

A Mathematical Analysis of Conflict Prevention Information

Jeffrey M. Maddalon* Ricky W. Butler*

César A. Muñoz*

NASA Langley Research Center, Hampton, VA, USA

Gilles Dowek†

Ecole Polytechnique, 91128 Palaiseau Cedex, France.

In air traffic management, conflict prevention information refers to the guidance maneuvers, which if taken, ensure that an aircraft’s path is conflict-free. These guidance maneuvers take the form of changes to track angle or ground speed. Conflict prevention information may be assembled into prevention bands that advise the crew on maneuvers that should not be taken. Unlike conflict resolution systems, which presume that the aircraft already has a conflict, conflict prevention systems show conflicts for any maneuver, giving the pilot confidence that if a maneuver is made, then no near-term conflicts will result. Because near-term conflicts can lead to safety concerns, strong verification of information correctness is required. This paper presents a mathematical framework to analyze the correctness of algorithms that produce conflict prevention information incorporating an arbitrary number of traffic aircraft and with both a near-term and intermediate-term lookahead times. The framework is illustrated with a formally verified algorithm for 2-dimensional track angle prevention bands.

Nomenclature

\mathcal{G}_{amber}	set of track angle bands without a loss of separation within time T_{red} , but within time T_{amber}
\mathcal{G}_{green}	set of track angle bands without a loss of separation within time T_{amber}
\mathcal{G}_{red}	set of track angle bands with a loss of separation within time T_{red}
$\mathcal{G}_{<T}$	set of track angles with a loss of separation within time T
$\mathcal{G}_{<T}^{o,i}$	set of track angles with a loss of separation within T between the ownship and traffic aircraft
\mathbf{s}	ownship aircraft position relative to the traffic aircraft, $\mathbf{s} = \mathbf{s}_o - \mathbf{s}_i$
\mathbf{s}_i	initial position of the traffic aircraft
\mathbf{s}_o	initial position of the ownship aircraft
T_{amber}	intermediate-term lookahead time
T_{red}	near-term lookahead time
T	generic lookahead time
\mathbf{v}_i	initial velocity of the traffic aircraft
\mathbf{v}_o	initial velocity of the ownship aircraft
\mathbf{v}'_o	new velocity vector for the ownship. We seek this vector.

I. Introduction

Different types of information are used to help aircraft maintain separation standards. At the lowest level, information is needed to indicate if separation standards will be violated in the near future—called a *conflict*. Once a conflict is detected, then conflict resolution information may be used to create a new path

*Research Engineer, NASA Langley Research Center, m/s 130, Hampton, VA 23681, USA.

†Professor, Ecole Polytechnique, 91128 Palaiseau Cedex, France.

in which there is no conflict. Most future airspace concepts propose using computer algorithms to produce this information. Both conflict detection and resolution algorithms usually work in a pair-wise fashion: the ownship aircraft and one other aircraft. In situations where traffic density is low, this pair-wise assumption does not significantly impact operations. However, when traffic density is high, resolving one conflict may result in new near-term conflicts—called *secondary conflicts*. These secondary conflicts may be nearer (in time) than the original conflict being addressed; so, the safety of the aircraft depends on avoiding these conflicts. More generally, any time an aircraft maneuvers there is the potential to create new conflicts, which must be avoided both for safety and efficiency.

Information to avoid potential conflicts involves analyzing possible maneuvers of the aircraft. There are two basic approaches to tactical^a airborne conflict prevention: probing and bands. In the maneuver probing approach, the pilot or controller provides an individual maneuver, which is tested to ensure the proposed trajectory is conflict-free. In the bands approach, large groups of possible maneuvers are analyzed and the pilot is presented with ranges of track angles or ground speeds, which, if taken, will result in conflict-free trajectories. Alternatively, these ranges could represent avoidance or “don’t go” zones. These ranges of guidance maneuvers are referred to as *conflict-prevention information*. The National Aerospace Laboratory (NLR) refers to their conflict prevention capability as Predictive Airborne Separation Assurance System or Predictive ASAS.³ The NLR approach provides two sets of bands: near-term conflicts (within 3 minutes) are shown in red, while intermediate-term conflicts (within 5 minutes) are shown in amber as illustrated in figure 1. We do not directly analyze the NLR system, but we do use it to characterize input and output information from a typical conflict prevention system.

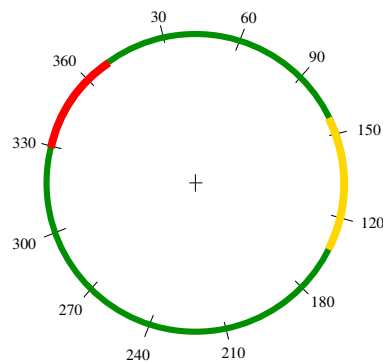


Figure 1. Compass Rose with Conflict Prevention Bands

Conflict prevention information may be presented directly to a pilot or controller or it may be supplied to another automated system. In either case, since aircraft safety may be threatened by incorrect conflict prevention information, a rigorous analysis of this information is needed. This paper provides this analysis. As with all such analysis, certain assumptions are made including ignoring the impact of aircraft performance, weather, or special use airspace; planned future work involves relaxing these assumptions. The correctness of this information is established through a mathematical characterization and analysis of the conflict-free regions for ground speed and track angles changes. The theory presented in this paper has been formalized and verified in the Prototype Verification System (PVS)⁶ and is publicly available.^b In this paper, we use standard mathematical notation make the paper accessible to a wider audience. To illustrate the general mathematical framework for the analysis of conflict prevention information, we present a 2-dimensional algorithm for track angle prevention bands. For completeness, a 2-dimensional algorithm for ground speed preventions bands is presented in the appendix. Both algorithms have been formally verified correct. We are currently working on candidate algorithms for 3-dimensional prevention bands.⁵

Although conflict-prevention systems have been used in several human-in-the-loop simulation experiments^{1,2,4} and their functionality has been described in other papers,^{3,7} we believe this is the first published analysis of such information. The primary focus of this work is conflict-prevention systems for airborne operation, but there is nothing inherent in this approach which precludes use in ground-based systems.

