Tactical Conflict Detection and Resolution in a 3-D Airspace*

Gilles Dowek† Alfons Geser‡ César Muñoz§

October 12, 2001

Abstract

This paper presents an algorithm for detection and resolution of air traffic conflicts between two aircraft, namely ownship and intruder, in a 3-dimensional airspace. The input to the algorithm is the state information, i.e., three-dimensional position and velocity vectors, of both aircraft. A conflict is predicted using a linear projection of the aircraft states. The algorithm outputs a set of solutions. Each solution is a single maneuver, to be performed by the ownship, that effectively keeps the required minimum separation. No cooperation of the intruder aircraft is required. Each solution modifies only one state parameter of the ownship: ground track, ground speed, or vertical speed. The algorithm accounts for the combined horizontal and vertical separation. Our aim is to have correct solutions under all circumstances. For this purpose we do a rigorous analysis of all special cases. To reinforce this reliability quest, we use formal arguments to derive the algorithm from a mathematical definition of the problem.

*This work was supported by the National Aeronautics and Space Administration under NASA Contract No. NAS1-97046 while the authors were in residence at ICASE, NASA Langley Research Center, Hampton, VA 23681-2199, USA.

†INRIA, Domaine de Voluceau - Rocquencourt - B.P. 105, 78153 Le Chesnay Cedex, France, e-mail: Gilles.Dowek@inria.fr.

‡ICASE, Mail Stop 132C, NASA Langley Research Center, Hampton, VA 23681-2199, USA, e-mail: geser@icase.edu.

§ICASE, Mail Stop 132C, NASA Langley Research Center, Hampton, VA 23681-2199, USA, e-mail: munoz@icase.edu.

1 Introduction

One of the main elements of the Free-Flight concept [12] is the redistribution of responsibilities for air traffic separation. Under Free-Flight rules, each aircraft with an appropriate level of equipment is responsible for keeping separation with other aircraft in the vicinity. To support this mode of operation, several automated decision support systems are being proposed. In this context, Conflict Detection and Resolution (CD&R) algorithms are designed to warn pilots about an imminent loss of separation, and to assist them to perform a corrective maneuver.

In this paper, we present a *tactical* CD&R algorithm for *two* aircraft in a 3-D space. In CD&R-related literature, *tactical* refers to the exclusive use of state information to project aircraft trajectories. Due to this intentionally limited source of information, they are rather used with short lookahead times (a few minutes, typically 5-10) during which aircraft are assumed to follow straight flight paths. *Strategic* approaches, in contrast, use intent information such as flight plans, and are aware of hazards such as weather conditions. They may have lookahead windows of several minutes and even hours. For a survey on CD&R methods see [8].

Distributed Air/Ground Traffic Management (DAG-TM) [1] is a set of concept elements, developed within the Advanced Air Transportation Technologies project at NASA, that defines modes of operation supporting the Free-Flight paradigm. Prototype tools such as the Autonomous Operations Planner (AOP) are being developed at NASA Langley to study the feasibility of self separation under a variety of operational constraints. Systems with related goals have been proposed in other research laboratories, e.g., the Future ATM Concepts Evaluation Tool (FACET) [3] at NASA Ames[1] and the Airborne Separation Assurance System (ASAS) [7] at the National Aerospace Laboratory (NLR) in the Netherlands. All these tools implement CD&R algorithms.

Safety assessment of new air traffic management systems is a main issue in DAG-TM. Standard safety assessment techniques such as testing and simulation have serious limitations in new, significantly more autonomous, systems. Failure in rare special cases may go unnoticed even after thorough simulation and testing. Even if such failures are unlikely, they are unacceptable. Given the critical nature of the problem, we plea that safety statements are made and verified formally, and that proofs are checked by machine. We are convinced that only formal verification can provide the demanded degree of reliability. Our algorithm is designed so as to be verified formally.

Two aircraft are said to be in *conflict* if their vertical separation is strictly less than H and their horizontal separation is strictly less than D. Values of $H = 1000$ ft (feet) and $D = 5$ nm (nautical miles) are commonly used. A body in 3-D space, called *protected zone*, is assigned to each aircraft such that a conflict is equivalent to an intrusion of another airplane into its protected zone. The protected zone forms a cylinder of altitude $2H$ and radius D around the position of the aircraft. Note that the boundaries are not considered part of the protected zone. We will see later that this choice enables optimal ownship maneuvers that solve potential conflicts by touching the boundaries of the intruder protected zone.

