Conflict Probability Estimation Generalized to Non-Level Flight

Russell A. Paielli and Heinz Erzberger
NASA Ames Research Center

Abstract: A method known as conflict probability estimation (CPE) is presented to estimate the probability of conflict for pairs of aircraft with uncertain predicted trajectories. It is a generalization of a previous method that applied to level flight only. The generalized method is equivalent to the previous exact analytical solution for the special case of level flight, but it also gives an approximate analytical solution for non-level flight. The trajectory prediction errors are modeled as Gaussian, and the two error covariances for an aircraft pair are combined into a single, equivalent covariance of the relative position. A coordinate transformation is then used to derive the analytical solution. A Monte Carlo simulation demonstrates that the CPE algorithm is accurate, assuming the validity of the simplifying assumptions and the accuracy of the underlying prediction error model. Numerical examples are presented.

Introduction

The economics and efficiency of air transportation in the continental U.S. will improve significantly when the current routing restrictions are relaxed to allow more direct or wind-optimal trajectories. The current system of static jet routes imposes structure on the en-route airspace, which helps to maintain an orderly flow of traffic, but it also forces aircraft to fly inefficient, zig-zag routes. Fortunately, new air traffic management (ATM) systems are being developed that will allow safety to be maintained with a less constrained airspace structure. The improved routing efficiency could save the airline industry several billion dollars per year. The methods presented in this paper apply to the current system, but they will be even more useful in a future, less constrained, ATM system.

The safety and efficiency of ATM can benefit now, and even more so in the future, from automated conflict prediction and resolution advisories. Conflict prediction is based on inexact trajectory prediction, however, and is itself therefore inexact. The farther in advance a prediction is made, the less certain it is, particularly in the along-track direction. For better efficiency, aircraft are usually flown at constant airspeed or Mach number rather than constant ground-speed, and the uncompensated effects of wind modeling and prediction errors accumulate with time. A method is needed, therefore, to estimate the probability of conflict, where a conflict is defined as two or more aircraft coming within the minimum allowed separation distance of each other. The minimum allowed horizontal separation for en-route airspace is currently 5 nmi. The vertical separation requirement above 29,000 ft altitude (FL290) is currently 2000 ft; below FL290 it is 1000 ft.

Conflict resolution can be broadly classified as strategic or tactical, depending on how far in advance the resolution maneuver is executed. Early conflict resolution is considered strategic, and later resolution is considered tactical, where the boundary between early and late is approximately six to eight minutes before the potential conflict. Strategic conflict resolution is concerned primarily with efficiency, whereas tactical conflict resolution is more concerned with safety. The optimal time to initiate a conflict resolution maneuver is a trade-off between efficiency and certainty. The farther in advance a maneuver is initiated, the more efficient it is likely to be in terms of time and/or fuel, but the less certain will be exactly what maneuver is required or whether a maneuver is required at all. The later a maneuver is initiated, on the other hand, the more certain will be exactly what maneuver is required, but the less efficient and less subtle the maneuver is likely to be. The determination of the optimal time to initiate a maneuver, therefore, requires an estimate of conflict probability.

For efficient strategic conflict resolution, the determination of the maneuver itself also requires a method of estimating conflict probability, because the objective is to reduce the conflict probability to some acceptable level. Strategic conflict resolution cannot reduce the conflict probability to zero without introducing unnecessary inefficiency, but it does not need to do so because it is not the last line of defense. Human air traffic controllers still have, and will continue to have, responsibility for aircraft separation, and the Conflict Alert system gives them an automatic three-minute warning of projected conflicts. As an additional safety measure, the radar-based Traffic alert and Collision Avoidance System (TCAS) onboard aircraft warns pilots of potential conflicts so they can execute emergency collision-avoidance maneuvers when necessary.

A method known as conflict probability estimation (CPE) is developed in this paper to estimate the probabil-

[0]This paper was published in the Air Traffic Control Quarterly, vol. 7, no. 3, October, 1999. Russ Paielli can be reached at NASA Ames Research Center, 210-10, Moffett Field, CA 94035-1000 or by email at rpaielli@mail.arc.nasa.gov

ity of conflict for pairs of aircraft with uncertain predicted trajectories. It is a generalization to non-level flight of the CPE method presented in [1] for level flight, and it supercedes that method. To make this paper complete, some steps of the earlier method must be essentially repeated. The trajectory prediction errors are modeled as Gaussian (i.e., normally distributed), and the two error covariances for an aircraft pair are combined into a single, equivalent covariance of the relative position, as before. A coordinate transformation is then used to derive an analytical solution. That solution is exact for level flight, given certain assumptions, and is approximate for non-level flight.

The CPE algorithm has been programmed in C++ as an independent module that can be used in any ATM conflict detection and resolution system. It has also been integrated into the Center-TRACON Automation System (CTAS) [2, 7], an ATM decision support system that has been developed at NASA Ames and installed for testing at two of the Air Route Traffic Control Centers (ARTCC) operated by the Federal Aviation Administration (FAA). In CTAS, the conflict probability is not displayed directly to air traffic controllers, but it is used to determine when to alert controllers of a potential conflict and what color code to use for the alert. The main focus of this paper is theoretical rather than operational, but some of the operational considerations are discussed in [3]. As mentioned earlier, CPE could also eventually be used in CTAS to help resolve conflicts efficiently, and that potential application motivates the current work. However, both the theoretical and operational aspects of applying CPE to conflict resolution are beyond the scope of this paper.

Other CPE algorithms have also appeared in the literature. Warren presented a CPE algorithm similar to the one presented here, but it was relegated to the appendix of a report [5], with an incomplete derivation and no validation. Krozel et al. [6] derived a CPE algorithm based on a first-order Taylor series expansion, but they did not present any validation, and their brief discussion of the stochastic error model was solely in terms of navigational accuracy, with no consideration of the larger effects of wind modeling error. Another approach was taken by Yang and Kuchar [4], who used a Monte Carlo simulation to estimate conflict probability. This approach is non-parametric and has the advantage of not requiring any particular error distribution or trajectory constraints, but it requires approximately three orders of magnitude more computation than the algorithm presented here requires, and its feasibility in an operational ATM system is questionable.

Conflict Probability Estimation

Aircraft trajectory prediction is a surprisingly complex modeling and software problem that has been addressed in CTAS. It requires current estimated position and velocity, flight plan, and predicted winds aloft. It is inexact, primarily because of wind modeling and prediction error and secondarily because of tracking, navigation and control error. The positions and velocities are currently based on radar tracking, and are provided, along with the flight plans, by the FAA at their ARTCC facilities. The wind predictions are provided by the Rapid Update Cycle (RUC) [10], a weather prediction system operated by the National Center for Environmental Prediction (NCEP) for the National Oceanic and Atmospheric Administration (NOAA). The algorithm to be presented in this paper requires predictions of position and velocity for pairs of aircraft at their point of minimum separation. Those predictions are provided by the CTAS Trajectory Synthesizer [8].

The CPE algorithm also requires a statistical model of the distribution of errors on the predicted positions at the point of minimum separation. Although any form of distribution can be used in principle, the Gaussian (normal) distribution simplifies the derivation of the algorithm and provides the potential for very efficient analytical forms of computation. Many other estimation algorithms, such as the classical Kalman filter, are also based on the Gaussian model. Recent empirical studies [9] have validated the Gaussian model of the trajectory prediction errors. For purposes of this paper, the magnitudes of the prediction error covariances and their growth rates are merely parameters that can be estimated empirically.

