Project Report
ATC-408

Optimized Airborne Collision Avoidance in Mixed Equipage Environments

D.M. Asmar
M.J. Kochenderfer

22 May 2013

Lincoln Laboratory
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LEXINGTON, MASSACHUSETTS

Prepared for the Federal Aviation Administration,
Washington, D.C. 20591

This document is available to the public through
the National Technical Information Service,
Springfield, Virginia 22161

TECHNICAL REPORT STANDARD TITLE PAGE

1. Report No. ATC-408	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle Optimized Airborne Collision Avoidance in Mixed Equipage Environments		5. Report Date 22 May 2013 6. Performing Organization Code
7. Author(s) D.M. Asmar and M.J. Kochenderfer		8. Performing Organization Report No. ATC-408
9. Performing Organization Name and Address MIT Lincoln Laboratory 244 Wood Street Lexington, MA 02420-9108		10. Work Unit No. (TRAIS) 11. Contract or Grant No. FA8721-05-C-0002
12. Sponsoring Agency Name and Address Department of Transportation Federal Aviation Administration 800 Independence Ave., S.W. Washington, DC 20591		13. Type of Report and Period Covered Project Report 14. Sponsoring Agency Code

15. Supplementary Notes

This report is based on studies performed at Lincoln Laboratory, a federally funded research and development center operated by Massachusetts Institute of Technology, under Air Force Contract FA8721-05-C-0002.

16. Abstract

Developing robust collision avoidance logic that reliably prevents collision without excessive alerting is challenging due to sensor error and uncertainty in the future paths of the aircraft. Over the past few years, research has focused on the use of a computational method known as dynamic programming for producing an optimized decision logic for airborne collision avoidance. This report focuses on recent research on coordination, interoperability, and multiple-threat encounters. The methodology presented in this report results in logic that is safer and performs better than legacy TCAS. Modeling and simulation indicate that the proposed methodology can bring significant benefit to the current airspace and can support the need for safe, non-disruptive collision protection as the airspace continues to evolve.

17. Key Words	18. Distribution Statement This document is available to the public through the National Technical Information Service, Springfield, VA 22161.

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	60	

FORM DOT F 1700.7 (8-72) Reproduction of completed page authorized

This page intentionally left blank.

EXECUTIVE SUMMARY

The Traffic Alert and Collision Avoidance System (TCAS), currently mandated on all large transport and cargo aircraft, has been shown to significantly reduce the risk of midair collision. TCAS uses an on-board surveillance system to monitor the local air traffic. The surveillance information is used by the threat resolution logic to determine whether to alert the pilots of a potential collision. If necessary, TCAS will issue a resolution advisory to the pilots to climb or descend at a particular rate to prevent a collision.

Developing robust collision avoidance logic that reliably prevents collision without excessive alerting is challenging due to sensor error and uncertainty in the future paths of the aircraft. The current TCAS logic was the result of decades of development and involved the careful engineering of many heuristic rules. The complexity of the logic makes it difficult to revise to accommodate the evolution of the airspace and to support the introduction of new surveillance technologies and procedures.

Over the past few years, research has focused on the use of a computational method known as dynamic programming for producing an optimized decision logic for airborne collision avoidance. There have been a series of technical reports, conference papers, and journal articles summarizing the research up to this point, but they have primarily focused on two-aircraft encounters with only one aircraft equipped with a collision avoidance system.

This report focuses on recent research on coordination, interoperability, and multiple-threat encounters. In situations where an aircraft encounters another aircraft with a collision avoidance system, it is important that the resolution advisories provided to the pilots be coordinated so that both aircraft are not instructed to maneuver in the same direction. Interoperability is a related consideration since new collision avoidance systems will be occupying the same airspace as legacy systems. Resolving encounters with multiple intruders will become increasingly important as the airspace becomes more dense, but poses some computational challenges addressed in this report.

The methodology presented in this report results in logic that is safer and performs better than legacy TCAS. To assess the performance of the system, this report uses U.S. airspace encounter models. The results indicate that the proposed methodology can bring significant benefit to the current airspace and can support the need for safe, non-disruptive collision protection as the airspace continues to evolve.

This page intentionally left blank.

ACKNOWLEDGMENTS

This report is the result of research and development sponsored by the TCAS Program Office at the FAA. The authors appreciate the assistance provided by the TCAS Program Manager, Neal Suchy. This research has greatly benefited from prior work by James P. Chryssanthacopoulos and discussions with and assistance from Ann Drumm, Matthew Edwards, Alex Eyler, Jessica Holland, Wes Olson, and Dave Spencer.

This page intentionally left blank.

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

LIST OF FIGURES

This page intentionally left blank.

LIST OF TABLES

This page intentionally left blank.

1 INTRODUCTION

The Traffic Alert and Collision Avoidance System (TCAS), currently mandated on all large transport and cargo aircraft, has been shown to significantly reduce the risk of mid-air collision. TCAS uses an on-board surveillance system to monitor the local air traffic. The surveillance information is then provided to the threat resolution logic to determine whether to alert the pilots of a potential collision. TCAS will issue a resolution advisory to the pilots to climb or descend at a particular rate to prevent a collision.

Developing robust collision avoidance logic that reliably prevents collision without excessive alerting is challenging due to sensor error and the uncertain future paths of the aircraft. The current TCAS logic was the result of many years of development and involved the careful engineering of many heuristic rules. Unfortunately, due to the complexity of the logic, it is difficult to revise to accommodate the evolution of the airspace and the introduction of new surveillance technologies and procedures.

Over the past few years, research has focused on the use of a computational method known as dynamic programming for producing an optimized decision logic for airborne collision avoidance. This research has resulted in the establishment of the Airborne Collision Avoidance System X (ACAS X) program and is being targeted to become the next international standard for collision avoidance for both manned and unmanned aircraft. Research up to this point has primarily focused on scenarios with a single unequipped intruder.

This report focuses on recent research on coordination, interoperability, and multiple threat encounters. In situations where an aircraft encounters another aircraft with a collision avoidance system, it is important that the resolution advisories provided to the pilots be coordinated so that both aircraft are not instructed to maneuver in the same direction. Interoperability is a related consideration since new collision avoidance systems will be occupying the same airspace as legacy systems. Resolving encounters with multiple intruders will become increasingly important as the airspace becomes more dense, but poses some computational challenges addressed in this report.

