Ninth USA/Europe Air Traffic Management Research and Development Seminar (ATM2011)

A New Approach for Designing Safer Collision Avoidance Systems

Mykel J. Kochenderfer, James P. Chryssanthacopoulos, and Roland E. Weibel
Lincoln Laboratory
Massachusetts Institute of Technology
Lexington, Massachusetts, USA

***Abstract*—The Traffic Alert and Collision Avoidance System (TCAS) has been shown to significantly reduce the risk of mid-air collision and is currently mandated worldwide on all large transport aircraft. Engineering the collision avoidance logic was a very costly undertaking that spanned several decades. The development followed an iterative process where the logic was specified using pseudocode, evaluated on encounters in simulation, and revised based on performance against a set of metrics. Modifying the logic to get the desired behavior is difficult because the pseudocode contains many heuristic rules that interact with each other in complex ways. Over the years, the TCAS logic has become challenging to maintain. With the anticipated introduction of next-generation air traffic control procedures and surveillance systems, the logic will require significant revision to prevent unnecessary alerts. Recent work has explored a new approach for designing collision avoidance systems that has the potential to shorten the development cycle, improve maintainability, and enhance safety with fewer false alerts. The approach involves leveraging recent advances in computation to automatically derive optimized collision avoidance logic directly from encounter models and performance metrics. This paper outlines the general approach and discusses the anticipated impact on development, safety, and operation.**

***Index Terms*—Collision avoidance systems, Traffic Alert and Collision Avoidance System (TCAS), probabilistic models, optimization.**

I. Introduction

The Traffic Alert and Collision Avoidance System (TCAS) is currently mandated worldwide on large transport aircraft to reduce the risk of mid-air collision. TCAS uses onboard beacon radar surveillance to monitor local air traffic. Embedded in the system is logic that determines when to alert pilots to potential collision and which vertical maneuver to recommend to pilots [1]. The logic has evolved over the course of many years beginning in the 1970s. The iterative development process (top half of Figure 1) involved specifying the logic using pseudocode and evaluating the system in simulation using encounter models. These encounter models were based on operational data that capture the properties of close encounters in the airspace. The simulation results were evaluated against a set of performance metrics, and the logic was revised manually to address potential performance issues. The evolutionary development process of the collision avoidance logic has resulted in complex pseudocode with many heuristic rules and parameter settings whose justification has been lost over the years. Unfortunately, due to the complexity of the system, correcting issues without introducing new vulnerabilities is very difficult and costly. Next-generation procedures and new sensor systems will require reengineering much of the logic and tuning many parameters.

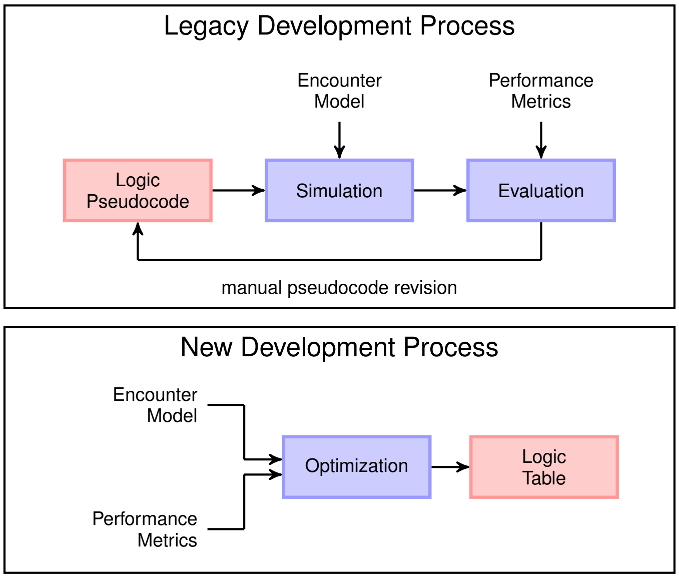


Fig. 1. Collision avoidance logic development processes.

Several factors make building a robust collision avoidance system difficult. The system must account for state uncertainty due to sensor noise, dynamic uncertainty in the future trajectories of the aircraft, aircraft performance constraints, and the variability in the delay and strength of the pilot response. The system must act conservatively to ensure an exceptionally high degree of safety, but it should minimize disruption of normal operations. Nuisance alerts negatively impact pilot compliance, result in unnecessary course deviation, and can induce conflicts with other aircraft. Deciding exactly when an alert is necessary requires carefully accounting for many

This work is sponsored by the Federal Aviation Administration under Air Force Contract #FA8721-05-C-0002. Opinions, interpretations, conclusions, and recommendations are those of the authors and are not necessarily endorsed by the United States Government.

constraints and sources of uncertainty, something that human designers are not especially well equipped to do on their own when building a complex system.

Recent work (e.g., [2]–[5]) has pursued a new model-based optimization approach to developing logic that has the potential to shorten the development cycle, improve maintainability, and enhance safety with fewer nuisance alerts. This new approach involves using computers to directly optimize the logic based on encounter models of traffic and performance metrics (bottom half of Figure 1). Such an approach allows humans to focus their development effort on building models and performance metrics instead of the difficult task of optimizing the logic. The computer-generated logic will still have to undergo rigorous safety analysis that may result in modifications to the model or metrics. However, the development cycle will be shortened because the logic does not require manual revision. Because the computer is able to take into account all possible future aircraft trajectories and their likelihood, the optimization is able to produce safer logic with fewer nuisance alerts than the current version of TCAS (Version 7.1). An early prototype system has demonstrated the strength of this approach in simulation.