The Gaussian prediction errors can be represented as ellipses in the horizontal plane or as ellipsoids in three-dimensional space. Those error ellipsoids tend to have their principal axes in the along-track, cross-track, and vertical directions. Figure 1 shows a typical example of prediction error ellipses in the horizontal plane, which tend to grow in the along-track direction with time of prediction. Because aircraft are usually flown at constant airspeed or Mach number for better efficiency, the uncompensated effects of wind modeling and prediction errors accumulate with time in the along-track direction. Position feedback in the cross-track and vertical axes keeps the prediction errors from growing in those directions. (The uncertainty ellipsoid for a Gaussian random variable x is defined as the solution of $z^T Z^{-1} z = c^2$, where $z \equiv x - E(x)$, $Z \equiv \mathrm{cov}(z) \equiv E(zz^T)$, E is the expected value, and c is a constant that can be assumed to be unity unless otherwise noted.)

The remainder of this section is divided into four subsections. First, the method of combining two covariances into a single covariance of the relative position prediction error is discussed. Next, coordinate transformations are derived that normalize the combined covariance and rotate the coordinate system to simplify the probability calculation. Then the analytical solution is developed. Finally, some heuristics and caveats are discussed.

Combined Error Covariance

The combining of error covariances is essentially the same as in the earlier CPE procedure for level flight only, except

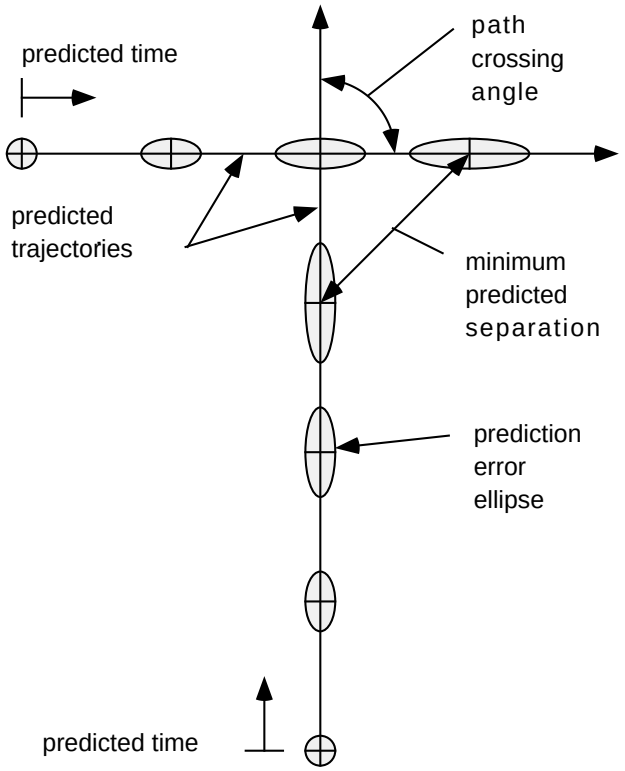

Figure 1: Encounter geometry in the horizontal plane

that now it applies in three dimensions rather than two. In vector/matrix notation, that difference is mathematically implicit. Hence, this section is very similar to the corresponding section in the earlier paper [1], but it is necessary to make the paper complete.

The trajectory prediction error for an aircraft will be modeled as Gaussian with zero mean and a covariance that has eigenvectors in the along-track, cross-track, and vertical directions, as explained previously. The covariance matrix is therefore diagonal in a coordinate system aligned with the aircraft heading and the vertical. If q is the aircraft position in such a coordinate system, and $\bar{q}$ is the corresponding prediction, then the prediction error is $\tilde{q} \equiv q - \bar{q}$ and the corresponding diagonal covariance matrix is

$$S \equiv \mathrm{cov}(\tilde{q}) \tag{1}$$

If R is the rotation matrix that transforms the heading-aligned coordinates to the reference coordinates, then the position prediction error in the reference coordinate system is $\tilde{p} = R\tilde{q}$ and the corresponding covariance matrix is

$$Q \equiv \mathrm{cov}(\tilde{p}) = RSR^T \tag{2}$$

Unlike S, the covariance matrix Q is not diagonal, but it is block-diagonal because the vertical error is still decoupled from the horizontal error.

Because the trajectory prediction errors are modeled as Gaussian, the two error covariances for an aircraft pair can be easily combined into a single equivalent covariance of the position difference or the relative position of one aircraft with respect to the other. For present purposes, this combined covariance can be assigned to one of the aircraft, referred to as the "stochastic" aircraft, and the other aircraft, referred to as the "reference" aircraft, can be regarded as having no position uncertainty.

Subscripts S and R will be used to designate the stochastic and reference aircraft, respectively. If the position difference in the reference coordinate system is Δp and the corresponding prediction is $\Delta\bar{p}$, then the prediction error is $\Delta\tilde{p} \equiv \Delta p - \Delta\bar{p} = \tilde{p}_S - \tilde{p}_R$. The combined prediction error covariance is

$$C \equiv \mathrm{cov}(\Delta\tilde{p}) = Q_S + Q_R - Q_{SR} \tag{3}$$

where Q_S and Q_R are the individual covariances based on (2), and the cross-correlation term Q_{SR} is defined as

$$Q_{SR} \equiv E(\tilde{p}_S\tilde{p}_R^T + \tilde{p}_R\tilde{p}_S^T) \tag{4}$$

In general, the combined error ellipsoid corresponding to C does not have principal axes aligned with the along-track and cross-track directions of either aircraft, but the vertical axis is still a principal axis. The vertical error is therefore still decoupled from the horizontal error.

The cross-correlation of prediction errors between aircraft, which is represented by Q_{SR}, can be important for aircraft pairs that are close together and heading in similar directions. The cross-correlation is more difficult to model than the individual covariances because it requires a spatial wind-error correlation model. That correlation model will be a function of both separation distance and heading angular difference. The correlation is determined empirically in [11] as a function of separation distance, but not as a function of heading angular difference. Aircraft pairs with nearly perpendicular flight paths tend to have weakly cross-correlated prediction errors because their along-track positions are affected by different wind components. Aircraft pairs that are close together and heading in similar directions, on the other hand, tend to have more strongly cross-correlated prediction errors, both because they are affected by a common wind component and because they spend a relatively long time close together. The portion of those errors in common will cancel in the position difference, reducing the relative position error.

It will be assumed that the aircraft velocities and prediction errors can be approximated as constant during the encounter (period of potential conflict). It will also be assumed that the velocities during the encounter are known. These assumptions are reasonable for aircraft in steady cruise. Aircraft in climb or descent usually have changing speed, but the change is usually small enough during the encounter that the value at the predicted point of minimum separation can be used. Note that *planned* maneuvers of any kind *prior to* the encounter do *not* violate these assumptions (assuming, of course, that the trajectory synthesizer knows about them). However, such maneuvers tend to increase the uncertainty of the predicted trajectory, and that uncertainty is reflected in the covariance.

The combined error ellipsoid is centered on the stochastic aircraft, and the conflict zone is centered on the reference aircraft. The conflict zone is a vertical cylinder 10 nmi in diameter and 4000 ft high (or 2000 ft high for altitudes under FL290). The extended conflict zone is defined as the volume swept out by the conflict zone as the encounter evolves. It is the projection of the conflict zone along the line of relative velocity. Because the relative velocity is assumed to be constant, the extended conflict zone is a straight tube. The total conflict probability is equal to the portion of the probability mass density within the extended conflict zone, where the total mass is unity. The coordinate transformations to be presented in the next section allow this probability to be determined analytically for level flight or approximated analytically for non-level flight.