II. Problem Assessment

In this section, we analyze conflict prevention information to discover its essential elements, then we show how these essential elements may be analyzed mathematically for safety. Conflict prevention information consists of collections of guidance maneuvers displayed in one-dimensional ranges. These ranges define changes to the ownship’s velocity in either vertical speed, ground speed, or track angle. In the interest of space, this paper only describes changes to track angle in a 2-dimensional space; prevention bands for all the parameters in a 3-dimensional airspace are addressed in a related technical report.⁵

^aWe use the term *tactical* to mean a system which only uses near-term predictions of aircraft behavior without incorporating pilot or controller intent.

^b<http://research.nianet.org/fm-at-nia/ACCoRD>.

A. Colors of Track Angles

If an aircraft restricts its movement based on all aircraft within its surveillance range, then many relatively safe maneuvers will be unnecessarily avoided. Instead, only those aircraft that will cause near-term conflicts should be included in the computation of conflict prevention information.

We follow the approach from Predictive ASAS³ by introducing two parameters, T_{red} and T_{amber} , which divide the set of conflicts based on their nearness (in time) to a loss of separation. Predictive ASAS uses 3 minutes for T_{red} and 5 minutes for T_{amber} ; our analysis leaves these as parameters. Thus, the track-only guidance maneuvers may be partitioned into three sets of track angles, which are denoted: \mathcal{G}_{green} , \mathcal{G}_{amber} , \mathcal{G}_{red} . These “colors” of sets mean that all of the track angles within these sets maintain a specific property as follows: the green maneuvers mean that if the ownship flies one of these paths, then there will be no loss of separation within time T_{amber} ; the amber maneuvers mean that if the ownship flies one of these paths, then there will be no loss of separation before T_{red} , but there will be a loss of separation before T_{amber} ; finally, the red maneuvers mean that if the ownship flies one of these paths, then there will be a loss of separation before time T_{red} .

Suppose we had a way to determine the set track angles that have a loss of separation within time T , denoted $\mathcal{G}_{<T}$. Then since $T_{red} < T_{amber}$, we may define our sets of track angles in terms of this new set:

$$\begin{aligned}\mathcal{G}_{red} &= \mathcal{G}_{<T_{red}} \\ \mathcal{G}_{amber} &= \mathcal{G}_{<T_{amber}} - \mathcal{G}_{<T_{red}} \\ \mathcal{G}_{green} &= \{\alpha \mid 0^\circ \leq \alpha < 360^\circ\} - \mathcal{G}_{<T_{amber}}\end{aligned}$$

This observation simplifies the analysis, because now we only need to analyze one set, $\mathcal{G}_{<T}$.

B. Multiple Aircraft

We observe that each aircraft’s contribution to the set $\mathcal{G}_{<T}$ is independent of all other traffic; thus, the problem neatly divides into a series of aircraft pairs: the ownship and each traffic aircraft. If we use $\mathcal{G}_{<T}^{o,i}$ to represent the set of track angles which cause a loss of separation within time T between traffic aircraft i and the ownship o , then the set of track angles for all traffic is then be formed by

$$\mathcal{G}_{<T} = \bigcup_{i \in \text{traffic}} \mathcal{G}_{<T}^{o,i}$$

This observation simplifies the analysis again, because now we only need to find the track angles which cause a conflict between two aircraft, denoted by the set $\mathcal{G}_{<T}^{o,i}$. Before we examine the set $\mathcal{G}_{<T}^{o,i}$ in detail, we introduce some mathematical modeling concepts.

III. Modeling Considerations

The position and velocity vectors of the ownship are denoted by \mathbf{s}_o and \mathbf{v}_o , respectively, and the position and velocity vectors of the traffic aircraft are denoted by \mathbf{s}_i and \mathbf{v}_i , respectively. The components of each position and velocity vector are scalar values, so they are represented without the bold-face font, for example $\mathbf{s}_o = (s_{ox}, s_{oy})$. For notational convenience, for a given vector \mathbf{v} , we use $\mathbf{v}^2 = \mathbf{v} \cdot \mathbf{v}$, $\mathbf{v}^\perp = (v_y, -v_x)$, and $\mathbf{0} = (0, 0)$.

As typical of state-based approaches, speeds are presumed to be ground-relative. The use of ground speed was chosen to correspond to position and velocity reports coming from Automated Dependent Surveillance (ADS-B) systems. The impact of differences between ground speed and air speed is left for future work. As a simplifying assumption, we regard the position and velocity vectors as accurate and without error. We also assume the aircraft are moving with respect to the ground, i.e., $\mathbf{v}_o \neq \mathbf{0}$ and $\mathbf{v}_i \neq \mathbf{0}$. However, relative velocity vectors can be $\mathbf{0}$, e.g., when both aircraft fly parallel to each other at the same ground speed.

In a 2-dimensional airspace system, the separation criterion is specified as a minimum horizontal separation D (in much of the airspace D is 5 nautical miles). It is convenient to develop the theory using a translated coordinate system where the traffic aircraft is at the center of the coordinate system and does not move, and the ownship is located at $\mathbf{s}_o - \mathbf{s}_i$ and, until a maneuver is taken, the aircraft move relative to each other with velocity $\mathbf{v}_o - \mathbf{v}_i$. Henceforth, we denote the initial relative position of the aircraft by \mathbf{s} , i.e., $\mathbf{s} = \mathbf{s}_o - \mathbf{s}_i$.

