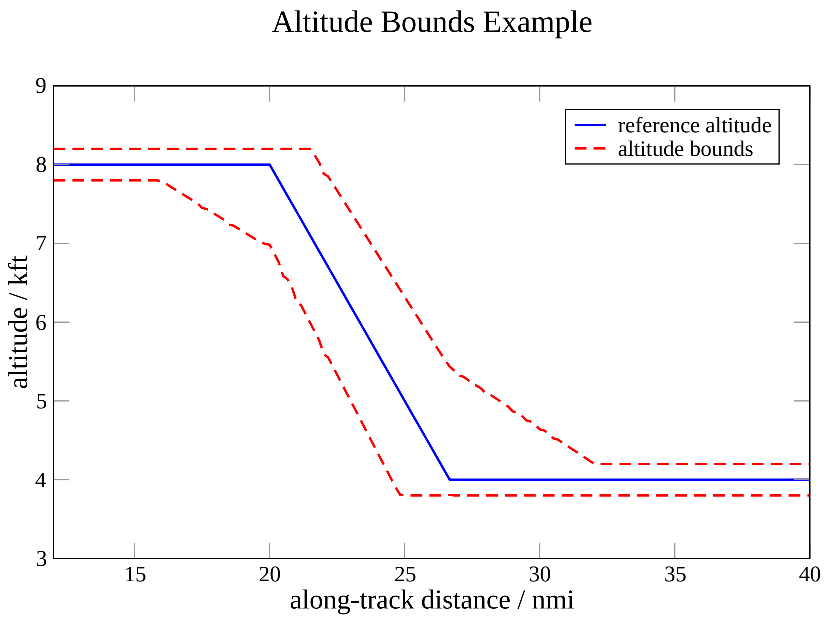

Figure 4. Simplified example of altitude bounds as a function of distance along route

B. Airborne FMC Component

The trajectory specification concept will require a new generation of FMCs and some changes to current flight control methodology. Whereas *any* change to flight control methods and systems requires a substantial effort, including certification, these particular changes are technically feasible and relatively basic. No technological breakthroughs are needed. The latest generation of FMCs can already conform to specified altitude bounds at a specified position, and they can control to a specified arrival time at a meter fix or runway. The trajectory specification concept simply generalizes thise capabilities to the entire route.

The only required change to lateral flight control (i.e., turning) is the use of circular turn arcs of constant radius. Constant-radius turns are not absolutely necessary for the concept, but they simplify the precise definition of the route, and many FMCs in current use can already fly them.[6, 13] Note that a coordinated turn at constant radius in a wind field requires a varying bank angle. If constant-radius turns are ultimately not considered acceptable, alternatives are available.

More extensive changes are required for longitudinal flight control to stay within the altitude and along-track bounds in the longitudinal plane as shown in Fig. 2. Figure 5 shows a simplified block diagram of aircraft longitudinal control. The inputs are the desired airspeed (CAS and/or Mach) and the desired rate of change of altitude; the outputs are the altitude and the distance along track (along route). The main feedback variable is the measured airspeed in terms of CAS or Mach (CAS at lower altitudes, including the entire terminal area, or Mach above the CAS/Mach crossover altitude). The main control variable is the elevator angle, which is used to maintain the commanded airspeed by varying the pitch angle. In level flight the altitude rate is also fed back and nulled to keep the flight level. For most aircraft, however, no attempt is made to close the loop on the rate of change of altitude (vertical speed or flightpath angle) in climb or descent (except during final approach). Instead, the throttle is set to some fixed position (or a variable schedule), and any reasonable resulting altitude change rate is considered acceptable. The throttle and elevator settings are the inputs to the engine and airframe as shown in Fig. 5.

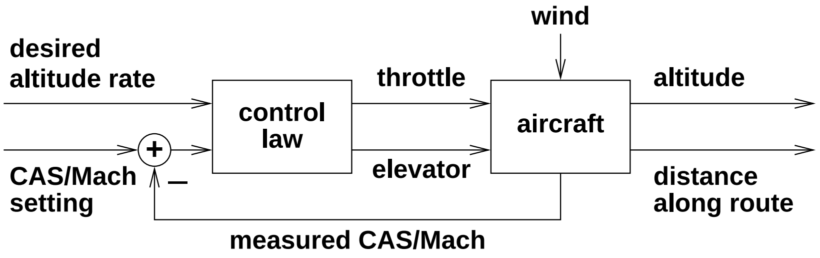

Figure 5. Simplified aircraft longitudinal control

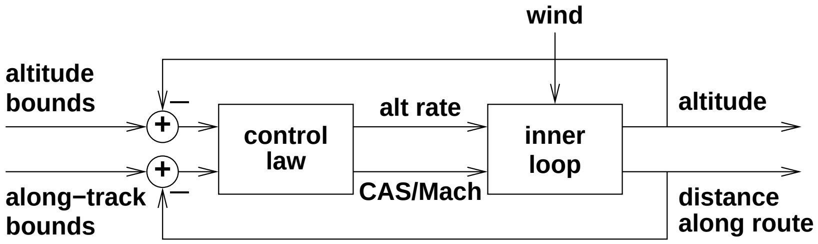

Figure 6. Aircraft longitudinal control with a low-bandwidth outer-loop added to bound altitude and along-track distance (the inner loop represents a simplified version of Fig. 5)

This open-loop approach to longitudinal flight control is adequate for an individual flight with no other traffic around, but the resulting trajectory prediction uncertainty causes problems for conflict detection and can ultimately limit airspace capacity. It can also cause problems in the case of ATC system failure as mentioned earlier because the trajectories are not guaranteed to be free of conflicts for any specific length of time.

Errors in the wind data cause errors in the FMC prediction of ground-speed and vertical speed, which integrate over time to become errors in altitude and along-track distance. If the errors start to approach the allowed tolerances in their respective axes, adjustments in airspeed and/or thrust are needed to keep the flight within its assigned trajectory bounds. Airspeed mainly affects the along-track error, and thrust mainly affects the altitude error, but some cross-coupling between the two axes may occur. Those adjustments should be at a relatively low rate to avoid excessive engine transients and passenger discomfort. (How close the errors should be allowed to get to their bounds before an airspeed or thrust adjustment is made is an important question but is outside the scope of this paper.)