Coordinate Transformations

The conflict probability is difficult or impossible to determine analytically in the original coordinate system. It can be determined numerically, but a numerical solution is likely to be much less efficient than an analytical solution. Such inefficiency is undesirable for an algorithm that is intended to run in an operational system at a high rate. Fortunately, coordinate transformations have been found that allow an exact analytical solution for the case of level flight and a good analytical approximation for the case of non-level flight.

The coordinate transformations consist of a non-orthogonal transformation followed by an orthogonal transformation or rotation of the coordinate system. The non-orthogonal transformation transforms the combined error ellipsoid into the standard form of a unit sphere. The orthogonal transformation aligns the relative velocity with one of the coordinate axes. These coordinate transformations must be applied to the relative position and velocity, the combined error ellipsoid, and the cylindrical conflict zone.

Non-Orthogonal Transformation

A non-orthogonal transformation distorts the shapes of the error ellipsoid and the cylindrical conflict zone. The purpose of this transformation is to transform the error ellipsoid into the standard form of a unit sphere to simplify the probability computations. In this form, the corresponding three-dimensional probability density function decouples into the product of three identical scalar functions.

A general linear coordinate transformation of the relative position and its error $\Delta\tilde{p}$ is of the form

$$\Delta\tilde{p}_T = T\Delta\tilde{p} \tag{5}$$

where T is a transformation matrix yet to be determined. In the transformed coordinate system, the mean error is still zero and the combined error covariance is

$$\mathrm{cov}(\Delta\tilde{p}_T) = TCT^T \tag{6}$$

where C is the combined error covariance in the original coordinate system from (3).

A Cholesky decomposition [12] or "square-root" factorization of the combined error covariance C is of the form

$$C = LL^T \tag{7}$$

where L is lower triangular. By choosing

$$T = L^{-1} \tag{8}$$

(6) can be simplified to the standard form

$$\mathrm{cov}(\Delta\tilde{p}_T) = I \tag{9}$$

where I is the identity matrix. This transformation therefore results in the transformation of the combined error ellipsoid into the standard form of a unit sphere.

Because the vertical error is decoupled from the horizontal error, the combined error covariance matrix is a block-diagonal matrix of the form

$$C \equiv \begin{bmatrix} C_h & 0 \\ 0 & \sigma_z^2 \end{bmatrix} \tag{10}$$

where the horizontal covariance is

$$C_h \equiv \begin{bmatrix} \sigma_x^2 & \rho_{xy} \\ \rho_{xy} & \sigma_y^2 \end{bmatrix} \tag{11}$$

The Cholesky factor of C in (7) is found to be

$$L \equiv \begin{bmatrix} L_h & 0 \\ 0 & \sigma_z \end{bmatrix} \tag{12}$$

with

$$L_h = \begin{bmatrix} \sigma_x & 0 \\ l_{xy} & l_y \end{bmatrix} \tag{13}$$

where $l_{xy} \equiv \rho_{xy}/\sigma_x$ and $l_y \equiv (\sigma_y^2 - \rho_{xy}^2/\sigma_x^2)^{1/2}$. The transformation matrix is then of the form

$$T = L^{-1} \equiv \begin{bmatrix} T_h & 0 \\ 0 & 1/\sigma_z \end{bmatrix} \tag{14}$$

where

$$T_h = L_h^{-1} = \begin{bmatrix} 1/\sigma_x & 0 \\ t_{xy} & 1/l_y \end{bmatrix} \tag{15}$$

with $t_{xy} \equiv -(\rho_{xy}/\sigma_x)/l_y$.

The transformation T transforms the combined error ellipsoid into a unit sphere. The same transformation must also be applied consistently to the relative velocity and to the cylindrical conflict zone. The transformation of the relative velocity is of the same form as the transformation of the relative position (a simple matrix multiplication). The transformation of the conflict zone is more complicated, however, and will now be discussed.

The conflict zone around each aircraft is a vertical right circular cylinder of radius R and half-height H. For the legal separation requirements above FL290, the dimensions of the cylinder are $R = 5$ nmi and $H = 2000$ ft. The

position p of points on the horizontal circles at any cross-section of the conflict zone satisfy the equation $p^T E p = 1$, where

$$E \equiv \begin{bmatrix} 1/R^2 & 0 \\ 0 & 1/R^2 \end{bmatrix} \tag{16}$$

The matrix E specifies the horizontal cross-section of the conflict zone, which is circular.

The transformation of the horizontal position p is of the form $p_T \equiv T_h p$ and the inverse transformation is of the form $p = L_h p_T$. After substituting the inverse transformation into the equation for the circles, the new equation in the transformed coordinates becomes $p_T^T E_T p_T = 1$, where

$$E_T \equiv L_h^T E L_h \tag{17}$$

The matrix E_T specifies the horizontal cross-section of the transformed conflict zone, which is an ellipse. Hence the conflict zone has been transformed from a vertical right circular cylinder into a vertical right elliptical cylinder. The vertical component of the transformation is decoupled from the horizontal component and is simply $z_T = z/\sigma_z$.

As mentioned above, the transformation of the error ellipsoid into a unit sphere simplifies the probability computations considerably because the corresponding three-dimensional probability density function decouples into the product of three identical scalar functions: $p(x,y,z) = p(x)p(y)p(z)$, where $p(s) \equiv \exp(-s^2/2)/\sqrt{2\pi}$. The probability density function can be represented as a radially symmetric volume of variable mass density, where the total mass is unity. The density is largest in the center and decreases in any outward radial direction according to a Gaussian "bell curve".

Orthogonal Transformation

In the transformed coordinate system, the extended conflict zone still extends in the direction of the (transformed) relative velocity. To simplify the probability computation, an orthogonal transformation will be used to rotate the transformed coordinate system such that the relative velocity is in the direction of one of the coordinate axes. An orthogonal transformation is a rotation of the coordinate system and will therefore not further distort the error sphere or the conflict zone.

The orthogonal transformation consists of two separate rotations. The first rotation is about the vertical axis to align one of the vertical coordinate planes with the relative velocity. The second rotation is about an axis normal to that coordinate plane to align one of the coordinate axes with the relative velocity. The second rotation is unnecessary if both aircraft are in level flight throughout the encounter.

If the relative velocity in the original coordinate system is Δv, the (non-orthogonally) transformed relative velocity is

$$\Delta v_T \equiv T \Delta v \equiv \begin{bmatrix} \nu_x & \nu_y & \nu_z \end{bmatrix}^T \tag{18}$$

where T is defined in (14). After the first rotation of coordinates about the vertical z axis, the relative velocity is

$$\Delta v_r \equiv R_z \Delta v_T \equiv \begin{bmatrix} \nu_h & 0 & \nu_z \end{bmatrix}^T \tag{19}$$

where

$$R_z \equiv \begin{bmatrix} R_h & 0 \\ 0 & 1 \end{bmatrix} \tag{20}$$

with

$$R_h \equiv \frac{1}{\nu_h} \begin{bmatrix} \nu_x & \nu_y \\ -\nu_y & \nu_x \end{bmatrix} \tag{21}$$

and $\nu_h \equiv (\nu_x^2 + \nu_y^2)^{1/2}$. The subscript r designates the coordinate system after this first rotation of coordinates.