For example in figure 2, the blue (upper) dot represents the ownship with its velocity vector and the magenta vector (lower) is the velocity vector of the traffic. In the translated coordinate system, these vectors combine to form a single relative vector, also shown in blue in figure 2. The separation criterion defines a disc of radius D around the traffic aircraft. This disc is called the *protected zone*.

In figure 2, the relative velocity vector (blue) defines a half-line that intersects the protected zone, meaning that in some future time the ownship will enter the protected zone around the traffic aircraft. If the intersection point occurs at a time less or equal than T the aircraft are in conflict. In this figure, the green vectors show possible resolution vectors.

Formally, a 2-dimensional conflict occurs if there exists a time $0 \leq t \leq T$ when the aircraft will lose separation, i.e., the positions $\mathbf{s}_o + t\mathbf{v}_o$ and $\mathbf{s}_i + t\mathbf{v}_i$ are within a horizontal distance D of each other. In the relative coordinate system, we define

Definition 1 (conflict?).

$$\text{conflict?}(\mathbf{s}, \mathbf{v}_o, \mathbf{v}_i) \equiv \exists 0 \leq t \leq T : (\mathbf{s} + t(\mathbf{v}_o - \mathbf{v}_i))^2 < D^2. \quad (1)$$

In this definition, we model future aircraft positions as a linear projection of the aircraft's velocity from its current position. An aircraft's accelerations—both positional and angular—are not modeled. The distance between two aircraft at time t is equal to $\|\mathbf{s} + t(\mathbf{v}_o - \mathbf{v}_i)\|$.

With the specification of a conflict in definition 1, we now develop a function to determine if a conflict exists. It is not difficult to prove that if $\mathbf{v}_o \neq \mathbf{v}_i$ the minimum separation between two aircraft occurs at the time of closest approach $\tau = -\mathbf{s} \cdot (\mathbf{v}_o - \mathbf{v}_i) / (\mathbf{v}_o - \mathbf{v}_i)^2$. Since conflicts are predicted for times $0 \leq t \leq T$, we have to consider the minimum distance between the aircraft during that interval of time. In particular, if $\tau \leq 0$ or if $\mathbf{v}_o = \mathbf{v}_i$, then the minimum separation between the aircraft is achieved at time 0. Similarly, if $\tau \geq T$, then the minimum separation between the aircraft occurs at time T . Therefore, the aircraft minimum separation during the time interval $[0, T]$ is defined as follows.

Definition 2 (Minimum distance function, Ω).

$$\Omega(\mathbf{s}, \mathbf{v}_o, \mathbf{v}_i) \equiv \begin{cases} \|\mathbf{s}\| & \text{if } \tau \leq 0 \text{ or } \mathbf{v}_o = \mathbf{v}_i, \\ \|\mathbf{s} + \min(\tau, T)(\mathbf{v}_o - \mathbf{v}_i)\| & \text{otherwise.} \end{cases}$$

We can use this minimum distance function to completely characterize the predicate `conflict?`.

Definition 3 (Conflict detection function, `cd2d`).

$$\text{cd2d}(\mathbf{s}, \mathbf{v}_o, \mathbf{v}_i) \equiv \Omega(\mathbf{s}, \mathbf{v}_o, \mathbf{v}_i)^2 < D^2.$$

Theorem 1 (Completeness of `cd2d`). `cd2d`($\mathbf{s}, \mathbf{v}_o, \mathbf{v}_i$) returns `true` if and only if `conflict?`($\mathbf{s}, \mathbf{v}_o, \mathbf{v}_i$) holds.

IV. Pair-wise Set of Track Angles

We now investigate $\mathcal{G}_{<T}^{o,i}$, the set of ownship track angles that result in a loss of separation with aircraft i within time T . The minimum separation as function of the ownship's track angle, denoted Ω_α , for a given relative position \mathbf{s} , traffic's velocity \mathbf{v}_i , and an ownship ground speed of $\|\mathbf{v}_o\|$ corresponds to

$$\Omega_\alpha \equiv \Omega(\mathbf{s}, \|\mathbf{v}_o\| \angle \alpha, \mathbf{v}_i) \quad (2)$$

where $m \angle \alpha = (m \sin \alpha, m \cos \alpha)$. It may appear that the sin and cos are reversed in this expression, but recall that α is a track angle defined as clockwise degrees from north, instead of the mathematical convention of counterclockwise from east.

If we plot the function Ω_α , then we will get a chart somewhat similar to figure 3. We observe that the points that cross the line labeled D (the horizontal separation minimum), are precisely the *transition* points between angles which should or should not be included in the set $\mathcal{G}_{<T}^{o,i}$. By figure 3, only those angles where the minimum distance is beneath the line D are in this set.

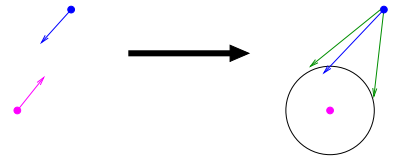


Figure 2. Translated Coordinate System

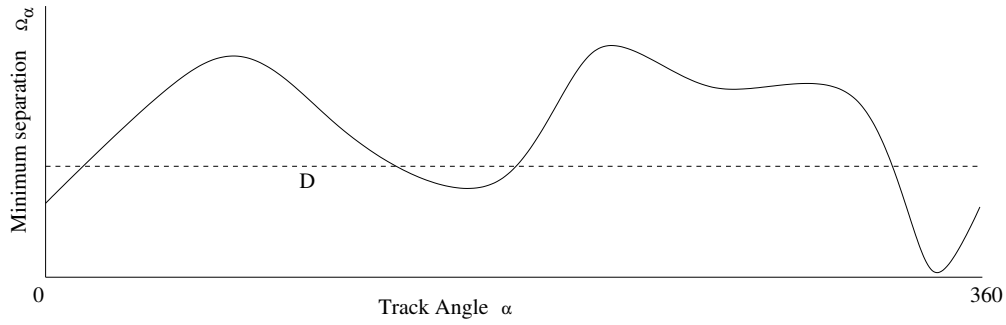


Figure 3. Example of Minimum Separation as a Function of Ownship Track Angle

Figure 3 implies that the function Ω_α is continuous and indeed it is; however the proof of this result is beyond the scope of this paper. The continuity of this function allows us to characterize intervals of track angles with a single angle. We observe this standard result from the mathematics of continuous functions.

