NASA/TM-2009-215768

A Mathematical Basis for the Safety Analysis of Conflict Prevention Algorithms

Jeffrey M. Maddalon, Ricky W. Butler, and César A. Muñoz
Langley Research Center, Hampton, Virginia

Gilles Dowek
Ecole Polytechnique, France

June 2009

NASA STI Program . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA scientific and technical information (STI) program plays a key part in helping NASA maintain this important role.

The NASA STI program operates under the auspices of the Agency Chief Information Officer. It collects, organizes, provides for archiving, and disseminates NASA's STI. The NASA STI program provides access to the NASA Aeronautics and Space Database and its public interface, the NASA Technical Report Server, thus providing one of the largest collections of aeronautical and space science STI in the world. Results are published in both non-NASA channels and by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers, but having less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services also include creating custom thesauri, building customized databases, and organizing and publishing research results.

For more information about the NASA STI program, see the following:

- Access the NASA STI program home page at *http://www.sti.nasa.gov*

- E-mail your question via the Internet to help@sti.nasa.gov

- Fax your question to the NASA STI Help Desk at 443-757-5803

- Phone the NASA STI Help Desk at 443-757-5802

- Write to:
  NASA STI Help Desk
  NASA Center for AeroSpace Information
  7115 Standard Drive
  Hanover, MD 21076-1320

NASA/TM-2009-215768

A Mathematical Basis for the Safety Analysis of Conflict Prevention Algorithms

Jeffrey M. Maddalon, Ricky W. Butler, and César A. Muñoz
Langley Research Center, Hampton, Virginia

Gilles Dowek
Ecole Polytechnique, France

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23681-2199

June 2009

Available from:

NASA Center for AeroSpace Information
7115 Standard Drive
Hanover, MD 21076-1320
443-757-5802

Abstract

In air traffic management systems, a conflict prevention system examines the traffic and provides ranges of guidance maneuvers that avoid conflicts. This guidance takes the form of ranges of track angles, vertical speeds, or ground speeds. These ranges may be assembled into prevention bands: 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 all maneuvers. Without conflict prevention information, a pilot might perform a maneuver that causes a near-term conflict. Because near-term conflicts can lead to safety concerns, strong verification of correct operation is required. This paper presents a mathematical framework to analyze the correctness of algorithms that produce conflict prevention information. This paper examines multiple mathematical approaches: iterative, vector algebraic, and trigonometric. The correctness theories are structured first to analyze conflict prevention information for all aircraft. Next, these theories are augmented to consider aircraft which will create a conflict within a given lookahead time. Certain key functions for a candidate algorithm, which satisfy this mathematical basis are presented; however, the proof that a full algorithm using these functions completely satisfies the definition of safety is not provided.

Acknowledgement: The author César Muñoz performed much of his contribution to this work as a staff scientist at the National Institute for Aerospace in Hampton, Virginia.

Contents

List of Figures

1 Introduction

Many tools have been proposed to help aircraft maintain separation standards. At the lowest level, algorithms are developed that detect situations where separation standards will be violated in the near future (called a *conflict*). Once a conflict is detected, then a conflict resolution algorithm can be applied to create a new path in which there is no conflict. 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. This conflict prevention information allows the pilot to perform conflict-free maneuvering.

Avoiding potential conflicts involves analyzing possible maneuvers of the aircraft. There are two basic approaches to tactical[1] 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, ground speeds, or vertical 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*. This information may be used for probing, bands, or even future highly-automated approaches. The National Aerospace Laboratory (NLR) refers to their conflict prevention capability as Predictive Airborne Separation Assurance System or Predictive ASAS [5]. 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 for the definition of the algorithm's input and output and also as a prototype of a conflict prevention system's behavior.

Given the near-term nature of these conflicts, it is critical that systems provide correct conflict-prevention information. The correctness of these solutions is established through a mathematical characterization and analysis of the conflict-free regions for vertical speed, ground speed and track angles changes. Most of the theory presented in this paper has been formalized and verified in the Prototype Verification System (PVS) [7]. We present the theorems and proofs in standard notation to make them accessible to a wider audience. The PVS theories and proofs are available at the web site.[2]

In addition to a general mathematical framework for the analysis of conflict

[1] We use the term tactical to mean a system which only uses near-term predictions of aircraft behavior without incorporating pilot or controller intent.

[2] `http://research.nianet.org/fm-at-nia/ACCoRD/`

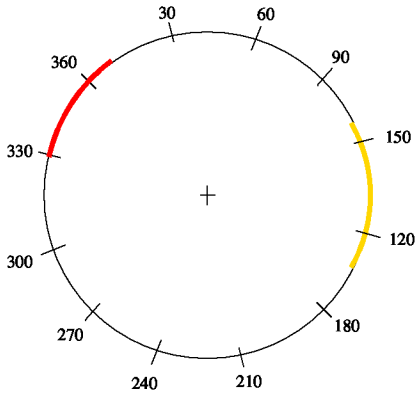

Figure 1. Compass Rose with Conflict Prevention Bands

prevention information, we present candidate algorithms for vertical speed, ground speed, and track angle bands, along with preliminary mathematical analysis of these approaches.

Although conflict-prevention systems have been used in several human-in-the-loop simulation experiments [1, 2, 6] and their functionality has been described in other papers [5,8], we believe this is the first published analysis of such information. The primary focus of this work has been conflict-prevention systems for airborne operation, but there is nothing inherent in this approach which precludes use in ground-based systems.

2 Iterative Solution

The formally-verified conflict detection and resolution algorithm `KB3D` [3] contains a conflict probe called `CD3D` (the `CD3D` algorithm is described in appendix A). Using this conflict probe, one can develop a simple iterative algorithm to compute prevention bands. For example, the track angle algorithm could be as shown in figure 2. Similar iterative algorithms could be developed for the vertical speed and ground speed guidance maneuvers.

The `CD3D` algorithm has been formally proven correct and complete, i.e., the function `CD3D` returns the Boolean value `TRUE` if and only if there is a predicted loss of separation. However, the correctness and completeness of `CD3D` does not imply the correctness nor the completeness of this simple iterative algorithm. Indeed, for any given step size, there are track angles that should be colored red or yellow that are missed. Conversely, not all the angles in a red or yellow band necessarily indicate a predicted loss of separation. To minimize this problem, the track step may be set to a very small angle. However, in that case, the efficiency of the algorithm suffers due to the large number of points to be checked. For these reasons, we seek an analytical approach that precisely computes the complete set of preventions bands.

```
input: ownship state,
       list of traffic aircraft,
       minimum horizontal separation D,
       minimum vertical separation H,
       track discretization step

Track = 0;

while Track < 360 do
   DisplayColor = green

   NewOwnship.position = Ownship.position
   NewOwnship.velocity = compute new velocity vector which has
                         the magnitude of 'Ownship.velocity' and
                         points in the direction of 'Track'

   for each traffic aircraft, Traffic:
     RelativePosition = NewOwnship.position - Traffic.position
     RelativeVelocity = NewOwnship.velocity - Traffic.velocity

     if CD3D(RelativePosition, RelativeVelocity, D, H, 3)
        then DisplayColor = red
     else if CD3D(RelativePosition, RelativeVelocity, D, H, 5)
        then DisplayColor = amber
     end if

   draw point on compass at angle 'Track' with color 'DisplayColor'

   Track = Track + step

end while
```

Figure 2. Track Angle Iterative Algorithm

3 Problem Assessment

Each aircraft's contribution to the prevention band is independent of all other traffic; thus, the problem neatly divides into a series of aircraft pairs: the ownship and each traffic aircraft. The conflict-free regions for each aircraft may then be merged into an overall picture of the airspace around the ownship. Analysis of the algorithm to merge bands is left for future work.

3.1 Modeling Considerations

Throughout this paper, we use the following notation to represent the pair of aircraft.

$\mathbf{s}_o$	3D vector	initial position of the ownship aircraft
$\mathbf{v}_o$	3D vector	initial velocity of the ownship aircraft
$\mathbf{s}_i$	3D vector	initial position of the traffic aircraft
$\mathbf{v}_i$	3D vector	initial velocity of the traffic aircraft

The components of each vector are scalar values, so they are represented without the bold-face font, for example $\mathbf{s}_o = (s_{ox}, s_{oy}, s_{oz})$. 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. For notational convenience, we use $\mathbf{v}^2 = \mathbf{v} \cdot \mathbf{v}$.

In the airspace system, the separation criteria are specified as a minimum horizontal separation D and a minimum vertical separation H (in much of the airspace D is 5 nautical miles and H is 1000 feet). It is convenient to develop the theory using a translated coordinate system. The relative position $\mathbf{s}$ is defined to be $\mathbf{s} = \mathbf{s}_o - \mathbf{s}_i$ and relative velocity of the ownship with respect to the traffic aircraft is denoted by $\mathbf{v}$. With these vectors the traffic aircraft is at the center of the coordinate system and does not move. For example in figure 3, 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 3. The separation criteria defines a cylinder of radius D and half-height H around the traffic aircraft. This cylinder is called the *protected zone.*

In figure 3, the green vectors show possible resolution vectors. In this figure, there is a horizontal conflict because 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. More formally, a horizontal conflict occurs if there exists a future time t where the aircraft 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 3.1 (`horizontal_conflict?`).

$$\texttt{horizontal\_conflict?}(\mathbf{s}, \mathbf{v}) \equiv \quad \exists\, t \geq 0 : (\mathbf{s} + t\mathbf{v})^2 < D^2, \tag{1}$$

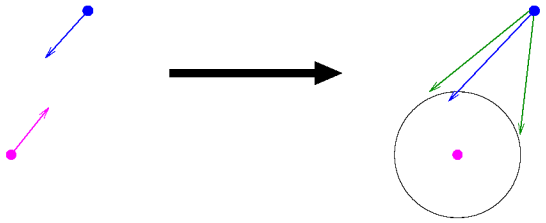

Figure 3. Translated Coordinate System

where $\mathbf{s}$ and $\mathbf{v}$ are, respectively, the projections on the horizontal plane of the relative position and velocity vectors of the ownship with respect to the traffic.

In this definition, we model future aircraft positions as a linear projection of the aircraft's velocity from its current position. An aircraft's acceleration—both positional and angular—are not modeled. It is easy to show that `horizontal_conflict?` precisely defines a minimum separation D between the points $\mathbf{s}_o + t\mathbf{v}_o$ and $\mathbf{s}_i + t\mathbf{v}_i$:

$$\begin{aligned} \sqrt{((s_{ox} + v_{ox}t) - (s_{ix} + v_{ix}t))^2 + ((s_{oy} + v_{oy}t) - (s_{iy} + v_{iy}t))^2} &< D \\ \iff (s_x + v_x t)^2 + (s_y + v_y t)^2 &< D^2 \\ \iff (\mathbf{s} + t\mathbf{v})^2 &< D^2. \end{aligned}$$

A vertical conflict occurs if there exists a future time t where the aircraft are within distance H of each other, i.e.,

$$|(s_{oz} + tv_{oz}) - (s_{iz} + tv_{iz})| < H.$$

Two aircraft are in a *conflict* if there exists a future time t where the two aircraft have *both* horizontal and vertical conflicts—that is, there is a predicted loss of separation.

3.2 Lookahead Time

If an aircraft restricts its movement based on all aircraft within its ADS-B range, then many relatively safe maneuvers will be unnecessarily avoided.[3] 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 ASAS [5] 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. The Predictive ASAS uses 3 minutes for T_{red} and 5 minutes for T_{amber}; our analysis leaves these as parameters. From a prevention band approach, if a loss of separation will occur within T_{red}, then the region is colored red. On

[3] ADS-B range over the ocean may reach 200 nautical miles.

the other hand, if a loss of separation will occur after T_{red}, but before T_{amber}, then the region is colored amber, otherwise it is painted green. Since $T_{red} < T_{amber}$, the boundaries for a guidance maneuver (ranges of track angles, etc.) for a conflict that is within T_{red} are completely contained within the boundaries for a conflict that is within T_{amber}. With this observation, the majority of the analysis can use a single lookahead time T, and T_{red} and T_{amber} can be accounted for by their relationship.

When we consider the lookahead time we notice that there are three distinct cases, as illustrated in figure 4. In figure 4a, all points in the protected zone can be

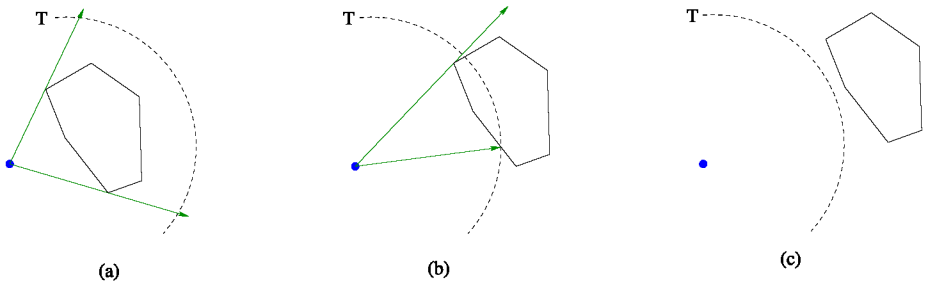

Figure 4. Relationship of Encounter Geometry and Lookahead Time

reached within the lookahead time. In this case the lookahead time does not enter into the analysis, so the aircrafts' encounter geometry can be evaluated without regard to a lookahead time. In figure 4c, where no point in the protected zone can be reached within the lookahead time, this conflict is too far away to be considered, so it does not restrict the ownship's maneuvers—essentially the traffic is ignored. It is only in figure 4b, where some points in the protected zone may be reached within the lookahead time, that lookahead time considerations must be addressed. We label situations like figure 4a as *encounter geometry* cases. We label the cases of figure 4b as *lookahead time* cases. The lookahead time cases involve analysis of both geometric and time considerations. Although the principle illustrated by this figure still holds, the protected zone is not an arbitrary regular polygon; rather, it is a rectangle for vertical speed maneuvers (incorporating the horizontal and vertical distance between the two aircraft) or a circle for both track angle and ground speed maneuvers (incorporating both horizontal dimensions).

4 Vertical Speed Prevention Bands

A vertical speed prevention band algorithm determines those vertical speeds that will result in a conflict with another aircraft. The algorithm can be used to paint a display similar to figure 5: the red region indicates those vertical speeds that will result in a conflict, while green indicates speeds that will not cause a conflict within the lookahead time.

It is assumed that the aircraft's track angle and ground speed remain unchanged while the vertical speed is varied. Consequently, if the current horizontal velocity does not have a horizontal conflict, then the solution is trivial, i.e., the region is

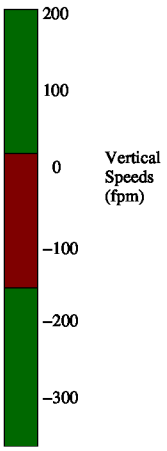

Figure 5. Vertical Speed Bands

green for all vertical speeds. When the current track does have a horizontal conflict then we are faced with the situation shown in figure 6. In this translated frame of reference the traffic aircraft is centered in the coordinate system with a vertical protected zone from $-H$ to $+H$. In this figure there are several relative velocity vectors are shown, some in conflict (red) and some that are not (green).

Computing conflict prevention bands involves whole ranges of vertical speeds. Figure 6 illustrates the problem. In this figure, the horizontal dimension represents along track distance between the two aircraft and the vertical dimension represents vertical distance between the two aircraft. The blue dot represents the position of the ownship and the the magenta dot represents the position of the traffic. The arrows represent potential velocity vectors for the ownship (varying the vertical speed). The red vectors indicate velocities where the protected zone is violated and the green vectors indicate conflict-free trajectories. Instead of computing individual trajectories, we want to analytically determine which vectors are green and which are red as the vertical speed is varied. In this figure the two relative velocity vectors $\mathbf{v}_1$ and $\mathbf{v}_2$ are the vectors that sit at the transition point between colors. If we can compute these, then we can accurately paint the vertical speed prevention band. Recognizing that vertical speed cannot be unlimited, the trajectory with the maximum vertical speed is indicated with `max_vs` and the vector with the minimum vertical speed is indicated by `-max_vs`..