This paper provides a high-level overview of how the encounter models and performance metrics are used in the computer-based optimization of the logic, and it explains how the logic is used on the aircraft. It then presents results from various safety studies comparing the performance of the new system with the current version of TCAS. The latter part of the paper discusses the impact this new development process has on the safety approval process and maintenance. The paper concludes with a discussion of other collision avoidance approaches and areas for further research.

II. Logic Development

The new approach requires specifying an encounter model and using computational methods to find the logic that optimizes performance against a set of metrics. The logic is represented as a numerical table that is used during flight to determine the expected cost of different actions (e.g., no alert, climb, or descend) available to the alerting system. This section discusses the process for constructing the table (left half of Figure 2).

A. Encounter Models

Success of a collision avoidance system depends on the ability of the logic to predict the future trajectories of aircraft given the current state (which includes position and velocity information) of the aircraft. Due to variability in aircraft dynamics and pilot response to advisories, it is impossible to exactly predict the future path of the aircraft involved in an encounter. Given the same initial state, a wide variety of different future trajectories are possible, some more likely than others. When issuing an advisory, it is important to account for the full spectrum of possible future trajectories and their likelihood. A probabilistic model of future trajectories can be constructed from domain expertise and recorded data [6]–[8].

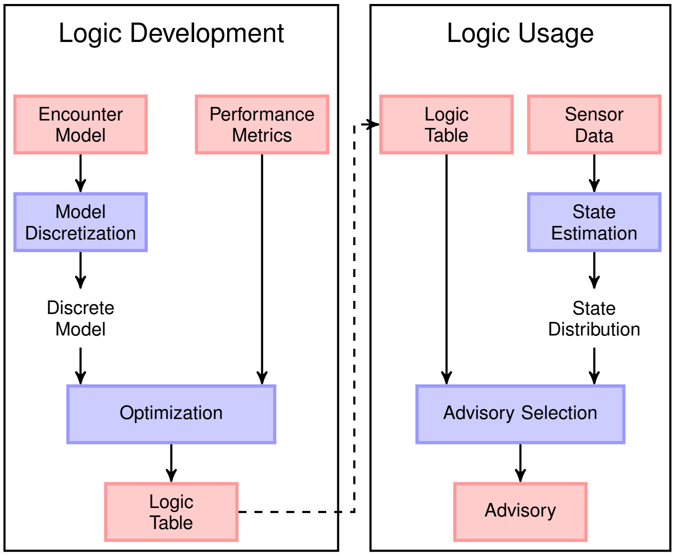


Fig. 2. Logic development and usage.

B. Performance Metrics

One of the standard safety metrics used for evaluating TCAS is the probability of near mid-air collision (NMAC), historically defined to be when an intruder comes within 500 ft horizontally and 100 ft vertically [9]. In addition to safety metrics, there are also operational metrics, such as the alert rate, that have been used to measure how disruptive the system is to normal operations. Additional metrics might include the rate of advisory strengthening (e.g., changing a climb at 1500 ft/min to a climb at 2500 ft/min) and reversing (e.g., changing a climb to a descend advisory).

A variety of different performance metrics can be combined into a single cost function by weighting the various components. The weights influence, for example, how much more important it is to prevent NMAC versus an unnecessary alert. The optimal collision avoidance logic is the one that is expected to provide the lowest expected cost. Because the cost function defines optimality, it is important that it truly reflects the priorities of the designers. One way to choose the relative weighting of the performance metrics is to vary the weighting until some safety threshold is met while maintaining operational acceptability in terms of, for example, alert, strengthening, and reversal rate [2].

C. Optimization

Several different computational techniques may be applied for optimizing decision logic based on a probabilistic model and cost function [10]. One approach, called dynamic programming (DP), has been shown to work well for airborne collision avoidance [2]. DP requires the dynamics to be described as a finite state model. However, the various state variables, such as the climb rates of the aircraft involved in the encounter, are naturally continuous quantities. The continuous state variables must therefore be discretized into a finite set of values. The finer the discretization, the better

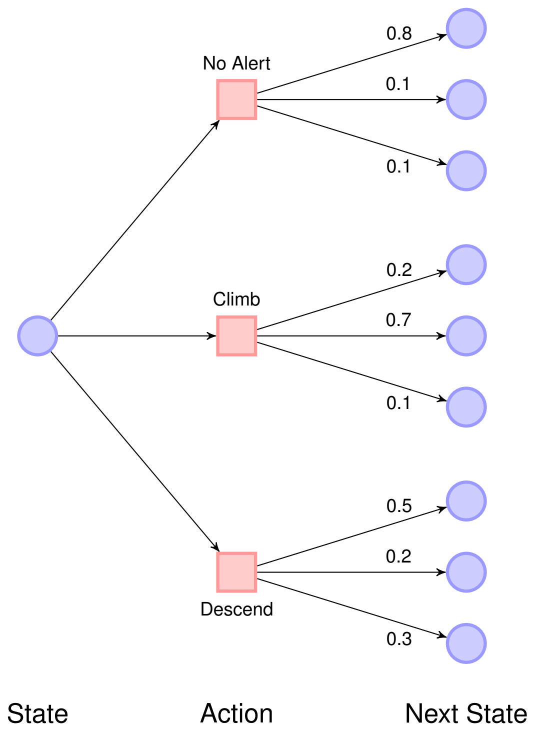


Fig. 3. Discrete transition model from a single state. The transition probabilities in the figure are notional, but they would be derived from the encounter model. In a realistic model, non-zero probability may be assigned to several dozen next states instead of only three as shown in the figure.

the finite state model represents the dynamics, but it comes at the cost of additional computation and memory storage. In order to adequately model the aircraft dynamics for collision avoidance, millions of discrete states are required.