Figure 6 shows a simplified block diagram of such a control system, where the block labeled “inner loop” is a simplified version of Fig. 5. This design “wraps” an outer (low-bandwidth) feedback loop around an existing flight control system, with no changes to the existing system. Other designs involving more fundamental changes to the flight control system may also make sense, but they would be more complicated and are outside the scope of this paper.

A possible near-term application of trajectory specification would be as an advanced separation buffer. In that application, the tolerances would not be enforced by the FMC but would simply provide a buffer to account for uncertainty in the process of checking for conflicts between predicted trajectories (including trajectories that have already been assigned and trajectories with maneuvers being considered for conflict resolution). This approach could account for conflict geometry better than the usual approach of simply adding an altitude buffer and a horizontal separation buffer (e.g., requiring 4 nmi horizontal separation rather than the minimal 3 nmi).

A more advanced near-term application that might also be worth considering would send low-rate guidance commands or altitude constraints through CPDLC[12] or voice synthesis. That approach would essentially replace the control law shown in Fig. 6 with CPDLC messages from the ATC system. A few airspeed adjustments in the TRACON could improve arrival time accuracy, for example, and altitude bounds at specified points could be used instead of thrust adjustments to maintain altitude conformance. Such updates could significantly improve trajectory accuracy and thereby increase airspace capacity, but they could not guarantee proper spacing and separation as the full trajectory specification concept could. Yet another nearer-term alternative might be to provide RNP alerting and graphical guidance, showing the aircraft position relative to the bounds, and allowing the flight crew to close the loop by adjusting thrust and/or airspeed.

The certification of an FMC model is an expensive and lengthy process that can take several years. The FMC that is originally installed on an aircraft typically stays on that aircraft for its entire service life of 30 years or more. As a result, installed FMC computers typically lag commercial off-the-shelf (COTS) computer technology by roughly the age of the aircraft plus ten years or more. (Software updates occur approximately once every 18 months on the newer FMCs, and less frequently on the older ones, but they are not major upgrades.) Moreover, airline companies have no financial incentive to upgrade FMCs until the ATC system can realistically provide a return on the investment. It's a classic "chicken and egg" problem: the airlines can't afford to upgrade until the relevant ATC capabilities are available, but the ATC upgrades won't be effective until the airlines upgrade.

The FMC technology lag problem makes any major new ATC concept that requires new FMC capabilities a very long-term prospect. However, that situation could start changing for several reasons. First, the computer hardware that is certified for FMCs will eventually become much more powerful, just as COTS hardware did a decade or two ago. Also, as the hardware becomes more powerful, major software upgrades in the field may become feasible without replacing or even removing the FMC. That hardly seems radical, considering that minor software updates are already done that way (and smart phones are routinely upgraded wirelessly with the push of a button, but wireless upgrades for FMCs would pose an unnecessary security risk). Once that capability is in place, the upgrade cycle should be dramatically reduced. However long it takes to reach that point, planning should begin now to utilize advanced FMC computer technology to maximize airspace capacity, airport throughput, and safety.

III. Application to Runway Arrival Spacing

As mentioned earlier, trajectory specification is applicable to both enroute airspace and terminal areas. The same basic principles apply in both cases, but the requirements differ considerably. The terminal area around a major airport tends to have a higher traffic density, larger turn angles, and more constraints than enroute airspace. For those reasons, the potential benefit of trajectory specification is likely to be larger in the terminal area. This paper therefore focuses on the terminal area and, more specifically, on final spacing for flights arriving at the runway.

The schedule of arriving flights into the terminal area around a major airport is created by an arrival manager. The arrival manager that is currently in use in US is called the Traffic Management Adviser (TMA).[15] Originally developed by NASA, TMA provides an arrival schedule to maximize throughput without overloading the runways. Runway capacity is limited by the wake-vortex spacing requirements and runway occupancy times.[16] The spacing required between intrail arrivals depends on the weight classes (small, large, B757, or heavy) of the leading and trailing aircraft, and it ranges from 2.5 to 6 nmi, but in most cases it is 3 nmi. The minimum separation standard in the terminal area is also 3 nmi horizontally or 1,000 ft vertically. Other more complicated airspace rules and constraints also apply,[16,17] but they are not considered in this preliminary study.

The guidance provided by TMA is based on imperfect trajectory predictions, and the execution by controllers and pilots is also imperfect. Moreover, TMA does not eliminate separation conflicts. Both spacing and separation conflicts occur, therefore, and controllers are needed to resolve them. Currently, that control tends to be tactical, with controllers issuing vectors, speeds reductions, and temporary altitude holds. That also forces pilots to fly tactically, with the FMC disengaged, manually entering speeds, altitudes, and headings. The objective of trajectory specification is to put flights on automatic, efficient, precisely specified trajectories that are guaranteed to be free of conflicts all the way through the terminal area.

In the past, NASA developed an advisory system for terminal area controllers called the Final Approach Spacing Tool (FAST).[18,19] Due to the complex interactions with the controller, however, an advisory system is in some respects more difficult to develop than an automated system. An automated system has more maneuvering flexibility, and interactions with the controller can be ignored, at least in the preliminary design. Ultimately the possibility of controller intervention must be considered, but that should be a relatively rare off-nominal condition rather than the norm. The certification of an automated system will be a major effort, of course, but there may be no way around it if terminal airspace capacity and airport throughput are ever to be truly maximized.

NASA is currently developing a research prototype software system to automate sequencing and conflict resolution in the terminal area.[3] This system has been shown to resolve virtually all spacing and separation conflicts in fast-time simulations with current traffic levels. While that is an important advancement of the state of the art, the system has not yet been tested in a more realistic environment with trajectory prediction error and pilots in the loop. Trajectory prediction error is not easy to simulate realistically. A Gaussian or uniform distribution typically does not model outliers well, but outliers pose the most difficult challenge. If the errors can be bounded, the problem becomes much more manageable. But that is essentially what trajectory specification does by imposing explicit tolerances.