1.1 ARCHITECTURE

ACAS X is composed of a surveillance and tracking module (STM), a threat resolution module (TRM), and a display. The STM is responsible for detecting and tracking intruders and providing the information to the TRM. The TRM uses this information to determine whether to generate an alert and what advisory to display. If an advisory is generated, information about the advisory is shared with other aircraft over a crosslink to ensure compatible advisories are generated. The display provides traffic alerts (TAs) to notify the pilots of a potential threat and resolution advisories (RAs) to instruct the pilots to climb or descend to avoid a collision.

Although the surveillance system in legacy TCAS has been tailored specifically for beacon-based surveillance systems with particular error characteristics, the interface between the STM and TRM in ACAS X has been defined generically to allow for other surveillance systems, such as those based on broadcast satellite navigation information, radar, and electro-optical/infrared sensors. In

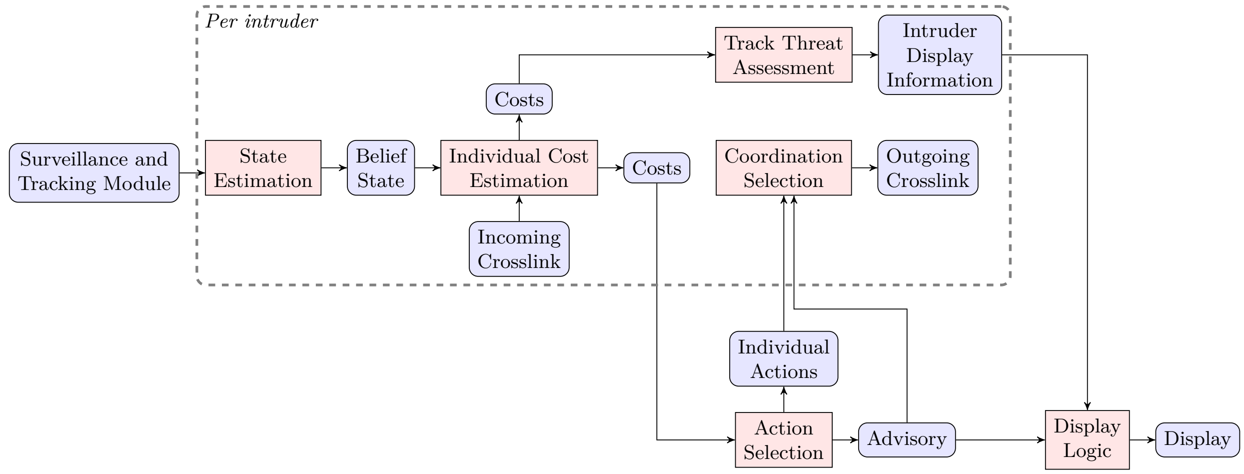


Figure 1. ACAS X architecture.

contrast with TCAS, which uses only a single estimate of the most likely state, ACAS X takes into account state uncertainty. It represents this uncertainty as a set of weighted samples. These weighted samples are passed to the TRM.

The focus of this report is on the TRM since it represents a major departure from legacy TCAS. Figure 1 provides an overview of the processing involved. The first step involves estimating a belief state based on the weighted position and velocity samples provided by the STM. The belief state is also represented by a set of weighted samples, but these samples represent a joint distribution over the time to potential collision, the relative altitude of the intruder, the intruder vertical rate, the own vertical rate, and the state of the RA response.

A separate belief state is associated with each intruder. For each intruder, the TRM estimates the costs of the available actions, such as no advisory, climb 1500, and descend 1500. These costs are determined through table lookup and interpolation. These costs are then used to determine whether to generate a TA. The action costs for all the intruders are used to select both the action that is appropriate for the individual intruders in isolation as well as the global action that is best overall. The individual actions are used in the coordination process with other aircraft. The global action is what is used to drive the display.

1.2 COORDINATION MECHANISM

This report is primarily concerned with handing scenarios with mixed equipages. TCAS has only been mandated on aircraft above a certain weight or seat threshold, and most general aviation aircraft are not equipped with any form of collision avoidance system. In certain categories of airspace, an aircraft carrying ACAS X is most likely to encounter aircraft equipped only with a Mode C or Mode S transponder. Much of the prior work on ACAS X has focused on encounters with these kinds of aircraft. When ACAS X encounters another aircraft with a collision avoidance system, it will need to coordinate maneuvers.

The coordination mechanism is exactly the same as with legacy TCAS when both aircraft are equipped with Mode S interrogators. If the system determines that a climb is necessary, it will transmit a do not climb coordination interrogation to the intruder. The intruder responds with a coordination reply that confirms agreement. Typically, the first aircraft to select a direction (often referred to as "sense") is granted its choice. However, if both aircraft happen to select conflicting senses, then the 24 bit Mode S address is used as a tiebreaker, and the aircraft with the lower address prevails. The other aircraft will be required to switch sense. Coordination interrogations continue to be sent once per second for the duration of the RA. Upon termination, the aircraft sends an additional coordination interrogation to notify the intruder.

The Mode S coordination process is quite robust, and no coordination failures are known to have occurred in TCAS operation. The Mode S link has a built-in parity protection feature, and the coordination information is protected by an additional parity field within the coordination message. In addition, coordination interrogations are transmitted at full power and are repeated 6–12 times (if necessary) over a 100 ms interval to ensure extremely high reception probability.

A flavor of ACAS X is being designed for general aviation aircraft [1], which are likely not equipped with relatively expensive Mode S interrogators. Hence, these aircraft cannot rely upon the same mechanism as legacy TCAS. If the aircraft are equipped with Automatic Dependent Surveillance-Broadcast (ADS-B), then a purely passive coordination technique known as Active Coordination Emulation (ACE) can be used. There are some limitations to ACE and it does not provide the same level of reliability as Mode S coordination required of large commercial aircraft. The details of this passive coordination mechanism will not be discussed in this report, but it is based on the same concept of sending coordination messages between equipped aircraft.