2 Related Work

Our algorithm takes ideas from several previously proposed algorithms for air traffic conflict detection and resolution.

2.1 Conflict Detection

The input to the *strategic AOP conflict detection algorithm* [9] is a set of flight paths, one of which is the ownship flight path. A flight path is a list of points, where each point is assumed to be joined to its successor by a straight line segment. Each such point is assigned the intended aircraft state at this point. Conflicts are detected by checking the distance between the flight paths at time steps of Δ seconds. Since the algorithm does not compute the actual time when the first loss of separation occurs, the choice of Δ is crucial. Indeed, if Δ is too large, near misses can occur without being detected.

Tactical conflict detection algorithms, as for example the ones implemented in ASAS and FACET [7, 2], reason in analytical geometry. In a 2-D framework, separation is lost if the horizontal distance between two aircraft is strictly less than D. Thus one can construct, and solve, a quadratic equation that has solutions when a conflict occurs. We have successfully verified a conflict alerting algorithm using a geometric trajectory model [10]. On the other hand, we have experienced that discretization makes formal verification difficult [4].

2.2 Conflict Resolution

The *Modified Potential Field resolution algorithm* (originally due to Eby [6] and implemented in ASAS [7]) computes the time τ of closest approach between the two aircraft. The algorithm proposes a new velocity vector such that at time τ the distance between the aircraft is D. It is easy to see that this maneuver improves the separation between ownship and intruder, but several maneuvers are required to solve the conflict.

This problem does not appear in the 2-D *Geometric Optimization Resolution algorithm*, proposed by K. Bilimoria [2] and implemented in FACET [3]. The intruder is considered fixed in space. The ownship position and velocity vector are taken relative to the intruder. A new relative velocity vector for the ownship solves the conflict if it does not intersect the protected zone. Among the infinitely many new velocity vectors that solve the conflict, the algorithm

[1] FACET is a CD&R analysis tool rather than a flight deck decision support tool.

chooses those which minimize their angle to the original velocity vector. Such velocity vectors are called *optimal*.

Since the protected zone is a disk, the optimal solutions lie on tangents to the disk. Any other solution requires a greater change of the ownship ground track. Given the direction of the new relative velocity vector, its length may be chosen arbitrarily. Bilimoria's algorithm proposes a solution that minimizes the change of the velocity vector.

3 An Algorithm for 3-D CD&R

In this paper, we only give a sketch of our new algorithm. The reader may refer to the technical report [5] for more details.

In the Modified Potential Field algorithm implemented in ASAS, 3-D conflicts are decomposed into horizontal and vertical conflicts. The solution to the horizontal conflict is then merged with the solution to the vertical conflict in order to provide a 3-D maneuver.

We present a true 3-D extension to the Geometric Optimization algorithm which is based on an accurate mathematical development and which offers a complete analysis of special cases.

3.1 3-D conflict detection

We start from a 3-D coordinate system where the intruder is fixed at the origin. Position, $\vec{a}$, and speed, $\vec{v}$, of the ownship are given relative to the intruder. We define the ownship trajectory as the half line originating from the ownship current position, going along its velocity vector as time t passes: $\vec{a} + t\vec{v}$, $t > 0$.

An ownship at position (x, y, z) is in *conflict* with the intruder if it is in the protected zone, defined by

$$P \quad = \quad \{(x, y, z) \mid x^2 + y^2 < D^2 \text{ and } -H < z < H\}.$$

The two aircraft are *predicted* to be in *conflict* if the ownship trajectory intersects the protected zone P. That is, $\vec{a} + t\vec{v} \in P$ for some $t > 0$. The conflict detection problem is to find out whether such t exist. It can be solved as follows. We assume that there is no conflict at time $t = 0$. The trajectory intersects the interior of the flat cylinder, P, if it intersects twice the boundary of P. The two intersections with the boundary determine the time interval of intrusion to the protected zone. The cylinder boundary consists of (1) its lateral surface

$$P_1 \quad = \quad \{(x, y, z) \mid x^2 + y^2 = D^2 \text{ and } -H \leq z \leq H\}$$

and (2) its top and bottom bases

$$P_2 \quad = \quad \{(x, y, z) \mid x^2 + y^2 < D^2 \text{ and } |z| = H\}.$$

Note that the top and bottom disks belong to P_1.

