NASA/TP–2004–213015

Formal Verification of a Conflict Resolution and Recovery Algorithm

Jeffrey Maddalon and Ricky Butler
Langley Research Center, Hampton, Virginia

Alfons Geser and César Muñoz
National Institute of Aerospace, Hampton, Virginia

April 2004

The NASA STI Program Office . . . 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 Office plays a key part in helping NASA maintain this important role.

The NASA STI Program Office is operated by Langley Research Center, the lead center for NASA's scientific and technical information. The NASA STI Program Office provides access to the NASA STI Database, the largest collection of aeronautical and space science STI in the world. The Program Office is also NASA's institutional mechanism for disseminating the results of its research and development activities. These results are published 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 that complement the STI Program Office's diverse offerings include creating custom thesauri, building customized databases, organizing and publishing research results . . . even providing videos.

For more information about the NASA STI Program Office, 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 (301) 621–0134
- Phone the NASA STI Help Desk at (301) 621–0390
- Write to:
  NASA STI Help Desk
  NASA Center for AeroSpace Information
  7121 Standard Drive
  Hanover, MD 21076–1320

NASA/TP–2004–213015

Formal Verification of a Conflict Resolution and Recovery Algorithm

Jeffrey Maddalon and Ricky Butler
Langley Research Center, Hampton, Virginia

Alfons Geser and César Muñoz
National Institute of Aerospace, Hampton, Virginia

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23681–2199

April 2004

The use of trademarks or names of manufacturers in this report is for accurate reporting and does not constitute an offical endorsement, either expressed or implied, of such products or manufacturers by the National Aeronautics and Space Administration.

Available from:

NASA Center for AeroSpace Information (CASI)
7121 Standard Drive
Hanover, MD 21076–1320
(301) 621–0390

National Technical Information Service (NTIS)
5285 Port Royal Road
Springfield, VA 22161–2171
(703) 605–6000

Contents

Abbreviations

ADS-B Automatic Dependent Surveillance-Broadcast

ATM Air Traffic Management

ATSP Air Traffic Service Provider

CD&R Conflict Detection and Resolution

CPU Central Processing Unit

CTAS Center TRACON Automation System

DAG-TM Distributed Air-Ground Traffic Management

FAA Federal Aviation Administration

GPS Global Positioning System

ICAO International Civil Aviation Organization

NASA National Aeronautics and Space Administration

PVS Prototype Verification System

RNAV area navigation

RNP Required Navigation Performance

RR3D name of a three dimensional resolution and recovery algorithm described in [1]

TRACON Terminal Radar Approach Control

URET User Request Evaluation Tool

1 Introduction

Air Traffic Management (ATM) has two fundamental objectives: provide safe separation between aircraft and maximize the efficiency of the airspace system. Today, the responsibility to maintain appropriate traffic separation resides in a central human authority within each sector, the Air Traffic Service Provider (ATSP). The ATSP monitors the airspace, issues clearances to all controlled aircraft in the sector, and expects the aircraft to follow these clearances. In the current system, as traffic levels approach capacity, efficiency is sacrificed for safety and there is little room for user preferences. Novel approaches to ATM, e.g., Distributed Air-Ground Traffic Management (DAG-TM) [2, 3] and Free-flight [4, 5], address capacity problems of the current airspace system by distributing the responsibility for traffic separation among specially-equipped aircraft in the airspace. In these approaches, on-board hardware and ATM software provide surveillance information, alerting for possible loss of separation, and advisories for corrective maneuvers.

On-board conflict detection, resolution, and recovery systems are critical components of new ATM concepts. *Conflict detection* determines if the path of the aircraft conflicts with any other aircraft. *Conflict resolution* creates a new path that avoids conflicts with other aircraft. *Conflict recovery* creates a path to guide the aircraft back to its original destination. The algorithm examined in this paper combines conflict resolution and recovery.

Safety assessment of the correctness of an ATM algorithm amounts to verifying that for every possible scenario, conflicts are detected and effectively resolved. Traditionally, this is done via testing, human-in-the-loop simulations, and flight experiments. The traditional techniques are not sufficient for a comprehensive safety assessment given the enormous number of interactions present in this new distributed environment. Testing, simulations and flight experiments are still valuable for defining requirements, assessing feasibility, and gaining experience with safety and efficiency issues. Some limitations of these techniques for safety assessment include:

- Simulations can only represent phenomena that have been specifically modeled.
- Biased selection of scenarios may limit the correctness of any generalized claims made from a collection of simulation results.
- Flight experiments are too expensive to obtain statistically significant results.
- The set of possible scenarios is too large to obtain reasonable coverage with testing, simulation, and experimentation.

In this paper we propose the first critical step—algorithm verification—in a formal approach to the safety assessment of future ATM systems; we then provide an extended example of this step. *Formal* indicates that the model of the ATM system and its properties are stated unambiguously by mathematical formulae, and that all claims are accompanied by rigorous proofs. When the formal proof is checked by a computer program we refer to this as a *mechanically checked proof* or a *mechanical verification.*[1]

As an illustration of this approach, the formal verification of an algorithm for air traffic conflict resolution and recovery, called RR3D [1], is presented. A conflict resolution and recovery algorithm can be considered a state-based geometric conflict detection and resolution (CD&R) algorithm that satisfies arrival time constraints, see [6]. Such an algorithm may be seen as a building block for strategic conflict resolution [7]. In [1] Geser, et al. present a proof of the RR3D algorithm; this paper formalizes this proof in the mechanical verification system PVS [8]. A proof that has not been mechanically verified may contain non-obvious errors that are difficult for humans to recognize. A proof that is checked by a computer ensures every detail of the proof is throughly examined.

This paper is organized as follows. Section 2 discusses the rationale for a formal safety assessment methodology. Section 3 presents an overview of CD&R modeling techniques. Section 4 introduces the resolution and recovery algorithm RR3D. In Section 5, RR3D serves as a case study for our formal approach to safety analysis. Section 6 summarizes the paper and discusses future research directions. Appendix A.1 lists minor errors and missing assumptions in the original proof. Appendix A.2 includes additional lemmas used in the verification. Appendix A.3 maps the notations used in this document to the textual representation in PVS.

2 Rationale for Formal Assessment of ATM Systems

Digital avionics systems have been used since the early 1970's. A fly-by-wire aircraft such as the Boeing 777 employs safety-critical software in the flight control computers. This type of software is largely derived from control theory based on rigorous mathematical methods that provide assurance of key properties such as stability. Moreover, the basic stability of the aircraft provides protection from occasional glitches in the control software.

On the ground side, most of the software associated with ATM is packaged into decision support tools for air traffic controllers, e.g., Center TRACON Automation System (CTAS) [9] and User Request Evaluation Tool (URET) [10]. This software provides information to controllers in a convenient format to aid them in managing the trajectories of the aircraft in their sector. The failure of this software is mitigated by human intelligence that has many sources of information about the aircraft under ATM control including the analog display of radar data. Consequently, the safety risk resides primarily in the human controllers. The main question to be asked about such software is whether the software helps the controllers achieve their operational goals. This question is best answered by qualitative human-factors oriented,

[1]A computer program that checks proofs is called a *theorem prover.* A theorem prover rigorously enforces the rules of mathematical logic and ensures that every step of the proof follows directly from primitive inference rules of the logic. Traditional mathematical proofs are checked through a social peer-review process which over decades identifies any errors in these proofs. Since proofs of software systems are inherently tedious and uninteresting, a social process is not feasible. Therefore, we rely on theorem provers to discover errors in our proofs.

statistical assessments.

Future ATM concepts under development will utilize software in ways that are fundamentally different from the past. Many of these concepts move the safety risk directly into executing software. A near-term influence is the ICAO's (International Civil Aviation Organization) Required Navigation Performance (RNP) initiative. RNP-based area navigation (RNAV) extends the capabilities of modern airplanes by providing more accurate and precise navigation capability leading to more flexible airspace routes and procedures in both visual and instrument conditions. Although the RNP-based RNAV system should provide greater accuracy, it will necessarily rely on more sophisticated on-board software and external infrastructure such as Global Positioning System (GPS) and their associated augmentations. In RNP-based RNAV environments the safety risk associated with ATM may migrate from radar and controllers to on-board software and critical technologies, such as GPS, that are also dependent upon software systems. Software consequently may have new safety implications because it can fail in ways that cannot be mitigated by a human. Hence, it is reasonable to re-examine the methods by which we determine that software is correct and reliable.

The safety assessment of ATM systems cannot be accomplished using simulation and experimentation alone. To verify that a system containing safety-critical software is safe, one must ensure that either there are no sequences of inputs that encounter a hazard-inducing bug in the software or that any errors due to non-verified sequences of inputs are mitigated by system level mechanisms. Unfortunately, the state space of complex systems is intractably large. The input space must cover the 3-D airspace in the vicinity of an aircraft and all possible pilot inputs. Even if these are discretized, the number of test cases that must be examined to cover the input domain would require millions of years of experimentation.[2] Extensive simulation can only establish that selected states, from the enormous set of possible states, are safe. It is unrealistic to infer that *all* states, or that *most* states, are also safe. The case is even worse with flight experiments. The number of input cases covered is so minuscule that its usefulness for this purpose is virtually nil. Hence the idea that a simulation or a flight experiment can establish the safety of an air traffic management concept must be rejected. A complete coverage of the set of possible states and a rigorous assessment of safety properties is only possible through a *complete mathematical proof.* The purpose of this paper is to elaborate this type of approach. Within this approach, simulation and flight experiments serve a critical new role in formal safety assessment, as we will point out below.

It is impossible to guarantee that an ATM system, like any physical system, works perfectly. There are too many unpredictable elements: changing weather, system failures, human errors, etc. It has been argued that it is impossible to achieve any guarantee about the behavior of an ATM system, and hence that a formal analysis of an ATM system is pointless. We disagree with this generalization.

[2]For example, even a tiny program consisting of five 10-bit inputs and ten 10-bit internal variables has 2^{150} states. If each state could be tested in one microsecond, then complete testing would require $4.5 \times 10^{+31}$ years.

Indeed formal techniques can guarantee that an algorithm is correct[3] for all possible scenarios *under reasonable, well-defined assumptions*. As we will explain later, this set of assumptions is a by-product of the formal verification process. We claim that a formal verification is an essential step in the validation process of avionics systems.

Traditional engineering practice involves making predictions about an extremely complex and unpredictable environment. This is accomplished by bringing mathematical rigor to the system's domain as much as possible, thus minimizing uncertainty in the system. Because software systems are intrinsically mathematical, one might think that there are no unpredictable elements in them. But, the behavior of embedded computer systems is dependent on *assumptions* about the environment in which the system operates and the *logic* contained within the system. If the behavior of the computer system is incorrect then either the assumptions or the logic must be incorrect. Formal verification ensures that the logic is correct but does not address the validity of assumptions. However, formal verification does provide a comprehensive list of assumptions and a framework wherein experts can validate these assumptions. A formally verified system may still fail, but only if the assumptions were not valid.[4] It is therefore critical to validate the assumptions on which the system was built. This requires experienced, technical judgment. Human inspection, flight experiments and simulation can provide this validation. For ATM systems, extensive simulations must be conducted to establish that the operational procedures that govern the new airspace concept are adequate to sustain the assumptions that go into the formal analysis of the software algorithms. Flight experiments must also be performed to corroborate the assumptions of the simulations (such as the effects of winds, dynamics, datalink behavior, etc). A flight experiment provides an essential capability by uncovering shortcomings and errors in the assumptions. When problems are discovered in flight, the formal analysis must be adjusted to reflect the different characteristics of the environment, or the operational procedures must be modified in order to rule-out the discovered problem.

A credible safety case for an advanced ATM system will be a massive endeavor. It should be noted that much of the current ATM research is based upon comparative studies. In other words, a new concept is promoted by comparing it to an existing capability rather than rigorously establishing that the concept achieves specific safety and efficiency objectives. The reason for this is that establishing objective, absolute safety and efficiency properties is extremely difficult. The following is only a rudimentary list of some of the key characteristics of a comprehensive safety case.

- All of the requirements for safety have been captured and expressed in a rigorous manner.

- Verifiable algorithms and designs have been used whose behavior is fully explicated via mathematical theorems.

[3]By *correct* we mean there is a mathematical specification of the algorithm's intended functionality and for all possible inputs it provides that functionality.

[4]By a formally verified system we mean that not only the algorithm has been shown to be correct, but its refinement into software has also been shown to be correct.

- Software programs have been developed in accordance with certification standards, such as DO-178B, and shown to be faithful implementations of the formally verified algorithms using code-level verification.
- The operating system on which the software implementation executes must provide guarantees of integrity and performance.
- The probability of failure due to physical faults of critical components and in the infrastructure systems have been shown to meet reliability requirements.
- The adequacy of the fault-tolerance strategies have been established using fault-trees and Markovian analysis as well as laboratory experimentation.
- Operational procedures have been shown to be complete and safe and have been extensively simulated.
- Assumptions of the formal analysis have been subjected to extensive investigation through simulation and flight experimentation.
- The pilot and controller workloads have been shown to be reasonable via simulated and flight experiments.
- Environmental testing requirements, such as DO-160, have been performed.

We believe that the existing incremental approach to system safety is not sufficient to convince regulatory agencies, such as the Federal Aviation Administration (FAA), that these systems are certifiably safe. We believe that safety cases built on the foundation of provably correct algorithms and designs is the only viable approach for future ATM systems.

As a first step toward a safety case of an advanced ATM concept, this paper presents the mechanical verification of an algorithm for conflict resolution and recovery, called RR3D [1]. The original presentation of this algorithm contained a hand-written proof of its correctness. Although the documented algorithm is correct, the mechanical verification revealed missing assumptions and a few errors in the hand-written proof. This supports our belief that mechanical verification is valuable even when the system has been diligently analyzed using pencil-and-paper.

Without a mechanical proof it is almost impossible to find these kinds of errors. A missing assumption, for example, could result in a fatal error in a real implementation. Since the algorithm has been formally verified, one may be confident that it is logically correct. Nevertheless, this algorithm must be translated into a machine-executable language such as Ada or C. This will necessitate several more steps of logical design, each potentially vulnerable to errors being introduced. There are many issues that must be addressed as this is done:

1. The algorithm operates within the domain of real numbers; an implementation operates within the domain of floating point numbers. Therefore, the executable code must address overflow, underflow, and all of the usual numerical problems.

2. The algorithm assumes no errors are present in the input data. But even the best sensors provide only approximate values. Communication systems, such as ADS-B, introduce errors by way of interference, latency, drop-outs, etc. The effect of these errors must be handled in a trustworthy manner. Also the system must be able to handle some number of computer or device failure conditions, i.e., it must be fault-tolerant. Mechanisms to handle these errors inevitably are implemented with software, which must also be rigorously verified.

3. The algorithm operates in a real-time environment, so one must establish that the system on which the algorithm executes, has a sufficient CPU time budget (under all possible scenarios) to complete the algorithm.

This process of design refinement can be understood as a sequence of more and more complete formal models; from the last model, an implementation can be synthesized. Each of these formal models can be shown to satisfy all properties of its predecessor model. This process is usually referred to as *design proof* and the final verification of the implementation code is called *code verification.* If the last step is accomplished using synthesis, then the auto-code tool must be verified or its output verified against the detailed design. This paper accomplishes the first step, namely, the proof that the mathematical algorithm meets its specified properties. Future work will also address the system level issues. If all of the refinement proofs are accomplished in addition to the algorithm proofs, then we can be assured that an implementation *that complies with the formal assumptions* (and this has to be checked with testing and simulation) will be free of software design errors.

3 Conflict Detection and Resolution

Conflict detection and conflict resolution algorithms are designed to warn about potential loss of air traffic separation and to produce avoidance maneuvers to be flown by the aircraft. There is a wide variety of approaches to CD&R because there are different ways to (1) predict the future trajectories, (2) define what constitutes close proximity of trajectories, (3) calculate the resolution trajectories, and (4) gain assurance about the safety and effectiveness of the algorithms. Algorithms also differ in the domain of application: (1) how far ahead in time should a conflict be detected, (2) whether the algorithm deals with only one conflict at a time or handles multiple simultaneous conflicts, and (3) the amount of coordination and communication needed to implement the algorithm.

In [11], Kuchar and Yang propose a taxonomy of CD&R algorithms. For completeness, we give an overview of that taxonomy.

3.1 Kuchar/Yang Taxonomy of CD&R Algorithms

The Kuchar/Yang taxonomy classifies CD&R algorithms based upon the following criteria: (1) state propagation method, (2) dimensions of the state information,

(3) detection alert issued, (4) resolution method, (5) dimensionality of resolution maneuver, (6) method for handling multiple alerts, and (7) other elements.

The state propagation method criteria classifies each algorithm as *nominal*, *worst-case*, or *probabilistic*. If the future course of aircraft is represented as the projected trajectory based on the current state, the algorithm is said to be *nominal*. If all possible future trajectories, subject to only physical constraints (e.g. maximum turn rate) are considered, then the algorithm is said to be *worst-case*. If possible future trajectories are assigned probabilities from which a conflict probability is calculated, the algorithm is said to be *probabilistic*.

The state dimensions criteria classifies an algorithm on the basis of the dimensions analyzed: horizontal plane only (H), vertical plane only (V), or both (HV). The detection alert criteria is just a boolean flag (T/F) which is true if the algorithm provides an explicit alert. The resolution criteria classifies an algorithm as Prescribed (P), Optimized (O), Force field (F), Manual (M), or None (-). Prescribed algorithms provide simple resolutions such as "pull up" that require no on-board calculation. Optimization approaches provide explicit calculated trajectories that remove the conflict. Force field approaches treat each aircraft as a charged particle and use modified electrostatic models from which resolution trajectories are calculated. This means that the closer two aircraft are to each other the more dramatic the maneuvers to escape from each other. Manual algorithms allow the pilot to present a trial solution and provide feedback indicating whether the proposed solution avoids conflict. If the algorithm does not provide a resolution, then it is designated as "None".

The resolution dimensionality criteria classifies an algorithm using four letters: T for Turns, V for Vertical maneuvers, S for Speed changes, and C for combined. This criteria is best explained by example. The notation TV indicates that resolutions produced by the algorithm involve turns or vertical maneuvers but not both at the same time. The notation C(TV) indicates that a simultaneous climbing or descending turn may be produced. The multiple conflicts criteria can be Pairwise (P) for algorithms where multiple conflicts are handled sequentially in pairs or Global (G) where all of the conflicts are handled at the same time.

In this taxonomy, "other elements" include how much information is known about the current state of the aircraft, how uncertainty of input data is handled, and the degree to which coordination between aircraft is required.

3.2 Classification of RR3D

It is straightforward to classify RR3D according to the Kuchar and Yang taxonomy. RR3D is a nominal, 3-dimensional algorithm (HV) which produces an alert if a conflict is detected, but does not provide the detection capability itself. It is designed to be used in conjunction with other detection algorithms. Therefore the RR3D algorithm should be classified as not providing conflict detection, i.e. (F).

The RR3D algorithm produces optimal solutions, i.e., minimal change, that are guaranteed to maintain separation and thus is an (O) algorithm. The resolution trajectories produced by RR3D only affect one parameter at a time and hence it is a STV algorithm. This is a deliberate design decision. The rationale is that a pilot will have reduced workload executing a maneuver if only one dimension changes. RR3D also produces recovery trajectories that return the aircraft to its next way-point using a second maneuver. The recovery trajectories may involve the change of ground speed along with a heading change or an altitude change. Currently, RR3D is a pairwise algorithm (P) though work is under way to establish properties of some of its solution trajectories in the context of multiple aircraft. Formal proofs are under development that the RR3D algorithm is complementary in a systems context without any explicit information being passed between aircraft. In other words, the evasive maneuvers provided by RR3D, which are executed independently on different aircraft, are guaranteed to resolve all conflicts.

With regard to the "other elements," the only information that RR3D requires is the position and velocity of the own-ship aircraft and any surrounding aircraft. The algorithm does not require any other data-exchange or handshakes between the aircraft, nor does it use information about the intent of the aircraft. RR3D currently does not take input data error into consideration. We envision future versions that incorporate support for bounded data errors.

In summary, RR3D is a nominal, HV, F, O, STV, P algorithm according to the Kuchar and Yang taxonomy.

3.3 Geometric CD&R

In recent years, new approaches for CD&R have been proposed that use non-standard programming techniques such as genetic algorithms [12–14], neural networks [15], game theory [16], graph theory [17], and semi-definite programming [18]. Given the computational complexity of some of these techniques, they usually require costly time and space discretizations. In contrast to these approaches, the geometric approach [5,6,19,20] is based on standard and well-understood analytical techniques.

In Kuchar & Yang's taxonomy, the geometric modeling correspond to nominal trajectories with either optimized or force field resolutions. Nominal trajectories are linear projections of the current position and velocity vectors. The conflict resolution problem is then expressed as a set of polynomial equations that are solved using analytical techniques. Since linear projections produce prediction errors that are negligible for short look-ahead times, this approach is also referred to as *tactical.* For large look-ahead times a more strategic approach, that uses the other pilot's intent (e.g., flight plan), is in order. While tactical approaches have well-understood geometric descriptions that allow for efficient and clear algorithms, they may fall short of pilots' expectations [3, 21].

3.4 Resolution and Recovery

Resolution and recovery algorithms—also called resolution with arrival time constraints in [22]—generate, in addition to the avoidance maneuver, merging trajectories that bring an aircraft back to its nominal path on schedule.

Figure 1 illustrates the position of conflict resolution and recovery in an abstract distributed ATM environment. On-board sensors capture the current state of the aircraft and broadcast this information to all nearby aircraft. When the conflict detection module [23] detects a conflict within a look-ahead time, the resolution and recovery module computes a list of escape and recovery maneuvers. The list of maneuvers is displayed through the cockpit interface for pilot selection or it may be forwarded to a navigation system that automatically selects one of the maneuvers.

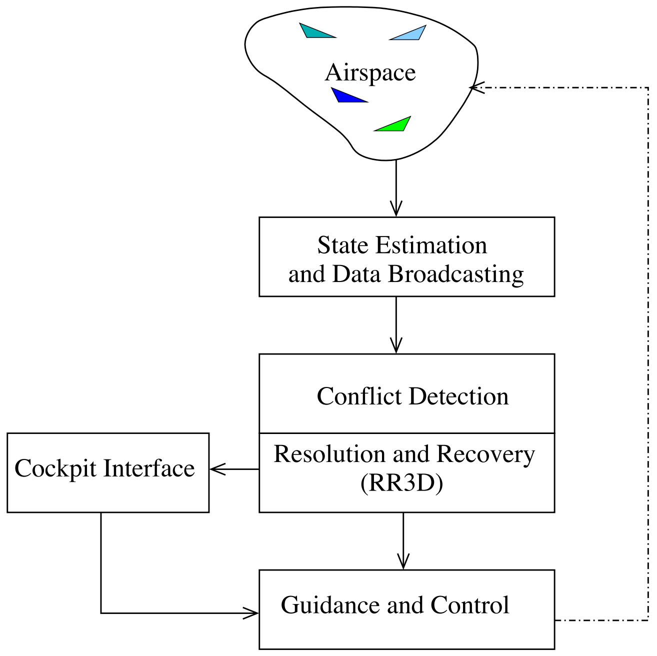

Airspace
State Estimation and Data Broadcasting
Conflict Detection
Resolution and Recovery (RR3D)
Cockpit Interface
Guidance and Control

Figure 1. On-board Processing of an ATM System

4 RR3D Algorithm

In RR3D aircraft are represented by a kinematic particle model with the center of gravity as the coordinate point of the particle. Trajectories are assumed to be composed of linear segments: speed is constant within a segment and from one segment to another acceleration is instantaneous.

RR3D resolves conflicts between a pair of aircraft: the *ownship* aircraft executing the algorithm onboard and another aircraft, also called the *intruder*. The intruder is surrounded by a cylindrical protected zone P of diameter $2D$ and height $2H$, where D is the required horizontal separation and H is the required vertical separation. A *conflict* is an intrusion of the ownship in the intruder's protected zone. RR3D computes conflict-free, easily performed escape and recovery maneuvers that result in trajectories that are tangential to the intruder's protected zone. The path will remain conflict-free, assuming the ownship aircraft follows the recommended path

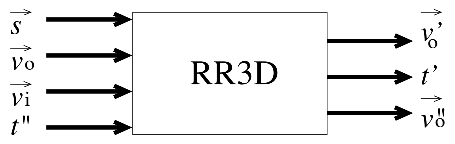


Figure 2. RR3D: Input/Outputs

and the intruder does not change its path. If the intruder maneuvers, then new paths may need to be computed.

For simplicity, we choose a relative Cartesian coordinate system where the intruder aircraft is fixed at the origin.[5] RR3D has the following inputs (see Figure 2 and Figure 3):

- the relative position s of ownship with respect to intruder.
- the velocity vector of ownship v_o.
- the velocity vector of intruder aircraft v_i.
- the arrival time t'' at the target point.