The horizontal cross-section of the conflict zone is now an ellipse specified by the matrix

$$E_r \equiv R_h E_T R_h^T \equiv \begin{bmatrix} s_x & s_{xy} \\ s_{xy} & s_y \end{bmatrix} \tag{22}$$

where E_T is defined in (17). The transformed conflict zone is still a vertical right elliptical cylinder. After differentiating the quadratic form by each of the independent variables x_r and y_r, then setting the derivatives to zero, the extremal value of y_r on the ellipse is found to be

$$y_{max} = \sqrt{s_x / \det} \tag{23}$$

where

$$\det \equiv s_x s_y - s_{xy}^2 \tag{24}$$

These results will be used in the next subsection.

If either aircraft is in non-level flight during the encounter, a second rotation of coordinates about the y_r axis is necessary to align the x_r axis with the relative velocity. After this rotation, the relative velocity is

$$\Delta v_R \equiv R \Delta v_T \equiv \begin{bmatrix} \nu_t & 0 & 0 \end{bmatrix}^T \tag{25}$$

where the subscript R denotes the final rotated and transformed coordinate system, and where $R \equiv R_y R_z$, with

$$R_y \equiv \frac{1}{\nu_t} \begin{bmatrix} \nu_h & 0 & \nu_z \\ 0 & \nu_t & 0 \\ -\nu_z & 0 & \nu_h \end{bmatrix} \tag{26}$$

and $\nu_t \equiv (\nu_h^2 + \nu_z^2)^{1/2}$. The relative velocity is therefore aligned with the x_R axis. In this coordinate system, the conflict zone is still a right elliptical cylinder, but it is "tilted" about the y_r axis and is therefore not aligned with the z_r axis.

An interactive Matlab program [13] has been developed to visualize transformed encounter geometries. It allows the user to enter a path angle, a prediction time, and a predicted minimum separation, among other parameters. It was used to generate Figures 2 and 3, which illustrate examples of a level encounter and a non-level (level/descent) encounter, respectively. In each case, the path crossing angle is 45 deg, the prediction time is 10 min, and the predicted minimum horizontal separation is 2 nmi. For the level case, the nominal vertical separation is a constant 1000 ft; for the non-level case it is 1000 ft at the point

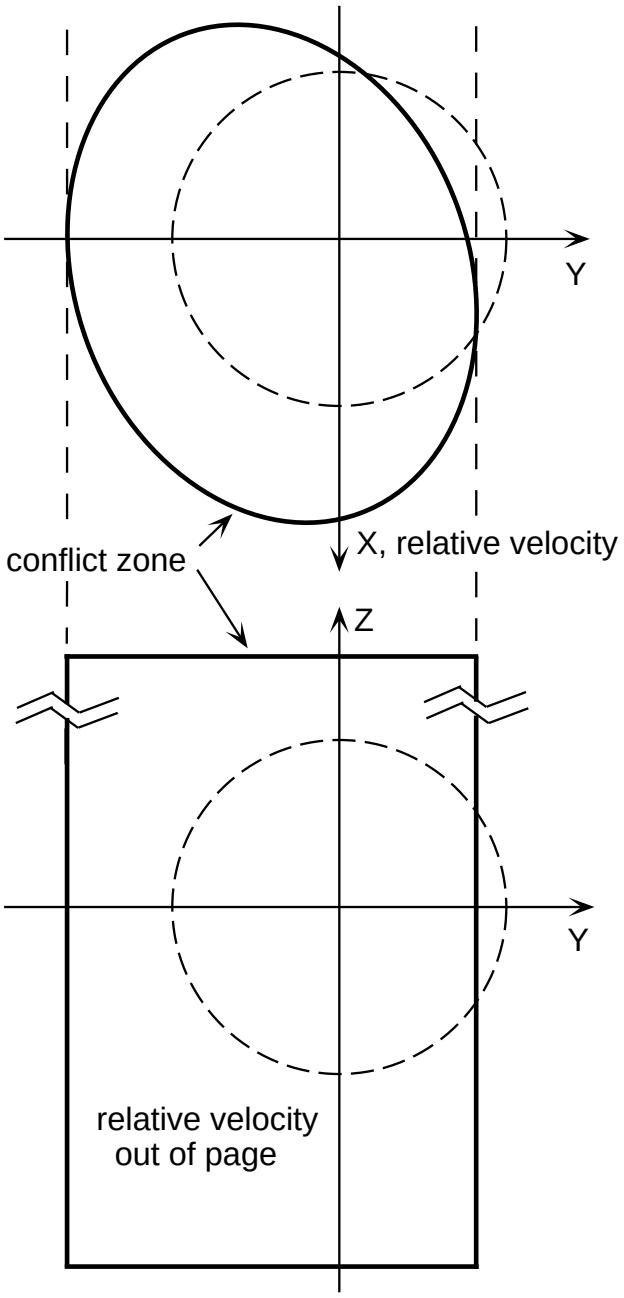

Figure 2: Transformed encounter geometry for a level encounter (dashed circles represent unit error sphere)

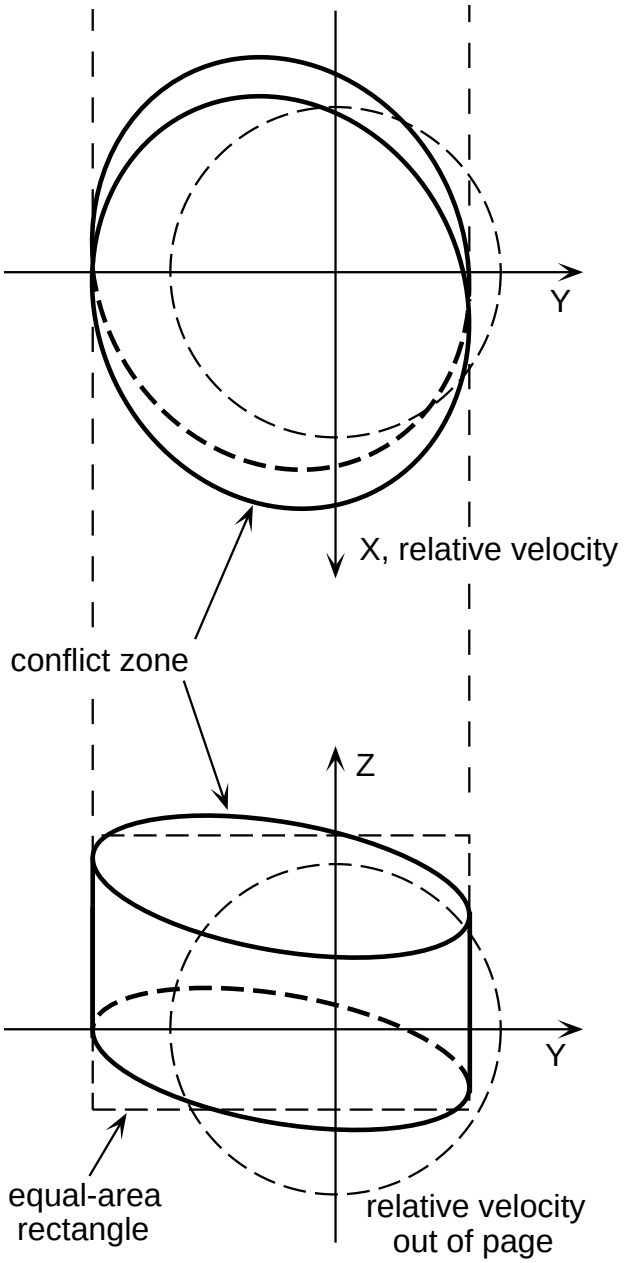

Figure 3: Transformed encounter geometry for a non-level (level/descent) encounter (dashed circles represent unit error sphere)

of minimum horizontal separation, with a descent rate of 1000 ft/min. The unit circles represent the error sphere, of course. (For simplicity, the axes in Figures 2 and 3 are labeled simply X, Y, and Z rather than x_R, y_R, and z_R.)