Now, either the vertical velocity component, v_z, is zero, in which case we have the special 2-D case. Or it is non-zero, in which case we can determine the times t_1 and t_2 of intersection of the ownship trajectory with the horizontal planes at altitudes $\pm H$. We distinguish cases whether or not the distances d_1, d_2 of the ownship to the intruder at times t_1 and t_2, respectively, exceeds D. In the case of a predicted conflict, we will get either 0, 1, or 2 intersections of the trajectory with the bases, and 2, 1, or 0 intersections, respectively, with the lateral surface. In no conflict is predicted, there are no intersections.

3.2 3-D conflict resolution

We assume that at time $t = 0$ ownship and intruder are not in conflict, but that they are predicted to be in conflict. We want a modified velocity vector $\vec{v'}$ such that the new trajectory $\vec{a} + t\vec{v'}$, $t > 0$ is not predicted to be in conflict.

Among the various solutions, we are interested in the optimal ones, i.e., those where the angle between $\vec{v}$ and $\vec{v'}$ is minimal in the half plane containing these vectors. Positive multiples of optimal solutions are again optimal solutions. Hence, the kernel of our algorithm is a procedure that characterizes the *directions* of optimal non-zero solutions.

In an optimal non-zero solution the half line $\vec{a} + t\vec{v'}$ must touch the boundary of the protected zone, and since it is a solution, it must not enter the protected zone. We characterize each solution by one of its touching points at the cylinder's boundary. We call such points *target points*, and the set of all target points the *target set*.

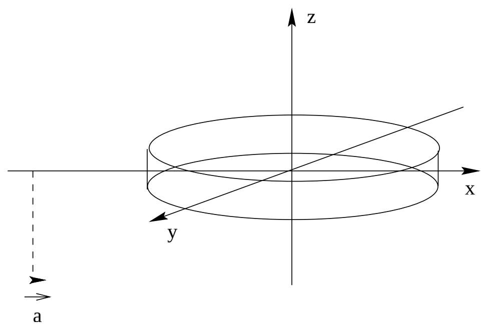

Figure 1: 3-D coordinate system and protected zone

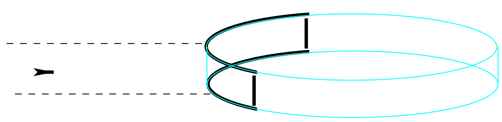

Figure 2: Target set. Case $-H < a_z$ and $-a_x > D$

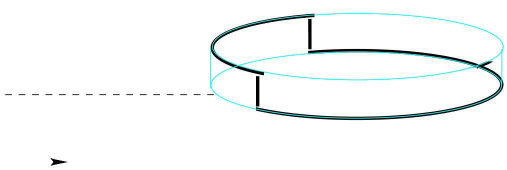

Figure 3: Target set. Case $a_z < -H$ and $-a_x \geq D$

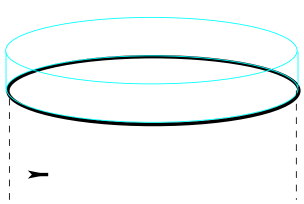

Figure 4: Target set. Case $a_z < -H$ and $-a_x < D$

For convenience we assume the coordinate system rotated around the z axis so as to ensure $a_x \leq 0$ and $a_y = 0$ for the relative initial ownship position, (a_x, a_y, a_z). By symmetry, we may furthermore assume that $a_z \leq 0$. Otherwise, we mirror the whole arrangement at the xy plane. See Figure 1.

We get the following description of the target set.

1. Case $-H < a_z$ and $-a_x > D$. In this case, the target set is the set of points (x, y, z) of P_1 such that $x = D^2/a_x$ or $(z = H$ and $x < D^2/a_x)$ or $(z = -H$ and $x < D^2/a_x)$. See Figure 2.

2. Case $a_z \leq -H$ and $-a_x \geq D$. In this case, the target set is the set of points (x, y, z) of P_1 such that $x = D^2/a_x$ or $(z = H$ and $x < D^2/a_x)$ or $(z = -H$ and $x > D^2/a_x)$. See Figure 3.