Theorem 2 (Intermediate Value). *If f is continuous on the open interval (x_l, x_u) and $f(x_l) = 0$, $f(x_u) = 0$ and there are no other zeros between x_l and x_u , then f is either positive at every point in the interval or negative everywhere in the interval.*

By this theorem, all angles between the zeros of the distance function $\Omega_\alpha - D$ will all share the same conflict status. Since the transition points precisely correspond to the zeros of this function, we may select any angle between two consecutive transition points and the result of `cd2d` on that point determines the conflict status of *all* of the angles between the transition points.

Our challenge to define set $\mathcal{G}_{<T}^{o,i}$ now involves finding all the transition points. Henceforth, we consider that the aircraft are initially separated, i.e., $s^2 \geq D^2$. If this is not the case, then separation has already been lost; so, set $\mathcal{G}_{<T}^{o,i}$ is the whole range of track angles since no maneuver can avoid a preexisting loss of separation.

V. Transition Points

As described in the last section, our problem of analyzing the correctness of a conflict prevention bands algorithm comes down to finding all the transition track angles—the zeros in theorem 2. We note that an algorithm that finds a superset of transition points is still correct. However, for algorithmic efficiency we would like to ignore all non-transition points.

Finding all the zeros of the function $\Omega_\alpha - D$ has shown to be difficult as it requires complex trigonometric reasoning. Rather than following a trigonometric approach, we use an algebraic method where we first find all the vectors \mathbf{v}'_o such that

$$\mathbf{v}'_o{}^2 = \mathbf{v}_o{}^2, \quad (3)$$

$$\Omega_\alpha(\mathbf{s}, \mathbf{v}'_o, \mathbf{v}_i)^2 = D^2, \quad (4)$$

for given vectors $\mathbf{s}, \mathbf{v}_o, \mathbf{v}_i$. Equation (3) guarantees that \mathbf{v}'_o has the same norm, i.e., ground speed, as \mathbf{v}_o . This ensures that \mathbf{v}'_o is only a rotation of \mathbf{v}_o . Equation (4) ensures that \mathbf{v}'_o is a zero of $\Omega_\alpha - D$. Then we can get the transition angles by `track`(\mathbf{v}'_o) where `track` is the angle of \mathbf{v}'_o measured in clockwise degrees from north in the range $[0, 360)$.^c

Using theorem 1 and the definition of `conflict?`, equation (4) may be solved in two ways: (a) the half-line $\mathbf{s} + t(\mathbf{v}_o - \mathbf{v}_i)$ does not intersect the protected zone for any value of $t \geq 0$ or (b) it intersects the protected zone at a time $t \geq T$. The first case determines a set of transition points defined by vectors \mathbf{v}'_o such that the relative velocity vector $\mathbf{v}'_o - \mathbf{v}_i$ yields a tangent trajectory to the protected zone. These transition points are called *horizontal transition points*. The second case determines a set of a set of transition points defined by vectors \mathbf{v}'_o such that the relative vector $\mathbf{v}'_o - \mathbf{v}_i$ yields a trajectory that intersects the protected zone at exactly time T . These transition points are called *lookahead time transition points*. In the sections below, we discuss how to compute these two types of transition points.

^cWe assume that `track`($\mathbf{0}$) = 0.

A. Horizontal Transition Points

The horizontal transition points are those values of the ownship's track angle that produce a *relative* velocity vector that is tangent to the protected zone, as indicated by the green vectors in figure 2. The P-ASAS report³ describes an interesting situation where a single aircraft pair produces two prohibited regions. We have formally verified that there are at most four different angles that define a trajectory tangent to the protected zone.

Horizontal transition points are determined by vectors \mathbf{v}'_o such that the half-line $\mathbf{s} + t(\mathbf{v}'_o - \mathbf{v}_i)$, for $t \geq 0$, is tangent to the protected zone and $\mathbf{v}'_o{}^2 = \mathbf{v}_o^2$. In order to find such vectors \mathbf{v}'_o we proceed in several steps. First, we note that there at most two half-lines that are tangent to the protected zone. Then, for each one of these half-lines, we find a non-null vector \mathbf{u} that lies in the half-line and points in the same direction. This vector will simplify the analysis provided it is only determined by the geometry of the encounter—the aircraft velocities are not involved. The vector \mathbf{u} characterizes the half-line $k\mathbf{u}$, for $k \geq 0$. Therefore, the vectors \mathbf{v}'_o that determine the horizontal transition points are those that satisfy (3) and for some $k \geq 0$,

$$k\mathbf{u} = \mathbf{v}'_o - \mathbf{v}_i. \quad (5)$$

Thus, we can find a \mathbf{v}'_o provided we find the tangent vector \mathbf{u} and k . Finding these two quantities is the subject of the next two sections. The third section collects these results into a function.