In Figure 2, because both aircraft are in level flight, the relative velocity is horizontal and the transformed conflict zone is a vertical right elliptical cylinder. When viewed from the direction of the relative velocity (bottom portion of Figure 2), the conflict zone appears as a rectangle, and when viewed from above (top portion of Figure 2) it appears as an ellipse. In level flight, the vertical rms error tends to be much smaller than the horizontal rms error, hence the coordinate transformation tends to drastically stretch the vertical axis relative to the horizontal axes. This distortion makes the conflict zone appear very tall and narrow, with a typical height of 5 to 20 times its diameter, but this effect is suppressed in Figure 2 to maintain a reasonable scale.

In Figure 3, because one of the aircraft is descending, the relative velocity is no longer horizontal, and the transformed cylindrical conflict zone is no longer perpendicular to the relative velocity. When viewed from the direction of the relative velocity (bottom portion of Figure 3), the conflict zone appears as an oblique view of a hockey puck. Because it is tilted about the y_R axis, however, its sides still appear vertical when viewed from the direction of the relative velocity along the x_R axis. When viewed from above (top portion of Figure 3) the top and bottom of the conflict zone appear as an offset pair of ellipses. In descent, the vertical rms error is modeled as growing at 300 ft/min, hence at a prediction time of 10 min it is 3000 ft. This magnitude of error is much larger than it is in level flight, so although the coordinate transformation still stretches the vertical axis relative to the horizontal axes, it does not do so nearly as much as it does in level flight. The same distortion also increases the vertical component of velocity and, hence, the descent angle. That is why the conflict zone is tilted significantly more than the actual descent angle, which is typically around 3 deg.

Analytical Solution

The extended conflict zone is the volume swept out by the conflict zone as the encounter progresses. Its cross-section is the projection of the cylindrical conflict zone onto the plane normal to the relative velocity. For the case of level flight, that cross-section is rectangular, as shown in Figure 2. The conflict probability therefore decouples into the product of vertical and horizontal conflict probabilities and can be determined exactly. For level flight, the solution to be presented in this paper gives the same result as the algorithm presented previously [1].

For non-level flight, on the other hand, the projection of the cylindrical conflict zone is non-rectangular, as shown in Figure 3. For the non-level case, an exact analytical solution cannot be found. The projection could be "sliced" vertically and an iterative numerical solution could be developed, but that was not found to be necessary. Instead, a very good analytical approximation was obtained by using a bounding rectangle of equal-area around the projection of the cylindrical conflict zone, as illustrated in Figure 3.

The cross-sectional area of the projected conflict zone can be determined by integration. The details will not be presented here, but the area is

$$A_p = \mid \nu_z/\nu_t \mid \pi/\sqrt{\det} + 4H(\nu_h/\nu_t)y_{max}/\sigma_z \qquad (27)$$

where det was defined in (24). The width w of the projection is not changed by the last rotation, hence it is still y_{max} from (23). Hence, the height h of the equal-area bounding rectangle can be determined by simply dividing the area by the width. The result is

$$\begin{aligned} w/2 &= y_{max} \\ h/2 &= A_p/(4y_{max}) \end{aligned} \qquad (28)$$

The conflict probability can then be expressed as an integral in which the bounds of integration are the locations (in the transformed coordinate system) of the edges of the bounding rectangle:

$$\begin{aligned} y_0, y_1 &\equiv \Delta y \pm w/2 \\ z_0, z_1 &\equiv \Delta z \pm h/2 \end{aligned} \qquad (29)$$

where Δy and Δz are the transformed relative position coordinates, defined according to

$$\Delta p_R \equiv R\Delta p_T \equiv RT\Delta p \equiv \left[\begin{array}{ccc} \Delta x & \Delta y & \Delta z \end{array} \right]^T \qquad (30)$$

Because the three-dimensional probability density function decouples into the product of three scalar functions, $p(x,y,z) = p(x)p(y)p(z)$, and the sides of the bounding rectangle are parallel to the coordinate axes, the integral expression for the conflict probability P_c can be simplified as follows:

$$\begin{aligned} P_c &= \int_{z_0}^{z_1}\int_{y_0}^{y_1}\int_{-\infty}^{\infty} p(x,y,z)\,dx\,dy\,dz \\ &= \int_{y_0}^{y_1} p(y)\,dy \int_{z_0}^{z_1} p(z)\,dz \int_{-\infty}^{\infty} p(x)\,dx \\ &= \int_{y_0}^{y_1} p(y)\,dy \int_{z_0}^{z_1} p(z)\,dz \\ &= P(y_0,y_1)P(z_0,z_1) \\ &\equiv P_{hc}P_{vc} \end{aligned} \qquad (31)$$

where, for level flight, P_{hc} is the horizontal conflict probability and P_{vc} is the vertical conflict probability. For non-level flight, P_{vc} corresponds to the direction normal to the relative velocity and in the vertical plane that contains the relative velocity; P_{hc} corresponds to the plane normal to that direction. The function P is the cumulative normal probability function, defined as $P(a,b) \equiv \int_a^b p(s)ds$, where $p(s) \equiv \exp(-s^2/2)/\sqrt{2\pi}$. This function can be approximated analytically [12] with high accuracy and efficiency.

Computational efficiency is an important concern in an operational system. Basic timing tests were performed on the conflict probability algorithm running on a Sun Ultra 1 workstation to estimate the computation time required. These tests were for the CPE algorithm only and did not include wind modeling, trajectory prediction, conflict probing, or any other part of the problem. The average computation time per aircraft pair was approximately 0.55 milliseconds. This computation time is two to four orders of magnitude faster than a numerical solution, depending on the method and level of resolution of the numerical integration. The high speed and precision of the algorithm make it a very convenient analytical tool for generating plots. More importantly, the algorithm is easily efficient enough to be used in an operational ATM system. It can even be used repeatedly in an iterative conflict resolution algorithm, if necessary, to find a maneuver that will reduce the conflict probability to some specified level.

Heuristics and Caveats

It is assumed that the aircraft velocities and prediction errors are constant (in both magnitude and direction) during the encounter (reasonable period of potential conflict) and that the velocities are known during the encounter. As free flight is phased in, trajectories will tend to become more direct and have fewer turns, so the assumption of constant velocity will be reasonably accurate for the vast majority of encounters. Please note that the assumption of constant velocity during the encounter does *not* preclude planned turns or other maneuvers *prior to* the encounter. If the trajectory predictor knows about the planned maneuvers, and if those maneuvers end sufficiently far in advance of the encounter, they do not violate the assumption of constant velocity throughout the encounter. After a planned turn or other type of maneuver, any uncertainty in the exact maneuver can be accounted for by increasing the error covariance, but the details are beyond the scope of this paper.

It is worth noting that trajectory predictions are not required at the exact point of predicted minimum separation. If the predictions are in discrete time increments of several seconds, for example, the predicted positions nearest to minimum separation can be used as follows to determine the "exact" predicted time of minimum separation. For constant velocity, the time at which the minimum predicted separation occurs is

$$t_m = t_0 + \frac{\Delta p_0^T \Delta v}{\Delta v^T \Delta v} \tag{32}$$

where Δp_0 is the position difference at time t_0, and Δv is the constant velocity difference, both in terms of Cartesian coordinates. The position difference at minimum separation is then

$$\Delta p_m = \Delta p_0 + (t_m - t_0)\Delta v = \left[I + \frac{\Delta v \Delta v^T}{\Delta v^T \Delta v} \right] \Delta p_0 \tag{33}$$

The minimum separation distance itself is the magnitude of Δp_m.