In general, spacing conflicts are easier to resolve than separation conflicts because spacing is usually done in one dimension whereas separation is done in three dimensions. (Spacing is slightly more complicated with dependent runways, but all runways are independent in this study.) Also, spacing involves only the along-track tolerances, whereas general separation involves the tolerances in all three axes. Fortunately, the resolution of spacing conflicts at the runway threshold tends to also resolve most earlier spacing conflicts and separation conflicts for flights in the same arrival stream. The general strategy, therefore, is to first resolve spacing conflicts at the runway threshold, then resolve earlier spacing and separation conflicts, as was done in the earlier study.[3] This paper focuses on arrival spacing at the runway threshold and defers general separation for a future paper.

The arrival trajectories used in this study were generated with the Kinematic Trajectory Generator (KTG),[20] which is part of the Airspace Concept Evaluation System (ACES),[21] a fast-time simulation program developed for NASA. KTG uses aircraft performance data from the Base of Aircraft Data (BADA)[22] developed by Eurocontrol.

The analysis and results below are based on 150 arrival trajectories to the DFW and DAL airports. The trajectories were generated by running ACES/KTG *without conflict resolution* and storing the resulting trajectories. These trajectories are what aircraft would fly (on standard arrival routes and nominal interior routes) with no other traffic around. 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 following sections discuss the methods for modifying the trajectories to resolve spacing conflicts by imposing the required delay.

A. Arrival Delay Methods

Once an ordering of arrivals to each runway is determined, the required spacing between flights is normally realized by applying a delay to the trailing flight of each consecutive pair when necessary. Spacing requirements are given in terms of distance, but delay is specified in terms of time, so a conversion from required spacing distance to delay time is necessary. Trajectory specification allows the exact delay to be determined to realize a given spacing even with arbitrary speed variations.

As in previous work,[3] delay is achieved in this study by first reducing speed, and if the maximum permissible speed reduction is insufficient, various types of "path stretching" are used. The first type of path stretching is an extension of final approach, and the second type is a symmetric path stretch. These delay methods are explained in more detail below. The resulting overall delay algorithm is intended to be an example of how a specified delay can be systematically realized, but it is not the only reasonable way. If more delay is needed, a more general form of (asymmetric) path stretching can be used, followed by holding patterns, but those methods will not be discussed in this paper.

The methods for imposing delay in this study are similar to the methods used in previous work,[3] but the *algorithms* are significantly different because they work directly with the original trajectory data rather than calling a trajectory generator (KTG) for each variation of trajectory parameters to be tried. In other words, the trajectory generator is used to generate the original trajectories but is never used again. That difference is important because in this study the original trajectories are supposed to represent downlinked trajectory requests from an FMC, and the ATC system is not expected to have its own copy of the same trajectory predictor that each FMC uses. That means that each iteration of trajectory parameters required to realize a given delay would require a call from the ATC system to the airborne FMC, which is clearly not feasible. But even if the ATC system had an exact copy of the trajectory predictor in the FMS, the approach used here is much more efficient computationally, which allows for more accurate solutions.

The computation times given below are for an Intel 2.7 GHz processor with no parallel computation. Because the software was written in functional programming style (in Scala), the algorithms should be relatively easy to parallelize on modern multicore processors for even faster computation if necessary.

1. Speed Reduction

As mentioned earlier, the first delay method that is tried for each flight is speed reduction. During descent, speed is normally reduced by steps in Calibrated Airspeed (CAS) (and even at constant CAS, true airspeed (TAS) decreases with altitude during descent). A typical CAS at entry into the Terminal Radar Approach Control (TRACON) area is 250 kn, and that CAS value is normally held constant for several minutes before the step-downs begin to a landing speed typically in the range of 140-150 kn. A delay in the runway arrival time can be realized by reducing the CAS to a lower value during the initial period of constant CAS.

Figure 7 shows an example of delays by speed reduction for a particular flight. In this case, the flight enters the TRACON at a CAS of 250 kn as shown by the top (blue) curve and stays at that speed for approximately 8 minutes. The lower dashed (red) curves show CAS reductions in steps of 10 kn to a maximum reduction of 40 kn. The speed reduction starts 1.0 min after the entry into the TRACON with a nominal deceleration of 0.06 g. As expected, these curves are very similar in shape to the original speed profile except for the initial speed reduction and the extension to the right on the time line, which represents the desired delay in arrival time. The maximum allowed airspeed reduction of 40 kn in CAS is a simplification and would be replaced in a real system with a value based on the actual aircraft flight envelope.

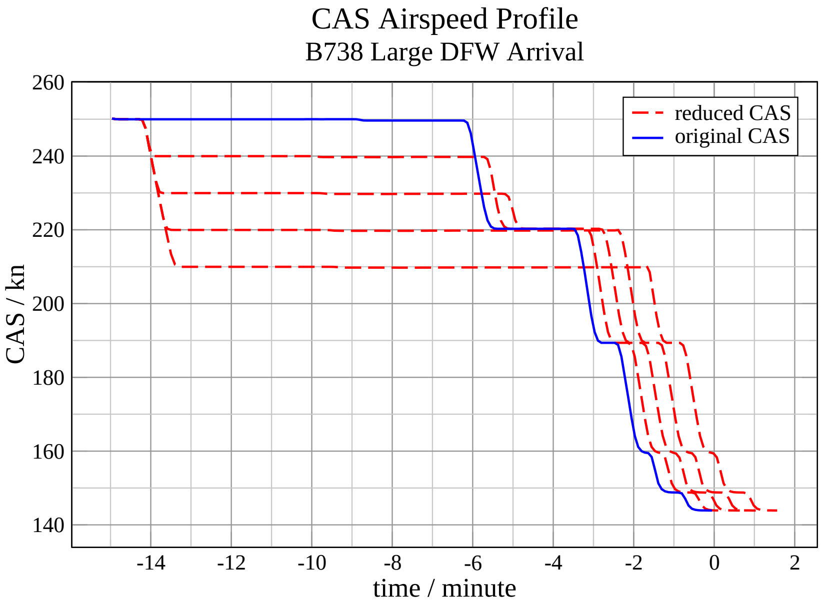

Figure 7. Example of CAS profiles for delay by speed reduction

As mentioned earlier, the tolerances would depend on the aircraft navigational capabilities and the current traffic situation. The determination of appropriate tolerances is a major topic in itself and is beyond the scope of this paper. The bounds shown in Figures 8 and 9 are examples of possible altitude and along-track bounds. The along-track tolerances would depend on the proximity of other nearby traffic in the same arrival stream. The altitude tolerances, on the other hand, would more likely depend on departing cross traffic that will pass over or under the arriving flight. Note that the cross-track and altitude bounds are superseded by the localizer and glideslope on final approach.