1. Finding Tangents

Since $\mathbf{s}^2 \geq D^2$, we consider two cases: (a) $\mathbf{s}^2 = D^2$ and (b) $\mathbf{s}^2 > D^2$. In the first case, any vector that is perpendicular to \mathbf{s} lies in a half-line that is tangent to the protected zone. Therefore, we define the vectors $\mathbf{u}_\epsilon = \epsilon\mathbf{s}^\perp$, for $\epsilon = \pm 1$. In the second case, for each one the two half-lines, we define a point on the protected zone that is tangent on the line from \mathbf{s} . Since there are at most two half-lines that are tangent to the protected zone, there are at most two such points, which we call \mathbf{Q}_ϵ .

Definition 4 (\mathbf{Q}_ϵ).

$$\mathbf{Q}_\epsilon(\mathbf{s}) \equiv \frac{D^2}{\mathbf{s}^2}\mathbf{s} + \epsilon \frac{D\sqrt{\mathbf{s}^2 - D^2}}{\mathbf{s}^2}\mathbf{s}^\perp,$$

for $\epsilon \pm 1$.

Then, we define the vectors $\mathbf{u}_\epsilon = \mathbf{Q}_\epsilon(\mathbf{s}) - \mathbf{s}$. These two vectors completely characterize the two half-lines that are tangent to the protected zone when $\mathbf{s}^2 > D^2$.

We combine the two cases into the function `tangent_line` to compute the tangent vectors.

Definition 5 (`tangent_line`).

$$\begin{aligned} \text{tangent_line}(\mathbf{s}, \epsilon) \equiv & \text{ IF } \mathbf{s}^2 = D^2 \text{ THEN} \\ & \epsilon \mathbf{s}^\perp \\ & \text{ ELSE} \\ & \mathbf{Q}_\epsilon(\mathbf{s}) - \mathbf{s} \\ & \text{ ENDIF} \end{aligned}$$

Lemma 1 (`tangent_line`). *The relative velocity vector \mathbf{v} yields a trajectory that is tangent to the protected zone if and only if $\mathbf{v} = k \text{tangent_line}(\mathbf{s}, \epsilon)$ for some $k \geq 0$.*

2. Magnitude of the Tangents

From lemma 1, we know that for any $k \geq 0$, the relative velocity vector $k \text{tangent_line}(\mathbf{s}, \epsilon)$ yields a trajectory that is tangent to the protected zone. Now we must find a $k \geq 0$ such that $\mathbf{v}'_o{}^2 = \mathbf{v}_o^2$ when $k \text{tangent_line}(\mathbf{s}, \epsilon) = \mathbf{v}'_o - \mathbf{v}_i$. Using equation (3), equation (5) can be transformed into a quadratic equation in k

$$(k\mathbf{u} + \mathbf{v}_i)^2 = \mathbf{v}_o^2, \quad (6)$$

which has at most two solutions for k . The function `trk_only_line` solves this quadratic equation.

Definition 6 (`trk_only_line`).

$$\begin{aligned} \text{trk_only_line}(\mathbf{u}, \mathbf{v}_o, \mathbf{v}_i, \epsilon) \equiv & \text{ LET } a = \mathbf{u}^2, \\ & b = 2(\mathbf{u} \cdot \mathbf{v}_i), \\ & c = \mathbf{v}_i^2 - \mathbf{v}_o^2, \\ & \Delta = b^2 - 4ac \text{ IN} \\ & \text{ IF } \Delta \geq 0 \text{ THEN LET } k = \frac{-b + \epsilon\sqrt{\Delta}}{2a} \text{ IN } \langle k, k\mathbf{u} + \mathbf{v}_i \rangle \\ & \text{ ELSE } \langle 0, \mathbf{0} \rangle \end{aligned}$$

The function `trk_only_line` returns the pair $\langle 0, \mathbf{0} \rangle$ when there are no solutions for k and \mathbf{v}'_o .

Lemma 2 (Completeness of `trk_only_line`). *A number k and a non-null vector \mathbf{v}'_o satisfy equations (3) and (5) if and only if $\langle k, \mathbf{v}'_o \rangle = \text{trk_only_line}(\mathbf{u}, \mathbf{v}_o, \mathbf{v}_i, \epsilon)$.*

3. Transition Points from Tangents

Using lemmas 1 and 2, we have that the horizontal transition points are exactly defined by the vectors \mathbf{v}'_o that satisfy

$$\langle k, \mathbf{v}'_o \rangle = \text{trk_only_line}(\text{tangent_line}(\mathbf{s}, \epsilon_1), \mathbf{v}_o, \mathbf{v}_i, \epsilon_2), \quad (7)$$

for $k \geq 0$, $\epsilon_1 = \pm 1$, $\epsilon_2 = \pm 1$. We observe that this equation uses ϵ_1 and ϵ_2 , each may be either -1 or $+1$; thus, there are four possible solutions to this equation.

Combining these results, we define

Definition 7 (`trk_line`).

$$\begin{aligned} \text{trk_line}(\mathbf{s}, \mathbf{v}_o, \mathbf{v}_i, \epsilon_1, \epsilon_2) \equiv & \text{ LET } \langle k, \mathbf{v}'_o \rangle = \text{trk_only_line}(\text{tangent_line}(\mathbf{s}, \epsilon_1), \mathbf{v}_o, \mathbf{v}_i, \epsilon_2) \text{ IN} \\ & \text{ IF } k \geq 0 \text{ THEN } \mathbf{v}'_o \\ & \text{ ELSE } \mathbf{0} \end{aligned}$$

The function `trk_line` returns the vector $\mathbf{0}$ when there are no solutions for \mathbf{v}'_o .