Once the state space has been discretized, the probabilistic transitions between the states for the different actions must be inferred from the encounter model. Figure 3 shows a notional example of the transitions available from one of the states. From the state on the left, three actions are available: no alert, climb, and descend. The next state is selected randomly according to the probabilities shown in the model. The figure shows only three possible resulting states for each action, but in a realistic model, there may be dozens. The methodology for determining these transition probabilities is detailed in [2].

Once the cost function has been chosen and the model discretized, DP may be applied to solve for the best possible action to execute from every possible state. The best action to take from a particular state is the one that minimizes the expected cost. In the process of determining the best action from each state, DP computes the expected cost of executing each action from every state for one step and then continuing with the optimal strategy. The expected cost associated with every state and action pair is stored in a large table. This table, which essentially represents the logic, is what is used in real time on the aircraft to decide when and how to alert.

One potential limitation of this optimization approach that relies upon the discretization of the encounter model is that the number of states grows exponentially with the number of state variables. However, by leveraging the structure inherent in the problem, the complexity of the problem can be reduced [11]. To adequately represent logic for encounters in three-dimensions using a naïve discretization, a table on the order of 1 TB is necessary. By carefully decoupling the horizontal and vertical dynamics, the storage and computational requirements are reduced by a factor of 2000. The current prototype logic requires only 500 MB of memory, which is certainly manageable with current technology, and it is feasible to accommodate many more states if it is later determined that modeling additional variables is necessary. The total time required for computing the optimal logic for the current prototype system is three minutes on a single processor.

Since the collision avoidance logic is critical to safety, it is important for humans to understand and be able to anticipate the behavior of the system. One way to visualize the logic is as plots of the alerting regions over slices of the state space. These logic plots play an important role in the development phase, allowing the designers to quickly assess how changing different cost or model parameters affects the behavior of the logic without having to rely on simulation.

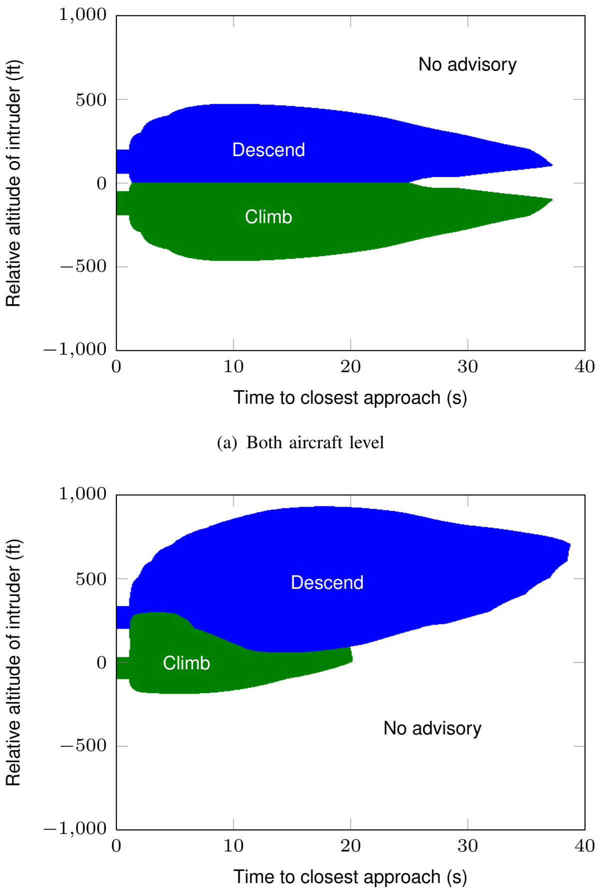
Figure 4 shows two examples of such plots for a logic optimized according to a particular cost function and dynamic model. In the first plot, both aircraft are initially level. In the second plot, the own aircraft is initially climbing at 1500 ft/min. The blue region indicates where the logic will issue a descend advisory, and the green region indicates where the logic will issue a climb advisory. Figure 4(a) indicates that the own aircraft should descend when the intruder is above and climb when the intruder is below. The logic does not alert in the notch on the right side of the alerting region even though the vertical separation is small. The logic delays alerting until it is more certain whether the intruder will end up above or below the own aircraft. In Figure 4(b), the logic issues a climb advisory in some cases when the intruder is above the own aircraft when there is insufficient time for the own aircraft to accelerate downwards to miss the intruder.

III. Logic Usage

As discussed in the previous section, the logic table is computed offline during the development phase. All of the complexity of the logic is represented as a table of expected costs instead of heuristic rules specified by pseudocode. The usage of the logic table during flight (right half of Figure 2) is outlined in this section.