3. Case $a_z < -H$ and $-a_x < D$. In this case, the target set is the set of points (x, y, z) of P_1 such that $z = -H$. See Figure 4.

4 Constrained Solutions

Every point from the infinite set of target points defines a direction. For each direction, we may chose an arbitrary length for the relative velocity vector $\vec{v'}$. Each $\vec{v'}$ determines a maneuver that optimally solves the conflict. Some of these maneuvers are more desirable than others. For instance, one might have only one flight parameter changed. If a change of ground speed is capable of solving the conflict, then one may disregard a combined change of ground speed and ground track. In what follows, we use the subscripts o and i to indicate state values associated to the ownship and the intruder aircraft, respectively.

We have done an exhaustive analysis for each of the following constraints on the new (absolute!) ownship velocity vector $\vec{v'_o} = (v'_{ox}, v'_{oy}, v'_{oz})$:

- only ground speed may change: $\vec{v'_o} = (kv_{ox}, kv_{oy}, v_{oz})$ for some $k > 0$,
- only ground track may change: $v'^2_{ox} + v'^2_{oy} = v^2_{ox} + v^2_{oy}$, $v'_{oz} = v_{oz}$,
- only vertical speed may change: $v'_{ox} = v_{ox}$, $v'_{oy} = v_{oy}$.

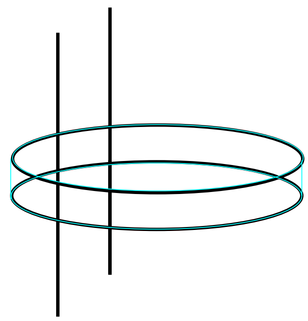

Figure 5: Superset of target points

For the analysis we utilize the fact that the target set is a subset of the point set drawn in Figure 5. For each of the three constraints, the analysis considers cases:

1. target points on the boundary circles of the lateral surface of Figure 5; these are defined by the system of equations
$$x^2 + y^2 = D^2 \quad (1)$$
$$z = \varepsilon H \quad \varepsilon = \pm 1 \quad (2)$$
2. target points on the vertical lines of Figure 5; these are defined by the system of equations
$$x = \frac{D^2}{a_x} \quad (3)$$
$$y = \frac{-\varepsilon D}{a_x}\sqrt{a_x^2 - D^2} \quad \varepsilon = \pm 1 \quad (4)$$
3. and also the zero solution $\vec{v'} = \vec{0}$, i.e., $\vec{v'_o} = \vec{v_i}$.

In each of these cases, quadratic equations obtained from Equations 1 and 2 or from Equations 3 and 4, respectively, are solved, and the solutions are checked for the target point property.

4.1 Ground Speed Change Only

We have $\vec{v'}_o = (k\ v_{ox}, k\ v_{oy}, v_{oz})$ for some $k > 0$. We must determine the possible k positive such that at some time t, the point $(x, y, z) = (a_x + t(v'_{ox} - v_{ix}), t(v'_{oy} - v_{iy}), a_z + t(v'_{oz} - v_{iz}))$ is on the target set.

4.1.1 Points on the circles

Equation 2 yields
$$t = (\varepsilon H - a_z)/(v_{oz} - v_{iz}).$$
Equation 1 is instantiated to
$$(a_x + t(kv_{ox} - v_{ix}))^2 + (t(kv_{oy} - v_{iy}))^2 = D^2,$$
which rewrites to
$$\begin{aligned} & t^2(v_{ox}^2 + v_{oy}^2)k^2 \\ + \; & (2(a_x - tv_{ix})tv_{ox} - 2t^2 v_{oy} v_{iy})k \\ + \; & (a_x - tv_{ix})^2 + t^2 v_{iy}^2 - D^2 = 0. \end{aligned}$$
We solve this equation for k.

We analyze the singularities of this case.

- Case $v_{oz} = v_{iz}$. The vector $\vec{v_o} - \vec{v_i}$ is horizontal and will remain horizontal if we change the ground speed of the ownship. Solutions, if any, are tangential to the circle and also belong to a line and those are handled in case 2.

  Therefore, there are no solutions.
- Case $(v_{ox}^2 + v_{oy}^2)D^2 < ((a_x - tv_{ix})v_{oy} + tv_{ox}v_{iy})^2$. In this case the quadratic equation for k has no solutions. Therefore, there are no solutions.