The target point s'' is defined as

$$s'' = s + t''(v_o - v_i).$$

RR3D outputs a choice of escape and recovery maneuvers for the ownship, i.e., triples (v_o', t', v_o'') where v_o' is the escape velocity vector, t' is the time of turn, and v_o'' is the recovery velocity vector. Figure 2 illustrates RR3D's functionality for a single output.

In order to reduce the pilot's workload, the escape and recovery maneuvers are constrained in such a way that both v_o' and v_o'' satisfy one of the following conditions:

1. *Change of vertical speed only.* The ownship's vertical speed may change but not its heading or ground speed, i.e., $v_{ox}' = v_{ox} = v_{ox}''$ and $v_{oy}' = v_{oy} = v_{oy}''$.
2. *Change of ground speed only.* The ownship's ground speed may change but not its heading or vertical speed. Formally, there is a $k > 0$ such that $v_{ox}' = kv_{ox}$, $v_{oy}' = kv_{oy}$, and $v_{oz}' = v_{oz}$, and there is a $j > 0$ such that $v_{ox}'' = jv_{ox}$, $v_{oy}'' = jv_{oy}$, and $v_{oz}'' = v_{oz}$.
3. *Change of heading.* In the two dimensional projection, the escape course and the recovery course (each in absolute coordinates) form a triangle. By the triangle inequality, the escape course and the recovery course together are longer than the original course. To arrive at the target point at time t'', the ownship has to compensate by using a greater average ground speed as opposed to its original ground speed. Hence, maneuvers where only heading changes

[5]We are assuming perfect knowledge of the location and velocity of the intruder.

are allowed cannot reach the target point in time. In this case, we propose a change of heading combined with a change of ground speed at time t'. In the escape course, the ownship's heading may change, but not its ground speed or vertical speed; for the recovery course one must allow for a change of ground speed as well as well as the heading change. Formally, $v'^2_{ox} + v'^2_{oy} = v^2_{ox} + v^2_{oy}$, $v'_{oz} = v_{oz}$, and $v''_{oz} = v_{oz}$.

Furthermore, we require that the escape and recovery courses are tangential to the lateral surface of the protected zone. Tangential courses solve a conflict in an optimal way. They require the least effort to correct the original trajectory such that the ownship arrives at the next way point[6] at the scheduled time while maintaining separation. Original, escape, and recovery courses are illustrated in Figure 3.

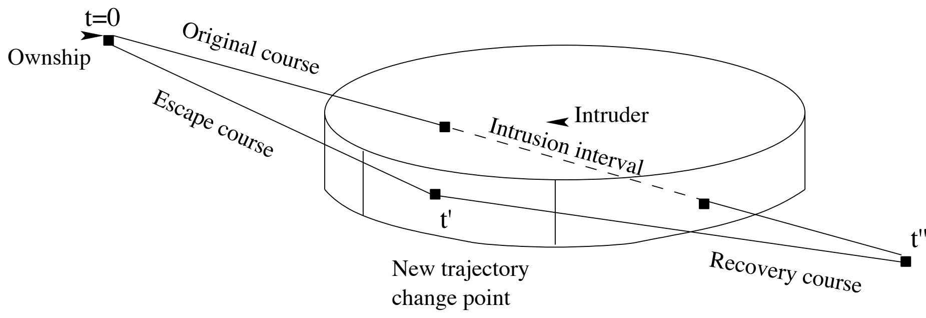


Figure 3. Relative movement of the ownship w.r.t. the intruder

The RR3D algorithm is presented as a set of solutions to polynomial equations that represent the initial assumptions, the correctness conditions, one of three constraints listed above, and the tangential requirement. The solutions are categorized according to the part of the surface of the protected zone P that is touched during the escape and recovery courses. The following cases are identified: line/line (Figure 4), line/circle (Figure 5), circle/line (Figure 6), one-circle (Figure 7), circle/circle (Figure 8), in-circle (Figure 9), and out-circle (Figure 10).

The RR3D algorithm is required to satisfy the following properties:

- **Correctness of the Escape Course:** The ownship maintains separation during the escape course. Let $v' = v'_o - v_i$; then for all times $0 \leq t \leq t'$

$$s + tv' \notin P \tag{1}$$

- **Correctness of the Recovery Course:** The ownship maintains separation during the recovery course. Let $v'' = v''_o - v_i$; then for all times $t' \leq t \leq t''$

$$s + t'v' + (t - t')v'' \notin P \tag{2}$$

[6]RR3D does not consider way points beyond the next one. RR3D could be used in conjunction with a strategic planner that alters subsequent way points to meet higher-level objectives such as flow management or weather avoidance.

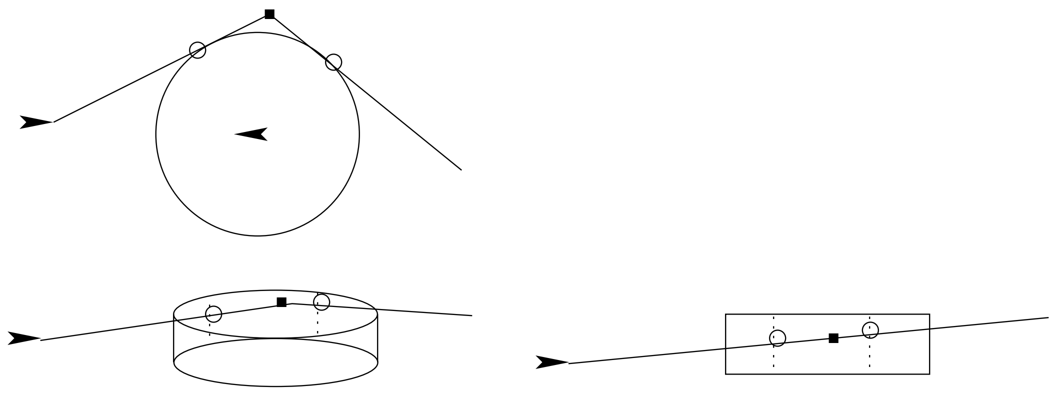


Figure 4. Line/line (top view, perspective view, and side view)

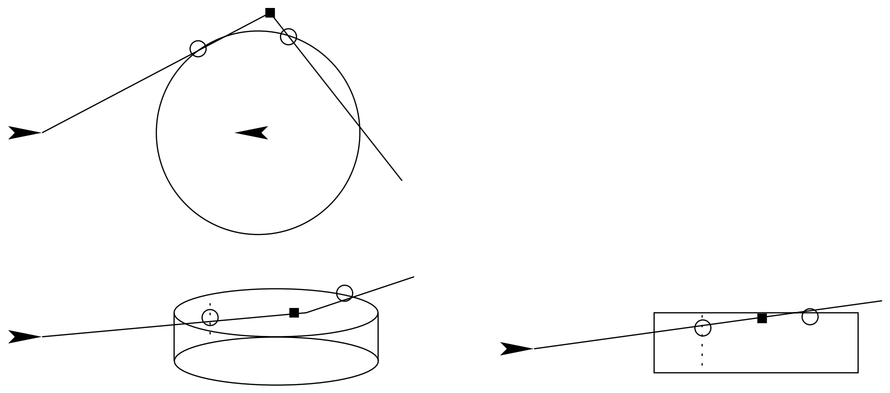


Figure 5. Line/circle (top view, perspective view, and side view)

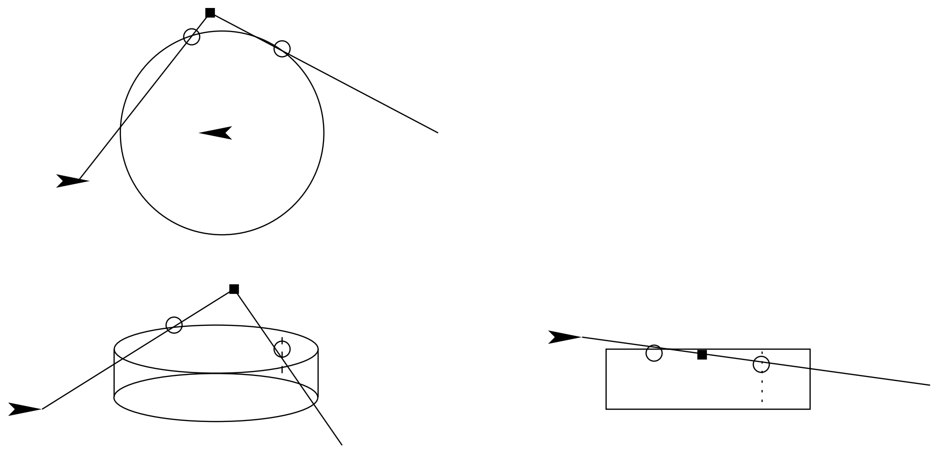
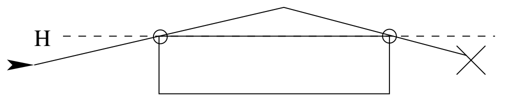


Figure 6. Circle/line (top view, perspective view, and side view)



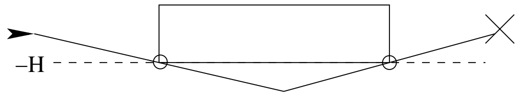


Figure 7. One-circle cases (side views)

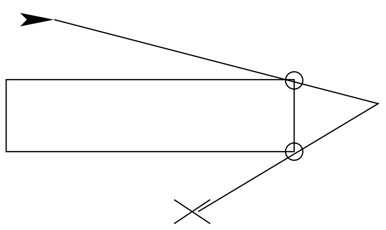
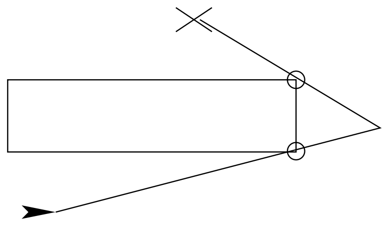
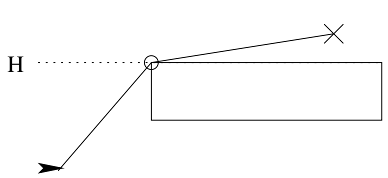



Figure 8. Circle-circle cases (side views)



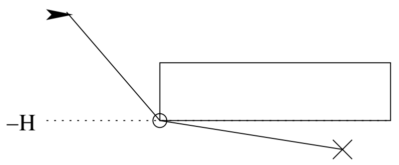
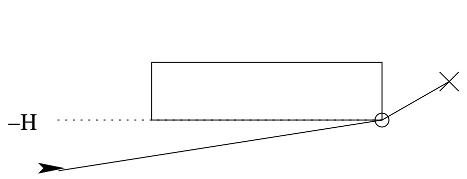


Figure 9. In-circle cases (side views)



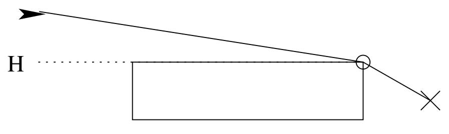


Figure 10. Out-circle cases (side views)

- **Timeliness:** The ownship arrives at the target point at the prescribed time.

$$s + t'v' + (t'' - t')v'' = s'' \tag{3}$$

Geser et al. [1] present a proof that the RR3D algorithm is *correct*, i.e., satisfies the required properties (1), (2), and (3).

We describe in Section 5 how the paper-and-pencil proofs of [1] are mechanized in PVS.

5 Formal Verification of RR3D

This presentation of the formal verification of RR3D is organized as follows. First we define a few predicates to express the separation requirements and some geometric properties, and useful statements about them. Then we prove correctness of the escape course, correctness of the recovery course, and timeliness for each case that RR3D defines. We divide the cases according to the constraint they satisfy: Vertical, Ground-Speed, and Heading.

5.1 Basic Definitions and Common Lemmas

In this section we use s, v, t in an generic way, i.e., they do not necessarily refer to the relative variables.

5.1.1 Horizontal and Vertical Separation

The *infinite cylinder* is the set of points

$$P_\infty = \{(x, y, z) \mid x^2 + y^2 < D^2\},$$

and the *infinite slice* is the set of points

$$S_\infty = \{(x, y, z) \mid |z| < H\}.$$

Associated with these regions, we define three predicates about aircraft separation in the PVS specification.

$$\textsf{hor\_sep?}(s) = {s_x}^2 + {s_y}^2 \geq D^2 \tag{4}$$

$$\textsf{vert\_sep?}(s) = |s_z| \geq H \tag{5}$$

$$\textsf{separation?}(s, v) = \forall t : \textsf{hor\_sep?}(s + tv) \ \vee \ \textsf{vert\_sep?}(s + tv) \tag{6}$$

We also define a notion of separation over an interval of time:

$$\textsf{pred\_sep?}(s, v, t'') = \forall t : 0 < t < t'' \ \supset \ \textsf{hor\_sep?}(s + tv) \ \vee \ \textsf{vert\_sep?}(s + tv) \tag{7}$$

The following useful lemma enables one to translate the starting point:

Lemma 1 (separation_lem)

$$\mathsf{separation?}(s,v) \Leftrightarrow \mathsf{separation?}(s+tv,v) \tag{8}$$

Proof. Case 1 [$\mathsf{separation?}(s,v) \supset \mathsf{separation?}(s+tv,v)$] We need to prove that

$$\mathsf{hor\_sep?}(s+tv+Tv) \lor \mathsf{vert\_sep?}(s+tv+Tv)$$

for an arbitrary T. From the premise we have

$$\forall t'' : \mathsf{hor\_sep?}(s+t''v) \lor \mathsf{vert\_sep?}(s+t''v)$$

Substituting $t+T$ for t'' we have the desired result.
Case 2 [$\mathsf{separation?}(s+tv,v) \supset \mathsf{separation?}(s,v)$] Proof similar to Case 1.
□

5.1.2 Correctness Criteria

A point s at the boundary of the infinite cylinder and moving with velocity v may move into or out of the infinite cylinder. The direction is determined by the sign of the dot product $(s_x,s_y)\cdot(v_x,v_y)$. In formulas (9-11) we provide convenient names for each direction.

$$\mathsf{entry?}(s,v) = s_xv_x + s_yv_y \le 0 \tag{9}$$
$$\mathsf{exit?}(s,v) = s_xv_x + s_yv_y \ge 0 \tag{10}$$
$$\mathsf{tangent?}(s,v) = s_xv_x + s_yv_y = 0 \tag{11}$$

For convenience the tangent case is included in the entry? and exit? definitions. The predicates $\mathsf{entry\_point?}(s,v)$, $\mathsf{exit\_point?}(s,v)$, and $\mathsf{tangent\_point?}(s,v)$ are defined as the conjunction of $s_x^2+s_y^2=D^2$ with $\mathsf{entry?}(s,v)$, $\mathsf{exit?}(s,v)$, and $\mathsf{tangent?}(s,v)$, respectively.

We provide correctness criteria for the line and circle cases.

Theorem 2 (Line Case Correctness)

$$\mathsf{tangent\_point?}(s,v) \supset \mathsf{separation?}(s,v)$$

Proof. Let $s+tv$ be a moving point such that s is tangent to P_∞. Then, by properties of tangent lines, $(s_x+tv_x)^2+(s_y+tv_y)^2 \ge D^2$ for all times t.
□

Theorem 3 (Circle Case Correctness)

$$\begin{aligned}&\mathsf{hor\_sep?}(s) \land \mathsf{vert\_sep?}(s)\\ \land\ &(\mathsf{entry\_point?}(s,v) \land s_zv_z \ge 0 \lor \mathsf{entry\_point?}(s,v) \land s_zv_z \le 0)\\ \supset\ &\mathsf{separation?}(s,v)\end{aligned}$$

Proof. Let $s+tv$ be a moving point such that $s_x^2+s_y^2=D^2$, $|s_z|=H$, and either (1) $s_xv_x+s_yv_y \le 0$ and $s_zv_z \ge 0$ or (2) $s_xv_x+s_yv_y \ge 0$ and $s_zv_z \le 0$. Then, for all times t, either (a) horizontal separation: $(s_x+tv_x)^2+(s_y+tv_y)^2 \ge D^2$ or (b) vertical separation: $|s_z+tv_z| \ge H$.
□

5.1.3 Times of Intersection with the Cylinder Lateral Surface

In order to use the correctness criteria, we have to determine the times t at which a moving point $s + tv$ intersects the lateral surface of the infinite cylinder. These times are given as the solutions of

$$(s_x + tv_x)^2 + (s_y + tv_y)^2 \quad = \quad D^2. \tag{12}$$

The predicate hor_speed_gt_0? expresses that the horizontal speed is greater than zero:

$$\mathsf{hor\_speed\_gt\_0?}(v) \;\Leftrightarrow\; (v_x^2 + v_y^2 > 0). \tag{13}$$

If hor_speed_gt_0?(v) holds then (12) reduces to a quadratic equation in t:

$$t^2(v_x^2 + v_y^2) + 2t(s_x v_x + s_y v_y) + s_x^2 + s_y^2 - D^2 = 0. \tag{14}$$

The discriminant $\Delta(s, v)$ is defined as

$$\begin{aligned}\Delta(s, v) &= 2^2(s_x v_x + s_y v_y)^2 - 4(v_x^2 + v_y^2)(s_x^2 + s_y^2 - D^2) \\ &= 4D^2(v_x^2 + v_y^2) - 4(s_x v_y - s_y v_x)^2.\end{aligned} \tag{15}$$

If $\Delta(s, v) \leq 0$ then the moving point does not intersect P_∞. In particular, if $\Delta(s, v) = 0$ we have the tangent case. We define a predicate tangent_condition? by

$$\mathsf{tangent\_condition?}(s, v) \;\Leftrightarrow\; (D^2(v_x^2 + v_y^2) = (s_x v_y - s_y v_x)^2). \tag{16}$$

If tangent_condition?(s, v) holds then the time $\tau(s, v)$ of closest approach in the horizontal plane is the unique solution of (14):

$$\tau(s, v) = -\frac{s_x v_x + s_y v_y}{v_x^2 + v_y^2}. \tag{17}$$

The following lemma establishes the fundamental property of τ: if the ownship is on a course satisfying the tangent condition, then it is at the tangent point at time τ.

Lemma 4 (tau_is_tangent_pt)

$$\begin{array}{ll} & \mathsf{hor\_speed\_gt\_0?}(v') \\ \wedge & \mathsf{tangent\_condition?}(s, v') \\ \supset & \mathsf{tangent\_point?}(s + v'\tau(s, v'), v') \end{array}$$

Proof. Expanding the tangent_point? predicate yields two claims:

$$(s_x + \tau(s, v')v'_x)^2 + (s_y + \tau(s, v')v'_y)^2 = D^2, \tag{18}$$

$$(s_x + \tau(s, v')v'_x)v'_x + (s_y + \tau(s, v')v'_y)v'_y = 0. \tag{19}$$

Proof of (18). Since ${v'}_x^2 + {v'}_y^2 \neq 0$ by $\mathsf{hor\_speed\_gt\_0?}(v')$, the tangent condition (16) can be expressed as

$$D^2 = \frac{(s_x v'_y - s_y v'_x)}{{v'}_x^2 + {v'}_y^2}.$$

Substituting this into equation (18) and expanding the definition of $\tau(s, v')$ yields

$$\left(s_x + \left[-\frac{s_x v'_x + s_y v'_y}{{v'_x}^2 + {v'_y}^2}\right] v'_x\right)^2 + \left(s_y + \left[-\frac{s_x v'_x + s_y v'_y}{{v'_x}^2 + {v'_y}^2}\right] v'_y\right)^2 = \frac{(s_x v'_y - s_y v'_x)}{{v'}_x^2 + {v'}_y^2}.$$

Algebraic simplification verifies this equality. This concludes the proof of (18).

Expanding the definition of $\tau(s, v')$ in equation (19) yields

$$\left(s_x + \left[-\frac{s_x v'_x + s_y v'_y}{{v'_x}^2 + {v'_y}^2}\right] v'_x\right) v'_x + \left(s_y + \left[-\frac{s_x v'_x + s_y v'_y}{{v'_x}^2 + {v'_y}^2}\right] v'_y\right) v'_y = 0.$$

Algebraic simplification verifies this equality.
□

5.1.4 Entering and leaving P_∞

If $\Delta(s, v) > 0$, we get two solutions for (14) which we call $\Theta^-(s, v)$ and $\Theta^+(s, v)$, respectively:

$$\Theta^-(s, v) = \frac{-2s_x v_x - 2s_y v_y - \sqrt{\Delta(s, v)}}{2v_x^2 + 2v_y^2}, \tag{20}$$

$$\Theta^+(s, v) = \frac{-2s_x v_x - 2s_y v_y + \sqrt{\Delta(s, v)}}{2v_x^2 + 2v_y^2}. \tag{21}$$

By definition, $\Theta^-(s, v) < \Theta^+(s, v)$.

To facilitate this definition in PVS, a predicate clash? is defined as follows:

$$\mathsf{clash?}(s, v) \quad = \quad {v_x}^2 + {v_y}^2 > 0 \quad \wedge \quad \Delta(s, v) > 0 \tag{22}$$

Thus we have

$$\Theta^{\pm}(s : \mathsf{vector}, v : (\mathsf{clash?})) = \frac{-2s_x v_x - 2s_y v_y \pm \sqrt{\Delta(s, v)}}{2{v_x}^2 + 2{v_y}^2} \tag{23}$$

Before we continue, we need to digress to the solution of quadratic equations. The following formula characterizes the solutions of a quadratic equation:

$$ax^2 + bx + c = 0 \;\Leftrightarrow\; \mathsf{discr}(a, b, c) \geq 0 \;\wedge\; (x = \mathsf{root}(-1, a, b, c) \;\vee\; x = \mathsf{root}(1, a, b, c)). \tag{24}$$

The discriminant $\mathsf{discr}(a,b,c)$ and the solutions $\mathsf{root}(\varepsilon,a,b,c)$ for $\varepsilon = \pm 1$ are defined by

$$\mathsf{discr}(a,b,c) = b^2 - 4ac, \tag{25}$$

$$\mathsf{root}(\varepsilon,a,b,c) = \frac{-b + \varepsilon\sqrt{\mathsf{discr}(a,b,c)}}{2a}. \tag{26}$$

The following lemma establishes the key property about $\Theta^{\pm}$:

Lemma 5 (THETA_main)

$$\mathsf{clash?}(s,v) \ \wedge \ t = \Theta^{\pm}(s,v) \ \supset \ (s_x + tv_x)^2 + (s_y + tv_y)^2 = D^2$$

Proof. Application of (24) to (14).
□

The following two lemmas establish that Θ^- is an entry point and that Θ^+ is an exit point:

Lemma 6 (entry_it_is)

$$\begin{aligned} &\mathsf{hor\_sep?}(s) \ \wedge \ \mathsf{clash?}(s,v) \ \wedge \ \neg\mathsf{pred\_sep?}(s,v,t'') \\ &\qquad\qquad \supset \ \mathsf{entry\_point?}(s + v\Theta^-(s,v), v) \end{aligned}$$

Proof. To show that $s + v\Theta^-(s,v)$ is an entry point we show that

$$(s_x + \Theta^-(s,v)v_x)^2 + (s_y + \Theta^-(s,v)v_y)^2 = D^2, \tag{27}$$

$$(s_x + \Theta^-(s,v)v_x)v_x + (s_y + \Theta^-(s,v)v_y)v_y \le 0. \tag{28}$$

THETA_main [Lemma 5] discharges (27). For the claim (28) let us consider the derivative of the distance between the two aircraft: $2(s_x + tv_x)v_x + 2(s_y + tv_y)v_y$, which is equal to $2t(v_x^2 + v_y^2) + 2(s_xv_x + s_yv_y)$. We first show that this is non-positive for all $t \le \tau(s,v)$.

$$\begin{aligned} & t \le \tau(s,v) \\ \supset \quad & t \le -\frac{s_xv_x + s_yv_y}{v_x^2 + v_y^2} \\ \supset \quad & 2t(v_x^2 + v_y^2) \le -2(s_xv_x + s_yv_y) \\ \supset \quad & 2t(v_x^2 + v_y^2) + 2(s_xv_x + s_yv_y) \le 0 \end{aligned}$$

From (20) and (17) it follows trivially that $\Theta^-(s,v) < \tau(s,v)$. Thus we can substitute $\Theta^-(s,v)$ for t in the previous inequality to get

$$2\Theta^-(s,v)(v_x^2 + v_y^2) + 2(s_xv_x + s_yv_y) \le 0$$

which simplifies to (28).
□

Lemma 7 (exit_it_is)

$$\begin{aligned}\mathsf{hor\_sep?}(s) \ \wedge \ \mathsf{clash?}(s,v) \ \wedge \ \neg\mathsf{pred\_sep?}(s,v,t'') \\ \supset \ \mathsf{exit\_point?}(s + v\Theta^{+}(s,v), v)\end{aligned}$$

Proof. Proof is similar to proof of entry_it_is except that $\Theta^{+}(s,v)$ is used and the derivative of the distance is non-negative for $t \geq \tau(s,v)$ is shown.
□

Lemma 8 (exploit_pred_conflict)

$$t'' > 0 \ \wedge \ \mathsf{hor\_sep?}(s) \ \wedge \ \neg\mathsf{pred\_sep?}(s,v,t'') \ \supset \ \mathsf{clash?}(s,v)$$

Proof. The following chain of implications provides the proof:

$$\begin{aligned}&\neg\mathsf{pred\_sep?}(s,v,t'') \\ &\supset \ \neg(\forall t : 0 \leq t \leq t'' \ \supset \ \mathsf{hor\_sep?}(s + vt)) \\ &\supset \ (\mathsf{hor\_speed\_gt\_0?}(v) \ \wedge \ \Delta(s,v) > 0 \ \wedge \ 0 < \Theta^{+}(s,v) \ \wedge \ \Theta^{-}(s,v) < t'') \\ &\supset \ \mathsf{clash?}(s,v)\end{aligned}$$

The second implication above follows from a characterization, similar to (24), of the solutions of the quadratic inequality $at^2 + bt + c \geq 0$ where $a = v_x^2 + v_y^2$ and $b = 2(s_x v_x + s_y v_y)$ and $c = s_x^2 + s_y^2 - D^2$ derived from (14).
□

Lemma 9 (vert_pred)

$$\begin{aligned}& s'' = s + t''v \\ \wedge \quad & ((s_z \geq H \ \wedge \ s''_z \geq H) \ \vee \ (s_z \leq -H \ \wedge \ s''_z \leq -H)) \\ \supset \quad & \mathsf{pred\_sep?}(s,v,t'')\end{aligned}$$

Proof. In order to show pred_sep?(s,v,t'') it is sufficient to show $|s_z + t''v_z| \geq H$.
Case 1 [$s_z \geq H \ \wedge \ s''_z \geq H$]: From the first premise and the case conditions we get $s''_z - t''v_z \geq H$ and $s_z + t''v_z \geq H$. Now if $v_z \geq 0$ we have $s_z + t''v_z = |s_z + t''v_z|$ and hence vertical separation. Otherwise, since s''_z is positive, $|s''_z| = s''_z = s_z + t''v_z$ and hence $|s_z + t''v_z| \geq H$.
Case 2 [$s_z \leq -H \ \wedge \ s''_z \leq -H$] Same approach as Case 1 only substituting $-s_z$ for s_z.
□

We will need (16) and (17) instantiated with the parameters of the escape and the recovery courses. For the escape course we get tangent_condition?(s,v') and the time of closest approach in the horizontal plane $\tau(s,v')$. The moving point $s'' + (t - t'')v''$ describes the recovery course in a translated time $t - t''$. Therefore, for the recovery course we get tangent_condition?(s'',v'') and the time of closest approach in the horizontal plane $\tau(s'',v'') + t''$.