A. State Estimation

The aircraft receives sensor measurements of the local traffic environment. Due to sensor limitations and noise, it is not possible to infer the state of the environment exactly. The current TCAS sensor provides fairly accurate range measurements of intruding aircraft, but the bearing measurements are relatively noisy and the altitude of the intruder is quantized [12]. TCAS currently uses a set of filters to estimate the state of the environment from the noisy measurements.



(a) Both aircraft level

(b) Own aircraft climbing at 1500 ft/min, intruder level

Fig. 4. Example logic plots for slices of the state space where no advisory has yet been issued. The shape and size of the alerting regions depend on the cost function and model.

The prototype system can be adapted to accommodate different surveillance systems with different error characteristics. The state estimation process requires a model of the sensor that specifies the probability of receiving different sensor measurements given different true states of the environment. A process known as recursive Bayesian estimation uses the sensor model and dynamic model to efficiently infer a probability distribution over the state space from a sequence of observations [13]. The current TCAS system does not infer a full probability distribution over states; it uses a single point estimate. By explicitly taking into account state uncertainty, the system can be made more robust to sensor error, resulting in a lower alert rate and improved safety [14].

There are different ways to represent the state distribution. One method that is flexible enough to accommodate non-Gaussian distributions is to use weighted, deterministically-chosen state samples [15]. Such an approach has been widely used for tracking and state estimation, and it has been shown to work well in practice for collision avoidance. Using a weighted sample scheme (illustrated in Figure 5) makes advisory selection based on expected cost straightforward.

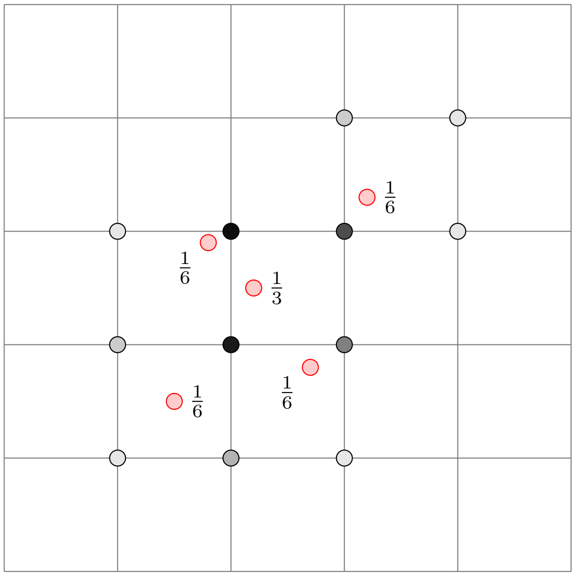


Fig. 5. Notional state distribution representation. The vertices of the grid correspond to discrete states. The red points correspond to state distribution samples with the specified weights. Determining the expected costs for the various samples requires interpolating values in the logic table at the discrete states enclosing the samples. The relative contributions of the enclosing discrete states to the value associated with the samples are indicated by their shading. To determine the overall cost for each action given this state distribution, the interpolated values are averaged together using the weights assigned to the samples.

B. *Advisory Selection*

Choosing the best advisory involves computing the expected cost for each of the advisories available to the system given the current state distribution. For each of the weighted samples representing this distribution, the expected cost of each action is determined by interpolating values in the logic table. An overall expected cost for each action is computed by averaging the expected costs using the weights associated with the samples. The system then simply executes the action with the lowest overall expected cost. Because this process involves simple table lookups and interpolation, deciding whether to alert and which advisory to issue is extremely fast.

C. *Coordination*

In situations where both aircraft in an encounter situation are equipped with a collision avoidance system, it is important that advisories be coordinated to reduce the risk of inducing collision. For example, if two aircraft have slightly different views of the world due to sensor error, there is some risk that the two collision avoidance systems may independently issue climb advisories. To prevent such situations, TCAS currently sends a coordination message to the other TCAS unit warning it to not issue advisories in the same direction. Conflicts can still occur due to timing, and are resolved based on a unique address number assigned to each aircraft. In the event of a multiple-threat encounter as described below, such coordination is performed pairwise with each TCAS-equipped threat, coordinating only the component of the composite advisory that is due to that other aircraft.

It is straightforward to apply the same general coordination strategy as TCAS to the new expected cost approach [2]. If the system receives a message to not climb, for example, it will remove all inconsistent advisories from consideration and issue whichever remaining advisory has the lowest expected cost. Such a simple strategy is able to significantly reduce the risk of collision. Adopting the same coordination strategy as TCAS may simplify interoperability of next-generation systems with existing TCAS equipment.

D. Multiple Threat Resolution

Although encounters involving two or more intruders are relatively rare in the current airspace, the collision avoidance logic must be robust to such situations. The TCAS pseudocode that governs the resolution of multiple threats is the most complicated part of the logic. The logic involves choosing advisories for each of the threats independently and then arbitrating between them. A similar strategy can be used with the new method, but some recent research has explored simpler methods that involve combining action costs associated with different threats and then choosing the action with the lowest overall expected cost [2]. The truly optimal solution would involve incorporating all intruders into a dynamic model and solving for the optimal logic, but the additional intruders can make the number of discrete states explode. However, simply combining the costs associated with the various intruders in isolation has been shown to result in better performance than the current version of TCAS [2].