Small variations in aircraft velocity due to wind disturbances or wind-optimal routing will have only a small effect in the immediate vicinity of an encounter, so they will usually not significantly violate the assumption of constant velocity. The predicted velocity at the point of minimum predicted separation is tangent to the flight-path and can be considered a first-order linear approximation to the actual trajectory at that point. In the unlikely event that a large heading, speed, or altitude change is scheduled in the vicinity of a potential conflict, the analytical solution for conflict probability will not be accurate and should not be used.

Three potential problems can occur for predicted conflicts with small relative velocity (similar headings and similar speeds). First, the wind-error cross-correlation may be high because wind error affects both aircraft similarly. A significant portion of the prediction error, therefore, cancels in the position difference, reducing the effect of the wind error. This reduction is desirable, but it may be difficult to model accurately. Second, the direction of the small relative velocity vector will be more sensitive to velocity errors, and the conflict probability estimate depends strongly on the direction of the relative velocity. Third, the small relative velocity also means that encounters of this type tend to have longer duration, so the assumptions of constant velocity and constant covariance during the encounter are more likely to be inaccurate. For these reasons, the CPE algorithm should probably not be used for extreme cases of this type. Fortunately, these types of conflicts develop slowly and give the controllers more time to respond, so CPE is not as important for these cases as it is otherwise.

Finally, a note about vertical conflict probability for level flight. In the current ATM system, aircraft altitudes are based on static pressure readings onboard the aircraft. Those readings, which are discretized in units of 100 ft (flight levels), are sent by the aircraft transponder to the ground tracking radar, which relays them to the host computer in the ARTCC Centers. For level flight, if the altitude is within 200 ft of the cleared altitude, the host computer rounds the measured altitude to the cleared altitude. Above FL290, the required altitude separation is 2000 ft, and if the nominal altitude separation is 2000 ft or more, the two aircraft are not considered to be in vertical conflict, even though they could possibly have only 1600 ft of altitude separation. However, if the nominal and required altitude separations are both 2000 ft, the CPE procedure computes a vertical conflict probability of 50%. Clearly, this is not a proper use of CPE. In the current ATM system, the vertical conflict probability is not very useful for level flight. However, it can be replaced very easily with discrete logic that simply sets the vertical conflict probability to 0 or 1, depending on the altitude separation.

path crossing angle, deg	minimum predicted separation, nmi	time to min separation, min			
		4	8	12	20
15	0	−.000	+.001	−.005	−.005
15	5	+.002	+.002	+.002	+.006
45	0	−.002	+.004	+.004	−.007
45	5	+.004	+.006	−.001	−.007
90	0	+.001	+.002	+.002	+.001
90	5	+.001	−.006	−.006	−.001
180	0	+.002	−.004	−.000	+.002
180	5	−.002	−.003	+.006	+.005

Table 1: Selected Monte Carlo simulation results for level flight: computed minus empirical conflict probability based on 10,000 samples per entry

path crossing angle, deg	minimum predicted separation, nmi	time to min separation, min			
		4	8	12	20
15	0	+.002	+.000	−.000	−.002
15	5	+.003	−.006	+.002	−.002
45	0	−.000	+.002	+.002	−.004
45	5	+.002	−.002	−.001	+.002
90	0	+.001	+.002	+.001	−.001
90	5	−.002	+.000	+.010	−.001
180	0	+.000	+.001	+.001	−.001
180	5	−.002	+.005	−.001	+.005

Table 2: Selected Monte Carlo simulation results for non-level flight: computed minus empirical conflict probability based on 10,000 samples per entry

Monte Carlo Simulation

The Gaussian prediction error model on which the CPE algorithm is based was determined empirically by analyzing actual air traffic data [9]. More work is being done in this area, but it is outside the scope of this paper. A Monte Carlo simulation was used to test the algorithm itself. The simulation was not intended to test the assumptions and the error model. In the Monte Carlo simulation, combinations of path-crossing angles, minimum predicted separations, and prediction times (time to minimum predicted separation) were generated. For each combination, the conflict probability was computed and nominal trajectories were generated. Then the nominal trajectories were perturbed by a series of random prediction errors, each consisting of constant cross-track position error and an along-track position error that increases linearly with prediction time. These randomly generated errors had the same expected statistics used in the conflict probability algorithm: 2 nmi rms cross-track error and 0.25 nmi/min rms along-track error growth rate. Wind-error cross-correlation was not modeled. The empirical fraction of cases in which conflicts resulted was compared with the computed conflict probability.

Table 1 shows a representative selection of the differences between the computed conflict probabilities and the Monte Carlo simulation results for level flight. The flight conditions are typical for a commercial transport aircraft in level flight. Each entry in Table 1 corresponds to a particular encounter geometry, and 10,000 Monte Carlo samples were run for each entry. Although not all cases are shown in Table 1, the path crossing angle was incremented in 15 deg steps from 15 deg to 180 deg, and the minimum predicted separation was incremented in steps of 2.5 nmi from 0 to 10 nmi, and the prediction time was incremented in steps of 4 min from 4 min to 24 min. The algorithmic results match well with the simulation results. The largest difference for all cases tested was 0.015 or 1.5% (it does not appear in Table 1, which gives only selected results). Given the accuracy of the underlying error model and the requirements of the application, this result is more than adequate. A maximum difference of perhaps 5% would have been considered adequate.

The expected standard deviation for each entry of Table 1 is $\sigma \equiv \sqrt{P_c(1-P_c)/N}$, where P_c is the conflict probability and N is the number of samples. Note that $P_c(1-P_c) = 0$ if $P_c = 0$ or $P_c = 1$, and the maximum possible value of $\sqrt{P_c(1-P_c)}$ is $1/2$ when $P_c = 1/2$. Thus, the maximum expected standard deviation for any entry cannot exceed 0.005. For each entry (and for the many other encounter geometries not listed in Table 1), the magnitude of the entry was divided by its corresponding σ to determine the normalized difference between the computed and empirical conflict rates. The maximum normalized difference was approximately 3.6. This result is reasonably consistent with the claim the CPE algorithm determines an exact solution for level flight under the previously stated assumptions. The algorithm is therefore validated for level flight.

Table 2 shows the same information as Table 1 except for non-level flight rather than level flight. The altitude separation at minimum horizontal separation is zero, and the descent rate of one of the aircraft is 1500 ft/min (the other aircraft is flying level). The rms vertical error growth rate in descent is 300 ft/min. These parameters are typical for encounters between an aircraft flying level and one in descent. As before, 10,000 Monte Carlo samples were run for each entry. Although not all cases are shown in Table 2, the path crossing angle was incremented in 15 deg steps from 15 deg to 180 deg, and the minimum predicted separation was incremented in steps of 2.5 nmi from 0 to 10 nmi, and the prediction time was incremented in steps of 4 min from 4 min to 24 min. Again, the algorithmic results matched well with the simulation results. The largest difference for all cases tested was 0.019, or less than 2%. The maximum normalized difference was 3.8. As before, this result is more than adequate. The equal-area