5.1.5 Reaching altitude H or $-H$

If $v_z \neq 0$ then the times when the ownship reaches altitude H or $-H$ are the solutions of $|s_z + tv_z| = H$ for t, which we call $\theta^-(s_z, v_z)$ and $\theta^-(s_z, v_z)$, respectively:

$$\theta^-(s_z, v_z) = \frac{-\mathsf{sign}(v_z)H - s_z}{v_z} \tag{29}$$

$$\theta^+(s_z, v_z) = \frac{\mathsf{sign}(v_z)H - s_z}{v_z} \tag{30}$$

The following lemma establishes the main property of $\theta^\pm$: the ownship is at the top or bottom of the infinite slice.

Lemma 10 (reaching_H_theta)

$$v_z \neq 0 \;\supset\; |s_z + \theta^\pm(s_z, v_z)v_z| = H$$

Proof. The condition $v_z \neq 0$ is only required to ensure that $\theta^\pm$ is defined. If $v_z > 0$ then by (29) or (30) we get

$$s_z + \theta^\pm(s_z, v_z)v_z = s_z + \pm H - s_z = \pm H,$$

the absolute value of which is H. If $v_z < 0$ then

$$s_z + \theta^\pm(s_z, v_z)v_z = s_z - \pm H - s_z = - \pm H$$

the absolute value of which is H.
□

The next lemma establishes another important property of the $\theta^\pm$ function: at time $\theta^+(s, v)$ the ownship is *leaving* the infinite slice and at time $\theta^-(s, v)$ it is *entering* the infinite slice.

Lemma 11 (vertical_entry_exit_condition)

$$v_z \neq 0 \;\supset\; (s_z + \theta^+(s_z, v_z)v_z)v_z \geq 0 \;\wedge\; (s_z + \theta^-(s_z, v_z)v_z)v_z \leq 0$$

Proof. The condition $v_z \neq 0$ is required to ensure that the function $\theta^\pm$ is defined. We use the fact that $H > 0$.

Case 1 [$v_z > 0$]. By (30) for $v_z > 0$,

$$s_z + \theta^+(s_z, v_z)v_z = s_z + H - s_z = H.$$

By replacement, the first claim reduces to $Hv_z \geq 0$, which trivially holds. Likewise, by (29),

$$s_z + \theta^-(s_z, v_z)v_z = s_z - H - s_z = -H.$$

By replacement, the second claim reduces to $-Hv_z \leq 0$, which trivially holds.

Case 2 [$v_z < 0$]. By (30) for $v_z < 0$,

$$s_z + \theta^+(s_z, v_z)v_z = s_z - H - s_z = -H.$$

By replacement, the first claim reduces to $-Hv_z \geq 0$, which trivially holds. Likewise, by (29),

$$s_z + \theta^-(s_z, v_z)v_z = s_z + H - s_z = H.$$

By replacement, the second claim reduces to $Hv_z \leq 0$, which trivially holds.
□

The next lemma states that $\theta^\pm$ values can be translated in time.

Lemma 12 (theta_translation)

$$v_z \neq 0 \ \supset \ \theta^\pm(s_z + t''v_z, v_z) = \theta^\pm(s_z, v_z) - t''$$

Proof. The condition $v_z \neq 0$ is required to ensure that the terms $\theta^\pm(s_z + t''v_z, v_z)$ and $\theta^\pm(s_z, v_z)$ are defined. Replacement by (29) and (30) yields

$$\frac{\pm\mathsf{sign}(v_z)H - (s_z + t''v_z)}{v_z} = \frac{\pm\mathsf{sign}(v_z)H - s_z}{v_z} - t'',$$

which resolves by algebraic simplification.
□

5.1.6 Time of Switch

The time t' is the time at which the ownship switches from the escape course to the recovery course. This time satisfies

$$t'(v' - v'') = t''(v - v''),$$

or in coordinate notation

$$t'(v'_x - v''_x) = t''(v_x - v''_x), \tag{31}$$
$$t'(v'_y - v''_y) = t''(v_y - v''_y), \tag{32}$$
$$t'(v'_z - v''_z) = t''(v_z - v''_z). \tag{33}$$

Equations (31) and (32) allow us to express v''_y and v''_x in terms of $t', t'', v_x, v'_x, v_y, v''_y$ which allows us to compute the velocity vector from the arrival time.

$$v''_x = \frac{t''v_x - t'v'_x}{t'' - t'}, \tag{34}$$
$$v''_y = \frac{t''v_y - t'v'_y}{t'' - t'}. \tag{35}$$

5.2 Correctness of Vertical Speed Case

We impose the constraint that only the vertical component of the velocity vector may change. Formally, we define a predicate vertical_change? as follows

$$\mathsf{vertical\_change?}(v, w) \ \Leftrightarrow \ (v_x = w_x) \ \wedge \ (v_y = w_y) \tag{36}$$

Constraining both v' and v'', we have:

$$\mathsf{vertical\_change?}(v, v') \ \wedge \ \mathsf{vertical\_change?}(v', v'')$$

In terms of absolute coordinates, we have:

$$v'_{ox} = v_{ox} = v''_{ox} \qquad \text{and} \qquad v'_{oy} = v_{oy} = v''_{oy}. \tag{37}$$

If the relative ground speed is zero ($v_x^2 + v_y^2 = 0$) then either the ownship is inside the infinite cylinder ($s_x^2 + s_y^2 < D^2$), and there is no vertical solution, or else there is no conflict. Otherwise, $\Theta^-(s, v)$ and $\Theta^+(s, v)$ are defined as in equations (20) and (21), and we may have the following independent solutions.

5.2.1 In-circle

If $0 < \Theta^-(s, v) < t''$ and $|s''_z| \geq H$ then there is an in-circle solution (Figure 9). It is given by $t' = \Theta^-(s, v)$,

$$\begin{aligned} v''_{oz} &= v_{iz} + \frac{-\mathsf{sign}(s''_z)H - s''_z}{\Theta^-(s, v) - t''}, \quad \text{and} \\ v'_{oz} &= v_{iz} + \frac{t''(v_{oz} - v_{iz}) - (t'' - \Theta^-(s, v))(v''_{oz} - v_{iz})}{\Theta^-(s, v)} \\ &= \frac{t''(v_{oz} - v''_{oz})}{\Theta^-(s, v)} + v''_{oz}. \end{aligned}$$

The following theorem has been formally verified for this maneuver:

Theorem 13 (vert_in_circle_correctness)

$$\begin{array}{ll} & \mathsf{hor\_sep?}(s) \\ \wedge & \neg\mathsf{pred\_sep?}(s, v, t'') \\ \wedge & \mathsf{vertical\_change?}(v + v_i, v' + v_i) \\ \wedge & \mathsf{vertical\_change?}(v' + v_i, v'' + v_i) \\ \wedge & t' > 0 \ \wedge \ t' < t'' \\ \wedge & t' = \Theta^-(s, v) \\ \wedge & s'' = s + t''v \\ \wedge & |s''_z| \geq H \\ \wedge & v''_z = \dfrac{\mathsf{sign}(s''_z)H - s''_z}{t' - t''} \\ \wedge & v'_z = \dfrac{t''v_z - (t'' - t')v''_z}{t'} \\ \supset & \mathsf{separation?}(s, v') \ \wedge \ \mathsf{separation?}(s + t'v', v'') \end{array}$$

Proof. First we use exploit_pred_conflict [Lemma 8] to obtain clash?(s, v). Next we observe that

$$s_z + t' v'_z = \mathsf{sign}(s''_z) H \tag{38}$$

by cross multiplying the formulas for v''_z and v'_z in the premise and using some algebra.

Part 1 [Establish separation?(s, v')] Using separation_lem [Lemma 1] we change the goal to establishing separation at $s + t'v'$, i.e., to separation?$(s + t'v', v')$. Application of Circle Case Correctness [Theorem 3] at $s + t'v'$ will give us the desired result, provided that we discharge its premises. We do so by proving that

$$|s_z + t'v'_z| = H, \tag{39}$$

$$\mathsf{entry\_point?}(s + t'v', v'), \tag{40}$$

$$(s_z + t'v'_z)v'_z \geq 0. \tag{41}$$

The claim (39) follows from (38).

To show (40), we establish entry_point?$(s+\Theta^-(s, v)v, v)$ by entry_it_is [Lemma 6]. Since entry_point? only involves the x and y components of the vector, and we have vertical_change? (v, v'), we also get entry_point?$(s + \Theta^-(s, v)v', v')$. The claim (40) follows by $t' = \Theta^-(s, v)$.

This leaves us to establish (41). Replacing with (38). This reduces to $\mathsf{sign}(s''_z)Hv'_z \geq 0$. To prove this goal we perform a case split on $s''_z \geq 0$.

Case [$s''_z \geq 0$]: Expanding sign and using the fact that H is positive, the goal becomes $v'_z \geq 0$. Using the formula for v'_z in the premise, and using $t' > 0$, the goal becomes

$$t'v''_z - t''v''_z + t''v_z \geq 0$$

From the formula for v''_z in the premise, we obtain $t'v''_z - t''v''_z = H - s''_z$, which can be used to simplify the goal to

$$H - s''_z + t''v_z \geq 0$$

Using $s'' = s + t''v$, we get:

$$H - s_z \geq 0$$

From the premise $|s''_z| \geq H$, we get $s''_z \geq H$. From vert_pred [Lemma 9], we get $(s_z \geq H \ \wedge \ s''_z \geq H) \ \vee \ (s_z \leq -H \ \wedge \ s''_z \leq -H)$ which suffices to finish off this case.

Case [$s''_z < 0$]: Analogous.

Part 2 [Establish separation?$(s+t'v', v'')$] Since $s'' = s+t''v$, the goal can be rewritten as:

$$\mathsf{separation?}(s'' - v''(t'' - t'), v'')$$

An application of $\mathsf{Circle\ Case\ Correctness}$ [Theorem 3] at $s'' - v''(t'' - t')$ will give us the desired result, provided that we can discharge its premises. We do so by proving that

$$|s''_z - (t'' - t')v''_z| = H, \tag{42}$$

$$\mathsf{entry\_point?}(s'' - (t'' - t')v'', v''), \tag{43}$$

$$(s''_z - (t'' - t')v''_z)v''_z \geq 0. \tag{44}$$

Substitution of the definition of v''_z in (42) and algebraic simplification yields $|\mathsf{sign}(s''_z)H| = H$ which is trivially true.

For (43), we first show $\mathsf{entry\_point?}(s + \Theta^-(s, v)v, v)$ using $\mathsf{entry\_it\_is}$ [Lemma 6]. Then the claim (43) follows by $\mathsf{vertical\_change?}(v, v')$, $\mathsf{vertical\_change?}(v', v'')$, $t' = \Theta^-(s, v)$, and algebra.

Finally let us prove (44). We first cross-multiply the premise that defines v''_z to get:

$$(t' - t'')v''_z = \mathsf{sign}(s''_z)H - s''_z. \tag{45}$$

Substituting in (44) and simplifying yields

$$\mathsf{sign}(s''_z)v''_z H \geq 0 \tag{46}$$

Case splitting on the argument to sign:

Case 1 [$s''_z \geq 0$]: From the premise $|s''_z| \geq H$ we get $s''_z \geq H$. Expanding sign in (45) we have $(t' - t'')v''_z = H - s''_z$. Thus $(t' - t'')v''_z \leq 0$; hence $v''_z \geq 0$. The claim (46) follows.

Case 2 [$s''_z < 0$]: From the premise $|s''_z| \geq H$, we get $s_z + t''v_z \leq -H$. Expanding sign in (45) we have $(t' - t'')v''_z = -H - s_z$. Thus $(t' - t'')v''_z \geq 0$; hence $v''_z \leq 0$. The claim (46) follows.
□

5.2.2 Out-circle

If $0 < \Theta^+(s, v) < t''$ and $|s_z| \geq H$ then there is an out-circle solution (Figure 10). It is given by $t' = \Theta^+(s, v)$,

$$v'_{oz} = v_{iz} + \frac{-\mathsf{sign}(v_z)H - s_z}{\Theta^+(s, v)}, \quad \text{and}$$

$$\begin{aligned} v''_{oz} &= v_{iz} + \frac{t''(v_{oz} - v_{iz}) - \Theta^+(s, v)(v'_{oz} - v_{iz})}{t'' - \Theta^+(s, v)} \\ &= \frac{t''v_{oz} - \Theta^+(s, v)v'_{oz}}{t'' - \Theta^+(s, v)}. \end{aligned}$$

The verification of this solution was facilitated by the proof of the following lemmas about the signs of the vectors:

Lemma 14 (signs_are_opposite)

$$\begin{aligned}&\neg\mathsf{pred\_sep?}(s,v,t'') \ \wedge\ |s_z| \geq H\\ &\qquad\supset\ \mathsf{sign}(s_z) = -\mathsf{sign}(v_z)\end{aligned}$$

Lemma 15 (signs_ve_z)

$$\begin{aligned}&\neg\mathsf{pred\_sep?}(s,v,t'') \ \wedge\ |s_z| \geq H \ \wedge\ C > 0\\ &\wedge\ v'_z = \frac{-\mathsf{sign}(v_z)H - s_z}{C} \ \wedge\ v'_z \neq 0\\ &\quad\supset\ \mathsf{sign}(v'_z) = \mathsf{sign}(v_z)\end{aligned}$$

Lemma 16 (signs_vr_z)

$$\begin{array}{ll} & \neg\mathsf{pred\_sep?}(s,v,t'')\\ \wedge & |s_z| \geq H\\ \wedge & v'_z = \dfrac{-\mathsf{sign}(v_z)H - s_z}{C}\\ \wedge & t'' - C > 0 \ \wedge\ C > 0\\ \wedge & v''_z = \dfrac{t''v_z - v'_zC}{t'' - C}\\ & \quad\supset\ \mathsf{sign}(v''_z) = -\mathsf{sign}(s_z)\end{array}$$

Proofs of these lemmas are given in Appendix 7.2.

The following theorem has been formally verified for this maneuver:

Theorem 17 (vert_out_circle_correctness)

$$\begin{array}{ll} & \mathsf{hor\_sep?}(s)\\ \wedge & \mathsf{vertical\_change?}(v,v') \ \wedge\ \mathsf{vertical\_change?}(v',v'')\\ \wedge & 0 < \Theta^+(s,v) \ \wedge\ \Theta^+(s,v) < t''\\ \wedge & |s_z| \geq H\\ \wedge & v'_z = \dfrac{-\mathsf{sign}(v_z)H - s_z}{\Theta^+(s,v)}\\ \wedge & v''_z = \dfrac{t''v_z - \Theta^+(s,v)v'_z}{t'' - \Theta^+(s,v)}\\ \wedge & \neg\mathsf{pred\_sep?}(s,v,t'')\\ \supset & \mathsf{separation?}(s,v') \ \wedge\ \mathsf{separation?}(s + v'\Theta^+(s,v), v'')\end{array}$$

Proof. First we use exploit_pred_conflict [Lemma 8] to obtain clash?(s,v). Next, cross-multiplying the premise that defines v'_z yields

$$\Theta^+(s,v)v'_z = -\mathsf{sign}(v_z)H - s_z. \qquad (47)$$

Part 1 [$\mathsf{separation?}(s, v')$]: First we use $\mathsf{separation\_lem}$ [Lemma 1] to translate the starting point to $s + \Theta^+(s, v)v'$. The goal becomes:

$$\mathsf{separation?}(s + \Theta^+(s, v)v', v')$$

An application of $\mathsf{Circle\ Case\ Correctness}$ [Theorem 3] at $s + v'\Theta^+(s, v)$ will give us the desired result, provided that we can discharge its premises. We do so by proving that

$$|s_z + \Theta^+(s, v)v'_z| = H, \tag{48}$$
$$\mathsf{exit\_point?}(s + \Theta^+(s, v)v', v'), \tag{49}$$
$$(s_z + \Theta^+(s, v)v'_z)v'_z \le 0. \tag{50}$$

The claim (48) follows trivially from (47).

Next let us prove (49). The lemma $\mathsf{exit\_it\_is}$ [Lemma 7] is used to show $\mathsf{exit\_point?}(s + \Theta^+(s, v)v, v)$. But since $\mathsf{exit\_point?}$ only involves the x and y components of the vector, and we have $\mathsf{vertical\_change?}(v, v')$, we also get (49).

This leaves us to prove (50). The case $v'_z = 0$ is trivial, so assume $v'_z \ne 0$. First, lemma $\mathsf{signs\_ve\_z}$ [Lemma 15] yields $\mathsf{sign}(v'_z) = \mathsf{sign}(v_z)$. Substituting this and (47) in (50) and simplifying yields

$$-\mathsf{sign}(v'_z)Hv'_z \le 0 \tag{51}$$

A case split whether or not $v'_z \ge 0$, and expanding the definition of sign completes this part.

Part 2 [$\mathsf{separation?}(s + v'\Theta^+(s, v), v'')$]:
An application of $\mathsf{Circle\ Case\ Correctness}$ [Theorem 3] at $s + v'\Theta^+(s, v)$ will give us the desired result, provided that we can discharge its premises. We do so by proving that

$$|s_z + \Theta^+(s, v)v'_z| = H, \tag{52}$$
$$\mathsf{exit\_point?}(s + \Theta^+(s, v)v'_z, v''), \tag{53}$$
$$(s_z + \Theta^+(s, v)v'_z)v''_z \le 0. \tag{54}$$

The claim (52) follows trivially from (47).

Next let us prove (53). The lemma $\mathsf{exit\_it\_is}$ [Lemma 7] establishes $\mathsf{exit\_point?}(s + \Theta^+(s, v)v, v)$. We use the independence of x and y coordinates and the premises $\mathsf{vertical\_change?}\ (v, v')$ and $\mathsf{vertical\_change?}(v', v'')$ to derive (53).

This leaves to prove (54). First we simplify to get the goal:

$$s_z v''_z + v'_z v''_z \Theta^+(s, v) \le 0 \tag{55}$$

Next, we use $\mathsf{signs\_are\_opposite}$ [Lemma 14] and $\mathsf{signs\_vr\_z}$ [Lemma 16] to obtain $\mathsf{sign}(s_z) = -\mathsf{sign}(v_z)$ and $\mathsf{sign}(v''_z) = -\mathsf{sign}(s_z)$, respectively. Substituting these and (47) in (54) and simplifying yields

$$-\mathsf{sign}(v''_z)v''_z H \le 0.$$

A case split whether or not $v''_z \geq 0$, and expanding the definition of sign completes the proof.
□

We also prove a theorem that states the arrival in time:

Theorem 18 (vert_out_circle_timeliness)

$$\begin{aligned}
&\mathsf{hor\_sep?(s)} \ \wedge \ \neg\mathsf{pred\_sep?}(s, v, t'') \ \wedge \\
&\mathsf{vertical\_change?}(v, v') \ \wedge \ \mathsf{vertical\_change?}(v', v'') \ \wedge \\
&0 < t'' \ \wedge \ 0 < \Theta^+(s, v) \ \wedge \ \Theta^+(s, v) < t'' \ \wedge \\
&v'_z = \frac{-\mathsf{sign}(v_z)H - s_z}{\Theta^+(s, v)} \ \wedge \\
&v''_z = \frac{t''v_z - \Theta^+(s, v)v'_z}{t'' - \Theta^+(s, v)} \\
&\qquad \supset \ s + t''v = s + \Theta^+(s, v)v' + (t'' - \Theta^+(s, v))v''
\end{aligned}$$

Proof. First, we use $\mathsf{exploit\_pred\_conflict}$ [Lemma 8] to obtain $\mathsf{clash?(s,\ v)}$. Cross-multiplying the definition of v'_z yields: $v'_z\Theta^+(s, v) = -\mathsf{sign}(v_z)H - s_z$. Cross-multiplying the definition of v''_z yields: $v''_z(t'' - \Theta^+(s, v)) = t''v_z - \Theta^+(s, v)v'_z$. Then algebraic simplifications and rewriting will finish the proof.
□

5.2.3 One-circle

If $0 < \Theta^-(s, v)$ and $\Theta^+(s, v) < t''$ then for both $\varepsilon \in \{-1, 1\}$ there may be a one-circle solution. Figure 7 shows the case where a one-circle solution exists for each $\varepsilon = 1$ (left) and $\varepsilon = -1$ (right). If $\varepsilon s_z < H$ and $\varepsilon s''_z < H$, then we compute the vertical speeds

$$\begin{aligned}
v'_{oz} &= v_{iz} + \frac{\varepsilon H - s_z}{\Theta^-(s, v)}, \\
v''_{oz} &= v_{iz} + \frac{\varepsilon H - s''_z}{\Theta^+(s, v) - t''}.
\end{aligned}$$

If $v'_{oz} \neq v''_{oz}$, then t' is given by (33) which simplifies to

$$t' \ = \ t''\frac{v_{oz} - v''_{oz}}{v'_{oz} - v''_{oz}}.$$

In this case, there is a one-circle solution for ε given by v'_{oz}, v''_{oz}, and t'.

We remark that there are no vertical solutions that touch the lines, nor circle-circle solutions. The following theorem has been proved in PVS:

Theorem 19 (vert_one_circle_correctness)

$$
\begin{aligned}
&\textsf{hor\_sep?(s)} \ \wedge\ \neg\textsf{pred\_sep?}(s, v, t'') \ \wedge \\
&\textsf{vertical\_change?}(v + vi, v' + vi) \ \wedge\ \textsf{vertical\_change?}(v' + vi, v'' + vi) \ \wedge \\
&0 < t' < t'' \ \wedge\ 0 < \Theta^-(s, v) \ \wedge\ \Theta^+(s, v) < t'' \ \wedge \\
&s'' = s + t''v \ \wedge\ \varepsilon s_z < H \ \wedge\ \varepsilon s''_z < H \ \wedge \\
&v'_z = \frac{\varepsilon H - s_z}{\Theta^-(s, v)} \ \wedge \\
&v''_z = \frac{\varepsilon H - s''_z}{\Theta^+(s, v) - t''} \ \wedge \\
&v'_z \neq v''_z \ \wedge\ t' = t'' \frac{v_z - v''_z}{v'_z - v''_z} \\
&\qquad \supset\ \textsf{separation?}(s, v') \ \wedge\ \textsf{separation?}(s + t'v', v'')
\end{aligned}
$$

Proof. First, we use exploit_pred_conflict [Lemma 8] to obtain clash?(s, v).