Encounters involving multiple simultaneous threats are relatively rare in the airspace, but their frequency is likely to increase as the airspace becomes more dense. In multiple threat encounters, it is possible that coordination messages of different senses should be sent to different aircraft. Messages with different senses can arise in "sandwich" geometries, where one aircraft is sandwiched between two other aircraft. Depending on the vertical rates involved, the aircraft in the middle may want to instruct the aircraft above it to not descend and the aircraft below it to not climb. The middle aircraft may generate the level-off advisory.

1.3 RECENT ADVANCES

Besides carefully studying and enhancing the coordination and multithreat logic in ACAS X, there have been a number of important enhancements to ACAS X since the publication of the last technical report [2]. The intruder vertical tracker has been modified to explicitly take into account quantization. Especially for intruders with 100 ft altitude encoding, the improved tracker can lead to much better performance. A number of online and offline costs have been added to account for various considerations that have emerged from the operational assessment and tuning process. These changes are summarized in this report.

1.4 METRICS

The performance of a collision avoidance system (CAS) can be separated into safety performance and operational performance. The primary objective of a collision avoidance system is to increase safety. However, a collision avoidance system should not interfere with normal, safe flight operations. Excessive alerts and changes in the advisories affect the efficiency of a collision avoidance system. The metrics used to evaluate collision avoidance systems in this report include:

- *Near Mid-Air Collision.* A near mid-air collision (NMAC) occurs when two aircraft come within 500 ft horizontally and 100 ft vertically. NMACs fall into two categories: induced and unresolved. An induced NMAC is one that occurs with a CAS but does not occur without a CAS. An unresolved NMAC is one that occurs both with and without a CAS.
- *Risk Ratio.* The risk ratio is defined as the probability of an NMAC given the aircraft is equipped with a CAS divided by the probability of an NMAC given no aircraft is equipped:

$$\text{Risk Ratio} = \frac{\Pr(\text{NMAC} \mid \text{aircraft with CAS})}{\Pr(\text{NMAC} \mid \text{aircraft without CAS})}. \tag{1}$$

- *Alert.* An alert is defined as when a CAS issues an advisory during an encounter.
- *Strengthening.* A strengthening is any change in a commanded vertical rate to a greater vertical rate in the same direction of the previous advisory.
- *Reversal.* A reversal is any advisory that changes the sense of a previous advisory (e.g., climb to descend).
- *Restart.* A restart is when a CAS reissues an advisory within 20 s.

1.5 OVERVIEW

Section 2 provides a brief overview of Markov decision processes as a way to model collision avoidance problems. This section discusses both the online and offline costs used in the logic. Optimal policies are compared to TCAS and a few example encounters are presented.

Section 3 discusses coordinated encounters with a single intruder. Solution methods are presented and a coordination scheme similar to TCAS, called forced cooperation, is explored. This section discusses online costs used to improve forced cooperation. Simulation results are presented for situations with both ACAS X and TCAS. Robustness to delayed pilot responses and non-compliant intruders is analyzed.

Section 4 generalizes the logic to handle encounters with multiple intruder aircraft. Different approximation methods are examined that treat intruders independently. Utility fusion is explored as a potential solution method. This section discusses how to include multithreat level-offs as an action without increasing the size of the action space. Simulation results are presented for various equipage scenarios.

Section 5 concludes the report, summarizing the major contributions, and outlines directions for future work.

2 OPTIMIZATION PROCESS

Markov decision processes (MDPs) have been well studied since the 1950s and have been applied to a wide variety of problems [3, 4, 5]. MDPs require that the dynamic model be Markovian, meaning that the probability of transitioning to state s' depends only on the current state s and action a. The probability of a state transition from s to s' by action a is denoted $T(s, a, s')$. The function T is often called the state-transition function. So long as sufficient information about the problem can be encoded in the state, the Markov assumption typically holds. The set of possible states is denoted $\mathcal{S}$ and the set of possible actions is denoted $\mathcal{A}$.

Solving an MDP involves searching for a strategy for choosing actions, also called a policy, that maximizes a performance metric. Under certain assumptions regarding the structure of the performance metric, it is sufficient to only consider policies that deterministically depend only on the current state without losing optimality [6]. Given a policy π, the action to execute from state s is denoted $\pi(s)$.

There are several common performance metrics, also called optimality criteria, typically used with MDPs. One common metric is the expected sum of instantaneous reward up to some fixed horizon. The optimal policy is the one that maximizes this metric. Because collision avoidance typically involves avoiding particular events, such as collision and alerting, it is more convenient to define the metric in terms of positive costs instead of negative rewards. The objective, then, is to minimize the expected sum of instantaneous costs. In this report, the word "cost" is used to mean sum of the instantaneous costs. In other texts, the expected sum of instantaneous costs is sometimes called "cost-to-go."

2.1 ACTION SPACE

The current version of TCAS issues advisories to the pilot through an aural annunciation, such as "climb, climb," and through a visual display. The visual display varies, but it is typically implemented on a vertical speed indicator, a vertical speed tape, or a pitch cue on the primary flight display. The set of advisories issued by TCAS can be interpreted as target vertical rate ranges. If the current vertical rate is outside the target vertical rate range, the pilot should maneuver to come within the required range. If the current vertical rate is within the target range, a corrective maneuver is not required, but the pilot should be careful not to maneuver outside the range.

The current version of TCAS assumes that the pilot will respond to initial advisories with a 1/4 g maneuver to come within the target vertical range. Depending on how the encounter evolves, TCAS has the option to strengthen the original advisory or reverse the sense of the advisory. Once an advisory has been strengthened, TCAS may later weaken the advisory. All subsequent advisories are assumed to be responded to with a stronger 1/3 g maneuver.

