ATIO-2007-7737
7th AIAA Aviation Technology, Integration and Operations Conference, 18–20 September 2007, Belfast, Northern Ireland

Conflict Detection and Resolution for 1,2,...,N Aircraft

Gilles Dowek* César Muñoz†

This paper presents a mathematical framework for the formal specification of aircraft conflict detection and resolution algorithms and the verification of their properties. The framework is illustrated with original algorithms that detect conflicts with multiple aircraft and effectively solve all them via vertical speed-only maneuvers.

I. Introduction

Formal methods in computer science refers to the use of logic and mathematics to verify that a system design and its implementation satisfy functional requirements and safety properties.[1] Despite the fact that several Conflict Detection and Resolution (CD&R) systems have been proposed in the past few years,[2] very few of these systems have been described and analyzed using formal methods. Therefore, it is not always clear what properties these algorithms satisfy. For instance, the notion of *cooperative resolution* is often used in the CD&R literature expressing different and, sometimes, contradictory concepts, e.g., it expresses exchange of intent information,[3–6] degree of coordination,[2,7] sharing of effort to achieve separation,[8] and temporary delegation of responsibility for separation.[9] This confusion could be avoided by providing rigorous definitions of the concepts and properties that are fundamental to all CD&R systems.

This paper presents a mathematical framework for the formal specification and analysis of conflict detection and resolution algorithms and their properties. This framework applies to *state-based*, *pairwise*, *geometric*, and *distributed* conflict detection and resolution algorithms. State-based (or *tactical*) refers to the use of aircraft state information, e.g., position and velocity vectors, as opposed to strategic approaches that use intent information, e.g., flight plan. Pairwise refers the use of a conflict detection and resolution logic for two distinguished aircraft: the *ownship* and the *traffic* aircraft. Geometric refers to the use of linear projections to predict aircraft trajectories as opposed to probabilistic or performance-based trajectories. Finally, distributed refers to systems that are deployed on several aircraft detecting and solving conflicts for the ownship as opposed to centralized systems that detect and simultaneously solve conflicts for a large set of aircraft. The distributed approach that we consider in this framework requires minimal communication between the aircraft. In particular, the only information that is periodically exchanged between the aircraft is the state information, e.g., position and velocity vectors of the aircraft. Examples of algorithms and approaches that are included in this framework are the self-organizational approach,[10] the modified potential algorithm,[7] the geometric optimization approach,[8] and the KB3D algorithm and its extensions.[11–14]

This paper is organized as follows. Section II provides the geometric background for state-based conflict detection and resolution. The CD&R framework is presented in Section III, for conflict detection, and Section IV, for conflict resolution. These sections illustrate the use of the framework with original conflict detection and resolution algorithms for multiple aircraft. The mathematical development presented in this paper has been formally verified in the Program Verification System (PVS).[15]

II. Geometric Background

We assume a flat earth geometry where position and velocity vectors are given in Cartesian coordinates. Aircraft dynamics is represented by a point moving at constant speed along a linear trajectory. Therefore,

*Professor, École polytechnique, 91128 Palaiseau Cedex, France.
†Senior Staff Scientist, National Institute of Aerospace, Hampton, VA 23666, USA.

aircraft trajectories can be described by a position, a velocity vector, and a time interval. We also assume that in order to solve a conflict an aircraft can instantaneously change course and speed. These assumptions are typical of tactical CD&R systems with short lookahead times (usually, 5 minutes).

The state information of an aircraft is represented by an identifier, a position, and a velocity vector. Identifiers are assumed to be unique and invariant, i.e., (1) there are no two different aircraft in the airspace with the same identifier, and (2) the identifier of an aircraft does not change over time. By abuse of notation, we will usually refer to an aircraft by its state. Henceforth, we use letters $A, B, \ldots$ to denote aircraft states, **boldface** letters to denote vector variables, e.g., $\mathbf{s}$ denotes position and $\mathbf{v}$ denotes velocity, and `typewriter` type for algorithms. When a vector variable refers to a particular aircraft, the variable is sub-indicated by the aircraft state. For instance the current position and velocity vectors of an aircraft A are denoted $\mathbf{s}_A$ and $\mathbf{v}_A$, respectively. As usual, sub-indices x, y, and z denote axis values, e.g., v_{Ax}, v_{Ay} are the horizontal components of the velocity vector $\mathbf{v}_A$.

Each aircraft A in the airspace is surrounded by an imaginary region called the *protected zone*. The protected zone defines a minimum safe separation distance between aircraft. In a 2-dimensional geometry, the protected zone is usually defined as a circle of diameter D around $\mathbf{s}_A$, i.e., the set points (x, y) that satisfy

$$\sqrt{(x - s_{Ax})^2 + (y - s_{Ay})^2} \quad < \quad \frac{D}{2}.$$

In a 3-dimensional geometry, a spherical protected zone can defined as the set of points (x, y, z) that satisfy

$$\sqrt{(x - s_{Ax})^2 + (y - s_{Ay})^2 + (z - s_{Az})^2} \quad < \quad \frac{D}{2}.$$

However, the preferred 3-dimensional shape of the protected zone is a flat cylinder of diameter D and height H. In this case, the cylindrical volume around an aircraft A is defined as the set of points (x, y, z) that satisfy

$$\begin{aligned} \sqrt{(x - s_{Ax})^2 + (y - s_{Ay})^2} \quad &< \quad \frac{D}{2}, \text{ and} \\ |z - s_{Az}| \quad &< \quad \frac{H}{2}. \end{aligned}$$

All these concrete definitions of protected zone are particular instances of a more general abstract definition.