Part 1 [separation?(s, v')]: First we cross-multiply the premise that defines v'_z to get:

$$\Theta^-(s, v)v'_z = \varepsilon H - s_z \tag{56}$$

Next we use separation_lem [Lemma 1] to translate the starting point to $s+\Theta^-(s, v)v'$. The goal becomes:

$$\textsf{separation?}(s + \Theta^-(s, v)v', v')$$

An application of theorem Circle Case Correctness [Theorem 3] at $s + v'\Theta^-(s, v)$ will give us the desired result, provided that we can discharge its premises. We do so by proving that

$$|s_z + \Theta^-(s, v)v'_z| = H, \tag{57}$$
$$\textsf{entry\_point?}(s + \Theta^-(s, v)v', v'), \tag{58}$$
$$(s_z + \Theta^-(s, v)v'_z)v'_z \geq 0. \tag{59}$$

The claim (57) follows immediately from (56).

Next let us prove (58). The lemma entry_it_is [Lemma 6] is used to show entry_point?$(s + \Theta^-(s, v)v, v)$. But since entry_point? only involves the x and y components of the vector, and vertical_change?(v, v') holds, we also get (58).

This leaves to prove (59). Substituting (56) simplifies the goal to

$$\varepsilon H v'_z \geq 0$$

From the premise $\varepsilon s_z < H$, equation (56) and the fact that $\varepsilon = 1 \ \vee\ \varepsilon = -1$ we obtain: $\varepsilon v'_z > 0$ from which the goal trivially follows.

Part 2 [separation?$(s + t'v', v'')$]: Cross-multiplying the premise that contains the definition of v''_z yields

$$(\Theta^+(s, v) - t'')v''_z = \varepsilon H - s''_z. \tag{60}$$

First we note that

$$s + t'v' = s'' - (t'' - t')v'' \tag{61}$$

This is easily put together from the premise $s'' = s + t''v$, the cross-multiplied definition of t', and the fact that the the x and y components of v, v' and v'' are the same. We use (61) to change the goal to

$$\mathsf{separation?}(s'' - (t'' - t')v'', v'').$$

Next we use separation_lem [Lemma 1] to translate the starting point to $s'' - (t'' - t')v'' + (\Theta^+(s, v) - t')v''$. Applying the equality

$$s'' - (t'' - t')v'' + (\Theta^+(s, v) - t')v'' = s'' + (\Theta^+(s, v) - t'')v''$$

this yields

$$\mathsf{separation?}(s'' + (\Theta^+(s, v) - t'')v'', v'').$$

An application of theorem Circle Case Correctness [Theorem 3] at $s'' + (\Theta^+(s, v) - t'')v''$ will give us the desired result, provided that we can discharge all its premises. We do so by proving that

$$|s''_z + (\Theta^+(s, v) - t'')v''_z| = H, \tag{62}$$

$$\mathsf{exit\_point?}(s'' + (\Theta^+(s, v) - t'')v'', v''), \tag{63}$$

$$(s''_z + (\Theta^+(s, v) - t'')v''_z)v''_z \leq 0. \tag{64}$$

The claim (62) reduces by (60) to the trivial $|\varepsilon H| = H$.

Next let us prove (63). The lemma exit_it_is [Lemma 7] shows $\mathsf{exit\_point?}(s'' + (\Theta^+(s, v) - t'')v, v)$. Then we exploit the fact that the x and y components are the same (because this is a vertical maneuver). This shows (63).

This leaves to show (64). Substituting (60) in (64) yields

$$\varepsilon H v''_z \leq 0 \tag{65}$$

Multiplication of (60) by ε and rewriting by $\varepsilon\varepsilon = 1$ yields

$$\varepsilon(\Theta^+(s, v) - t'')v''_z = H - \varepsilon s''_z.$$

By the premise $\varepsilon s''_z < H$, this is positive, so $\varepsilon v''_z \geq 0$ and so (65) follows.
□

5.3 Ground-Speed Cases

The ground-speed cases contain six independent solutions. There are four line and circle cases: line/line (Figure 4), line/circle (Figure 5), circle/line (Figure 6), and circle/circle (Figure 8) and two more cases: in-circle (Figure 9) and out-circle (Figure 10). Each case is proven separately; however, the line and circle cases are so similar that two intermediate lemmas (line_correctness and circle_correctness) are proven that greatly aid the proof of the more general theorems. For each case, three conditions must be proven—the correctness of the escape course, the correctness of the recovery course, and the timeliness of the complete maneuver. Correctness refers to the property that the aircraft do not violate vertical and horizontal separation criteria and timeliness refers to the property that the aircraft complete the maneuver at the time of the original operation.

All cases of the RR3D algorithm, we assume that there is a conflict along the original course[7] and that the relative velocity is defined as the ownship velocity minus the intruder velocity. These two conditions are captured in the $\mathsf{RR3D\_criteria?}(s, v, v_o, v_i, t'')$ predicate:

$$\begin{aligned}&\mathsf{RR3D\_criteria?}(s, v, v_o, v_i, t'') \;\Leftrightarrow \\ &\qquad \neg\mathsf{pred\_sep?}(s, v, t'') \;\wedge\; v = v_o - v_i. \end{aligned} \tag{66}$$

For the ground-speed only cases, we impose the constraint that only the ground speed of the ownship changes in each step. Formally, there are factors $k, j > 0$, such that

$$v'_{ox} = kv_{ox}, \qquad v'_{oy} = kv_{oy}, \qquad v'_{oz} = v_{oz}, \tag{67}$$

$$v''_{ox} = jv_{ox}, \qquad v''_{oy} = jv_{oy}, \qquad v''_{oz} = v_{oz}. \tag{68}$$

By the definition of the relative velocity we define the $\mathsf{ground\_speed\_only\_absolute?}(v, \lambda, v_o, v_i)$ predicate as follows

$$\begin{aligned}&\mathsf{ground\_speed\_only\_absolute?}(v, \lambda, v_o, v_i) \;\Leftrightarrow \\ &\quad \lambda > 0 \;\wedge\; v_x = \lambda v_{ox} - v_{ix} \;\wedge\; v_y = \lambda v_{oy} - v_{iy} \;\wedge\; v_z = v_{oz} - v_{iz}\end{aligned} \tag{69}$$

Using this predicate and the definitions in (67) and (68), we can constrain the relative escape and recovery velocities for the ground-speed only cases by

$$\mathsf{ground\_speed\_only\_absolute?}(v', k, v_o, v_i) \;\wedge\; \mathsf{ground\_speed\_only\_absolute?}(v'', j, v_o, v_i)$$

Occasionally we will use the derived property $v'_z = v_z = v''_z$ which is proven in the following lemma:

Lemma 20 (vert_speeds_equal)

$$\begin{array}{ll} & \mathsf{RR3D\_criteria?}(s, v, v_o, v_i, t'') \\ \wedge & \mathsf{ground\_speed\_only\_absolute?}(v', k, v_o, v_i) \\ \wedge & \mathsf{ground\_speed\_only\_absolute?}(v'', j, v_o, v_i) \\ \supset & v_z = v'_z \;\wedge\; v_z = v''_z \end{array}$$

Proof. From the ground_speed_only_absolute? premises we derive that v'_z and v''_z are equal to $v_{oz} - v_{iz}$. We also know from (66) that the relative velocity v is equal to

[7]in other words, there is not predicted separation along the original course

$v_o - v_i$. Breaking this equation into its z coordinates we see that $v_z = v_{oz} - v_{iz}$.
□

During the development of correctness and timeliness properties, we will need some properties common to all ground speed only cases. The time_definition? predicate combines the equations (31) and (32). It is defined as

$$\begin{aligned}&\mathsf{time\_definition?}(v, v', v'', t', t'') \;\Leftrightarrow\\ &\quad t'(v'_x - v''_x) = t''(v_x - v''_x) \;\wedge\; t'(v'_y - v''_y) = t''(v_y - v''_y).\end{aligned} \tag{70}$$

First we observe that $k \neq j$:

Lemma 21 (constants_not_equal)

$$\begin{array}{ll} & \mathsf{RR3D\_criteria?}(s, v, v_o, v_i, t'') \\ \wedge & \mathsf{hor\_speed\_gt\_0?}(v_o) \\ \wedge & \mathsf{ground\_speed\_only\_absolute?}(v', k, v_o, v_i) \\ \wedge & \mathsf{ground\_speed\_only\_absolute?}(v'', j, v_o, v_i) \\ \wedge & \mathsf{time\_definition?}(v, v', v'', t', t'') \\ \wedge & (\mathsf{separation?}(s, v') \;\vee\; \mathsf{separation?}(s + t''v, v'')) \\ \supset & k \neq j \end{array}$$

Proof. Since the ownship's ground speed must be different from zero (by the predicate hor_speed_gt_0?), either $v_{ox} \neq 0$ or $v_{oy} \neq 0$. If $v_{ox} \neq 0$ then we get

$$t'(k - j) = t''(1 - j) \tag{71}$$

from (31). If $v_{oy} \neq 0$ then we get (71) from (32).

We proceed with a proof by contradiction. Assume $k = j$. Observe that $t'' \geq 0$ follows from (66). If $t'' = 0$, then by (66), we must start and end in a conflict. Therefore neither of the two separation conditions can be true. This is a contradiction.

If $t'' > 0$ and $k = j$, then $0 = 1 - j$ follows from (71). So $k = j = 1$ which means that $v = v' = v''$ by (67) and (68). This contradicts the premise $\neg$pred_sep?(s, v, t''). Thus we have $k \neq j$.
□

If $k \neq j$ then t' is defined uniquely by (71) which is equivalent to

$$t' = \frac{t''(1 - j)}{k - j}. \tag{72}$$

In PVS, this is established in the following lemma.

Lemma 22 (escape_time_defined)

$$\begin{array}{ll} & \mathsf{RR3D\_criteria?}(s, v, v_o, v_i, t'') \\ \wedge & \mathsf{hor\_speed\_gt\_0?}(v_o) \\ \wedge & \mathsf{ground\_speed\_only\_absolute?}(v', k, v_o, v_i) \\ \wedge & \mathsf{ground\_speed\_only\_absolute?}(v'', j, v_o, v_i) \\ \wedge & \mathsf{time\_definition?}(v, v', v'', t', t'') \\ \wedge & k \neq j \\ \supset & t' = \dfrac{t''(1-j)}{k-j} \end{array}$$

Proof. Since the ownship's ground speed must be different from zero (by the predicate hor_speed_gt_0?), either $v_{ox} \neq 0$ or $v_{oy} \neq 0$. If $v_{ox} \neq 0$ then we get (71) from (31). If $v_{oy} \neq 0$ then we get (71) from (32). Since $k \neq j$ by assumption, using algebra we get the claim.
□

5.3.1 Timeliness

Recall the that timeliness condition states that the maneuver is completed at the same time as the original course and the resulting position is the same as the original ending position. The lemma that proves the timeliness condition is presented below. Since this lemma does not depend on the specific definitions of the k and j constants, all six of the ground-speed-only cases use the same timeliness lemma.

Lemma 23 (gs_timeliness)

$$\begin{array}{ll} & \mathsf{ground\_speed\_only\_absolute?}(v', k, v_o, v_i) \\ \wedge & \mathsf{ground\_speed\_only\_absolute?}(v'', j, v_o, v_i) \\ \wedge & v = v_o - v_i \\ \wedge & k \neq j \\ \wedge & t' = \dfrac{t''(1-j)}{(k-j)} \\ \supset & s + vt'' = (s + v't') + v''(t'' - t') \end{array}$$

Proof. Expand both ground_speed_only_absolute? predicates, then substitute the definitions of v' and v'' into the implication. Next, substitute the definition of v (provided in the assumptions) into the implication. Separate the implication into its x, y, and z coordinates and the result will be these three equations

$$\begin{aligned} v_{ox}t'' - v_{ix}t'' &= (jv_{ox} - v_{ix})t'' + \frac{t'' - jt''}{k-j}(kv_{ox} - v_{ix}) - \frac{t'' - jt''}{k-j}(jv_{ox} - v_{ix}), \\ v_{oy}t'' - v_{iy}t'' &= (jv_{oy} - v_{iy})t'' + \frac{t'' - jt''}{k-j}(kv_{oy} - v_{iy}) - \frac{t'' - jt''}{k-j}(jv_{oy} - v_{iy}), \\ v_{oz}t'' - v_{iz}t'' &= (v_{oz} - v_{iz})t'' + \frac{t'' - jt''}{k-j}(v_{oz} - v_{iz}) - \frac{t'' - jt''}{k-j}(v_{oz} - v_{iz}), \end{aligned}$$

each of which can be reduced by algebra.
□

5.3.2 Line and Circle Correctness

There are four line and circle cases line/line (Figure 4), line/circle (Figure 5), circle/line (Figure 6), and circle/circle (Figure 8). These cases are quite similar to each other and will be described together. Each of these cases can be viewed as a combination of an escape line subcase or an in-circle subcase combined with either a recovery line subcase or a out-circle subcase. If we prove the correctness of each of these four subcases then the subcases can be suitably assembled into proofs for each of the four line and circle cases. Recall that correctness means that during the escape or recovery course, there will be no violations of both horizontal and vertical separation constraints. The escape and recovery line subcases are proven with line_correctness [Lemma 24]. The in-circle and out-circle subcases are proven with circle_correctness [Lemma 27].

The conditions for both the escape and recovery line subcases can be covered with a single predicate line_case? which is defined as

$$\begin{aligned}&\mathsf{line\_case?}(s,v) \;\Leftrightarrow \\ &\qquad \mathsf{hor\_speed\_gt\_0?}(v) \;\wedge\; \mathsf{tangent\_condition?}(s,v).\end{aligned} \tag{73}$$

Instantiating this predicate as line_case?(s, v') yields an escape line subcase and instantiating it as line_case?$(s + t''v, v'')$ yields a recovery line case. Correctness can be proven without relying on either of these two instantiations: any parameters may be used. To prove the correctness of line subcases we use the lemma line_correctness.

Lemma 24 (line_correctness)

$$\begin{aligned}&\mathsf{hor\_speed\_gt\_0?}(v)\\ \wedge\;&\mathsf{tangent\_condition?}(s,v)\\ \supset\;&\mathsf{separation?}(s,v)\end{aligned}$$

Proof. From tau_is_tangent_pt [Lemma 4], we can show tangent_point?$(s+\tau(s,v)v, v)$ provided that hor_speed_gt_0?(v) and tangent_condition?(s, v). These two conditions are met since they are assumptions of line_correctness. Then by the line_case_correctness theorem [Theorem 2], tangent_point?$(s+\tau(s,v)v, v)$ implies separation?$(s+\tau(s,v)v, v)$. Finally observe that by separation_lem [Lemma 1], separation?$(s+\tau(s,v)v, v)$ is equivalent to separation?(s, v).
□

In the original paper [1] the line subcases are defined by the solutions of the equation

$$\begin{aligned}&k^2[D^2(v_{ox}^2 + v_{oy}^2) - (s_x v_{oy} - s_y v_{ox})^2]+\\ &2k[-D^2(v_{ox}v_{ix} + v_{oy}v_{iy}) + (s_x v_{oy} - s_y v_{ox})(s_x v_{iy} - s_y v_{ix})]+\\ &D^2(v_{ix}^2 + v_{iy}^2) - (s_x v_{iy} - s_y v_{ix})^2 = 0.\end{aligned} \tag{74}$$

In order to use line_correctness [Lemma 24] for them, we must show that equation (74) implies the tangent condition.

Lemma 25 (constant_for_line)

$$\begin{array}{ll} & \text{ground\_speed\_only\_absolute?}(v, k, v_o, v_i) \\ \wedge & a = D^2(v_{ox}^2 + v_{oy}^2) - (s_x v_{oy} - s_y v_{ox})^2 \\ \wedge & b = 2(-D^2(v_{ox}v_{ix} + v_{oy}v_{iy}) + (s_x v_{oy} - s_y v_{ox})(s_x v_{iy} - s_y v_{ix})) \\ \wedge & c = D^2(v_{ix}^2 + v_{iy}^2) - (s_x v_{iy} - s_y v_{ix})^2 \\ \wedge & 0 = ak^2 + bk + c \\ \supset & \text{tangent\_condition?}(s, v) \end{array}$$

Proof. Expanding the definitions of ground_speed_only_absolute? and tangent_condition? followed by extensive algebraic manipulation proves this lemma.
□

An alternate form of this lemma is useful when one is computing the roots of the quadratic instead of assuming that the quadratic relationship already holds. This alternate lemma is used in the proofs of the algorithmic form of the ground-speed only solutions.

Lemma 26 (constant_for_line_alt)

$$\begin{array}{ll} & \text{ground\_speed\_only\_absolute?}(v, k, v_o, v_i) \\ \wedge & a = D^2(v_{ox}^2 + v_{oy}^2) - (s_x v_{oy} - s_y v_{ox})^2 \\ \wedge & b = 2(-D^2(v_{ox}v_{ix} + v_{oy}v_{iy}) + (s_x v_{oy} - s_y v_{ox})(s_x v_{iy} - s_y v_{ix})) \\ \wedge & c = D^2(v_{ix}^2 + v_{iy}^2) - (s_x v_{iy} - s_y v_{ix})^2 \\ \wedge & (a = 0 \ \wedge \ b \neq 0 \ \wedge \ k = -c/b \ \vee \\ & a \neq 0 \ \wedge \ b^2 - 4ac \geq 0 \ \wedge \ (k = \text{root}(-1, a, b, c) \ \vee \ k = \text{root}(1, a, b, c))) \\ \supset & \text{tangent\_condition?}(s, v) \end{array}$$

Recall that $\text{root}(-1, a, b, c)$ and $\text{root}(1, a, b, c)$ denote the two roots of the quadratic equation with coefficients a, b, and c.
Proof. The proof proceeds as two cases.

Case 1 [$a = 0 \ \wedge \ b \neq 0 \ \wedge \ k = -c/b$]. Instantiate constant_for_line [Lemma 25] then substitute the definitions $a = 0$ and $k = -c/b$ into the quadratic equation from this lemma. Reduce with algebra.

Case 2 [$a \neq 0 \ \wedge \ b^2 - 4ac \geq 0 \ \wedge \ (k = \text{root}(-1, a, b, c) \ \vee \ k = \text{root}(1, a, b, c))$]. Using (24), we get $ak^2 + bk + c = 0$. Then instantiating constant_for_line [Lemma 25] discharges this proof.
□

The correctness of both the in-circle and out-circle subcases are proven in circle_correctness [Lemma 27]. The conditions for an in-circle course are captured in the predicate in_circle_case? (s, v, v'', t''), which is defined as:

$$\begin{aligned}&\textsf{in\_circle\_case?}(s, v, v'', t'') \Leftrightarrow \\ &\quad v_z \neq 0 \,\wedge\, \textsf{entry\_point?}((s + t''v) + (\theta^+(s_z, v_z) - t'')v'', v'')\end{aligned} \tag{75}$$

The conditions for a out-circle course are captured in the predicate out_circle_case?(s, v'), which is defined as:

$$\begin{aligned}&\textsf{out\_circle\_case?}(s, v') \Leftrightarrow \\ &\quad v'_z \neq 0 \,\wedge\, \textsf{exit\_point?}(s + \theta^-(s_z, v'_z)v', v')\end{aligned} \tag{76}$$

Observing the similarities between the two circle subcases allows the definition and proof of a single lemma, circle_correctness, that will help in each subcase. This lemma should be instantiated at the point s along the v' vector for an escape course (the out_circle_case? case) and at the point $s + t''v$ along the v'' vector for a recovery course (the in_circle_case? case).

Lemma 27 (circle_correctness)

$$\begin{array}{ll} & v_z \neq 0 \\ \wedge & (\textsf{exit\_point?}(s + \theta^-(s_z, v_z)v, v) \,\vee\, \textsf{entry\_point?}(s + \theta^+(s_z, v_z)v, v)) \\ \supset & \textsf{separation?}(s, v)\end{array}$$

Proof. The proof proceeds as one of two cases: either the point is an entry point or an exit point. For each case, the $v_z \neq 0$ condition is required to ensure that the $\theta^{\pm}(s_z, v_z)$ expression is defined.

Case 1 [exit_point?$(s + \theta^-(s_z, v_z)v, v)$]. Instantiating the circle_case_correctness theorem [Theorem 3] at the point $s+\theta^-(s_z, v_z)v$ along the vector v implies separation?$(s+\theta^-(s_z, v_z)v, v)$, provided that we can discharge its premises. We do so by proving that

$$|s_z + \theta^-(s_z, v_z)v_z| \geq H, \tag{77}$$
$$\textsf{exit\_point?}(s + \theta^-(s_z, v_z)v, v), \tag{78}$$
$$(s_z + \theta^-(s_z, v_z)v_z)v_z \leq 0. \tag{79}$$

Condition (77) is met by applying reaching_H_theta [Lemma 10]. The lemma states that $|s_z + \theta^-(s_z, v_z)v_z| = H$, and we have $H \geq H$. Condition (78) is met trivially by the exit_point? assumption. Lemma vertical_entry_exit_condition [Lemma 11] discharges (79). Since these three conditions have been met, the circle_case_correctness theorem yields separation?$(s + \theta^-(s_z, v_z)v, v)$. Applying separation_lem [Lemma 1], separation?$(s + \theta^-(s_z, v_z)v, v)$ is equivalent to separation?(s, v).

Case 2 [entry_point?$(s+\theta^+(s_z, v_z)v, v)$]. Like Case 1, but with θ^+ and entry_point? instead of θ^- and exit_point?, respectively.
□

Circle subcases are defined in the original paper [1] by certain defining equations. Therefore, we must show that those equations imply an escape course or a recovery course. First we will show that the quadratic presented in the paper

$$\begin{aligned}&\lambda^2 t^2(v_{ox}^2 + v_{oy}^2)+\\ &2\lambda t(s_x v_{ox} - t v_{ix} v_{ox} + s_y v_{oy} - t v_{iy} v_{oy})+\\ &(s_x - t v_{ix})^2 + (s_y - t v_{iy})^2 - D^2 = 0.\end{aligned} \tag{80}$$

for both subcases implies that $s+tv$ is at the cylinder lateral surface. For convenience we introduce a predicate on_cyl? for this purpose, defined by

$$\text{on\_cyl?}(s) \;\Leftrightarrow\; s_x^2 + s_y^2 = D^2. \tag{81}$$

The value t is instantiated by $\theta^-(s_z, v_z)$ for an escape course and by $\theta^+(s_z, v_z)-t''$ for a recovery course. The value λ can be the constant k for an escape course or the constant j for a recovery course.

Lemma 28 (constant_for_circle)

$$\begin{array}{ll} & \text{ground\_speed\_only\_absolute?}(v, \lambda, v_o, v_i)\\ \wedge & a = t^2(v_{ox}^2 + v_{oy}^2)\\ \wedge & b = 2t(s_x v_{ox} - t v_{ix} v_{ox} + s_y v_{oy} - t v_{iy} v_{oy})\\ \wedge & c = (s_x - t v_{ix})^2 + (s_y - t v_{iy})^2 - D^2\\ \wedge & 0 = a\lambda^2 + b\lambda + c\\ \supset & \text{on\_cyl?}(s + tv)\end{array}$$

Proof. Expanding the definitions of ground_speed_only_absolute?, and on_cyl? followed by extensive algebraic manipulation proves this lemma.
□

In a similar way to how both lemmas constant_for_line [Lemma 25] and constant_for_line_alt [Lemma 26] are developed to define the constants of a line case, an alternate form of constant_for_circle [Lemma 28] is useful when one is computing the roots of the quadratic instead of assuming that the quadratic relationship already holds. This alternate lemma is used in the proofs of the algorithmic form of the ground-speed only solutions.

Lemma 29 (constant_for_circle_alt)

$$\begin{array}{ll} & \textsf{ground\_speed\_only\_absolute?}(v, \lambda, v_o, v_i) \\ \wedge & a = t^2(v_{ox}^2 + v_{oy}^2) \\ \wedge & b = 2t(s_x v_{ox} - t v_{ix} v_{ox} + s_y v_{oy} - t v_{iy} v_{oy}) \\ \wedge & c = (s_x - t v_{ix})^2 + (s_y - t v_{iy})^2 - D^2 \\ \wedge & (a = 0 \ \wedge \ b \neq 0 \ \wedge \ \lambda = -c/b \ \vee \\ & a \neq 0 \ \wedge \ b^2 - 4ac \geq 0 \ \wedge \ (\lambda = \textsf{root}(-1, a, b, c) \ \vee \ \lambda = \textsf{root}(1, a, b, c))) \\ \supset & \textsf{on\_cyl?}(s + tv) \end{array}$$

Proof. The proof proceeds as two cases.

Case 1 $[a = 0 \ \wedge \ b \neq 0 \ \wedge \ \lambda = -c/b]$. Instantiate constant_for_circle [Lemma 28] then substitute the definitions $a = 0$ and $\lambda = -c/b$ into the quadratic equation from this lemma. Reduce with algebra.

Case 2 $[a \neq 0 \ \wedge \ b^2 - 4ac \geq 0 \ \wedge \ (\lambda = \textsf{root}(-1, a, b, c) \ \vee \ \lambda = \textsf{root}(1, a, b, c))]$. Using (24) we get $a\lambda^2 + b\lambda + c = 0$. Then instantiating constant_for_circle [Lemma 28] discharges this proof.