IV. Example Encounter

Figure 6 shows the vertical profile of a randomly generated encounter where TCAS induces an NMAC. The own aircraft is shown flying from the left, and the intruder approaches from the right. The optimized logic issues a descend to pass below the intruder 17 s into the encounter. The expected cost of issuing a descend advisory is approximately 0.00928, lower than the expected cost for issuing a climb advisory (0.0113) or for not issuing an advisory (0.00972). After the descend advisory, the intruder begins to increase its descent, causing the optimized logic to reverse the descend to a climb. The pilot begins climbing three seconds later, resulting in 595 ft of vertical separation at closest approach.

TCAS initially issues a climb advisory 4 s into the encounter because it anticipates, using its deterministic dynamic model, that by climbing it can safely pass above the intruder. Later, when the own aircraft is executing its climb advisory, TCAS reverses the climb to a descend because it projects that maintaining the climb will not provide the required separation. TCAS strengthens the advisory 3 s later, but fails to resolve the conflict. The aircraft miss each other by 44 ft vertically.

Although this is just one example encounter, it illustrates the behavior that is typical of the optimized and TCAS systems. The optimized system generally waits a little longer than TCAS before it commits to a particular advisory, allowing it to provide a greater level of safety while alerting less frequently, as discussed in the next section.

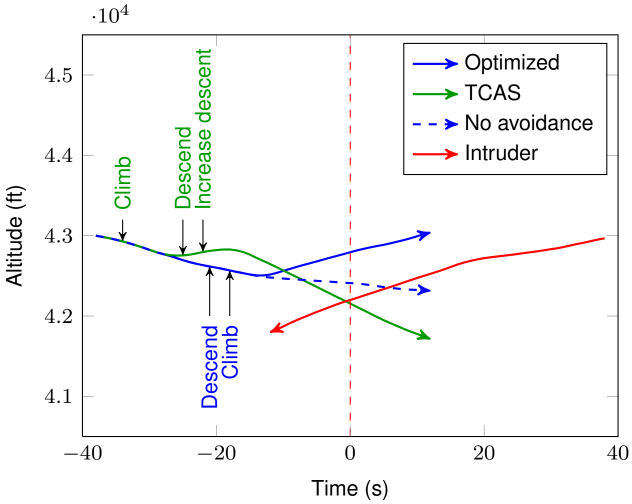


Fig. 6. Example encounter comparing the optimized system to TCAS.

V. Performance Evaluation

Due to their safety-critical nature, collision avoidance systems must undergo rigorous analysis before deployment in the airspace [9]. Civil aviation authorities require a combination of flight tests and detailed simulation studies to ensure effectiveness and safety. Flight tests are useful for evaluating the system in actual operation, but only a few situations can be examined due to time, cost, and safety constraints. Simulation studies are required for testing the robustness of the system to a wide variety of encounters. If the encounter model is representative of those expected to be found in the airspace, then the actual probabilities of different events, such as NMAC, can be estimated through Monte Carlo simulation. The results of such simulations were a major factor contributing to the certification and eventual mandate of TCAS. Any new system would also need to be evaluated in this way, and now there are higher-fidelity models based on much larger volumes of radar data to support safety analysis [7].

Preliminary safety studies have been conducted on the prototype logic [2]. Table I summarizes results from one million simulated encounters with a single intruder equipped with a transponder but no collision avoidance system. The same surveillance model and encounter scenarios were used for evaluating both systems. For the particular cost function chosen for Table I, the optimized logic is twice as safe as TCAS while alerting less than half the time. The optimized logic also reversed less than a quarter as often as TCAS, but it strengthened its advisories significantly more frequently. Operationally, strengthening is more acceptable than reversing. The cost function, though, can be adjusted based on safety community feedback to reduce the rate of strengthening advisories. However, as can be expected, reducing the rate of strengthening by increasing its cost relative to the other metrics will result in more NMACs, alerts, or reversals. A longer report [2] discusses results of simulations with equipped intruders, multiple intruders, and different sensor noise levels.

TABLE I
PERFORMANCE EVALUATION

	TCAS	Optimized System	Ratio
Pr(NMAC)	$1.43 \cdot 10^{-4}$	$6.98 \cdot 10^{-5}$	0.49
Pr(Alert)	$5.03 \cdot 10^{-1}$	$2.01 \cdot 10^{-1}$	0.40
Pr(Strengthening)	$1.18 \cdot 10^{-2}$	$1.07 \cdot 10^{-1}$	9.04
Pr(Reversal)	$3.26 \cdot 10^{-3}$	$7.99 \cdot 10^{-4}$	0.25

Safety curves are one way to evaluate different systems without committing to a particular setting of parameters [16]. Figure 7 shows safety curves generated using a simplified encounter model. One curve shows the performance of the optimized system while varying the alert cost parameter. The other curve shows TCAS performance as the sensitivity level parameter is varied. Since there are points on the optimized curve that are above and to the left of all the points in the TCAS curve, the optimized system dominates the TCAS system with respect to the safety and alert metrics.

Choosing appropriate cost parameters will require extensive discussion within the safety community. However, deciding the trades between, for example, strengthening rate and reversal rate is much more straightforward than deciding whether to add an additional rule to the TCAS pseudocode or change a parameter in the logic. The focus of human effort in the new approach is on balancing concrete objectives rather than designing the actual logic.