Definition 1 (Protected Zone) *The protected zone of an aircraft A is defined as the set of points P_A, such that*

$$P_A \quad = \quad \{ \; \mathbf{x} \quad | \quad \|\mathbf{s}_A - \mathbf{x}\| < \frac{1}{2} \; \}, \tag{1}$$

where $\|.\|$ is a vector norm*, i.e., an operation from vectors to real number that satisfies:*

1. *Positivity:* $\|\mathbf{x}\| \geq 0$.
2. *Nullity:* $\|\mathbf{x}\| = 0$ *if and only if* $\mathbf{x} = \mathbf{0}$.
3. *Scalability:* $\|k\mathbf{x}\| = |k|\|\mathbf{x}\|$.
4. *Triangular inequality:* $\|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|$.

If the protected zone is a circle or a sphere of diameter D, the norm of a vector $\mathbf{x}$ is defined as the Euclidean norm of the vector $\mathbf{x}$ divided by D:

$$\|(x, y, z)\| \quad = \quad \frac{\sqrt{x^2 + y^2}}{D}, \text{ in a 2-dimensional geometry, and} \tag{2}$$

$$\|(x, y, z)\| \quad = \quad \frac{\sqrt{x^2 + y^2 + z^2}}{D}, \text{ in a 3-dimensional geometry.} \tag{3}$$

In the case of a cylinder of diameter D and height H, the norm is defined as

$$\|(x, y, z)\| \quad = \quad \max(\frac{\sqrt{x^2 + y^2}}{D}, \frac{|z|}{H}). \tag{4}$$

Proposition 1 *Formulas (2), (3), and (4) are well-defined vector norms, i.e., they satisfy positivity, nullity, scalability, and triangular inequality.*

The use of a norm allows for definitions of aircraft distance and loss of separation that are independent of the shape of the protected zone.

Definition 2 (Distance) *The* distance *between aircraft A and B is defined as*

$$\Delta(A, B) \quad = \quad \|\mathbf{s}_A - \mathbf{s}_B\|. \tag{5}$$

Definition 3 (Loss of Separation) *A* loss of separation *or* violation *occurs when the protected zones of two aircraft overlap, e.g.,*

$$P_A \cap P_B \quad \neq \quad \emptyset. \tag{6}$$

The notions of loss of separation and aircraft distance are related by the following proposition.

Proposition 2 *Aircraft A and B are in loss of separation if and only if*

$$\Delta(A, B) \quad < \quad 1.$$

The linear trajectory of an aircraft is predicted from the current state of the aircraft, where the velocity of the aircraft is assumed to be constant.

Definition 4 (Predicted State) *The* predicted state *of an aircraft A at future time t relative to a current time 0, denoted $A(t)$, satisfies:*

$$\begin{aligned} \mathbf{s}_{A(t)} &= \mathbf{s}_A + t\,\mathbf{v}_A, \\ \mathbf{v}_{A(t)} &= \mathbf{v}_A. \end{aligned}$$

A conflict is a predicted loss of separation between two aircraft within a *lookahead time T*.

Definition 5 (Conflict) *Aircraft A and B are said to be in* conflict *if there is a time t in the future but before the lookahead time, i.e., $0 \leq t \leq T$, such that the distance between the aircraft at time t is strictly less than 1:*

$$\Delta(A(t), B(t)) \quad < \quad 1. \tag{7}$$

Conflicts are usually checked by comparing the predicted closest separation of two aircraft against the minimum safe distance defined by the protected zone.

Definition 6 (Time of Closest Separation) *The* time of closest separation *between aircraft A and B within a lookahead time T, denoted $\tau_T(A, B)$, is the smallest time τ, $0 \leq \tau \leq T$, such that for any time t, $0 \leq t \leq T$:*

$$\Delta(A(\tau), B(\tau)) \quad \leq \quad \Delta(A(t), B(t)). \tag{8}$$

Note that the time of closest separation always exists because Δ is a continuous function and the time range $0 \leq t \leq T$ defines a close interval.

Definition 7 (Distance of Closest Separation) *The* distance of closest separation *between two aircraft A and B, within a lookahead time T, is the distance at the time of closest separation, i.e.,*

$$\delta_T(A, B) \quad = \quad \Delta(A(\tau_T(A, B)), B(\tau_T(A, B))). \tag{9}$$

The following proposition provides an alternative definition of conflict based on distance of closest separation.

Proposition 3 *Aircraft A and B are in* conflict *if and only if $\delta_T(A, B) < 1$.*

In our framework, a resolution maneuver is modeled as an instantaneous change of the velocity vector of the current aircraft state.

Definition 8 (Resolution Maneuver) *A* resolution maneuver $\mathcal{R}_A$ *is a new state for an aircraft A that (only) modifies the value of the current velocity vector. The new velocity vector for aircraft A under maneuver* $\mathcal{R}_A$ *is denoted* $\mathbf{v}_{\mathcal{R}_A}$.

Resolution maneuvers can be classified according to the parameters of the velocity vector that they modify. Assuming a 3-dimensional geometry:

- $\mathcal{R}_A$ is a *track-only* maneuver if ${v_{\mathcal{R}_A x}}^2 + {v_{\mathcal{R}_A y}}^2 = {v_{Ax}}^2 + {v_{Ay}}^2$ and $v_{\mathcal{R}_A z} = v_{Az}$.
- $\mathcal{R}_A$ is a *ground speed-only* maneuver if $v_{\mathcal{R}_A z} = v_{Az}$ and there is a positive k such that $(v_{\mathcal{R}_A x}, v_{\mathcal{R}_A y}) = k(v_{Ax}, v_{Ay})$.
- $\mathcal{R}_A$ is a *vertical speed-only* maneuver if $v_{\mathcal{R}_A x} = v_{Ax}$ and $v_{\mathcal{R}_A y} = v_{Ay}$.