4.1.2 Points on the lines

From Equation 3 and Equation 4 one derives
$$t(kv_{ox} - v_{ix}) = \frac{D^2}{a_x} - a_x \quad (5)$$
$$t(kv_{oy} - v_{iy}) = \frac{-\varepsilon D\sqrt{a_x^2 - D^2}}{a_x}. \quad (6)$$
Multiplication of Equation 5 with opposite sides of Equation 6 and cancellation yields
$$\sqrt{a_x^2 - D^2}(kv_{oy} - v_{iy}) = \varepsilon D(kv_{ox} - v_{ix}) \quad (7)$$
Thus, we deduce
$$k = \frac{v_{iy}\sqrt{a_x^2 - D^2} - v_{ix}\varepsilon D}{v_{oy}\sqrt{a_x^2 - D^2} - v_{ox}\varepsilon D}, \quad (8)$$
$$t = \frac{-\varepsilon D v_{ox}\sqrt{a_x^2 - D^2} + (a_x^2 - D^2)v_{oy}}{a_x(v_{ix}v_{oy} - v_{ox}v_{iy})}\ . \quad (9)$$
We analyze the singularities of this case.

- Case $-a_x < D$. No solution, since there are no lines.

- Case $-a_x = D$. There is a single line and the relative initial position of the ownship is on that line. The only way to reach a target point on that line is if $\vec{v'}$ is vertical or null. If $v_{ix}v_{oy} - v_{ox}v_{iy} = 0$ then there is a solution: $v'_{ox} = v_{ix}, v'_{oy} = v_{iy}$. Otherwise, there is none.

- Case $v_{ix}v_{oy} - v_{ox}v_{iy} = 0$. The horizontal components of $\vec{v_i}$ and $\vec{v_o}$ are parallel. The direction of $\vec{v'}$ is independent of k and, by predicted conflict for $k = 1$, is not tangent to the lateral surface of the cylinder.

- Case $\varepsilon D v_{ox} = v_{oy}\sqrt{a_x^2 - D^2}$. No solution, since $t = 0$.

4.1.3 Special case $\vec{v'} = \vec{0}$

The only way to reach $\vec{v'} = \vec{0}$ by changing the ownship ground speed is to have $v_{ix}v_{oy} - v_{ox}v_{iy} = 0$ and $v_{oz} = v_{iz}$.

4.2 Ground Track Change Only

Since ground speed and vertical speed are constant, we have

$$\begin{aligned} v'^2_{ox} + v'^2_{oy} &= v^2_{ox} + v^2_{oy} && (10)\\ v'_{oz} &= v_{oz} && (11)\end{aligned}$$

and we must determine the possible v'_{ox} and v'_{oy} such that at some time t, the point $(x, y, z) = (a_x + t(v'_{ox} - v_{ix}), t(v'_{oy} - v_{iy}), a_z + t(v'_{oz} - v_{iz}))$ is on the target set.

4.2.1 Points on the circles

Equation 2 gives $t = (\varepsilon H - a_z)/(v_{oz} - v_{iz})$. Equation 1 rewrites to

$$2t^2 v'_{oy} v_{iy} = E + 2(a_x - tv_{ix})tv'_{ox}, \qquad (12)$$

where

$$\begin{aligned} E &= (a_x - tv_{ix})^2 + t^2v^2_{iy} + t^2v'^2_{ox} + t^2v'^2_{oy} - D^2\\ &= (a_x - tv_{ix})^2 + t^2v^2_{iy} + t^2v^2_{ox} + t^2v^2_{oy} - D^2.\end{aligned}$$

Squaring Equation 12 yields

$$4t^4 v'^2_{oy} v^2_{iy} = (E + 2(a_x - tv_{ix})tv'_{ox})^2.$$

Using Equation 10, we get

$$4t^4 v^2_{iy}(v^2_{ox} + v^2_{oy} - v'^2_{ox}) = (E + 2(a_x - tv_{ix})tv'_{ox})^2$$

Hence,

$$\begin{aligned} & 4t^2((a_x - tv_{ix})^2 + t^2v^2_{iy})v'^2_{ox} && (13)\\ +\; & 4E(a_x - tv_{ix})tv'_{ox} && (14)\\ +\; & E^2 - 4t^4v^2_{iy}(v^2_{ox} + v^2_{oy}) = 0. && (15)\end{aligned}$$