4.1 Encounter Geometry Analysis

To determine the transition vectors for the encounter geometry, we consider four cases of the two aircraft. Examples of these four cases are shown in figure 7. These four cases correspond to: (a) no horizontal conflict, (b) vertical loss of separation, i.e., $-H < s_z < H$, but not a horizontal loss of separation, i.e., $s_x^2 + s_y^2 > D^2$, (c) a horizontal loss of separation, but not a vertical loss of separation, and (d) neither vertical nor horizontal loss of separation. These cases are exhaustive because, by

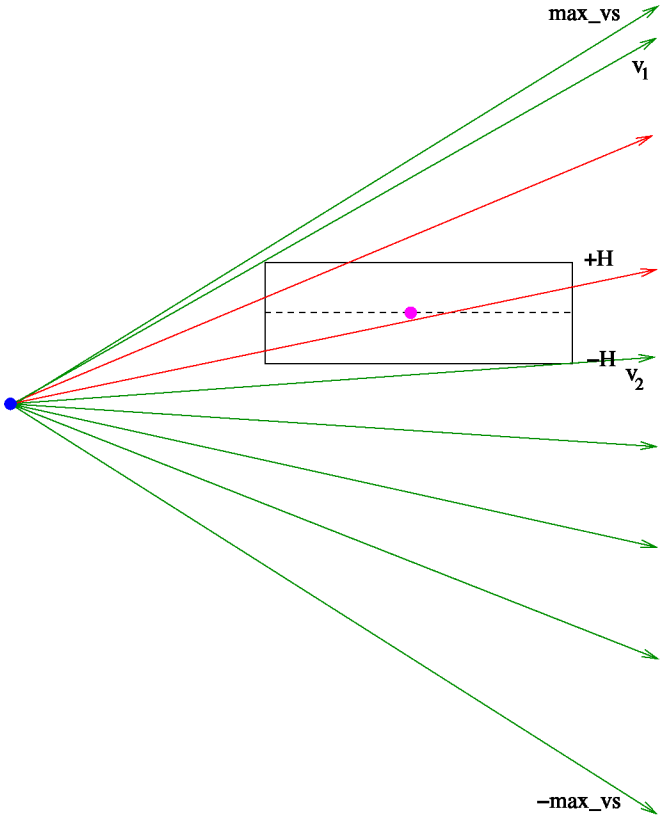

Figure 6. Vertical Speed Trajectories

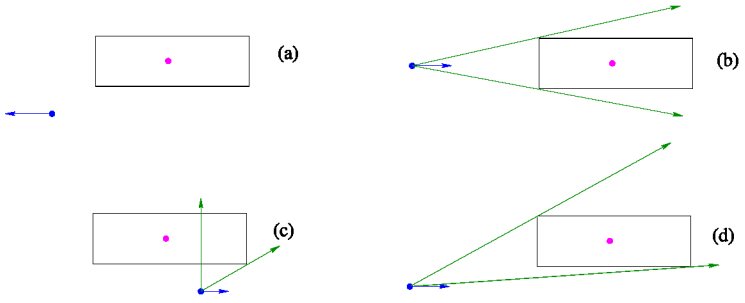

Figure 7. Vertical Encounter Geometries

assumption, we are excluding the case where there is a loss of separation both vertically and horizontally. The first case is trivial since all vertical speeds avoid conflicts and therefore none need to be prevented. The other three cases rely on the time to enter or exit the protected zone *horizontally.* To find these times, we perform a linear prediction of the relative position of the two aircraft into the future. We know the protected zone is entered when this position equals D, that is, an equation representing the intersection of a trajectory with a circular protected zone. Mathematically, we solve the following equation for t:

$$||\mathbf{s} + t\mathbf{v}|| = D, \tag{2}$$

where, for this derivation, $\mathbf{s}$ and $\mathbf{v}$ are two dimensional vectors. This equation expands to

$$(\mathbf{s} + t\mathbf{v})^2 = D^2,$$

which further expands to

$$t^2\mathbf{v}^2 + 2t(\mathbf{s} \cdot \mathbf{v}) + \mathbf{s}^2 - D^2 = 0. \tag{3}$$

This equation is a quadratic in t, with

$$a = \mathbf{v}^2, \quad b = \mathbf{s} \cdot \mathbf{v}, \quad c = \mathbf{s}^2 - D^2.$$

The roots of this equation give us the entry and exit times into the protected zone. These are named $\Theta(\mathbf{s}, \mathbf{v}, -1)$ and $\Theta(\mathbf{s}, \mathbf{v}, +1)$, respectively:

$$\Theta(\mathbf{s}, \mathbf{v}, -1) = \frac{-b - \sqrt{b^2 - ac}}{a}, \tag{4}$$

$$\Theta(\mathbf{s}, \mathbf{v}, +1) = \frac{-b + \sqrt{b^2 - ac}}{a}. \tag{5}$$

Where it is unambiguous, we will abbreviate these as Θ_- and Θ_+. If the relative ground speed is 0, i.e., $\mathbf{v} = 0$, then $\Theta_\pm$ is undefined, which corresponds to the first case (no horizontal conflict), unless the two aircraft are not horizontally separated.

4.1.1 Vertical Loss of Separation

The situation where there is vertical loss of separation but not a horizontal loss of separation (the second case, mentioned above) can be solved by computing the vertical speed required to enter the protected zone (that is, when the time is Θ_-) at its top and bottom edges. These two speeds can be represented with the introduction of a parameter ϵ, which is defined as

$\epsilon = +1$	top of the vertical protected zone
$\epsilon = -1$	bottom of the vertical protected zone

Using ϵ the top and bottom edges can be represented as ϵH. The vertical speed is computed with the following derivation.

$$\begin{aligned} s_z + v_z \Theta_- &= \epsilon H, \\ v_z \Theta_- &= \epsilon H - s_z, \\ v_z &= \frac{\epsilon H - s_z}{\Theta_-}, \\ v_{oz} - v_{iz} &= \frac{\epsilon H - s_z}{\Theta_-}, \\ v_{oz} &= v_{iz} + \frac{\epsilon H - s_z}{\Theta_-}. \end{aligned}$$

The right side of this equation is called $\texttt{vertical\_THETA1}(\mathbf{s}, \mathbf{v}_o, \mathbf{v}_i, \epsilon)$. We name the solutions from this function as follows

$$\begin{aligned} V_{1m} &= \texttt{vertical\_THETA1}(\mathbf{s}, \mathbf{v}_o, \mathbf{v}_i, -1), \\ V_{1p} &= \texttt{vertical\_THETA1}(\mathbf{s}, \mathbf{v}_o, \mathbf{v}_i, +1), \end{aligned}$$

where $\mathbf{s}, \mathbf{v}_o, \mathbf{v}_i$ are three dimensional vectors. The vertical bands should be colored as follows for an arbitrary vertical speed v_{oz}:

$\texttt{-max\_vs} \leq v_{oz} \leq V_{1m}$	green
$V_{1m} < v_{oz} < V_{1p}$	red
$V_{1p} \leq v_{oz} \leq \texttt{max\_vs}$	green

where `max_vs` is the maximum vertical speed.

4.1.2 Horizontal Loss of Separation

The third subcase occurs when $\mathbf{s}^2 < D^2$. In this case, we are interested in the time to exit the horizontal protected zone. If the aircraft is below the protected zone, then we must go no higher than $-H$ vertically. On the other hand, if the aircraft is above the protected zone, we must go no lower than $+H$. Both of these destinations can be captured by the formula $\texttt{sign}(s_z)H$. To compute the vertical speed we use the derivation

$$\begin{aligned} s_z + v_z \Theta_+ &= \texttt{sign}(s_z)H, \\ v_z \Theta_+ &= \texttt{sign}(s_z)H - s_z, \\ v_z &= \frac{\texttt{sign}(s_z)H - s_z}{\Theta_+}, \\ v_{oz} - v_{iz} &= \frac{\texttt{sign}(s_z)H - s_z}{\Theta_+}, \\ v_{oz} &= v_{iz} + \frac{\texttt{sign}(s_z)H - s_z}{\Theta_+}. \end{aligned}$$

The right side of the last equation is called $\texttt{vertical\_THETA2}(\mathbf{s}, \mathbf{v}_o, \mathbf{v}_i)$, which we abbreviate as V_2. The regions are colored as follows:

$s_z \geq 0$		$s_z < 0$	
$\texttt{-max\_vs} \leq v_{oz} < V_2$	red	$\texttt{-max\_vs} \leq v_{oz} \leq V_2$	green
$V_2 \leq v_{oz} \leq \texttt{max\_vs}$	green	$V_2 < v_{oz} \leq \texttt{max\_vs}$	red

4.1.3 No Loss of Separation

The fourth case is a combination of the equations developed in the two previous cases. The regions are colored as follows

$s_z \geq 0$		$s_z < 0$
$\texttt{-max\_vs} \leq v_{oz} \leq V_{1m}$	green	$\texttt{-max\_vs} \leq v_{oz} \leq V_2$
$V_{1m} < v_{oz} < V_2$	red	$V_2 < v_{oz} < V_{1p}$
$V_2 \leq v_{oz} \leq \texttt{max\_vs}$	green	$V_{1p} \leq v_{oz} \leq \texttt{max\_vs}$

4.1.4 Algorithm for Encounter Geometry

To develop an algorithm to compute the vertical speed bands, we must combine the four cases into a single solution. But a simple observation suggests a simple algorithm. We notice that there are only a small number of critical points where the vertical bands may change colors: V_{1m}, V_{1p} and V_2. We observe that if we evaluate the conflict status around these points, then we can characterize the whole range of vertical speeds. The computation of these three points can be combined into a single function:

$$
\begin{array}{l}
\texttt{vertical\_speed\_circle}(\mathbf{s}, \mathbf{v}_o, \mathbf{v}_i, \epsilon) = \\
\quad \texttt{IF } (v_{ox} - v_{ix})^2 + (v_{oy} - v_{iy})^2 = 0 \texttt{ THEN} \\
\quad\quad (v_{ox}, v_{oy}, v_{iz}) \\
\quad \texttt{ELSIF } \epsilon\, s_z < H \texttt{ AND } s_x^2 + s_y^2 > D^2 \texttt{ THEN} \\
\quad\quad (v_{ox}, v_{oy}, \texttt{vertical\_THETA1}(\mathbf{s}, \mathbf{v}_o, \mathbf{v}_i, \epsilon)) \\
\quad \texttt{ELSIF } \epsilon\, s_z \geq H \texttt{ THEN} \\
\quad\quad (v_{ox}, v_{oy}, \texttt{vertical\_THETA2}(\mathbf{s}, \mathbf{v}_o, \mathbf{v}_i)) \\
\quad \texttt{ELSE} \\
\quad\quad (0, 0, 0) \\
\quad \texttt{ENDIF}
\end{array}
$$

where

$$
\texttt{vertical\_THETA1}(\mathbf{s}, \mathbf{v}_o, \mathbf{v}_i, \epsilon) = v_{iz} + \frac{\epsilon H - s_z}{\Theta_-}
$$

and

$$
\texttt{vertical\_THETA2}(\mathbf{s}, \mathbf{v}_o, \mathbf{v}_i) = v_{iz} + \frac{\texttt{sign}(s_z) H - s_z}{\Theta_+}
$$

Due to geometric limitations all three points are never needed; instead, at most, two are required:

$$
\begin{aligned}
\mathbf{Z}_m &= \texttt{vertical\_speed\_circle}(\mathbf{s}, \mathbf{v}_o, \mathbf{v}_i, -1) \\
\mathbf{Z}_p &= \texttt{vertical\_speed\_circle}(\mathbf{s}, \mathbf{v}_o, \mathbf{v}_i, +1)
\end{aligned}
$$

We use the z component of these vectors and add the minimum (`-max_vs`) and maximum (`+max_vs`) vertical speeds to a list of four values. If we sort this list, we end up with four ascending values:

$$v_0 \leq v_1 \leq v_2 \leq v_3$$

Usually v_0 is the minimum vertical speed and v_3 is the maximum vertical speed; however, this is not required. These four values correspond to three possible regions and the color of these regions can be determined using a conflict probe like CD3D (see appendix A).

v_0 to v_1	`cd3d`$(\frac{v_0+v_1}{2})$
v_1 to v_2	`cd3d`$(\frac{v_1+v_2}{2})$
v_2 to v_3	`cd3d`$(\frac{v_2+v_3}{2})$

4.2 Lookahead Time Analysis

As introduced in section 3.2, we use the two lookahead times representing near-term conflicts (T_{red}) and mid-term conflicts (T_{amber}). We first consider a single lookahead time (T), then later add the second time.

In the finite lookahead problem, we adjust the limits of the vertical speed based on the relationship between the lookahead time and the time to enter or exit the protected zone horizontally ($\Theta_{\pm}$). Comparing the lookahead to the entry and exit times results in three cases:

- Lookahead time is before the entry time ($T \leq \Theta_-$).
- Lookahead time is between the entry and exit times ($\Theta_- < T < \Theta_+$).
- Lookahead time is after the exit time ($\Theta_+ \leq T$).

These cases correspond to situations where all points within the protected zone can reached within the lookahead time, some points can be reached within the lookahead time, and no point may be reached within the lookahead time. The first and third cases are elementary. In the first case, no vertical speeds are prevented and in the third case, the results from the encounter geometry solution can be used. The second case (figure 8) requires additional analysis. In this illustration, the horizontal axis represents time rather than distance. Here the lookahead time falls between the entry and exit times. The vector $\mathbf{v}$ is drawn at the point where T is the entry point into the protected zone. Following a development similar to `vertical_THETA2`, this vector is computed by

$$\begin{aligned} s_z + v_z T &= \epsilon H, \\ v_z T &= \epsilon H - s_z, \\ v_z &= \frac{\epsilon H - s_z}{T}, \\ v_{oz} - v_{iz} &= \frac{\epsilon H - s_z}{T}, \\ v_{oz} &= v_{iz} + \frac{\epsilon H - s_z}{T}. \end{aligned}$$

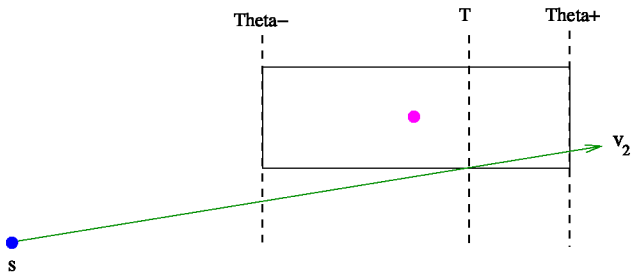

Figure 8. Vertical Speed Range with Lookahead Time

Since we are excluding the case where $T \geq \Theta_+$, we can combine this with the previous definition of `vertical_THETA2` to get a single expression

$$\texttt{vertical\_THETA2}(\mathbf{s}, \mathbf{v}_o, \mathbf{v}_i, T) \quad = \quad v_{iz} + \frac{\texttt{sign}(s_z)H - s_z}{\texttt{min}(\Theta_+, T)}. \tag{6}$$

With this new definition, the functions `vertical_THETA2` and `vertical_speed_circle` can be extended with a lookahead time.

4.3 Algorithm for Lookahead Time

We first generalize `vertical_speed_circle` to include the lookahead parameter as follows:

$$\begin{array}{l}
\texttt{vertical_speed_circle}(\mathbf{s}, \mathbf{v}_o, \mathbf{v}_i, T, \epsilon) = \\
\quad \texttt{IF } (v_{ox} - v_{ix})^2 + (v_{oy} - v_{iy})^2 = 0 \texttt{ THEN} \\
\quad\quad (v_{ox}, v_{oy}, v_{iz}) \\
\quad \texttt{ELSIF } \epsilon\, s_z < H \texttt{ AND } s_x^2 + s_y^2 > D^2 \texttt{ THEN} \\
\quad\quad (v_{ox}, v_{oy}, \texttt{vertical_THETA1}(\mathbf{s}, \mathbf{v}_o, \mathbf{v}_i, \epsilon)) \\
\quad \texttt{ELSIF } \epsilon\, s_z \geq H \texttt{ THEN} \\
\quad\quad (v_{ox}, v_{oy}, \texttt{vertical_THETA2}(\mathbf{s}, \mathbf{v}_o, \mathbf{v}_i, T)) \\
\quad \texttt{ELSE} \\
\quad\quad (0, 0, 0) \\
\quad \texttt{ENDIF}
\end{array}$$

where `vertical_THETA2` is defined by (6). We instantiate this function using T_{red} and T_{amber} yielding four possible values

$$\begin{array}{l}
\mathbf{Z}_{rm} = \texttt{vertical_speed_circle}(\mathbf{s}, \mathbf{v}_o, \mathbf{v}_i, T_{red}, -1) \\
\mathbf{Z}_{rp} = \texttt{vertical_speed_circle}(\mathbf{s}, \mathbf{v}_o, \mathbf{v}_i, T_{red}, +1) \\
\mathbf{Z}_{am} = \texttt{vertical_speed_circle}(\mathbf{s}, \mathbf{v}_o, \mathbf{v}_i, T_{amber}, -1) \\
\mathbf{Z}_{ap} = \texttt{vertical_speed_circle}(\mathbf{s}, \mathbf{v}_o, \mathbf{v}_i, T_{amber}, +1)
\end{array}$$

Once again we use the z component of these vectors and add the minimum and maximum vertical speed to create a list of six values. If we sort this list, we end up with six ascending values: $v_0 \leq v_1 \leq v_2 \leq v_3 \leq v_4 \leq v_5$. These six values correspond to five possible regions, where the color can be determined using a conflict probe like CD3D (see appendix A).

v_0 to v_1	$\texttt{cd3d}(\frac{v_0+v_1}{2})$
v_1 to v_2	$\texttt{cd3d}(\frac{v_1+v_2}{2})$
v_2 to v_3	$\texttt{cd3d}(\frac{v_2+v_3}{2})$
v_3 to v_4	$\texttt{cd3d}(\frac{v_3+v_3}{2})$
v_4 to v_5	$\texttt{cd3d}(\frac{v_4+v_5}{2})$

5 Horizontal Considerations

Both the ground speed and track prevention bands are horizontal problems. Similar to the vertical case, if there is not a vertical conflict, then all ground speeds and track angles should be allowed. So we implicitly consider only those cases where there is already a vertical conflict. For the ground speed prevention band, we vary the ground speed of the ownship and determine those ground speeds leading to a conflict. We denote the ownship velocity vector with ground speed κ as

$$\mathbf{v}_\kappa = \frac{\kappa}{||\mathbf{v}_o||}(v_{ox},\, v_{oy}). \tag{7}$$

For the track prevention bands, we vary the track angle of the ownship velocity and determine those track angles leading to a conflict. We denote the ownship velocity vector with track α as

$$\mathbf{v}_\alpha = (\omega \cos\alpha,\ \omega \sin\alpha), \tag{8}$$

where ω is constrained by $\omega^2 = v_{ox}^2 + v_{oy}^2$. In both of these definitions, the vertical speed is unchanged.

In both cases, the boundaries are determined by the vectors that are tangent to the protected zone in the relative frame of reference. The next two sections involve developing forms to express and compute these tangents. Both cases deal primarily in the horizontal dimensions; so, the vectors in this section are the two dimensional versions of the vectors presented in section 3.1. Both $\mathbf{v}_\kappa$ and $\mathbf{v}_\alpha$ may be extended to three dimensional vectors by including the original vertical speed: $(v_{\kappa x}, v_{\kappa y}, v_{oz})$ or $(v_{\alpha x}, v_{\alpha y}, v_{oz})$.