The speed reduction algorithm works as follows. First, the ground speed is calculated by back-differencing the trajectory points and dividing the distance between successive points by the time difference (5 sec in this study). Note that because the trajectory is a predicted trajectory it contains negligible noise (numerical roundoff but no measurement noise), so back-differencing is acceptable. Note also that deriving speed directly from position ensures consistency between position and speed. The resulting ground speed is then converted to true airspeed by adding the wind speed in the direction of flight, then the true airspeed is converted to CAS using standard formulas. The CAS in the initial segment of constant CAS is then reduced to the desired value by increasing the time step by the inverse of the ratio of the change in CAS, and the accumulated change in time is tracked. The time for each trajectory point is thus changed while the position remains unchanged. A deceleration limit of 0.06 g is maintained throughout the procedure by limiting the change in CAS at each time step. Once the point in the trajectory is reached where the original CAS is below the new reduced initial CAS, no additional delay is added.

The resulting trajectory is then composed of points with varying time steps, but it is converted to equal time steps by interpolation as discussed earlier (for fast access to altitude and other variables

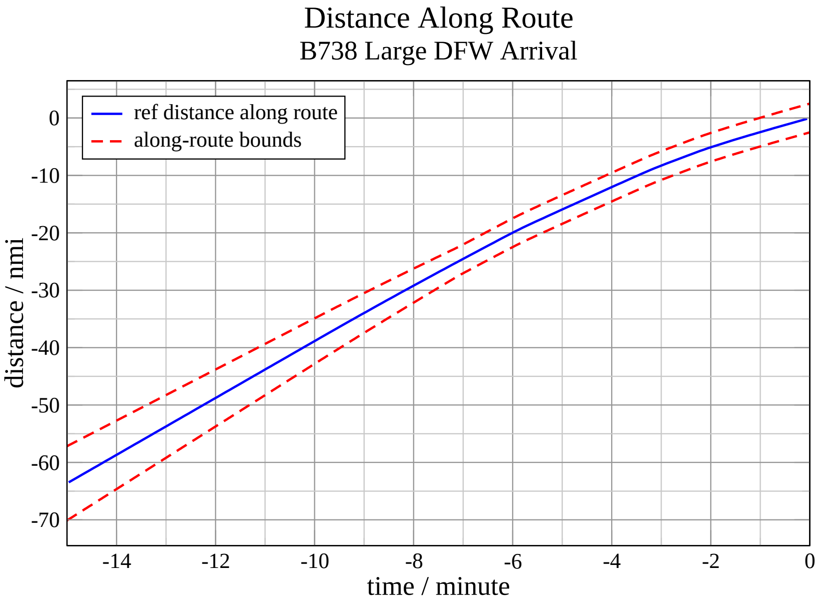

Figure 8. Reference distance along track and bounds as a function of time

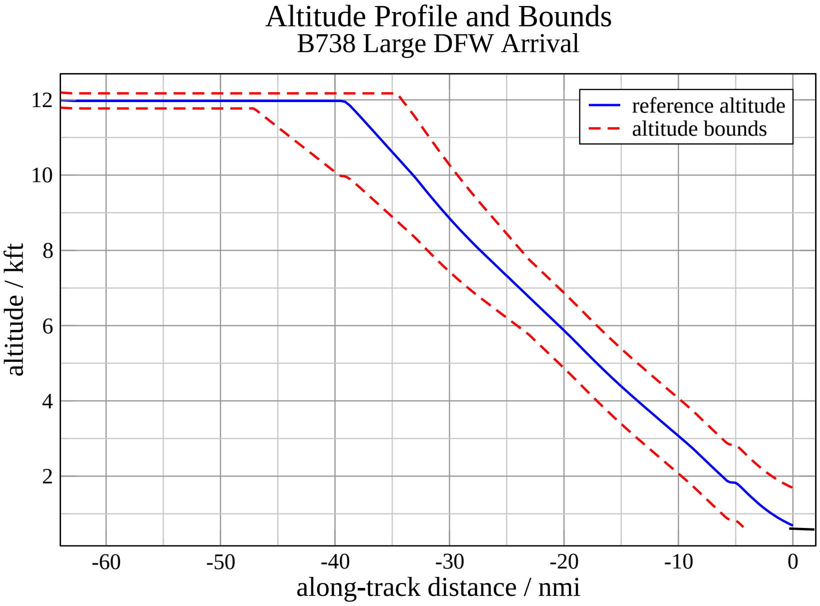

Figure 9. Reference altitude profile and bounds as a function of along-track distance

as a function of time and distance). Because the times of the trajectory points are changed but not the positions, the altitude profile as a function of along-track distance is unchanged, but the altitude profile as a function of time is delayed as necessary. A more refined delay algorithm could take into account the aerodynamic parameters of each aircraft type (and current weight) to determine the maximum acceptable speed reduction and also perhaps to modify the altitude profile together with the speed profile, if necessary, to avoid stall or other undesirable conditions.

The speed reduction algorithm determines the resulting delay. What is needed, however, is the inverse of that function, the speed reduction required to realize a specified delay. Figure 10 shows the time delays for one particular flight as a function of the reduced initial value of CAS, with the time of the start of the speed reduction as a parameter. In this study, the CAS reductions were started 1.0 minute after entry into the TRACON, as represented by the top curve. The largest delay in this case was slightly less than 1.5 min. As the figure shows, the curves are slightly nonlinear but monotonic. The delay accuracy requirement has not yet been rigorously determined, but an accuracy of 1 sec or less is almost certainly more than adequate.