To solve Equation 13 on v'_{ox}, we compute its discriminant $\Delta = 16v^2_{iy}t^4\Delta'$, where

$$\Delta' = -E^2 + 4t^2(v^2_{ox} + v^2_{oy})((a_x - tv_{ix})^2 + v^2_{iy}t^2).$$

If $\Delta' \geq 0$, the solutions are

$$v'_{ox} = \frac{-E(a_x - tv_{ix}) + \varepsilon' v_{iy} t\sqrt{\Delta'}}{2t((a_x - tv_{ix})^2 + t^2v^2_{iy})}, \qquad (16)$$

where $\varepsilon' = \pm 1$. Then, we get

$$\begin{aligned} v'_{oy} &= \frac{E + 2(a_x - tv_{ix})tv'_{ox}}{2t^2v_{iy}} && (17)\\ &= \frac{Etv_{iy} + \varepsilon'(a_x - tv_{ix})\sqrt{\Delta'}}{2t((a_x - tv_{ix})^2 + t^2v^2_{iy})}. && (18)\end{aligned}$$

We analyze the singularities.

- Case $v_{oz} = v_{iz}$. The vector $\vec{v_o} - \vec{v_i}$ is horizontal and will remain horizontal if we change the ground track of the ownship. Solutions, if any, are tangential to the circle and also belong to a line and those are handled in case 2.

- Case $v_{iy} = 0$ and $a_x \neq tv_{ix}$. Then Equation 12 yields one solution for v'_{ox} which meets Equation 16. Equation 10 gives two solutions for v'_{oy} which meet Equation 18.

- Case $v_{iy} = 0$ and $a_x = tv_{ix}$. At time t, the intruder will be where the ownship is at time 0. Changing the ownship ground track does not affect the horizontal distance between the aircraft at time t (that will be $t\sqrt{v^2_{ox} + v^2_{oy}}$ in all cases). Therefore, it does not help to solve the conflict.

4.2.2 Points on the lines

If $-a_x \neq D$ then $v'_{ox} = v_{ix}$ is not a solution and Equation 3 gives t in function of v'_{ox}

$$t = \frac{D^2 - a_x^2}{a_x(v'_{ox} - v_{ix})}.$$

Equation 4 rewrites to

$$\begin{aligned} \frac{(D^2 - a_x^2)(v'_{oy} - v_{iy})}{a_x(v'_{ox} - v_{ix})} &= -\frac{\varepsilon D}{a_x}\sqrt{a_x^2 - D^2} \\ \frac{v'_{oy} - v_{iy}}{v'_{ox} - v_{ix}} &= \frac{\varepsilon D}{\sqrt{a_x^2 - D^2}} \\ v'_{oy} &= \frac{\varepsilon D(v'_{ox} - v_{ix})}{\sqrt{a_x^2 - D^2}} + v_{iy} \end{aligned}$$

Replacing v'_{oy} in Equation 10, we get

$$\left(\frac{\varepsilon D(v'_{ox} - v_{ix})}{\sqrt{a_x^2 - D^2}} + v_{iy}\right)^2 + v'^2_{ox} = v_{ox}^2 + v_{oy}^2.$$

We solve this equation and get v'_{ox} and then v'_{oy}.

We analyze the singularities.

- Case $-a_x < D$. No solution since there are no lines.
- Case $-a_x = D$. The only solution is $v'_{ox} = v_{ix}$, $v'_{oy} = v_{iy}$. In this case, we must have $v_{ix}^2 + v_{iy}^2 = v_{ox}^2 + v_{oy}^2$. Hence, $\vec{v'}$ is vertical or $\vec{0}$.

4.2.3 Special case $\vec{v'} = \vec{0}$

The only way to get $\vec{v'} = \vec{0}$ by changing the ownship ground track is to have $v_{ix}^2 + v_{iy}^2 = v_{ox}^2 + v_{oy}^2$ and $v_{oz} = v_{iz}$. In this case, we take $\vec{v'}_o = \vec{v}_i$.

4.3 Vertical Speed Change Only

If the target point is on a line then it is also on a circle. So we have only two cases here.

