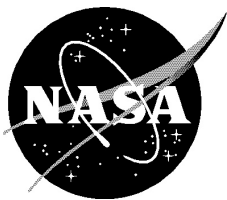


NASA/TM-2004-212825



Trajectory Specification for High-Capacity Air Traffic Control

Russell A. Paielli

September 2004

The NASA STI Program Office . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) Program Office plays a key part in helping NASA maintain this important role.

The NASA STI Program Office is operated by Langley Research Center, the Lead Center for NASA's scientific and technical information. The NASA STI Program Office provides access to the NASA STI Database, the largest collection of aeronautical and space science STI in the world. The Program Office is also NASA's institutional mechanism for disseminating the results of its research and development activities. These results are published by NASA in the NASA STI Report Series, which includes the following report types:

- **TECHNICAL PUBLICATION.** Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA's counterpart of peer-reviewed formal professional papers but has less stringent limitations on manuscript length and extent of graphic presentations.
- **TECHNICAL MEMORANDUM.** Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.
- **CONTRACTOR REPORT.** Scientific and technical findings by NASA-sponsored contractors and grantees.

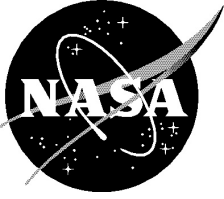
- **CONFERENCE PUBLICATION.** Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or cosponsored by NASA.
- **SPECIAL PUBLICATION.** Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.
- **TECHNICAL TRANSLATION.** English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services that complement the STI Program Office's diverse offerings include creating custom thesauri, building customized databases, organizing and publishing research results . . . even providing videos.

For more information about the NASA STI Program Office, see the following:

- Access the NASA STI Program Home Page at <http://www.sti.nasa.gov>
- E-mail your question via the Internet to help@sti.nasa.gov
- Fax your question to the NASA Access Help Desk at (301) 621-0134
- Telephone the NASA Access Help Desk at (301) 621-0390
- Write to:
NASA Access Help Desk
NASA Center for AeroSpace Information
7121 Standard Drive
Hanover, MD 21076-1320

NASA/TM-2004-212825



Trajectory Specification for High-Capacity Air Traffic Control

*Russell A. Paielli
Ames Research Center
Moffett Field, California*

National Aeronautics and
Space Administration

Ames Research Center
Moffett Field, California 94035-1000

September 2004

Available from:

NASA Center for AeroSpace Information
7121 Standard Drive
Hanover, MD 21076-1320
(301) 621-0390

National Technical Information Service
5285 Port Royal Road
Springfield, VA 22161
(703) 487-4650

TRAJECTORY SPECIFICATION FOR HIGH-CAPACITY AIR TRAFFIC CONTROL

Russell A. Paielli
Ames Research Center

Abstract— The doubling or tripling of airspace capacity that will be needed over the next couple of decades will require that tactical separation guidance be automated for appropriately equipped aircraft in high-density airspace. Four-dimensional trajectory assignment (three-dimensional position as a function of time) will facilitate such automation. A standard trajectory specification format based on XML, the Extensible Markup Language, is proposed for that purpose. Trajectories are composed of a series of trajectory segments. The horizontal path consists of a series of straight (great-circle) segments connected by turns of specified radius. Along-track position is specified as a low-order polynomial function of time, and vertical profiles for climb and descent are specified as low-order polynomial functions of along-track position. Flight technical error tolerances in the along-track, cross-track, and vertical axes determine a bounding space, at each point in time, in which the aircraft is required to be contained. Periodic updates in the along-track axis will adjust for errors in the predicted along-track winds. An important safety benefit of this regimen is that the traffic will be able to fly free of conflicts for at least several minutes even if all ground systems and the entire communication infrastructure fail.

INTRODUCTION

As the demand for air transportation increases, the capacity of the current U.S. air traffic management (ATM) system will eventually be stressed to its limits. New technologies in communication, navigation, and surveillance (CNS), along with new decision support systems and an evolutionary development of the ATM system architecture [1], can extend the capacity of the current system for several years, but a revolutionary new approach will be needed within perhaps twenty years to meet the growing demand.

An often misunderstood or overlooked fact about the current ATM system is that sector capacities are a function of controller workload rather than the airspace itself. In other words, current airspace capacity (as distinguished from *airport* capacity) is limited by the *cognitive capacity of human controllers* to maintain safe separation with high reliability. A controller can handle only approximately fifteen aircraft with the ultra-high reli-

bility that is required. However, earlier studies [2, 3] found that traffic in high-density sectors could be at least doubled or tripled over current limits without saturating the actual capacity of the airspace itself. Airspace capacity is difficult to define precisely, but it involves the rate at which conflicts can be reliably resolved without causing more conflicts.

Airspace capacity could conceivably be increased by reducing sector sizes, but that causes other problems. First, it increases the handoff workload because traffic will cross sector boundaries more often. Second, it reduces the amount of space that controllers have available to resolve conflicts within their own sector, hence more coordination is required as aircraft are diverted through adjacent sectors to resolve conflicts. The current sectorization of the airspace has already reached the point of diminishing or negative return on reduction of sector sizes, so that option cannot yield the needed increases in capacity.

One way to substantially increase airspace capacity is to automate separation monitoring and guidance, thereby relieving human controllers of the primary responsibility for safe separation. Four-dimensional (4D) trajectories (three-dimensional position as a function of time) will greatly facilitate such automation. The concept of 4D trajectories was proposed at least as far back as 1972 [8], and it was a key idea in the Eurocontrol PHARE program [9], for example. Several advanced ATM concepts intended for the 2025 time frame [4, 5, 6, 7] are based on 4D trajectories. However, a standard format for specifying continuous 4D trajectories, with error tolerances in all three axes, does not exist, nor is one currently being developed by any major standards organization.

The stringent regimen implied by assigned 4D trajectories may seem to contradict the notion of “free flight,” but it actually does not. The objective is not to restrict routing options any more than necessary, but rather to keep precise and reliable track of intent. Rather than trying to “predict” the trajectory of each aircraft, with no guarantee of correctness or even attempted conformance, trajectories will be assigned, and conformance will be mandated. Without such a regimen, airspace capacity can never be safely maximized. Note, however, that pilots and airlines will be allowed to request trajectory revisions at any time (within reason), and if the requested trajectory is free of conflicts and consistent

with the traffic flow requirements, it will be approved. Flight will therefore be as “free” as it can be without jeopardizing safety.

A major safety benefit of the proposed regimen is that each equipped aircraft will always have a conflict-free trajectory to fly for several minutes or more, even during a complete failure of all ground systems and the entire communication infrastructure. The duration of this conflict-free period will depend on how long aircraft can stay in conformance without updates, which in turn depends mainly on wind-modeling accuracy. During periods of high accuracy, the conflict-free period could be indefinitely long. This benefit could ultimately prove to be critical for the acceptance of automated tactical separation in high-density airspace. Without 4D trajectory assignments, a ground computer failure could dump the responsibility for safe separation onto human controllers, which would be unacceptable because some sectors will contain more traffic than a controller can safely handle. Recall that increasing the traffic density beyond what a human controller can reliably handle is precisely the objective of automated separation.

A standard called Controller/Pilot Datalink Communication (CPDLC) [10] is currently being developed for communicating specific maneuvers using standard message types, but it is not designed to specify 4D trajectories. Barrer proposed the concept of path objects [11], which constitute a simple “path language” for expressing 3D route patterns such as straight segments, turns, S-turn delays, holding patterns, etc. CPDLC and path objects are potentially useful, particularly in the period of time before 4D trajectory assignment can be implemented, and for aircraft that are not equipped for 4D guidance, but they do not actually specify continuous 4D trajectories.

Barhydt and Warren [13] have recently proposed “The Development of an Information Structure for Reliable Communication of Airborne Intent and Aircraft Trajectory Prediction.” Their proposal is associated with Automatic Dependent Surveillance-Broadcast (ADS-B) [14, 15]. Because it is intended for implementation within the next few years, it is constrained by current and next-generation FMS (Flight Management System) capabilities. The ADS-B Trajectory Intent Bus gives discrete 4D waypoints but does not precisely specify a 3D reference position as a continuous function of time, nor does it precisely specify a 3D bounding space at each point in time. ADS-B was designed for state dissemination rather than detailed trajectory specification, and attempting to use it for the latter will be awkward at best.

One problem with 4D waypoints is that they do not specify a continuous trajectory. The points can be generated at a sufficiently high density to approximate a continuous trajectory, of course, but that might require an inefficient use of bandwidth. Another problem is that they fail to capture the structure of the trajectory. Real

trajectories consist of a series of segment types, such as climb at constant CAS (Calibrated Airspeed), cruise at constant Mach, etc., but discrete 4D points do not convey that structure. This paper will propose a structured, parametric approach based on straight (great-circle) segments, constant-radius turn segments, and low-order polynomial approximation. In addition to data compression, polynomials also provide other desirable features such as analytic differentiation and simplified interpolation.

A more fundamental problem with simply using a sequence of discrete 4D points is that along-track position error couples into altitude. A trajectory can be visualized as a 3D tube through which the aircraft flies, where the scalar along-track position in the tube constitutes the fourth dimension. The tube itself should be fixed relative to the earth. Making altitude a function of time causes the tube to shift, relative to the earth, as a function of the along-track error. In other words, if the aircraft is ahead of or behind schedule, its target altitude should be a function of along-track position rather than time. Altitude should therefore be specified as a function of along-track position rather than time, which cannot be done with 4D waypoints.

The objective of this paper is to propose a standard and a parametric format for specifying 4D aircraft trajectories, a standard 4D “path language.” The specified trajectories could be entire flights from takeoff to landing, or any portion thereof. This standard will be used to precisely specify the assigned 3D reference position and flight technical error tolerances as a continuous function of time. At each point in time a 3D bounding space will be determined in which the aircraft is required to be contained. This bounding space is similar in principle to the PHARE “contract tube” [12], but it will typically be larger and more flexible, particularly in the along-track direction. Trajectories will be synthesized to guarantee the minimum required separation for any pair of aircraft as long as both conform to their assigned trajectories within the specified tolerances.

A key aspect of the format proposed in this paper is that it is based on XML, the Extensible Markup Language. XML is rapidly replacing binary formats for automated business-to-business transactions and is being widely used for computing standards such as Scalable Vector Graphics (SVG). Whereas binary formats typically require the same data to be transferred in the same precise order every time, XML provides more flexibility in the selection and ordering of the data fields. The flexibility of XML will be indispensable for trajectory specification because each trajectory can have a variable number of segments of various types. XML also allows aircraft characteristics and flight preferences to be easily specified.

The remainder of the paper is organized as follows. First, the basic requirements of the proposed trajectory specification standard are discussed. Then the neces-

sary coordinate systems and transformations are outlined. Next, polynomial approximation of vertical profiles and along-track position is discussed. The proposed XML format itself is then presented, after which routine along-track trajectory updates are discussed. Finally, the requirements for trajectory software objects are outlined, followed by some concluding remarks.

REQUIREMENTS

The trajectory specification standard to be proposed in this paper is intended to be used for communicating trajectories between air and ground. Pilots or Airline Operation Centers (AOC) should be able to use it to downlink requested trajectories, and ground systems should be able to use it to uplink assigned trajectories. The basic requirements are that it be:

- able to precisely specify any “reasonable” 4D reference trajectory.
- able to precisely specify error tolerances relative to the reference trajectory.
- based on a global earth-fixed coordinate system.
- parametric and reasonably compact.
- based on a text format readable by humans.
- suitable for an international standard.

The first requirement is that the format be able to precisely specify any “reasonable” 4D trajectory (3D position as a function of time). A unique 3D position must be precisely determined at each point in time, and the set of specifiable trajectories must not be unreasonably restrictive. Efficient climbs and descents must be allowed, for example, and turns must be allowed during climb and descent. The horizontal path will be restricted to straight (greatcircle) segments connected by turns of constant radius to simplify computations and conformance monitoring. These restrictions should not significantly limit practical routing flexibility. Note that wind-optimal routes can be approximated with sufficient accuracy for practical purposes using greatcircle segments of, say, 100 to 200 nmi in length (depending on the length of the flight). More general horizontal path segment types can be added later if desired.

The second requirement is the ability to specify error tolerances for the flight technical error in each of the three axes: along-track, cross-track, and vertical. The error tolerances will precisely determine a 3D bounding space in which the aircraft is required to be contained at any point in time. Those bounds will be the key to assuring that the minimum required separation is maintained at all times without the attention of a human controller. If an aircraft fails to conform, or is expected to fail shortly, its status will be temporarily downgraded

to unequipped, and if necessary it will be automatically issued a basic heading or altitude resolution advisory (by a system outside the scope of this paper). After the situation is under control, the aircraft will either reacquire a new assigned trajectory or be handed off to a human controller for conventional separation monitoring until it is able to reacquire a new assigned trajectory.

Trajectories will be synthesized to guarantee the minimum required separation for a specified period of time called the conflict time horizon, which could be perhaps fifteen minutes. The key point is that, if the trajectories are correctly synthesized, conformance by any two aircraft will guarantee the minimum required separation between them for a specified period of time, regardless of where each aircraft is within its bounding space. In other words, the bounding spaces themselves will always maintain the minimum required separation. Note that minimum separation standards are specified in terms of the separation *distance* between aircraft, regardless of velocities or higher-order dynamics. Hence, the trajectory error tolerances will also be specified in terms of distance or length. Velocity and acceleration can obviously affect future conformance, but actual current conformance will not depend on them. Nevertheless, a conformance monitoring system is free to use velocity and acceleration to try to predict impending nonconformance.