VI. Certification Process

Avionics for collision avoidance require certification by civil aviation authorities. The certification of systems built using the new approach is expected to follow a process similar to that used for legacy TCAS. The safety approval process for collision avoidance systems differs from typical avionics development due to the complexity of the logic, liability issues associated with collision avoidance, and the benefits the system provides to the international civil aviation system. For TCAS, the logic was specified in the Minimum Operational Performance Standards [1]. These common standards were developed by a community of experts and are recognized as the certification standard for TCAS equipment. Individual vendors implement the TCAS specification, which is then certified by civil authorities. Thus, it is useful to divide the discussion of certification into development and implementation phases.

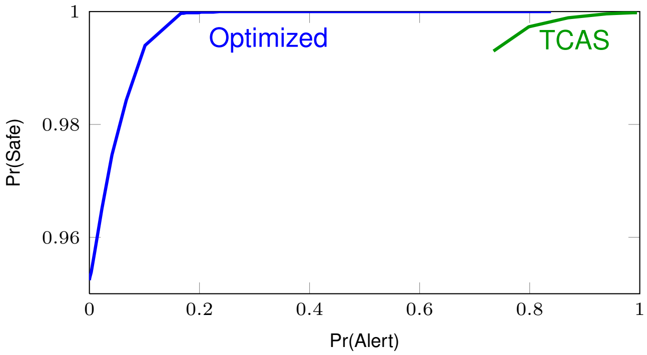


Fig. 7. Safety curves comparing the alerting and safety performance of legacy TCAS with an optimized collision avoidance system. The curve for TCAS was traced by varying the sensitivity level, and the curve for the optimized system was traced by varying the cost of alerting.

A. Certification of Logic Specification

Development of new logic for collision avoidance will require the participation of aviation stakeholders, including civil aviation authorities, operators, avionics and airframe manufacturers, and technical experts. The community standards development process differs from that of a single vendor applying for certification in that it takes place in a more collaborative atmosphere without strictly defined process standards for certification. Stakeholders and regulators agree on minimum acceptable specifications and safety performance by vetting analysis and proposals through a standards development organization. For complex avionics, the standards development process typically results in certification to higher-level safety standards, such as a target level of safety, rather than to lower-level regulations.

Certification assures that avionics performance will be safe and operationally acceptable when implemented. Historically, acceptability of collision risk has been judged by using a risk ratio, defined as the ratio of NMACs with an adopted change to NMACs without. More recent TCAS changes have been approved against a threshold level of risk (e.g., the rate of NMACs per hour) [9]. Operational acceptability has been measured by false alert rates, reversal and strengthening rates, and flight path deviations.

Tradeoffs between safety and operational metrics can be incorporated directly into the optimization by choosing weights in the cost function, making tradeoffs more transparent. Enhanced transparency facilitates technical review and confidence in a community setting. An additional advantage of the model-based approach is that the resulting safety performance is guaranteed to be optimal under the modeling assumptions.

Airworthiness certification policies require the assurance of integrity of airborne software. One acceptable means, and the one typically used for avionics, is provided by RTCA DO-178B [17]. DO-178B specifies several objectives to be met at different levels of software specification from high-level requirements to executable code. The decomposition presumed by the standard does not readily apply to the generation of a logic table through model-based optimization. In that approach, high-level requirements in safety and operational acceptability are framed in the cost function and optimized directly. Verification that the objectives are met can be done through Monte Carlo simulation or direct evaluation of the logic table, reviewed by community experts. The evaluation is assured directly from the outputs of the simulation, not by tracing the implementation of requirements. In this implementation, the rigor of assurance measures required of the computer analysis generating the lookup table is unclear, and will require guidance from approval authorities.

To be implemented by vendors, the logic must be specified using some representation. Past versions of TCAS were specified using both pseudocode and state charts, both sufficiently complex to be understood only by experts. The new logic uses a lookup table that deterministically and unambiguously specifies the optimal alerting action for every possible state. The process of generating this lookup table is similar to generating navigational databases, where software tools are used to encode derived data and ensure its integrity when transmitting to different users. Similar methods should be investigated to ensure the integrity of the logic table [18].

B. Certification of Logic Implementation

It is expected that vendors will implement state estimation and collision logic with sensors, displays, and other system elements that comply with a common minimum specification. Avionics must be certified as compliant with the minimum specification to be used on aircraft. Several methods are used to test compliance. One method currently used with TCAS is to test the system in simulation on a limited number of test cases and compare it to a certified reference system. With legacy TCAS, it was not possible to rigorously verify the full implementation. With the new approach, it may be possible to efficiently test the entire state space to ensure that the logic is implemented correctly. A logic table reduces the amount of code required to implement and verify, lowering vendor cost.

VII. Maintenance

The airspace will continue to evolve with the introduction of new procedures, aircraft capabilities, and sensor systems. To ensure a high degree of safety while maintaining operational acceptability, collision avoidance systems will also need to evolve. Changing the TCAS pseudocode is very difficult due to the complexity of the logic and the ways in which the various rules and parameters interact with each other. The changes incorporated since TCAS was originally introduced required tremendous effort, and it is not clear that the process of making small updates to the pseudocode will scale well into the future.