4.3.1 Points on the circles

Equation 1 rewrites to

$$(a_x + t(v_{ox} - v_{ix}))^2 + (t(v_{oy} - v_{iy}))^2 = D^2.$$

We solve this equation to get t. Equation 2 yields

$$v'_{oz} = v_{iz} + (\varepsilon H - a_z)/t.$$

If $v_{ox} = v_{ix}, v_{oy} = v_{iy}$, Equation 1 has a solution only when $-a_x = D$ and in this case there is no conflict.

4.3.2 Special case $\vec{v'} = \vec{0}$

The only way to reach $\vec{v'} = \vec{0}$ by changing the ownship vertical speed is to have $v_{ox} = v_{ix}$ and $v_{oy} = v_{iy}$. We take $v'_{oz} = v_{iz}$.

5 A Prototype Implementation

We have implemented the algorithm in a prototype written in Java. The prototype is a few hundred lines of code containing assignments and conditionals. Expressions use the four basic arithmetic operations and square root, but no trigonometric functions. We do use trigonometric functions in the interface, to print the ground track of the aircraft from the computed Cartesian coordinates of the velocity vector. The implementation is available at `http://www.icase.edu/~munoz/sources.html`.

Here is a typical execution: we have two aircraft flying at the same altitude with a horizontal separation of 10 nm. In the coordinate system where the intruder is at the origin and the ownship at coordinates $(-10, 0, 0)$, the ground track of the ownship is 0 and the ground track of the intruder is 180°. The ground speed of the ownship is 400 nm/h and that of the intruder is 300 nm/h. The ownship is climbing at a vertical speed of 1000 ft/mn and the intruder is descending at a vertical speed of -1000 ft/mn. The input to the algorithm is a file containing the following information.

```
Ground distance = 10 nm  Vertical distance = 0 ft
Ownship:    0 deg   400 nm/h   1000 ft/mn
Intruder: 180 deg   300 nm/h   -1000 ft/mn
```

The programs detects a conflict and proposes five solutions:

```
Conflict in the time interval (25.7143,29.1456)

There are 5 solutions.

Modify GROUND SPEED     317.5889 nm/h  (TOP)
Modify GROUND TRACK      29.1888 deg   (TOP)
Modify GROUND TRACK     -29.1888 deg   (TOP)
Modify VERTICAL SPEED  1266.8799 ft/mn (TOP)
Modify VERTICAL SPEED -3266.8799 ft/mn (BOTTOM)
```

The first solution is to reduce ground speed to 317 nm/h. The second and third modify ground track. The last two solutions modify vertical speed. On the other hand, in the first four solutions, the target points are on the top circle of the target set. In the last solution, the target point is on the bottom circle.

Notice that some solutions may not be physically possible. For instance, the last solution proposes an absolute change of vertical speed of more than 4000 ft/mn. Our algorithm does not check the solutions for satisfaction of aircraft capabilities. Such checks may however easily be added.

6 Conclusion

We have given a complete and rigorous analysis of tactical detection and resolution of air traffic conflicts in the 3-dimensional space and described a new CD&R algorithm that produces a set of solutions. Each solution proposed by the algorithm is a constrained single maneuver that, when performed by the ownship, solves the conflict without collaboration of the intruder aircraft. Experiments have indicated that our algorithm always yields at least two solutions. After thousands of randomly generated examples the average was three solutions per conflict.

Our algorithm can be integrated within a more comprehensive system, such as AOP, to detect and solve conflicts in piece-wise linear flight plans. It is well suited to serve this purpose. First, it is efficient. Particularly, it does not contain loops nor calls to trigonometric functions. Moreover, extra information such as intent information, aircraft performance information, and area hazard information may be used to chose one maneuver from the set that our algorithm offers.

We designed our algorithm so as to enable its formal verification. Our next step is to prove in PVS [11] formally that every proposed maneuver indeed solves the conflict and that the set of proposed maneuvers is always non-empty.

Acknowledgment

The authors are grateful to David Wing and Rick Butler for their helpful comments on preliminary versions of this manuscript.

References

[1] Advanced Air Transportation Technologies (AATT) Project. Concept definition for Distributed Air/Ground Traffic Management (DAG-TM), version 1.0. NASA Ames Research Center - NASA Langley Research Center, 1999.

[2] K. Bilimoria. A geometric optimization approach to aircraft conflict resolution. In *Guidance, Navigation, and Control Conference*, volume AIAA 2000-4265, Denver, CO, August 2000.