5.1 Tangent Condition

We begin by examining the necessary conditions to ensure a relative velocity vector $\mathbf{v}$ is tangent to the protected zone. One condition on the vector $\mathbf{v}$ is that, there must exist some time t that results in a position at the edge of the protected zone. We start with (2) and follow the same derivation to the quadratic (3). The roots of this equation give us the entry and exit times into the protected zone, which were previously labeled as Θ_- in (4) and Θ_+ in (5). Equation (3) provides the general

solution for the intersection of a trajectory and the protected zone. To find the resolution trajectory, we must constrain $\mathbf{v}$ to only those velocity vectors that are tangent to the protected zone. This will occur when the entrance and exit times to the protected zone are equal, $\Theta_- = \Theta_+$. These times will be equal when the discriminant of (3) is zero, that is,

$$(\mathbf{s} \cdot \mathbf{v})^2 - \mathbf{v}^2[\mathbf{s}^2 - D^2] \;=\; 0. \tag{9}$$

During the development of KB3D several simplifications to the discriminant were discovered. The key to these simplifications is the lemma **sq_det**:

Lemma 5.1 (sq_det)**.**

$$(\mathbf{s} \cdot \mathbf{v})^2 = \mathbf{v}^2\mathbf{s}^2 - \mathtt{det}(\mathbf{s}, \mathbf{v})^2, \tag{10}$$

$$\text{where, } \mathtt{det}(\mathbf{s}, \mathbf{v}) \equiv \mathbf{s}^\perp \cdot \mathbf{v} \text{ and } \mathbf{s}^\perp = (-s_y, s_x). \tag{11}$$

Proof. Algebraic simplification, which establishes

$$\begin{aligned} s_x^2 v_x^2 + 2 s_x v_x s_y v_y + s_y^2 v_y^2 - s_x^2 v_x^2 + -s_x^2 v_y^2 - s_y^2 v_x^2 + -s_y^2 v_y^2 + \mathbf{v}^2 D^2 = \\ -(s_x v_y - s_y v_x)^2 + \mathbf{v}^2 D^2. \end{aligned}$$

□

Using the **sq_det** lemma twice, (9) can be simplified as follows

$$\begin{aligned} (\mathbf{s} \cdot \mathbf{v})^2 - \mathbf{v}^2[\mathbf{s}^2 - D^2] &= 0, \\ \mathbf{v}^2\mathbf{s}^2 - \mathtt{det}(\mathbf{s}, \mathbf{v})^2 - \mathbf{v}^2[\mathbf{s}^2 - D^2] &= 0, \\ -\mathtt{det}(\mathbf{s}, \mathbf{v})^2 + D^2\mathbf{v}^2 &= 0, \\ D^2\mathbf{v}^2 &= \mathtt{det}(\mathbf{s}, \mathbf{v})^2, \\ D^2\mathbf{v}^2\mathbf{s}^2 - D^2\mathtt{det}(\mathbf{s}, \mathbf{v})^2 &= \mathtt{det}(\mathbf{s}, \mathbf{v})^2\mathbf{s}^2 - D^2\mathtt{det}(\mathbf{s}, \mathbf{v})^2, \\ D^2[\mathbf{v}^2\mathbf{s}^2 - \mathtt{det}(\mathbf{s}, \mathbf{v})^2] &= \mathtt{det}(\mathbf{s}, \mathbf{v})^2[\mathbf{s}^2 - D^2], \\ D^2(\mathbf{s} \cdot \mathbf{v})^2 &= \mathtt{det}(\mathbf{s}, \mathbf{v})^2[\mathbf{s}^2 - D^2], \\ (\mathbf{s} \cdot \mathbf{v})^2 &= R^2\mathtt{det}(\mathbf{s}, \mathbf{v})^2, \\ |\mathbf{s} \cdot \mathbf{v}| &= R\,|\mathtt{det}(\mathbf{s}, \mathbf{v})|, \\ \mathbf{s} \cdot \mathbf{v} &= R\epsilon\,\mathtt{det}(\mathbf{s}, \mathbf{v}), \end{aligned} \tag{12}$$

where $\epsilon \in \{-1, +1\}$ and

$$R \;=\; \frac{\sqrt{\mathbf{s}^2 - D^2}}{D}. \tag{13}$$

Any $\mathbf{s}$ and $\mathbf{v}$ vector that satisfies (12) forms a trajectory that is tangent to the protected zone. This remarkably succinct result factors out the squared terms of (9) into a linear expression using a simple signed variable ϵ. Formula (12) efficiently defines tangent vectors but it does not filter out tangents that have occurred in the past, i.e., that point away from the protected zone. To perform this filtering, we rely on the following lemma to find the time of closest approach.

Lemma 5.2 (`closest_approach`). *The minimal separation occurs at time τ, where*

$$\tau = -\frac{\mathbf{s} \cdot \mathbf{v}}{\mathbf{v}^2}. \tag{14}$$

Proof. The distance between the aircraft $d(t) = ||\mathbf{s} + t\mathbf{v}||$ achieves a minimum when

$$\begin{aligned} d^2(t) &= (\mathbf{s} + t\mathbf{v})^2 \\ &= \mathbf{s}^2 + 2t(\mathbf{s} \cdot \mathbf{v}) + t^2\mathbf{v}^2. \end{aligned}$$

achieves a minimum. Since this distance function has no maximum (at infinite time, the distance is infinite), the minimum occurs when the derivative of $d^2(t)$ equals zero:

$$2(\mathbf{s} \cdot \mathbf{v}) + 2t\mathbf{v}^2 = 0.$$

Solving for t we have

$$t = -\frac{\mathbf{s} \cdot \mathbf{v}}{\mathbf{v}^2}.$$

□

Therefore $\mathbf{s} \cdot \mathbf{v}$ is negative if the tangent occurs in the future. We now define a predicate that defines tangents that are towards the protected zone:

Definition 5.3 (`line_solution?`).

$$\texttt{line\_solution?}(\mathbf{s}, \mathbf{v}, \epsilon) \equiv R\,\epsilon\,\texttt{det}(\mathbf{s}, \mathbf{v}) = \mathbf{s} \cdot \mathbf{v} \texttt{ AND } \epsilon\,\texttt{det}(\mathbf{s}, \mathbf{v}) \leq 0. \tag{15}$$

It is easy to see that $\texttt{line\_solution?}(\mathbf{s}, \mathbf{v}, \epsilon)$ implies that $\mathbf{s} \cdot \mathbf{v} \leq 0$.

5.2 Computing Tangents

In this section we develop a simple way to find a point $\mathbf{Q}$ on the line from $\mathbf{s}$ that is tangent to the protected zone. An important aspect of the definition of $\mathbf{Q}$ is that it is only determined by the geometry of the encounter—the aircraft velocities are not involved. We restrict our attention to situations where there has not yet been a loss of separation, i.e., $\mathbf{s}^2 - D^2 > 0$. We define two auxiliary functions $\hat{\alpha}(\mathbf{s})$ and $\hat{\beta}(\mathbf{s})$ as follows:

$$\begin{aligned} \hat{\alpha}(\mathbf{s}) &= \frac{D^2}{\mathbf{s}^2}, \\ \hat{\beta}(\mathbf{s}) &= \frac{D\sqrt{\mathbf{s}^2 - D^2}}{\mathbf{s}^2} = R\,\hat{\alpha}(\mathbf{s}), \end{aligned}$$

where, R is defined as in (13). We will abbreviate these values as $\hat{\alpha}$ and $\hat{\beta}$ when it is clear. The following equation is a simple algebraic consequence of these definitions

$$-\hat{\beta} = \frac{\hat{\alpha} - 1}{R}. \tag{16}$$

We then define the components of $\mathbf{Q}(\mathbf{s}, \epsilon)$ as follows

$$Q_x(\mathbf{s}, \epsilon) = \hat{\alpha}(\mathbf{s})\, s_x + \epsilon\, \hat{\beta}(\mathbf{s})\, s_y,$$
$$Q_y(\mathbf{s}, \epsilon) = \hat{\alpha}(\mathbf{s})\, s_y - \epsilon\, \hat{\beta}(\mathbf{s})\, s_x.$$

The vector $\mathbf{Q}(\mathbf{s}, \epsilon) - \mathbf{s}$ is the desired tangent vector.

We would also like to compute a suitable tangent for the special case where $\mathbf{s}^2 = D^2$. In this case, an appropriate vector is the perpendicular vector. We define the function `tangent_line` in figure 9 to capture this special case. We now define

```
tangent_line(s, ε) =
    IF s² = D² THEN
        − ε s⊥
    ELSE
        Q(s, ε) − s
    ENDIF
```

Figure 9. Tangent Vector Computation

a predicate `tangent_line?` to represent the set of all tangent velocity vectors:

Definition 5.4 (`tangent_line?`)**.**

$$\texttt{tangent\_line?}(\mathbf{s}, \mathbf{v}, \epsilon) \equiv \quad \exists\, k \geq 0 : \mathbf{v} = k\, \texttt{tangent\_line}(\mathbf{s}, \epsilon) \tag{17}$$

Next we will show that all of these vectors satisfy `line_solution?` and hence are tangent to the protected zone.

Theorem 5.5 (`tangent_line_solution`)**.**

$$\texttt{tangent\_line?}(\mathbf{s}, \mathbf{v}, \epsilon) \iff \texttt{line\_solution?}(\mathbf{s}, \mathbf{v}, \epsilon).$$

Proof. The $\mathbf{s}^2 = D^2$ case is trivial, so we focus on the cases where $\mathbf{s}^2 > D^2$. First we prove $\texttt{tangent\_line?}(\mathbf{s}, \mathbf{v}, \epsilon) \implies \texttt{line\_solution?}(\mathbf{s}, \mathbf{v}, \epsilon)$. Since $\mathbf{v} = k[\mathbf{Q}(\mathbf{s}, \epsilon) - \mathbf{s}]$, we can use many algebraic simplifications and (16) to obtain

$$\begin{aligned}
\mathbf{s} \cdot \mathbf{v} &= \mathbf{s} \cdot k(\mathbf{Q} - \mathbf{s}) \\
&= k\,(\hat{\alpha} - 1)\mathbf{s}^2 \\
&= k\,(-\hat{\beta} R)\mathbf{s}^2 \\
&= R\epsilon\,(\mathbf{s}^\perp \cdot k(\mathbf{Q} - \mathbf{s})) \\
&= R\epsilon\,(\mathbf{s}^\perp \cdot \mathbf{v}) \\
&= R\epsilon\,\mathbf{det}(\mathbf{s}, \mathbf{v}).
\end{aligned}$$

Also, we need to show that $\epsilon\,\mathtt{det}(\mathbf{s},\mathbf{v}) \leq 0$. Since $\hat{\beta} \geq 0$ we have $-\hat{\beta}\mathbf{s}^2 \leq 0$. The following algebraic manipulation

$$\begin{aligned}
-\hat{\beta}\mathbf{s}^2 \leq 0 &\iff \epsilon(-\epsilon\,\hat{\beta}\,\mathbf{s}^2) \leq 0 \\
&\iff \epsilon\,[\mathbf{s}^{\perp}\cdot(\mathbf{Q}-\mathbf{s})] \leq 0 \\
&\iff \epsilon\,(\mathbf{s}^{\perp}\cdot\mathbf{v}) \leq 0 \\
&\iff \epsilon\,\mathtt{det}(\mathbf{s},\mathbf{v}) \leq 0 \\
&\iff \epsilon\,\frac{\mathbf{s}\cdot\mathbf{v}}{R} \leq 0.
\end{aligned}$$

establishes the needed result.

Now we prove that $\mathtt{line\_solution?}(\mathbf{s},\mathbf{v},\epsilon) \implies \mathtt{tangent\_line?}(\mathbf{s},\mathbf{v},\epsilon)$. We could prove that $\mathbf{v}$ is a multiple of $\mathbf{Q}(\mathbf{s},\epsilon)-\mathbf{s}$, or more succinctly, we could prove that $\mathbf{v}$ and $\mathbf{Q}(\mathbf{s},\epsilon)-\mathbf{s}$ are parallel. To show that two vectors are parallel it is sufficient to show that their 2-dimensional determinant (11) is zero. So we will show that $\mathtt{det}(\mathbf{v},\mathbf{Q}-\mathbf{s}) = 0$, which is equivalent to $\mathtt{det}(\mathbf{v},\mathbf{Q}) = \mathtt{det}(\mathbf{v},\mathbf{s})$, since $\mathtt{det}(\mathbf{v},\mathbf{Q}-\mathbf{s}) = \mathtt{det}(\mathbf{v},\mathbf{Q}) - \mathtt{det}(\mathbf{v},\mathbf{s})$. Thus,

$$\begin{aligned}
\mathtt{det}(\mathbf{v},\mathbf{Q}) &= \mathbf{v}^{\perp}\cdot\mathbf{Q} \\
&= v_x Q_y - v_y Q_x \\
&= v_x[\hat{\alpha}\,s_y - \epsilon\,\hat{\beta}\,s_x] - v_y[\hat{\alpha}\,s_x + \epsilon\,\hat{\beta}\,s_y] \\
&= \hat{\alpha}(\mathbf{v}^{\perp}\cdot\mathbf{s}) - \epsilon\hat{\beta}(\mathbf{s}\cdot\mathbf{v}).
\end{aligned}$$

Using (16) and $\mathtt{line\_solution?}$, we further simplify:

$$\begin{aligned}
\hat{\alpha}(\mathbf{v}^{\perp}\cdot\mathbf{s}) - \epsilon\hat{\beta}(\mathbf{s}\cdot\mathbf{v}) &= \hat{\alpha}(\mathbf{v}^{\perp}\cdot\mathbf{s}) + \epsilon\,\frac{(\hat{\alpha}-1)}{R}(\mathbf{s}\cdot\mathbf{v}) \\
&= \hat{\alpha}(\mathbf{v}^{\perp}\cdot\mathbf{s}) + (\hat{\alpha}-1)(\mathbf{s}^{\perp}\cdot\mathbf{v}) \\
&= \hat{\alpha}(\mathbf{v}^{\perp}\cdot\mathbf{s}) - (\hat{\alpha}-1)(\mathbf{v}^{\perp}\cdot\mathbf{s}) \\
&= \mathbf{v}^{\perp}\cdot\mathbf{s} \\
&= \mathtt{det}(\mathbf{v},\mathbf{s}).
\end{aligned}$$

□

This ends the groundwork for computing tangents in the horizontal plane. These results will be used to find the prevention bands for ranges of either ground speeds or track angles.

5.3 A Simple Formula for Horizontal Conflict

It is necessary for us to detect whether a given vector is in horizontal conflict (1). The following theorem provides the needed relationship between horizontal conflict and a simple formula that does not include a quantification over time. Instead, this theorem provides an equivalence between `horizontal_conflict?(s,v)` and the conjunction of two simple inequalities.

Theorem 5.6 (horizontal_solution).

$$\texttt{horizontal\_conflict?}(\mathbf{s},\mathbf{v}) \iff \mathbf{s}\cdot\mathbf{v} < R\,\texttt{det}(\mathbf{v},\mathbf{s}) < -\mathbf{s}\cdot\mathbf{v},$$

where, R is defined as in (13).

Proof. We note that another way to describe horizontal_conflict? is as a trajectory, which points towards the protected zone. Any trajectory that points towards the protected zone (and has not already entered the protected zone) will have entry and exit times into the protected zone—called Θ_- and Θ_+. These times are only defined when the discriminant of the quadratic (9) is greater than zero.

Now, we show that when the discriminate is greater than zero, then certain inequalities are satisfied. We follow a derivation similar to the development of (12).

$$\begin{aligned}
(\mathbf{s}\cdot\mathbf{v})^2 - \mathbf{v}^2(\mathbf{s}^2 - D^2) &> 0,\\
(\mathbf{s}\cdot\mathbf{v})^2 &> \mathbf{v}^2(\mathbf{s}^2 - D^2),\\
\mathbf{v}^2\mathbf{s}^2 - \texttt{det}(\mathbf{v},\mathbf{s})^2 &> \mathbf{v}^2(\mathbf{s}^2 - D^2),\\
D^2\mathbf{v}^2 &> \texttt{det}(\mathbf{v},\mathbf{s})^2,\\
D^2\mathbf{v}^2\mathbf{s}^2 - D^2\texttt{det}(\mathbf{v},\mathbf{s})^2 &> \texttt{det}(\mathbf{v},\mathbf{s})^2\mathbf{s}^2 - D^2\texttt{det}(\mathbf{v},\mathbf{s})^2,\\
D^2[\mathbf{v}^2\mathbf{s}^2 - \texttt{det}(\mathbf{v},\mathbf{s})^2] &> \texttt{det}(\mathbf{v},\mathbf{s})^2[\mathbf{s}^2 - D^2],\\
D^2(\mathbf{s}\cdot\mathbf{v})^2 &> \texttt{det}(\mathbf{v},\mathbf{s})^2[\mathbf{s}^2 - D^2],\\
(\mathbf{s}\cdot\mathbf{v})^2 &> \texttt{det}(\mathbf{v},\mathbf{s})^2\frac{\mathbf{s}^2 - D^2}{D^2},\\
(\mathbf{s}\cdot\mathbf{v})^2 &> R^2\,\texttt{det}(\mathbf{v},\mathbf{s})^2,\\
|\mathbf{s}\cdot\mathbf{v}| &> R\,|\texttt{det}(\mathbf{v},\mathbf{s})|,
\end{aligned}$$

which can be expanded into the following conjunction of inequalities

$$-|\mathbf{s}\cdot\mathbf{v}| < R\,\texttt{det}(\mathbf{v},\mathbf{s}) < |\mathbf{s}\cdot\mathbf{v}|.$$

The initial situation is in horizontal_conflict?, so $\mathbf{s}\cdot\mathbf{v}$ must be less than 0, thus

$$\mathbf{s}\cdot\mathbf{v} < R\,\texttt{det}(\mathbf{v},\mathbf{s}) < -\mathbf{s}\cdot\mathbf{v}.$$

The proof steps can be reversed to obtain the implication in the other direction. □

It is convenient to define

$$\texttt{horizontal\_criterion?}(\mathbf{s},\mathbf{v},\epsilon) \equiv \mathbf{s}\cdot\mathbf{v} \geq R\,\epsilon\,\texttt{det}(\mathbf{v},\mathbf{s}).$$

Using this definition we immediately obtain:

Theorem 5.7 (horizontal_criterion_independence).

$$\begin{aligned}
&\texttt{NOT horizontal_conflict?}(\mathbf{s},\mathbf{v}) \iff\\
&\quad \texttt{horizontal_criterion?}(\mathbf{s},\mathbf{v},-1)\ \texttt{OR}\\
&\quad \texttt{horizontal_criterion?}(\mathbf{s},\mathbf{v},+1).
\end{aligned}$$

6 Ground Speed Prevention Bands

A ground speed prevention band algorithm calculates the ground speeds of the ownship aircraft which will result in a conflict with another aircraft. Figure 10 gives

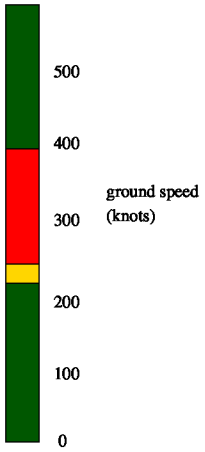

Figure 10. Ground Speed Prevention Bands

an example display of ground speed prevention information. As described earlier, we use two parameters to represent the lookahead times (T_{red}, T_{amber}) that are used to filter conflicts that are too far in the future. The red region in figure 10 indicates which ground speeds will result in a loss of separation within T_{red} while green indicates ground speeds that will not result in a loss of separation before T_{amber}. The bands algorithm varies the ground speed while holding vertical speed and track angle constant. For some ground speed the aircraft will be in conflict with the traffic and for other values it will not be, as illustrated in figure 11. One

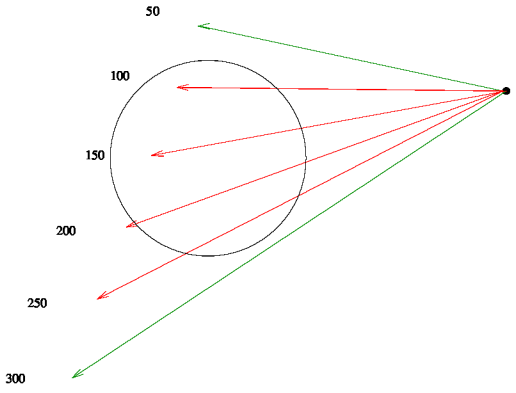

Figure 11. Ground Speed Trajectories

may recall that a change in the ownship's ground speed often results in a change in

the *direction* of the velocity vector in the relative frame of reference. The transition from green to red and vice versa occurs at the tangent vectors.