□

From the original paper [1], the equations used to define a circle subcase for an escape course include equation (80) and require the translated location multiplied by the escape velocity must be greater than or equal to zero, that is,

$$(s_x + t(\lambda v_{ox} - v_{ix}))(\lambda v_{ox} - v_{ix}) + (s_y + t(\lambda v_{oy} - v_{iy}))(\lambda v_{oy} - v_{iy}) \geq 0. \qquad (82)$$

In this paper, we say that an out-circle subcase (76) must be an exit point. Since we have already shown that (80) implies on_cyl?, we now need to show that the on_cyl? predicate and (82) imply an exit point.

Lemma 30 (constant_for_circle_exit)

$$\begin{array}{ll} & \textsf{ground\_speed\_only\_absolute?}(v, \lambda, v_o, v_i) \\ \wedge & \textsf{on\_cyl?}(s + tv) \\ \wedge & (s_x + t(\lambda v_{ox} - v_{ix}))(\lambda v_{ox} - v_{ix}) + (s_y + t(\lambda v_{oy} - v_{iy}))(\lambda v_{oy} - v_{iy}) \geq 0 \\ \supset & \textsf{exit\_point?}(s + tv, v) \end{array}$$

Proof. Expansion of exit_point? and on_cyl? solves the goal.

□

From the original paper [1], the equations used to define a circle subcase for an recovery course include equation (80) and require

$$(s_x + t(\lambda v_{ox} - v_{ix}))(\lambda v_{ox} - v_{ix}) + (s_y + t(\lambda v_{oy} - v_{iy}))(\lambda v_{oy} - v_{iy}) \leq 0. \qquad (83)$$

In this paper we say that an in-circle subcase (75) must be an entry point. Since we have already shown that (80) implies on_cyl?, we now need to show that the on_cyl? predicate and (83) imply an entry point.

Lemma 31 (constant_for_circle_entry)

$$\begin{array}{ll} & \textsf{ground\_speed\_only\_absolute?}(v, \lambda, v_o, v_i) \\ \wedge & \textsf{on\_cyl?}(s + tv) \\ \wedge & (s_x + t(\lambda v_{ox} - v_{ix}))(\lambda v_{ox} - v_{ix}) + (s_y + t(\lambda v_{oy} - v_{iy}))(\lambda v_{oy} - v_{iy}) \leq 0 \\ \supset & \textsf{entry\_point?}(s + tv, v) \end{array}$$

Proof. Expansion of entry_point? and on_cyl? solves the goal.
□

5.3.3 Line and Circle Cases

We next present the proofs of the four line and circle cases line_line [Theorem 32], circle_line [Theorem 33], and line_circle [Theorem 34], circle_circle [Theorem 35]. For each case three conditions must be proven: the correctness of escape course, the correctness of the recovery course, and the timeliness of the maneuver. Recall that correctness refers to the property that the aircraft do not violate vertical and horizontal separation criteria and timeliness refers to the aircraft completing the maneuver at the time of the original operation. To prove correctness for a line course (either escape or recovery) we use line_correctness [Lemma 24]. To prove correctness for a circle course (either escape or recovery) we use circle_correctness [Lemma 27]. Finally, to prove timeliness we use gs_timeliness [Lemma 23]. Three predicates are used to define the type of escape and recovery course: line_case? predicate (73) in_circle_case? predicate (75) out_circle_case? predicate (76)

For the cases involving an escape line course, we check for sanity that $0 < \tau(s, v') < t'$. For the cases involving a recovery line course, we check for sanity that $t' < \tau(s'', v'') + t'' < t''$. Furthermore, for the cases involving a circle course, we assume that relative vertical speed is not zero, i.e., $v_z \neq 0$; otherwise, there is no solution. In all the cases, we check for sanity that $k, j > 0$.

The first case we will consider is the case with a line escape course and a line recovery course.

Theorem 32 (line_line)

$$\begin{array}{rl} & \mathsf{RR3D\_criteria?}(s, v, v_o, v_i, t'') \\ \wedge & \mathsf{hor\_speed\_gt\_0?}(v_o) \\ \wedge & \mathsf{ground\_speed\_only\_absolute?}(v', k, v_o, v_i) \\ \wedge & \mathsf{ground\_speed\_only\_absolute?}(v'', j, v_o, v_i) \\ \wedge & \mathsf{line\_case?}(s, v') \\ \wedge & \mathsf{line\_case?}(s + t''v, v'') \\ \wedge & \mathsf{time\_definition?}(v, v', v'', t', t'') \\ \supset & \\ & \mathsf{separation?}(s, v') \\ \wedge & \mathsf{separation?}(s + t''v, v'') \\ \wedge & s + t''v = (s + t'v') + (t'' - t')v'' \end{array}$$

Proof. Step 1 [Escape Correctness]. First instantiate line_correctness? [Lemma 24] for the escape course. This discharges the claim separation?(s, v').

Step 2 [Recovery Correctness]. Next use [Lemma 24] again for the recovery course. That is, instantiate the starting point is set to $s + t''v$, and the velocity to v''. This discharges the claim separation?$(s + t''v, v'')$.

Step 3 [Timeliness]. Lemma constants_not_equal [Lemma 21] supplies $k \neq j$. With it, escape_time_defined [Lemma 22] yields a formula for t'. Both formulas are in turn used by gs_timeliness [Lemma 23] to yield the claim $s + t''v = (s + t'v') + (t'' - t')v''$. The condition $v = v_o - v_i$ is discharged by expanding the RR3D_criteria? premise (66).
□

Next we will consider the case with a circle escape course and a line recovery course.

Theorem 33 (circle_line)

$$\begin{array}{rl} & \mathsf{RR3D\_criteria?}(s, v, v_o, v_i, t'') \\ \wedge & \mathsf{hor\_speed\_gt\_0?}(v_o) \\ \wedge & \mathsf{ground\_speed\_only\_absolute?}(v', k, v_o, v_i) \\ \wedge & \mathsf{ground\_speed\_only\_absolute?}(v'', j, v_o, v_i) \\ \wedge & \mathsf{out\_circle\_case?}(s, v') \\ \wedge & \mathsf{line\_case?}(s + t''v, v'') \\ \wedge & \mathsf{time\_definition?}(v, v', v'', t', t'') \\ \supset & \\ & \mathsf{separation?}(s, v') \\ \wedge & \mathsf{separation?}(s + t''v, v'') \\ \wedge & s + t''v = (s + t'v') + (t'' - t')v'' \end{array}$$

Proof. Step 1 [Escape Correctness]. First instantiate $\mathsf{circle\_correctness}$ [Lemma 27] at the starting point s along the velocity vector v'. This lemma discharges the $\mathsf{separation?}(s, v')$ predicate provided that we can discharge its premises. We do so by proving that

$$v'_z \neq 0,$$
$$\mathsf{exit\_point?}(s + \theta^-(s_z, v'_z)v', v').$$

Both conditions are given by the $\mathsf{out\_circle\_case?}$ premise (76).

Step 2 [Recovery Correctness]. Next $\mathsf{line\_correctness}$ [Lemma 24] is used for the recovery course. That is, instantiate the starting point is set to $s + t''v$, and the velocity with v''. This discharges the claim $\mathsf{separation?}(s + t''v, v'')$.

Step 3 [Timeliness]. Exactly as in [Theorem 32].

□

Next we will consider the case with a line escape course and a circle recovery course.

Theorem 34 (line_circle)

$$\begin{array}{rl}
 & \mathsf{RR3D\_criteria?}(s, v, v_o, v_i, t'') \\
\wedge & \mathsf{hor\_speed\_gt\_0?}(v_o) \\
\wedge & \mathsf{ground\_speed\_only\_absolute?}(v', k, v_o, v_i) \\
\wedge & \mathsf{ground\_speed\_only\_absolute?}(v'', j, v_o, v_i) \\
\wedge & \mathsf{line\_case?}(s, v') \\
\wedge & \mathsf{in\_circle\_case?}(s, v, v'', t'') \\
\wedge & \mathsf{time\_definition?}(v, v', v'', t', t'') \\
\supset & \\
 & \mathsf{separation?}(s, v') \\
\wedge & \mathsf{separation?}(s + t''v, v'') \\
\wedge & s + t''v = (s + t'v') + (t'' - t')v''
\end{array}$$

Proof. Step 1 [Escape Correctness]. First instantiate $\mathsf{line\_correctness}$ [Lemma 24] for the escape course. This discharges the $\mathsf{separation?}(s, v')$ predicate.

Step 2 [Recovery Correctness]. Lemma $\mathsf{circle\_correctness}$ [Lemma 27] is used for the recovery course. This lemma is instantiated with a starting point of $s + t''v$ along the v'' velocity vector. This lemma discharges the $\mathsf{separation?}(s + t''v, v'')$ claim provided that we can discharge its premises. We do so by proving

$$v''_z \neq 0, \qquad (84)$$
$$\mathsf{entry\_point?}((s + t''v) + \theta^+(s_z + t''v_z, v''_z)v'', v''). \qquad (85)$$

From the first part of the $\mathsf{in\_circle\_case?}$ predicate (75), we see $v_z \neq 0$. Using $\mathsf{vert\_speeds\_equal}$ [Lemma 20] we see that $v_z = v''_z$. Since $v_z \neq 0$ and $v_z = v''_z$, condition (84) is satisfied.

In order to prove (85), we expand the in_circle_case? premise, and show that

$$(s + t''v) + \theta^+(s_z + t''v_z, v''_z)v'' = (s + t''v) + (\theta^+(s_z, v_z) - t'')v''. \tag{86}$$

For, if (86) is true then the two entry_point? statements coincide. Lemma theta_translation [Lemma 12] turns (86) into

$$(s + t''v) + (\theta^+(s_z, v''_z) - t'')v'' = (s + t''v) + (\theta^+(s_z, v_z) - t'')v''$$

Since $v''_z = v_z$ by [Lemma 20], the two sides of the equation are equal; so condition (85) holds. This implies separation?$(s + t''v, v'')$.

Step 3 [Timeliness]. Exactly as in [Theorem 32].

□

Finally we prove correctness in the case of a circle escape course and a circle recovery course.

Theorem 35 (circle_circle)

$$\begin{array}{ll}
 & \mathsf{RR3D\_criteria?}(s, v, v_o, v_i, t'') \\
\wedge & \mathsf{hor\_speed\_gt\_0?}(v_o) \\
\wedge & \mathsf{ground\_speed\_only\_absolute?}(v', k, v_o, v_i) \\
\wedge & \mathsf{ground\_speed\_only\_absolute?}(v'', j, v_o, v_i) \\
\wedge & \mathsf{out\_circle\_case?}(s, v') \\
\wedge & \mathsf{in\_circle\_case?}(s, v, v'', t'') \\
\wedge & \mathsf{time\_definition?}(v, v', v'', t', t'') \\
\supset & \\
 & \mathsf{separation?}(s, v') \\
\wedge & \mathsf{separation?}(s + t''v, v'') \\
\wedge & s + t''v = (s + t'v') + (t'' - t')v''
\end{array}$$

Proof. Step 1 [Escape Correctness]. First instantiate circle_correctness [Lemma 27] at the starting point s along the velocity vector v'. This lemma discharges the separation?(s, v') predicate provided that (A) $v'_z \neq 0$ and that (B) the point $s + \theta^-(s_z, v'_z)v'$ along the v' velocity vector is an exit_point?. Both of these conditions are given in the out_circle_case? predicate (76).

Step 2 [Recovery Correctness]. Lemma circle_correctness [Lemma 27] is instantiated with a starting point of $s + t''v$ along the v'' velocity vector. This lemma discharges the separation?$(s+t''v, v'')$ claim provided that we can discharge its premises. We do so by proving

$$v''_z \neq 0, \tag{87}$$

$$\mathsf{entry\_point?}((s + vt'') + \theta^+(s_z + t''v_z, v''_z)v'', v''). \tag{88}$$

From the first part of the in_circle_case? predicate (75), we get $v_z \neq 0$. Using vert_speeds_equal [Lemma 20] we see that $v_z = v''_z$. Since $v_z \neq 0$ and $v_z = v''_z$, the condition (87) is satisfied.

In order to prove (88), we expand the in_circle_case? premise, and show that

$$(s + vt'') + \theta^+(z(s + t''v), v''_z)v'' = (s + vt'') + (\theta^+(s_z, v_z) - t'')v''. \qquad (89)$$

For, if (89) is true, then the two entry_point? statements coincide. Lemma theta_translation [Lemma 12] turns (89) into

$$(s + vt'') + (\theta^+(s_z, v''_z) - t'')v'' = (s + vt'') + (\theta^+(s_z, v_z) - t'')v''$$

Since $v''_z = v_z$ by [Lemma 20], the two sides of the equation are equal; hence condition (88) holds. This proves separation?$(s + t''v, v'')$.

□

5.3.4 In-Circle Case

This section contains the proof of the in-circle case (Figure 9). Like the proofs of the line and circle cases, three conditions must be proven: the correctness of escape course, the correctness of the recovery course, and the timeliness of the maneuver. Recall that correctness refers to the property that the aircraft do not violate vertical and horizontal separation criteria and timeliness refers to the completing the maneuver at the time of the original operation. To prove correctness of the escape course we use circle_correctness [Lemma 27]. and to prove timeliness we use gs_timeliness [Lemma 23]. To prove the correctness of the recovery course, neither of the correctness lemmas can help, so the proofs are developed from lower level lemmas. These lemmas are presented here.

Lemma 36 (vertical_criterion_sz_vz_ge_0)

$$\begin{aligned} &|s_z| = H \ \wedge \ s_z v_z \geq 0 \\ &\qquad \supset \ \forall T : T \geq 0 \ \supset \ |s_z + T v_z| \geq H \end{aligned}$$

Proof. Consider four cases:
Case 1 [$v_z \leq 0 \ \wedge \ s_z + Tv_z < 0$]: Then $s_z \leq 0$ by $s_z v_z \geq 0$, so $s_z = -H$. The goal $-(s_z + Tv_z) \geq H$ follows from $T \geq 0$ and $v_z \leq 0$.
Case 2 [$v_z \leq 0 \ \wedge \ s_z + Tv_z > 0$]: From $T \geq 0$ we get $Tv_z \leq 0$. Adding this to $s_z \leq 0$, it yields $s_z + Tv_z \leq 0$ which contradicts the assumption. So this case is impossible.
Case 3 [$v_z \geq 0 \ \wedge \ s_z + Tv_z < 0$]: Then $s_z \geq 0$ by $s_z v_z \geq 0$. From $T \geq 0$ we get $Tv_z \geq 0$. Adding this to $s_z \geq 0$, it yields $s_z + Tv_z \geq 0$ which contradicts the assumption $s_z + Tv_z < 0$. So this case is impossible.
Case 4 [$v_z \geq 0 \ \wedge \ s_z + Tv_z > 0$]: Then $s_z \geq 0$ by $s_z v_z \geq 0$, so $s_z = H$. The goal $s_z + Tv_z \geq H$ follows from $T \geq 0$ and $v_z \geq 0$.

□

Lemma 37 (vertical_criterion_sz_vz_le_0)

$$\begin{aligned} &|s_z| = H \ \wedge \ s_z v_z \leq 0 \\ &\qquad \supset \ \forall T : T \leq 0 \ \supset \ |s_z + T v_z| \geq H \end{aligned}$$

Proof. The proof is similar to vertical_criterion_sz_vz_ge_0.
□

Theorem 38 (in_circle)

$$\begin{array}{ll} & \mathsf{RR3D\_criteria?}(s, v, v_o, v_i, t'') \\ \wedge & \mathsf{hor\_speed\_gt\_0?}(v_o) \\ \wedge & \mathsf{ground\_speed\_only\_absolute?}(v', k, v_o, v_i) \\ \wedge & \mathsf{ground\_speed\_only\_absolute?}(v'', j, v_o, v_i) \\ \wedge & v_z \neq 0 \\ \wedge & 0 < \theta^+(s_z, v_z) \\ \wedge & \theta^+(s_z, v_z) < t'' \\ \wedge & \mathsf{entry\_point?}(s + \theta^+(s_z, v_z)v', v') \\ \wedge & \mathsf{time\_definition?}(v, v', v'', t', t'') \\ \wedge & t' = \theta^+(s_z, v_z) \end{array}$$

$$\supset$$

$$\begin{array}{ll} & \mathsf{separation?}(s, v') \\ \wedge & \mathsf{pred\_sep?}(s + t'v', v'', t'' - t') \\ \wedge & s + t''v = (s + t'v') + (t'' - t')v'' \end{array}$$

Observe, that the pred_sep? condition is used here instead of the separation? condition. separation? says that the if the aircraft continue to fly with the same relative velocity, then the two aircraft will be separated for all time. This is a much stronger condition than is required. The pred_sep? condition says that the aircraft will be separated for (at least) the given amount of time.

Proof. Step 1 [Escape Correctness]. First instantiate circle_correctness [Lemma 27] at the starting point s along the velocity vector v'. This lemma discharges the separation?(s, v') predicate provided that $v'_z \neq 0$. This can be verified since one assumption is $v_z \neq 0$ and since $v_z = v'_z$ from [Lemma 20].

Step 2 [Recovery Correctness]. Expanding pred_sep? leaves us with

$$\mathsf{hor\_sep?}((s_z + t'v'_z) + tv''_z),$$
$$\mathsf{vert\_sep?}((s_z + t'v'_z) + tv''_z). \tag{90}$$

We only need to prove one of these conditions and we choose to prove (90). Expansion of vert_sep? and usage of the definition of t' leaves to prove that for all $0 < t < t'' - t'$,

$$|(s_z + \theta^+(s_z, v_z)v'_z) + tv''_z| \geq H. \tag{91}$$

Let us instantiate vertical_criterion_sz_vz_ge_0 [Lemma 36] at the point $s+\theta^+(s_z, v''_z)v''$ along the velocity vector v''. Then (91) will be satisfied by vertical_criterion_sz_vz_ge_0 provided that (A) $|(s_z+\theta^+(s_z, v_z)v'_z)+tv''_z| = H$ and that (B) $(s_z+\theta^+(s_z, v''_z)v''_z)v''_z \geq 0$. Condition (A) is met by applying reaching_H_theta [Lemma 10]. Condition (B) is met by applying vertical_entry_exit_condition [Lemma 11].

Step 3 [Timeliness]. Exactly as in [Theorem 32].
□

5.3.5 Out-Circle Case

This section contains the proof of the out-circle case (Figure 10). Like the proofs of the line and circle cases, three conditions must be proven: the correctness of escape course, the correctness of the recovery course, and the timeliness of the maneuver. Recall that correctness refers to the property that the aircraft do not violate vertical and horizontal separation criteria and timeliness refers to the completing the maneuver at the time of the original operation. To prove correctness of the recovery course we use circle_correctness [Lemma 27]. and to prove timeliness we use gs_timeliness [Lemma 23]. To prove the correctness of the escape course, neither of the correctness lemmas can help, so the proofs are developed from lower level lemmas. Unlike all the other ground-speed only cases, this case must explicitly state the premise that $k \neq j$.

Theorem 39 (out_circle)

$$
\begin{array}{ll}
& \textsf{RR3D\_criteria?}(s, v, v_o, v_i, t'') \\
\wedge & \textsf{hor\_speed\_gt\_0?}(v_o) \\
\wedge & \textsf{ground\_speed\_only\_absolute?}(v', k, v_o, v_i) \\
\wedge & \textsf{ground\_speed\_only\_absolute?}(v'', j, v_o, v_i) \\
\wedge & v_z \neq 0 \\
\wedge & 0 < \theta^-(s_z, v_z) \\
\wedge & \theta^-(s_z, v_z) < t'' \\
\wedge & \textsf{exit\_point?}(s + \theta^-(s_z, v_z)v', v'') \\
\wedge & \textsf{time\_definition?}(v, v', v'', t', t'') \\
\wedge & t' = \theta^-(s_z, v_z) \\
\wedge & k \neq j \\
\supset & \\
& \textsf{pred\_sep?}(s, v', t') \\
\wedge & \textsf{separation?}(s + t''v, v'') \\
\wedge & s + t''v = (s + t'v') + (t'' - t')v''
\end{array}
$$

Observe, that the pred_sep? condition is used here instead of the separation? condition. separation? says that the if the aircraft continue to fly with the same relative velocity, then the two aircraft will be separated for all time. This is a much stronger condition than is required. The pred_sep? condition says that the aircraft will be separated for (at least) the given amount of time.

Proof. Step 1 [Escape Correctness]. Let us prove pred_sep?(s, v', t'). Expanding the pred_sep? leaves to prove one of

$$
\begin{array}{c}
\textsf{hor\_sep?}(s_z + tv'_z), \\
\textsf{vert\_sep?}(s_z + tv'_z).
\end{array}
\tag{92}
$$

We choose to prove (92). Expansion of vert_sep? leaves to prove

$$|s_z + tv'_z| \geq H \tag{93}$$

for all t such that $0 \leq t \leq t'$. If we instantiate vertical_criterion_sz_vz_le_0 [Lemma 37] at the point $s + \theta^-(s_z, v_z)v'$ along the velocity vector v', then it will yield the result

$$|s_z + \hat{t}v''_z + \theta^-(s_z, v''_z)v''_z| \geq H \tag{94}$$

for all $\hat{t} \leq 0$, provided that (A) $|s_z+\theta^-(s_z, v_z)v'_z| = H$ and (B) $(s_z+\theta^-(s_z, v_z)v'_z)v'_z \leq 0$. Condition (A) is met by $v_z = v'_z$ from [Lemma 20] and by reaching_H_theta [Lemma 10]. Condition (B) is met by vertical_entry_exit_condition [Lemma 11].

To finish the proof of pred_sep?(s, v', t') we choose $\hat{t} = t - \theta^-(s_z, v''_z)$. Then from (93) and since $v'_z = v''_z$

$$s_z + tv'_z = s_z + \hat{t}v'_z + \theta^-(s_z, v''_z)v'_z = s_z + \hat{t}v''_z + \theta^-(s_z, v''_z)v''_z \tag{95}$$

From this result the inequalities (94) and (93) coincide. We moreover have $\hat{t} \leq 0$ since $t \leq t'$ and $t' = \theta^-(s_z, v''_z)$.

Step 2 [Recovery Correctness]. Lemma circle_correctness [Lemma 27] is instantiated with a starting point of $s + t''v$ along the v'' velocity vector. This lemma discharges the separation?$(s+t''v, v'')$ claim provided that we can discharge its premises. We do so by proving

$$v''_z \neq 0, \tag{96}$$

$$\text{exit\_point?}((s + t''v) + \theta^-(s_z + t''v_z, v''_z)v'', v''). \tag{97}$$

Lemma vert_speeds_equal [Lemma 20] yields $v_z = v''_z$. Since $v_z \neq 0$ by premise, the condition (96) holds. In order to prove (97), we show

$$(s + t''v) + \theta^-(s_z + t''v_z, v''_z)v'' = s + \theta^-(s_z, v_z)v', \tag{98}$$

which entails the exit_point? premise and claim (97) coincide. Lemma theta_translation [Lemma 12] yields $\theta^-(s_z + t''v_z, v''_z) = \theta^-(s_z, v''_z) - t''$. Since $v''_z = v_z$ by [Lemma 20], the latter is equal to $\theta^-(s_z, v_z) - t''$. This turns (98) into

$$(s + t''v) + (\theta^-(s_z, v_z) - t'')v'' = s + \theta^-(s_z, v_z)v'$$

which is a restatement of the timeliness condition and can be proven with gs_timeliness [Lemma 23] as in Step 3 below. Hence condition (97) holds. This finishes the proof of separation?$(s + t''v, v'')$.

Step 3 [Timeliness]. Exactly as in [Theorem 32].

□

5.4 Correctness of Heading Case

In this section we prove correctness of escape courses that only change the heading, and recovery courses that only change the heading and the ground speed, of the ownship's velocity vector. This is expressed formally as

$$v'^2_{ox} + v'^2_{oy} = v^2_{ox} + v^2_{oy} \qquad \text{and} \qquad v'_{oz} = v_{oz} = v''_{oz}.$$

For the various solutions, satisfaction of this property is not obvious; it has to be explicitly verified.

We have 6 independent solution categories: line/line (ll), line/circle (lc), circle/line (cl), circle/circle (cc), in-circle (ic), and out-circle (oc).

5.4.1 Important Lemmas

The proofs of the main theorems for these categories are facilitated by correctness and heading only lemmas for each of the following cases: line escape, line recovery, circle escape, circle recovery, and in_circle recovery, which are reused several times to establish the main results. Several timeliness lemmas (timeliness, alpha_timeliness, vor_timeliness) establish that the evasive maneuver reaches the final destination at the same time as the original trajectory.