[3] K. Bilimoria, B. Sridhar, G. Chatterji, K. Sheth, and S. Grabbe. FACET: Future ATM concepts evaluation tool. In *3rd USA/Europe Air Traffic Management R&D Seminar*, Naples, Italy, June 2000.

[4] V. Carreño and C. Muñoz. Aircraft trajectory modeling and alerting algorithm verification. In J. Harrison and M. Aagaard, editors, *Theorem Proving in Higher Order Logics: 13th International Conference, TPHOLs 2000*, volume 1869 of *Lecture Notes in Computer Science*, pages 90–105. Springer-Verlag, 2000. An earlier version appears as report NASA/CR-2000-210097 ICASE No. 2000-16.

[5] G. Dowek, C. Muñoz, and A. Geser. Tactical conflict detection and resolution in a 3-D airspace. Technical Report NASA/CR-2001-210853 ICASE Report No. 2001-7, ICASE-

NASA Langley, ICASE Mail Stop 132C, NASA Langley Research Center, Hampton VA 23681-2199, USA, April 2001.

[6] M.S. Eby. A self-organizational approach for resolving air traffic conflicts. *Lincoln Laboratory Journal*, 7(2), 1994.

[7] J. Hoekstra, R. Ruigrok, R. van Gent, J. Visser, B. Gijsbers, M. Valenti, W. Heesbeen, B. Hilburn, J. Groeneweg, and F. Bussink. Overview of NLR free flight project 1997-1999. Technical Report NLR-CR-2000-227, National Aerospace Laboratory (NLR), May 2000.

[8] J. Kuchar and L. Yang. Survey of conflict detection and resolution modeling methods. In *AIAA Guidance, Navigation, and Control Conference*, volume AIAA-97-3732, pages 1388–1397, New Orleans, LA, August 1997.

[9] J.A. McDonald and R.A. Vivona. Strategic airborne conflict detection of air traffic and area hazards. Technical Report NASA Contract: NAS2-98005 RTO-29, TITAN Systems Corporation, SRC Division, November 2000.

[10] C. Muñoz, R.W. Butler, V. Carreño, and G. Dowek. On the verification of conflict detection algorithms. Technical Report NASA/TM-2001-210864, NASA Langley Research Center, NASA LaRC,Hampton VA 23681-2199, USA, May 2001.

[11] S. Owre, J. M. Rushby, and N. Shankar. PVS: A prototype verification system. In Deepak Kapur, editor, *11th International Conference on Automated Deduction (CADE)*, volume 607 of *Lecture Notes in Artificial Intelligence*, pages 748–752, Saratoga, NY, June 1992. Springer-Verlag.

[12] Radio Technical Commission for Aeronautics. Final report of the RTCA board of directors' select committee on free flight. Technical Report Issued 1-18-95, RTCA, Washington, DC, 1995.

Dr. Gilles Dowek is the head of the Logical Project at INRIA (France). He is interested in the formalization of mathematics, in proof processing systems, and in the applications to the design of programming languages. He has been involved the organization of several conferences, including LICS, TPHOLs, and TLCA. He is a member of the steering committee the European working group TYPES and he is teaching at the École Polytechnique. In 2001, he has been visiting the Formal Methods group at ICASE-NASA Langley research in Hampton (Virginia).

Dr. Alfons Geser is interested in inductive theorem proving, term rewriting, and model checking, and their practical application. He worked as a research assistant at the Universities of Ulm, Passau, and Tuebingen, and as an ASIC Design Engineer at a hardware design company in Passau, Germany. He joined ICASE as a Senior Staff Scientist in Jan 2001. Currently he is involved in several NASA Langley formal verification projects.

Dr. César Muñoz received his Ph.D. in Computer Science from the University of Paris 7 in 1997. After completing his Ph.D., he spent one and a half years as an International Fellow in the Formal Methods Group of the the Computer Science Laboratory at SRI International in Menlo Park. He joined ICASE - NASA Langley as a Staff Scientist in May 1999. His research work focuses on the application of higher order logic to formal specification and verification of digital systems that are of interest to NASA. Currently, he is coordinating the formal verification group at ICASE, whose main objective is the formal safety analysis of Air Traffic Management Systems.