Theorem 3 (Completeness of `trk_line`). *A non-null vector \mathbf{v}'_o satisfy equations (3) and the half-line $\mathbf{s} + t(\mathbf{v}'_o - \mathbf{v}_i)$, for $t \geq 0$, is tangent to the protected zone if and only if $\mathbf{v}'_o = \text{trk_line}(\mathbf{s}, \mathbf{v}_o, \mathbf{v}_i, \epsilon_1, \epsilon_2)$ for some $\epsilon_1 = \pm 1$, $\epsilon_2 = \pm 1$.*

B. Lookahead Time Transition Points

We now consider ways to use the lookahead time to ignore those aircraft that are “too far” away. First, we find the velocity vectors where the ownship will touch the protected zone at precisely the lookahead time, as illustrated by figure 4.

Lookahead time transition points are determined by vectors \mathbf{v}'_o that satisfy three equations: equation (3) along with

$$(\mathbf{s} + T(\mathbf{v}'_o - \mathbf{v}_i))^2 = D^2, \quad (8)$$

$$(\mathbf{s} + T(\mathbf{v}'_o - \mathbf{v}_i)) \cdot (\mathbf{v}'_o - \mathbf{v}_i) \leq 0. \quad (9)$$

Equation (3) states that \mathbf{v}'_o has the same norm as \mathbf{v}_o , equation (8) states that the ownship is at the border of the protected zone at time T , and equation (9) states that immediately after time T the ownship enters the protected zone.

Solutions to equations (3) and (8) correspond to the intersection of two circles, which may have zero, one, two, or infinite solutions. We first consider the special case of infinite solutions.

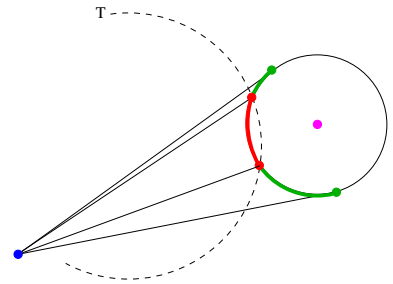


Figure 4. Lookahead Time Transition Points

1. Special Case

First we consider the case where there is an infinite number of solutions. For the remainder of this section we will assume $\mathbf{s}, \mathbf{v}_o, \mathbf{v}_i$ satisfy `trk_special_case?`.

Definition 8 (`trk_special_case?`).

$$\text{trk_special_case?}(\mathbf{s}, \mathbf{v}_o, \mathbf{v}_i) \equiv \mathbf{s} = T\mathbf{v}_i \text{ and } \mathbf{v}_o^2 = \frac{D^2}{T^2}. \quad (10)$$

Within this special case and given equation (3), then equation (8) is satisfied. Next, we have formally proven the following result

Lemma 3 (Special Case). *Equation (9) holds if and only if `¬conflict?(s, v'_o, v_i)`.*

By lemma 3, we no longer need to solve (9) but only a restricted instance of it

$$(\mathbf{s} + T(\mathbf{v}'_o - \mathbf{v}_i)) \cdot (\mathbf{v}'_o - \mathbf{v}_i) = 0, \quad (11)$$

which has at most two solutions. To solve equation (11), we transform it into the equivalent form:

$$\mathbf{v}_i \cdot \mathbf{v}'_o - \frac{D^2}{T^2} = 0, \quad (12)$$

In our mathematical development, we have defined functions to solve these kinds of equations. In particular, we have found,

Lemma 4 (Completeness of `trk_only_dot`). *Let \mathbf{a}, \mathbf{b}' be non-null vectors and $e > 0$.*

$$\mathbf{a} \cdot (\mathbf{b}' - \mathbf{c}) = e \text{ and } \mathbf{b}^2 = \mathbf{b}'^2 \text{ if and only if } \mathbf{b}' = \text{trk_only_dot}(\mathbf{a}, \mathbf{b}, \mathbf{c}, e, \epsilon)$$

where,

$$\text{trk_only_dot}(\mathbf{a}, \mathbf{b}, \mathbf{c}, e, \epsilon) \equiv \text{LET } \langle k, \mathbf{b}' \rangle = \text{trk_only_line}(\mathbf{a}^\perp, \mathbf{b}, \mathbf{c} + \frac{e}{\mathbf{a}^2} \mathbf{a}, \epsilon) \text{ IN } \mathbf{b}',$$

provided $\mathbf{a} \neq \mathbf{0}$, $e > 0$, and $\epsilon = \pm 1$.

We can cast equation (12) in the form of lemma 4; so, the lookahead transitions points for the special case are exactly defined by the vectors $\mathbf{v}'_o = \text{trk_only_dot}(\mathbf{v}_i, \mathbf{v}_o, \mathbf{0}, \frac{D^2}{T^2}, \epsilon)$ for $\epsilon = \pm 1$.

2. General Case

Now we consider the case of vectors $\mathbf{s}, \mathbf{v}_o, \mathbf{v}_i$ that do not satisfy `trk_special_case?(s, v_o, v_i)`. In this case, there are at most two solutions to equations (3) and (8). To find these solutions, we note that when equation (3) holds, equation (8) can be written as

$$\mathbf{w} \cdot (\mathbf{v}'_o - \mathbf{v}_i) = e, \quad (13)$$

where $\mathbf{w} = \mathbf{s} - T\mathbf{v}_i$ and $e = \frac{(D^2 - \mathbf{s}^2 - T^2)(\mathbf{v}_o^2 - \mathbf{v}_i^2)}{2T}$.

As in the special case, equation (13) can be resolved using the function `trk_only_dot`. Using lemma 4, the lookahead transition points for the general case are exactly defined by the vectors \mathbf{v}'_o that satisfy equation (9) and

$$\mathbf{v}'_o = \text{trk_only_dot}(\mathbf{w}, \mathbf{v}_o, \mathbf{v}_i, e, \epsilon), \quad (14)$$

for $\epsilon = \pm 1$.