In the current air traffic system, standard navigational conformance bounds of ± 4 nmi in cross-track define a lane width of 8 nmi. However, those bounds are routinely violated for various reasons, such as loose piloting or controllers issuing heading “vectors” but not entering them into the system. In the vertical axis, conformance bounds apply only in level flight, and no bounds apply in the along-track axis (except arrival time constraints). The lack of rigorous conformance bounds in the current system makes conformance monitoring a “fuzzy” problem, which Reynolds and Hansman [16] have attempted to solve using fault detection methods. But conformance monitoring itself is precisely defined if conformance bounds are based on position only and specified precisely, as proposed in this paper. The more difficult and “fuzzy” problem is the detection of faults that could lead to imminent non-conformance, and that is where Reynolds’ approach could still apply.

The error tolerances will be based on Required Navigation Performance (RNP) specifications [17], but they could be relaxed in sparse traffic. Because winds cannot be modeled exactly, the most challenging axis for which to set tolerances is the along-track axis. Tightening the along-track tolerance increases airspace capacity, but it also increases the probability that aircraft will be required to fly at inefficient or even unflyable airspeeds. Along-track position error tolerances must be set as a compromise between those two effects. For more flexibility, it will be allowed to vary linearly with time. Also, the along-track assigned position and velocity will be up-

dated periodically to compensate for errors in modeling and prediction of along-track wind magnitudes, but only when doing so does not result in a conflict. The management of along-track position assignments and tolerances will be discussed in more detail later in this paper.

The next requirement is that the format be based on a global earth-fixed coordinate system, which will provide a common reference. Local coordinate systems, such as the (pseudo-Cartesian) stereographic projection used within each Air Route Traffic Control Center (ARTCC), are inappropriate for enroute airspace because they are each valid only within one Center. The complexity of switching coordinate systems for each Center would be unnecessarily complicated. The standard WGS84 geodetic coordinate system (latitude, longitude, and altitude) will be used as the reference coordinate system for enroute airspace. Local coordinate systems might be convenient in terminal areas however, so that option will be available too. Also, a curvilinear flight-path coordinate system will be introduced in the next section for specifying and monitoring the flight technical error tolerances.

The fourth item in the requirements list is that the format be parametric and reasonably compact. A continuous 4D trajectory can be approximated by a simple sequence of discrete 4D points (t,x,y,z) , but such an approach tends to be inefficient in terms of storage and bandwidth usage. It also fails to capture the structure of the trajectory. Real trajectories consist of discrete segment types, such as climb at constant CAS (Calibrated Airspeed), cruise at constant Mach, etc., but discrete 4D points do not convey that structure. This paper will propose a structured, parametric approach based on straight (greatcircle) segments, constant-radius turn segments, and low-order polynomial approximation. In addition to data compression, polynomials also provide other desirable features such as analytic differentiation and simplified interpolation.

A more fundamental problem with using a sequence of discrete 4D points is that along-track position error couples into altitude. Suppose, for example, that an aircraft is on approach for landing and is one minute ahead of schedule (but still within tolerance). If altitude is specified as a function of time, the aircraft will be required to land several miles before it reaches the runway! On the other hand, if altitude is a function of along-track position, the aircraft will be required to land at the runway regardless of its status with respect to its schedule. Clearly the latter is preferable. While discrete 4D points are good for specifying trajectories that have already been flown, they are simply not the best choice for specifying trajectories, with error tolerances, yet to be flown.

The fifth requirement listed above is that the format be in plain text, readable by humans. The traditional standard for text is ASCII (American Standard Code for Information Interchange), but the new more general

standard is Unicode. For the purposes of this paper, the ASCII subset of Unicode is sufficient. Text-based formats typically provide less efficient storage than binary formats, but they also tend to be more flexible and less prone to error. Also, text-based formats are more convenient because they can be read directly by humans. Text can be compressed and encrypted into a binary format for efficient and secure radio transmission, but then it would be decompressed and decrypted at the receiving end to recover the original text. XML, the Extensible Markup Language [18], is the new standard text-based format for specifying structured data and transferring it across platforms, and it will be used for the format proposed in this paper.

XML is designed for creating application-specific or domain-specific standards for data specification and transfer. The resulting text-based standards are intended to be independent of any particular computer platform or language. XML is rapidly replacing binary formats for automated business-to-business transactions and is being widely used for computing standards such as Scalable Vector Graphics (SVG). Whereas binary formats typically require the same data to be transferred in the same precise order every time, XML provides more flexibility in the selection and ordering of the data fields. The flexibility of XML will be indispensable for trajectory specification because each trajectory can have a variable number of segments of various types. The flexibility will also allow trajectories to be updated without repeating all the data that remains unchanged from the previous update, which could more than compensate for the inherent inefficiency of text-based data. Note also that XML text compresses well for efficient use of bandwidth.

The final requirement listed above for the proposed trajectory specification standard is that it be suitable for an international standard that is recognized by, and can be automatically flown by, any standard FMS. The standard will be used onboard aircraft to downlink requested trajectories constructed by the FMS or constructed by the pilot using a graphical user interface. The standard will also be used on the ground to check for conflicts and to uplink assigned trajectories. Developing a consensus for an international standard is obviously a major challenge, but such a common language can greatly simplify the logistics of high-capacity ATM. With a common trajectory language, the chances of miscommunication will be much less than they would be without one. If adopted, the actual communication mechanism would probably be an extension of CPDLC [10] or a new datalink message over the Aeronautical Telecommunication Network (ATN).

COORDINATE SYSTEMS AND TRANSFORMATIONS

An assigned trajectory consists of a 4D reference trajectory and flight technical error tolerances. The reference trajectory is the precise 4D trajectory the aircraft would fly in the ideal case of zero flight technical error. It is a precise 3D position that varies as a function of time, and the position at any point in time will be referred to as the reference position. The error tolerances, on the other hand, are the maximum allowed error in each of the three axes: along-track, cross-track, and vertical. These tolerances define a 3D bounding space around the reference position, at each point in time, that the aircraft must stay within to be in conformance.

As explained in the previous section, the WGS84 geodetic coordinate system will be used as a global standard for specifying reference trajectories. Straight (i.e., minimum distance) segments between geodetic points are great circles in general, but for short segments (away from the earth's poles) a greatcircle is close to linear in latitude and longitude. Geodetic coordinates are inconvenient for specifying and monitoring error tolerances, however. For that purpose, a curvilinear flightpath coordinate system, which follows the assigned trajectory, will be used. An example of a segment of such a curvilinear flightpath coordinate system is illustrated in figure 1, showing the along-track and cross-track grid.

A curvilinear flightpath coordinate system is a combination of Cartesian and polar coordinate systems. The first step in converting from WGS84 coordinates to the curvilinear coordinates is to determine the type of the local coordinate region, which is Cartesian in the (assigned) straight segments and polar (or cylindrical in 3D) in the (assigned) turn segments, as shown in the figure. Actually, these regions are not strictly Cartesian or polar, because they follow the curvature of the earth, but for practical purposes they are Cartesian or polar within the local region of reasonable flight technical errors. The key point is that *each segment defines its own local coordinate system*, which is Cartesian for straight segments and polar for turn segments.

Note that the bounding space is based on the definition of the along-track and cross-track error coordinates. Thus, the bounding space follows the curvature of the flightpath, as shown in figure 1. An alternative way to define the error coordinates would be in terms of a time-varying Cartesian coordinate system that follows the aircraft as it turns, but that would not work very well. With that definition, the bounding space shown in figure 1, for example, would be rectangular and would not conform to the curvature of the flightpath. As the reference position progresses around the turn, the bounding space would be a rectangle that rotates with the reference track angle, which would clearly be inappropriate.

Coordinate transformations will be needed to transform the geodetic coordinates of an aircraft position

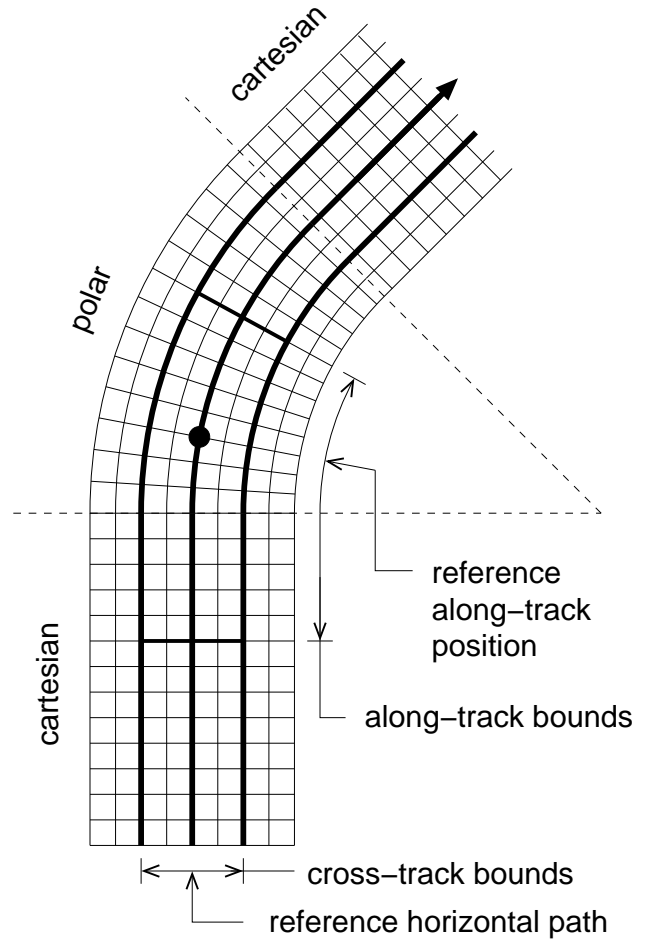


Figure 1: Curvilinear flightpath coordinate system with along-track and cross-track grid.

to the along-track and cross-track coordinates in the curvilinear flightpath coordinate system. In the straight (greatcircle) segments of the assigned trajectory, the local flightpath coordinate system is approximately Cartesian within the range of practical error tolerances, and the along-track and cross-track coordinates of a point can be determined with established greatcircle algorithms.

The earth is nearly but not quite spherical. The equatorial and polar radii differ by approximately 12 nmi, or about 1/300th of the nominal radius. Great-circle algorithms come in two forms: those based on a simplified spherical model of the earth, and those based on a more accurate but more complex ellipsoidal model. The spherical model yields closed-form analytic solutions, whereas the ellipsoidal model yields more accurate but more complicated iterative algorithms. For the purposes of this paper, the cross-track errors will normally be a few nmi at most and are well approximated with the spherical equations. However, the spherical equations for along-track distances can be off by several nmi within the continental U.S., which may be marginally unacceptable

for this application, hence algorithms based on the ellipsoidal model may be required in practice.

The ellipsoidal greatcircle algorithms are too complicated to be presented in this paper, but a few of the key spherical equations are presented in the Appendix. For the purposes of this paper, the important formulas determine the along-track and cross-track coordinates of a given point, relative to a greatcircle from one given point to another. These formulas utilize functions that are also given in the Appendix.

The greatcircle equations apply only in the Cartesian-coordinate regions of the curvilinear flight-path coordinate system, which correspond to the straight (greatcircle) segments of the assigned trajectory. However, they can easily be adapted for use in the polar-coordinate regions too, which correspond to the turning segments of the assigned trajectory. The trick is to start out by computing the along-track and cross-track coordinates as if the point were still in the preceding Cartesian region, then convert to polar coordinates. The origin of the polar coordinate system will be the center of the turn arc, and the reference azimuth angle will be at the start of the turn. The actual cross-track coordinate will be the radial coordinate minus the nominal radius of the turn, so that the reference cross-track coordinate is always zero (consistent with the straight segments). The along-track coordinate will be the angle from the start of the turn, multiplied by the nominal radius of the turn. Note that if the aircraft is flying the turn with a cross-track error, the actual radius of the turn will be different than the nominal radius, hence the actual along-track distance traveled by the aircraft will be different than the along-track coordinate.

A 4D trajectory also includes a vertical profile, which is altitude as a function of time or along-track position. While either time or along-track position could be used as the independent variable, along-track position provides a critical advantage: it fixes the reference trajectory in the earth-fixed coordinate system, which simplifies conflict calculations. Using time as the independent variable, on the other hand, would allow the reference trajectory to drift (relative to the earth-fixed coordinate system) with the along-track position error. As mentioned earlier, a trajectory can be visualized as a 3D tube in space, through which the aircraft flies, with the along-track position in the tube constituting the fourth dimension. The tube itself should be fixed in space. Making altitude a function of time causes the tube to shift in space as a function of the along-track error.

The along-track position will be specified as a low-order polynomial function of time for each segment, as will be discussed in the next section. Also, altitude will be specified as a low-order polynomial function of the *actual* (as opposed to *reference*) along-track position, as illustrated in figure 2. The figure shows the reference trajectory as the solid curve with a dot on the curve

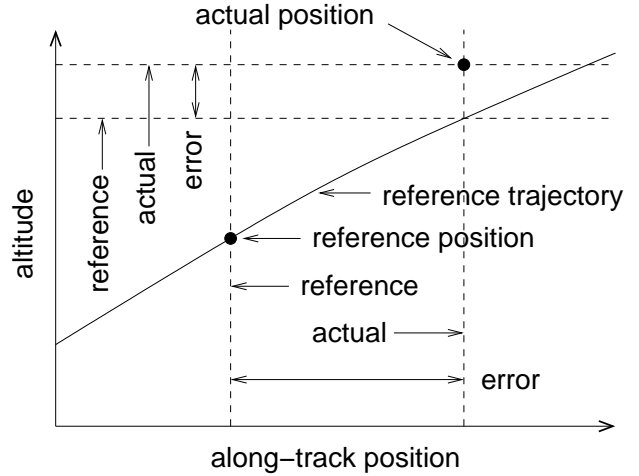


Figure 2: Example showing that reference altitude is a function of *actual* (not reference) along-track position.

to indicate the reference position at a particular time. The other dot in the upper right portion of the figure indicates the *actual* position of the aircraft. The along-track position error is shown as the difference between the actual and reference positions. Similarly, the altitude error is shown as the difference between the actual and reference altitudes. The key point of the figure is that the reference altitude is a function of the *actual*, rather than reference, along-track position. As mentioned earlier, this fixes the 3D flight tube in the earth-fixed coordinate system.