Resolution maneuvers can also be classified according to the separation criteria that they satisfy. Let $\mathcal{R}_A$ and $\mathcal{R}_B$ be resolution maneuvers for aircraft A and B,

- $\mathcal{R}_A$ is an *independent resolution* for aircraft A (with respect to aircraft B) if $\delta_T(\mathcal{R}_A, B) \geq 1$. If $\delta_T(\mathcal{R}_A, B) = 1$, $\mathcal{R}_A$ is also *tangential*.
- $\mathcal{R}_A$ and $\mathcal{R}_B$ are *coordinated* if $\delta_T(\mathcal{R}_A, \mathcal{R}_B) \geq 1$. If $\delta_T(\mathcal{R}_A, \mathcal{R}_B) = 1$, they are also *minimal*.
- $\mathcal{R}_A$ and $\mathcal{R}_B$ are *repulsive* if $\delta_T(\mathcal{R}_A, \mathcal{R}_B) > \delta_T(A, B)$.

III. Conflict Detection

This section presents our mathematical framework for the specification of conflict detection algorithms and their safety properties. It illustrates the use of the framework with a three dimensional conflict detection algorithm for multiple aircraft.

III.A. Pairwise Conflict Detection

A *(pairwise) conflict detection* algorithm takes as parameters the current state of two aircraft, and returns a Boolean value that indicates a predicted loss of separation between them.

Definition 9 (Conflict Detection Correctness and Completeness) *Given a lookahead time T, a conflict detection algorithm, namely* `cd`*, is* correct *if only conflicts are detected, i.e., for all aircraft states A and B, if* `cd`(A, B) *returns* `true` *then A and B are in conflict. The algorithm is* complete *if all conflicts are detected, i.e., if A and B are in conflict then* `cd`(A, B) *returns* `true`.

Note that according to the previous definition, an algorithm that always returns `true` is complete, but not correct, and an algorithm that always returns `false` if correct, but not complete.

Proposition 4 *If* `cd` *is correct and complete then* `cd` *is* symmetric*, i.e., for all aircraft states A and B,* `cd`(A, B) *if and only if* `cd`(B, A).

III.B. Conflict Detection for Multiple Aircraft

A *detection algorithm for multiple aircraft* $\vec{\texttt{cd}}$ takes as parameters the current state of the ownship A and a set of aircraft states α where $A \notin \alpha$, and returns a Boolean value that indicates a predicted loss of separation between A and any aircraft in α.

Definition 10 (Conflict Detection Correctness and Completeness for Multiple Aircraft) *Given a lookahead time T, a conflict detection algorithm for multiple aircraft, namely* $\vec{\texttt{cd}}$*, is* correct *if for all aircraft state A and set of aircraft states α, where $A \notin \alpha$, if* $\vec{\texttt{cd}}(A, \alpha)$ *returns* `true` *then A is in conflict with an aircraft $B \in \alpha$. The algorithm is* complete *if every time that $A \notin \alpha$ and $B \in \alpha$ are in conflict then* $\vec{\texttt{cd}}(A, \alpha)$ *returns* `true`.

Definition 11 *A pairwise conflict detection algorithm* `cd` *can be transformed into a conflict detection algorithm for multiple aircraft* $\vec{\texttt{cd}}$ *as follows:*

```
cd⃗(A,α): bool =
  if α = ∅ then
    false
  else let B ∈ α in
    cd(A,B) or cd⃗(A,α \ {B})
  endif
```

The algorithm $\vec{\texttt{cd}}$ is recursive. However, it is well-defined since the cardinality of the set of aircraft strictly decreases at each iteration.

Proposition 5 *If* `cd` *is a pairwise correct and complete conflict detection algorithm then* $\vec{\texttt{cd}}$ *is a correct and complete conflict detection algorithm for multiple aircraft.*

III.C. A 3-Dimensional Detection Algorithm for Multiple Aircraft

KB3D is a pairwise state-based CD&R system co-developed by the authors[11] and formally verified in the Program Verification System (PVS).[15] In this section, we illustrate our conflict detection framework with a simplified version of the conflict detection algorithm used by KB3D, which we call `cd3d`.

The protected zone in KB3D is cylinder of diameter D and height H. Therefore, the norm is defined as in Formula (4). The conflict detection logic considers the following cases:

- The relative horizontal velocity of the aircraft is zero, i.e., ${v_x}^2 + {v_y}^2 = 0$, and the aircraft have already lost horizontal separation, i.e., ${s_x}^2 + {s_y}^2 < D^2$. In this case, the aircraft are in conflict if they have already lost vertical separation, i.e., $|s_z| < H$, or if they will lose vertical separation within the lookahead time T, i.e., $v_z s_z < 0$ and $-H < \texttt{sign}(v_z)(Tv_z + s_z)$.

- The relative vertical speed of the aircraft is zero, i.e., $v_z = 0$, but the aircraft are horizontally separated. First, we compute the time interval of horizontal conflict $[t_{D_{in}}, t_{D_{out}}]$ from the quadratic equation on t:

$$(s_x + tv_x)^2 + (s_y + tv_y)^2 = D^2.$$

  In this case, there is a conflict if the aircraft have already lost vertical separation, i.e., $|s_z| < H$, and if the time interval of horizontal conflict is within the lookahead time T, i.e., $t_{D_{out}} > 0$ and $t_{D_{in}} < T$.

- In the general case, we compute the time interval of horizontal conflict $[t_{D_{in}}, t_{D_{out}}]$ as in the previous case, and the time interval of vertical conflict $[t_{H_{in}}, t_{H_{out}}]$ from the equation on t:

$$|s_z + tv_z| = H.$$

  From these intervals, we compute the time interval of conflict $[t_{in}, t_{out}]$, where

$$\begin{aligned} t_{in} &= \max(t_{D_{in}}, t_{H_{in}}), \\ t_{out} &= \min(t_{D_{out}}, t_{H_{out}}). \end{aligned}$$

  In this case, there is a conflict if this interval is within the lookahead time T, i.e., $t_{in} < t_{out}$ and $t_{out} > 0$ and $t_{in} < T$.