Adopting the new development approach will likely require equipping aircraft with new hardware, although the surveillance system can be left intact if so desired. The current TCAS equipment does not possess the memory capacity to hold the logic tables. However, once the investment has been made in equipping aircraft with hardware that accommodates a tabular representation of the logic, the system can be updated as necessary by uploading revised tables.

Reoptimizing the logic in response to changes in the model or metrics is much easier than trying to incorporate changes in the pseudocode. Of course, Monte Carlo analysis is still required to validate the reoptimized logic table before deployment. Manufacturers will not need to implement any new code, which will lower long-term development cost and speed the deployment of logic changes. Although the short-term development cost for introducing new logic might be more significant than evolving the current TCAS logic, there are important long-term benefits for such an investment.

VIII. Alternative Approaches

A wide variety of other approaches to collision avoidance have been proposed [19]. Some approaches (e.g., [20]) use nominal trajectory prediction, like TCAS, where a deterministic model is used to extrapolate the positions of the aircraft into the future. If an intruder is predicted to come within some protected zone in the near future, the system alerts. The problem with this approach is that it does not explicitly account for low-probability events that can lead to collision. To make the logic robust to deviation, such systems need to incorporate complex heuristics. Another approach is to use worst-case trajectory prediction, where the system examines a range of possible future maneuvers and determines whether any of them results in NMAC. One disadvantage of this approach is that it can cause excessive alerting.

Probabilistic trajectory propagation, which is central to the approach proposed in this paper, tends to result in much more robust behavior because it takes into account the full spectrum of future trajectories as well as their likelihood [19]. However, there have been probabilistic modeling approaches suggested in the literature that do not use the same optimization method presented here. Many systems (e.g., [21]–[23]) adopt a threshold-based alerting strategy where an alert is issued if the probability of collision passes above some fixed value during the encounter. Such systems tend to perform well in practice, but they are not optimal in general.

It should be mentioned that there are other methods for computing optimal strategies given a probabilistic model and cost function that do not involve discretizing the state space [24]. There has been a tremendous amount of research on such methods in the artificial intelligence and operations research communities that could be leveraged for building better collision avoidance systems. However, the discretization approach with its expected cost table works well and is conceptually easier to understand than some of the other methods. The main limitation of the discretization approach is that the number of discrete states grows exponentially with the number of state variables. Current hardware easily supports the number of states required by the model to represent the logic table, so there is little motivation to pursue other methods at this time.

IX. Conclusions and Further Work

This paper has summarized ongoing work exploring a new approach for automatically deriving robust airborne collision avoidance logic from encounter models and performance metrics. Experiments have demonstrated that the approach has the potential to significantly improve safety while reducing the rate of unnecessary alerts compared to the current TCAS logic. The approach focuses human effort in the areas where it is needed most, in building models and deciding on performance metrics, and leaving the difficult task of optimizing the logic to computers. Such an approach to developing and maintaining logic will become increasingly important as the airspace evolves with the introduction of new surveillance systems, procedures, and aircraft.

Key challenges in certifying the new approach will be software certification and cost function development. Equivalent approaches to current software certification approaches will need to be coordinated, but will overlap well with current community development approaches. The logic offers several certification advantages, including direct specification of safety and operational tradeoffs, reduced code to develop and inspect, and robustness to modeling assumptions.

Software for computing the logic tables and evaluating them in simulation has been implemented and validated. Additional research is required to further improve coordination between aircraft and the handling of multiple simultaneous threats. Additional analysis is required to ensure interoperability with existing TCAS systems. Before committing to a particular encounter model or cost parameters, it is important to ensure that there is consensus within the community that the model captures the features of real encounters and that the cost parameters adequately balance performance considerations.

This new approach is a significant departure from how people have thought about collision avoidance in the past. The idea of computer optimization of decision logic using probabilistic models will likely become increasingly important as the air traffic system becomes more complex. Such an approach will be useful in the design of airborne and ground-based sense-and-avoid systems for unmanned aircraft as well as other decision support tools for air traffic management.

Acknowledgments

The authors greatly appreciate the support of Neal Suchy from the TCAS Program Office at the Federal Aviation Administration. This work has benefited from discussions with David Spencer and Ann Drumm from MIT Lincoln Laboratory.

References

[1] RTCA, “Minimum operational performance standards for Traffic Alert and Collision Avoidance System II (TCAS II),” DO-185B, RTCA, Inc., Washington, D.C., Jun. 2008.

[2] M. J. Kochenderfer and J. P. Chryssanthacopoulos, “Robust airborne collision avoidance through dynamic programming,” Massachusetts Institute of Technology, Lincoln Laboratory, Project Report ATC-371, 2011. [Online]. Available: http://www.ll.mit.edu/mission/aviation/publications/publication-files/atc-reports/Kochenderfer_2011_ATC-371_WW-21458.pdf

[3] M. J. Kochenderfer, J. P. Chryssanthacopoulos, L. P. Kaelbling, and T. Lozano-Perez, “Model-based optimization of airborne collision avoidance logic,” Massachusetts Institute of Technology, Lincoln Laboratory, Project Report ATC-360, 2010. [Online]. Available: http://www.ll.mit.edu/mission/aviation/publications/publication-files/atc-reports/Kochenderfer_2010_ATC-360_WW-18658.pdf

[4] S. Temizer, M. J. Kochenderfer, L. P. Kaelbling, T. Lozano-Pérez, and J. K. Kuchar, “Collision avoidance for unmanned aircraft using Markov decision processes,” in *AIAA Guidance, Navigation, and Control Conference*, Toronto, Canada, 2010.