Combining this result with the special case we define,

Definition 9 (`trk_circle`).

```

trk_circle(s, v_o, v_i, ε) ≡ IF trk_special_case?(s, v_o, v_i) THEN
    trk_only_dot(v_i, v_o, 0,  $\frac{D^2}{T^2}$ , ε)
ELSE
    LET v'_o = trk_only_dot(s - T v_i, v_o, v_i,  $\frac{(D^2 - s^2 - T^2)(v_o^2 - v_i^2)}{2T}$ , ε) IN
    IF (s + T(v'_o - v_i)) · (v'_o - v_i) ≤ 0 THEN
        v'_o
    ELSE
        0
    ENDIF
ENDIF

```

The function `trk_circle` returns the vector $\mathbf{0}$ when there are no solutions for \mathbf{v}'_o .

Theorem 4 (Completeness of `trk_circle`).

- Case `trk_special_case?(s, v_o, v_i)`: A non-null vector \mathbf{v}'_o satisfy equations (3), (8), and (12) if and only if $\mathbf{v}'_o = \text{trk_circle}(s, \mathbf{v}_o, \mathbf{v}_i, \epsilon)$ for some $\epsilon = \pm 1$.
- Case `¬trk_special_case?(s, v_o, v_i)`: A non-null vector \mathbf{v}'_o satisfy equations (3), (8), and (9) if and only if $\mathbf{v}'_o = \text{trk_circle}(s, \mathbf{v}_o, \mathbf{v}_i, \epsilon)$ for some $\epsilon = \pm 1$.

VI. Track Angle Prevention Bands Algorithm

In this section, we provide the mathematical basis that guarantee the correctness of the track angle prevention bands algorithm. First, we define a set of transition vectors and a set transition angles as follows:

$$\begin{aligned} \mathcal{V} &\equiv \{ \mathbf{v}'_o \neq \mathbf{0} \mid (\mathbf{v}'_o = \text{trk_line}(s, \mathbf{v}_o, \mathbf{v}_i, \epsilon_1, \epsilon_2) \text{ for some } \epsilon_1 = \pm 1, \epsilon_2 = \pm 1) \text{ or} \\ &\quad (\mathbf{v}'_o = \text{trk_circle}(s, \mathbf{v}_o, \mathbf{v}_i, \epsilon) \text{ for some } \epsilon = \pm 1) \} \\ \mathcal{A} &\equiv \{ \alpha \mid \alpha = \text{track}(\mathbf{v}'_o) \text{ for some } \mathbf{v}'_o \in \mathcal{V} \}. \end{aligned}$$

An open interval (α_1, α_2) is a *track angle band* if it does not contain transition points, i.e., $0 \leq \alpha_1 \leq \alpha_2 < 360$ and for all $\beta \in \mathcal{A} : \beta \notin (\alpha_1, \alpha_2)$. We have formally proven the basic correctness result for track angle conflict prevention information, theorem 5.

Theorem 5 (Completeness of Track Angle Bands). *Let (α_1, α_2) be a track angle band and let $\alpha \in (\alpha_1, \alpha_2)$.*

- If `cd2d(s, ||v_o||∠α, v_i)` returns `true`, then for all $\gamma \in (\alpha_1, \alpha_2) : \text{conflict?}(s, ||\mathbf{v}_o||\angle\gamma, \mathbf{v}_i)$,
- If `cd2d(s, ||v_o||∠α, v_i)` returns `false`, then for all $\gamma \in (\alpha_1, \alpha_2) : \neg \text{conflict?}(s, ||\mathbf{v}_o||\angle\gamma, \mathbf{v}_i)$.

The proof of theorem 5 involves the fact that the function Ω_α is continuous, the Intermediate Value theorem (theorem 2), theorems 1, 3, and 4, and the fact that the set \mathcal{A} contains all the transition track angles.

Based on theorem 5, we propose the following algorithm that computes $\mathcal{G}_{<T}^{o,i}$. The algorithm first constructs a list that contains all the elements in \mathcal{A} . Then, the elements 0 and 360 are added and the list is sorted. Next, the list is pair-wise iterated to get each band and the `cd2d` algorithm is used on the mid-point of the band to determine if the band should be in the set $\mathcal{G}_{<T}^{o,i}$. In this algorithm, we are not making any special attempt to handle implementation concerns such as floating point round-off errors.

Definition 10 (Pair-wise Algorithm for Track Angle Prevention Bands).

```

trk_bands(s, vo, vi) ≡
    angle[0] := track(trk_line(s, vo, vi, -1, -1));
    angle[1] := track(trk_line(s, vo, vi, -1, +1));
    angle[2] := track(trk_line(s, vo, vi, +1, -1));
    angle[3] := track(trk_line(s, vo, vi, +1, +1));

    angle[4] := track(trk_circle(s, vo, vi, -1));
    angle[5] := track(trk_circle(s, vo, vi, +1));

    angle[6] := 0;
    angle[7] := 360;

    sort(angle);
     $\mathcal{G}_{<T}^{o,i} := \emptyset;$ 
    FOR EACH  $\alpha_i, \alpha_{i+1}$  IN angle
        LET  $\alpha = (\alpha_i + \alpha_{i+1})/2$  IN
        IF cd2d(s,  $\|\mathbf{v}_o\| \angle \alpha, \mathbf{v}_i$ ) THEN
             $\mathcal{G}_{<T}^{o,i} := \mathcal{G}_{<T}^{o,i} \cup (\alpha_i, \alpha_{i+1});$ 
        ENDIF
    RETURN  $\mathcal{G}_{<T}^{o,i};$ 

```

For efficiency, we could remove from the list `angle` the points where the functions `trk_line` and `trk_circle` return the vector $\mathbf{0}$ because this vector does not determine a transition point. However, since `track(0)` is assumed to be 0, the proposed algorithm is correct.