The set of advisories used in this report for ACAS X is summarized in Table 1. The advisories are similar to those in previous reports, with the addition of DNC, DND, and the maintain advisories [2]. In the table, COC stands for "clear of conflict," which means that no advisory has been issued or that there is no longer a threat. DNC and DND stand for "do not climb" and "do not descend,"

TABLE 1

Advisory Set

Name	Vertical rate (ft/min) Minimum	Vertical rate (ft/min) Maximum	Strength (g)	Available from
COC	$-\infty$	∞	0	All
DNC	$-\infty$	0	1/4	COC, DES1500, SDES1500, SDES2500, MDESrate
DND	0	∞	1/4	COC, CL1500, SCL1500, SCL2500, MCLrate
MDES	$-\infty$	$\dot{h}_{\text{curr}}$	1/4	COC, DND, CL1500, SCL1500, SCL2500
MCL	$\dot{h}_{\text{curr}}$	∞	1/4	COC, DNC, DES1500, SDES1500, SDES2500
DES1500	$-\infty$	-1500	1/4	COC
CL1500	1500	∞	1/4	COC
SDES1500	$-\infty$	-1500	1/3	DNC, DND, MCL, CL1500, SCL1500, SCL2500
SCL1500	1500	∞	1/3	DNC, DND, MDES, DES1500, SDES1500, SDES2500
SDES2500	$-\infty$	-2500	1/3	MDES, DES1500, SDES1500
SCL2500	2500	∞	1/3	MCL, CL1500, SCL1500

respectively. The sense of all other advisories are labeled either CL or DES, for either climb or descend, respectively. The prefix “M” stands for maintain. The maintain advisories are issued only when the magnitude of the current vertical rate of the own aircraft ($\dot{h}_{\text{curr}}$) is greater than 1500 ft/min. The maintain advisory is issued with the current rate as the upper or lower bound on the commanded rate. The prefix “S” indicates that a stronger response is assumed. Only the minimum and maximum rates are typically displayed to the pilot, and not the “strength” of the response, so the eleven advisories in the table only correspond to nine different advisories to be displayed to the pilot. However, it is useful to distinguish the advisories according to the assumed strength of the maneuver when developing the MDP model.

Table 1 also indicates the availability of each advisory given the current advisory on display. For example, COC can be issued at any time. However, because DES1500 and CL1500 are initial advisories, they can only be issued if COC is on display to the pilot. The advisory SDES1500 can be issued following DND, MCL, CL1500, SCL1500, and SCL2500, in which case it acts as a reversal, or following DNC, in which case it acts as a strengthening. Because SDES1500 is a subsequent advisory, it cannot be issued following COC. It also cannot be issued following DES1500 because they are fundamentally the same advisory, differing only in strength. The allowed transitions are modeled after those of TCAS.

The advisories in Table 1 are a subset of the advisories available in the current version of TCAS. Although it captures most of the advisories issued by TCAS, it does not contain certain rate limit preventive advisories. One advisory that is incorporated into the set for mulithreat purposes is a multithreat level-off (MTLO). The MTLO is not included in the action state but is still allowed to be issued, as discussed in Section 4.3. Although this report does not incorporate all the advisories, it is straightforward to include them if deemed necessary. The computational and storage requirements of the MDP approach scale linearly with the addition of new actions.

When the collision avoidance system issues an initial advisory, the pilot responds after a 5 s delay with a 1/4 g maneuver to achieve the required vertical rate. Following the initial advisory,

TABLE 2

Discretization Scheme

Variable	Minimum	Maximum	Number of values
h	−4000 ft	4000 ft	31
$\dot{h}_0$	−10000 ft/min	10000 ft/min	25
$\dot{h}_1$	−10000 ft/min	10000 ft/min	25
τ	0 s	40 s	41
s_{RA}	N/A	N/A	19

the system may either terminate the advisory, strengthen the advisory, weaken the advisory, or reverse the advisory. For subsequent advisories, the pilot maneuvers at 1/3 g after a 3 s delay [7].

2.2 STATE SPACE

The state is represented using five variables.

- h: altitude of the intruder relative to the own aircraft
- $\dot{h}_0$: vertical rate of the own aircraft
- $\dot{h}_1$: vertical rate of the intruder aircraft
- τ: time to potential NMAC
- s_{RA}: the state of the resolution advisory

The state space is discretized using a multidimensional grid. This report uses the discretization shown in Table 2, which results in 15 million grid vertices that correspond to discrete states. The discretization can be made finer to improve the quality of the discrete model approximation, but it would be at the expense of additional computation and storage. Prior studies have found this level of discretization acceptable [2].

2.3 DYNAMIC MODEL

The dynamics of the aircraft involved in the encounter are governed by sequences of accelerations. These accelerations are used to update the vertical rates of the aircraft and, consequently, their positions. The maximum vertical rate of both aircraft is assumed to be ±10000 ft/min, although this is easily changed at the expense of a larger state space. In a real system, the limits can be adjusted to meet the performance constraints for the particular aircraft. The dynamics are transformed into a discrete state MDP using the multilinear-interpolation and sigma-point sampling scheme described in an earlier technical report [2].

TABLE 3

Event Costs

Description	Cost
NMAC	1
Alert	0.0025
Reversal	0.008
Strengthening	0.009
Weakening	0.001
Change in $\dot{h}$	3×10^{-5}
Corrective advisories when relative altitude is greater than 500 ft	0.01
Corrective advisories when relative altitude is greater than 1000 ft	0.03
Maintain advisories less than 1500 ft/min	1
Switching advisories	0.001
Crossing encounters when relative altitude is greater than 500 ft	0.01
Prohibited transitions	1
Issue COC	-1×10^{-9}
Issue DNC	-0.0001
Issue DND	-0.0001
Issue Maintain	-0.0004
Preventive advisories during crossing scenarios	1

2.4 COST FUNCTION

The cost function used in the MDP only depends on the current state and action. The cost of executing action a from state s is denoted $C(s, a)$ and costs of various events are summarized in Table 3. The costs were chosen after several iterations of tuning the logic.

The primary safety metric for evaluating TCAS historically has been Pr(NMAC), and so NMACs are assigned high cost. The small negative costs associated with COC is awarded at every time step the system is not alerting to provide some incentive to discontinue alerting after an encounter has been resolved. The negative costs for the advisories DND, DNC, and maintain are to provide incentives to issue less aggressive advisories.

This report does not focus on varying or adding offline costs to address safety or operational considerations. If the state space does not require expanding, the computation required to construct the expected cost table grows linearly with the number of cost factors and the storage required remains constant. Online execution of the logic also remains constant.