The pairwise algorithm `cd3d` is defined as follows.

```
cd3d(A,B): bool =
  let s = s_A − s_B in
  let v = v_A − v_B in
    if  v_x^2 + v_y^2 = 0 and s_x^2 + s_y^2 < D^2 then
        |s_z| < H or
        (v_z s_z < 0 and −H < sign(v_z) (T v_z + s_z))
    else
      let d = 2 s_x v_x s_y v_y + D^2 (v_x^2 + v_y^2) − (s_x^2 v_y^2 + s_y^2 v_x^2) in
```

```
  if d > 0 then
    let a = v_x^2 + v_y^2 in
    let b = s_x v_x + s_y v_y in
    let t_{D_in} = (−b − √d)/a in
    let t_{D_out} = (−b + √d)/a in
    if v_z = 0 then
      |s_z| < H and t_{D_out} > 0 and t_{D_in} < T
    else
      let t_{H_in} = (−sign(v_z) H − s_z)/v_z in
      let t_{H_out} = (sign(v_z) H − s_z)/v_z in
      let t_in = max(t_{D_in}, t_{H_in}) in
      let t_out = min(t_{D_out}, t_{H_out}) in
        t_in < t_out and t_out > 0 and t_in < T
    endif
  else false
  endif
endif
```

Proposition 6 *The conflict detection algorithm* `cd3d` *is correct and complete.*

The following theorem is a corollary of Propositions 5and 6.

Theorem 7 *The conflict detection algorithm for multiple aircraft* $\vec{\texttt{cd3d}}$*, from Definition 11, is correct and complete.*

IV. Conflict Resolution

This section presents our mathematical framework for the specification of conflict resolution algorithms and their safety properties. Based on this framework, we propose an original vertical-only resolution algorithm for multiple aircraft.

IV.A. Pairwise Conflict Resolution

A *(pairwise) conflict resolution* algorithm takes as parameters the current state of two aircraft, the ownship and the traffic aircraft, and returns a set of possible resolution maneuvers for the ownship. Henceforth, we assume that conflict resolution algorithms are only applied to *conflicting aircraft*, i.e., aircraft A and B that are in conflict within a lookahead time T: $\delta_T(A,B) < 1$, but not in violation: $\Delta(A,B) \geq 1$.

Completeness of a conflict resolution algorithm refers to the existence of at least one resolution maneuver for every pair of conflicting aircraft.

Definition 12 (Conflict Resolution Completeness) *A conflict resolution algorithm, namely* `cr`*, is* complete *if for all conflicting aircraft A and B,* $\texttt{cr}(A,B)$ *is non-empty.*

We define several notions of conflict resolution correctness depending on the number of aircraft that maneuver to resolve a predicted conflict. For one aircraft, the correctness property is called *independence*, for two or more aircraft is called *coordination*.

IV.B. Conflict Resolution for One Aircraft

A basic assumption in a distributed conflict resolution approach is that only one aircraft, e.g., the ownship, maneuvers. In this case, the traffic aircraft is assumed to keep its current trajectory.

Definition 13 (Independent Conflict Resolution) *A conflict resolution algorithm, namely* `cr`*, is* independent *if and only if all resolutions computed by the algorithm are independent for the ownship, i.e., for all conflicting aircraft A and B:*

$$\mathcal{R}_A \in \texttt{cr}(A,B) \quad \textit{implies} \quad \delta_T(\mathcal{R}_A, B) \geq 1.$$

Note that a conflict resolution algorithm that is independent guarantees a minimum safe separation distance between the ownship and traffic aircraft when only the ownship maneuvers.

Definition 14 (Tangential Resolution) *A conflict resolution algorithm* $\mathtt{cr}$ *is* tangential *if all resolutions computed by the algorithm are tangential, i.e., for all conflicting aircraft A and B:*

$$\mathcal{R}_A \in \mathtt{cr}(A,B) \quad \textit{implies} \quad \delta_T(\mathcal{R}_A, B) = 1.$$

The following property is a corollary of Definitions 13 and 14.

Proposition 8 *All tangential algorithms are independent.*

IV.C. Conflict Resolution for Two Aircraft

In contrast to independent conflict resolution algorithms, where the resolution effort relies on the ownship, a coordinated algorithm assumes that both the ownship and traffic aircraft potentially maneuver to avoid an impending conflict.

Definition 15 (Coordinated Conflict Resolution) *A conflict resolution algorithm* $\mathtt{cr}$ *is* coordinated *if all resolutions computed by the algorithm are pairwise coordinated, i.e., for all conflicting aircraft A and B:*

$$\mathcal{R}_A \in \mathtt{cr}(A,B) \textit{ and } \mathcal{R}_B \in \mathtt{cr}(B,A) \quad \textit{implies} \quad \delta_T(\mathcal{R}_A, \mathcal{R}_B) \geq 1.$$

Definition 16 (Minimal Resolution) *A conflict resolution algorithm* $\mathtt{cr}$ *is* minimal *if all resolutions computed by the algorithm are pairwise minimal, i.e., for all conflicting aircraft A and B:*

$$\mathcal{R}_A \in \mathtt{cr}(A,B) \textit{ and } \mathcal{R}_B \in \mathtt{cr}(B,A) \quad \textit{implies} \quad \delta_T(\mathcal{R}_A, \mathcal{R}_B) = 1.$$

The following property is a corollary of Definitions 15 and 16.