6.1 Encounter Geometry Analysis

Using the **Q** theory developed in section 5.2, we need to find a k such that

$$\lambda \mathbf{v}_o - \mathbf{v}_i = k\,\texttt{tangent\_line}(\mathbf{s}, \epsilon), \tag{18}$$

where $\lambda \mathbf{v}_o$ represents a change to (only) the ground speed of the ownship's velocity. Recall that `tangent_line` (figure 9) returns a vector that is tangent to the protected zone. Therefore, (18) represents a change to the ownship's ground speed, which results in a relative velocity vector that is tangent to the protected zone. We now wish to find a way to calculate k and λ assuming the term `tangent_line` is represented as $\mathbf{v}$.

Lemma 6.1 (`k_l_det`)**.**

$$\begin{aligned} &k\mathbf{v} + \mathbf{v}_i = \lambda \mathbf{v}_o \iff \\ &\quad k\,\texttt{det}(\mathbf{v}_o, \mathbf{v}) = \texttt{det}(\mathbf{v}_i, \mathbf{v}_o) \ \texttt{AND} \ \lambda\,\texttt{det}(\mathbf{v}_o, \mathbf{v}) = \texttt{det}(\mathbf{v}_i, \mathbf{v}). \end{aligned}$$

Proof. The components of $k\mathbf{v} + \mathbf{v}_i = \lambda \mathbf{v}_o$ are

$$\begin{aligned} kv_x + v_{ix} &= \lambda v_{ox}, \\ kv_y + v_{iy} &= \lambda v_{oy}. \end{aligned}$$

Multiplying both side with v_{oy} and v_{ox}, we obtain:

$$\begin{aligned} kv_x v_{oy} + v_{ix} v_{oy} &= \lambda v_{ox} v_{oy}. \\ kv_y v_{ox} + v_{iy} v_{ox} &= \lambda v_{oy} v_{ox}. \end{aligned}$$

Subtracting gives us:

$$k(v_x v_{oy} - v_y v_{ox}) + v_{ix} v_{oy} - v_{iy} v_{ox} = 0,$$

or

$$k\,\texttt{det}(\mathbf{v}, \mathbf{v}_o) = \texttt{det}(\mathbf{v}_o, \mathbf{v}_i),$$

which is equivalent to

$$k\,\texttt{det}(\mathbf{v}_o, \mathbf{v}) = \texttt{det}(\mathbf{v}_i, \mathbf{v}_o).$$

Using a similar derivation, yields the following relationship for λ:

$$\lambda\,\texttt{det}(\mathbf{v}_o, \mathbf{v}) = \texttt{det}(\mathbf{v}_i, \mathbf{v}).$$

□

```
gs_line_eps(s, v_o, v_i, ε) =
    v = tangent_line(s, ε)
    IF det(v_o, v) ≠ 0 THEN
        k = det(v_i, v_o) / det(v_o, v)
        λ = det(v_i, v) / det(v_o, v)
        IF λ > 0 THEN
            λ v_o
        ELSE
            (0, 0)
        ENDIF
    ELSE
        (0, 0)
    ENDIF
```

Figure 12. Ground Speed Scale Factor Computation

Therefore, if we instantiate $\mathbf{v}$ in this lemma with a known tangent vector—that is, the result from `tangent_line` in figure 9—we have the needed k that satisfies (18). With this result we can compute the tangent vector with the function `gs_line_eps` (figure 12). Note that if $\mathtt{det}(\mathbf{v}_o, \mathbf{v}) = 0$, then there is no ground speed change in the direction of $\mathbf{v}_o$ that produces a tangent. The $\lambda > 0$ test eliminates solutions that are in the opposite direction.

Using the definition of `gs_line_eps` in figure 12 we obtain two potential boundary ground speeds:

$$G_m = ||\mathtt{gs\_line\_eps}(\mathbf{s}, \mathbf{v}_o, \mathbf{v}_i, -1)||,$$
$$G_p = ||\mathtt{gs\_line\_eps}(\mathbf{s}, \mathbf{v}_o, \mathbf{v}_i, 1)||.$$

Either of these speeds may be undefined, that is, the `gs_line_eps` function returns a zero vector. If G_m is undefined, we set it to 0. If G_p is undefined, we set it equal to `max_gs`, the maximum ground speed to be displayed. With the definition of these two speeds, the display can be painted as follows:

$gs \leq G_m$	green
$G_m < gs < G_p$	red
$G_p \leq gs$	green

6.2 Lookahead Time Analysis

We now incorporate a finite lookahead time into the analysis of ground speed bands. Using the general outline from section 3.2, if the entire protected zone is inside the lookahead time, then the bands are easily computed using the technique from section 6.1. However, special techniques must be used if the lookahead time crosses the protected zone as show in figure 13. We must calculate the intersection of the

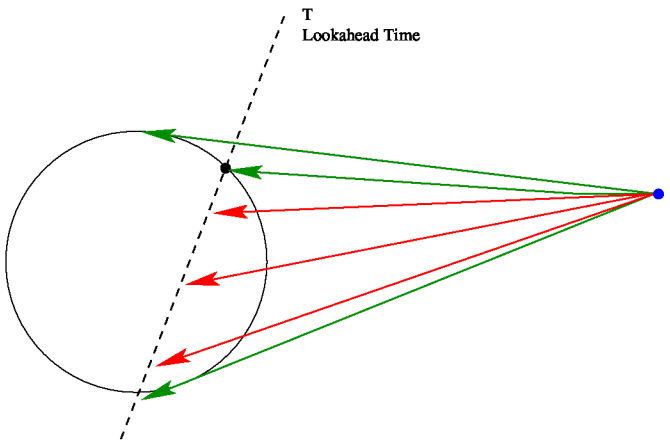

Figure 13. Ground Speed Trajectories with Lookahead Time

protected zone with the lookahead time. The relative velocity for a ground speed change is given by $\mathbf{v} = \lambda \mathbf{v}_o + \mathbf{v}_i$, where λ is a factor to adjust the ownship's ground speed and equals $\frac{\kappa}{||\mathbf{v}_o||}$, which is consistent with (7). To find the point where this velocity intersects the protected zone, we need to find a λ such that $||\mathbf{s} + T\mathbf{v}|| = D$, where T is the lookahead time.

$$
\begin{aligned}
&||\mathbf{s} + \mathbf{v}T|| = D \\
&(\mathbf{s} + \mathbf{v}T)^2 = D^2 \\
&(\mathbf{s} + (\lambda \mathbf{v}_o - \mathbf{v}_i)T)^2 = D^2
\end{aligned}
$$

If we let $\mathbf{W} = \mathbf{s} - T\mathbf{v}_i$, then

$$
\begin{aligned}
&(\mathbf{W} + \lambda \mathbf{v}_o T)^2 = D^2 \\
&(\mathbf{W} + \lambda \mathbf{v}_o T) \cdot (\mathbf{W} + \lambda \mathbf{v}_o T) = D^2
\end{aligned}
$$

Expanding we obtain a quadratic equation in λ with

$$a = T^2\,\mathbf{v}_o^2, \quad b = 2T\,(\mathbf{W} \cdot \mathbf{v}_o), \quad c = \mathbf{W}^2 - D^2.$$

When the discriminant is greater than or equal to zero, then the intersection between the circle and the lookahead time exists. We only want solutions where $\lambda \geq 0$. The function `gs_circle` (figure 14) solves this quadratic and returns a nonzero vector when an intersection point exists.

```
gs_circle(s, v_o, v_i, t, irt) =
    W = s − t v_i
    a = t^2 v_o^2
    b = 2t (W · v_o)
    c = W^2 − D^2
    IF discr(a, b, c) ≥ 0 THEN
        λ = root(a, b, c, irt)
        IF λ ≥ 0 THEN
            λ v_o
        ELSE
            (0, 0)
        ENDIF
    ELSE
        (0, 0)
    ENDIF
```

Figure 14. Ground Speed Circle Algorithm

6.3 Vertical Considerations

Up to this point we have ignored the impact of the aircraft's vertical speed. If the aircraft are in conflict horizontally but not vertically for a particular ground speed, then that ground speed should be allowed. Only if there is *both* horizontal and vertical conflict should that ground speed be prevented (i.e., painted red). For example, the aircraft could be vertically and horizontally separated, but the loss of separation occurs once the vertical standard is violated, as illustrated in figure 15. In this case the time to *enter* vertically is critical to the solution. Alternately, we

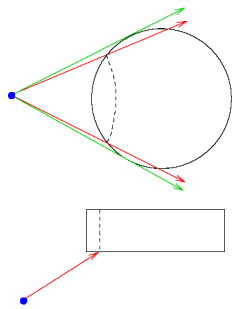

Figure 15. Vertical Entry Time Considerations

could have the same situation horizontally, but due to different vertical constraints, there is no conflict, as illustrated in figure 16. In this case, the time to *exit* vertically is important.

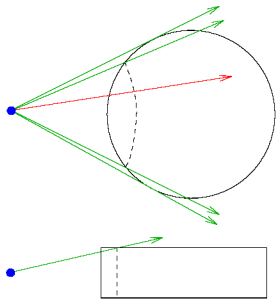

Figure 16. Vertical Exit Time Considerations

To address these situations, we must calculate the time to enter and exit the protected zone vertically:

$$t_{entry} = \frac{-\mathtt{sign}(v_z)H - s_z}{v_z}, \tag{19}$$

$$t_{exit} = \frac{\mathtt{sign}(v_z)H - s_z}{v_z}. \tag{20}$$

Even though the aircraft are currently vertically separated they will lose vertical separation after t_{entry}. Consequently, the vectors that intersect the circle at these vertical entry points are velocity vectors where separation may be lost. If the aircraft are currently not vertically separated, they may become separated at time t_{exit}. Therefore, the velocity vectors that intersect the circle at these vertical exit points may be vectors where separation is regained.

The ground speeds that corresponds to these times may be found using the Ground Speed Circle algorithm (figure 14) with $t_{entry/exit}$ for the time parameter. In some cases these times may be negative, meaning the aircraft has already entered the vertical protected zone or that the aircraft will never enter the protected zone vertically. In these cases, a constraint on the entry/exit time is ignored. Likewise, if the vertical speed (v_z) is zero, then these times are undefined and the corresponding constraint is ignored.

6.4 Sketch of a Ground Speed Algorithm

The basic idea of a ground speed algorithm is to create a sorted list of the ground speeds from the magnitude of the *non-zero* vectors representing the key points in

the encounter:

$$
\begin{aligned}
\mathbf{R}_m &= \texttt{gs\_line\_eps}(\mathbf{s}, \mathbf{v}_o, \mathbf{v}_i, -1),\\
\mathbf{R}_p &= \texttt{gs\_line\_eps}(\mathbf{s}, \mathbf{v}_o, \mathbf{v}_i, +1),\\
\mathbf{C}_{mr} &= \texttt{gs\_circle}(\mathbf{s}, \mathbf{v}_o, \mathbf{v}_i, T_{red}, -1),\\
\mathbf{C}_{pr} &= \texttt{gs\_circle}(\mathbf{s}, \mathbf{v}_o, \mathbf{v}_i, T_{red}, +1),\\
\mathbf{C}_{ma} &= \texttt{gs\_circle}(\mathbf{s}, \mathbf{v}_o, \mathbf{v}_i, T_{amber}, -1),\\
\mathbf{C}_{pa} &= \texttt{gs\_circle}(\mathbf{s}, \mathbf{v}_o, \mathbf{v}_i, T_{amber}, +1),\\
\mathbf{C}_{mn} &= \texttt{gs\_circle}(\mathbf{s}, \mathbf{v}_o, \mathbf{v}_i, t_{entry}, -1),\\
\mathbf{C}_{pn} &= \texttt{gs\_circle}(\mathbf{s}, \mathbf{v}_o, \mathbf{v}_i, t_{entry}, +1),\\
\mathbf{C}_{mx} &= \texttt{gs\_circle}(\mathbf{s}, \mathbf{v}_o, \mathbf{v}_i, t_{exit}, -1),\\
\mathbf{C}_{px} &= \texttt{gs\_circle}(\mathbf{s}, \mathbf{v}_o, \mathbf{v}_i, t_{exit}, +1).
\end{aligned}
$$

For completeness, 0 and the maximum ground speed are added to this list. A ground speed is chosen between every two sorted elements in this list and this ground speed is checked with a conflict probe like CD3D (see appendix A) to determine the color of the region. We do not offer a proof of correctness of this algorithm in this paper; this will be pursued in future work. A formal specification of the `gs_line_eps` function is presented in appendix B.1 and a formal specification of the `gs_circle` function is presented in appendix B.2.

7 Track Angle Prevention Bands

Using the definition of $\mathbf{v}_\alpha$ from (8), we define a *relative* velocity vector that is a function of α as

$$
\begin{aligned}
\mathbf{v} &= \mathbf{v}_\alpha - \mathbf{v}_i \\
&= (\omega \cos\alpha - v_{ix},\ \omega \sin\alpha - v_{iy},\ v_{oz} - v_{iz}),
\end{aligned}
\tag{21}
$$

where ω is the ground speed and is constrained by $\omega^2 = v_{ox}^2 + v_{oy}^2$. The boundaries of the track angle prevention regions are those values of α that produce relative velocity vectors that are tangent to the protected zone. In this section we explore three ways to find these angles.

Reference [5] describes an interesting situation when two prohibited regions are produced by a single aircraft pair. This situation is illustrated in the absolute reference frame by figure 17. As a check on our work, any equations we derive must be able to produce zero, two, or four α angles for certain encounter geometries.

This section begins by looking at multiple ways to find the boundary angles for a pair of aircraft only considering the encounter geometry. Then we develop inequality reasoning that enables us to properly identify which regions are the conflict regions and which are conflict-free. Finally, we examine solution methods which restrict the regions based on a lookahead time.

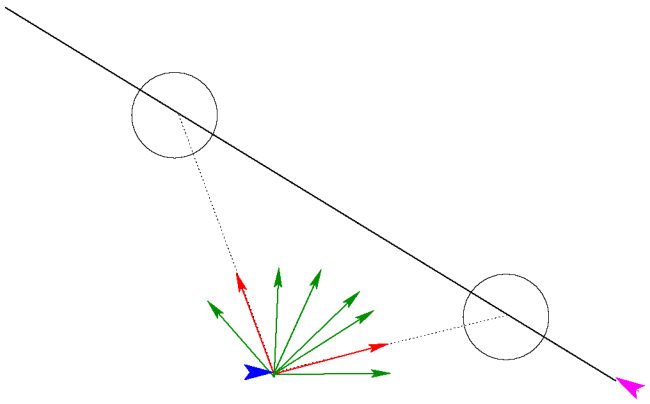

Figure 17. Two Prohibited Regions

7.1 Boundaries for Encounter Geometry Cases

We know that tangent velocity vectors must satisfy (9). The most obvious way to find these tangent vectors involves substituting the definition of $\mathbf{v}$ from (21) into (9) and solving for the track angle, α. This approach yields a complex formula involving $\mathtt{sin}\,\alpha$ and $\mathtt{cos}\,\alpha$. We can use the equality $\mathtt{cos}\,\alpha = \sqrt{1 - \mathtt{sin}^2\alpha}$ and get a formula involving just $\mathtt{sin}\,\alpha$, but to remove the square root terms, we will have a large, fourth order polynomial in $\mathtt{sin}\,\alpha$. However, if instead of using (9) directly, we use the simplification captured in `line_solution?` (15), we arrive at a quadratic function of α, whose solution is much more practical. We pursue multiple approaches to solve this equation. The first two approaches find the vector components of $\mathbf{v} = (v_x, v_y)$. Then by (21), we can obtain α, as follows[4]

$$\alpha = \mathtt{atan}(v_x + v_{ix}, v_y + v_{iy}). \tag{22}$$

In other words, we defer the use of trigonometry until the end of the derivation. Of the two approaches that use (22), one is based on the development of the $\mathbf{Q}$ vector in section 5.2. The other method finds the vector components directly and is called the *algebraic* approach. The final approach finds α directly through complex trigonometric reasoning. Not surprisingly, this approach is called the *trigonometric* approach. The first approach is the most mathematically succinct. The other approaches are presented in an attempt to be comprehensive.