2.5 DYNAMIC PROGRAMMING

Due to the assumptions made by the dynamic model and the cost function outlined, solutions may be found efficiently using a computational technique called dynamic programming (DP). DP can be used to compute the expected cost from every discrete state when following an optimal K-step horizon policy π_K^*. The optimal expected cost function J_K can be used to determine the optimal policy. Computing J_K is done using an iterative process, starting with initializing the function $J_0(s) = 0$ for all states s. Given the function J_{k-1}, the function J_k is as follows:

$$J_k(s) = \min_a \left[C(s,a) + \sum_{s'} T(s,a,s') J_{k-1}(s') \right]. \tag{2}$$

The state-action cost $J_K(s,a)$ with a K-step horizon is given by

$$J_K(s,a) = \min_a \left[C(s,a) + \sum_{s'} T(s,a,s') J_{K-1}(s') \right]. \tag{3}$$

An optimal K-step policy satisfies

$$\pi_K^*(s) = \arg\min_a J_K(s,a). \tag{4}$$

In general, there may be multiple actions that satisfy the equality above from a particular state. In this work, it is assumed that ties are broken according to some ordering over the set of available actions, resulting in a unique optimal policy.

Because the actions are categorical, one cannot simply interpolate the optimal policy directly to determine the optimal action from non-discrete states. Instead, one can interpolate the state-action function and from that determine the optimal action. Storing the optimal state-action function requires $|\mathcal{S}| \times |\mathcal{A}|$ entries, whereas storing the optimal policy requires only $|\mathcal{S}|$ entries. Because the number of actions available from a particular state is typically small in the collision avoidance domain, the storage of the state-action table can be compressed. The implementation used for the experiments in this report stores the state-action costs only for valid state-action pairs and uses an index file for fast lookups.

2.6 ENTRY DISTRIBUTION

The state variable τ, which represents the time to potential NMAC, is a quantity that is based on a future event. Due to the fact that the aircraft can maneuver horizontally, it is impossible to know this value with certainty. One approach for accounting for this uncertainty is to model the horizontal dynamics directly in the MDP. Directly modeling the horizontal dynamics would require adding several variables into the state space, which is not feasible given current memory and processing constraints. The approach taken in ACAS X is to apply dynamic programming assuming that τ is known and decrements deterministically. During execution, however, the system infers a distribution over τ and chooses the action with the lowest expected cost [8, 9].

2.7 ONLINE COSTS

The cost function was designed to be based on only the current state and action, $C(s,a)$. Therefore, any cost that requires "memory" would have to be implemented by introducing new state variables. Online costs were introduced to penalize actions in real-time without introducing new state variables. During execution, the expected costs calculated from the offline optimization

TABLE 4

Online Cost Parameters

Description	Cost / Value
Altitude Inhibit Cost	
All advisories lower hysteresis bound	900 ft
All advisories upper hysteresis bound	1100 ft
All advisories cost	∞
SDES2500, MDESrate (rate > 2500 ft/min) lower hysteresis bound	1450 ft
SDES2500, MDESrate (rate > 2500 ft/min) upper hysteresis bound	1650 ft
SDES2500, MDESrate (rate > 2500 ft/min) cost	∞
DES1500, SDES1500, MDESrate (rate > 1500 ft/min) lower hysteresis bound	1000 ft
DES1500, SDES1500, MDESrate (rate > 1500 ft/min) upper hysteresis bound	1200 ft
DES1500, SDES1500, MDESrate (rate > 1500 ft/min) cost	∞
DNC lower hysteresis bound	1050 ft
DNC upper hysteresis bound	1150 ft
DNC cost	0.005
Advisory Switch Cost	
Time the online cost is active after an advisory is issued	10 s
Switch to a reversal	0.05
Switch to COC	0.05
Switch to any advisory except for COC or a reversal	0.025
Advisory Restart Cost	
Time the online cost is active after an advisory is terminated	10 s
Restart cost	0.05
Initialization Cost	
Time to incur online cost starting from the start of the track	3 s
Initialization cost for all advisories except COC	∞
Switch Within Altitude Threshold	
Altitude threshold	200 ft
Cost to switch to DNC or DND after previous non-preventive advisory	∞
Time the online cost is active after a differing VRC	10 s
Switching to COC within altitude threshold and time limit	∞
Master aircraft switching to a reversal within altitude threshold and time limit	∞
Multithreat COC Incentive	
Cost for COC in special multithreat situations	-0.0025
Forced Cooperation Cost	
Slave cost for non-cooperative advisory	∞
Master cost for non-cooperative first advisory	∞
Master cost for non-cooperative subsequent advisory	0.025
Restrict COC Due To Reversal	
COC cost due to a forced reversal by a master aircraft	∞

are added to the online cost associated with that action. Various online costs are discussed in this section. Table 4 summarizes the parameters and costs.

- *Altitude Inhibit Cost.* The altitude inhibit cost penalizes advisories below certain altitudes. Hysteresis is implemented to prevent chatter. Advisories are prohibited if the aircraft starts below the upper threshold and remains prohibited until it crosses above that threshold. From the other direction, advisories are allowed until the aircraft flies below the lower threshold.

- *Advisory Switch Cost.* The advisory switch cost penalizes actions that represent a change from the current advisory within a certain number of seconds after it is issued. Switching to different advisories such as opposite sense advisories are penalized differently. This online cost requires memory of the previous advisory issued and the duration it has been active.

- *Advisory Restart Cost.* The advisory restart cost penalizes advisories whenever a prior advisory has been terminated for fewer than a certain number of processing cycles. This cost requires memory of whether an advisory has been issued, whether an advisory has terminated, and the amount of time for which an advisory has been terminated.

- *Initialization Cost.* The initialization cost prohibits advisories from being issued for some number of processing cycles. The initialization period allows for the trackers to stabilize before issuing an advisory.

- *Switch within Altitude Threshold.* The switch within altitude threshold cost is similar to the advisory switch cost. However, this cost is only nonzero when the intruder is within a certain altitude of the own aircraft and an advisory has already been issued. This online cost improves performance in coordinated encounters. First, this cost penalizes switching to preventive advisories (DNC or DND) if the intruder aircraft is within the altitude threshold. Second, it penalizes a master aircraft from switching its advisory to a reversal or to COC within a certain number of processing cycles after receiving a coordination message if the intruder is within the altitude threshold. The latter part of this online cost is discussed more in Section 3.3.2.