Proposition 9 *All minimal algorithms are coordinated.*

The concepts of independent and coordinated conflict resolution are orthogonal. Section V presents examples of resolution algorithms that are independent without being coordinated, coordinated without being independent, and coordinated and independent simultaneously. Furthermore, any independent conflict resolution algorithm can be transformed into a coordinated algorithm by using a priority relation between aircraft. If the priority relation is known to all aircraft, coordination can be achieved without a central authority. The priority relation could be as simple as an order over aircraft identifiers. It does not need to be constant under time. However, at a given time all aircraft have to use the same priority relationship.

Formally, a *priority relation* is a strict order over aircraft states. We write $A \prec B$ to denote that aircraft A has priority over aircraft B at current time.

Definition 17 *Let* $\mathtt{cr}$ *be a conflict resolution algorithm, the resolution algorithm* $\mathtt{cr}_\prec$ *is defined as follows.*

```
cr≺(A,B): set =
  if  A ≺ B  then
    {A}
  else
    cr(A,B)
  endif
```

Proposition 10 *If the pairwise conflict resolution* $\mathtt{cr}$ *is independent, then* $\mathtt{cr}_\prec$ *is coordinated.*

Definition 18 (Repulsive Resolution) *A conflict resolution algorithm* $\mathtt{cr}$ *is* repulsive *if all resolutions computed by the algorithm are pairwise repulsive, i.e., for all conflicting aircraft A and B:*

$$\mathcal{R}_A \in \mathtt{cr}(A,B) \textit{ and } \mathcal{R}_B \in \mathtt{cr}(B,A) \quad \textit{implies} \quad \delta_T(\mathcal{R}_A, \mathcal{R}_B) > \delta_T(A,B).$$

In contrast to coordinated resolutions, repulsive resolutions separate but not necessarily solve a conflict. However, the iteration of a repulsive algorithm is expected to eventually solve the conflict.

Definition 19 *Let* `cr` *be a conflict resolution algorithm, we define the algorithm* $\texttt{cr}^n$ *recursively on* $n \geq 0$*:*

```
cr^n(A,B): set =
  if  n = 0 then
    {A}
  else
    R := ∅;
    foreach R_A ∈ cr(A,B) do
      foreach R_B ∈ cr(B,A) do
        R := R ∪ cr^{n-1}(R_A,R_B)
      done
    done
    R
  endif
```

Despite the expectation, the iteration of a repulsive conflict resolution algorithm does not always yield a coordinated algorithm. This is not even the case if the iteration is performed infinitely many times. We provide sufficient conditions for finite and infinite convergence of repulsive algorithms into coordinated algorithms.

Proposition 11 (Definite Convergence) *Let* `cr` *be a repulsive conflict resolution algorithm and* ϵ *be a strictly positive constant, i.e.,* $\epsilon > 0$*. If for any pair of conflicting aircraft* A *and* B*, and resolution maneuvers* $\mathcal{R}_A \in \texttt{cr}(A,B)$ *and* $\mathcal{R}_B \in \texttt{cr}(B,A)$*, it holds that*

$$\delta_T(\mathcal{R}_A, \mathcal{R}_B) - \delta_T(A,B) \quad \geq \quad \epsilon,$$

then there exists a natural number N *such that* $\texttt{cr}^N$ *is a coordinated conflict resolution algorithm.*

Proposition 12 (Eventual Convergence) *Let* `cr` *be a repulsive conflict resolution algorithm and* γ *be a positive constant strictly less than* 1*, i.e.,* $0 \leq \gamma < 1$*. If for any pair of conflicting aircraft* A *and* B*, and resolution maneuvers* $\mathcal{R}_A \in \texttt{cr}(A,B)$ *and* $\mathcal{R}_B \in \texttt{cr}(B,A)$*, it holds that*

$$\gamma(1 - \delta_T(A,B)) + \delta_T(\mathcal{R}_A, \mathcal{R}_B) \quad \geq \quad 1,$$

then there exists a number $N \leq \infty$ *such that* $\texttt{cr}^N$ *is a coordinated conflict resolution algorithm.*

IV.D. Conflict Resolution for Multiple Aircraft

Pairwise conflict resolution algorithms are not, in general, sufficient to solve multiple aircraft conflicts. Indeed, if an aircraft A is in conflict with aircraft B and C, it may happen that none of the pairwise resolutions for A solves both conflicts simultaneously. Multiple aircraft conflicts require resolution algorithms that take into account the state of several aircraft at the same time.

A *resolution algorithm for multiple aircraft* $\vec{\texttt{cr}}$ takes as parameters the current state of the ownship A and a set of aircraft states α where $A \notin \alpha$, and returns a set of possible resolution maneuvers for the ownship.

Definition 20 (Separated and Solved Aircraft) *A set of aircraft states* α *is* separated *if aircraft in* α *are pairwise separated, i.e., for all* $A, B \in \alpha$*, where* A *and* B *are different aircraft,* $\Delta(A,B) \geq 1$ *holds. The set* α *is* solved *if aircraft are pairwise free of conflicts, i.e., for all* $A, B \in \alpha$*, where* A *and* B *are different aircraft,* $\delta_T(A,B) \geq 1$ *holds.*

Definition 21 (Independent Resolution for Multiple Aircraft) *A resolution algorithm for multiple aircraft* $\vec{\texttt{cr}}$ *is* independent for the ownship A and the set of aircraft α*, where* $A \notin \alpha$*, if for any resolution* $\mathcal{R}_A \in \vec{\texttt{cr}}(A, \alpha)$*, and for any aircraft* $B \in \alpha$*,* $\delta_T(\mathcal{R}_A, B) \geq 1$*.*

Definition 22 (Coordinated Resolution for Multiple Aircraft) *A resolution algorithm for multiple aircraft* $\vec{\texttt{cr}}$ *is* coordinated for N aircraft *if for any set of separated aircraft* α *containing at most* N *aircraft, all sets of the form* $\{\mathcal{R}_{A_0}, \ldots, \mathcal{R}_{A_m}\}$*, where* $\mathcal{R}_{A_i} \in \vec{\texttt{cr}}(A_i, \alpha \setminus \{A_i\})$*, are solved.*

We note that pairwise resolution algorithms are a special case of resolution algorithms for multiple aircraft where the set of aircraft is a singleton. Indeed, the pairwise coordinated algorithms are the coordinated algorithms for 2 aircraft.