Figure 3 illustrates the same concept from a different perspective. The figure shows a notional block diagram of a system in which the main input is time, and the outputs are latitude, longitude, and altitude. For each segment, the along-track reference position is specified as a function of time (a low-order polynomial function to be discussed later). The key point here is that, whereas the latitude and longitude can be considered functions of the along-track *reference* position, the altitude is a function of the *actual* along-track position (another low-order polynomial function to be discussed later). The actual along-track position is the sum of the reference along-track position and the along-track position error, as shown in the figure. Thus, the along-track position error constitutes a second input, and the reference altitude is a function of along-track position rather than time. The time-referenced altitude would be the actual reference altitude only if the along-track position error were zero.

In case this distinction is still unclear or seems unnecessary, a simple thought experiment should help clarify and justify it. Consider a trajectory that is specified all the way through final approach to actual landing. Now suppose the flight is one minute behind schedule (but still within tolerance). If altitude is specified only

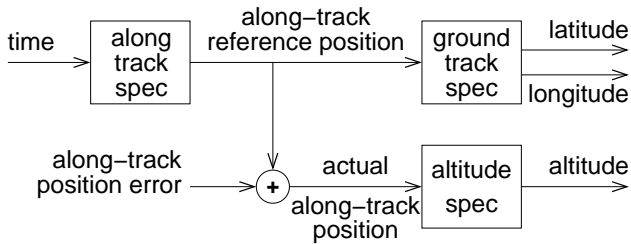


Figure 3: Notional block diagram showing that altitude is a function of *actual* rather than reference along-track position.

as a function of time, then the aircraft will be required to touch down one minute—or several miles—before it reaches the runway! That obviously won’t work. But if altitude is specified as a function of the scalar along-track position, the flight will be required to touch down at the correct point on the runway regardless of how far ahead of or behind schedule it might be. Clearly, the latter approach is correct.

POLYNOMIAL APPROXIMATION

In the current air traffic system, vertical profiles are difficult to predict accurately based on information available on the ground. Part of the problem is that weight and thrust (or throttle setting) are not accurately known on the ground. Another major source of altitude prediction uncertainty is the lack of knowledge of the actual time of initiation of an altitude transition. When told to climb or descend, the time taken by the pilot to initiate the maneuver can vary by up to nearly a minute. As a result, controllers must reserve a large block of airspace around any aircraft that is in, or is about to enter, an altitude transition. With better information available on the ground, altitude can be assigned more precisely, which will increase airspace capacity.

The objective of specifying a vertical profile is to provide reasonable bounds on altitude without significantly compromising efficiency. The idea is to approximate the vertical profile that the aircraft would be most likely to fly if unconstrained (or with normal arrival time constraints), then assign that profile, along with reasonable error tolerances. In climb and descent, commercial transport airplanes normally fly with the throttle fixed and with feedback to the elevator to maintain constant CAS (at lower altitudes) or constant Mach (at higher altitudes). In the future, the intended CAS/Mach schedule will be known on the ground, as will the throttle setting and the estimated weight of the aircraft. The predicted wind, temperature, and pressure fields will also be available from a centralized weather service. Given this data, the vertical profile can usually be predicted fairly accurately, and an approximation of the predicted profile can

be used as the assigned profile. If the wind data is reasonably accurate, and if the altitude tolerances are reasonable, the aircraft should be able to conform to the specified trajectory by flying as usual with the specified power and CAS/Mach schedule.

The mechanics of the procedure will be similar to what is currently done in the Center/TRACON Automation System (CTAS) [19], but with a few key differences. CTAS is a suite of ATC/ATM decision support tools that is being developed at NASA Ames Research Center. CTAS currently has to guess at the weight and the CAS/Mach schedule to be flown, but those data will be available from the aircraft or from the AOC. A more fundamental difference is that the predicted trajectory will actually become the assigned trajectory if it is free of conflicts; otherwise it will be modified to eliminate any conflicts before becoming the assigned trajectory. The current ATC system has no such precisely defined vertical profiles, and a CTAS prediction is no guarantee of actual, or even attempted, conformance. In fact, the notion of vertical conformance itself isn’t even defined for altitude transition.

The CTAS software process that predicts trajectories is called the Trajectory Synthesizer (TS) [20]. The TS contains performance models of all major aircraft types, and types that are not modeled directly are approximated with similar available models. The inputs to the TS for each aircraft include the aircraft type and weight, CAS/Mach values, throttle settings, the flight-plan, and the current weather data file from the Rapid Update Cycle (RUC) [21]. The output is the predicted 4D trajectory in the form of a discrete series of points in which the time increment varies with the dynamic state. The TS or its functional equivalent can be used to construct the assigned trajectory as closely as possible to the trajectory that would have been flown without the constraints. The common trajectory modeling capability currently being discussed by the FAA and Eurocontrol [22] could eventually be applied here.

Figure 4 shows the altitude profile synthesized by the TS for a constant-CAS climb segment of a Boeing 757 from an altitude of about 12,000 ft, where it exits the TRACON, to 34,000 ft, in a randomly selected wind field. The solid line represents the best-fit parabola, and the dashed lines represent an example error tolerance of ± 2000 ft relative to the parabola. The constant-CAS segment is followed by a short constant-Mach segment (not shown), which would require its own curve fit. In most cases, the aircraft should be able to fly the specified CAS of 296 knots without altitude feedback or throttle modulation, and stay within the specified altitude range. Only if the TS is substantially in error would the aircraft need to use feedback of altitude, and perhaps also throttle modulation, to stay within tolerance. Such error could be due to errors in wind, thrust, and/or weight. Altitude feedback, and perhaps throttle modulation, could be activated when the altitude deviation from

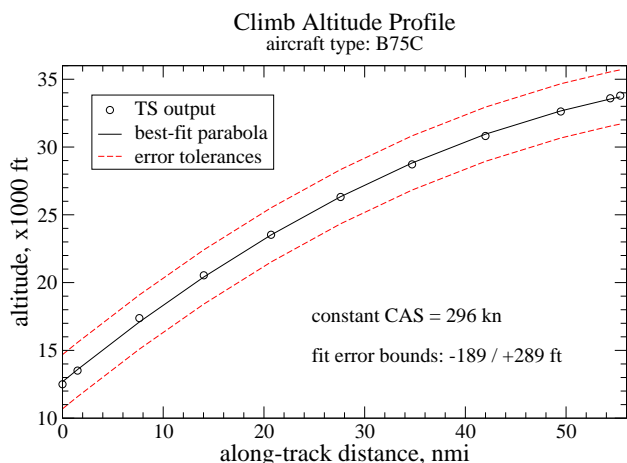


Figure 4: Synthesized altitude profile and best-fit parabola for the constant-CAS segment of a Boeing 757 in climb.

the reference reaches a threshold value of, say, 1000 ft. Alternatively, an FMS could be programmed to use a more refined criterion.

The curve fit error bounds of the parabola in figure 4 are -189 to $+289$ ft, for a total range of 478 ft (the curve could be offset to make the error bounds symmetric, but that was not done here). With a vertical error tolerance of ± 2000 ft, that fit allows a worst-case altitude deviation, relative to the TS output, of $189 - 2000 = -1811$ to $2000 - 289 = +1711$ ft, which is probably sufficient. However, if the error tolerance were tighter, say ± 1000 ft, that quadratic fit would only leave a worst-case altitude deviation of -811 to $+711$ ft, which might not be considered sufficient. In that case, the segment could be divided into two or more segments, or a cubic or quartic polynomial could be used for a better fit. For this example, a cubic polynomial gives fit error bounds of -178 to $+94$ ft (272 ft range), and a quartic gives -102 to $+68$ ft (170 ft range). Polynomials of fifth order or higher could have numerical problems and should be avoided, but polynomials of fourth order or less should not suffer from significant numerical roundoff errors if a consistent numerical precision of 64 bits is used in both ground-based and airborne computers.

In the case of engine problems that prevent climbing at a normal rate, the aircraft should notify the ground immediately of the expected non-conformance, and the ground will then reroute any affected aircraft. When routing under climbing aircraft, precautions could be taken to minimize the chance of a conflict in the case of engine failure. For example, trajectories could be determined for all levels of engine power from full power down to one engine out, and those trajectories could be avoided. The lower altitude tolerance, which will be dis-

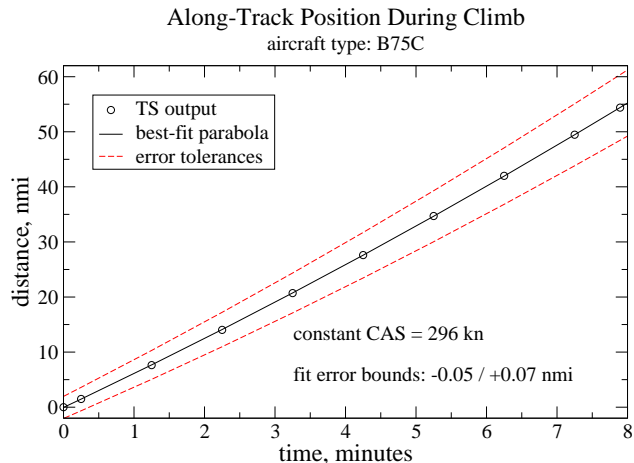


Figure 5: Synthesized along-track position and best-fit parabola for the constant-CAS segment of a Boeing 757 in climb.

cussed later, could be used for this purpose.

Figure 5 shows the along-track position associated with the climb of figure 4. The best-fit parabola fits with error bounds of -0.05 to $+0.07$ nmi, which is close enough for all practical purposes. The example error tolerances represented by the dashed lines start out at ± 2 nmi and grow linearly with time at a rate of 0.5 nmi/min to ± 6 nmi at 8 min from the start of the climb.

Figure 6 shows the altitude profile synthesized by the TS for a constant-CAS, idle-thrust descent segment of a Boeing 727 from an altitude of about 30,000 ft to 11,000 ft, where it enters the TRACON, in a randomly selected wind field. Again, the solid line represents the best-fit parabola, and the dashed lines represent a hypothetical error tolerance of ± 1500 ft. The constant-CAS segment is preceded by a short constant-Mach segment (not shown), which would require its own curve fit. Again, the aircraft should normally be able to fly the constant CAS of 280 knots without altitude feedback or throttle modulation and stay within the specified altitude range. As before, altitude feedback, and perhaps throttle modulation, could be activated when the altitude deviation reaches some threshold value. With error bounds of -53 to $+116$ ft, the curve fit for this descent is much more accurate than for the climb of figure 4. Descents tend to be more nearly linear than long climbs, and are usually well modeled with a parabola. In general, an arrival descent would be followed by a short level cruise segment into the meter fix, which would allow the aircraft to cross the meter fix at a precise level altitude.

Figure 7 shows the along-track position associated with the descent of figure 6. The best-fit parabola fits with error bounds of -0.06 to $+0.06$ nmi, which is close enough for all practical purposes. The example error

mat to be presented in the next section.

PROPOSED XML FORMAT

The purpose of XML (Extensible Markup Language) is to create application-specific standards for platform-independent data specification and transfer. Unlike its more specialized sibling HTML (Hyper-Text Markup Language), XML allows standards designers to define their own data structures. XML is rapidly replacing binary formats for automated business-to-business transactions and is being widely used for computing standards such as Scalable Vector Graphics (SVG). XML provides flexibility in the selection and ordering of the data fields, which is indispensable for trajectory specification because each trajectory can have a variable number of segments of various types. The flexibility also allows trajectories to be updated without repeating all of the data that remains unchanged from the previous update. Note that the use of XML is not required for this application, but its versatility, popularity, and standardization seem to make it a good choice at this time. That could change in the future, of course.

The structure and form of an XML document can be formally described by another document called a Document Type Definition (DTD), which itself has a standard format. However, DTDs are expected to eventually be superseded by XML Schema, which are considered more versatile. XML Schema use XML itself to define the allowed structure and form of other XML documents. Other alternatives to DTD and Schema are also being considered for standardization, but these alternatives are outside the scope of this paper. The objective of this paper is not to formally define an XML format but rather to suggest how the format might look and what information it should contain. A formal XML Schema (or equivalent) would be premature at this time and will not be presented. Instead, example XML code will be presented and discussed informally, but in sufficient detail to provide high-level design requirements for a Schema.