VII. Conclusions

The mathematics underlying conflict prevention systems is more subtle than expected. Instead of a trigonometric analysis that yields fourth-order polynomial equations, a vector-based approach is developed. We have found a clean decomposition of the problem. First the problem is divided along the dimensions of ground speed and track angle. Next, the problem is divided into near-term and intermediate-term time horizons. Then the problem may be reduced from an N-aircraft problem, to a pair-wise problem. Since the maximum number of transition points is fixed, this pair-wise division of the problem results in an algorithm that scales linearly with the number of traffic aircraft.

In this paper, we have focused on a 2-dimensional prevention bands algorithm for track angle. Techniques in this paper have been used to address the ground speed dimensions of conflict prevention information too. The major distinction between the mathematical development for these dimensions and those presented in this paper is that different transition points are used. For completeness, a 2-dimensional ground speed prevention band algorithm is presented in the appendix. We have formally verified the correctness of the mathematical basis of these algorithms. Due to safety concerns associated with incorrect information from a conflict prevention system, we believe this further step of a formal verification is justified.

An open question is should a 2-dimensional algorithm be used or a 3-dimensional one. This question cannot be answered formally, but rather is based on pilot acceptance. Would a pilot, looking at a horizontal display, expect that separation must occur only in the horizontal dimension? If so, then a 2-dimensional algorithm is preferred. In any case, an extension of this work to 3-dimensional algorithms is work in progress.⁵ Extending this analysis is not straightforward. First, we need a new definition of the minimum distance function Ω that characterizes 3-dimensional conflicts. Since in a 3-dimensional space the protected zone is a cylinder rather than a sphere a distance function based on 3-dimensional Euclidean distance does not seem to be appropriate. Furthermore, we need to find the new transition points added by the vertical dimension

and to prove that we have found all of them. These issues add considerable complexity to the formal proofs.

Beyond these developments, we also intend to use the mathematics developed in this paper as a foundation for evaluating the traffic complexity and to develop techniques for down-selecting solutions from conflict detection and resolution algorithms, which produce multiple conflict resolutions.

References

¹Bryan Barmore, Edward Johnson, David Wing, and Richard Barhydt. Airborne conflict management within confined airspace in a piloted simulation of DAG-TM autonomous aircraft operations. In *The 5th USA-Europe ATM Seminar*, 2003.

²N.A. Doble, R. Barhydt, and J.M. Hitt. Distributed conflict management in en route airspace: human-in-the-loop results. *The 24th Digital Avionics Systems Conference*, Oct./Nov. 2005.

³J. Hoekstra, R. Ruigrok, R. van Gent, J. Visser, B. Gijssbers, 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.

⁴J. M. Hoekstra. Designing for safety: The free flight air traffic management concept. Technical Report 90-806343-2-8, Technische Universiteit Delft, November 2001.

⁵Jeffrey M. Maddalon, Ricky W. Butler, César Muñoz, and Gilles Dowek. A mathematical basis for the safety analysis of conflict prevention algorithms. Technical Report TM-2009-215768, NASA Langley Research Center, June 2009.

⁶S. Owre, J. Rushby, and N. Shankar. PVS: A prototype verification system. In Deepak Kapur, editor, *Proc. 11th Int. Conf. on Automated Deduction*, volume 607 of *Lecture Notes in Artificial Intelligence*, pages 748–752. Springer-Verlag, June 1992.

⁷T.W. Rand and M.S. Eby. Algorithms for airborne conflict detection, prevention, and resolution. *The 23rd Digital Avionics Systems Conference*, October 2004.

VIII. Appendix

This appendix defines algorithms to compute 2-dimensional ground speed conflict prevention bands. The mathematical development that led to these algorithms is presented in the related technical report.⁵

Definition 11 (`det`).

$$\det(\mathbf{v}, \mathbf{u}) \equiv v_x u_y - v_y u_x$$

Definition 12 (`gs_line`).

```

gs_line(s, v_o, v_i, ε) ≡
  LET v = tangent_line(s, ε) IN
  IF det(v_o, v) ≠ 0 THEN
    LET k = det(v_i, v_o) / det(v_o, v),
        λ = det(v_i, v) / det(v_o, v) IN
    IF λ > 0 AND k ≥ 0 THEN
      λ v_o
    ELSE
      0
    ENDIF
  ELSE
    0
  ENDIF

```

Definition 13 (gs_circle).

```

gs_circle(s, v_o, v_i, ε) ≡
  LET w = s - T v_i,
      a = T2 v_o2,
      b = 2T (w · v_o),
      c = w2 - D2 IN
  IF discr(a, b, c) ≥ 0 THEN
    LET λ = root(a, b, c, ε) IN
    IF λ ≥ 0 THEN
      LET v' = λ v_o - v_i IN
      IF (s + T v') · v' ≤ 0 THEN
        λ v_o
      ELSE
        0
      ENDIF
    ELSE
      0
    ENDIF
  ELSE
    0
  ENDIF

```

Definition 14 (gs_bands).

```

gs_bands(s, v_o, v_i) ≡
  gs[0] := ||gs_line(s, v_o, v_i, -1)||;
  gs[1] := ||gs_line(s, v_o, v_i, +1)||;
  gs[2] := ||gs_circle(s, v_o, v_i, -1)||;
  gs[3] := ||gs_circle(s, v_o, v_i, +1)||;
  gs[4] := min_ground_speed;
  gs[5] := max_ground_speed;
  sort(gs);
  G<To,i := ∅;
  FOR EACH gsi, gsi+1 IN gs
    LET λ =  $\frac{gs_i + gs_{i+1}}{2 ||v_o||}$ ,
        v'_o = (λ vox, λ voy) IN
    IF cd2d(s, v'_o, v_i) THEN
      G<To,i := G<To,i ∪ (gsi, gsi+1);
    ENDIF
  RETURN G<To,i;

```