As in the case of pairwise conflict resolution, resolution algorithms for multiple aircraft can be iterated.

Definition 23 *Let $\vec{cr}$ be a resolution algorithm for multiple aircraft, we define the algorithm $\vec{cr}^n$ by induction on $n \geq 0$ as follows.*

```
cr⃗^n(A,{A_0,...,A_m}): set =
  let α = {A_0,...,A_m} in
  if  n = 0 then
    {A}
  else
    R := ∅;
    foreach R_A ∈ cr⃗(A,α) do
      foreach R_A0 ∈ cr⃗(A_0, α \ A_0 ∪ {A}) ... R_Am ∈ cr⃗(A_m, α \ A_m ∪ {A}) do
        R := R ∪ cr⃗^(n-1)(R_A,{R_A0,...,R_Am})
      done
    done
    R
  endif
```

The rest of this section gives a sufficient condition that guarantees that the iteration of a resolution algorithm for multiple aircraft is coordinated.

Definition 24 ($\prec$-Segment) *Let $\prec$ be a priority relation that is stable by velocity changes. The $\prec$-segments of a set of aircraft states α are the downward close subsets of α. That is, β is a* $\prec$-segment *of α if and only if*

1. *$\beta \subseteq \alpha$, and*
2. *for all $A \in \alpha$ and $B \in \beta$, if $A \prec B$ then $A \in \beta$.*

The cardinality of the greatest solved $\prec$-segment of α is denoted $[\alpha]$.

Definition 25 (Progressive Resolution) *A resolution algorithm for multiple aircraft $\vec{cr}$ is* progressive *if it increases the size of solved $\prec$-segments, i.e., for all α and all sets of the form $\{\mathcal{R}_{A_0}, \ldots, \mathcal{R}_{A_m}\}$, where $\mathcal{R}_{A_i} \in \vec{cr}(A_i, \alpha \setminus \{A_i\})$, it holds that $[\alpha] < [\{\mathcal{R}_{A_0}, \ldots, \mathcal{R}_{A_m}\}]$.*

Proposition 13 *If the conflict resolution algorithm for multiple aircraft $\vec{cr}$ is* progressive*, then $\vec{cr}^N$ is coordinated for N aircraft.*

IV.E. A Vertical Resolution Algorithm for Multiple Aircraft

Given a conflict between the ownship and a traffic aircraft, KB3D outputs a set of maneuvers for the ownship that modify only one parameter of the aircraft, e.g., ground speed-only, ground track-only, and vertical velocity-only. It has been formally shown that KB3D supports independent and coordinated resolutions.[13] The algorithm has also been extended with recovery maneuvers that satisfy time arrival constraints.[16] This section presents a simple coordinated algorithm for N aircraft, based on the vertical resolutions computed by KB3D.

Let $\prec$ be the priority relation defined by $A \prec B$ if and only if

$$s_{Az} < s_{Bz} \text{ or } (s_{Az} = s_{Bz} \text{ and } s_{Ax} < s_{Bx}) \text{ or } (s_{Az} = s_{Bz} \text{ and } s_{Ax} = s_{Bx} \text{ and } s_{Ay} < s_{By}).$$

As two aircraft cannot be at the same time at the same position, if A and B are different aircraft then either $A \prec B$ or $B \prec A$.

Consider two conflicting aircraft: ownship and traffic, denoted respectively by O and I. We assume that $I \prec O$, i.e., the traffic aircraft is lower than the ownship. In other words, we assume that the ownship aircraft maneuvers because the traffic aircraft has the right of way.

A tangential independent vertical-only resolution maneuver $\mathcal{R}_O$ for the ownship satisfies

$$\begin{aligned}\delta_T(\mathcal{R}_O, I) &= 1,\\ v_{\mathcal{R}_O x} &= v_{Ox}, \text{ and}\\ v_{\mathcal{R}_O y} &= v_{Oy}.\end{aligned}$$

Therefore,

$$\mathbf{s}_{\mathcal{R}_O(t)} = (s_{Ox} + tv_{Ox}, s_{Oy} + tv_{Oy}, s_{\mathcal{R}_O z} + tv_{\mathcal{R}_O z}) \quad \text{for all } t. \tag{10}$$

From Formulas (9), $\Delta(\mathcal{R}_O(\tau), I(\tau)) = 1$, where τ is the time of closest separation for aircraft states $\mathcal{R}_O$ and I. Let $\mathbf{s} = \mathbf{s}_O - \mathbf{s}_I$, $\mathbf{v} = \mathbf{v}_O - \mathbf{v}_I$, $\mathbf{v}' = \mathbf{v}_{\mathcal{R}_O} - \mathbf{v}_I$,

$$\begin{aligned}\Delta(\mathcal{R}_O(\tau), I(\tau)) &= 1\\ \|\mathbf{s}_{I(\tau)} - \mathbf{s}_{\mathcal{R}_O(\tau)}\| &= 1, \text{ from Formula (5)}\\ \max(\frac{\sqrt{s_x^2(\tau) + s_y^2(\tau)}}{D}, \frac{|s_z + \tau v'_z|}{H}) &= 1, \text{ from Formulas (4) and (10)}\\ \max(\frac{\sqrt{(s_x + \tau v_x)^2 + (s_y + \tau v_y)^2}}{D}, \frac{|s_z + \tau v'_z|}{H}) &= 1.\end{aligned}$$