At the most basic level, an XML document consists of a hierarchy of elements, each of which can contain subelements and/or attributes. Consider, for example, the following XML fragment:

```
<elem attr="yes">
  <sub attr2="100" attr3="no"/>
</elem>
```

The main delimiters in XML are “<” and “>,” which enclose the opening and closing tags of each element or subelement. This example shows an element called “elem,” which has an attribute called “attr” and a subelement called “sub.” The names of elements and attributes are selected for each particular application and are case sensitive. All attributes are specified in the opening tag of an element, and the closing tag contains

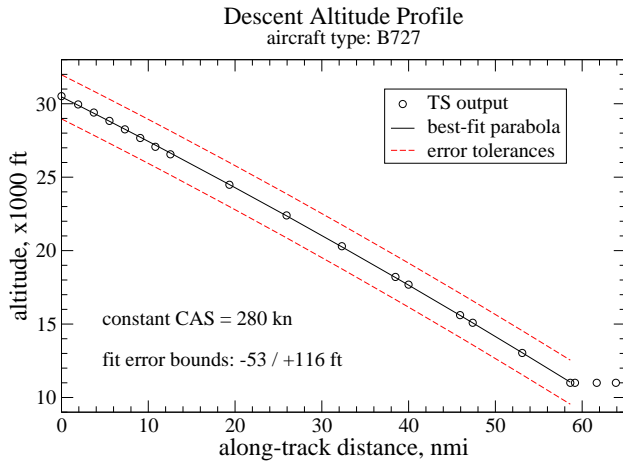


Figure 6: Synthesized altitude profile and best-fit parabola for the constant-CAS segment of a Boeing 727 in descent.

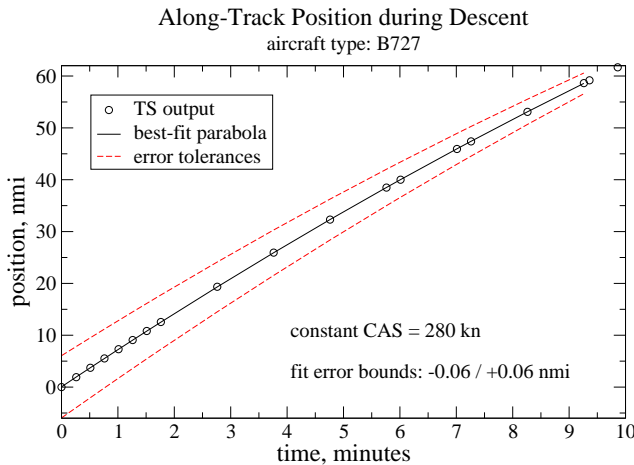


Figure 7: Synthesized along-track position and best-fit parabola for the constant-CAS segment of a Boeing 727 in descent.

tolerances represented by the dashed lines start out at ± 6 nmi and contract linearly with time at a rate of 0.5 nmi/min to slightly less than ± 2 nmi at 9 min from the start of descent. Climbing aircraft usually only need to control airspeed using the elevator (with throttle fixed). However, descending arrivals need to control both altitude and precise time of arrival at the meter fix, so they need tighter control using not only the elevator but possibly also the throttle or speed brake.

The polynomial approximations discussed in this section will be used in the trajectory specification for-

Table 1: DEFAULT PHYSICAL UNITS

quantity	unit
time	hh:mm:ss (XML time format)
horizontal distance	nautical miles (nmi)
altitude	100 feet (100 ft)
angles	degrees (deg)
horizontal speed	knots
vertical speed	feet/minute (ft/min)
weight	1000 pounds (klbs)

```
<flight ...>
  <aircraft ...> ... </aircraft>
  <preferences> ... </preferences>
  <trajectory ...> ... </trajectory>
</flight>
```

XML Sample 1: Top-level structure

the element name preceded by a forward slash, such as “</elem>” above. Elements or subelements that contain only attributes and no subelements can end the opening tag with “/>” in place of a separate closing tag, as shown above. Attribute values must always be in quotes, and the allowed values can be restricted to a specified list. Attribute values can also be of specified types, such as character strings, integers, and decimal numbers.

In this application, XML will be used to specify several physical quantities such as times, distances, speeds, etc. The units could be specified explicitly, but that would be unnecessary in most instances because standard units are well established in aviation. Horizontal distances are virtually always specified in units of nautical miles (nmi), for example, and altitudes are always specified in units of feet (ft), 100 ft, or 1000 ft. In this paper, the default units will be as shown in table 1 unless otherwise specified. In the unlikely event of a switchover to metric units, the default English units can simply be overridden using attributes. Time will be specified in the standard XML format of “hh:mm:ss” (two digits each for hours, minutes, and seconds).

At the top level, the proposed XML trajectory specification format appears as shown in XML sample 1. Ellipses (“...”) represent text that has been omitted for simplicity. The root element is “flight,” and it contains the top-level elements “aircraft,” “preferences,” and “trajectory.” Note that these element names (and those to follow) could be abbreviated if datalink bandwidth is a problem, but full names will usually be used in this paper for clarity. The “aircraft” element gives information about the aircraft itself. The “preferences” element provides information about the airline or pilot preferred flight parameters. Finally, the “trajectory”

```
<flight ID="AAL2332/SFO" dest="JFK",
  date="2004-02-04" time="13:25:00"
  CID="324459" bcode="2187"
  rev="0.0.0" status="request">
```

XML Sample 2: Flight element attributes

element specifies the trajectory itself. Each of these elements will be discussed in more detail below. Note that the “aircraft” and “preferences” elements can be specified once and need not be repeated each time the trajectory is revised, unless they are revised too.

The root element “flight” has several attributes, as shown in XML sample 2. The “ID” attribute gives the standard flight identification number with the originating airport code (for the current leg of the flight) appended after a slash. The originating airport could be another attribute, but appending it to the flight identification number helps prevent confusion with previous or subsequent legs of the same flight (which could be in the system at the same time). The “dest” attribute gives the destination airport code. The “date” and “time” attributes specify the scheduled departure date and time. The “CID” and “bcode” attributes give a unique computer identification number and the aircraft transponder beacon code, respectively. The “rev” element gives the revision number of the trajectory in a format to be discussed later.

The “status” attribute tells whether the information to follow is a “request” or is actually “assigned.” Another possible status might be “mandatory,” which might apply when an imminent conflict is being resolved. The question of whether or when a pilot has veto power over an assigned trajectory is an operational issue that is outside the scope of this paper. Note, however, that if a trajectory is tentatively assigned but pending approval by the pilot, then both the tentative trajectory and the active trajectory need to be kept clear of new conflicts (as a result of new trajectory assignments) until the pilot decides whether or not to approve.

The pilot or Airline Operations Center (AOC) need not specify an actual trajectory request if they are not equipped to do so or if they do not wish to do so. They can simply specify their aircraft and flight preferences and let the ground systems specify and assign a trajectory. To do so, the pilot or AOC would use the “aircraft” and “preferences” elements and omit the “trajectory” element. The revision number of “0.0.0” shown above applies in that case.

When the trajectory is initially assigned, the revision number in the “rev” attribute will become “1.0.0,” and the “status” attribute will change from “request” to “assigned.” The initial assigned trajectory could be fully specified from start to finish, or it could be fully

```
<flight ID="AAL2332/SF0" CID="324459"
  assigntime="14:05:32"
  devtime="14:09:52" rev="1.0.2"
  status="assigned">
```

XML Sample 3: Flight element attributes for trajectory revision

specified for, say, the first hour, and only the horizontal route tentatively specified for the remainder of the flight, pending later, more precise specification. When a trajectory is actually assigned, the root element “flight” will have another attribute called “assigntime,” which gives the assignment uplink time.

When the trajectory is revised, a trajectory deviation time will also be specified in an attribute called “devtime.” The revised trajectory must be continuous with the old trajectory so the aircraft can maintain continuous conformance, and “devtime” specifies the time at which the new trajectory actually deviates from the old. The deviation time must follow the assignment time by a sufficient margin (yet to be determined) to allow the new trajectory to be uplinked, accepted, and processed onboard the aircraft. An example of the “flight” tag for a trajectory revision appears in XML sample 3. Note that the scheduled departure date and time need not be repeated because they are constant, and the same applies to the destination airport and beacon code, assuming they haven’t changed. However, the flight identification and the computer identification number need to be given for positive identification.

During flight, a trajectory modification request can be specified and downlinked to the ground for approval at any time. The requested trajectory will then be checked on the ground, and if it is free of conflicts for some specified time horizon on the order of, say, 15 min, and if it is also consistent with the longer range traffic flow plan, it will be approved. In that case, no assigned trajectory needs to be uplinked. Instead, an MD5 or SHA-1 [23] message digest signal can be uplinked to approve the request and guarantee integrity. However, if the requested trajectory needs to be further modified to resolve conflicts or for some other reason, the modified trajectory will be uplinked. In that case, the aircraft will downlink an MD5 or SHA-1 message digest of the assigned trajectory to guarantee integrity.

Aircraft

As mentioned above, the “aircraft” element, a top-level subelement of the root element, will be used by the AOC or the pilot to specify the aircraft model and parameters. An example is shown in XML sample 4.

The “tail” attribute of the “aircraft” element gives the tail number of the aircraft. The “model” at-

```
<aircraft tail="N788" model="MD80">
  <weight unit="klbs" value="135"/>
  <fuel unit="gal" amount="5226"/>
  <engine model="JT9D" factor="0.98"/>
  <equip code="GAF" status="1"/>
</aircraft>
```

XML Sample 4: Aircraft element

tribute gives the aircraft model, which will be selected from an approved list of several hundred models. The ground systems will have performance models of each aircraft type, which will be used to construct an efficient trajectory when a fully specified trajectory request is not received from the aircraft or its AOC.

The “weight” element specifies the takeoff weight of the aircraft in units specified by the “unit” attribute. The units of “klbs” (1000 lbs) shown in the example could be the default, and other options such as “kg,” for kilograms, might be allowed. The “fuel” element gives the amount of fuel stored at takeoff and could be used to update the fuel status if it becomes a concern. The “engine” element has attributes “model” and “factor,” which specify the engine model and an optional thrust factor, which defaults to “1.0.” The thrust factor scales the nominal maximum thrust for that engine model, and it could be used to provide a more precise maximum thrust for that particular engine, if known. The “equip” element gives an avionics equipment code and a functional status code that could be considered optional if everything is functioning properly.

Flight Preferences

As mentioned above, the “preferences” element, a top-level subelement of the root element, is used by the AOC or pilot to specify preferred flight parameters. It gives the ground systems the basic parameters needed to construct an efficient trajectory consistent with the airline or pilot preferences. An example is given in XML sample 5.

Most of the information in the “preferences” element is self-explanatory. The “climb” and “descent” elements each have a “thrust” attribute to specify a preferred thrust power setting in percent of maximum thrust, where maximum thrust is determined by the engine type as specified in the “engine” element discussed previously. Climbs and descents usually consist of a long constant-CAS (Calibrated Airspeed) segment at low altitudes and, for jets, a constant-Mach segment above the CAS/Mach transition altitude. The “climb” and “descent” elements each have “CAS” and “Mach” attributes to specify the constant CAS and constant Mach to be flown. The “cruise” element specifies the preferred cruising altitude and Mach or CAS.

```

<preferences>
  <climb thrust="90" Mach="0.74"
    CAS="280"/>
  <descent thrust="5" Mach="0.74"
    CAS="290"/>
  <cruise alt="310" Mach="0.76"/>
  <turn radius="9.0"/>
  <depart runway="10L" order="23"/>
  <arrive runway="22L" order="14"/>
  <route>xxx.xxx,xxx.xxx
    xxx.xxx,xxx.xxx
    xxx.xxx,xxx.xxx
    xxx.xxx,xxx.xxx</route>
</preferences>

```

XML Sample 5: Preferences element

The “turn” element has a “radius” attribute to specify the preferred default turn radius. Alternatively, a maximum bank angle for a coordinated turn could be specified using a “bank” attribute. The “depart” and “arrive” elements each have a “runway” attribute to make known the preferred runways. They also each have an optional “order” element to specify the preferred takeoff or landing order relative to other flights of the same airline company. Finally, the “route” element can be used to specify the desired horizontal route waypoints in terms of WGS84 (latitude and longitude) points. Each waypoint could be a comma-separated pair, and the waypoints could be separated by spaces, the same format used to represent polygon vertexes in the XML standard for Scalable Vector Graphics (SVG).

Trajectory

The actual trajectory is specified in the “trajectory” element, a top-level subelement of the root element. It could be specified by the aircraft as a request, or it could be specified on the ground as an assignment. The “trajectory” element consists of a preamble, followed by a series of “segment” elements, each of which specify a trajectory segment. An example is shown in XML sample 6.

The “trajectory” element has attributes “reftime” and “wthr.” The “reftime” attribute specifies the trajectory reference time, relative to which all other times in the trajectory will be specified. By changing this reference time, the entire trajectory can be shifted in time when necessary, which will be discussed later. The zero reference for along-track position will correspond to the assigned position of the aircraft at this trajectory reference time. As the flight progresses, past segments can be dropped and the reference time can be moved up to the beginning of the current trajectory segment, if desired. The “wthr” attribute specifies the

```

<trajectory reftime="13:46:17"
  wthr="2004-02-04T13:00">

  <depart runway="10L"/>
  <tolerances>
    <cross tol="2.0"/>
    <vert tol="2"/>
    <along tol="-2.0 2.0"
      rate="-10 10" time0="0:32"
      max="-10 10"/>
  </tolerances>

  <segment ...> ... </segment>
  <segment ...> ... </segment>
  ...
  <segment ...> ... </segment>

</trajectory>

```

XML Sample 6: Trajectory element

exact weather data that was used in the construction of the trajectory. The “depart” element has an attribute called “runway” that is used to specify the departure runway, and an “arrive” element could be added later in the flight to specify the arrival runway.

The “tolerances” subelement of “trajectory” is used to specify the default trajectory tolerances. It contains “along,” “cross,” and “vert” subelements to specify the default along-track, cross-track, and vertical tolerances, which can be overridden in each trajectory segment. Each of those elements has a “tol” attribute that is used to specify the actual tolerances. The default cross-track tolerance will always be symmetric left and right, so only a single value is given, in the default units of nmi. The default vertical tolerance will apply only to level flight, and it will also be symmetric up and down, so a single value is given in the default units of 100 ft.