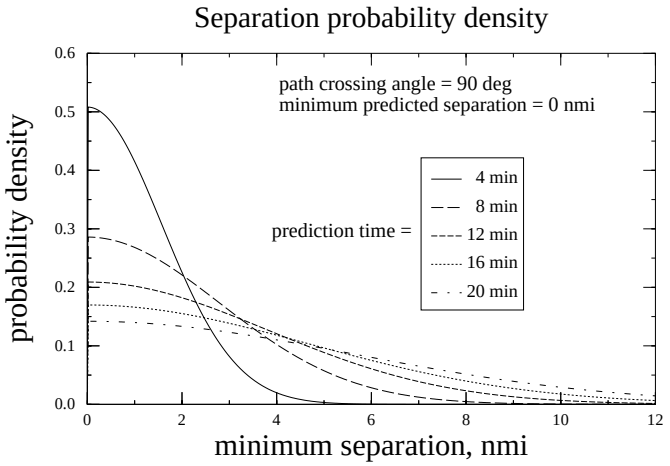

Figure 4: Separation probability density for level flight, with prediction time as a parameter

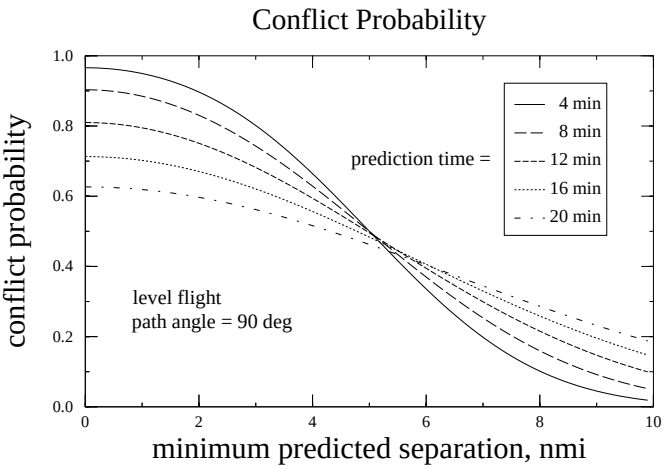

Figure 5: Conflict probability as a function of minimum predicted separation for level flight with a path crossing angle of 90 deg, with prediction time as a parameter

bounding-rectangle approximation for non-level flight (as shown in Figure 3) is therefore justified, and the algorithm is validated for non-level flight.

It was noted above that, for Table 2, the altitude separation at minimum horizontal separation was zero. Because the CPE accuracy could depend on this parameter, the simulation was rerun several times with different values. The altitude separation at minimum horizontal separation was varied by increments of 250 ft from -4000 to 4000 ft, and the equivalent of Table 2 was generated for each case. The worst case difference occurred when the altitude separation at minimum horizontal separation was 3000 ft, and the difference between the computed and empirical conflict rates was under 5%. Again, that is more than adequate.

Numerical Examples

A set of numerical examples of conflict probabilities and related quantities were generated and plotted for a wide range of encounter geometries. The efficient and precise analytical solution made these plots much faster to generate than they would have been if numerical integration were required. The aircraft speeds were 500 kn for steady, level flight and 300 kn for descent, with a nominal descent rate of 1500 ft/min. These are typical speeds for commercial transport aircraft. The conflict separation distance was 5 nmi horizontally and 2000 ft vertically, the current legal separation requirements above FL290. For level flight, the vertical rms error was 100 ft, the cross-track rms error was 2 nmi, and the along-track rms error started at 0.25 nmi and grew linearly at a rate of 0.25 nmi/min. For descent, prediction errors tend to be larger than in cruise: the vertical error grew at a rate of 300 ft/min, and the along-track rms error grew at a rate of 0.333 nmi/min. Wind-error cross correlation between aircraft was not modeled.

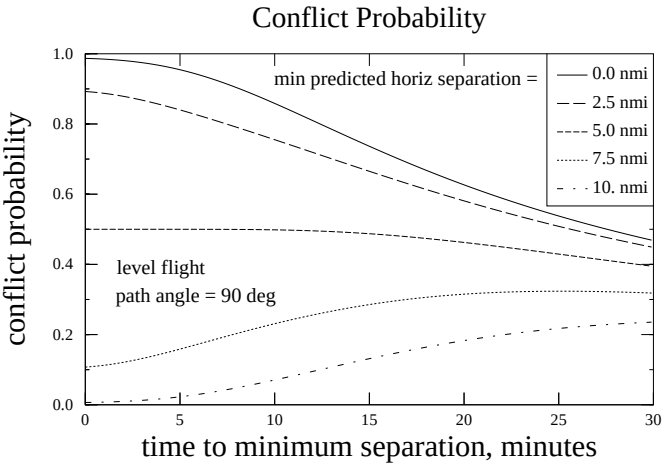

Figure 6: Conflict probability as a function of prediction time for level flight, with minimum predicted separation as a parameter

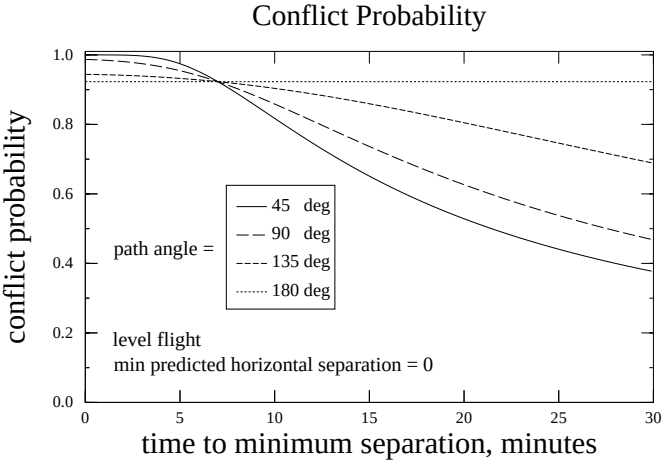

Figure 7: Conflict probability as a function of prediction time for level flight, with path crossing angle as a parameter

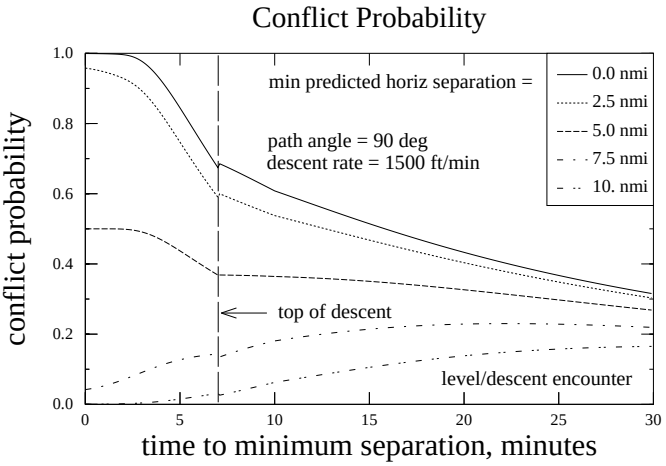

Figure 8: Conflict probability as a function of prediction time for non-level flight, with minimum predicted separation as a parameter

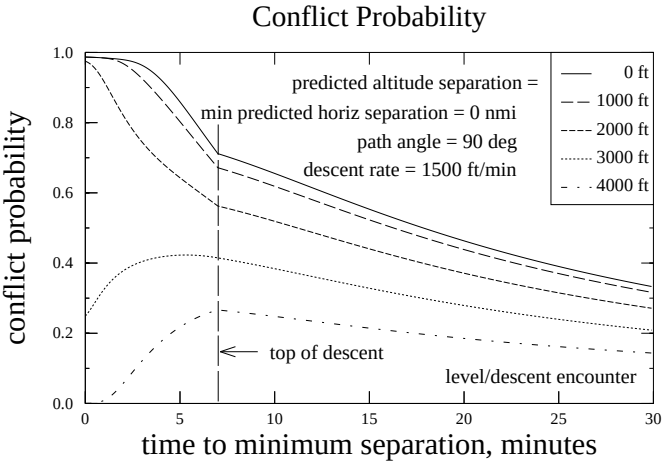

Figure 9: Conflict probability as a function of prediction time for non-level flight, with altitude separation as a parameter

Figure 4 shows the separation probability density for level flight, with prediction time (time to minimum predicted separation) as a parameter, where the path-crossing angle is 90 deg and the minimum predicted separation is zero nmi (an exact collision). This density function is the derivative, with respect to minimum separation, of the cumulative separation probability. It was determined by numerical differentiation of results from the CPE algorithm. The plot shows how the density function spreads out as prediction time increases. Note also that separation probability density is not Gaussian even though the error model on which it is based is Gaussian. This is because separation is a *nonlinear* function (root-sum-square) of the relative position coordinates.