Thus, both expressions $\frac{\sqrt{(s_x+\tau v_x)^2+(s_y+\tau v_y)^2}}{D}$ and $\frac{|s_z+\tau v'_z|}{H}$ are less than or equal to 1 and one of them is equal to 1. Studying the variation of these expressions in the neighborhood of τ we can prove that actually both expressions are equal to 1. Therefore, the solutions touch the top or the bottom circle of the protected zone, i.e.,

$$\frac{\sqrt{(s_x + \tau v_x)^2 + (s_y + \tau v_y)^2}}{D} = 1, \text{ and} \tag{11}$$

$$\frac{|s_z + \tau v'_z|}{H} = 1. \tag{12}$$

Formula (11) is a quadratic equation with two possible solutions for τ:

$$\tau^{\pm} = \frac{-s_x v_x - s_y v_y \pm \sqrt{\texttt{discr}(s_x, s_y, v_x, v_y)}}{v_x^2 + v_y^2},$$

where

$$\texttt{discr}(s_x, s_y, v_x, v_y) = D^2(v_x^2 + v_y^2) - (s_x v_y - s_y v_x)^2.$$

Since the aircraft are in conflict but not in violation, there is a time t where the aircraft lose horizontal separation, i.e.,

$$\frac{\sqrt{(s_x + tv_x)^2 + (s_y + tv_y)^2}}{D} < 1.$$

Thus, $\texttt{discr}(s_x, s_y, v_x, v_y) > 0$. If the relative ground speed of the aircraft is non-zero, i.e., $v_x^2 + v_y^2 \neq 0$, the two solutions for τ exists and $\tau^- < \tau^+$. Consider the solution for τ that touches the top circle of the protected zone. Figures 1 and 2 illustrate the cases $s_z \geq H$ and $s_z < H$, respectively. The time τ that yields a vertical solution is given by

$$\tau = \begin{cases} \tau^+ & \text{if } s_z \geq H,\\ \tau- & \text{otherwise.}\end{cases}$$

When the relative ground speed of the aircraft is non-zero, Formula (12) yields the following vertical resolution maneuver that always touches the top circle of the protected zone:

$$v_{\mathcal{R}_O z} = v_{Iz} + \frac{H - s_z}{\tau} \quad \text{if } v_x^2 + v_y^2 \neq 0. \tag{13}$$

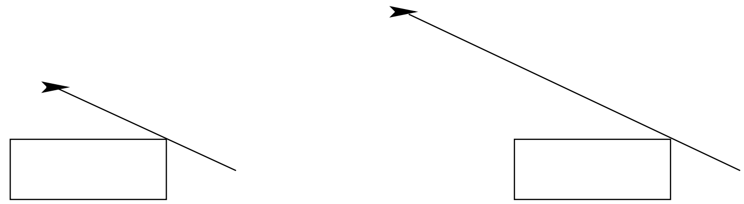

Figure 1. $s_z \geq H$

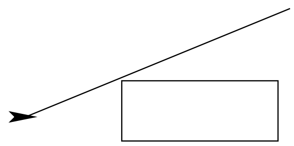

Figure 2. $s_z < H$

In the case where the relative ground speed is zero, a possible vertical solution maneuver is:

$$v_{\mathcal{R}_O z} \quad = \quad v_{Iz} \quad \text{if } v_x^2 + v_y^2 = 0. \tag{14}$$

The vertical speed-only resolution algorithm kv3d is defined by Formulas (13) and (14) as follows.

```
kv3d(O,I): set =
  let s = s_O − s_I in
  let v = v_O − v_I in
  let v_{R_O z} =
    if v_x^2 + v_y^2 = 0 then
      v_{Iz}
    else
      v_{Iz} + (H − s_z)/τ
    endif in
  {(v_x, v_y, v_{R_O z})}
```

Proposition 14 *The pairwise algorithm kv3d is a tangential independent vertical speed-only resolution algorithm.*

Finally, we define the resolution algorithm for multiple aircraft $\vec{\text{kv3d}}$ as follows.

```
kv3d⃗(O,α): set =
  let β = {A | A ≺ O and δ_T(O,A) < 1} in
  if β = ∅ then
    {O}
  else
    let I ∈ β in
    let O′ = kv3d(O,I) in
    kv3d⃗(O′, α \ {I})
  endif
```

The algorithm $\vec{\text{kv3d}}$ is recursive. However, it is well-defined since the cardinality of the set of aircraft strictly decreases at each iteration. Moreover, the algorithm is non-deterministic as an arbitrary aircraft I is chosen at each iteration. Once this conflict is solved, solving a conflict with another aircraft will only increase the vertical speed further. Therefore, when the recursion terminates the ownship has solved all conflicts with the aircraft below it. From this, it can be proved that $\vec{\text{kv3d}}$ is progressive.

Proposition 15

1. $\vec{kv3d}$ *is independent for the ownship and the set of aircraft that are below it.*

2. $\vec{kv3d}$ *is progressive.*

The following theorem is a corollary of Propositions 13 and 15.

Theorem 16 *The conflict resolution algorithm for multiple aircraft* $\vec{kv3d}^N$*, from Definition 23, is coordinated for* N *aircraft.*

V. Conclusion

We have presented a framework, i.e., a set of mathematical definitions, that enables the characterization of state-based conflict detection and resolution algorithms for different assumptions on the number of aircraft that maneuver to resolve a conflict. Furthermore, we identify sufficient conditions that guarantee that these algorithms satisfy properties such as correctness and completeness. The framework is illustrated with examples of original conflict detection and conflict resolution algorithms for multiple aircraft that are complete and correct.

The characterization of CD&R algorithms that we propose in this work, complements less rigorous, but more general, characterizations as the one proposed by Kuchar and Yang.[2,17] For instance, Table 1 classifies the following pairwise state-based conflict resolution algorithms according to the properties they satisfy.