The along-track tolerance is slightly more complicated. While unexpected cross-winds can be compensated for with only minor loss of efficiency, the same is not necessarily true of unexpected along-track winds (headwinds or tailwinds). Attempting to compensate for errors in the along-track wind predictions to maintain a specified groundspeed can result in flight at an inefficient airspeed. Tightening the along-track tolerances increases airspace capacity, but it also increases the probability that aircraft will be required to fly at inefficient or perhaps even unflyable airspeeds. Along-track position error tolerances must be set as a compromise between those two effects. To allow more flexibility, along-track tolerances will be allowed to vary linearly with time and to be asymmetric. Also, the along-track assigned position, tolerances, and speed will be updated periodically


```

<segment number="1" vtype="climb"
  htype="straight" stype="constCAS">

  <time start="0:08:42" duration="7:42"/>
  <begin coords="WGS84" lat="xxx.xxxx"
    lon="xxx.xxxx"/>
  <end coords="WGS84" lat="xxx.xxxx"
    lon="xxx.xxxx"/>
  <along coeffs="xxx.xxx xxx.xxx"
    CAS="280" length="27.815"/>
  <alt coeffs="126.8 21.609 4.1417e-3"
    thrust="90" end="270" max="272"/>
  <tolerances>
    <along rate="-15 15"/>
    <vert tol="-15 10"
      rate="-1.5 1"/>
  </tolerances>
</segment>

```

XML Sample 7: Trajectory segment element

to compensate for errors in modeling and prediction of along-track winds, as will be discussed later.

The “tol” attribute of the “along” element, therefore, has two values: the forward and rear tolerances, in units of nautical miles. These are the initial tolerances at the time specified in the “time0” element, which is relative to the trajectory reference time specified in the “reftime” attribute of the “trajectory” element discussed earlier. The actual tolerances vary as a function of time at a rate specified in the optional “rate” attribute, if specified, which also has two values, in units of knots, that default to zero. However, the maximum magnitude of the along-track tolerances is capped by the optional “max” attribute, if specified.

Trajectory Segments

The actual trajectory is specified as a series of segments, each of which is specified in a “segment” subelement of the “trajectory” element, which is itself a top-level subelement of the root element. An example of a “segment” subelement is given in XML sample 7.

The “number” attribute of “segment” gives the segment number in the sequence. This is redundant information if the segments are listed in order, but it provides a handy reference tag and minimizes the chance of confusion. The “vtype,” “htype,” and “stype” attributes specify the type of the segment, which determines which other subelements apply. The “vtype” element specifies the vertical type, “htype” specifies the heading type, and “stype” specifies the speed type. The allowed types are listed in table 2. The vertical types are climb, level, and descent. The heading types are right turn, left turn, and straight. The speed types are constant CAS, constant

Table 2: TRAJECTORY SEGMENT TYPES

name	description
vtype	vertical type
climb	increasing altitude
level	constant altitude
descent	decreasing altitude
htype	heading type
rturn	right turn
lturn	left turn
straight	greatcircle path
stype	speed type
constCAS	constant CAS
constMach	constant Mach
speedup	increasing CAS or Mach
slowdown	decreasing CAS or Mach
dependent	dependent CAS or Mach

Mach, speedup (increasing CAS or Mach), slowdown (decreasing CAS or Mach), and dependent. Note that the speed types are based on airspeed (CAS or Mach) rather than groundspeed, which changes during altitude transition even when CAS or Mach are constant with no winds. Whether speedup or slowdown refer to CAS or Mach will depend on which is specified, which will be discussed shortly. The dependent type is a catch-all for cases where CAS or Mach could vary depending on other requirements, such as a constant flightpath angle for climb or descent. The total number of permutations of possible segment types is $3 \times 3 \times 5 = 45$, though some will rarely or never be used. The rationale behind these trajectory segment types will be discussed in more detail later.

The “time” subelement of “segment” specifies the precise time range of the segments. The “start” attribute gives the starting time of the segment relative to the trajectory reference time, and the “duration” attribute gives the duration. The “begin” and “end” subelements of “segment” specify the exact ground position of the beginning and end of the segment. The attribute “coords=“WGS84”” shown in the example indicate that the coordinate are latitude and longitude in the WGS84 geodetic reference system, which could be considered the default. (The beginning of a segment must match the end of the previous segment, so it is redundant information that could perhaps not be required if bandwidth is an issue.)

Local coordinate systems could also be convenient in terminal areas around major airports. A standard stereographic (pseudo-Cartesian) coordinate system can be defined for each major airport or TRACON. The “coords” attribute will be the airport code, and the coordinates will be specified, in units of nautical miles, with “x” and “y” attributes, as they are traditionally called. For example:

```
<begin coords="DFW" x="34.344" y="9.439"/>
```

All straight segments will be assumed to follow greatcircles, regardless of the coordinate system used to specify their endpoints. Note that when the segment length is less than, say, 15 nmi, the difference between a greatcircle and a straight line in stereographic coordinates is very small. Finally, conventional named waypoints will still be allowed, and will be designated with “`coords="named".`” For example:

```
<end coords="named" name="GREGS"/>
```

Named waypoints could be useful for arrival meter fixes, as in the current system.

Referring back to XML sample 7, the “`along`” subelement of “`segment`” specifies the along-track position as a polynomial function of time. The “`CAS`” attribute gives the nominal CAS that is expected to be flown. It could be replaced by a “`Mach`” attribute when appropriate, of course. The “`length`” attribute gives the along-track length of the segment. The “`coeffs`” attribute lists the coefficients of the polynomial in order of increasing powers, starting with the constant term. The distance units will be nautical miles for all coefficients. The linear coefficient should be in standard units of knots (nmi/hr) for readability, but the succeeding coefficients should probably be in units of knots/min, knots/min/min, etc., for better scaling. Consistent units are desirable, but readability and reasonable scaling are more important. As long as the units are clearly specified in the standard, the mixing of hours and minutes should not be a problem. The value of time used as the argument of the polynomial will be the time *relative to the start of the segment*. Thus, the first (constant) coefficient will give the along-track (scalar) starting position of the segment, where the zero reference for along-track position corresponds to the assigned reference position of the aircraft at the trajectory reference time. The second coefficient is the groundspeed, in units of knots, at the start of the segment.

Referring back again to XML sample 7, the “`alt`” subelement of “`segment`” specifies the altitude as a function of along-track (scalar) position. The “`thrust`” attribute gives the nominal thrust for the climb segment, in percent of full power, and the “`end`” attribute gives the assigned altitude at the end of the segment. The “`max`” attribute gives an upper altitude limit that overrides the vertical tolerance near the end of the climb, as will be discussed later. The “`coeffs`” attribute gives the coefficients of the altitude as a function of *actual* (not reference) along-track position, as was illustrated in Figs. 2 and 3. As with the “`along`” element, the coefficients will be listed in order of increasing powers. The distance used as the argument of the polynomial will be the along-track position, in units of nmi, relative to the start of the segment. Thus, the first (constant) coefficient will be the altitude, in hundreds of feet, at the start of the segment. The second coefficient will be the

```
<tolerances>
  ...
  <along type="relative" tol="-2.0 -3.0"
    rate="-10" time0="0:32"/>
</tolerances>
```

XML Sample 8: Relative along-track tolerance

altitude rate at the start of the segment, in units of 100 ft/min.

An interesting side issue is whether altitude should be specified in terms of geometric altitude or pressure altitude. In the United States, pressure altitude is currently used above the transition altitude of 18,000 ft, and aircraft performance charts are based on pressure altitude. However, geometric altitude can be determined by GPS/WAAS (Global Positioning System with Wide Area Augmentation System) much more accurately than pressure altitude can be measured using a baro-altimeter. GPS/WAAS can also determine geometric altitude *rate* much more accurately than pressure altitude rate can be determined. Accurate altitude and altitude rate determination are indispensable for rapid detection of trajectory nonconformance or impending nonconformance. Accurate altitude control is also very desirable for the “continuous altitude rule” that was proposed in reference [24]. This rule designates cruising altitudes as a continuous function of course angle, which spreads cruising traffic vertically, greatly reducing the chance of collision. For these reasons, geometric altitude should eventually replace pressure altitude as the standard for trajectory assignment.

Referring back yet again to XML sample 7, the “`tolerances`” subelement of “`segment`” specifies the trajectory error tolerances for the segment. It has the same basic format as the “`tolerances`” subelement of the “`trajectory`” element one level up in the XML tree, which was used to specify the *default* tolerances for all segments. If no tolerances are specified for a particular segment, the defaults apply. If tolerances are specified for a particular segment, they apply only to that segment.

During climb and cruise, the along-track position error tolerance will typically expand with time in both directions, which means that the lower bound of the along-track tolerance rate will be negative and the upper bound will be positive. That would be reversed for descent, however, if the aircraft targets a precise arrival time at a meter fix. For in-trail arrivals on the same flightpath, along-track tolerances could conceivably be relative. That is, the forward along-track tolerance for one aircraft could be specified relative to the preceding aircraft. This would obviously require that the following aircraft be equipped to track the leading

```

<segment number="1" vtype="level"
  htype="rturn" stype="constMach">

  <time .../>
  <begin .../>
  <end .../>
  <alt .../>

  <turn begin="84.6" angle="7.3"
    end="91.9" radius="15.0">
    <wind speed="46" dir="243"/>
  </turn>

  <tolerances> ... </tolerances>
</segment>

```

XML Sample 9: Turn segment

aircraft. Such a relative along-track tolerance could be specified, for example, as shown in XML sample 8. The attribute “tol=-2.0 -3.0” would be interpreted as follows. The -2.0 is the along-track rear tolerance, as before. The forward tolerance of -3.0, on the other hand, indicates that the aircraft must stay at least 3.0 nmi behind the preceding aircraft.

Also, for non-level (climbing or descending) segments, the vertical tolerances can be asymmetric and will be allowed to vary linearly for enhanced flexibility. Hence, the “vert” subelement has a “tol” attribute that specifies both a lower and an upper tolerance in units of 100 ft, and it also has a “rate” attribute that specifies the rate of change of those tolerances in units of 100 ft/nmi. The implied along-track (scalar) reference position for vertical segments is the beginning of the segment. This is different from the along-track reference time, which is specified independently of any particular segment, as discussed earlier.

The “along” subelement of “segment,” as shown in XML sample 7, applies to straight (greatcircle) segments. For turning segments, all the same elements apply except “along,” which is replaced with “turn.” An example of a turn segment is shown in XML sample 9. The elements in common with all segment types are shown in abbreviated form for simplicity.

The “turn” subelement has attributes “begin,” “end,” “angle,” and “radius.” The “begin” and “end” attributes specify the course angle at the beginning and end of the turn, and the optional “angle” gives the angle of the turn, where positive is to the right (“angle” is for the convenience of the human reader). Note that course angle is the angle of the groundtrack, independent of winds, where zero is due North and 90 deg is due East. The “radius” attribute gives the radius of the turn in units of nautical miles, as usual, and it should

probably be limited to something like 100 nmi. The “turn” element has one subelement, “wind,” with attributes “speed” and “dir” to specify the speed and direction of the wind in units of knots and degrees, respectively. Since turns are relatively short in duration, the wind field will be assumed to be uniform throughout the turn. These parameters precisely determine the horizontal position of the aircraft as a function of time throughout the turn.

The wind vector is added to the airspeed “vector” (based on airspeed and heading relative to the wind) to determine the varying groundspeed of the aircraft as it progresses through a turn. The law of cosines is then used to compute the resulting groundspeed, which can be numerically integrated to precisely determine the along-track position as a function of time. In any non-zero wind field the groundspeed will vary through the turn, and the bank angle must be varied accordingly to maintain a coordinated turn (in which gravity and centrifugal acceleration add vectorially to a force normal to the floor of the aircraft). The necessary bank angle, ϕ , is given in terms of the varying groundspeed, v , by $\phi = \text{atan}(v^2/(rg))$, where r is the (constant) turn radius, and g is gravitational acceleration. The radius of the turn can be selected so that the maximum bank angle through the turn does not exceed a specified magnitude.

A few additional points are worth mentioning with regard to trajectory segment types listed in table 2. Higher-order turn dynamics could be modeled with additional trajectory segment types, but the small improvement in accuracy they would provide is not likely to justify the added complexity. Segment types could be added to model the roll dynamics at the start and end of turn segments, for example, but those roll dynamics last only a few seconds and their effect can simply be absorbed into the cross-track error tolerance. Turns of small magnitude should probably be given a large radius (but not more than, say, 100 nmi) to minimize the bank angle and the resulting modeling error due to the roll dynamics. Note also that an FMS can compensate for the effective delay simply by starting the turn a few seconds early. The turn lead time is approximately half of the time needed to bank over (bank angle divided by nominal roll rate).

The normal (vertical) acceleration dynamics at the start and end of altitude transitions are another matter. Those dynamics can last up to approximately twenty seconds, and their effect on the trajectory can be significant. The same sort of lead compensation discussed in the preceding paragraph for roll dynamics can be used, but that may not match the assigned trajectory closely enough. In that case, a short climbing or descending segment of ten to twenty seconds can be used to model the vertical acceleration dynamics. That is, a simple parabolic segment can be used to provide a smooth transition between the level segment and the climbing or descending segment that leads or follows it.

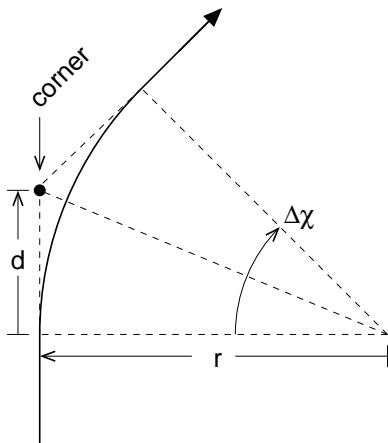


Figure 8: Turn geometry.