7.1.1 Boundaries by Q-theory

From theorem 5.5, we know that $k\,\mathtt{tangent\_line}(\mathbf{s}, \epsilon)$ is a velocity vector that is tangent to the protected zone and therefore satisfies (15). Computing the boundary track angles involves finding a k that satisfies

$$\mathbf{v}_\alpha - \mathbf{v}_i = k\,\mathtt{tangent\_line}(\mathbf{s}, \epsilon). \tag{23}$$

[4]We use the arctangent, $\mathtt{atan}(a, b)$, as the arctangent of b/a. The arctangent function in some programming languages reverses the order of these parameters.

Once this k is found, the boundary angles for each ϵ can be found by (22). We claim that the needed k may be computed with the function in figure 18 provided the $\mathbf{v}'$ parameter is $\texttt{tangent\_line}(\mathbf{s}, \epsilon)$. This calculation masks the negative

$$
\begin{aligned}
&\texttt{calc\_k}(\mathbf{v}', \mathbf{v}_o, \mathbf{v}_i, \epsilon) = \\
&\quad a = (\mathbf{v}')^2 \\
&\quad b = 2(\mathbf{v}' \cdot \mathbf{v}_i) \\
&\quad c = \mathbf{v}_i^2 - \mathbf{v}_o^2 \\
&\quad \Delta = b^2 - 4ac \\
&\quad \texttt{IF } \Delta \geq 0 \texttt{ THEN } \frac{-b + \epsilon\sqrt{\Delta}}{2a} \\
&\qquad \texttt{ELSE } -1 \\
&\quad \texttt{ENDIF}
\end{aligned}
$$

Figure 18. Computation of k for Track Solutions

roots because they represent velocities in the opposite direction and hence they will never be used. To show this vector satisfies (23), it is sufficient to show that $k\,\texttt{tangent\_line}(\mathbf{s}, \epsilon) + \mathbf{v}_i$ is a velocity vector with the same speed as $\mathbf{v}_o$. Formally, this is stated as theorem 7.1.

Theorem 7.1 (`Calc_k`). *Letting k be* $\texttt{calc\_k}(\mathbf{v}', \mathbf{v}_o, \mathbf{v}_i, \texttt{irt})$, *then*

$$||k\,\mathbf{v}' + \mathbf{v}_i|| = ||\mathbf{v}_o||$$

when k is non-negative.

Proof. `calc_k` is a solution of the following quadratic equation:

$$
\begin{aligned}
&ak^2 + bk + c = 0, \\
&(\mathbf{v}')^2k^2 + 2(\mathbf{v}' \cdot \mathbf{v}_i)k + \mathbf{v}_i^2 = \mathbf{v}_o^2, \\
&(k\mathbf{v}' + \mathbf{v}_i) \cdot (k\mathbf{v}' + \mathbf{v}_i) = \mathbf{v}_o^2, \\
&(k\mathbf{v}' + \mathbf{v}_i)^2 = \mathbf{v}_o^2,
\end{aligned}
$$

or $||k\mathbf{v}' + \mathbf{v}_i|| = ||\mathbf{v}_o||$ as required. □

We know that `calc_k` will satisfy (23), provided $\mathbf{v}'$ is $\texttt{tangent\_line}(\mathbf{s}, \epsilon)$. We can combine these concepts may into a function (figure 19) to compute a velocity vector that is tangent to the protected zone.

$$
\begin{array}{l}
\texttt{track\_line}(\mathbf{s}, \mathbf{v}_o, \mathbf{v}_i, \epsilon, \texttt{irt}) = \\
\quad \text{LET } \mathbf{v}' = \texttt{tangent\_line}(\mathbf{s}, \epsilon), \\
\quad\quad k = \texttt{calc\_k}(\mathbf{v}', \mathbf{v}_o, \mathbf{v}_i, \texttt{irt}) \text{ IN} \\
\quad k\mathbf{v}'
\end{array}
$$

Figure 19. Computation of Track Line Solutions

7.1.2 Boundaries by Direct Algebra

Now we pursue an algebraic approach to solve for the vector components of $\mathbf{v}$. Starting from `line_solution?` (15), we have:

$$
\begin{aligned}
R\epsilon\,(\mathbf{s}^{\perp} \cdot \mathbf{v}) &= \mathbf{s} \cdot \mathbf{v}, \\
R\epsilon\,(s_x v_y - s_y v_x) &= s_x v_x + s_y v_y. \qquad (24)
\end{aligned}
$$

Assuming $v_x \neq 0$, we let $\rho = \frac{v_y}{v_x}$, we obtain:

$$
\begin{aligned}
&R\epsilon\,(s_x \rho v_x - s_y v_x) = s_x v_x + s_y \rho v_x, \\
&R\epsilon\,(s_x \rho - s_y) = s_x + s_y \rho, \\
&\rho = \frac{R\epsilon s_y + s_x}{R\epsilon s_x - s_y}.
\end{aligned}
$$

Since $\mathbf{v} + \mathbf{v}_i = \mathbf{v}_\alpha$ and $||\mathbf{v}_\alpha||^2 = ||\mathbf{v}_o||^2$, then following a development similar to that presented in reference [4],

$$
\begin{aligned}
&||\mathbf{v}_o||^2 = ||\mathbf{v} + \mathbf{v}_i||^2, \\
&\mathbf{v}_o^2 = \mathbf{v}^2 + 2(\mathbf{v} \cdot \mathbf{v}_i) + \mathbf{v}_i^2, \\
&0 = v_x^2(1 + \rho^2) + 2v_x(v_{ix} + \rho v_{iy}) + \mathbf{v}_i^2 - \mathbf{v}_o^2.
\end{aligned}
$$

This is a quadratic in v_x with:

$$
a = (1 + \rho^2), \quad b = 2(v_{ix} + \rho v_{iy}), \quad c = \mathbf{v}_i^2 - \mathbf{v}_o^2.
$$

The root of this quadratic will give a solution to v_x and from the definition of ρ we can obtain v_y. Since $\mathbf{v}_i$ is known, we can obtain $\mathbf{v}_\alpha$ from $\mathbf{v} = \mathbf{v}_\alpha - \mathbf{v}_i$. Four values of v_x are possible since each value of ϵ produces two ρ's and for each of these ρ's there are two solutions to the quadratic equation. Finally, we must address the special case when $v_x = 0$. From (24) we have

$$
R\epsilon\, s_x v_y = s_y v_y.
$$

If $R\epsilon\, s_x = s_y$, then any v_y can be used. If it is not equal, then $\mathbf{v}$ must be $(0, 0)$.

7.1.3 Boundaries by Trigonometry

This section presents a trigonometric approach to finding the region's boundaries. Using `line_solution?` (15) and $\mathbf{v} = \mathbf{v}_\alpha - \mathbf{v}_i$, we obtain:

$$\begin{aligned}
R\epsilon\,(\mathbf{s}^\perp \cdot \mathbf{v}) &= \mathbf{s} \cdot \mathbf{v}, \\
R\epsilon\,(\mathbf{s}^\perp \cdot (\mathbf{v}_\alpha - \mathbf{v}_i)) &= \mathbf{s} \cdot (\mathbf{v}_\alpha - \mathbf{v}_i), \\
R\epsilon\,(\mathbf{s}^\perp \cdot \mathbf{v}_\alpha) - R\epsilon\,(\mathbf{s}^\perp \cdot \mathbf{v}_i) &= \mathbf{s} \cdot \mathbf{v}_\alpha - \mathbf{s} \cdot \mathbf{v}_i, \\
R\epsilon\,(\mathbf{s}^\perp \cdot \mathbf{v}_\alpha) - \mathbf{s} \cdot \mathbf{v}_\alpha &= R\epsilon\,(\mathbf{s}^\perp \cdot \mathbf{v}_i) - \mathbf{s} \cdot \mathbf{v}_i, \\
\mathbf{v}_\alpha \cdot (R\epsilon\,\mathbf{s}^\perp - \mathbf{s}) &= \mathbf{v}_i \cdot (R\epsilon\,\mathbf{s}^\perp - \mathbf{s}).
\end{aligned}$$

Using (8), we have

$$(\omega \mathtt{cos}\alpha,\ \omega \mathtt{sin}\alpha) \cdot (-R\epsilon s_y - s_x,\ R\epsilon s_x - s_y) = \mathbf{v}_i \cdot (R\epsilon\,\mathbf{s}^\perp - \mathbf{s}).$$

Rearranging terms we have

$$\omega(R\epsilon s_x - s_y)\,\mathtt{sin}\,\alpha - \omega(R\epsilon s_y + s_x)\,\mathtt{cos}\,\alpha = \mathbf{v}_i \cdot (R\epsilon\,\mathbf{s}^\perp - \mathbf{s}).$$

Letting

$$E = \omega(R\epsilon s_x - s_y), \quad F = -\omega(R\epsilon s_y + s_x), \quad G = \mathbf{v}_i \cdot (R\epsilon\,\mathbf{s}^\perp - \mathbf{s}),$$

we have:

$$E\,\mathtt{sin}\,\alpha + F\,\mathtt{cos}\,\alpha = G. \tag{25}$$

Using a standard trigonometric identity (see appendix C), which is true as long as E and F are not both 0, we obtain

$$E\,\mathtt{sin}\,\alpha + F\,\mathtt{cos}\,\alpha = \sqrt{E^2 + F^2}\,\mathtt{sin}(\alpha + \mathtt{atan}(E, F)).$$

From which we get

$$\mathtt{sin}(\alpha + \mathtt{atan}(E, F)) = \frac{G}{\sqrt{E^2 + F^2}}.$$

If $\left|\frac{G}{\sqrt{E^2+F^2}}\right| \leq 1$ then in some 2π range, we have

$$\begin{aligned}
\alpha_1 &= \mathtt{asin}\left(\frac{G}{\sqrt{E^2 + F^2}}\right) - \mathtt{atan}(E, F), \\
\alpha_2 &= \pi - \mathtt{asin}\left(\frac{G}{\sqrt{E^2 + F^2}}\right) - \mathtt{atan}(E, F),
\end{aligned}$$

since $\mathtt{sin}(\alpha + \mathtt{atan}(E, F)) = \mathtt{sin}(\pi - \alpha - \mathtt{atan}(E, F))$. These two angles are the boundaries of the region. Since E, F, and G are all functions of ϵ, we have two pairs of α_1 and α_2 or four total angles. In specific cases, we may have fewer unique angles, for instance when $\left|\frac{G}{\sqrt{E^2+F^2}}\right| > 1$ or when the regions are adjacent (i.e., two α's are equal).

7.2 Regions for Encounter Geometry

We intuitively know that if the mid-angle of a region produces a conflict within T_{red} then all other angles in that region (between the boundary angles) will also produce conflicts within T_{red} in the encounter geometry case. We need to prove this formally. In other words, we seek a theorem of the form:

Theorem 7.2 (MidAngle).

$$
\begin{aligned}
&\alpha_1 < \alpha_m < \alpha_2 \ \texttt{AND} \\
&\alpha_m = \texttt{atan}(\mathbf{v}_m) \ \texttt{AND} \\
&\alpha_1 = \texttt{atan}(\mathbf{v}_1) \ \texttt{AND} \\
&\alpha_2 = \texttt{atan}(\mathbf{v}_2) \ \texttt{AND} \\
&\alpha = \texttt{atan}(\mathbf{v}_\alpha) \ \texttt{AND} \\
&\texttt{line\_solution?}(\mathbf{s}, \mathbf{v}_1) \ \texttt{AND} \\
&\texttt{line\_solution?}(\mathbf{s}, \mathbf{v}_2) \ \texttt{AND} \\
&||\mathbf{v}_m|| = ||\mathbf{v}_1|| = ||\mathbf{v}_2|| = ||\mathbf{v}_\alpha|| \ \texttt{IMPLIES} \\
&\quad \texttt{horizontal\_conflict?}(\mathbf{s}, \mathbf{v}_m) \\
&\qquad \iff \\
&\quad \forall \alpha : \alpha_1 < \alpha < \alpha_2 : \texttt{horizontal\_conflict?}(\mathbf{s}, \mathbf{v}_\alpha)
\end{aligned}
$$

where $\texttt{atan}(\mathbf{v}) = \texttt{atan}(v_x, v_y)$.

We have not yet been able to prove this theorem. Therefore, we proceeded with a more trigonometric approach (similar to the approach in section 7.2.1) that provides a rigorous basis for using the mid-angle and which also provides direct inequalities that delineate the regions.

7.2.1 Regions From Trigonometry

To determine color information for the regions between the boundaries, we develop trigonometric inequalities analogous to (25). We know from theorem horizontal_solution (theorem 5.6) that a velocity vector that satisfies

$$\mathbf{s} \cdot \mathbf{v} < r(\mathbf{s}^\perp \cdot \mathbf{v}) < -\mathbf{s} \cdot \mathbf{v}$$

is in horizontal conflict. Now we develop a theorem that relates these inequalities to serve as a condition for testing an angle α.

Theorem 7.3 (EFG).

$$
\begin{aligned}
&\mathbf{s}^2 - D^2 > 0 \implies \\
&\quad \mathbf{s} \cdot \mathbf{v} < R(\mathbf{s}^\perp \cdot \mathbf{v}) < -\mathbf{s} \cdot \mathbf{v} \\
&\qquad \iff (E_1 \sin\alpha + F_1 \cos\alpha > G_1 \ \texttt{AND} \ E_2 \sin\alpha + F_2 \cos\alpha < G_2)
\end{aligned}
$$

where

$$\begin{array}{lll} R = \frac{\sqrt{\mathbf{s}^2 - D^2}}{D}, & E_1 = \omega(Rs_x - s_y), & E_2 = \omega(Rs_x + s_y), \\ \omega = ||\mathbf{v}_o||, & F_1 = -\omega(Rs_y + s_x), & F_2 = \omega(-Rs_y + s_x), \\ \mathbf{v} = (\omega\cos\alpha,\ \omega\sin\alpha) - \mathbf{v}_i, & G_1 = \mathbf{v}_i \cdot (R\mathbf{s}^{\perp} - \mathbf{s}), & G_2 = \mathbf{v}_i \cdot (R\mathbf{s}^{\perp} + \mathbf{s}). \end{array}$$

Proof. Although this looks complicated, algebraic manipulation after substituting all definitions proves the result. □

From a prevention band standpoint, we want to know the range of track angles (range of α) that cause these linear combinations

$$E_1\sin\alpha + F_1\cos\alpha > G_1, \tag{26}$$
$$E_2\sin\alpha + F_2\cos\alpha < G_2, \tag{27}$$

to be true—so we can avoid them. These results are trigonometric and can be solved without regard to the specific values of E, F, and G developed in theorem 7.3. We begin with the following theorem which given an angle in a certain 2π range and the satisfaction of a linear combination, then we know the angle is in a more narrow range.

Theorem 7.4 (`angle_limit_ge`). *Assuming $E^2 + F^2 > G^2$ and some integer i,*

$$\begin{aligned} &2\pi i - \frac{\pi}{2} \le x \le 2\pi(i+1) - \frac{\pi}{2} \quad \texttt{AND} \\ &E\texttt{sin}(x - \texttt{atan}(E,F)) + F\texttt{cos}(x - \texttt{atan}(E,F)) \ge G \\ &\iff \\ &2\pi i + \texttt{narrow}(E,F,G,-1) \le x \le 2\pi i + \texttt{narrow}(E,F,G,1) \end{aligned}$$

where

$$\texttt{narrow}(E,F,G,\epsilon) = -\epsilon\,\texttt{asin}\left(\frac{G}{\sqrt{E^2+F^2}}\right) + \frac{\pi}{2}(\epsilon+1),$$
where, $\epsilon = 1$ *or* $\epsilon = -1$.

Proof. Forward implication case: From the linear combination theorem (see appendix C), we get

$$\begin{aligned} &E\texttt{sin}(x - \texttt{atan}(E,F)) + F\texttt{cos}(x - \texttt{atan}(E,F)) \\ &\quad = \sqrt{E^2+F^2}\ \texttt{sin}(x - \texttt{atan}(E,F) + \texttt{atan}(E,F)) \\ &\quad = \sqrt{E^2+F^2}\ \texttt{sin}(x). \end{aligned}$$

Then we can follow with this derivation

$$\begin{aligned} \sqrt{E^2+F^2}\ \texttt{sin}(x) &\ge G, \\ \texttt{sin}(x) &\ge \frac{G}{\sqrt{E^2+F^2}}, \\ \texttt{asin}(\texttt{sin}(x)) &\ge \texttt{asin}\left(\frac{G}{\sqrt{E^2+F^2}}\right). \end{aligned}$$

At this point we have two possibilities, depending on where the angle $x - 2\pi i$ falls in a range of values. If $-\frac{\pi}{2} \leq x - 2\pi i \leq \frac{\pi}{2}$ then, $\texttt{asin}(\texttt{sin}(x)) = x - 2\pi i$. So,

$$\begin{aligned} x - 2\pi i &\geq \texttt{asin}\left(\frac{G}{\sqrt{E^2 + F^2}}\right), \\ x &\geq \texttt{asin}\left(\frac{G}{\sqrt{E^2 + F^2}}\right) + 2\pi i, \\ x &\geq \texttt{narrow}(E, F, G, -1) + 2\pi i, \end{aligned} \tag{28}$$

or if $\frac{\pi}{2} \leq x - 2\pi i \leq \frac{3\pi}{2}$, then $\texttt{asin}(\texttt{sin}(x)) = \pi - (x - 2\pi i)$. So,

$$\begin{aligned} \pi - (x - 2\pi i) &\geq \texttt{asin}\left(\frac{G}{\sqrt{E^2 + F^2}}\right), \\ -(x - 2\pi i) &\geq \texttt{asin}\left(\frac{G}{\sqrt{E^2 + F^2}}\right) - \pi, \\ x - 2\pi i &\leq -\texttt{asin}\left(\frac{G}{\sqrt{E^2 + F^2}}\right) + \pi, \\ x - 2\pi i &\leq \texttt{narrow}(E, F, G, 1), \\ x &\leq \texttt{narrow}(E, F, G, 1) + 2\pi i. \end{aligned} \tag{29}$$

With these two conditions (28) and (29), the forward implication is proven.