5.4.2 The alpha_calc Function

The following function is used throughout this section. This function is used in the computation of the heading change.

$$\begin{aligned}\mathsf{alpha\_calc}(\varepsilon, s) \quad \equiv \quad & \mathsf{IF}\ D^2 = {s_x}^2\ \mathsf{THEN}\ \frac{{s_y}^2 - D^2}{2 s_x s_y} \\ & \mathsf{ELSE}\ \frac{-s_x s_y + \varepsilon D \sqrt{{s_x}^2 + {s_y}^2 - D^2}}{D^2 - {s_x}^2} \\ & \mathsf{ENDIF}\end{aligned}$$

5.4.3 Frequently Appearing Premises

For the cases involving an escape line course, s must not be at the boundary of the infinite cylinder, i.e., $s_x^2 + s_y^2 > D^2$. The calculated time of closest approach $\tau(s, v')$ must satisfy $0 < \tau(s, v') < t'$. Symmetrically, for the cases involving a recovery line course, s'' must not be at the boundary of the infinite cylinder, i.e., $s_x''^2 + s_y''^2 > D^2$, and the time of closest approach $\tau(s'', v'') + t''$ must satisfy $t' < \tau(s'', v'') + t'' < t''$. For the cases involving a circle course, the initial relative vertical speed must not equal zero, i.e., $v_z \neq 0$; otherwise, there is no solution. In other cases horizontal speeds must not equal not zero. (e.g. hor_speed_gt_0?(v')). Finally, it is necessary to relate certain variables explicitly: $v' = v_{oe} - v_i$, $v = v_o - v_i$, and $s'' = s + t''v$.

5.4.4 The Line Escape Theorem

Theorem 40 (line_escape) *If* $\alpha' = \mathsf{alpha\_calc}(-1, s)$ *or* $\alpha' = \mathsf{alpha\_calc}(1, s)$ *holds, and the quadratic equation*

$$v_x'^2(1 + \alpha'^2) + 2v_x'(v_{ix} + \alpha' v_{iy}) + v_{ix}^2 + v_{iy}^2 - v_{ox}^2 - v_{oy}^2 \quad = \quad 0. \qquad (99)$$

has solutions x_1 *and* x_2*, i.e., the discriminant is non-negative:*

$$\mathsf{discr}(1 + {\alpha'}^2, 2v_{ix} + \alpha' v_{iy}, {v_{ix}}^2 + v_i y^2 - {v_{ox}}^2 - {v_{oy}}^2) \geq 0$$

then

$$\begin{array}{rl} & {s_x}^2 + {s_y}^2 > D^2 \\ \wedge & {v_{ox}}^2 + {v_{oy}}^2 \neq {v_{ix}}^2 + {v_{iy}}^2 \\ \wedge & \textsf{hor\_speed\_gt\_0?}(v') \\ \wedge & (v'_x = x_1 \ \vee \ v'_x = x_2) \\ \wedge & v'_y = \alpha' v'_x \\ & \ \supset \ \textsf{separation?}(s, v') \end{array}$$

Proof. First we establish that $v'_x \neq 0$: If $v'_x = 0$ then (99) simplifies to ${v_{ix}}^2 + {v_i y}^2 = {v_{ox}}^2 + {v_{oy}}^2$ which contradicts the second premise. Next using separation_lem [Lemma 1] we change the goal to separation?$(s + \tau(s, v')v', v')$. The goal is further reduced by line_case_correctness [Theorem 2] to tangent_point?$(s+\tau(s, v')v', v')$. Next using tau_is_tangent_pt [Lemma 4] the goal is simplified to tangent_condition?(s, v') which expands to

$$D^2({v'_x}^2 + {v'_y}^2) = (s_x v'_y - s_y v'_x)^2$$

Rewriting the goal with the last premise we get $D^2({v'_x}^2 + (\alpha' v'_x)^2) = (s_x \alpha' v'_x - s_y v'_x)^2$. Dividing both sides by v'^2_x yields:

$$D^2(1 + \alpha'^2) = (s_x \alpha' - s_y)^2$$

which can be rearranged into a quadratic equation in α':

$$\alpha'^2(D^2 - s_x^2) + 2\alpha' s_x s_y + D^2 - s_y^2 = 0. \tag{100}$$

If $D^2 = s_x^2$, the goal simplifies to

$$2\alpha' s_x s_y + D^2 - s_y^2 = 0. \tag{101}$$

which follows trivially from the definition of α' given by alpha_calc. Otherwise, Equation (100) has solutions

$$\alpha' = \frac{-s_x s_y + \varepsilon' D \sqrt{s_x^2 + s_y^2 - D^2}}{D^2 - s_x^2}$$

where $\varepsilon' \in \{-1, 1\}$. This also matches the definition of α' given by alpha_calc.
□

Theorem 41 (line_esc_hd_only) *If the quadratic equation* (99) *has solutions* x_1 *and* x_2*, then*

$$\begin{array}{rl} & {s_x}^2 + {s_y}^2 > D^2 \ \wedge \ {v_{ox}}^2 + {v_{oy}}^2 \neq {v_{ix}}^2 + {v_{iy}}^2 \\ \wedge & v' = v'_o - v_i \ \wedge \ (v'_x = x_1 \ \vee \ v'_x = x_2) \\ \wedge & v'_y = \alpha' v'_x \ \wedge \ v'_{oz} = v_{oz} \\ & \ \supset \ \textsf{heading\_only?}(v_o, v'_o) \end{array}$$

Proof. Equation (99) can be re-arranged to

$$v_x'^{\,2} + {v_{ix}}^2 + {v_{iy}}^2 + v_x'^{\,2}\alpha'^2 + 2v_x'v_{ix} - {v_{ox}}^2 - {v_{oy}}^2 + 2v_x'v_{iy}\alpha' = 0,$$

and then to

$$(v_x' + v_{ix})^2 + (\alpha' v_x' + v_{iy})^2 = {v_{ox}}^2 + {v_{oy}}^2.$$

Replacing with the premises $\alpha' v_x' = v_y'$ and $v' = v_o' - v_i$ turns this into

$$(v_{ox}' - v_{ix} + v_{ix})^2 + (v_{oy}' - v_{iy} + v_{iy})^2 = {v_{ox}}^2 + {v_{oy}}^2,$$

which simplifies to ${v_{ox}'}^2 + {v_{oy}'}^2 = {v_{ox}}^2 + {v_{oy}}^2$ which is the expanded $\mathsf{heading\_only?}(v_o, v_o')$.

□

5.4.5 The Line Recovery Theorem

Theorem 42 ($\mathsf{line\_recovery}$)

$$\begin{array}{ll}
 & \alpha'' = \mathsf{alpha\_calc}(\varepsilon, s'') \\
\wedge & s'' = s + t''v \;\wedge\; \mathsf{hor\_speed\_gt\_0?}(v'') \\
\wedge & t'' \neq t' \;\wedge\; s_x''^2 + s_y''^2 - D^2 \geq 0 \;\wedge\; s_y'' \neq 0 \\
\wedge & v_y' - \alpha'' v_x' \neq 0 \\
\wedge & t' = t'' \dfrac{v_y - \alpha'' v_x}{v_y' - \alpha'' v_x'} \\
\wedge & v_x'' = \dfrac{t'' v_x - t' v_x'}{t'' - t'} \\
\wedge & v_y'' = \alpha'' v_x'' \;\wedge\; v_z' = v_z \;\wedge\; v_z'' = v_z \\
 & \supset\; \mathsf{separation?}(s + t'v', v'')
\end{array}$$

Proof. First we show that

$$s + t'v' = s'' - (t'' - t')v''.. \tag{102}$$

Since $s'' = s + t''v$ we need only show that

$$(t'' - t')v'' = t''v - t'v'.$$

To prove this we show (31), (32), and (33). From the premise defining v_x'', we get

$$v_x''(t'' - t') = t''v_x - t'v_x' \tag{103}$$

by cross-multiplying, which establishes (31). From the premise defining t'', we get

$$v_y't' - v_x'\alpha''t' = v_yt'' - v_x\alpha''t''$$

by cross-multiplication. Rearrangement yields

$$v_y't' + \alpha''(v_xt'' - v_x't') = v_yt'',$$

which can be rewritten using the premise defining v''_x as:

$$v'_y t' + \alpha'' v''_x (t'' - t') = v_y t''.$$

By $v''_y = \alpha'' v''_x$, this turns into

$$v''_y (t'' - t') = t'' v_y - t' v'_y$$

which establishes (32). Since $v'_z = v_z$ and $v''_z = v_z$ we also have (33).
Now that we have proven (102), we may rewrite the goal of the theorem to

$$\mathsf{separation?}(s'' - (t'' - t')v'', v'')$$

Using separation_lem [Lemma 1] we change the starting point for the goal:

$$\mathsf{separation?}(s'' - (t'' - t')v'' + (\tau(s'', v'') - t' + t'')v'', v'')$$

which can be simplified to

$$\mathsf{separation?}(s'' + \tau(s'', v'')v'', v'')$$

Using line_case_correctness [Theorem 2] this goal can be reduced to tangent_point?$(s'' + \tau(s'', v'')v'', v'')$. Next using tau_is_tangent_pt [Lemma 4] the goal is simplified to proving tangent_condition?(s'', v'') which expands to

$$D^2(v''^2_x + v''^2_y) = (s''_x v''_y - s''_y v''_x)^2 \tag{104}$$

We consider two cases:
Case 1 [$D^2 = s''^2_x$]: In this case (104) reduces to

$$D^2 v''^2_x = -2 s''_x v''_y s''_y v''_x + s''^2_y v''^2_x. \tag{105}$$

Expansion of alpha_calc yields

$$\alpha'' = \frac{s''^2_y - D^2}{2 s''_x s''_y}$$

This and the premise $v''_y = \alpha''$ solve (105).
Case 2 [$D^2 \neq s''^2_x$]: In this case expansion of alpha_calc yields

$$\alpha'' = \frac{-s''_x s''_y + \varepsilon D \sqrt{s''^2_x + s''^2_y - D^2}}{D^2 - s''^2_x}$$

which solves the quadratic equation

$$D^2(1 + \alpha''^2) = (s''_x \alpha'' - s''_y)^2.$$

Multiplying both sides by v''^2_x yields

$$(1 + \alpha''^2) D^2 v''^2_x = (s''_x \alpha'' - s''_y)^2 v''^2_x.$$

from which (104) follows by the premise $v''_y = \alpha''$.
□

5.4.6 The Circle Escape Theorem

First we show that the constructed solution is at the cylinder boundary.

Lemma 43 (cir_esc_cyl) *If the quadratic equation*

$$v^2 A + vB + C \quad = \quad 0, \tag{106}$$

defined by

$$\begin{aligned} A &= 4\theta^2((s_x - \theta v_{ix})^2 + (s_y - \theta v_{iy})^2), \\ B &= 4(s_x - \theta v_{ix})\theta E, \\ C &= E^2 - 4(s_y - \theta v_{iy})^2\theta^2(v_{ox}^2 + v_{oy}^2), \\ E &= (s_x - \theta v_{ix})^2 + (s_y - \theta v_{iy})^2 + \theta^2 v_{ox}^2 + \theta^2 v_{oy}^2 - D^2 \end{aligned}$$

is a proper quadratic equation, i.e., $A \neq 0$, and has solutions v_1 and v_2, i.e. $\mathsf{discr}(A, B, C) \geq 0$*, then*

$$\begin{aligned} & v_z \neq 0 \\ \wedge \quad & \mathsf{sign}(-2(s_y - \theta v_{iy})\theta v'_{oy}) = \mathsf{sign}(E + 2(s_x - \theta v_{ix})\theta v'_{ox}) \\ \wedge \quad & {v_{ox}}^2 + {v_{oy}}^2 \geq {v'_{ox}}^2 \\ \wedge \quad & (v'_{ox} = v_1 \ \vee \ v'_{ox} = v_2) \\ \wedge \quad & v'_{oy} = \sqrt{{v_{ox}}^2 + {v_{oy}}^2 - {v'_{ox}}^2} \\ & \supset \ \mathsf{on\_cyl?}(s + \theta v') \end{aligned}$$

Proof. Since v'_{ox} is a solution of the quadratic equation (106), we have:

$$A{v'_{ox}}^2 + Bv'_{ox} + C = 0$$

Substituting A, B, C; using the squared premise that defines v'_{oy}:

$$v'^2_{oy} \quad = \quad v^2_{ox} + v^2_{oy} - v'^2_{ox};$$

and simplifying turns this into

$$[-2(s_y - \theta v_{iy})\theta v'_{oy}]^2 =$$
$$[(s_x - \theta v_{ix})^2 + (s_y - \theta v_{iy})^2 + 2(s_x - \theta v_{ix})\theta v'_{ox} + \theta'^2 v'^2_{ox} + \theta'^2 v'^2_{oy} - D^2]^2.$$

By premise,

$$\mathsf{sign}(-2(s_y - \theta v_{iy})\theta v'_{oy}) = \mathsf{sign}(E + 2(s_x - \theta v_{ix})\theta v'_{ox}),$$

so we can take the square-root of both sides and further simplify to obtain:

$$(s_x + \theta(v'_{ox} - v_{ix}))^2 + (s_y + \theta(v'_{oy} - v_{iy}))^2 \quad = D^2,$$

which is the expansion of $\mathsf{on\_cyl?}(s + \theta v')$.
□

Theorem 44 (circle_escape) *If the quadratic equation*

$$v^2A + vB + C \quad = \quad 0,$$

defined by

$$\begin{aligned}
A &= 4\theta^2((s_x - \theta v_{ix})^2 + (s_y - \theta v_{iy})^2), \\
B &= 4(s_x - \theta v_{ix})\theta E, \\
C &= E^2 - 4(s_y - \theta v_{iy})^2\theta^2(v_{ox}^2 + v_{oy}^2), \\
E &= (s_x - \theta v_{ix})^2 + (s_y - \theta v_{iy})^2 + \theta^2 v_{ox}^2 + \theta^2 v_{oy}^2 - D^2
\end{aligned}$$

is a proper quadratic equation, i.e., $A \neq 0$*, and has solutions* v_1 *and* v_2*, i.e.* $\mathsf{discr}(A, B, C) \geq 0$*, and*

$$\begin{aligned}
& v_z \neq 0 \ \wedge \ \theta = \theta^{\varepsilon}(s_z, v_z) \\
\wedge \quad & ((\mathsf{exit?}(s + \theta v', v') \ \wedge \ \varepsilon = -1) \ \vee \\
& \ (\mathsf{entry?}(s + \theta v', v') \ \wedge \ \varepsilon = 1)) \\
\wedge \quad & {v_{ox}}^2 + {v_{oy}}^2 \geq {v'_{ox}}^2 \\
\wedge \quad & (v'_{ox} = v_1 \ \vee \ v'_{ox} = v_2) \\
\wedge \quad & v'_{oy} = \sqrt{{v_{ox}}^2 + {v_{oy}}^2 - {v'_{ox}}^2} \\
\wedge \quad & v = v_o - v_i \ \wedge \ v' = v'_o - v_i \ \wedge \ v'_z = v_z \\
\wedge \quad & \mathsf{sign}(-2(s_y - \theta v_{iy})\theta v'_{oy}) = \mathsf{sign}(E + 2(s_x - \theta v_{ix})\theta v'_{ox})
\end{aligned}$$

then we have

$$\mathsf{separation?}(s + \theta v', v'')$$

Proof. Using circle_correctness [Lemma 27] the goal can be reduced to:

$$\mathsf{exit\_point?}(s + \theta^-(s_z, v'_z)v', v') \ \vee \ \mathsf{entry\_point?}(s + \theta^+(s_z, v'_z)v', v')$$

By expanding the definitions of entry_point? and exit_point? and by simplifying, the goal becomes:

$$\begin{aligned}
& (\mathsf{on\_cyl?}(s + \theta^{\varepsilon}(s_z, v'_z)v') \ \wedge \\
& ((\mathsf{exit?}(s + \theta^{\varepsilon}(s_z, v'_z)v', v') \ \wedge \ \varepsilon = -1) \ \vee \\
& \quad (\mathsf{entry?}(s + \theta^{\varepsilon}(s_z, v'_z)v', v') \ \wedge \ \varepsilon = 1))
\end{aligned}$$

Using the entry? and exit? premises we end up with the following subgoal:

$$\mathsf{on\_cyl?}(s + \theta^{\varepsilon}(s_z, v'_z)v')$$

This is discharged with cir_esc_cyl [Lemma 43] and by the premise $v'_z = v_z$.
□

Theorem 45 (circle_hd_only)

$$\begin{array}{rl} & {v_{ox}}^2 + {v_{oy}}^2 \geq {v'_{ox}}^2 \\ \wedge & v'_{oy} = \sqrt{{v_{ox}}^2 + {v_{oy}}^2 - {v'_{ox}}^2} \;\wedge\; v'_{oz} = v_{oz} \\ \wedge & v = v_o - v_i \;\wedge\; v' = v'_o - v_i \wedge v'_z = v_z \\ & \quad \supset \; \mathsf{heading\_only?}(v_o, v'_o) \end{array}$$

Proof. The result follows trivially by squaring both sides of the second premise and expanding the definition of heading_only?.
□

5.4.7 The Circle Recovery Theorem

First we prove that the constructed solution is on the boundary of the infinite cylinder.

Lemma 46 (cir_rec_lem) *If the quadratic equation $at^2 + bt + c = 0$, defined by*

$$\begin{array}{ll} \mathsf{A}_x = s''_x + (\theta'' - t'')v'_x, & \mathsf{A}_y = s''_y + (\theta'' - t'')v'_y, \\ \mathsf{B}_x = s_x + \theta'' v_x, & \mathsf{B}_y = s_y + \theta'' v_y, \end{array}$$

$$\begin{aligned} a &= {\mathsf{A}_x}^2 + {\mathsf{A}_y}^2 - D^2, \\ b &= 2t''(D^2 - \mathsf{A}_x\mathsf{B}_x - \mathsf{A}_y\mathsf{B}_y), \\ c &= t''^2({\mathsf{B}_x}^2 + {\mathsf{B}_y}^2 - D^2), \end{aligned}$$

is a proper quadratic equation, i.e., $a \neq 0$, and has solutions t_1 and/or t_2, i.e. $\mathsf{discr}(a, b, c) \geq 0$,

$$\begin{array}{rl} & (t' = t_1 \;\vee\; t' = t_2) \\ \wedge & s'' = s + t''v \;\wedge\; t' \neq t'' \\ \wedge & v''_x = \dfrac{t''v_x - t'v'_x}{t'' - t'} \\ \wedge & v''_y = \dfrac{t''v_y - t'v'_y}{t'' - t'} \\ & \quad \supset \; \mathsf{on\_cyl?}(s + t'v' + (\theta'' - t')v'') \end{array}$$

Proof. Expanding the coefficients of the quadratic equation yields

$$\begin{aligned} & t'^2[(s''_x + (\theta'' - t'')v'_x)^2 + (s''_y + (\theta'' - t'')v'_y)^2 - D^2] + \\ & \quad 2t't''[D^2 - (s''_x + (\theta'' - t'')v'_x)(s_x + \theta''v_x) - (s''_y + (\theta'' - t'')v'_y)(s_y + \theta''v_y)] + \\ & \quad t''^2((s_x + \theta''v_x)^2 + (s_y + \theta''v_y)^2 - D^2) = 0. \end{aligned}$$

Rewriting with $s'' = s + t''v$, and rearranging yields:

$$\begin{aligned} & [s_x(t'' - t') - v_xt't'' - v'_xt'\theta'' + v_x\theta''t'' + v'_x(t'' - t')t' + v'_xt't']^2 \\ & + [s_y(t'' - t') - v_yt't'' - v'_yt'\theta'' + v_y\theta''t'' + v'_y(t'' - t')t' + v'_yt't')]^2 \\ & \qquad = D^2(t'' - t')^2. \end{aligned}$$

From the premises defining v''_x and v''_y, we get by cross-multiplication:

$$\begin{aligned} v''_x(t''-t') &= t''v_x - t'v'_x, \\ v''_y(t''-t') &= t''v_y - t'v'_y. \end{aligned}$$

rewriting with these and re-arranging yields

$$\begin{aligned} &[(s_x + t'v'_x + (\theta'' - t')v''_x)(t''-t')]^2 \\ &+ [(s_y + t'v'_y + (\theta'' - t')v''_y)(t''-t')]^2 = D^2(t''-t')^2. \end{aligned}$$

Factoring out $(t''-t')^2$ and dividing both sides by $(t''-t')^2$ yields the on_cyl? claim when it is expanded.
□

Theorem 47 (circle_recovery) *If the quadratic equation $at^2+bt+c=0$, defined by:*

$$\begin{aligned} \mathsf{A}_x &= s''_x + (\theta''-t'')v'_x & \mathsf{A}_y &= s''_y + (\theta''-t'')v'_y \\ \mathsf{B}_x &= s_x + \theta''v_x & \mathsf{B}_y &= (s_y + \theta''v_y) \end{aligned}$$

$$\begin{aligned} a &= \mathsf{A}_x{}^2 + \mathsf{A}_y{}^2 - D^2 \\ b &= 2t''(D^2 - \mathsf{A}_x\mathsf{B}_x - \mathsf{A}_y\mathsf{B}_y) \\ c &= t''^2(\mathsf{B}_x{}^2 + \mathsf{B}_y{}^2 - D^2) \end{aligned}$$

is a proper quadratic equation, i.e., $a \neq 0$, and has solutions t_1 and t_2, i.e. $\mathsf{discr}(a,b,c) \geq 0$*, and the following hold*

$$\begin{aligned} &\mathsf{hor\_speed\_gt\_0?}(v') \ \wedge\ v' = v'_o - v_i \ \wedge\ s'' = s + t''v \\ \wedge\quad & v_z \neq 0 \ \wedge\ \theta'' = \theta^+(s_z, v_z) \\ \wedge\quad & \theta'' < t'' \\ \wedge\quad & (t' = t_1 \ \vee\ t' = t_2) \\ \wedge\quad & t' \neq t'' \ \wedge\ v''_x = \frac{t''v_x - t'v'_x}{t''-t'} \\ \wedge\quad & v''_y = \frac{t''v_y - t'v'_y}{t''-t'} \ \wedge\ v'_z = v_z \ \wedge\ v''_z = v_z \\ \wedge\quad & [s''_x + (\theta''-t'')v''_x]v''_x + [s''_y + (\theta''-t'')v''_y]v''_y \leq 0 \end{aligned}$$

then we have

$$\mathsf{separation?}(s + t'v', v'')$$

Proof. From the premises defining v''_x and v''_y, and the premises $v''_z = v_z = v'_z$, we get:

$$\begin{aligned} v''_x(t''-t') &= t''v_x - t'v'_x, & (107) \\ v''_y(t''-t') &= t''v_y - t'v'_y, & (108) \\ v''_y(t''-t') &= t''v_y - t'v'_y; \end{aligned}$$

Hence we have

$$v''(t'' - t') = t''v - t'v'.$$

Using $t''v = s'' - s$, and rearranging, we get

$$s + t'v' = s'' - (t'' - t')v''.$$

Thus the goal is reduced to $\mathsf{separation?}(s'' - (t'' - t')v'', v'')$. We use separation_lem [Lemma 1] to change the starting point for the goal to

$$\mathsf{separation?}(s'' - (t'' - t')v'' + (\theta'' - t')v'', v''),$$

which can be simplified to

$$\mathsf{separation?}(s'' + (\theta'' - t'')v'', v'').$$

Using circle_case_correctness [Theorem 3] this goal can be reduced to:

$$|s''_z + (\theta'' - t'')v''_z| \geq H, \tag{109}$$

$$\mathsf{entry\_point?}(s'' + (\theta'' - t'')v'', v''), \tag{110}$$

$$(s'' + (\theta'' - t'')v''_z)v''_z \geq 0. \tag{111}$$

Claim (109) follows directly from the definition of θ^+ (30). For (110), we expand the definition of entry_point. This produces the subgoals

$$\mathsf{on\_cyl?}(s'' + (\theta'' - t'')v'') \tag{112}$$

$$[s'' + (\theta'' - t'')v''_x]v''_x + [s'' + (\theta'' - t'')v''_y]v''_y \leq 0 \tag{113}$$

$$(s''_z + (\theta'' - t'')v''_z)v''_z \geq 0 \tag{114}$$

In (112), we expand on_cyl?. This reduces the goal to

$$[s''_x + (\theta'' - t'')v''_x]^2 + [s''_y + (\theta'' - t'')v''_y]^2 = D^2 \tag{115}$$

From cir_rec_lem [Lemma 46] we have:

$$[s_x + t'v'_x + (\theta'' - t')v''_x]^2 + [s_y + t'v'_y + (\theta'' - t')v''_y]^2 = D^2 \tag{116}$$

which is the horizontal distance to the origin at time θ''. Rearranging and substituting formulas (107) and (108) into (116), we obtain:

$$[s_x + v''_x(\theta'' - t'') + t''v_x]^2 + [s_y + v''_y(\theta'' - t'') + t''v_y]^2 = D^2$$

By $s'' = s + t''v$, we get (112).

The subgoal (113) is exactly the last premise.