In Figure 5 conflict probability is plotted as a function of minimum predicted separation for level flight with a path crossing angle of 90 deg, with prediction time as a parameter. This figure shows the dependence of conflict probability on the predicted minimum separation for a particular class of encounters. The nonlinear dependence is interesting because, before CPE was developed, the minimum predicted separation was the best indicator of conflict probability, and it was often used as an intuitive substitute for conflict probability. The controller had to create a crude mental model of the function, which was difficult enough without even considering the effect of prediction time and path crossing angle. Now that can all be done automatically for the controller.

In Figure 6 conflict probability is plotted as a function of prediction time for level flight, with the minimum predicted separation as a parameter, where the path-crossing angle is 90 deg. For small prediction times, the covariances are small and the conflict probabilities are a strong function of minimum predicted separation. For larger prediction times, the covariances grow and the conflict probability becomes a weaker function of the minimum predicted separation. The conflict probabilities converge and asymptotically approach zero as prediction time increases. When the minimum predicted separation is greater than the minimum allowed separation, the conflict probability decreases monotonically with prediction time. However, when the minimum predicted separation is less than the minimum allowed separation, the conflict probability first increases to a maximum and then decreases. This is because the error ellipses start apart and begin to overlap as they grow, but then as they continue to grow they become more diffuse.

In Figure 7 conflict probability is plotted again as a function of prediction time for level flight, but with the path-crossing angle as a parameter, where the predicted minimum separation is 0 nmi. As the prediction time increases, the conflict probability decreases faster for smaller path-crossing angles. If wind-error cross correlation were taken into account, however, these curves would be very different for small path crossing angles. A portion of the trajectory prediction error would cancel in the position difference, and the effective error growth rate would be smaller. Hence the conflict probabilities for smaller path angles would be higher than these shown in the Figure 7. The same information was also plotted with speed difference as a parameter, but they are not shown.

Figure 8 shows the first example of results for non-level flight. It shows the plot corresponding to Figure 6, but for a level/descent encounter. One of the aircraft is in a typical descent for the final 7 min before minimum separation, hence the top of descent appears at a prediction time of 7 min. The effect of the increased error growth rate in descent is apparent, and the accumulated effect extends back into the period before the descent even starts. The increased uncertainty in non-level encounters obviously tends to make them more difficult than level encounters for controllers. CPE can augment the controllers judgement by providing a systematic method of determining the appropriate time to alert them of a potential conflict and the

necessity for resolution.

In Figure 9 conflict probability is plotted again as a function of prediction time for a level/descent encounter, but this time with predicted altitude separation (at the point of minimum horizontal separation) as a parameter. This figure shows the effects of increased altitude uncertainty (as well as non-level flight path) for a typical non-level encounter. Note that when the predicted altitude separation is 3000 ft, or even 4000 ft, the probability of conflict is still significant, whereas in level flight it is insignificant when the altitude separation is the minimum required 2000 ft. CPE can therefore give controllers guidance on the altitude separation buffers required for encounters involving climbing and descending aircraft.

Conclusion

A method that had been previously developed to estimate the probability of conflict for aircraft pairs in level flight has been generalized to non-level flight. A Monte Carlo simulation was used to successfully test the accuracy of the algorithm (but not the validity of the assumptions or the accuracy of the underlying error model, which are outside the scope of this paper). The CPE algorithm is computationally efficient enough to be used directly in an operational ATM decision support system, and it has been integrated into CTAS and tested at two ARTCC facilities. It has also been tested and calibrated off-line on real air traffic prediction data from CTAS. In future work, CPE will be used to help air traffic controllers determine when and how to efficiently resolve conflicts.

References

[1] Paielli, R.A., Erzberger, H.: "Conflict Probability Estimation For Free Flight," *J. Guidance, Control, and Dynamics*, vol. 20, no. 3, May-June 1997, pp. 588-596.

[2] Erzberger, H.; Nedell, W.: "Design of Automated System for Management of Arrival Traffic", NASA Technical Memorandum TM-102201, June, 1989.

[3] Etzberger, H; Paielli, R.A.: "Conflict Prediction and Resolution In The Presence of Prediction Error," *1st USA/Europe Air Traffic Management R&D Seminar*, Saclay, France, June 17-20, 1997.

[4] Yang, L.C.; Kuchar, J.K.: "Prototype Conflict Alerting System for Free Flight," *J. Guidance, Control, and Dynamics*, vol. 20, no. 4, July-Aug 1997, pp 768-776.

[5] Warren, A.; Schwab, R.W.; Geels, T.J.; Shakarian, A.: "Conflict Probe Concepts Analysis in Support of Free Flight," Appendix B, NASA Contractor Report 201623, Jan. 1997. (available at http://techreports.larc.nasa.gov/ltrs/)

[6] Krozel, J.; Peters, M.E.; Hunter, G.: "Conflict Detection and Resolution for Future Air Transportation Management," NASA TR 97138-01, pp. 26-33, April 1997.

[7] Erzberger, H.; Davis, T.J.; Green, S.: "Design of Center-TRACON Automation System," *AGARD Guidance and Control Symposium on Machine Intelligence in Air Traffic Management*, Berlin, Germany, May 1993. (see also http://george.arc.nasa.gov/af/afa/ctas/welcome)

[8] Slattery, R.; Zhao, Y.: "Trajectory Synthesis For Air Traffic Automation," *J. Guidance, Control, and Dynamics*, vol. 20, no. 2, pp. 232-238, March-April 1997.

[9] Paielli, R.A.: "Empirical Test of Conflict Probability Estimation," *2nd USA/Europe Air Traffic Management R&D Seminar*, Orlando, Florida, Dec. 1-4, 1998.

[10] Benjamin, S.G.; Brundage, K.J.; and Morone, L.L.: "The Rapid Update Cycle, Part I: Analysis/Model Description," *Technical Proceedings Bulletin* no. 416, June 1994, NOAA/NWS. (http://www.fsl.noaa.gov/frd-bin/tpbruc.cgi)

[11] Cole, R.E.; Richard, C.; Kim, S.; Bailey, D.: "An Assessment of the 60 km Rapid Update Cycle (RUC) with Near Real-Time Aircraft Reports," Project Report NASA/A-1, MIT Lincoln Laboratory, July 15, 1998.

[12] Press, W.H.; Tuekolsky, S.A.; Vetterling, W.T.; Flannery, B.P.: *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University Press, 1992.

[13] Bach, R.E.: A Matlab program for visualizing transformed encounter geometries, Ames Research Center, Feb 1999. Available upon request.