To be valid, segments must be labeled with the correct type. For example, level segments must have a constant altitude, climb segments must have a positive altitude rate, and descent segments must have a negative altitude rate. Also, position, groundspeed, and course (groundtrack) angle must all be continuous to form a valid trajectory. This means that a turning segment is required between any two segments for which the course angle at the end of the first segment does not match the course angle at the start of the second. Similarly, a speedchange segment is required between any two segments for which the groundspeed at the end of the first segment does not match the groundspeed at the start of the second. A small discontinuity should be allowed in the vertical speed, however, at the beginning and end of altitude transitions and at CAS/Mach transitions.

The turn angle between adjacent straight (great-circle) segments can be determined using the “course” function given in the Appendix for determining the initial course angle of a greatcircle segment (the spherical-earth equations are plenty accurate for this purpose). The turn angle between greatcircles (A,B) and (B,C) is “course(B,C) – course(B,A) ± 180 deg.” Each turn must be tangent to the two greatcircle segments that it connects. The turn corner is at the intersection of the two greatcircle segments, and a turn of angle $\Delta\chi$ must start at a distance $d = r \tan(\Delta\chi/2)$ before the corner, as shown in figure 8.

The 45 combinations of segment types cover all normal flight modes. The types involving turns and/or air-speed changes will typically be of relatively short duration compared to the straight segment types with constant airspeed. The segment types involving a simultaneous turn and speed change will probably be used rarely but are available when needed. These segment types may seem to preclude certain combinations, such as a turn that continues through a transition from climb to level flight, for example. Note, however, that such a combination could be constructed, if necessary, by sim-

ply following a climbing turn segment with a level turn segment. Note also that sequential turn segments could be used for S-turn delays. They could also be used for unusual flight plans that go back and forth for aerial surveying or other specialized tasks in which the straight segments are parallel and closer together than the minimum acceptable turn radius of the aircraft.

The 45 different possible combinations of segment types are not all parametrically distinct. Climb and descent segments have the same parameters, for example, and differ only by the the sign of the altitude rate. Similarly, level segments are just a special case in which the altitude rate is zero. The climb, level, and descent designations are therefore not necessary for the actual construction of the trajectory; they are essentially for the convenience of the human reader. The only distinction as far as parametrization is concerned is the distinction between turning and straight segment types. The turning segment types need a few additional parameters such as the radius of the turn and the wind vector, which were discussed earlier.

A problem with transitioning from a non-level segment to a level segment is that the altitude tolerance will virtually always be discontinuous. A typical altitude tolerance for level flight might be ± 200 ft, but for climb or descent the tolerance could be ten times larger. Going from a level segment to a non-level segment is not a problem because the tolerance increases, but going from a non-level segment to a level segment is a problem because the altitude tolerances decrease sharply and discontinuously, as shown in figure 9. The linearly decreasing “transition tolerance” shown for the lower altitude tolerance is one possible approach for reducing the tolerance less abruptly. The along-track distance over which the tolerance decreases, which could be something like 5 to 10 nmi, will be specified in the “`ttdist`” (“tolerance transition distance”) subelement of the “`segment`” element for the level segment. Note that “`ttdist`” could also apply to the cross-track tolerance in transitioning from a turn segment to a straight segment. However, it does not apply to the along-track tolerance, which will always be continuous between segments.

Figure 9 also illustrates another problem with transitioning from a non-level segment to a level segment. The upper altitude tolerance during the climb segment allows the aircraft to go significantly above its intended cruising altitude, exposing it to potential conflicts with traffic at the next higher flight level. To prevent such exposure, an “altitude limit” can be specified in the “`max`” attribute of the “`alt`” subelement of “`segment`.” This maximum altitude limit overrides the upper altitude tolerance, as shown in the figure. For a descent segment, the “`max`” attribute would be replaced by “`min`.” This maximum altitude limits any “overshooting” of target altitude and will normally be the upper altitude tolerance for the level segment. The same geometry turned upside down applies to descent.

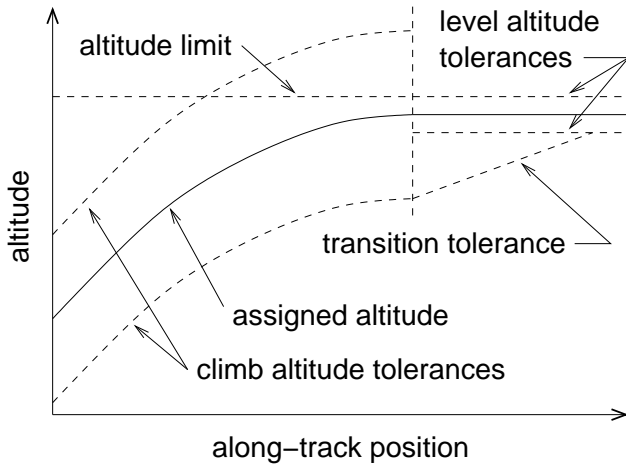


Figure 9: Altitude limit and transition tolerance for typical transition from climb to level segment with decreasing altitude tolerances.

As mentioned earlier, curved wind-optimal routes can be approximated as a series of waypoints connected by greatcircles. The closer together the waypoints are placed, the better the approximation will be, of course. As a practical matter, something like six straight segments are sufficient to maximize wind efficiency in crossing the entire North American continent. Nevertheless, better accuracy could be achieved for a given number of waypoints by using low-order (quadratic or cubic) polynomial cross-track offsets from nominal greatcircle segments. That would allow some curvature to be put in each segment for more flexible routing. It is not clear, however, that the benefits would justify the additional complexity, so it will not be pursued in this paper. Note that the added complexity is not just in defining the reference trajectory itself, but even more so in determining the bounding space. Recall that each segment determines its own local coordinate system.

Also, as mentioned earlier, the “rev” attribute of the root element “flight” gives the trajectory revision number in the form “i.j.k,” where “i” is the horizontal path revision number, “j” is the vertical profile revision number, and “k” is the along-track update number. The revision number will start as “1.0.0” for the first assigned trajectory. If the horizontal path is revised, then “i” will be incremented and “j” and “k” will each be reset to zero. Similarly, if the vertical profile is updated (as in a temporary altitude assignment to resolve a conflict), then “j” will be incremented, and “k” will be reset to 0, but “i” will remain unchanged. If an along-track update occurs, which will be the most common type of update (to correct for wind modeling and prediction errors), then “k” will be incremented and “i” and “j” will remain unchanged. Along-track updates are discussed in the next section.

ALONG-TRACK UPDATES

The three axes in which the error tolerances are specified are along-track, cross-track, and vertical. In terms of energy, the cheapest axis in which to maintain conformance is the cross-track axis. If the aircraft can fly the assigned groundtrack to within the cross-track tolerance, the marginal energy required to do so is usually small. In the case of unexpected crosswinds, the aircraft only has to “crab” to stay on track, and the main controls required are the ailerons and the rudder.

In cruise, vertical conformance is also cheap, but along-track conformance can be far more expensive, depending on how accurately the along-track winds can be modeled and predicted. Airplanes normally cruise at constant airspeed (CAS or Mach), and the groundspeed corresponding to the most efficient airspeed obviously varies with the along-track wind speed. If the wind field prediction is accurate, then an efficient groundspeed can be determined and assigned. If the predicted winds are substantially in error, however, the airspeed corresponding to the assigned groundspeed could be inefficient or even unflyable. The two relevant concerns here are the wind prediction accuracy and the speed range of the aircraft.

Reference [25] cites wind prediction accuracy results for the Rapid Update Cycle (RUC-1) [26] augmented with aircraft wind reports. The wind error vector magnitude was 7.85 m/s (15.3 knots) or less 90 percent of the time, and error vector magnitude exceeded 10 m/s (19.4 knots) only 4 percent of the time. The errors tend to be somewhat worse during the winter months because of higher wind speeds in general, but they are somewhat better than the quoted figures the rest of the year. The errors also tend to be larger at higher altitudes where the wind speeds are higher, but the figures quoted above are from actual aircraft at their operating altitudes. These performance figures can perhaps be expected to improve over the next twenty years. Note also that the along-track component of the wind error, which is the significant quantity here, is less than the vector magnitude. Averaged over all heading directions, the mean headwind error is $2/\pi$ (≈ 0.64) times the magnitude of the error vector (that probably overstates the average effective reduction, however, because neither heading directions nor wind error vectors are uniformly distributed).

The other relevant factor here is aircraft speed range. Figure 10 is a plot of the speed envelope for an MD-80 aircraft at a gross weight of 135,000 lb on a standard day. This figure is taken from reference [27] and is based on data that came from a McDonnell Douglas performance handbook. These data, as well as the aircraft weight, would be known by the ground systems for all equipped aircraft. Any uncertainties in the weight would need to be accounted for to guarantee that the speed range is not overestimated.

The speed bounds shown in figure 10 are the re-

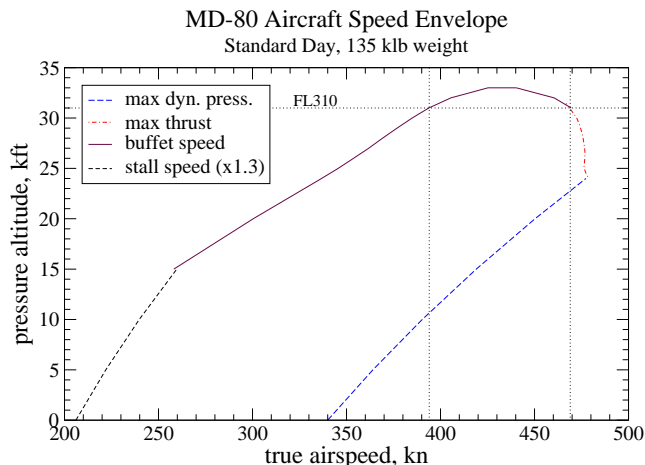


Figure 10: MD-80 Aircraft Speed Envelope for a standard day at 135 klb gross weight.

sult of four different effects, as shown in the figure. At lower altitudes, the upper speed is limited by the maximum dynamic pressure that the airframe can withstand. At higher altitudes, the upper speed is limited by the maximum cruise thrust, which is primarily a function of airframe drag and engine thrust. At the highest altitudes, speed is limited to the minimum and maximum buffet speeds, beyond which airframe vibration becomes excessive (the buffet speed limits in this particular plot may be somewhat conservative). Finally, at lower altitudes, the wing stall speed (multiplied by a safety factor of 1.3) is the effective lower bound on speed.

Figure 10 shows that the speed range falls off sharply as the pressure altitude ceiling of approximately FL330 (for this weight) is approached. At FL310 the speed range is approximately 75 knots, and at FL290 it is slightly below 100 knots. Suppose the aircraft is flying at FL310 at the recommended cruise speed of Mach 0.76, which is equivalent to 446 knots at that altitude. The speed envelope then goes from approximately 394 to 469 knots, as shown in figure 10, so the speed can be increased by a maximum of 23 knots or decreased by a maximum of 52 knots from the recommended speed. Thus, if the wind prediction error is within the range of -23 to $+52$ knots, the aircraft can maintain the assigned groundspeed exactly.

Although the aircraft can fly from 394 to 469 knots at FL310, it cannot fly efficiently over that entire range, of course. Suppose the efficiency is deemed “acceptable” from 430 to 454 knots at that altitude and weight. Then, as long as the wind prediction error is within -8 to $+16$ knots, the aircraft can maintain the assigned groundspeed exactly and still fly with acceptable efficiency. Outside that range, an incentive exists to fly at

an efficient airspeed until a point is reached at which the aircraft can no longer conform to the along-track error tolerances. Hence, additional rules may need to be established for how tightly an aircraft should track the assigned groundspeed, but such rules will not be discussed in this paper.

The aircraft isn’t required to fly the assigned groundspeed exactly; it is only required to stay within the along-track position bounds, which can vary linearly with time. Suppose the along-track tolerance rates are zero, the along-track position tolerances are ± 2 nmi, and the aircraft is centered within the along-track position bounds. If the along-track wind error is within ± 10 knots, the aircraft will be able to maintain its recommended airspeed for at least 12 min, worst case, before it can possibly fall out of conformance.

Alternatively, if the along-track tolerance rates are ± 12 knots, then the along-track tolerances will increase at that rate, which is 0.2 nmi per minute, in each direction. Thus, the aircraft will be able to fly at the recommended airspeed and maintain a constant or increasing margin from the along-track bounds even if the along-track wind predictions are in error by up to ± 12 knots.

Commercial transport airplanes normally climb with throttle fixed somewhere in the range of 85 to 95 percent of full throttle, with feedback to the elevator to maintain constant CAS (at lower altitudes) or constant Mach (at higher altitudes). If the errors in the prediction of the along-track wind speed are reasonably small, the aircraft should still be able to fly in that mode. However, when altitude or along-track position approach their bounds, the throttle may also need to be adjusted to maintain conformance. The feedback to the throttle could be programmed to start automatically when the deviation reaches some threshold magnitude. The error tolerances should be set to accommodate the entire range of potential wind errors to some high level of certainty, say 99.9%. This could make the error tolerances fairly large, but at least they will clearly bound the area (as a function of time) that needs to be avoided by other aircraft. In the current system, the lack of explicit bounds forces controllers to effectively block out excessively large amounts of airspace for climbing and descending aircraft, which reduces airspace capacity.