This leaves to prove (114). Lemma vertical_entry_exit_condition [Lemma 11] gives us:

$$s_z v_z + \theta'' v_z v_z \geq 0$$

Using $s_z = s''_z - t''v_z$ and $v_z = v''_z$, we get

$$(s''_z - t''v''_z)v''_z + \theta''v''_z v''_z \geq 0$$

Factoring this formula yields (114) as wanted. This also proves (111).
□

5.4.8 The In-Circle Recovery Theorem

Theorem 48 (in_circle_recovery)

$$\begin{aligned} & v_z \neq 0 \ \wedge \ t' = \theta^+(s_z, v_z) \ \wedge \\ & v'_z = v_z \ \wedge \ v''_z = v_z \\ & \qquad \supset \ \textsf{separation\_pos?}(s + t'v', v'') \end{aligned}$$

where separation_pos? *is defined by*

$$\textsf{separation\_pos?}(s, v) = \forall\, T \geq 0 : \textsf{hor\_sep?}(s + Tv) \vee \textsf{vert\_sep?}(s + Tv)$$

Proof. Since $t' = \theta^+(s_z, v_z)$, we seek to establish

$$\textsf{vert\_sep?}(s + \theta^+(s_z, v_z)v' + Tv'')$$

for all $T \geq 0$. By definition of vert_sep? this is

$$|s_z + \theta^+(s_z, v_z)v'_z + Tv''_z| \geq H \tag{117}$$

Case A $[v_z \geq 0]$: From definition of θ^+ (30) we get:

$$s_z + \theta^+(s_z, v_z)v_z = H$$

and since $v'_z = v_z$, the goal (117) becomes:

$$|H + Tv''_z| \geq H$$

By $v''_z = v_z$, we have $Tv''_z \geq 0$ and the result trivially follows.
Case B $[v_z < 0]$: From definition of θ^+ (30) we get:

$$s_z + \theta^+(s_z, v_z)v_z = -H$$

and since $v'_z = v_z$, the goal (117) becomes:

$$|-H + Tv''_z| \geq H$$

Now, since $v''_z = v_z$, we have $Tv''_z < 0$ and the result follows.
□

5.4.9 The Out-Circle Recovery Theorem

Theorem 49 (out_circle_recovery) *If the quadratic equation*

$$v^2A + vB + C \;=\; 0,$$

defined by

$$\begin{aligned} A &= 4\theta'^2((s_x - \theta' v_{ix})^2 + (s_y - \theta' v_{iy})^2), \\ B &= 4(s_x - \theta' v_{ix})\theta' E, \\ C &= E^2 - 4(s_y - \theta' v_{iy})^2\theta'^2(v_{ox}^2 + v_{oy}^2), \\ E &= (s_x - \theta' v_{ix})^2 + (s_y - \theta' v_{iy})^2 + \theta'^2 v_{ox}^2 + \theta'^2 v_{oy}^2 - D^2 \end{aligned}$$

is a proper quadratic equation, i.e., $A \neq 0$*, and has solutions* v_1 *and* v_2*, i.e.* $\mathsf{discr}(A, B, C) \geq 0$*, and*

$$\begin{array}{ll} & v_z \neq 0 \ \wedge \ \theta' = t' = \theta^-(s_z, v_z) \\ \wedge & s'' = s + t''v \\ \wedge & {v_{ox}}^2 + {v_{oy}}^2 \geq {v'_{ox}}^2 \\ \wedge & (v'_{ox} = v_1 \ \vee \ v'_{ox} = v_2) \\ \wedge & v'_{oy} = \sqrt{{v_{ox}}^2 + {v_{oy}}^2 - {v'_{ox}}^2} \\ \wedge & v' = v'_o - v_i \ \wedge \ v'_z = v_z \ \wedge \ v''_z = v_z \\ \wedge & \mathsf{sign}(-2(s_y - \theta' v_{iy})\theta' v'_{oy}) = \mathsf{sign}(E + 2(s_x - \theta' v_{ix})\theta' v'_{ox}) \\ \wedge & \mathsf{exit?}(s + \theta' v', v'') \\ & \quad \supset \ \mathsf{separation?}(s + t'v', v'') \end{array}$$

Proof. Using the lemma circle_case_correctness [Theorem 3], the goal is reduced to

$$|s_z + t'v'_z| \geq H, \tag{118}$$

$$\mathsf{exit\_point?}(s + t'v', v''), \tag{119}$$

$$(s_z + t'v'_z)v''_z \leq 0 \tag{120}$$

Claim (118) is easily discharged using reaching_H_theta [Lemma 10]. By the exit? premise, (119) reduces to on_cyl?$(s + t'v')$ which is proven by cir_esc_cyl [Lemma 43]. This leaves us with

$$(s_z + t'v'_z)v''_z \leq 0.$$

Substituting with $t' = \theta', v'_z = v_z$ and $v''_z = v_z$ yields:

$$(s_z + \theta' v_z)v_z \leq 0 \tag{121}$$

From the definition of $\theta^-(s_z, v_z)$ described in (29), we obtain

$$\theta' v_z = -s_z - \mathsf{sign}(v_z)H$$

and so the goal simplifies to

$$-\mathsf{sign}(v_z)Hv_z \leq 0. \tag{122}$$

Case analysis whether or not $v_z > 0$ solves this goal.
□

5.4.10 Timeliness Properties

Lemma 50 (timeliness)

$$\begin{array}{l} t' \neq t'' \ \wedge \ v''_x = \dfrac{t''v_x - t'v'_x}{t'' - t'} \ \wedge \\ v''_y = \dfrac{t''v_y - t'v'_y}{t'' - t'} \ \wedge \ v'_z = v_z \ \wedge \ v''_z = v_z \\ \qquad \supset \ s + t''v = s + t'v' + (t'' - t')v'' \end{array}$$

Proof. The result follows trivially by cross-multiplying premises 2 and 3.
□

Theorem 51 (alpha_timeliness)

$$\begin{aligned} & t' \neq t'' \ \wedge \ v''_x = \frac{t'' v_x - t' v'_x}{t'' - t'} \\ \wedge \quad & v''_y = \alpha v''_x \ \wedge \ v'_y - \alpha v'_x \neq 0 \\ \wedge \quad & t' = t'' \frac{v_y - \alpha v_x}{v'_y - \alpha v'_x} \\ \wedge \quad & v'_z = v_z \ \wedge \ v''_z = v_z \\ & \quad \supset \ s + t'' v = s + t' v' + (t'' - t') v'' \end{aligned}$$

Proof. Using timeliness [Theorem 50] the goal is reduced to

$$v''_y = \frac{t'' v_y - t' v'_y}{t'' - t'}$$

Cross-multiplication, replacement by $v''_y = \alpha v''_x$, and rearrangement reduces this to

$$\alpha v''_x (t'' - t') = t'' v_y - t' v'_y. \tag{123}$$

Multiplying both sides of the defining premise of v''_x by $(t'' - t')\alpha$ and re-arranging yields

$$v_x \alpha t'' = v''_x \alpha t'' - v''_x \alpha t' + v'_x \alpha t' \tag{124}$$

Cross-multiplying the defining premise of t' yields

$$v'_y t' - v'_x \alpha t' = v_y t'' - v_x \alpha t''$$

Replacing (124) in this equation and simplifying yields (123) as wanted.
□

Theorem 52 (vor_timeliness)

$$\begin{aligned} & t' \neq t'' \ \wedge \ v''_{ox} = \frac{t' v'_{ox} - t'' v_{ox}}{t' - t''} \\ \wedge \quad & v''_{oy} = \frac{t' v'_{oy} - t'' v_{oy}}{t' - t''} \ \wedge \ v'_z = v_z \ \wedge \ v''_z = v_z \\ \wedge \quad & v = v_o - v_i \ \wedge \ v' = v'_o - v_i \ \wedge \ v'' = v''_o - v_i \\ & \quad \supset \ s + t'' v = s + t' v' + (t'' - t') v'' \end{aligned}$$

Proof. Using lemma timeliness the goal is reduced to

$$v''_x = \frac{t' v'_x - t'' v_x}{t' - t''}$$
$$v''_y = \frac{t' v'_y - t'' v_y}{t' - t''}$$

Modulo the definitions of v, v', and v'', the cross-multiplied versions of these equations match the cross-multiplied versions of premises 2 and 3.
□

5.4.11 Line/line

The line/line situation is shown in Figure 4.

Theorem 53 (llhd) *If* $\alpha' = \mathsf{alpha\_calc}(-1, s)$ *or* $\alpha' = \mathsf{alpha\_calc}(1, s)$*, and the quadratic equation*

$$v_x'^2(1+\alpha'^2) + 2v_x'(v_{ix} + \alpha' v_{iy}) + v_{ix}^2 + v_{iy}^2 - v_{ox}^2 - v_{oy}^2 \quad = \quad 0$$

has solutions x_1 *and* x_2*, i.e., the discriminant is non-negative:*

$$\mathsf{discr}(1+\alpha'^2, 2v_{ix} + \alpha' v_{iy}, {v_{ix}}^2 + v_i y^2 - {v_{ox}}^2 - {v_{oy}}^2) \geq 0$$

then

$$\begin{array}{ll}
 & v' = v_o' - v_i \ \wedge \ v = v_o - v_i \ \wedge \ s'' = s + t''v \\
\wedge & {s_x}^2 + {s_y}^2 > D^2 \\
\wedge & {v_{ox}}^2 + {v_{oy}}^2 \neq {v_{ix}}^2 + {v_{iy}}^2 \\
\wedge & \mathsf{hor\_speed\_gt\_0?}(v') \ \wedge \ \mathsf{hor\_speed\_gt\_0?}(v'') \\
\wedge & (v_x' = x_1 \ \vee \ v_x' = x_2) \\
\wedge & v_y' = \alpha' v_x' \ \wedge \ t'' \neq t' \\
\wedge & s_x''^2 + s_y''^2 - D^2 \geq 0 \ \wedge \ s_y'' \neq 0 \\
\wedge & \alpha'' = \mathsf{alpha\_calc}(\varepsilon, s'') \ \wedge \ v_y' - \alpha'' v_x' \neq 0 \\
\wedge & t' = t'' \dfrac{v_y - \alpha'' v_x}{v_y' - \alpha'' v_x'} \\
\wedge & v_x'' = \dfrac{t'' v_x - t' v_x'}{t'' - t'} \\
\wedge & v_y'' = \alpha'' v_x'' \ \wedge \ v_z' = v_z \ \wedge \ v_z'' = v_z \\
 & \supset \\
 & \mathsf{separation?}(s, v') \ \wedge \\
 & \mathsf{separation?}(s + t'v', v'') \ \wedge \\
 & \mathsf{heading\_only?}(v_o, v_o') \ \wedge \\
 & s + t''v = s + t'v' + (t'' - t')v''
\end{array}$$

Proof. The theorem follows directly from the four theorems line_escape [Theorem 40], line_recovery [Theorem 42], line_esc_hd_only [Theorem 41], and alpha_timeliness [Theorem 51].
□

5.4.12 Line/circle

The line/circle situation is depicted in Figure 5.

Theorem 54 (lchd) *If* $\alpha' = \mathsf{alpha\_calc}(-1, s)$ *or* $\alpha' = \mathsf{alpha\_calc}(1, s)$*, and the quadratic equation*

$$v_x'^2(1+\alpha'^2) + 2v_x'(v_{ix} + \alpha' v_{iy}) + v_{ix}^2 + v_{iy}^2 - v_{ox}^2 - v_{oy}^2 \quad = \quad 0$$

has solutions x_1 *and* x_2*, i.e., the discriminant is non-negative:*

$$\mathsf{discr}(1+\alpha'^2, 2v_{ix} + \alpha' v_{iy}, {v_{ix}}^2 + v_i y^2 - {v_{ox}}^2 - {v_{oy}}^2) \geq 0$$

and the quadratic equation $at^2 + bt + c = 0$*, defined by*

$$\begin{aligned} \mathsf{A}_x &= s_x'' + (\theta'' - t'')v_x' & \qquad \mathsf{A}_y &= s_y'' + (\theta'' - t'')v_y' \\ \mathsf{B}_x &= s_x + \theta'' v_x & \qquad \mathsf{B}_y &= s_y + \theta'' v_y \end{aligned}$$

$$\begin{aligned} a &= {\mathsf{A}_x}^2 + {\mathsf{A}_y}^2 - D^2 \\ b &= 2t''(D^2 - \mathsf{A}_x\mathsf{B}_x - \mathsf{A}_y\mathsf{B}_y) \\ c &= t''^2({\mathsf{B}_x}^2 + {\mathsf{B}_y}^2 - D^2) \end{aligned}$$

is a proper quadratic equation, i.e., $a \neq 0$*, and has solutions* t_1 *and* t_2*, i.e.,* $\mathsf{discr}(a, b, c) \geq 0$*, then*

$$\begin{aligned} & v' = v_o' - v_i \ \wedge \ v = v_o - v_i \ \wedge \ s'' = s + t''v \\ \wedge \quad & {s_x}^2 + {s_y}^2 > D^2 \ \wedge \ {v_{ox}}^2 + {v_{oy}}^2 \neq {v_{ix}}^2 + {v_{iy}}^2 \\ \wedge \quad & \mathsf{hor\_speed\_gt\_0?}(v') \\ \wedge \quad & (v_x' = x_1 \ \vee \ v_x' = x_2) \\ \wedge \quad & v_y' = \alpha v_x' \ \wedge \ v_z' = v_z \ \wedge \ v_z \neq 0 \\ \wedge \quad & \theta'' = \theta^+(s_z, v_z) \wedge \theta'' < t'' \\ \wedge \quad & (t' = t_1 \ \vee \ t' = t_2) \\ \wedge \quad & t' \neq t'' \ \wedge \ v_x'' = \frac{t''v_x - t'v_x'}{t'' - t'} \\ \wedge \quad & v_y'' = \frac{t''v_y - t'v_y'}{t'' - t'} \ \wedge \ v_z' = v_z \ \wedge \ v_z'' = v_z \\ \wedge \quad & [s_x'' + (\theta'' - t'')v_x'']v_x'' + [s_y'' + (\theta'' - t'')v_y'']v_y'' \leq 0 \\ & \supset \\ & \mathsf{separation?}(s, v') \ \wedge \\ & \mathsf{separation?}(s + t'v', v'') \ \wedge \\ & \mathsf{heading\_only?}(v_o, v_o') \ \wedge \\ & s + t''v = s + t'v' + (t'' - t')v'' \end{aligned}$$

Proof. The theorem follows directly from line_escape [Theorem 40], circle_recovery [Theorem 47], line_esc_hd_only [Theorem 41], and timeliness [Theorem 50].
□

5.4.13 Circle/line

A circle/line maneuver is shown in Figure 6.

Theorem 55 (clhd) *If the quadratic equation*

$$v^2A + vB + C = 0,$$

defined by

$$\begin{aligned}
A &= 4\theta'^2((s_x - \theta' v_{ix})^2 + (s_y - \theta' v_{iy})^2),\\
B &= 4(s_x - \theta' v_{ix})\theta' E,\\
C &= E^2 - 4(s_y - \theta' v_{iy})^2\theta'^2(v_{ox}^2 + v_{oy}^2),\\
E &= (s_x - \theta' v_{ix})^2 + (s_y - \theta' v_{iy})^2 + \theta'^2 v_{ox}^2 + \theta'^2 v_{oy}^2 - D^2
\end{aligned}$$

is a proper quadratic equation, i.e., $A \neq 0$*, and has solutions* v_1 *and* v_2*, i.e.* $\mathsf{discr}(A, B, C) \geq 0$*, then*

$$\begin{aligned}
& v = v_o - v_i \;\wedge\; v' = v'_o - v_i \;\wedge\; s'' = s + t''v \;\wedge\; v_z \neq 0\\
\wedge\;\; & \theta' = \theta^-(s_z, v_z) \;\wedge\; \mathsf{exit?}(s + \theta' v', v')\\
\wedge\;\; & {v_{ox}}^2 + {v_{oy}}^2 \geq {v'_{ox}}^2\\
\wedge\;\; & v'_{ox} = v_1 \;\vee\; v'_{ox} = v_2\\
\wedge\;\; & v'_{oy} = \sqrt{{v_{ox}}^2 + {v_{oy}}^2 - {v'_{ox}}^2} \;\wedge\; v'_z = v_z\\
\wedge\;\; & \mathsf{sign}(-2(s_y - \theta' v_{iy})\theta' v'_{oy}) = \mathsf{sign}(E + 2(s_x - \theta' v_{ix})\theta' v'_{ox})\\
\wedge\;\; & \mathsf{hor\_speed\_gt\_0?}(v'') \;\wedge\; t'' \neq t'\\
\wedge\;\; & s''^2_x + s''^2_y - D^2 \geq 0 \;\wedge\; s''_y \neq 0\\
\wedge\;\; & \alpha'' = \mathsf{alpha\_calc}(\varepsilon, s'') \;\wedge\; v'_y - \alpha'' v'_x \neq 0\\
\wedge\;\; & t' = t'' \frac{v_y - \alpha'' v_x}{v'_y - \alpha'' v'_x}\\
\wedge\;\; & v''_x = \frac{t'' v_x - t' v'_x}{t'' - t'}\\
\wedge\;\; & v''_y = \alpha'' v''_x \;\wedge\; v'_z = v_z \;\wedge\; v''_z = v_z\\
& \supset\\
& \mathsf{separation?}(s, v') \;\wedge\\
& \mathsf{separation?}(s + t'v', v'') \;\wedge\\
& \mathsf{heading\_only?}(v_o, v'_o) \;\wedge\\
& s + t''v = s + t'v' + (t'' - t')v''
\end{aligned}$$

Proof. The theorem follows directly from circle_escape [Theorem 44], line_recovery [Theorem 42], circle_hd_only [Theorem 45], and alpha_timeliness [Theorem 51].
□

5.4.14 Circle/circle

Circle-circle solutions are shown in Figure 8.

Theorem 56 (cchd) *If the quadratic equation*

$$v^2A + vB + C \quad = \quad 0,$$

defined by

$$\begin{aligned}
A &= 4{\theta'}^2((s_x - \theta' v_{ix})^2 + (s_y - \theta' v_{iy})^2),\\
B &= 4(s_x - \theta' v_{ix})\theta' E,\\
C &= E^2 - 4(s_y - \theta' v_{iy})^2{\theta'}^2(v_{ox}^2 + v_{oy}^2),\\
E &= (s_x - \theta' v_{ix})^2 + (s_y - \theta' v_{iy})^2 + {\theta'}^2 v_{ox}^2 + {\theta'}^2 v_{oy}^2 - D^2
\end{aligned}$$

is a proper quadratic equation, i.e., $A \neq 0$, and has solutions v_1 and v_2, i.e. $\mathsf{discr}(A, B, C) \geq 0$*, and the quadratic equation $at^2 + bt + c = 0$, defined by*

$$\begin{aligned}
\mathsf{A}_x &= s''_x + (\theta'' - t'')v'_x \qquad & \mathsf{A}_y &= s''_y + (\theta'' - t'')v'_y\\
\mathsf{B}_x &= (s_x + \theta'' v_x) & \mathsf{B}_y &= (s_y + \theta'' v_y)
\end{aligned}$$

$$\begin{aligned}
a &= {\mathsf{A}_x}^2 + {\mathsf{A}_y}^2 - D^2\\
b &= 2t''(D^2 - \mathsf{A}_x\mathsf{B}_x - \mathsf{A}_y\mathsf{B}_y)\\
c &= {t''}^2({\mathsf{B}_x}^2 + {\mathsf{B}_y}^2 - D^2)
\end{aligned}$$

is a proper quadratic equation, i.e., $a \neq 0$, and has solutions t_1 and t_2, i.e. $\mathsf{discr}(a, b, c) \geq$

0, *then*

$$\begin{aligned}
& v = v_o - v_i \ \wedge \ v' = v'_o - v_i \ \wedge \ s'' = s + t''v \ \wedge \ v_z \neq 0 \\
\wedge \quad & \theta' = \theta^-(s_z, v_z) \ \wedge \ \mathsf{exit?}(s + \theta' v', v') \\
\wedge \quad & {v_{ox}}^2 + {v_{oy}}^2 \geq {v'_{ox}}^2 \ \wedge \ (v'_{ox} = v_1 \ \vee \ v'_{ox} = v_2) \\
\wedge \quad & v'_{oy} = \sqrt{{v_{ox}}^2 + {v_{oy}}^2 - {v'_{ox}}^2} \ \wedge \ v'_z = v_z \\
\wedge \quad & \mathsf{sign}(-2(s_y - \theta' v_{iy})\theta' v'_{oy}) = \mathsf{sign}(E + 2(s_x - \theta' v_{ix})\theta' v'_{ox}) \\
\wedge \quad & \mathsf{hor\_speed\_gt\_0?}(v') \ \wedge \ v_z \neq 0 \\
\wedge \quad & \theta'' = \theta^+(s_z, v_z) \ \wedge \ \theta'' < t'' \\
\wedge \quad & (t' = t_1 \ \vee \ t' = t_2) \\
\wedge \quad & t' \neq t'' \ \wedge \ v''_x = \frac{t''v_x - t'v'_x}{t'' - t'} \\
\wedge \quad & v''_y = \frac{t''v_y - t'v'_y}{t'' - t'} \ \wedge \ v'_z = v_z \ \wedge \ v''_z = v_z \\
\wedge \quad & [s''_x + (\theta'' - t'')v''_x]v''_x + [s''_y + (\theta'' - t'')v''_y]v''_y \leq 0 \\
& \supset \\
& \mathsf{separation?}(s, v') \ \wedge \\
& \mathsf{heading\_only?}(v_o, v'_o) \ \wedge \\
& \mathsf{separation?}(s + t'v', v'') \ \wedge \\
& s + t''v = s + t'v' + (t'' - t')v''
\end{aligned}$$

Proof. The theorem follows directly from circle_escape [Theorem 44], circle_recovery [Theorem 47], circle_hd_only [Theorem 45], and timeliness [Theorem 50].
□

5.4.15 In-circle

In-circle solutions are shown in Figure 9.

Theorem 57 (ichd) *If the quadratic equation*

$$v^2 A + vB + C \quad = \quad 0$$

defined by

$$\begin{aligned}
A &= 4\theta'^2((s_x - \theta' v_{ix})^2 + (s_y - \theta' v_{iy})^2), \\
B &= 4(s_x - \theta' v_{ix})\theta' E, \\
C &= E^2 - 4(s_y - \theta' v_{iy})^2 \theta'^2 (v_{ox}^2 + v_{oy}^2), \\
E &= (s_x - \theta' v_{ix})^2 + (s_y - \theta' v_{iy})^2 + \theta'^2 v_{ox}^2 + \theta'^2 v_{oy}^2 - D^2
\end{aligned}$$

is a proper quadratic equation, i.e., $a \neq 0$, and has solutions v_1 and v_2, i.e. $\mathsf{discr}(A, B, C) \geq$

0, *and*

$$\begin{array}{ll}
 & v = v_o - v_i \ \wedge \ v' = v'_o - v_i \ \wedge \ v'' = v''_o - v_i \\
\wedge & v_z \neq 0 \ \wedge \ \theta'' = \theta^{+}(s_z, v_z) \\
\wedge & \mathsf{entry?}(s + \theta'' v', v') \\
\wedge & {v_{ox}}^2 + {v_{oy}}^2 \geq {v'_{ox}}^2 \\
\wedge & (v'_{ox} = v_1 \ \vee \ v'_{ox} = v_2) \\
\wedge & v'_{oy} = \sqrt{{v_{ox}}^2 + {v_{oy}}^2 - {v'_{ox}}^2} \\
\wedge & \mathsf{sign}(-2(s_y - \theta'' v_{iy})\theta'' v'_{oy}) = \mathsf{sign}(E + 2(s_x - \theta'' v_{ix})\theta'' v'_{ox}) \\
\wedge & t' = \theta'' \wedge \theta'' < t'' \\
\wedge & v''_{ox} = \dfrac{t' v'_{ox} - t'' v_{ox}}{t' - t''} \\
\wedge & v''_{oy} = \dfrac{t' v'_{oy} - t'' v_{oy}}{t' - t''} \ \wedge \ v'_z = v_z \ \wedge \ v''_z = v_z \\
 & \supset \\
 & \mathsf{separation?}(s, v') \ \wedge \\
 & \mathsf{heading\_only?}(v_o, v'_o) \ \wedge \\
 & \mathsf{separation\_pos?}(s + t'v', v'') \ \wedge \\
 & s + t'' v = s + t' v' + (t'' - t') v''
\end{array}$$

Proof. The theorem follows directly from circle_escape [Theorem 44], in_circle_recovery [Theorem 48], circle_hd_only [Theorem 45], and vor_timeliness [Theorem 52]. Instead of separation_pos?, separation?$(s + t'v', v'')$ would be a more general result, but it is not required. All that is required is separation over the interval $[t', t'']$. This interval is covered with the separation_pos? predicate.
□

5.4.16 Out-circle

Out-circle solutions are shown in (Figure 10).