- VP: Self-organizational voltage potential algorithm.[10]
- MVP: Modified voltage potential algorithm.[7]
- GO: Geometric optimization algorithm, "non-cooperative" setting.[8]
- GO-coop: Geometric optimization algorithm, "cooperative" setting.[8]
- KB3D: Original version.[11]
- KB3D-coord: Coordinated version.[13]

Property	**VP**	**MVP**	**GO**	**GO-coop**	**KB3D**	**KB3D-coord**
Completeness					$\surd$	$\surd$
Independence		$\times$	$\surd$	$\times$	$\surd$	$\surd$
Tangentiality		$\times$	$\surd$	$\times$	$\surd$	$\surd$
Coordination		$\times$	$\times$	$\surd$	$\times$	$\surd$
Minimality		$\times$	$\times$	$\surd$	$\times$	$\times$
Repulsion	$\surd$	$\surd$	$\times$	$\surd$	$\times$	$\surd$

Table 1. Properties of CD&R algorithms

The check symbol ($\surd$) represents properties that are claimed to be satisfied by the developers of the algorithms, while the cross symbol ($\times$) represents properties that are known not to hold. Blanks represent properties whose satisfiability is unknown to the authors of this paper.

Finally, we stress that the mathematical development presented in this paper has been formally specified and verified in a mechanical theorem prover. Given the critical nature of CD&R systems, we believe that the use of formal methods is an essential step in the analysis, design, and implementation of the new generation of distributed air traffic management concepts.

References

[1]Monin, J.-F., *Understanding Formal Methods*, FACIT (Formal Approaches to Computing and Information Technology), Springer-Verlag, London, UK, 2003.

[2]Kuchar, J. and Yang, L., "A Review of Conflict Detection and Resolution Modeling Methods," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 1, No. 4, December 2000, pp. 179–189.

[3]Kosecka, J., Tomlin, C., Pappas, G., and Sastry, S., "Generation of conflict resolution maneuvers for air traffic management," *International Conference on Intelligent Robots and Systems*, Grenoble, 1997.

[4]Tomlin, C., Pappas, G., and Sastry, S., "Noncooperative Conflict Resolution," *IEEE International Conference on Decision and Control*, San Diego, CA, December 1997.

[5]Bicchi, A. and Pallottino, L., "On Optimal Cooperative Conflict Resolution for Air Traffic Management Systems," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 1, No. 4, December 2000, pp. 221–232.

[6]Schouwenaars, T., How, J., and Feron, E., "Decentralized Cooperative Trajectory Planning of Multiple Aircraft with Hard Safety Guarantees," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Providence, Rhode Island, August 2004.

[7]Hoekstra, J., Ruigrok, R., van Gent, R., Visser, J., Gijsbers, B., Valenti, M., Heesbeen, W., Hilburn, B., Groeneweg, J., and Bussink, F., "Overview of NLR Free Flight Project 1997-1999," Tech. Rep. NLR-CR-2000-227, National Aerospace Laboratory (NLR), May 2000.

[8]Bilimoria, K., "A Geometric Optimization Approach to Aircraft Conflict Resolution," *Guidance, Navigation, and Control Conference*, Vol. AIAA 2000-4265, Denver, CO, August 2000.

[9]International Civil Aviation Organization, *Eleventh Air Navigation Conference Report*, Montreal, 22 September – 3 October 2003.

[10]Eby, M., "A Self-Organizational Approach for Resolving Air Traffic Conflicts," *Lincoln Laboratory Journal*, Vol. 7, No. 2, 1994, pp. 239–254.

[11]Dowek, G., Muñoz, C., and Geser, A., "Tactical Conflict Detection and Resolution in a 3-D Airspace," Tech. Rep. 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.

[12]Geser, A., Muñoz, C., Dowek, G., and Kirchner, F., "Air Traffic Conflict Resolution and Recovery," Tech. Rep. ICASE Report No. 2002-12 NASA/CR-2002-211637, ICASE-NASA Langley, ICASE Mail Stop 132C, NASA Langley Research Center, Hampton VA 23681-2199, USA, May 2002.

[13]Dowek, G., Muñoz, C., and Carreño, V., "Provably Safe Coordinated Strategy for Distributed Conflict Resolution," *Proceedings of the AIAA Guidance Navigation, and Control Conference and Exhibit 2005, AIAA-2005-6047*, San Francisco, California, 2005.

[14]Galdino, A., Muñoz, C., and Ayala, M., "Formal Verification of an Optimal Air Traffic Conflict Resolution and Recovery Algorithm," *Proceedings of the 14th Workshop on Logic, Language, Information and Computation*, edited by D. Leivant and R. de Queiroz, Vol. 4576 of *Lecture Notes in Computer Science*, Springer-Verlag, Rio de Janeiro, Brazil, July 2007, pp. 177–188.

[15]Owre, S., Rushby, J. M., and Shankar, N., "PVS: A Prototype Verification System," *11th International Conference on Automated Deduction (CADE)*, edited by D. Kapur, Vol. 607 of *Lecture Notes in Artificial Intelligence*, Springer-Verlag, Saratoga, NY, June 1992, pp. 748–752.

[16]Maddalon, J., Butler, R., Geser, A., and Muñoz, C., "Formal Verification of a Conflict Resolution and Recovery Algorithm," Tech. Rep. NASA/TP-2004-213015, NASA Langley Research Center, NASA LaRC,Hampton VA 23681-2199, USA, April 2004.

[17]Kuchar, J. and Yang, L., "Survey of Conflict Detection and Resolution Modeling Methods," *AIAA Guidance, Navigation, and Control Conference*, Vol. AIAA-97-3732, New Orleans, LA, August 1997, pp. 1388–1397.