In the event of substantial errors in the prediction of along-track winds, aircraft could be forced to fly at grossly inefficient speeds to maintain along-track position conformance. Worse yet, they could reach a state in which they are aerodynamically incapable of flying at the speed necessary to maintain conformance. To avoid either of those two undesirable conditions, particularly the latter, the along-track assignments will be updated periodically. The updates could apply to position, speed, and error tolerances. With proper updates, the worst that should happen is that some traffic may be forced to fly inefficiently for short periods of time to avoid a

conflict, but they would obviously never be required to fly at speeds of which they are incapable of flying.

A complete 4D trajectory can (optionally) be filed by each participating aircraft or its AOC prior to takeoff. The trajectory reference time, which is specified by the “`reftime`” attribute of the “`trajectory`” element, will be defined as the time at which the aircraft is expected to cross some predefined marker, such as the end of the takeoff runway. Because takeoff time usually cannot be predicted exactly, the first along-track update will occur immediately after takeoff. When the aircraft crosses the reference marker, its reference time will be adjusted accordingly. Because all other times are relative to the reference time, no other times need to be changed. By adjusting the trajectory reference time, the entire trajectory can be effectively shifted in time.

After takeoff, the FMS will guide the aircraft along its assigned climb trajectory. As explained earlier, conventional feedback of speed error to the elevator will be used to maintain constant CAS or Mach, and feedback to the throttle will be used only if the vertical or along-track deviation reaches some threshold value, which shouldn’t happen often if the wind predictions are reasonably accurate and the error tolerances are reasonable. If the aircraft does drift away from its reference trajectory and approach its along-track error bounds, however, the ground can update the along-track assignment by changing the assigned position, speed, and/or error tolerances. Such updates would be done only if they do not cause a conflict within the conflict time horizon of, say, 15 min.

The most common type of along-track assignment update will be to change the assigned position to the current position and to simultaneously reset the position error tolerances to their initial values. The assigned groundspeed (the rate of change of the assigned along-track position) could also be changed if the wind model is determined to be significantly in error. This kind of update could be done periodically at a rate of, say, once per two minutes, except that it would *not* be done if it produces a potential conflict within the conflict time horizon. A potential conflict is defined as having the bounding spaces of two aircraft come closer together than the required minimum separation. In other words, conformance to their assigned trajectories by any two aircraft must guarantee the minimum required separation.

Because all the times given in the trajectory specification are relative to the trajectory reference time, the entire trajectory can be shifted in time by changing the reference time. That is equivalent to changing the assigned along-track position. As a flight progresses, trajectory segments that are completely in the past can be discarded. Because along-track updates will be a common operation, an abridged format is appropriate, where unchanged data is not repeated. An example of a simple but complete along-track update is shown in XML sample 10.

As explained earlier, the “`devtime`” attribute of

```
<flight ID="AAL2332/SFO" CID="324459"
  assigntime="14:05:32"
  devtime="14:06:32" rev="1.1.3"
  status="assigned">
  <trajectory reftime="14:02:17">
    <tolerances>
      <along tol="-2.0 2.0"
        rate="-10 10" time0="0:00"
        max="-10 10"/>
    </tolerances>
  </trajectory>
```

XML Sample 10: Along-track update

“`flight`” gives the trajectory deviation time, which is the time at which the new trajectory deviates from the previous trajectory. The new trajectory reference time adjusts the trajectory to the current along-track position of the flight, and the along-track tolerance is reset to its nominal initial value. Note also that the last field of the trajectory revision number in the “`rev`” attribute is incremented for an along-track update. Although the entire trajectory specification is not transmitted, it will be stored on the ground and reconstructed onboard the aircraft based on the update and the previously stored trajectory.

TRAJECTORY SOFTWARE OBJECTS

The trajectory specification concept could be the basic building block for the entire ATM system. It could be used for tactical and strategic separation as well as long-range traffic flow management (TFM). The proposed XML format represents trajectories independently of the computer platform and programming language, but it also translates directly into software data structures and objects. XML bindings are available for languages such as Ada, Java and C++. For example, Java Architecture for XML Binding (JAXB) provides a convenient way to bind an XML schema to a representation in Java code.

The methods or functions that will be needed for trajectory objects can be categorized as those needed only on the ground, those needed only in the air, and those needed both on the ground and in the air. In any case, the methods will need to:

- read and write the XML format and verify integrity using the MD5, SHA-1 [23], or similar algorithm.
- revise the XML document, given a full version and a partial (“delta”) revision.
- parse the XML format and convert the data into a corresponding object data structure.

- check the sequence of trajectory segments for validity, continuity, and flyability.
- compute the reference position and the bounding space as a function of time and current position.
- determine the along-track, cross-track, and vertical deviations of the current position from the reference position.
- generate and/or modify trajectories using a graphical user interface with user-friendly point-and-click functionality.
- determine the minimum horizontal separation between the bounding spaces for a pair of trajectories, and the minimum vertical separation while the horizontal separation is less than the prescribed standard.

The last item, although simple in concept, could be a challenge to implement efficiently. In principle it requires the determination of the minimum separation between any point in one (horizontal) bounding space and any point in the other at any point in time. Clearly the closest points will be on the boundary of the bounding spaces, but that still leaves a theoretically infinite number of possible points on each boundary that need to be checked against each of an infinite number of points on the other boundary. Efficient geometric algorithms will need to be developed to get a reasonably efficient and accurate approximation.

On the ground, more advanced methods will be needed. For example, methods will be needed to find a conflict-free trajectory “nearest” a requested trajectory in terms of pathlength, flight time, fuel consumption, or some other appropriate metric. Conflicts should be avoided with other aircraft as well as convective weather cells. In some cases, merely tightening the tolerances temporarily, without changing the reference trajectory, may resolve potential conflicts, and methods for doing that could be useful. Also useful would be methods for determining how much the error tolerances (particularly along-track) can be relaxed without causing a conflict in moderate or light traffic. Finally, methods for constructing efficient, conflict-free sets of trajectories could be useful. Existing multi-aircraft trajectory optimization methods can be adapted for this purpose.

Additionally, methods for detecting critical maneuvers and no-transgression zones would be useful. A critical maneuver is a planned maneuver (turn, climb, descent, leveloff, etc.) that must be executed on schedule to avoid an imminent conflict. No-transgression zones are areas that must be strictly avoided due to proximity to another aircraft. More generally, it will be useful to be able to determine and highlight trajectory “hotspots,” or points in a trajectory where separation approaches the minimum required separation, and conformance is critical to avoid a loss of separation. The situation is

analogous to driving a car on a two-lane road: the center line bounds the driving path, but the opposite lane becomes a no-transgression zone only when an oncoming car approaches.

As for the unique onboard requirements for trajectory objects, the main one is that the aircraft be capable of conforming to its assigned trajectory. This will most likely be an automated function of an advanced FMS. Another onboard requirement is to display the reference trajectory for the pilot, which could be done with an advanced CDTI (Cockpit Display of Traffic Information), perhaps as a head-up display. A CDTI could also conceivably be used for real-time guidance, instead of an FMS, to allow aircraft not equipped with an FMS to fly in high-density airspace for short periods of time for arrival at busy airports. This possibility will be discussed briefly in the next section.

TRAJECTORY DISPLAY

In twenty years, virtually all commercial transport aircraft are likely to be equipped with an advanced FMS capable of automatically flying a specified trajectory. Pilots will still need to be able to visualize their assigned trajectory and monitor conformance, however. New CNS technologies should be able to provide that capability at a reasonable cost. A technology called Cockpit Display of Traffic Information (CDTI) uses GPS/WAAS and ADS-B to provide an onboard visual representation of the local traffic environment. An advanced CDTI could also show assigned trajectories for improved situational awareness.

Such an advanced CDTI could also conceivably be used in place of an FMS for pilot guidance. Whereas an FMS must be developed for a particular aircraft model, generic CDTI units can be produced for any aircraft with a compatible slot in the cockpit, and are potentially cheaper to test and certify, hence a CDTI unit could potentially be cheaper than an FMS. This fact could allow lower-cost aircraft, perhaps even general aviation, to participate for short periods of time as equipped aircraft in high-density airspace, which could be critical in crowded terminal and transition airspace. The trajectory error tolerances for such aircraft would probably need to be larger than for aircraft equipped with an FMS, but they would still fly much more precise and predictable trajectories than aircraft not equipped with CDTI guidance.

Figure 11 shows how an advanced CDTI might be used for situational awareness. It shows the local traffic as any CDTI would, but it also shows the bounding space and the future assigned horizontal path of the user and the local traffic, so surprise maneuvers are minimized. The start of descent point is also indicated, and a vertical guidance bar appears on the right edge of the display so that all three axes are represented in one view. The small box in the upper-left corner shows current altitude,

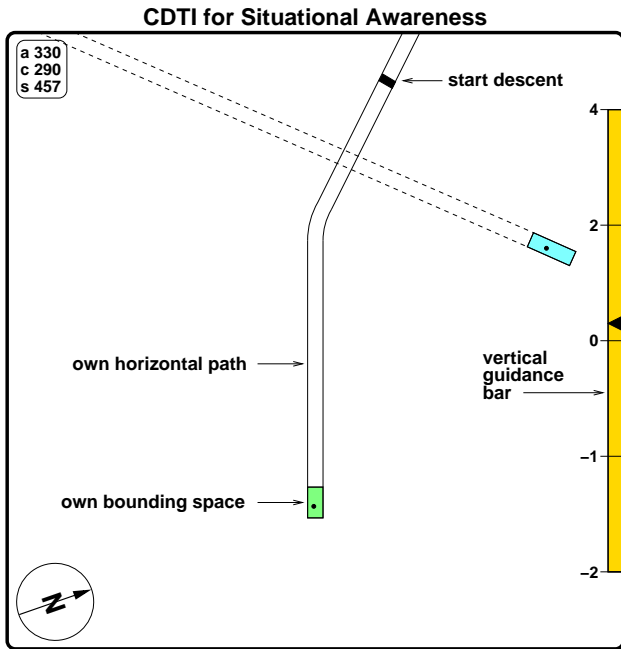


Figure 11: CDTI for situational awareness.

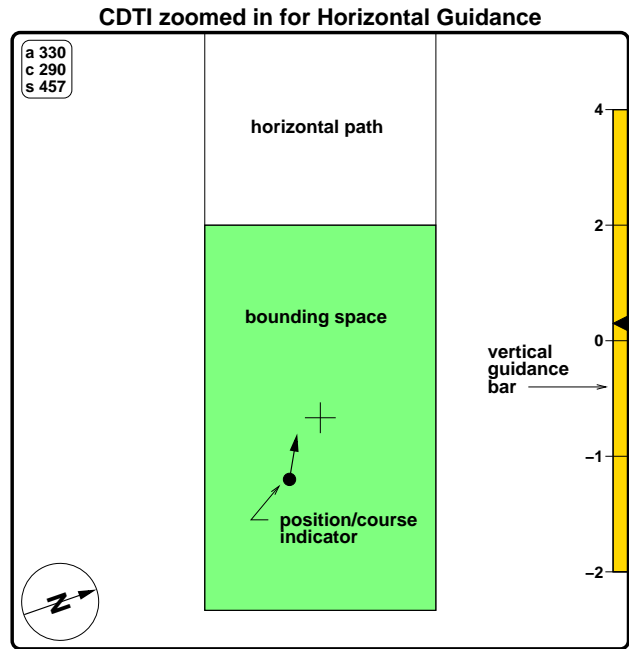


Figure 12: CDTI zoomed in for horizontal guidance.

course, and speed, and a compass pointer appears in the lower-left corner.

Figure 12 shows how the same CDTI could be zoomed in for finer guidance, again perhaps as a head-up display. The horizontal position of the user within the bounding space is clearly shown, as is the velocity relative to the reference velocity. Again, a vertical guidance bar appears on the right edge of the display, but more detailed vertical guidance could also be provided during climb and descent, as shown in figure 13. The vertical guidance display includes a cross-track guidance bar so that some guidance is always provided in all three axes.

CONCLUSION

An XML data format standard has been proposed for specifying trajectories for appropriately equipped aircraft in future high-capacity airspace. The format specifies a 4D reference trajectory along with error tolerances, which together define a bounding space, at each point in time, in which the aircraft is required to be contained. The assigned trajectories and tolerances will be constructed such that conformance by any two aircraft will guarantee the minimum required separation between them for a period of time known as the conflict time horizon, which could be something like 15 minutes.

Each AOC or pilot will be allowed to submit requests at any time for specific trajectories or trajectory revisions. Requested trajectories and revisions will be checked on the ground for conflicts within the conflict time horizon, and if they are free of conflicts and consistent with the longer range traffic flow plan, they will be

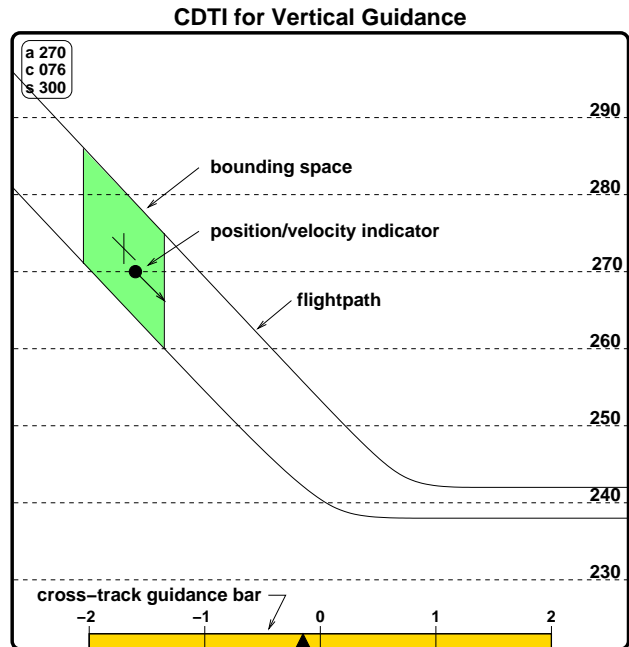


Figure 13: CDTI for vertical guidance.

approved. Otherwise, the requested trajectory will be minimally revised on the ground to resolve the conflict or other problem, then it will be uplinked as the assigned trajectory. Message digests such as MD5 or SHA-1 can be used to guarantee datalink integrity.