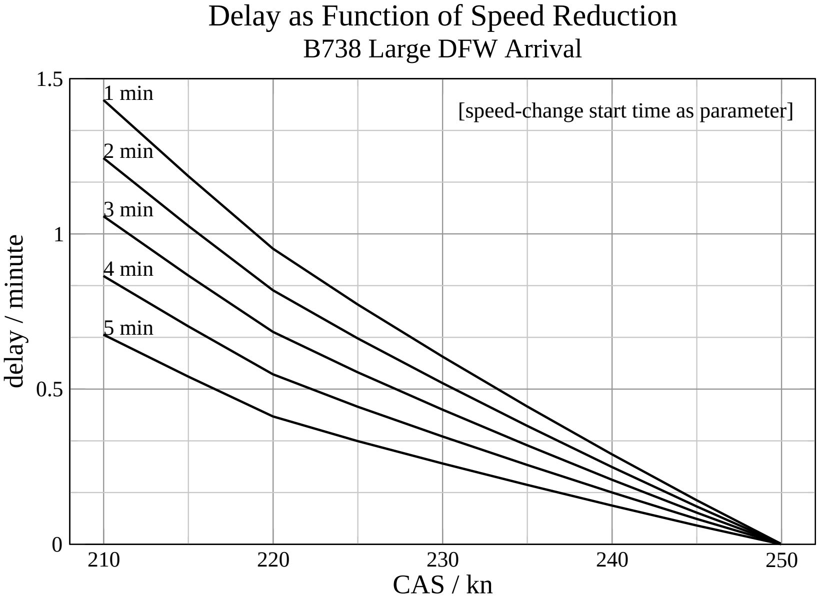

Figure 10. Example of delay as a function of speed reduction

To solve for the CAS required to realize a specified delay, an algorithm was implemented based on the bisection method followed by a final interpolation. This algorithm can achieve arbitrary accuracy at the cost of computation time by simply increasing the number bisection steps. Five bisection steps were used (followed by an interpolation), and the specified delay was incremented from zero to the maximum achievable delay in steps of 30 sec for each of the 150 arrivals. The largest resulting delay error magnitude was 0.34 sec, and the average time per test was 11 ms (where a test is the computation of one CAS for one delay of one flight). Those levels of accuracy and computation time are more than adequate.

Figure 11 shows a snapshot in the horizontal plane of an example of an arrival encounter with spacing by speed reduction. The red rectangles represent the horizontal bounds at a point in time, which are decreasing linearly to a tolerance of ±2 0.2 nmi at the runway. The green ovals (rectangles with rounded corners) must remain separated to maintain the required spacing and separation between the bounding rectangles, which are both 3 nmi in this example. The red line connecting the bounding rectangles is the line of minimum separation, which is 4.55 nmi at this point in time, which is 5.6 minutes before the leader is scheduled to arrive at the runway threshold.

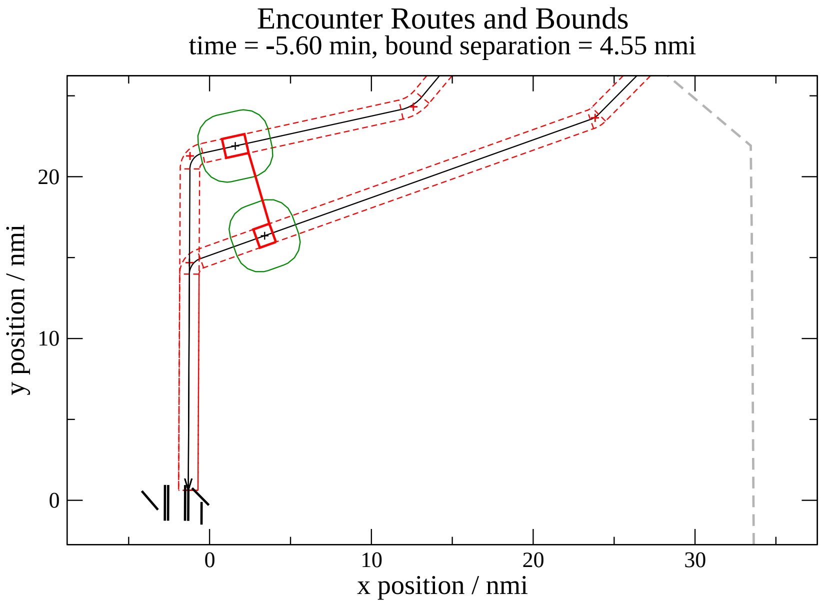

Figure 11. Snapshot in horizontal plane of an arrival encounter with spacing by speed reduction

Figure 12 shows the separation and spacing profile as a function of time for the arrival pair shown in Fig. 11. The red line at 3 nmi represents the minimum required separation and spacing for this pair of "large" aircraft. The upper (gray) curve represents the "point" separation and spacing of the reference trajectories, and the lower (blue) curve represents the "bound" separation and spacing of the horizontal bounding spaces. For each curve, the dashed segment on the left represents the horizontal separation, and the solid segment on the right represents the spacing while one flight is following directly behind the other on the same path (during which the spacing is equivalent to the separation). The along-track tolerances for each flight in this example are ±0.2 nmi (approximately ±5 sec) for each flight on final approach, hence the bound spacing is 0.4 nmi less than the point spacing. An additional buffer of 2 sec is also added between bounding spaces. The large variations in separation are a result of the turns to final. The spacing decreases as usual to a minimum at the runway threshold (at time zero at the right edge of the plot) as the speeds decrease to landing speed.

2. Extension of Final Approach

When speed reduction is insufficient to realize the required spacing, the next method used is extension of final approach. A waypoint is added to extend the final-approach leg back linearly by a specified distance within the TRACON boundary. The control variable is the distance by which the waypoint at turn to final is moved back. A base leg is then added if necessary to limit the turn to final approach to a maximum of 90 deg. Figure 13 shows an example for an arrival route with a downwind leg. The solid line represents the original route, and the dashed lines represent final extensions in steps of 4 nmi. The pattern here is the classic "trombone" maneuver. Figure 14