- *Multithreat COC Incentive.* The multithreat COC incentive cost provides a negative cost to COC in certain multithreat situations. Knowledge of the previous individual action and global action are needed. The negative cost for COC is only applied when the system has issued an advisory due to a different intruder and the lowest-cost action with respect to the intruder was COC. This online cost is discussed in more detail with an example encounter in Section 4.5.2.

- *Forced Cooperation Cost.* The forced cooperation cost penalizes actions that are incompatible with the sense of an intruder VRC message.

- *Restrict COC Due to Reversal.* This cost penalizes the transition to clear of conflict when a slave aircraft is forced to reverse due to coordination. The restriction of COC only occurs for the one processing cycle that follows a coordination message received from a master aircraft that forces an opposite sense advisory.

2.8 OPTIMAL POLICY

Since the collision avoidance logic is critical to safety, it is important for humans to understand and anticipate the behavior of the system. Because the logic makes decisions based on values in an expected cost table, which is not directly informative to a human, it is necessary to develop ways to visualize the logic. Visualization is also important in building confidence that the logic produced through computer optimization is sensible.

The policy plots generated in this report are based on simulations with different initial altitudes. The results are discretized into altitude bins for each time step. The action for every track at each time step is deposited in its respective bin. The policy plot is then displayed as the most frequent action for each bin. If there is a bin that no trajectory falls in, no action is displayed. The horizontal axis is time and the vertical axis is the altitude. All simulations were conducted with standard TCAS sensor uncertainty and pilot response [7].

Figure 2 shows a policy plot for ACAS X and TCAS. The encounter is between two aircraft with only the own aircraft equipped. Both aircraft in this scenario are flying level and directly at each other horizontally. The time of closest horizontal approach (TCA) occurs at 40 s. The ACAS X alerting region is much smaller than that of TCAS, delaying alerting by about 5 s. Besides the alerting region, the policies are very similar.

Figure 3 shows a policy plot for both ACAS X and TCAS when the own aircraft has a vertical rate. The encounter is between two aircraft with only the own aircraft being equipped. The intruder aircraft is flying level at 7500 ft and this encounter is head-on. The own aircraft has an initial vertical rate of 500 ft/min. Like the policy in Figure 2 the ACAS X alerting region is much smaller than that of TCAS. Again, besides the alerting region, the policies are very similar. However, the TCAS policy is a little more sloped with the own vertical rate compared to the ACAS X policy. The advisory color that is normally designated for MTLOs is used for vertical rate limits in the TCAS policies. Vertical rate limit advisories are seen at the bottom of the alerting region in the TCAS policy.

2.9 EXAMPLE ENCOUNTERS

Figure 4 is an example of an encounter that is not resolved by TCAS. The alert by TCAS is too late and an NMAC results with 84 ft of vertical separation. ACAS X resolves the encounter with 585 ft of vertical separation. In Figure 5, TCAS issues a reversal and induces an NMAC. After issuing a climb advisory, TCAS does not predict adequate separation will be attained and reverses the encounter before the intruder starts to slow its vertical rate. The result is an NMAC and a vertical separation of 59 ft. ACAS X resolves the encounter by issuing a climb advisory and then strengthens. The resulting vertical separation is 430 ft.