Backward implication case: two standard trigonometric results state that

$$-\frac{\pi}{2} \leq a, b \leq \frac{\pi}{2} \quad \texttt{IMPLIES} \quad \texttt{sin}(a) > \texttt{sin}(b) \iff a > b, \tag{30}$$

$$\frac{\pi}{2} \leq a, b \leq \frac{3\pi}{2} \quad \texttt{IMPLIES} \quad \texttt{sin}(a) < \texttt{sin}(b) \iff a > b. \tag{31}$$

By combining these two results with the antecedent of the backward implication—$\texttt{narrow}(E, F, G, -1) \leq x - 2\pi i$ and $x - 2\pi i \leq \texttt{narrow}(E, F, G, 1)$—we can state

$$\begin{aligned} \texttt{sin}(x - 2\pi i) &> \texttt{sin}(\texttt{narrow}(E, F, G, -1)), \\ \texttt{sin}(x - 2\pi i) &> \texttt{sin}(\texttt{narrow}(E, F, G, 1)). \end{aligned}$$

Depending on the value of x, the preconditions of only one of (30) or (31) applies. However, which one is true doesn't matter because

$$\texttt{sin}(\texttt{narrow}(E, F, G, -1)) = \texttt{sin}(\texttt{narrow}(E, F, G, 1)) = \frac{G}{\sqrt{E^2 + F^2}}.$$

Note also, that $\texttt{sin}(x - 2\pi i) = \texttt{sin}(x)$. Thus,

$$\begin{aligned} \texttt{sin}(x - 2\pi i) &> \texttt{sin}(\texttt{narrow}(E, F, G, -1 \text{ or } 1)), \\ \texttt{sin}(x) &> \frac{G}{\sqrt{E^2 + F^2}} \\ \sqrt{E^2 + F^2}\,\texttt{sin}(x) &> G. \end{aligned}$$

We can transform this last formula into

$$E\mathtt{sin}(x - \mathtt{atan}(E,F)) + F\mathtt{cos}(x - \mathtt{atan}(E,F)) > G,$$

by the linear combination theorem (see appendix C) and the backward implication is proved. □

By similar reasoning we can prove

Theorem 7.5 (`angle_limit_le`). *Assuming* $E^2 + F^2 > G^2$ *and some integer* i,

$$\begin{aligned}
&2\pi i + \frac{\pi}{2} \le x \le 2\pi(i+1) + \frac{\pi}{2} \quad \mathtt{AND} \\
&E\mathtt{sin}(x - \mathtt{atan}(E,F)) + F\mathtt{cos}(x - \mathtt{atan}(E,F)) \le G \\
&\iff \\
&2\pi i + \mathtt{narrow}(E,F,G,1) \le x \le 2\pi(i+1) + \mathtt{narrow}(E,F,G,-1).
\end{aligned}$$

Now we need to collect these results (`angle_limit_ge`, `angle_limit_le`) to develop a range of angles that will satisfy (26). Since α represents the track angle of the aircraft, we know that its range is between 0 and 2π. If we let $x = \alpha + \mathtt{atan}(E,F)$, then the precondition of `angle_limit_ge` changes to

$$2\pi i - \frac{\pi}{2} - \mathtt{atan}(E,F) \le \alpha \le 2\pi(i+1) - \frac{\pi}{2} - \mathtt{atan}(E,F),$$

for some value of i. Since the range of $\mathtt{atan}(E,F)$ is also between 0 and 2π, this precondition can be satisfied over the full range of α for three values of i: 0, 1, and 2. With these three values of i and using `angle_limit_ge`, we get three ranges of α

$$\begin{aligned}
\mathtt{narrow}(E,F,G,-1) - \mathtt{atan}(E,F) \le &\quad \alpha \quad \le \mathtt{narrow}(E,F,G,1) - \mathtt{atan}(E,F), \\
2\pi + \mathtt{narrow}(E,F,G,-1) - \mathtt{atan}(E,F) \le &\quad \alpha \quad \le 2\pi + \mathtt{narrow}(E,F,G,1) - \mathtt{atan}(E,F), \\
4\pi + \mathtt{narrow}(E,F,G,-1) - \mathtt{atan}(E,F) \le &\quad \alpha \quad \le 4\pi + \mathtt{narrow}(E,F,G,1) - \mathtt{atan}(E,F).
\end{aligned}$$

The last condition is met trivially, since α cannot be greater than 2π. Eliminating this conditions yields the ranges

$$\begin{aligned}
\alpha &\ge \mathtt{narrow}(E,F,G,-1) - \mathtt{atan}(E,F), \\
\alpha &\le \mathtt{narrow}(E,F,G,1) - \mathtt{atan}(E,F),
\end{aligned} \tag{32}$$

$$\begin{aligned}
\alpha &\ge 2\pi + \mathtt{narrow}(E,F,G,-1) - \mathtt{atan}(E,F), \\
\alpha &\le 2\pi + \mathtt{narrow}(E,F,G,1) - \mathtt{atan}(E,F),
\end{aligned} \tag{33}$$

$$\begin{aligned}
\alpha &\ge 4\pi + \mathtt{narrow}(E,F,G,-1) - \mathtt{atan}(E,F), \\
\alpha &< 2\pi.
\end{aligned} \tag{34}$$

Thus if a particular angle is within any of these ranges, then we know that the linear combination formula is satisfied at that angle.

Using a similar development, the precondition to `angle_limit_le`

$$2\pi i + \frac{\pi}{2} - \texttt{atan}(E, F) \leq \alpha \leq 2\pi(i+1) + \frac{\pi}{2} - \texttt{atan}(E, F)$$

can cover the entire 0 to 2π range of α if i is -1, 0, and 1. From the `angle_limit_le` theorem, these three values of i yield the following three ranges of α

$$\begin{aligned} \alpha &\geq 0 \\ \alpha &\leq \texttt{narrow}(E, F, G, -1) - \texttt{atan}(E, F), \end{aligned} \tag{35}$$

$$\begin{aligned} \alpha &\geq \texttt{narrow}(E, F, G, 1) - \texttt{atan}(E, F), \\ \alpha &\leq 2\pi + \texttt{narrow}(E, F, G, -1) - \texttt{atan}(E, F), \end{aligned} \tag{36}$$

$$\begin{aligned} \alpha &\geq 2\pi + \texttt{narrow}(E, F, G, 1) - \texttt{atan}(E, F), \\ \alpha &\leq 4\pi + \texttt{narrow}(E, F, G, -1) - \texttt{atan}(E, F), \end{aligned} \tag{37}$$

which will ensure the linear combination is satisfied.

To summarize all these results, if an angle α satisfies one condition among (32), (33), or (34) and it also satisfies one condition from (35), (36), or (37), then we know that both theorems `angle_limit_ge` and `angle_limit_le` are satisfied, hence theorem 7.3 is satisfied, and thus we know that this angle results in a conflict in the horizontal dimension (theorem 5.6).

A few comments should be made about these ranges of angles. First, it is not necessarily the case that all three conditions are defined. For some values of $\texttt{atan}(E, F)$, a condition may be unsatisfiable for any value of α in the range 0 to 2π. Also, these conditions include an "equal to" and are not a strict $<$ or $>$ relation. The "equal to" parts of these conditions exactly correspond to the boundaries developed in section 7.1.3 and thus can be ignored. Finally, one may wonder why three ranges of α are provided with these conditions when figure 17 and the surrounding text states that only two regions are possible. This is only an apparent problem. The discrepancy comes from forcing the angle α in the range $0..2\pi$. A prevention region could begin near 2π and "wrap around" to angles larger than 0. Thus two contiguous regions could span all three conditions.

7.2.2 Mid-Angle Approach Revisited

From the previous section we know that

$$\begin{aligned} &\texttt{horizontal\_conflict?}(\mathbf{s}, \mathbf{v}_\alpha) \\ &\iff \\ &2\pi i - \frac{\pi}{2} - \texttt{atan}(E, F) \leq \alpha \leq 2\pi(i+1) - \frac{\pi}{2} - \texttt{atan}(E, F). \end{aligned}$$

Consider the function representing the separation distance between the two aircraft minus the diameter of the protected zone. This function is closely related to

`horizontal_conflict?`. We claim, but have not yet proven, that this function is continuous. With this assumption, we can apply the following lemma to this function.

Lemma 7.6 (`zeros_lem`). *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.*

The zeros of f satisfy `line_solution?`. By the above lemma, all angles between the zeros will all share the same conflict status (either red or green). Therefore, any angle between them (for our purposes, the mid-angle) can be used to determine how the region should be colored. We will use the mid-angle as a representative angle in following section.

7.3 Lookahead Time Analysis

In section 7.2, we developed solution techniques for prevention bands considering only the encounter geometry; now we consider ways to ignore those aircraft that are "too far" away. As described in section 3.2, a protected zone that cannot be reached within the lookahead time can be ignored. When all points of a protected zone may be reached within the lookahead time, then the solutions from the encounter geometry calculations may be used without modification (see section 7.1.1). It is only when the protected zone intersects the lookahead time that a special solution must be found, which is the purpose of this section. We will examine several approaches to solve this problem.

Each of these approaches uses a conflict probe to determine whether an angle is in conflict or not. One question is whether a 3-dimensional conflict problem (CD3D, see appendix A) should be used or if a strictly horizontal, 2-dimensional probe should be used instead. This question cannot be answered formally, but rather is based on human factors knowledge. Would a pilot, looking at a horizontal display, expect that separation must occur only in the horizontal dimension? If so, then a 2-dimensional probe is preferred. We will not pursue this question any further, but for convenience of presentation we will use CD3D from now on.

7.3.1 Conservative Boundary Approach

The easiest way to handle the case where only part of the protected zone is within the lookahead period is to be conservative. We begin with a region that creates a conflict when considering only the encounter geometry (see section 7.2.2). Next we find the time when the ownship will reach the tangent point (represented by Θ, as defined in (4) and 5) at each of the boundaries for this region. If either time is within the lookahead time, then the whole region is prevented (i.e., painted red).

The advantage of this approach is its simplicity—especially in terms of analysis; the disadvantage is that certain track angles are safe, but are painted red. Other approaches attempt to avoid losing many of these potential solutions.

7.3.2 Iterative Approaches

A variation on the full iterative approach (see section 2) is to step through the angles, starting at the boundaries from the encounter geometry solution and call CD3D at each angle. If we examine the angles

$$\alpha_1,\ \alpha_2,\ \ldots,\ \alpha_m,\ \ldots,\ \alpha_n,\ \ldots,\ \alpha_{end}$$

and we find that α_m is the first angle with a conflict and angle α_n is the last angle with a conflict, then we can conclude that all the angles from α_{m-1} to α_{n+1} should be prohibited.

A similar, but more efficient approach is instead of using a constant step size between angles, bisect the angles to find the boundaries of the region. This should execute in logarithmic time, instead of linear time.

7.3.3 Algebraic Approach

In this section we develop an algebraic method to *solve* for the points where the lookahead time, T, intersects the protected zone. We begin by finding the velocity vectors where the ownship will touch the protected zone at precisely the lookahead time, as illustrated by figure 20. Depending on the traffic's velocity, the lookahead

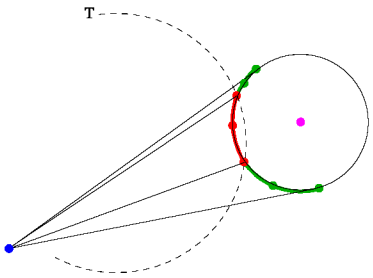

Figure 20. Track Angle with Lookahead Time

time may not be symmetric around the axis formed by the position of the two aircraft. The points where the lookahead time and the protected zone intersect are combined with the regional boundaries only considering the encounter geometry (section 7.1.1) and the midpoint between these points is tested with the conflict probe, CD3D. The result of the conflict probe is used to characterize the entire region (that is, to paint it red, amber, or green) between these points, as described in section 7.2.2. We observe that the intersection points between the lookahead time and the protected zone are precisely those points where the relative position vector at time T equals the diameter of the protected zone, or formally, when

$$||\mathbf{s} + T\mathbf{v}|| = D.$$

An equivalent formula, which is easier to solve is

$$||\mathbf{s} + T\mathbf{v}||^2 = D^2. \tag{38}$$

Since this equation involves a square, there will be two solutions for $\mathbf{v}$. Thus, to find the intersection points we must solve for the components of the vector $\mathbf{v}$ (i.e., v_x and v_y). With these components, the track angles from the ownship velocity vector, $\mathbf{v}_\alpha$, can be obtained by (22). The key theorem that enables us to compute these components is:

Theorem 7.7 (`TrackCircle`). *Let* $\mathbf{v} = \mathbf{v}_\alpha - \mathbf{v}_i$, $||\mathbf{v}_\alpha|| = ||\mathbf{v}_o||$, *and* T *be a time. Then* $||\mathbf{s}+T\mathbf{v}||^2 = D^2$, *provided* $v_{\alpha x}$ *is the solution of the quadratic equation* $Av_{\alpha x}^2 + Bv_{\alpha x} + C = 0$ *and* $\mathtt{sign}(-2P_y\,Tv_{\alpha y}) = \mathtt{sign}(E + 2P_x\,Tv_{\alpha x})$, *where*

$$\begin{array}{ll} \mathbf{P} = \mathbf{s} - T\mathbf{v}_i, & A = 4T^2\,||\mathbf{P}||^2, \\ E = ||\mathbf{P}||^2 + T^2\mathbf{v}_o^2 - D^2, & B = 4TE\,P_x, \\ & C = E^2 - 4\,P_y^2\,T^2\mathbf{v}_o^2. \end{array}$$

Proof. First, we establish that

$$||\mathbf{s} + T\mathbf{v}||^2 = D^2 \iff -2\,P_y\,Tv_{\alpha y} = E + 2\,P_x\,Tv_{\alpha x}.$$

This follows from substitution and simple algebraic manipulation:

$$\begin{array}{ll} ||\mathbf{s} + T\mathbf{v}||^2 = D^2, & \\ ||\mathbf{s} - T\mathbf{v}_i + T\mathbf{v}_\alpha||^2 = D^2, & \\ ||\mathbf{P} + T\mathbf{v}_\alpha||^2 = D^2, & \\ ||\mathbf{P}||^2 + 2T(\mathbf{P}\cdot\mathbf{v}_\alpha) + T^2\mathbf{v}_\alpha^2 = D^2, & \\ 0 = ||\mathbf{P}||^2 + T^2\mathbf{v}_o^2 - D^2 + 2T(\mathbf{P}\cdot\mathbf{v}_\alpha) & \text{recall: } \mathbf{v}_\alpha^2 = \mathbf{v}_o^2, \\ 0 = E + 2T(\mathbf{P}\cdot\mathbf{v}_\alpha), & \\ -2P_yTv_{\alpha y} = E + 2P_xTv_{\alpha x}. & (39) \end{array}$$

Next we square both sides of (39), to obtain

$$(-2\,P_y\,Tv_{\alpha y})^2 = (E + 2\,P_x\,Tv_{\alpha x})^2.$$

This step requires the condition $\mathtt{sign}(-2P_y\,Tv_{\alpha y}) = \mathtt{sign}(E + 2P_x\,Tv_{\alpha x})$. Next we substitute $v_{ox}^2 + v_{oy}^2 - v_{\alpha x}^2$ for $v_{\alpha y}^2$ (i.e., $||\mathbf{v}_\alpha|| = ||\mathbf{v}_o||$). This yields

$$4\,P_y^2\,T^2(v_{ox}^2 + v_{oy}^2 - v_{\alpha x}^2) = E^2 + 4\,P_x\,Tv_{\alpha x}E + 4\,P_x^2\,T^2v_{\alpha x}^2.$$

Solving for $v_{\alpha x}$ we obtain the quadratic with A, B, C as prescribed. □

We use the quadratic in the Track Circle theorem to obtain the two roots $v_{\alpha x}$. To complete the solution we solve for $v_{\alpha y}$ as follows

$$v_{\alpha y} = \pm\sqrt{v_{ox}^2 + v_{oy}^2 - v_{\alpha x}^2}.$$

Now we must decide whether to choose the positive root or the negative root for $v_{\alpha y}$. To satisfy the preconditions to Track Circle, we choose the sign of $v_{\alpha y}$ so that

$$\mathtt{sign}(-2\,P_y\,Tv_{\alpha y}) = \mathtt{sign}(E + 2\,P_x\,Tv_{\alpha x})$$

$$
\begin{aligned}
&\texttt{track\_circle}(\mathbf{s}, \mathbf{v}_o, \mathbf{v}_i, t, \texttt{irt}) = \\
&\qquad \mathbf{P} = \mathbf{s} - t\mathbf{v}_i, \\
&\qquad e = \mathbf{P}^2 + t^2\mathbf{v}_o^2 - D^2, \\
&\qquad a = 4t^2\mathbf{P}^2, \\
&\qquad b = 4t\, e\, P_x, \\
&\qquad c = e^2 - 4\, P_y^2\, t^2\, \mathbf{v}_o^2 \\
&\qquad \texttt{IF } \texttt{discr}(a, b, c) \geq 0 \texttt{ THEN} \\
&\qquad\quad v'_{ox} = \texttt{root}(a, b, c, \texttt{irt}), \\
&\qquad\quad \texttt{IF } \texttt{sign}(-P_y) = \texttt{sign}(e + 2\, P'_x, t\, v'_{ox}) \texttt{ THEN } \xi = 1 \\
&\qquad\quad \texttt{ELSE } \xi = -1 \\
&\qquad\quad \texttt{IF } \mathbf{v}_o^2 - (v'_{ox})^2 > 0 \texttt{ THEN} \\
&\qquad\qquad v'_{oy} = \xi\sqrt{\mathbf{v}_o^2 - (v'_{ox})^2} \\
&\qquad\qquad (v'_{ox}, v'_{oy}) \\
&\qquad\quad \texttt{ELSE} \\
&\qquad\qquad (0, 0) \\
&\qquad\quad \texttt{ENDIF} \\
&\qquad \texttt{ELSE} \\
&\qquad\quad (0, 0) \\
&\qquad \texttt{ENDIF}
\end{aligned}
$$

Figure 21. Track Circle Function

is satisfied. In this way, the two roots for $v_{\alpha x}$ with their companion $v_{\alpha y}$ give us two potential solutions, say $\mathbf{v}_{\alpha_1}$ and $\mathbf{v}_{\alpha_2}$. The function in figure 21 provides these solutions. This function produces a zero vector when the quadratic equation from the Track Circle theorem cannot be solved. The meaning of a zero vector in these cases is that the lookahead time boundary and the protected zone do not intersect. In these cases, only the boundaries from the encounter geometry are used, unless the protected zone is too far away, then it is ignored all together. The PVS version of the `track_circle` algorithm is presented in appendix B.4.