Trajectories are broken down into a series of trajectory segments of various types. With three vertical types (climb, level, and descent), three heading types (straight,

right turn, and left turn), and five speed types, the total number of possible combinations is 45. The horizontal path consists of a series of geodetic waypoints connected by great circles, and the great-circle segments are connected by turns of specified radius. Along-track position is specified as a low-order polynomial function of time, and vertical profiles for climb and descent are specified as low-order polynomial functions of *actual* (not reference) along-track position. Flight technical error tolerances in the along-track, cross-track, and vertical axes determine the bounding space. Periodic updates in the along-track axis adjust for errors in the predicted along-track winds.

This regimen of assigned 4D trajectories can eliminate the need for separation monitoring (of equipped aircraft) by human air traffic controllers. It can also guarantee that the equipped traffic will be able to fly free of conflicts for at least several minutes even if all ground systems and the entire communication infrastructure fail. This failsafe guarantee, along with the elimination of the human factor from the primary separation feedback loop, has the potential to greatly increase airspace capacity. That increase in capacity will be necessary within the next couple of decades if the U.S. air traffic system is to keep pace with the growing demand for air travel.

References

- [1] National Airspace System Operational Evolution Plan: A Foundation for Capacity Enhancement 2003-2013, Version 5.0, Executive Summary, Dec 2002. Available from <http://www.faa.gov/programs/oep>.
- [2] Krozel, J.; Peters, M.; Bilimoria, K.; Lee, C.; and Mitchell, J. S. B: System Performance Characteristics of Centralized and Decentralized Air Traffic Separation Strategies. 4th USA/Europe Air Traffic Management R&D Seminar, Santa Fe, NM, USA, Dec 3-7, 2001.
- [3] Andrews, J.: Capacity Benefits of the Automated Airspace Concept (AAC): A Preliminary Investigation. MIT Lincoln Laboratory Report No. 42PM-AATT-0014, Cambridge, MA, Aug. 2001.
- [4] Sorensen, J., et al.: Massive Point-to-Point (PTP) and On-Demand Air Transportation System Concept Description. Volume I, Version 1.1. TR03231-01, NASA Contract No. NAS2-02076, Seagull Technology, Inc., Campbell, California, January 15, 2003.
- [5] Rossow, V. J.: Use of Individual Flight Corridors to Avoid Vortex Wakes. *Journal of Aircraft*, vol. 40, no. 2, March-April 2003, pp. 225-231.
- [6] Arkind, K.: Maximum Capacity Terminal Area Operations in 2022. AIAA Aviation Technology, Integration, and Operations (ATIO) Forum, Denver, CO, Nov 17-19, 2003.
- [7] Erzberger, H.; and Paielli, R. A.: Concept for Next Generation Air Traffic Control System. *Air Traffic Control Quarterly*, vol. 10, no. 4, Winter 2002, pp. 355-378.
- [8] Erzberger, H.; and Pecsvaradi, T.: 4-D Guidance System Design with Application to STOL Air Traffic Control. (A72-38226 19-10), 13th Joint Automatic Control Conference, Stanford, CA, Aug 16-18, 1972, pp. 445-454.
- [9] Wilson, I.: PHARE Advanced Tools Project Final Report. DOC 98-70-18, Eurocontrol, 21 Nov 1999.
- [10] Concepts for Services Integrating Flight Operations and Air Traffic Management Using Addressed Data Link, RTCA DO-269, June 12, 2001.
- [11] Barrer, J. N.: Integrating the Cockpit with Air Traffic Management: The Concept of Path Objects. 3rd USA/Europe Air Traffic Management R&D Seminar, Napoli, Italy, June 13-16, 2000.
- [12] Wilson, I.: PHARE: Definition and Use of Tubes. DOC 96-70-18, Eurocontrol, 15 July, 1996.
- [13] Barhydt, R.; and Warren, A.: Development of an Information Structure for Reliable Communication of Airborne Intent and Aircraft Trajectory Prediction. *Air Traffic Control Quarterly*, vol. 11, no. 3, Fall 2003, pp. 251-274.
- [14] Advanced Flight Management Computer System, ARINC Characteristic 702A-1, prepared by Airlines Electronic Engineering Committee, Jan 31, 2000.
- [15] Minimum Aviation System Performance Standards for Automatic Dependent Surveillance Broadcast (ADS-B), RTCA DO-242A, RTCA, Inc., Washington, DC., 2002.
- [16] Reynolds, T. G.; and Hansman, R. J.: Investigating Conformance Monitoring Issues in Air Traffic Control Using Fault Detection Approaches. ICAT-2003-5, Intl. Center for Air Transportation, Massachusetts Institute of Technology, Nov. 2003.
- [17] Minimum Aviation System Performance Standards: Required Navigation Performance for Area Navigation, RTCA DO-236A, Sept. 13, 2000.
- [18] See <http://XML.org> for general information on XML. See <http://www.w3schools.com/xml> for a tutorial on XML.
- [19] Denery, D. G.; and Erzberger, H.: The Center-TRACON Automation System: Simulation and Field Testing. NASA/TM—1995-110366. See also <http://www.ctas.arc.nasa.gov/>
- [20] Slattery, R.; and Zhao, Y.: Trajectory Synthesis for Air Traffic Automation. *J. Guidance, Control, and Dynamics*, vol. 20, no. 2, March-April, 1997, pp. 232-238.
- [21] Benjamin, S. G.; J. M. Brown; K. J. Brundage; D. Devenyi; G. A. Grell; D. Kim; B. E. Schwartz; T. G. Smirnova; T. L. Smith; S. Weygandt; and G. S. Manikin: RUC20 - The 20-km version of the Rapid Update Cycle. NOAA Technical Memorandum OAR FSL-28, 2002 (available from <http://maps.fsl.noaa.gov>).

- [22] Lindsay, K. S.; Green, S. M.; Mondoloni, S.; and Paglione, M.: Common Trajectory Modeling for National Airspace System Decision Support Tools. MP 02W0000115, The MITRE Corporation, McLean, VA, 2003 (submitted to *Air Traffic Control Quarterly*).
- [23] See <http://www.faqs.org/rfcs/rfc1321.html> for information on MD5. See <http://www.itl.nist.gov/fipspubs/fip180-1.htm> for information on SHA-1.
- [24] Paielli, R. A.: A Linear Altitude Rule for Safer and More Efficient Enroute Air Traffic. *Air Traffic Control Quarterly*, vol. 8, no. 3, Fall 2000, pp. 195-221.
- [25] Cole, R. E.; Green, S.; Jardin, M.; Schwartz, B. E.; and Benjamin, S. G.: Wind Prediction Accuracy for Air Traffic Management Decision Support Tools. *3rd USA/Europe Air Traffic Management R&D Seminar*, Napoli, Italy, 13-16 June 2000.
- [26] Cole, R. E.; Richard, C.; Kim, S.; and Bailey, D.: An Assessment of the 60 km Rapid Update Cycle (RUC) with Near Real-Time Aircraft Reports. NASA/A-1, Lincoln Laboratory, 15 July 1998.
- [27] Mueller, K. T.; Schleicher, D. R.; and Bilimoria, K. D.: Conflict Detection and Resolution with Traffic Flow Constraints. *AIAA Guidance, Navigation, and Control Conf.*, Monterey, California, Aug 5-8, 2002.

APPENDIX: GREATCIRCLE FORMULAS

The earth is nearly but not quite spherical, with an eccentricity of about one part in 300. Greatcircle algorithms come in two forms: those based on a simplified spherical model of the earth, and those based on a more accurate but more complex ellipsoidal model. The spherical model yields closed-form analytic solutions, whereas the ellipsoidal model yields more accurate but more complicated iterative algorithms. The ellipsoidal greatcircle algorithms will be needed for accurate along-track distances but are too complicated to be presented in this paper. A few of the key spherical equations are presented here for reference. The geodetic coordinates of a point will be represented as (ϕ, λ) , where ϕ is latitude and λ is longitude.

Given a point A on the surface of the earth at (ϕ_A, λ_A) , and a greatcircle from point A with initial course (groundtrack) angle χ_A , the point a distance d (in radians) along the greatcircle at (ϕ, λ) is given by

$$\phi = \text{asin}(\sin \phi_A \cos d + \cos \phi_A \sin d \cos \chi_A) \quad (1)$$

$$\lambda = \text{mod}(\lambda_A - \gamma + \pi, 2\pi) - \pi \quad (2)$$

where

$$\gamma = \text{atan2}(\sin \chi_A \sin d \cos \phi_A, \cos d - \sin \phi_A \sin \phi) \quad (3)$$

Given points A and B on the surface of the earth at (ϕ_A, λ_A) and (ϕ_B, λ_B) , the angular distance between them is given by

$$\text{dist}(A, B) = 2 \text{asin}(\gamma_1^2 + \cos \phi_A \cos \phi_B \gamma_2^2) \quad (4)$$

where

$$\gamma_1 \equiv \sin((\phi_B - \phi_A)/2)$$

$$\gamma_2 \equiv \sin((\lambda_A - \lambda_B)/2)$$

The result must be multiplied by the radius of the earth to determine the actual length of the greatcircle. The official FAA earth radius is 3440.655273 nmi.

Given a greatcircle that starts at point A at (ϕ_A, λ_A) and ends at point B at (ϕ_B, λ_B) , the initial course (groundtrack) angle of the greatcircle at point A is given by

$$\text{course}(A, B) = \text{atan2}(\beta_1, \beta_2) \quad (5)$$

where

$$\beta_1 \equiv \sin(\lambda_A - \lambda_B) \cos \phi_B$$

$$\beta_2 \equiv \cos \phi_A \sin \phi_B - \sin \phi_A \cos \phi_B \cos(\lambda_A - \lambda_B)$$

Note that the course angle at the *end* of the greatcircle can be determined by simply reversing the order of the arguments of the course function, which is useful for determining the turn angle between consecutive greatcircle segments.

Given these utility functions, the cross-track error (positive to the right) of a point C that was supposed to be on the greatcircle from A to B is given by

$$\begin{aligned} \text{cross}(A, B, C) = & \text{asin}(\sin(A, C) \\ & \sin(\text{course}(A, C) - \text{course}(A, B))) \end{aligned} \quad (6)$$

where

$$\sin(A, C) \equiv \sin(\text{dist}(A, C))$$

The along-track position is given by

$$\begin{aligned} \text{along}(A, B, C) = & \text{asin}(((\sin(A, C))^2 - \\ & (\sin \text{cross}(A, B, C))^2)^{1/2} / \cos \text{course}(A, B, C)) \end{aligned} \quad (7)$$

As before, these results must be multiplied by the radius of the earth to convert them from angles (radians) to lengths.

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 0704-0188	
<p>The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.</p> <p>PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.</p>					
1. REPORT DATE (DD-MM-YYYY) 22-09-2004		2. REPORT TYPE Technical Memorandum		3. DATES COVERED (From - To)	
4. TITLE AND SUBTITLE Trajectory Specification for High-Capacity Air Traffic Control				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) Russell A. Paielli				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER 150-10-01	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Ames Research Center Moffett Field, California 94035-1000				8. PERFORMING ORGANIZATION REPORT NUMBER A-0412440	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration Washington, D. C. 20546-0001				10. SPONSORING/MONITOR'S ACRONYM(S) NASA	
				11. SPONSORING/MONITORING REPORT NUMBER NASA/TM—2004-212825	
12. DISTRIBUTION/AVAILABILITY STATEMENT Unclassified — Unlimited Distribution: Nonstandard Subject Category 03, 04 Availability: NASA CASI (301) 621-0390					
13. SUPPLEMENTARY NOTES Point of Contact: Russell A. Paielli, Ames Research Center, M.S. 210-10, Moffett Field, CA 94035-1000					
14. ABSTRACT In the current air traffic management system, the fundamental limitation on airspace capacity is the cognitive ability of human air traffic controllers to maintain safe separation with high reliability. The doubling or tripling of airspace capacity that will be needed over the next couple of decades will require that tactical separation be at least partially automated. Standardized conflict-free four-dimensional trajectory assignment will be needed to accomplish that objective. A trajectory specification format based on the Extensible Markup Language is proposed for that purpose. This format can be used to downlink a trajectory request, which can then be checked on the ground for conflicts and approved or modified, if necessary, then uplinked as the assigned trajectory. The horizontal path is specified as a series of geodetic waypoints connected by great circles, and the great-circle segments are connected by turns of specified radius. Vertical profiles for climb and descent are specified as low-order polynomial functions of along-track position, which is itself specified as a function of time. Flight technical error tolerances in the along-track, cross-track, and vertical axes define a bounding space around the reference trajectory, and conformance will guarantee the required separation for a period of time known as the conflict time horizon. An important safety benefit of this regimen is that the traffic will be able to fly free of conflicts for at least several minutes even if all ground systems and the entire communication infrastructure fail. Periodic updates in the along-track axis will adjust for errors in the predicted along-track winds.					
15. SUBJECT TERMS Air traffic management, Air traffic control, Trajectories, XML, Datalink, Flight management systems					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19b. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Russell A. Paielli
Unclassified	Unclassified	Unclassified	Unclassified	30	19b. TELEPHONE NUMBER (Include area code) (650) 604-5454