Theorem 58 (ochd) *If the quadratic equation*

$$v^2 A + vB + C \quad = \quad 0$$

defined by

$$\begin{array}{rcl}
A & = & 4\theta'^2((s_x - \theta' v_{ix})^2 + (s_y - \theta' v_{iy})^2), \\
B & = & 4(s_x - \theta' v_{ix})\theta' E, \\
C & = & E^2 - 4(s_y - \theta' v_{iy})^2 \theta'^2 (v_{ox}^2 + v_{oy}^2), \\
E & = & (s_x - \theta' v_{ix})^2 + (s_y - \theta' v_{iy})^2 + \theta'^2 v_{ox}^2 + \theta'^2 v_{oy}^2 - D^2
\end{array}$$

is a proper quadratic equation, i.e., $A \neq 0$, and has solutions v_1 and v_2, i.e. $\mathsf{discr}(A, B, C) \geq 0$, *and*

$$\begin{array}{ll} & v = v_o - v_i \ \wedge \ v' = v'_o - v_i \ \wedge \ v'' = v''_o - v_i \\ \wedge & s'' = s + t''v \ \wedge \ v_z \neq 0 \ \wedge \ \theta' = \theta^-(s_z, v_z) \\ \wedge & \mathsf{exit?}(s + \theta' v', v') \ \wedge \ {v_{ox}}^2 + {v_{oy}}^2 \geq {v'_{ox}}^2 \\ \wedge & (v'_{ox} = v_1 \ \vee \ v'_{ox} = v_2) \\ \wedge & v'_{oy} = \sqrt{{v_{ox}}^2 + {v_{oy}}^2 - {v'_{ox}}^2} \\ \wedge & \mathsf{sign}(-2(s_y - \theta' v_{iy})\theta' v'_{oy}) = \mathsf{sign}(E + 2(s_x - \theta' v_{ix})\theta' v'_{ox}) \\ \wedge & t' = \theta' \ \wedge \ \theta' < t'' \\ \wedge & \mathsf{exit?}(s + \theta' v', v'') \ \wedge \ [s''_z - (t'' - t')v''_z]v''_z \leq 0 \\ \wedge & v''_{ox} = \dfrac{t'v'_{ox} - t''v_{ox}}{t' - t''} \\ \wedge & v''_{oy} = \dfrac{t'v'_{oy} - t''v_{oy}}{t' - t''} \\ \wedge & v'_z = v_z \wedge v''_z = v_z \\ & \supset \\ & \mathsf{separation?}(s, v') \ \wedge \\ & \mathsf{heading\_only?}(v_o, v'_o) \ \wedge \\ & \mathsf{separation?}(s + t'v', v'') \ \wedge \\ & s + t''v = s + t'v' + (t'' - t')v'' \end{array}$$

Proof. The theorem follows directly from circle_escape [Theorem 44], out_circle_recovery [Theorem 49], circle_hd_only [Theorem 45], and vor_timeliness [Theorem 52].
□

5.4.17 Special Cases

Theorem 59 (line_escape_0)

$$\begin{array}{ll} & v' = v'_o - v_i \ \wedge \ {v_{ox}}^2 + {v_{oy}}^2 \geq {v_{ix}}^2 \\ \wedge & D^2 = {s_x}^2 \ \wedge \ v'_{ox} = v_{ix} \\ & \supset \ \mathsf{separation?}(s, v') \end{array}$$

Proof. First we use separation_lem [Lemma 1] to change the goal to

$$\mathsf{separation?}(s + \tau(s, v')v', v')$$

Next, using line_case_correctness [Theorem 2] we can establish this goal if we prove tangent_point? $(s + \tau(s, v')v', v')$. To do this we use the lemma tau_is_tangent_pt which reduces the goal to tangent_condition?(s, v'), which expands to

$$D^2({v'_x}^2 + {v'_y}^2) = (s_x v'_y - s_y v'_x)^2$$

From the premises $v' = v'_o - v_i$ and $v'_{ox} = v_{ix}$ we obtain: $v'_x = 0$. The goal follows by premise $D^2 = {s_x}^2$.
□

Theorem 60 (line_esc_0_hd_only)

$$\begin{gathered} {v_{ox}}^2 + {v_{oy}}^2 \geq {v_{ix}}^2 \;\wedge\; v'_{ox} = v_{ix} \;\wedge \\ v'_{oy} = \varepsilon\sqrt{{v_{ox}}^2 + {v_{oy}}^2 - {v_{ix}}^2} \;\wedge\; v'_{oz} = v_{oz} \\ \supset\; \mathsf{heading\_only?}(v_o, v'_o) \end{gathered}$$

Proof. To establish heading_only?(v_o, v'_o), we must show ${v'_{ox}}^2 + {v'_{oy}}^2 = {v_{ox}}^2 + {v_{oy}}^2 \;\wedge$ $v'_{oz} = v_{oz}$. Using the premises $v'_{ox} = v_{ix}$ and $v'_{oz} = v_{oz}$ we can reduce this to ${v_{ix}}^2 + {v'_{oy}}^2 = {v_{ox}}^2 + {v_{oy}}^2$. This follows trivially by squaring both sides of the definition of v'_{oy}.
□

The next theorem describes a strange maneuver. During the escape course, the ownship solves the conflict; during the recovery course, it only waits.

Theorem 61 (llhd_recovery_0)

$$\begin{array}{ll} & \mathsf{hor\_speed\_gt\_0?}(v'') \\ \wedge & D^2 = (s_x + t''v_x)^2 \\ \wedge & v'_x \neq 0 \;\wedge\; t' = t''\dfrac{v_x}{v'_x} \\ \wedge & v''_x = 0 \;\wedge\; v''_y = \dfrac{t''v_y - t'v'_y}{t'' - t'} \;\wedge\; v'_z = v_z \;\wedge\; v''_z = v_z \\ & \supset\; \mathsf{separation?}(s + t'v', v'') \end{array}$$

Proof. Define $s'' = s + t''v$. Using separation_lem [Lemma 1] we change the starting point for the goal:

$$\mathsf{separation?}(s'' - (t'' - t')v'' + (\tau(s'', v'') - t' + t'')v'', v'')$$

which can be simplified to

$$\mathsf{separation?}(s'' + \tau(s'', v'')v'', v'')$$

Using line_case_correctness [Theorem 2] this goal can be reduced to: tangent_point?$(s''+\tau(s'', v'')v'', v'')$.
Next using lemma tau_is_tangent_pt the goal is simplified to proving tangent_condition?(s'', v''), which expands to

$$D^2(v''^2_x + v''^2_y) = (s''_x v''_y - s''_y v''_x)^2.$$

Since $v''_x = 0$, this reduces to

$$D^2 v''^2_y = (s''_x v''_y)^2.$$

Since a premise provides that $s''^2_x = D^2$, this equality is true.
□

Theorem 62 (llhd_recovery_B)

$$
\begin{array}{ll}
 & \mathsf{hor\_speed\_gt\_0?}(v'') \ \wedge \ s'' = s + t''v \\
\wedge & D^2 = s_x''^2 \ \wedge \ v_x' \neq 0 \ \wedge \ t'' \neq t' \\
\wedge & t' = t''\dfrac{v_x}{v_x'} \ \wedge \ v_x'' = 0 \\
\wedge & v_y'' = \dfrac{t''v_y - t'v_y'}{t'' - t'} \\
\wedge & v_z' = v_z \ \wedge \ v_z'' = v_z \\
 & \quad \supset \ \mathsf{separation?}(s + t'v', v'')
\end{array}
$$

Proof. As in the proof of llhd_recovery_0 the goal can be reduced to

$$D^2(v_x''^2 + v_y''^2) = (s_x''v_y'' - s_y''v_x'')^2$$

This goal follows from the premises $v_x'' = 0$ and $D^2 = s_x''^2$ by simplification.
□

6 Conclusion

In this paper, a formal safety approach to the development of Air Traffic Management (ATM) systems is advocated. An example of the first step of this approach—the formal verification of a critical component of a distributed ATM concept (an air traffic resolution and recovery algorithm)—is presented.

One reason formal verification is valuable is that it provides the system designer with a much better means to handle the inherent unpredictability of complex systems. Once the behavior of some of the system's components is fully understood (through the formal verification process), the properties of other unpredictable components can be characterized more easily. Using a set of algorithms whose behavior is fully understood under explicitly enumerated assumptions greatly aids the designer of ATM operational concepts. Not only is the designer liberated from having to define contingency plans for failures of the algorithm, but by knowing the assumptions built into the algorithm, the designer has explicit knowledge about where to concentrate attention in order to produce a robust and safe operational concept. In this approach, human-in-the-loop simulation and expensive flight experiments are used to validate assumptions made during the formal verification. This is a major shift from traditional approaches where testing and simulation drive the safety validation and certification of avionics systems.

Proof-of-correctness of an algorithm does not guarantee a fault-free system. Complete verification of a system implementation must deal with many other considerations that do not arise in an abstract algorithm such as floating point overflow and underflow, validity of input data, meeting real-time deadlines, communication flaws, etc. Furthermore, at the system design level, additional algorithms are introduced to handle inter-aircraft communications (e.g. ADS-B), to detect and mask faulty input data, to format output data for pilot displays, to schedule execution,

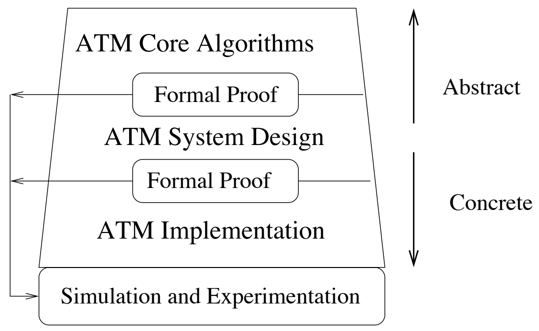


Figure 11. System Verification

and to coordinate with other systems. All of these algorithms must be shown to satisfy critical safety properties in addition to the core resolution and recovery algorithm.

Nevertheless, the verification of a resolution and recovery algorithm is a fundamental step toward complete verification of an ATM system. In particular, as the RR3D algorithm is refined into a high-level design and then translated into a programming language, additional formal proofs will be constructed. An ATM system that integrates an implementation of RR3D will be then formally supported by several layers of abstraction (Figure 11). These kinds of system design properties have not been addressed. Some issues related to system verification that will be addressed as the research continues include:

- **Strategic Resolution and Recovery**. RR3D is a state-based algorithm with minimal intent information. It propagates an aircraft trajectory based on current location, velocity, and an arrival time constraint. The arrival time constraint makes RR3D suitable for strategic CD&R. Indeed, Geser and Munoz describe in [7] an algorithm that incorporates RR3D to a conflict-free flight planner. The correctness of the flight planner is based on the correctness of RR3D. A separate resolution and recovery algorithm enables the decomposition of the flight planner into less complex parts and, more importantly, the decomposition of its correctness proof.

- **Geodesic Coordinates**. As with most geometric ATM algorithms, RR3D is presented in a Cartesian coordinate system that assumes a flat earth. An interface module has been developed that converts from geodesic coordinates to a Cartesian coordinates eliminating errors due to the flat earth assumption. The formalization and correctness proof of the coordinate transformation is in progress.

- **Floating Point Errors**. The formalization of RR3D assumes exact real arithmetic, whereas programming languages provide only floating point arithmetic. It is well-known that floating point numbers do not satisfy even elementary properties of real numbers. An interval analysis of RR3D that considers floating point inaccuracy errors, underflows, and overflows will complement a preliminary work on refinement of abstract algorithms into real-life programming languages.

A formal safety approach to the design and verification of ATM systems provides an intellectually defensible means to move advanced technology into the national airspace. Current research approaches—which center around comparative studies—can only establish some characteristics of a proposed system in preconceived scenarios. Formal analysis, including appropriate assumptions, provides an objective, absolute statement about a proposed system over all possible scenarios. Admittedly this analysis is a difficult and time-consuming endeavor. But, as unprecedented amounts of software are introduced in new safety critical roles, a more comprehensive assessment of safety is needed.

7 Appendix

7.1 Errors Found and Missing Assumptions

While checking the hand-written proof of RR3D presented in [1], we have discovered the following errors and missing assumptions (all formula and section numbers refer to [1]):

- Definition of C in Formula 4.16 should be:

  $$C \;=\; E^2 - 4(s_y - \theta' v_{iy})^2 \theta'^2 (v_{ox}^2 + v_{oy}^2)$$

  A square expression was missing in the original formula.

- The escape-circle case in Section 4.3 should refer to formula

  $$\begin{aligned}(s_x + \theta'(kv_{ox} - v_{ix}))(kv_{ox} - v_{ix}) +& \\ (s_y + \theta'(kv_{oy} - v_{iy}))(kv_{oy} - v_{iy}) &\leq 0\end{aligned}$$

  instead of Formula 4.28.

- The recovery-circle case in Section 4.3 should refer to formula

  $$\begin{aligned}(s''_x + (\theta'' - t'')(jv_{ox} - v_{ix}))(jv_{ox} - v_{ix}) +& \\ (s''_y + (\theta'' - t'')(jv_{oy} - v_{iy}))(jv_{oy} - v_{iy}) &\geq 0\end{aligned}$$

  instead of Formula 4.31.

- Solutions to Equation 4.42 are valid only under the assumption:

  $$\theta' v'_{oy}(s_y - \theta' v_{iy})(E + 2\theta' v'_{ox}(s_x - \theta' v_{ix})) \leq 0$$

- The correct definition of α'' in the solution of Formula 4.46 is

  $$\alpha'' \;=\; -(D^2 - s''^2_y)/(2 s''_x s''_y)$$

  A negative symbol is missing in the original formula.

- The circle/circle derivation in Section 4.4 is invalid. The formal verification reported in this paper contains a valid derivation.

7.2 Proofs Of Some Useful Lemmas

Lemma 14 (signs_are_opposite)

$$\begin{aligned}\neg\mathsf{pred\_sep?}(s, v, t'') \;\wedge\; |s_z| \geq H& \\ \supset\; \mathsf{sign}(s_z) = -\mathsf{sign}(v_z)&\end{aligned}$$

Proof. From $\neg\mathsf{pred\_sep?}(s, v, t'')$ we obtain $\neg\mathsf{vert\_sep?}(s + \hat{t}v)$ for some time $\hat{t}$, where $0 \le \hat{t} \le t''$. From definition of $\neg\mathsf{vert\_sep?}$ we get $|s_z + \hat{t}v_z| < H$.
Case 1 $[v_z > 0]$: If $\mathsf{sign}(s_z) = -1$, then the goal is reached trivially. If $\mathsf{sign}(s_z) = 1$, then $s_z > 0$, so by the second premise $H \le s_z$. We also observe that $\hat{t}v_z > 0$ so, $s_z + \hat{t}v_z > H$. This yields a contradiction with the $\neg\mathsf{vert\_sep?}(s + \hat{t}v)$ premise.
Case 2 $[v_z \le 0]$: If $\mathsf{sign}(s_z) = 1$, then the goal is reached trivially. If $\mathsf{sign}(s_z) = -1$, then $s_z < 0$, so by the second premise $s_z \le -H$. We also observe that $\hat{t}v_z < 0$ so, $s_z + \hat{t}v_z < -H$. This yields a contradiction with the $\neg\mathsf{vert\_sep?}(s + \hat{t}v)$ premise.
□

Lemma 15 (signs_ve_z)

$$\begin{aligned} &\neg\mathsf{pred\_sep?}(s, v, t'') \;\wedge\; |s_z| \ge H \;\wedge\; C > 0 \;\wedge \\ &v'_z = \frac{-\mathsf{sign}(v_z)H - s_z}{C} \;\wedge\; v'_z \ne 0 \\ &\qquad \supset\; \mathsf{sign}(v'_z) = \mathsf{sign}(v_z) \end{aligned}$$

Proof. Using lemma $\mathsf{signs\_are\_opposite}$ [14] we get $\mathsf{sign}(s_z) = -\mathsf{sign}(v_z)$. Rewriting the definition of v'_z with this and cross-multiplying we get:

$$v'_z C = \mathsf{sign}(s_z)H - s_z$$

Case 1: $[v'_z > 0]$: Expanding sign, we have: $v'_z C = H - s_z$ from which the result easily follows.
Case 2: $[v'_z \le 0]$: Expanding sign, we have: $v'_z C = -s_z - H$. Since magnitude of s_z is greater than H we have the desired result.
□

Lemma 16 (signs_vr_z)

$$\begin{aligned} &\neg\mathsf{pred\_sep?}(s, v, t'') \;\wedge\; |s_z| \ge H \\ \wedge \quad & v'_z = \frac{-\mathsf{sign}(v_z)H - s_z}{C} \\ \wedge \quad & t'' - C > 0 \;\wedge\; C > 0 \\ \wedge \quad & v''_z = \frac{t''v_z - v'_z C}{t'' - C} \\ & \quad \supset\; \mathsf{sign}(v''_z) = -\mathsf{sign}(s_z) \end{aligned}$$

Proof. Using lemma $\mathsf{signs\_are\_opposite}$ we get $\mathsf{sign}(s_z) = -\mathsf{sign}(v_z)$. Cross-multiplying definition of v'_z yields: $v'_z C = -\mathsf{sign}(v_z)H - s_z$. Cross-multiplying definition of v''_z yields: $v''_z(t'' - C) = t''v_z - Cv'_z$ Rewriting the second formula with the first formula

$$v''_z(t'' - C) = t''v_z - (-\mathsf{sign}(v_z)H - s_z).$$

From definition of $\mathsf{vert\_sep?}$ we get $|s_z + t^* v_z| < H$. One then case splits on both $v_z > 0$ and $v''_z > 0$ and simplifies all formulas. Inequality reasoning involving $v''_z t^*$

and $v_z t^*$ yields the desired result.
$\square$

7.3 Mapping of Notation to PVS

s	s	relative ownship position
s'	se	relative turn point
s''	sr	relative final position
v	v	relative ownship velocity
v'	ve	relative ownship escape velocity
v''	vr	relative ownship recovery velocity
v_x	v‘x	x component of relative ownship velocity
v'_y	ve‘y	y component of relative escape velocity
v''_z	vr‘z	z component of relative recovery velocity
t	t	time variable
t'	te	turn time
t''	tr	final time

References

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

2. NASA: Concept Definition for Distributed Air/Ground Traffic Management (DAG-TM), Version 1.0, 1999. Advanced Air Transportation Technologies (AATT) Project. NASA Ames Research Center. NASA Langley Research Center.

3. Wing, D. J.; Adams, R. J.; Duley, J. A.; Legan, B. M.; Barmore, B. E.; and Moses, D.: Airborne Use of Traffic Intent Information in a Distributed Air-Ground Traffic Management Concept: Experiment Design and Preliminary Results. NASA/TM-2001-211254, NASA Langley Research Center, Hampton VA 23681, USA, November 2001.

4. RTCA: Final Report of the RTCA Board of Directors' Select Committee on Free Flight. Issued 1-18-95, RTCA, Washington, DC, 1995.

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

6. Dowek, G.; Muñoz, C.; and Geser, A.: Tactical Conflict Detection and Resolution in 3-D Airspace. *4th USA/Europe Air Traffic Management R&D Seminar (ATM-2001)*, Santa Fe, New Mexico, 2001. Extended version available as ICASE Report No. 2002-12 NASA/CR-2002-211637.

7. Geser, A.; and Muñoz, C.: A Geometric Approach to Strategic Conflict Detection and Resolution, 2002. Proceedings of the 21st Digital Avionics Systems Conference.

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

9. Sanford, B.; Harwood, K.; Nowlin, S.; Bergeron, H.; Heinrichs, H.; Wells, G.; and Hart, M.: Center/TRACON automation system: Development and evaluation in the field. *38th Annual Air Traffic Control Association Conference*, October 1993.

10. Brudnicki, D.; Lindsay, K.; and McFarland, A.: Assessment of Field Trials, Algorithmic Performance, and Benefits of the User Request Evaluation Tool (URET) Conflict Probe. *16th Digital Avionics Systems Conference*, Irvine, CA, October 1997, pp. 9.3.35–9.3.44.

11. Kuchar, J.; and Yang, L.: A Review of Conflict Detection and Resolution Modeling Methods. *IEEE Transactions on Intelligent Transportation Systems*, vol. 1, no. 4, December 2000, pp. 179–189.

12. Durand, N.; Alliot, J.-M.; and Noailles, J.: Automatic aircraft conflict resolution using genetic algorithms. *Symposium on Applied Computing*, Philadelphia, PA, 1996.

13. Granger, G.; Durand, N.; and Alliot, J.-M.: Optimal resolution of en route conflicts. *4th USA/Europe Air Traffic Management R&D Seminar (ATM-2001)*, Santa Fe, New Mexico, 2001.

14. McDonald, J.; and Vivona, R.: Strategic Airborne Conflict Detection of Air Traffic and Area Hazards. NASA Contract: NAS2-98005 RTO-29, TITAN Systems Corporation, SRC Division, November 2000.

15. Durand, N.; Alliot, J.-M.; and Medioni, F.: Neural Nets trained by genetic algorithms for collision avoidance. *Applied Artificial Intelligence*, vol. 13(3), 2000.

16. Tomlin, C.; Pappas, G.; and Sastry, S.: Conflict Resolution for Air Traffic Management: A Study in Multi-Agent Hybrid Systems. *IEEE Transactions on Automatic Control*, vol. 43(4), 1998.

17. Chiang, Y.-J.; Klosowsky, J.; Lee, C.; and Mitchell, J.: Geometric Algorithms for Conflict Detection/Resolution in Air Traffic Management. *36th IEEE Conference on Decision and Control*, 1997.

18. Frazzoli, E.; Mao, Z.-H.; Oh, J.-H.; and Feron, E.: Resolution of Conflicts Involving Many Aircraft via Semidefinite Programming. *Journal of Guidance, Control, and Dynamics*, vol. 24, no. 1, 2001, pp. 79–86.

19. Eby, M.: A Self-Organizational Approach for Resolving Air Traffic Conflicts. *Lincoln Laboratory Journal*, vol. 7, no. 2, 1994, pp. 239–254.

20. Bilimoria, K.: A Geometric Optimization Approach to Aircraft Conflict Resolution. *Guidance, Navigation, and Control Conference*, Denver, CO, vol. AIAA 2000-4265, August 2000.

21. Wing, D.; Adams, R.; Barmore, B.; and Moses, D.: Airborne Use of Traffic Intent Information in a Distributed Air-Ground Traffic Management Concept: Experiment Design and Preliminary Results. *4th USA/Europe Air Traffic Management R&D Seminar (ATM-2001)*, Santa Fe, New Mexico, 2001.

22. Bilimoria, K.; and Lee, H.: Aircraft Conflict Resolution with an Arrival Time Constraint. *Guidance, Navigation, and Control Conference*, Monterey, CA, vol. AIAA 2002-4444, August 2002.

23. Muñoz, C.; Butler, R.; Carreño, V.; and Dowek, G.: On the verification of conflict detection algorithms. NASA/TM-2001-210864, NASA Langley Research Center, NASA LaRC,Hampton VA 23681-2199, USA, May 2001.

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704–0188*

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.
PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)* 01-04-2004	**2. REPORT TYPE** Technical Publication	**3. DATES COVERED** *(From - To)*
4. TITLE AND SUBTITLE Formal Verification of a Conflict Resolution and Recovery Algorithm		**5a. CONTRACT NUMBER**
		5b. GRANT NUMBER
		5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) Maddalon, Jeffrey; Butler, Ricky; Geser, Alfons; and Muñoz, César		**5d. PROJECT NUMBER**
		5e. TASK NUMBER
		5f. WORK UNIT NUMBER 23-727-01-26
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) NASA Langley Research Center Hampton, VA 23681-2199		**8. PERFORMING ORGANIZATION REPORT NUMBER** L–18323
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/TP–2004–213015

12. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified-Unlimited
Subject Category 63
Availability: NASA CASI (301) 621-0390 Distribution: Standard

13. SUPPLEMENTARY NOTES
An electronic version can be found at http://techreports.larc.nasa.gov/ltrs/ or http://ntrs.nasa.gov.

14. ABSTRACT
New air traffic management concepts distribute the duty of traffic separation among system participants. As a consequence, these concepts have a greater dependency and rely heavily on on-board software and hardware systems. One example of a new on-board capability in a distributed air traffic management system is air traffic conflict detection and resolution (CD&R). Traditional methods for safety assessment such as human-in-the-loop simulations, testing, and flight experiments may not be sufficient for this highly distributed system as the set of possible scenarios is too large to have a reasonable coverage. This paper proposes a new method for the safety assessment of avionics systems that makes use of formal methods to drive the development of critical systems. As a case study of this approach, the mechanical verification of an algorithm for air traffic conflict resolution and recovery called RR3D is presented. The RR3D algorithm uses a geometric optimization technique to provide a choice of resolution and recovery maneuvers. If the aircraft adheres to these maneuvers, they will bring the aircraft out of conflict and the aircraft will follow a conflict-free path to its original destination. Verification of RR3D is carried out using the Prototype Verification System (PVS).

15. SUBJECT TERMS
air traffic management, safety assessment, conflict resolution, conflict recovery, arrival time constraint, theorem proving, formal methods

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`)
U	U	U	UU	82	**19b. TELEPHONE NUMBER** ***(Include area code)*** (301) 621-0390

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39.18