The `track_circle` function finds both entry points and exit points to the protected zone. In many situations only one of these solutions are needed. Figure 22 illustrates a situation where only the entry point solution of the `track_circle` func-

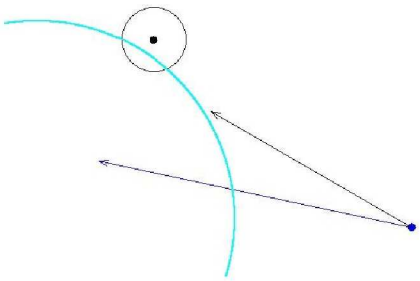

Figure 22. Exit Point from Track Circle

tion is relevant. This figure is in the translated frame of reference where the traffic aircraft (black dot) is at the origin. The blue dot shows the position of the ownship and the blue vector its current trajectory. The black circle shows the protected zone around the traffic. The cyan circle shows all possible relative locations at the lookahead time, if the ownship were to change only its track angle. The cyan circle intersects the protected zone in two places, but only the entry point (right-most intersection) affects the protected bands. The net result is that only one side of the bands is affected, as illustrated in figure 23. The green-red circle shows what the prevention bands would look like if the entire region between the tangent points are colored red. The outer blue-red-amber bands show the impact of using Track Circle. The amber region shows the part of the bands, which is no longer red because it falls outside of the finite lookahead time. The algorithm that is presented in this paper is not harmed by the inclusion of the extra solutions (e.g. the exit point in the above scenario). However, it is essential that no critical point is omitted. We have decided to not filter out the extra solutions (and potentially gain a little more efficiency) in our first version of the algorithm. After a fully formal proof has been constructed, we may filter out the extraneous solutions.

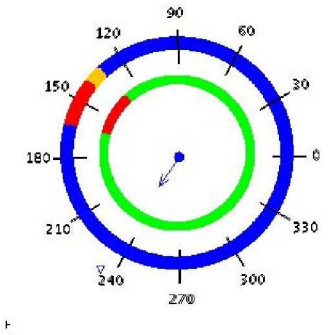

Figure 23. Bands with Exit Point

7.3.4 Trigonometric Approach

Like the algebraic approach, the trigonometric approach begins with a lookahead time T and (38), then proceeds in a similar way to the Track Circle theorem.

$$\begin{aligned}
&||\mathbf{s} + T\mathbf{v}||^2 = D^2, \\
&||\mathbf{s} - T\mathbf{v}_i + T\mathbf{v}_\alpha||^2 = D^2, \\
&||\mathbf{P} + T\mathbf{v}_\alpha||^2 = D^2, \\
&\mathbf{P}^2 + 2T(\mathbf{P} \cdot \mathbf{v}_\alpha) + T^2\mathbf{v}_\alpha^2 = D^2,
\end{aligned}$$

where $\mathbf{P} = \mathbf{s} - T\mathbf{v}_i$. Using $\mathbf{v}_\alpha = (\omega \cos\alpha, \omega \sin\alpha)$ and $\mathbf{v}_\alpha^2 = \omega^2$, we can write

$$\begin{aligned}
&2T(\mathbf{P} \cdot \mathbf{v}_\alpha) = D^2 - ||\mathbf{P}||^2 - T^2\omega^2, \\
&2TP_x\omega \cos\alpha + 2TP_y\omega \sin\alpha = D^2 - ||\mathbf{P}||^2 - T^2\omega^2.
\end{aligned}$$

We can now cast this formula in the form of (25),

$$E \cos\alpha + F \sin\alpha = G,$$

where

$$E = 2TP_x\omega, \quad F = 2TP_y\omega, \quad G = D^2 - ||\mathbf{P}||^2 - T^2\omega^2.$$

We can apply the linear combination theorem (see appendix C, pg. 56) in a manner similar to section 7.1.3, and get

$$\sin(\alpha + \operatorname{atan}(E, F)) = H,$$

where

$$H = \frac{G}{\sqrt{E^2 + F^2}}.$$

Thus, if $|H| \leq 1$ then in some 2π range, we have

$$\begin{aligned}
\alpha_1 &= \operatorname{asin}(H) - \operatorname{atan}(E, F), \\
\alpha_2 &= \pi - \operatorname{asin}(H) - \operatorname{atan}(E, F),
\end{aligned}$$

since $\sin(\alpha + \operatorname{atan}(E, F)) = \sin(\pi - \alpha - \operatorname{atan}(E, F))$. Just as in the algebraic case (section 7.3.3), we must evaluate these angles to ensure they yield entry points at the lookahead time.

7.4 Vertical Considerations

Similar to section 6.3, we have ignored the impact of the aircraft's vertical speed. If the aircraft are in conflict horizontally but not vertically for a particular track angle, the prevention band should be green at that angle. Only if there is both horizontal and vertical conflict should that track angle be prevented (i.e., painted red). We can use (19) and (20) to find the entry time and exit times into the vertical protected zone. To avoid unnecessarily prohibiting angles, we must ensure that all track angles that occur before the t_{entry} are allowed. Furthermore, if t_{exit} is before the time to reach the tangent point, then all track angles between these times should be allowed. The track angle that corresponds to these times may be found using the Track Circle Theorem (theorem 7.7) and `track_circle` function in figure 21 (see also algorithm `trk_circle`, presented in appendix B.4). The parameter T of the theorem is instantiated with $t_{entry/exit}$.

7.5 Sketch of a Track Angle Algorithm

The idea of the algorithm is to find the critical track angles from the encounter geometry, the lookahead time, and the vertical considerations. Then sort this list of angles and use the conflict probe (CD3D, see appendix A) at the angle between each of the critical angles to characterize the region (i.e., determine which color the region should be painted: green, amber, or red). These critical vectors may be computed as

$$
\begin{aligned}
\mathbf{R}_{mm} &= \texttt{track\_line}(\mathbf{s}, \mathbf{v}_o, \mathbf{v}_i, -1, -1),\\
\mathbf{R}_{mp} &= \texttt{track\_line}(\mathbf{s}, \mathbf{v}_o, \mathbf{v}_i, -1, +1),\\
\mathbf{R}_{pm} &= \texttt{track\_line}(\mathbf{s}, \mathbf{v}_o, \mathbf{v}_i, +1, -1),\\
\mathbf{R}_{pp} &= \texttt{track\_line}(\mathbf{s}, \mathbf{v}_o, \mathbf{v}_i, +1, +1),\\
\mathbf{C}_{rm} &= \texttt{track\_circle}(\mathbf{s}, \mathbf{v}_o, \mathbf{v}_i, T_{red}, -1),\\
\mathbf{C}_{rp} &= \texttt{track\_circle}(\mathbf{s}, \mathbf{v}_o, \mathbf{v}_i, T_{red}, +1),\\
\mathbf{C}_{am} &= \texttt{track\_circle}(\mathbf{s}, \mathbf{v}_o, \mathbf{v}_i, T_{amber}, -1),\\
\mathbf{C}_{ap} &= \texttt{track\_circle}(\mathbf{s}, \mathbf{v}_o, \mathbf{v}_i, T_{amber}, +1),\\
\mathbf{C}_{em} &= \texttt{track\_circle}(\mathbf{s}, \mathbf{v}_o, \mathbf{v}_i, t_{entry}, -1),\\
\mathbf{C}_{ep} &= \texttt{track\_circle}(\mathbf{s}, \mathbf{v}_o, \mathbf{v}_i, t_{entry}, +1),\\
\mathbf{C}_{xm} &= \texttt{track\_circle}(\mathbf{s}, \mathbf{v}_o, \mathbf{v}_i, t_{exit}, -1),\\
\mathbf{C}_{xp} &= \texttt{track\_circle}(\mathbf{s}, \mathbf{v}_o, \mathbf{v}_i, t_{exit}, +1).
\end{aligned}
$$

Some of these vectors may be zero vectors in which case they are ignored. Then the track angles are found using the `atan` function. Finally, to provide appropriate bounding, the angles 0 and 2π must be added.

We do not offer a proof of correctness of this algorithm in this paper. This will be pursued in future work.

8 Conclusions

The mathematics underlying conflict prevention systems is more subtle than expected. A direct approach (for the bands of track angles) easily carries one into the domain of fourth-order polynomial equations in $\sin\alpha$. In this paper, we have elaborated several different mathematical approaches to find regions of prevention bands. These approaches roughly fall into three categories: iterative, algebraic, and trigonometric. The mathematical development divides the problem into encounter geometry considerations, which incorporate conflicts from all aircraft, and lookahead time considerations, which filter conflicts based on their nearness (in time) to the ownship. Since the lookahead time is parametrized as T, the same mathematical development can be used to examine at immediate warnings (colored red), and near-term warnings (colored amber).

Using this strong mathematical basis, we intend to formally verify algorithms to implement these mathematical formulas. Initial algorithms have been sketched out, but have not been verified. In future work we will select from the methods provided in this paper to develop an efficient algorithm for the generation of prevention bands. At this point we favor the **Q**-theory algebraic approach because we see this as the most efficient approach and the most mathematically succinct. In addition, we expect the formal proofs of the algorithm will not be overly difficult. From an implementation standpoint, an iterative method may be preferred; however, as we mentioned earlier, simple iterative solutions such as presented in section 2 have both correctness and completeness issues. Due to safety concerns the correctness issues must be addressed and for efficiency reasons the completeness issues must be evaluated to determine their impact.

The non-iterative approaches set forth in this paper are limited to cases where only two aircraft are involved. As alluded to in section 3, an algorithm is needed to merge the regions of angles from each traffic aircraft into a single collection of conflict prevention information. Future work involves specifying, verifying, and implementing this algorithm.

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

1. Bryan 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.

2. N.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.

3. G. Dowek, C. Muñoz, and V. Carreño. Provably safe coordinated strategy for distributed conflict resolution. In *Proceedings of the AIAA Guidance Navigation, and Control Conference and Exhibit 2005, AIAA-2005-6047*, San Francisco, California, 2005.

4. A. Geser, C. Muñoz, G. Dowek, and F. Kirchner. Air traffic conflict resolution and recovery. Technical Report NASA/CR-2002-211637, ICASE-NASA Langley, NASA Langley Research Center, Hampton VA 23681-2199, USA, May 2002.

5. 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.

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

7. S. 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.

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

Appendix A

Conflict Detection in 3D Space: CD3D

This appendix describes a formally verified three dimensional conflict detection algorithm called CD3D. The algorithm works on a pair of aircraft: the ownship and a single traffic aircraft. By methodically calling this function for each traffic aircraft and combining the outputs, this algorithm may be used as the basis of a conflict probe. CD3D returns `TRUE` if and only if there is a time t less than the lookahead time T, where there is both vertical and horizontal loss of separation. To estimate future states of the aircraft, this algorithm performs a linear extrapolation of state information (position and velocity). The algorithm is decomposed into two cases:

- $v_z = 0$, i.e., no vertical speed
- $v_z \neq 0$

In the first case, the problem is essentially 2-dimensional. The second case is more difficult. We will develop some preliminary lemmas before we undertake these two cases.

A.1 Two Dimensional Preliminaries

The mathematics underlying horizontal 2D conflict detection is the intersection of the ownship's trajectory with the protected zone (a circle). As mentioned above, this problem is equivalent to finding if there is a time, t, where the trajectory is within the protected zone. A quadratic equation to find this time is

$$t^2\mathbf{v}^2 + 2(\mathbf{s}\cdot\mathbf{v})\,t + \mathbf{s}^2 - D^2 = 0,$$

with

$$a = \mathbf{v}^2, \quad b = \mathbf{s}\cdot\mathbf{v}, \quad c = \mathbf{s}^2 - D^2.$$

This equation is the same as (3) and the derivation of this equation is presented in section 4.1. The roots of this equation give us the exit and entry times into the protected zone. We repeat (4) and (5), which represent these times:

$$\Theta_- = \frac{-b - \sqrt{d}}{a}, \quad \Theta_+ = \frac{-b + \sqrt{d}}{a},$$

where d is the discriminant and equals $b^2 - ac$. The following expanded form of the discriminant is used by CD3D:

$$\begin{aligned} d &= (\mathbf{s}\cdot\mathbf{v})^2 + \mathbf{v}^2(\mathbf{s}^2 - D^2) \\ &= (s_x v_x + s_y v_y)^2 + \mathbf{v}^2 D^2 - (v_x^2 + v_y^2)(s_x^2 + s_y^2) \\ &= 2(s_x s_y v_x v_y) + \mathbf{v}^2 D^2 - (s_x^2 v_y^2 + v_x^2 s_y^2). \end{aligned}$$

This equation is equivalent to the form presented in (9) and the further development of this equation is not necessary for the conflict detection problem. The conflict *resolution* problem—represented by the development after (9)—attempts to find the velocity vector $\mathbf{v}$, where in the conflict *detection* problem, the vector $\mathbf{v}$ is known and is equal to $\mathbf{v}_o - \mathbf{v}_i$.

If the discriminant is less than or equal to zero, then the ownship's trajectory does not intersect the horizontal protected zone and there is no conflict. Thus, a condition for conflict detection is that $d > 0$.

We now proceed to develop the logic of CD3D for two cases: if $v_z = 0$ and if $v_z \neq 0$.

A.2 When $v_z = 0$

When $v_z = 0$, CD3D reduces to a 2-dimensional problem. For CD3D to return `TRUE`, first we must ensure that we have a conflict in the vertical dimension, formally, that

$$-H < s_z < H.$$

Next, we must ensure that $d > 0$. Finally, we determine that the lookahead time, T, is after the entrance time into the protected zone and that the exit time is not in the past. Formally, these conditions are $\Theta_- < T$ and $\Theta_+ > 0$. We claim that these conditions are satisfied by the following conditions:

$$\left(D^2 > \mathbf{s}^2 \ \texttt{OR} \ b \leq 0\right) \ \texttt{AND} \ \left(d > (aT+b)^2 \ \texttt{OR} \ (aT+b) \geq 0\right).$$

This relationship is proved by the lemmas A.1 and A.3. The lemmas are stated with a generic time t, because they will be used slightly differently in the next section.

Lemma A.1 (`theta1_lt_t`). *If the discriminant is non-negative, then*

$$\Theta_- < t \iff d > (at+b)^2 \ \texttt{OR} \ at+b \geq 0.$$

Proof.

$$\begin{aligned}
&\Theta_- < t \iff \\
&\frac{-b-\sqrt{d}}{a} < t \iff \\
&-\sqrt{d} < at+b \iff \\
&d > (at+b)^2 \ \texttt{OR} \ at+b \geq 0.
\end{aligned}$$

□

To resolve the second condition, we need to determine if $\Theta_+ > 0$. The proof of this result requires the proof of the next lemma.

Lemma A.2 (`theta2_gt_t`). *If the discriminant is non-negative, then*

$$d > (at+b)^2 \ \texttt{OR}\ at+b \le 0 \iff \Theta_+ > t.$$

Proof.

$$\begin{aligned}
&\Theta_+ > t \iff \\
&\frac{-b+\sqrt{d}}{a} > t \iff \\
&\sqrt{d} > at+b \iff \\
&d > (at+b)^2 \ \texttt{OR}\ at+b \le 0.
\end{aligned}$$

□

Lemma A.3 (`theta2_gt_0`). *If the discriminant is non-negative, then*

$$D^2 > \mathbf{s}^2 \ \texttt{OR}\ b \le 0 \iff \Theta_+ > 0.$$

Proof. Using lemma `theta2_gt_t` with $t = 0$, we have

$$d > b^2 \ \texttt{OR}\ b \le 0 \iff \Theta_+ > 0.$$

But the first condition $d > b^2$ can only occur when there is a loss of separation, i.e., $\mathbf{s}^2 < D^2$:

$$\begin{aligned}
&d > b^2 \iff \\
&b^2 - ac > b^2 \iff \\
&-ac > 0 \iff \\
&\mathbf{v}^2\left[\mathbf{s}^2 - D^2\right] < 0 \iff \\
&\mathbf{s}^2 - D^2 < 0 \iff && \text{since } \mathbf{v}^2 > 0, \\
&\mathbf{s}^2 < D^2.
\end{aligned}$$

□

A.3 When $v_z \neq 0$

If $v_z \neq 0$, then we must deal with both the two dimensional considerations in the previous section and we must also deal with the times when loss of separation occurs vertically. We repeat (19) and (20) that give the times to enter and exit the vertical protected zone:

$$\begin{aligned}
t_{entry} &= \frac{-\mathtt{sign}(v_z)H - s_z}{v_z}, \\
t_{exit} &= \frac{\mathtt{sign}(v_z)H - s_z}{v_z}.
\end{aligned}$$

If either of these times is negative, then this time is undefined; for instance, when an aircraft is below the protected zone and its vertical speed is negative. Furthermore, since $H > 0$, $t_{entry} < t_{exit}$.

This problem breaks into two cases: a special case is when the ground speed is zero ($v_x = 0$ and $v_y = 0$) and the general case when the ground speed is non-zero ($v_x \neq 0$ or $v_y \neq 0$). Recognize that the speed here is the relative ground speed between the two aircraft, so the special case is physically realistic. We handle the special case first.