[5] T. B. Wolf and M. J. Kochenderfer, “Aircraft collision avoidance using Monte Carlo real-time belief space search,” *Journal of Intelligent and Robotic Systems*, 2011, in press.

[6] M. J. Kochenderfer, L. P. Espindle, J. K. Kuchar, and J. D. Griffith, “Correlated encounter model for cooperative aircraft in the national airspace system,” Massachusetts Institute of Technology, Lincoln Laboratory, Project Report ATC-344, 2008. [Online]. Available: http://www.ll.mit.edu/mission/aviation/publications/publication-files/atc-reports/Kochenderfer_2008_ATC-344_WW-18099.pdf

[7] M. J. Kochenderfer, M. W. M. Edwards, L. P. Espindle, J. K. Kuchar, and J. D. Griffith, “Airspace encounter models for estimating collision risk,” *Journal of Guidance, Control, and Dynamics*, vol. 33, no. 2, pp. 487–499, 2010.

[8] M. J. Kochenderfer, L. P. Espindle, M. W. M. Edwards, J. K. Kuchar, and J. D. Griffith, “Airspace encounter models for conventional and unconventional aircraft,” in *Eighth USA/Europe Air Traffic Management Research and Development Seminar*, Napa, Calif., 2009. [Online]. Available: http://atm2003.eurocontrol.fr/8th-seminar-united-states-june-2009/papers/paper_012

[9] RTCA, “Safety analysis of proposed change to TCAS RA reversal logic,” DO-298, RTCA, Inc., Washington, D.C., Nov. 2005.

[10] S. M. LaValle, *Planning Algorithms*. Cambridge University Press, 2006.

[11] M. J. Kochenderfer and J. P. Chryssanthacopoulos, “Partially-controlled Markov decision processes for collision avoidance systems,” in *International Conference on Agents and Artificial Intelligence*, Rome, Italy, 2011.

[12] International Civil Aviation Organization, “Surveillance, radar and collision avoidance,” in *International Standards and Recommended Practices*, 4th ed., Jul. 2007, vol. IV, annex 10.

[13] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*. New York: Wiley, 2001.

[14] J. P. Chryssanthacopoulos and M. J. Kochenderfer, “Accounting for state uncertainty in collision avoidance,” *Journal of Guidance, Control, and Dynamics*, 2011, in press.

[15] S. J. Julier and J. K. Uhlmann, “Unscented filtering and nonlinear estimation,” *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–422, Mar. 2004.

[16] J. K. Kuchar, “Methodology for alerting-system performance evaluation,” *Journal of Guidance, Control, and Dynamics*, vol. 2, no. 2, pp. 438–444, Mar.–Apr. 1996.

[17] RTCA, “Software considerations in airborne systems and equipment certification,” DO-178B, RTCA, Inc., Washington, D.C., Dec. 1992.

[18] ——, “Standards for processing aeronautical data,” DO-200A, RTCA, Inc., Washington, D.C., Sep. 1998.

[19] J. K. Kuchar and L. C. Yang, “A review of conflict detection and resolution modeling methods,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 1, no. 4, pp. 179–189, Dec. 2000.

[20] R. Chamlou, “Future airborne collision avoidance: Design principles, analysis plan and algorithm development,” in *Digital Avionics Systems Conference*, Orlando, FL, Oct. 2009, pp. 6.E.2–1–6.E.2–17.

[21] J. Jansson and F. Gustafsson, “A framework and automotive application of collision avoidance decision making,” *Automatica*, vol. 44, pp. 2347–2351, 2008.

[22] B. D. Carpenter and J. K. Kuchar, “Probability-based collision alerting logic for closely-spaced parallel approach,” in *AIAA Aerospace Sciences Meeting*, Reno, NV, Jan. 1997.

[23] L. C. Yang and J. K. Kuchar, “Prototype conflict alerting system for free flight,” *Journal of Guidance, Control, and Dynamics*, vol. 20, no. 4, pp. 768–773, Jul.–Aug. 1997.

[24] W. B. Powell, *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. Hoboken, NJ: Wiley, 2007.

Mykel J. Kochenderfer (BS’03—MS’03—PhD’06) is a staff member at MIT Lincoln Laboratory. He received a PhD from the University of Edinburgh in informatics for research on model-based reinforcement learning, and BS and MS degrees in computer science from Stanford University. His current research activities include airspace modeling and aircraft collision avoidance.

James P. Chryssanthacopoulos (BS’08) is an assistant staff member at MIT Lincoln Laboratory. He has a bachelor’s degree in physics from Worcester Polytechnic Institute. His research has focused on development, simulation, and analysis of advanced algorithms for next-generation aircraft collision avoidance.

Roland E. Weibel (BS’02—MS’05—PhD’08) is a staff member at MIT Lincoln Laboratory. His research interests include safety assessment, unmanned aircraft sense and avoid, and air traffic control operations. He received a BS in Aerospace Engineering from the University of Kansas, and MS and PhD degrees in Aeronautics/Astronautics from MIT.