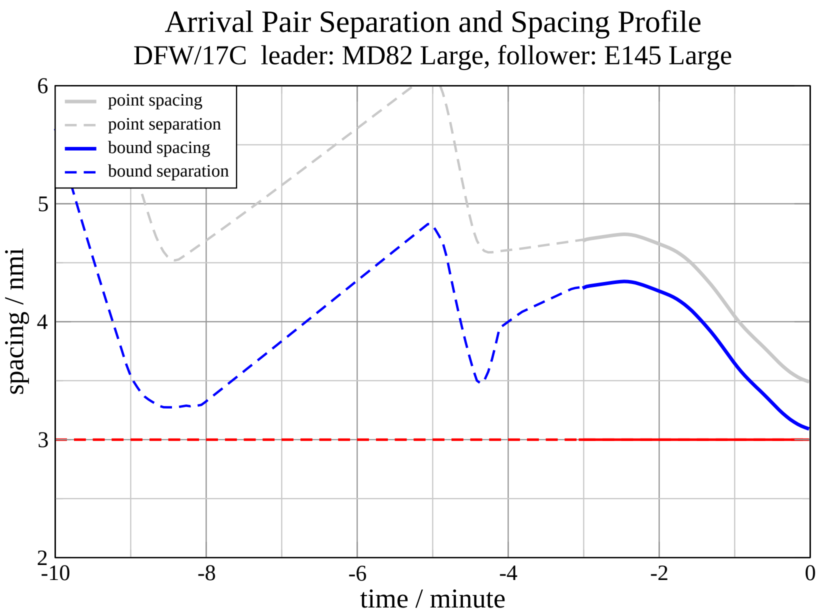

Figure 12. Arrival pair separation and spacing profile

shows another example with a base leg getting added to prevent the final turn angle from exceeding 90 deg.

In addition to modifying the route geometry as shown in the figures above, the algorithm also has to provide a longitudinal profile. The approach taken in this study is based on the simplifying assumption that the lateral and longitudinal dynamics are decoupled, which is normally a reasonable assumption for commercial passenger airplanes. The original longitudinal profile is therefore simply superimposed onto the new route starting from the runway and going back, and a section is added at the back of the route to fill the gap that is left due to the 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. This approach could also possibly result in a violation of airspace restrictions, but that was not considered in this preliminary study.

Plots of delay as a function of the final extension distance are similar to Figure 10 and will not be shown. To solve for the final extension distance required to realize a specified delay, an algorithm similar to the one discussed for speed reduction was implemented based on the bisection method followed by a final interpolation. Again, this algorithm can achieve arbitrary accuracy at the cost of computation time by simply increasing the number bisection steps. Each of the 150 arrivals was first reduced in speed by 40 kn, the maximum speed reduction allowed in this study, then an additional delay was specified in increments of 30 sec. The largest delay error magnitude was 0.06 sec at an average time per test of 16 ms. Again, those levels of accuracy and computation time are more than adequate.

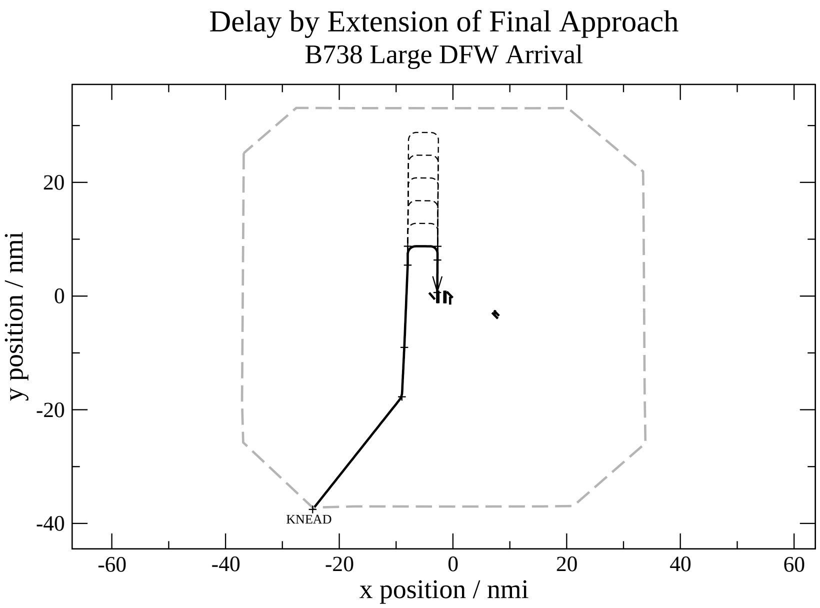

Figure 13. Examples of delay by extension of final approach: the “trombone” pattern

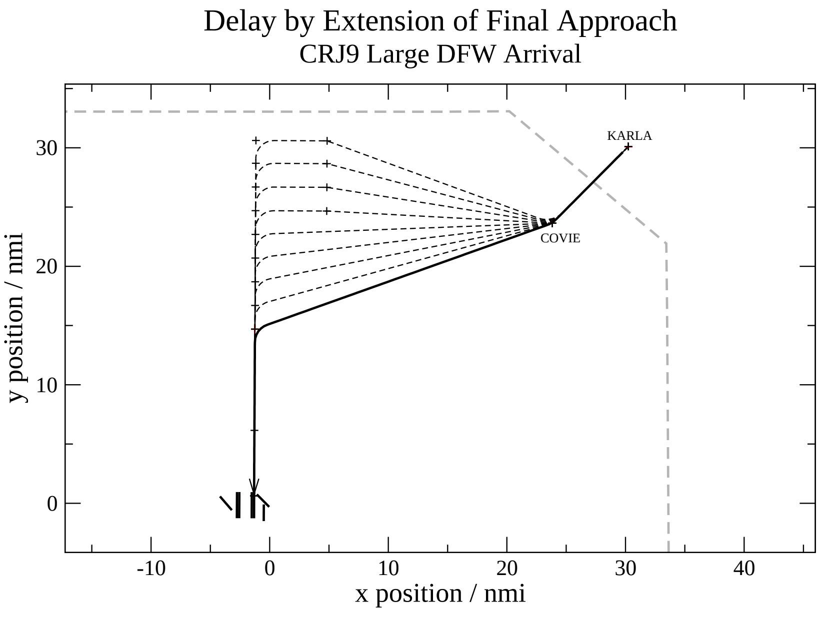

Figure 14. Examples of delay by extension of final approach

3. Direct Routes and Symmetric Path Stretching

When speed reduction and final-leg extension are insufficient to realize the required spacing, the next method used is symmetric path stretching. However, symmetric path stretching is best applied to direct routes. Direct routes are routes in which extraneous waypoints inside the TRACON have been removed. For a direct route, a waypoint is added just inside the TRACON boundary to avoid a discontinuous trajectory change at the start of the trajectory, and all subsequent waypoints are removed up to the turn to base or final, and the length of the final leg is a minimal 8 nmi. Ideally, all routes should start as direct routes, and the delay methods discussed above can be applied in sequence as explained above, starting with speed reduction. For various reasons including noise considerations, however, the acceptability of direct routes is not certain.