A.3.1 When $v_x = 0$ and $v_y = 0$

In this case there can only be a conflict if there is already a loss of separation horizontally, so the test $\mathbf{s}^2 < D^2$ is performed. If there is also a vertical loss of separation (i.e., $-H < s_z$ AND $s_z < H$) then there is a loss of separation, which—by definition—is a conflict. Otherwise, the vertical velocity will cause a conflict within the lookahead time, provided

$$0 < t_{exit} \text{ AND } t_{entry} < T.$$

By expanding the definition of t_{entry}, and performing algebraic manipulations, we have the conditions:

$$\begin{aligned} &v_z > 0 \text{ AND } s_z < 0 \text{ AND } -H < s_z + Tv_z \text{ OR} \\ &v_z < 0 \text{ AND } s_z > 0 \text{ AND } H > s_z + Tv_z. \end{aligned}$$

A.3.2 When $v_x \neq 0$ or $v_y \neq 0$

The general case involves all four key times associated with detection:

- Θ_-, the time to enter protected zone horizontally
- Θ_+, the time to leave protected zone horizontally
- t_{entry}, the time to enter protected zone vertically
- t_{exit}, the time to leave protected zone vertically

If these times are properly ordered then there is a conflict. First we must ensure that both a horizontal and vertical loss of separation occurs within the lookahead time, that is:

$$\begin{aligned} &0 < t_{exit} \text{ AND } t_{entry} < T, \\ &0 < \Theta_+ \text{ AND } \Theta_- < T. \end{aligned}$$

Finally, we must ensure that the two intervals of time overlap. This can be ensured with

$$\Theta_- < t_{exit} \text{ AND } t_{entry} < \Theta_+.$$

To summarized these six conditions

Condition	is equivalent to	by
$t_{entry} < T$		
$t_{exit} > 0$		
$\Theta_+ > 0$	$D^2 > s_x^2 + s_y^2$ `OR` $b \leq 0$	lemma `theta2_gt_0`
$T > \Theta_-$	$d > (aT + b)^2$ `OR` $aT + b \geq 0$	lemma `theta1_lt_t`
$t_{exit} > \Theta_-$	$d > (at_2 + b)^2$ `OR` $at_2 + b \geq 0$	lemma `theta1_lt_t`
$t_{entry} < \Theta_+$	$d > (at_1 + b)^2$ `OR` $at_1 + b \leq 0$	lemma `theta2_gt_t`

These are precisely the conditions that are checked in CD3D when $v_z \neq 0$.

A.4 CD3D Algorithm

The CD3D algorithm, expressed in the PVS specification language, is shown in figure A1.

```
s : VAR Vect3   %% Relative position of the ownship wrt traffic
v : VAR Vect3   %% Relative velocity of the ownship wrt traffic
D : VAR posreal %% Minimum horizontal separation
H : VAR posreal %% Minimum vertical separation
T : VAR posreal %% Lookahead time

CD3D(s,v,D,H,T) : bool =
  LET (sx,sy,sz) = s,
      (vx,vy,vz) = v IN
  IF  vx=0 AND vy=0 AND sx*sx+sy*sy < D*D THEN  %% No horizontal movement
      (-H < sz AND sz < H) OR                   %% Already in conflict
      (vz > 0 AND sz < 0 AND -H < T*vz + sz) OR %% Vert. conflict in future
      (vz < 0 AND sz > 0 AND  H > T*vz + sz)
  ELSE
    LET d = 2*sx*vx*sy*vy + D*D*(vx*vx + vy*vy) -
            (sx*sx*vy*vy + sy*sy*vx*vx) IN
    IF d > 0 THEN
      LET a = vx*vx + vy*vy,
          b = sx*vx + sy*vy IN
      %% theta1 = (-b - sqrt(d))/a, first intersection with D
      %% theta2 = (-b + sqrt(d))/a, second intersection with D
      IF vz = 0 THEN  %% Horizontal movement only
        -H < sz AND sz < H AND
        (D*D > sx*sx + sy*sy OR b <= 0) AND   %% theta2 > 0
        (d > (a*T+b) * (a*T+b) OR a*T+b >= 0) %% theta1 < T
      ELSE  %% General case
        LET t1 = (-sign(vz)*H-sz)/vz,
            t2 = (sign(vz)*H-sz)/vz IN
        %% t1 < t2
        (d > (a*t2+b) * (a*t2+b) OR a*t2+b >= 0) AND %% theta1 < t2
        (d > (a*t1+b) * (a*t1+b) OR a*t1+b <= 0) AND %% t1 < theta2
        %% max(theta1,t1) < min(theta2,t2)
        (D*D > sx*sx + sy*sy OR b <= 0)          AND %% theta2 > 0
        t2 > 0                                   AND %% min(theta2,t2) > 0
        (d > (a*T+b) * (a*T+b) OR a*T+b >= 0)    AND %% theta1 < T
        t1 < T                                       %% max(theta1,t1) < T
      ENDIF
    ELSE FALSE
    ENDIF
  ENDIF
```

Figure A1. CD3D Algorithm

Appendix B

Horizontal Algorithms

In this section the PVS versions of the horizontal algorithms are provided, These algorithms are abstract and are defined over the reals. We have not considered the issues associated with the finite precision provided by a floating point implementations, which are used in conventional programming languages such as C++ or Java. The algorithms are defined as functions with a highly constrained return type. In other words, the return types convey the key properties of the function. These return types rely on a few additional predicates:

```
v1,v2     : VAR Vect2

gs_only?(v1)(v2) : bool = EXISTS (l:posreal): v2 = l*v1
trk_only?(v1)(v2) : bool =  norm(v2) = norm(v1)
nz_vect2?(v: Vect2): bool = v /= zero
```

The following types are also used:

```
Sp_vect2 : TYPE = {s: Vect2 | sqv(s) >= sq(D)}
Ss_vect2 : TYPE = {s: Vect2 | sqv(s) > sq(D)}
Nz_vect2 : TYPE = {v: Vect2 | v /= zero}
Nzv2_vect3 : TYPE = {v | nz_vect2?(v)}
```

In addition to these definitions, the algorithms rely on utility functions providing vector operations and the roots of quadratic equations, see sections B.5 and B.6.

B.1 Ground Speed Line Algorithm

The ground speed line algorithm is defined as follows in PVS:

```
sp        : VAR Sp_vect2
ss        : VAR Ss_vect2
eps       : VAR Sign
vo,vi,v,
nvo,nvi   : VAR Vect2
nzv       : VAR Nz_vect2

alpha(ss) : {x:posreal | x < 1} =  sq(D)/sqv(ss)

beta(ss)  : nnreal =  D*sqrt(sqv(ss)-sq(D))/sqv(ss)

Qx(ss,eps) : MACRO real = alpha(ss)*ss`x + eps*beta(ss)*ss`y

Qy(ss,eps) : MACRO real = alpha(ss)*ss`y - eps*beta(ss)*ss`x
```

```
Q(ss,eps)  : Vect2 = (Qx(ss,eps),Qy(ss,eps))

Qs(ss,eps) : Vect2 =
  Q(ss,eps)-ss

tangent_line(sp,eps) : Nz_vect2 =
  IF on_D?(sp) THEN
    eps*perpR(sp)
  ELSE
    Qs(sp,eps)
  ENDIF

gs_only_line(v,vo,vi): {k:real,nvo:Vect2 | nz_vect2?(nvo) =>
                                           gs_only?(vo)(nvo) AND
                                           k*v = nvo-vi} =
  IF det(vo,v) /= 0 THEN
    LET k = det(vi,vo)/det(vo,v),
        l = det(vi,v)/det(vo,v) IN
    IF l > 0 THEN
      (k,l*vo)
    ELSE
      (0,zero)
    ENDIF
  ELSE
    (0,zero)
  ENDIF

gs_line_eps(sp,vo,vi,eps): {nvo | nz_vect2?(nvo) =>
                                  gs_only?(vo)(nvo)} =
  LET (k,nvo) = gs_only_line(tangent_line(sp,eps),vo,vi) IN
  IF nz_vect2?(nvo) AND k >= 0 THEN
    nvo
  ELSE
    zero
  ENDIF
```

B.2 Ground Speed Circle Algorithm

The ground speed circle algorithm is defined as follows in PVS:

```
s,nvo     : VAR Vect3
vo,vi     : VAR Nzv2_vect3
dir,irt   : VAR Sign
vnzo      : VAR Nz_vect2
d,t       : VAR posreal
```

```
gs_only_circle(s,vnzo,vi,d,t,irt): {nvo | nz_vect2?(nvo) =>
                                          gs_only?(vnzo)(nvo) AND
                                          sqv(s+t*(nvo-vi)) = sq(d)} =
  LET w = s-t*vi,
      a = sq(t)*sqv(vnzo),
      b = t*(w*vnzo),
      c = sqv(w)-sq(d) IN
  IF discr2b(a,b,c) >= 0 THEN
    LET l = root2b(a,b,c,irt) IN
    IF l > 0 THEN
      l*vnzo
    ELSE
      zero
    ENDIF
  ELSE
    zero
  ENDIF
```

B.3 Track Line Algorithm

The track line algorithm is defined as follows in PVS:

```
sp         : VAR Sp_vect2
u,vnz,vnzo : VAR Nz_vect2
eps,irt    : VAR Sign
vo,vi,nvo  : VAR Vect2


trk_only_line(vnz,vo,vi,irt): {k:real,nvo:Vect2 |
                               nz_vect2?(nvo) => trk_only?(vo)(nvo) AND
                                                 k*vnz = nvo-vi} =
  LET a = sqv(vnz),
      b = vnz*vi,
      c = sqv(vi) - sqv(vo) IN
  IF discr2b(a,b,c) >= 0 THEN
    LET k = root2b(a,b,c,irt) IN
    (k,k*vnz+vi)
  ELSE
    (0,zero)
  ENDIF

trk_line_eps_irt(sp,vo,vi,eps,irt): {nvo:Vect2 | nz_vect2?(nvo) =>
                                                 trk_only?(vo)(nvo)} =
  LET (k,nvo) = trk_only_line(tangent_line(sp,eps),vo,vi,irt) IN
  IF nz_vect2?(nvo) AND k >= 0 THEN
    nvo
  ELSE
    zero
  ENDIF
```

B.4 Track Circle Algorithm

The track circle algorithm is defined as follows in PVS:

```
s,nvo   : VAR Vect3
vo,vi   : VAR Nzv2_vect3
dir,irt : VAR Sign
t       : VAR posreal

horizontal_dir?(s: Vect2, v: Vect2,eps) : MACRO bool =
  eps*(s*v) >= 0

vertical_dir?(sz: real,vz: real, eps) : MACRO bool =
  eps*(sz*vz) >= 0

trk_circle(s,vo,vi,t,irt,eps): {nvo | nz_vect2?(nvo) =>
                                      trk_only?(vo)(nvo)} =
 LET P = s - t*vi,
     e = sqv(P) + sq(t)*sqv(vo) - sq(D),
     a = 4*sq(t)*sqv(P),
     b = 4*t*e*P`x,
     c = sq(e) - 4*sq(P`y)*sq(t)*sqv(vo) IN
 IF solvable?(a,b,c) THEN
   LET nvox = solution(a,b,c,irt),
       sgny = IF sign(-P`y) = sign(e+2*P`x*t*nvox) THEN 1 ELSE -1 ENDIF
   IN
   IF sqv(vo) - sq(nvox) > 0 THEN
     LET nvoy = sgny*sqrt(sqv(vo)-sq(nvox)) IN
     (nvox,nvoy)
   ELSE
     zero
   ENDIF
 ELSE
   zero  %% zero indicates no solution
 ENDIF
```

B.5 Vectors

The NASA PVS library[B1] contains two theories that define 2D vectors and 3D vectors, named `vectors_2D` and `vectors_3D`. These theories introduce the following record types.

```
Vect2: TYPE = [#  x, y: real  #]

Vect3: TYPE = [#  x, y, z: real  #]
```

Given the following variable declarations

```
v: VAR Vect2
w: VAR Vect3
```

[B1] Available at `http://shemesh.larc.nasa.gov/fm/ftp/larc/PVS-library/pvslib.html`.

the components of the vectors `v` and `w` are accessed using a ‘ character: `v‘x`, `v‘y`, `w‘x`, `w‘y`, and `w‘z`. The standard operations on vectors are defined: if `u` and `v` are vectors and `a` is a scalar, `u + v`, `-u`, `u - v`, `u * v`, and `a*v` denote addition, negation, subtraction, dot product, and scalar multiplication, respectively. We also define

```
sq(v)   : nnreal = sq(v‘x) + sq(v‘y)
sqv(v)  : nnreal = v*v
norm(v) : nnreal = sqrt(sq(v))
```

where `nnreal` is the type of non-negative real numbers. The zero vectors are defined as follows

```
zero: Vect2 = (0, 0)

zero: Vect3 = (0, 0, 0)
```

A `vect2D` function is provided that creates a 2D vector from a 3D vector by truncation, i.e., throwing away the z-component:

```
vect2D(v:Vect3): Vect2 = (v‘x,v‘y)
```

B.6 Roots

The `quadratic` theory in the `reals` library defines the following functions

```
sq(x: real) = x*x

quadratic(a,b,c: real)(x): real = a*sq(x) + b*x + c

discr(a,b,c) : real = sq(b) - 4*a*c

root(a:nonzero_real,b:real,c:real|discr(a,b,c)>=0,eps:Sign):real =
  (-b + eps*sqrt(discr(a,b,c)))/(2*a)

discr2b(a,b,c)  : real = sq(b) - a*c

root2b(a:nonzero_real,b:real,c:real|discr2b(a,b,c)>=0,eps:Sign):real =
  (-b + eps*sqrt(discr2b(a,b,c)))/a
```

Appendix C

Linear Combination

A linear combination of sine and cosine is a useful identity used throughout the trigonometric proofs.

Theorem C.1 (`linear combination`). *A linear combination of sine and cosine can be expressed with a single sine function, such that*

$$E\mathtt{sin}\,\alpha + F\mathtt{cos}\,\alpha = \sqrt{E^2+F^2}\ \mathtt{sin}(\alpha + \mathtt{atan}(E,F))$$

Provided that either E or F is not zero—the arctangent function is undefined in this case. The function $\mathtt{atan}(E,F)$ is defined as the arctangent of F/E.

Proof. The proof involves trigonometric reasoning. We proceed by transforming the right side of the equality into the left side.

$$\begin{aligned}
&\sqrt{E^2+F^2}\,\mathtt{sin}(\alpha + \mathtt{atan}(E,F)) \\
&\sqrt{E^2+F^2}\,(\mathtt{sin}(\alpha)\mathtt{cos}(\mathtt{atan}(E,F)) + \mathtt{cos}(\alpha)\mathtt{sin}(\mathtt{atan}(E,F)))
\end{aligned}$$

Using standard trigonometric identities,

$$\sqrt{E^2+F^2}\left(\mathtt{sin}(\alpha)\frac{\mathtt{sign}(E)}{\sqrt{1+\left(\frac{F}{E}\right)^2}} + \mathtt{cos}(\alpha)\frac{\frac{F\mathtt{sign}(E)}{E}}{\sqrt{1+\left(\frac{F}{E}\right)^2}}\right)$$

Then we use algebraic manipulations,

$$\begin{aligned}
&\sqrt{E^2+F^2}\left(\frac{\mathtt{sin}(\alpha)\mathtt{sign}(E) + \mathtt{cos}(\alpha)\frac{F\mathtt{sign}(E)}{E}}{\sqrt{1+\left(\frac{F}{E}\right)^2}}\right) \\
&\sqrt{E^2+F^2}\left(\frac{\mathtt{sin}(\alpha)\mathtt{sign}(E) + \mathtt{cos}(\alpha)\frac{F\mathtt{sign}(E)}{E}}{\frac{\sqrt{E^2+F^2}}{\mathtt{sign}(E)E}}\right) \\
&\mathtt{sign}(E)E\left(\mathtt{sin}(\alpha)\mathtt{sign}(E) + \mathtt{cos}(\alpha)\frac{F\mathtt{sign}(E)}{E}\right) \\
&E\mathtt{sin}\,\alpha + F\mathtt{cos}\,\alpha
\end{aligned}$$

since, $(\mathtt{sign}(E))^2 = 1$. □

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
01-06-2009	Technical Memorandum	

4. TITLE AND SUBTITLE
A Mathematical Basis for the Safety Analysis of Conflict Prevention Algorithms

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Maddalon, Jeffrey M.; Butler, Ricky W.; Muñoz, César A.; Dowek, Gilles

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER
411931.02.51.07.01

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
NASA Langley Research Center
Hampton, VA 23681-2199

8. PERFORMING ORGANIZATION REPORT NUMBER
L-19699

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Washington, DC 20546-0001

10. SPONSOR/MONITOR'S ACRONYM(S)
NASA

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
NASA/TM-2009-215768

12. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified - Unlimited
Subject Category 03
Availability: NASA CASI (443) 757-5802

13. SUPPLEMENTARY NOTES

14. ABSTRACT

In air traffic management systems, a conflict prevention system examines the traffic and provides ranges of guidance maneuvers that avoid conflicts. This guidance takes the form of ranges of track angles, vertical speeds, or ground speeds. These ranges may be assembled into prevention bands: 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 all maneuvers. Without conflict prevention information, a pilot might perform a maneuver that causes a near-term conflict. Because near-term conflicts can lead to safety concerns, strong verification of correct operation is required. This paper presents a mathematical framework to analyze the correctness of algorithms that produce conflict prevention information. This paper examines multiple mathematical approaches: iterative, vector algebraic, and trigonometric. The correctness theories are structured first to analyze conflict prevention information for all aircraft. Next, these theories are augmented to consider aircraft which will create a conflict within a given lookahead time. Certain key functions for a candidate algorithm, which satisfy this mathematical basis are presented; however, the proof that a full algorithm using these functions completely satisfies the definition of safety is not provided.

15. SUBJECT TERMS
Air traffic; Avoidance; Conflict; Detection; Resolution

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			STI Help Desk (email: help@sti.nasa.gov)
					19b. TELEPHONE NUMBER *(Include area code)*
U	U	U	UU	67	(443) 757-5802