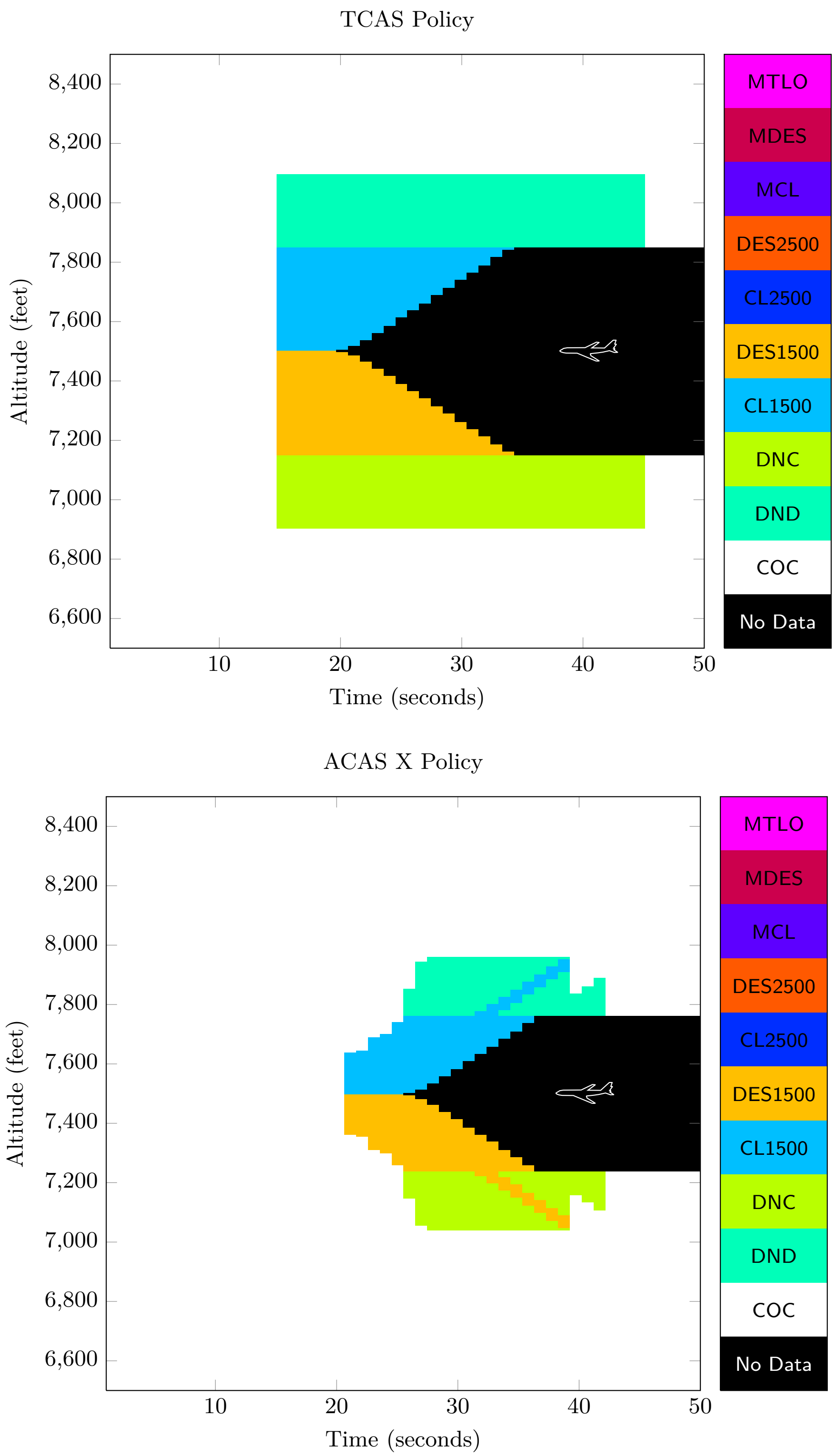


Figure 2. Policy plot of ACAS X and TCAS against an unequipped intruder. The intruder starts at 7500 ft and maintains level flight. The own aircraft is also initially level. TCA occurs at 40 s.

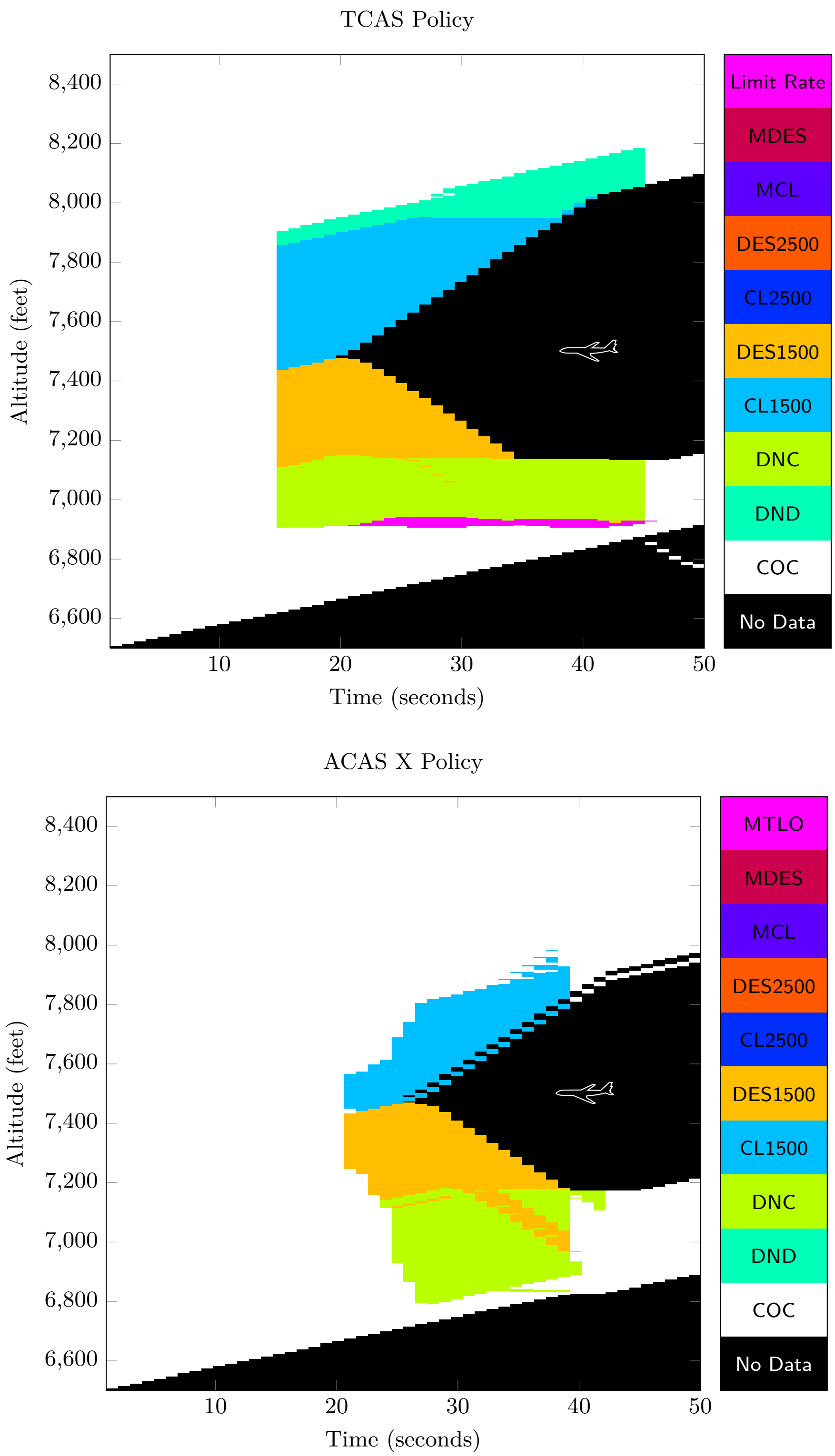
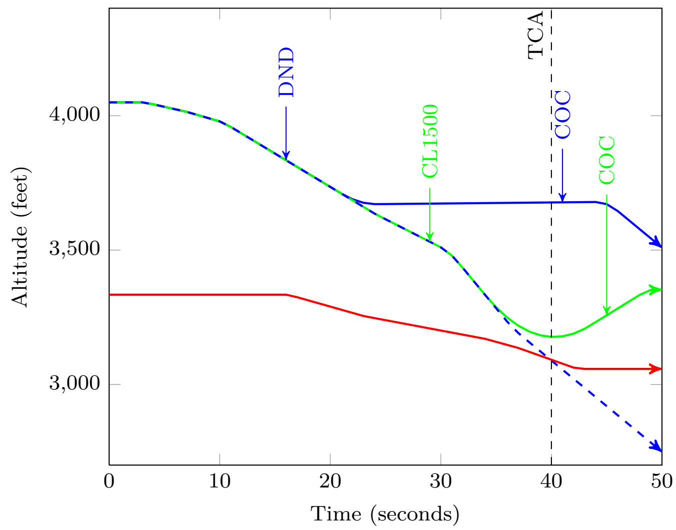
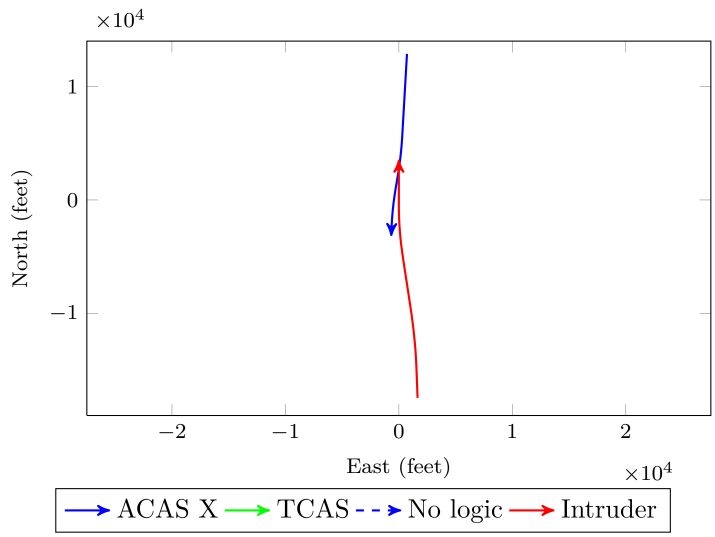