Figures 15 and 16 show examples of direct routes with maximum extension of final approach followed by symmetric path stretches. The control variable is the perpendicular offset distance from the line segment between the two endpoints, which can go in either direction. The new waypoint must be inside the TRACON boundary and is also constrained in two other ways. Firstly, the offset distance was not allowed to exceed 20 nmi or half the distance between the two original waypoints, whichever is smaller. Secondly, the new waypoint is not allowed to cross over the downwind leg, as shown in Figure 16, because that could interfere with other arrivals to, or departures from, the runway.

The original longitudinal profile is then superimposed onto the new route using the same algorithm that was discussed in the previous subsection. Also as before, plots of delay as a function of the symmetric path offset distance are similar to Figure 10 and will not be shown. To solve for the final extension distance required to realize a specified delay, an algorithm similar to the one discussed for speed reduction was implemented based on the bisection method followed by a final interpolation. Each of the 150 arrivals was reduced in speed by 40 kn as before, and also had its final leg extended to the limit, and then an additional delay was specified in increments of 30 sec. The largest delay error magnitude for the 150 arrivals was 0.04 sec at an average time per test of 6.7 ms. Again, those levels of accuracy and computation time are more than adequate.

B. Delay Limits

Figure 17 shows the cumulative distribution of achievable delay at the runway threshold using the methods discussed above for the 150 arrivals used in this study starting with direct routes to final approach. As can be seen, a maximum speed reduction of 40 kn can only guarantee approximately 40 sec of delay, and it can achieve 1.5 minutes of delay for only approximately 40% of arrivals. Extension of final approach is more effective and can achieve approximately 3 min of delay in all cases when added to speed reduction, or 5 min for 60% of cases. Symmetric path stretching adds approximately 1 to 2 min to the achievable delay.

As a final test, the combined delay algorithm (speed reduction followed by extension of final approach, then symmetric path stretching) was tested in delay steps from zero to the delay limits shown in Figure 17 in steps of 1 minute for each of the 150 arrival trajectories converted to a direct route. The largest delay error magnitude was 0.63 sec at an average time per test of 22 ms. As before, these results are more than adequate for practical use.

IV. Conclusions

This paper updates the trajectory specification concept that was first proposed nearly 10 years ago and starts the process of applying it to the terminal area around major airports. The main idea is that aircraft trajectories be explicitly bounded to a precisely defined volume of space at each point in time. It is a generalization of Required Navigation Performance (RNP) to the longitudinal plane,

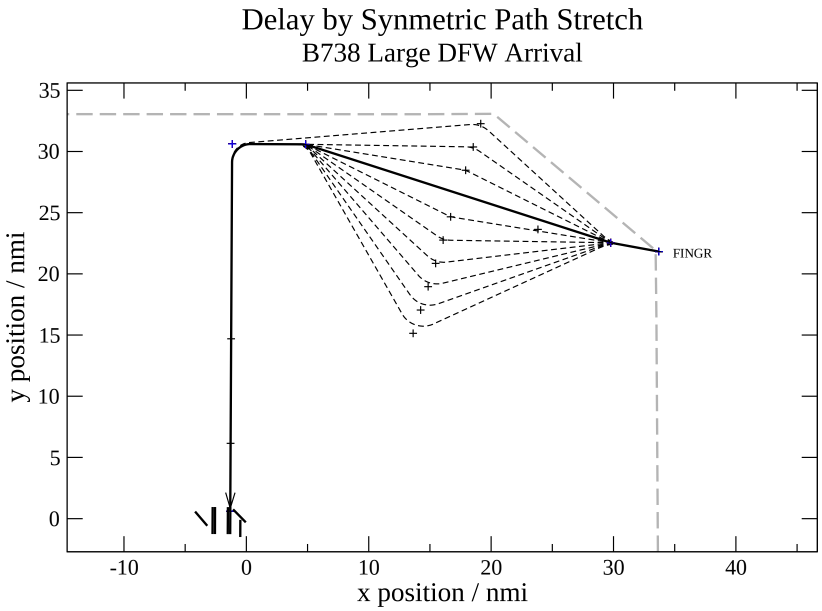

Figure 15. Examples of delay by symmetric path stretching

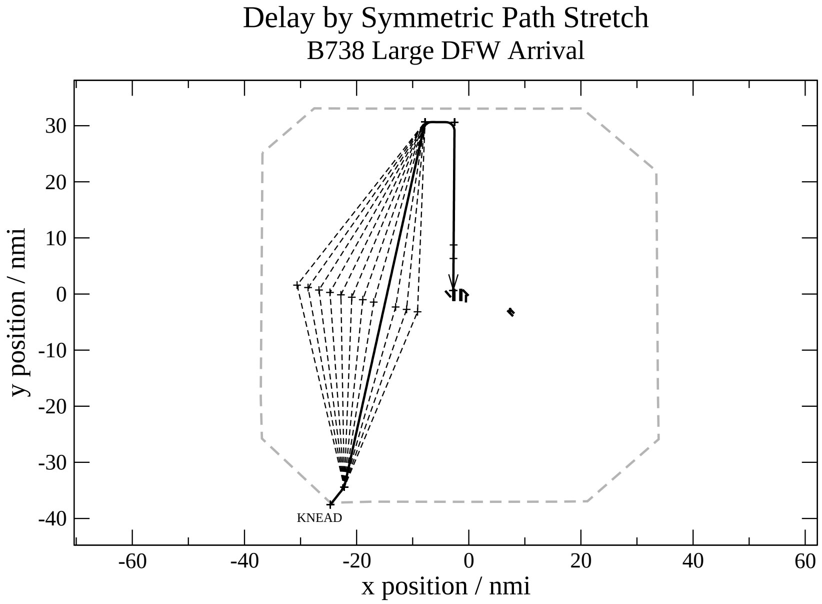

Figure 16. Examples of delay by symmetric path stretching

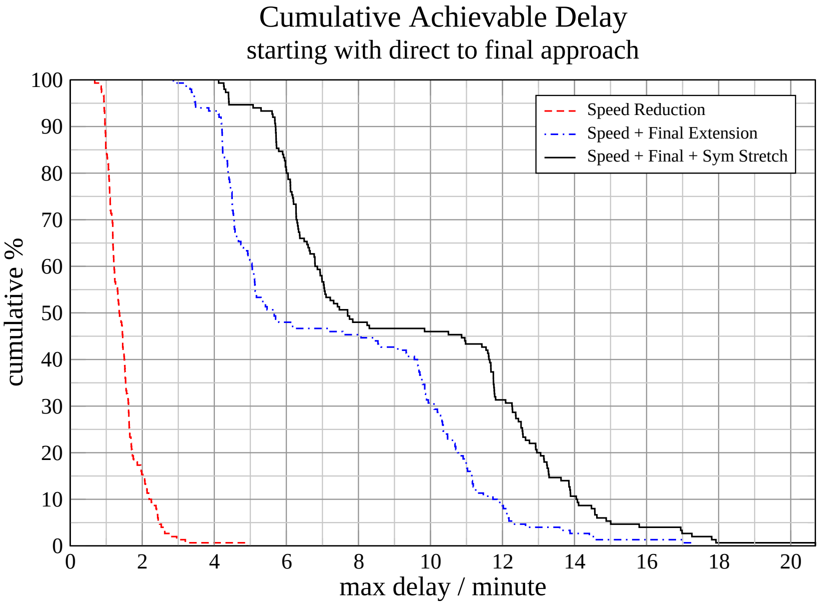

Figure 17. Cumulative distribution of achievable delay for arrivals

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 should be a key to achieving the high level of safety and reliability needed for ATC automation.

A standard language will be developed to communicate trajectory specifications (including the reference trajectory and the tolerance parameters) from ATC to airborne FMCs and vice versa. Taking all relevant airspace restrictions into account, FMCs will convert pilot intent (e.g., route) into a trajectory prediction and downlink it to ATC as a request. ATC will then assign tolerances and check for spacing and separation conflicts. If no conflicts are found, ATC will uplink the trajectory specification as an assignment; otherwise ATC will modify the trajectory (or possibly modify the assigned trajectory of another flight) to resolve conflicts before uplinking it. In a more advanced version, the FMC could also receive the trajectory assignments for other flights, and avoid conflicts with them when generating its own trajectory requests, but those trajectories would still be double-checked for conflicts by ATC.

Trajectory specification is applicable to both enroute airspace and the terminal area around airports, but this paper focused on arrival spacing in the terminal area. Realistic trajectories from a fast-time simulation were used to represent downlinked arrival trajectory requests, and an algorithm was developed to modify each trajectory to realize a specified delay. The delay algorithm first uses speed reduction in terms of calibrated airspeed, then it uses extension of final approach if necessary, and finally it uses symmetric path stretching. A simple algorithm based on the bisection method and interpolation was developed to determine the required delay parameters to realize a specified delay. Numerical testing showed that more than sufficient accuracy can be realized in a sufficiently short computation time.

The solution presented in this paper for the final spacing problem is an important step, but it is only one of several problems that must be solved before the trajectory specification concept is ready to be tested in simulation. The general separation problem and the selection of appropriate trajectory tolerances will be addressed in future work.

Acknowledgments

Huu Huynh provided technical assistance in obtaining the trajectory data that was used in this study. The author would also like to acknowledge Heinz Erzberger for many enlightening discussions and ideas on terminal area air traffic control.

References

[1] Erzberger, H.; Paielli, R.A.: "Concept for Next Generation Air Traffic Control System," *Air Traffic Control Quarterly*, Vol. 10(4)(2002), pp 355-378.

[2] Erzberger, 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)?

[3] Nikoleris, 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.

[4] Erzberger, H.; Nikoleris, T.; Paielli, R.A.; Chu, Y.C.: "Algorithms for Control of Arrival and Departure Traffic in Terminal Airspace," submitted to *Journal of Aerospace Engineering*

[5] Paielli, R.A.: "Trajectory Specification for High-Capacity Air Traffic Control," *AIAA Journal of Aerospace Computation, Information, and Communication*, vol. 2, no. 9, Sept 2005.

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

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

[8] Andrews, J.W.; Erzberger, H.; Welch, J.D.: "Safety Analysis for Advanced Separation Concepts," *Air Traffic Control Quarterly*, vol. 14, no. 1, 2006.

[9] Joulia, 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.

[10] Paielli, R.A.: "Evaluation of Tactical Conflict Resolution Algorithms for Enroute Airspace," AIAA Journal of Aircraft, vol. 48, no. 1, Jan-Feb 2011.

[11] Tang, 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.

[12] RTCA DO-258A: "Interoperability Requirements for ATS Applications Using ARINC 622 Data Communications," April 2005.

[13] Aeronautical Radio, Inc.: "Navigation System Database," ARINC Specification 424-20, Dec 2011.

[14] Cone, 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.

[15] Erzberger, H.; Ito, E.: "Design Principles and Algorithms for Air Traffic Arrival Scheduling," NASA/TP-2014-218302, May 2014.

[16] Isaacson, D.R.; Sadovsky, A.V.; Davis, D.: "Tactical Scheduling for Precision Air Traffic Operations: Past Research and Current Problems," *Journal of Aerospace Information Systems*, Vol. 11, No. 4, April 2014.

[17] "Air Traffic Control," FAA Order JO 7110.65V, Federal Aviation Administration, Dept. of Transportation, Washington, DC., 2014.

[18] Davis, T.J.; Krzeczowski, K.J.; Bergh, C.: "The Final Approach Spacing Tool," *Thirteenth IFAC Symposium on Automatic Control in Aerospace*, Palo Alto, CA., Sept 1994.

[19] Isaacson, D.R.; Robinson III, J.E.; "A Knowledge-based Conflict Resolution Algorithm for Terminal Area Air Traffic Control Advisory Generation," *AIAA Guidance, Navigation, and Control Conference*, Montreal, Canada, Aug. 2001.

[20] Zhang, 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.

[21] 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 Conf., AIAA-2011-6373, Portland, OR, 2011.

[22] Eurocontrol Experimental Centre, "Base of Aircraft Data (BADA) Aircraft Performance Modelling Report," EEC Technical/Scientific Report No. 2009-009, 2009.