Figure 3. Policy plot of ACAS X and TCAS against an unequipped intruder. The intruder starts at 7500 ft and maintains level flight. The own aircraft initially has a vertical rate of 500 ft/min. TCA occurs at 40 s.

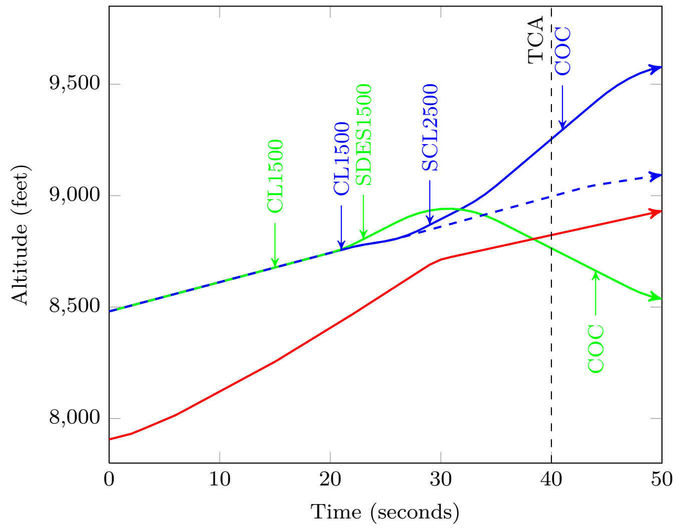


(a) Vertical profile for both ACAS X and TCAS.

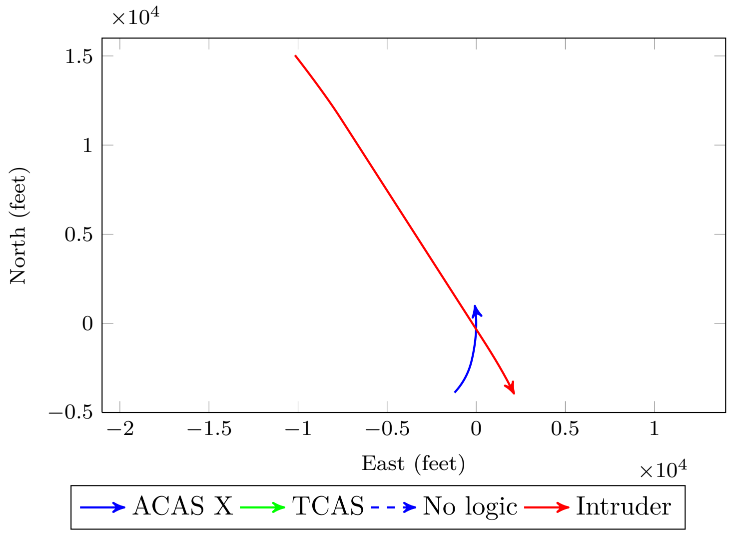


(b) Horizontal profile.

Figure 4. Example of a two aircraft encounter where ACAS X resolves the encounter and TCAS does not. The intruder is unequipped.



(a) Vertical profile for both ACAS X and TCAS.



(b) Horizontal profile.

Figure 5. Example of a two aircraft encounter where ACAS X resolves the encounter and TCAS induces an NMAC by reversing. The intruder is unequipped.

3 COORDINATION

Much of the development of ACAS X up to this point has focused primarily on encounters with a single unequipped intruder. If the intruder is equipped with a collision avoidance system, then safety can be significantly improved. However, the maneuvers recommended by the systems must be coordinated. If both the own aircraft and intruder issue the same advisory, then the likelihood of an NMAC increases significantly. The coordination of advisories requires communication between aircraft. Any next generation collision avoidance logic must use a communication architecture compatible with TCAS.

After selecting an advisory against a particular TCAS-equipped intruder, TCAS transmits a coordination interrogation to the intruder, called a Vertical Resolution Advisory Complement (VRC), through a dedicated communication channel. For example, if TCAS selects a climb advisory against the intruder, it sends a message to the intruder containing a do not climb VRC. The VRC is used by the intruder to select a compatible advisory. When the intruder is no longer deemed a threat, TCAS sends a Cancel Vertical Resolution Advisory Complement (CVC) message canceling the previous VRC message. Also contained in the coordination message sent to the intruder is a parity field, called the Vertical Sense Bits (VSB) field. The intruder checks to see if the VSB field is consistent with the VRC and CVC fields before using the coordination information in the logic.

3.1 POTENTIAL SOLUTION METHODS

Many approaches to coordinated collision avoidance have been proposed in the literature including those based on mixed-integer programs [10] and geometry [11]. These methods result in open-loop plans, which have certain drawbacks compared to closed loop plans that take into account a probabilistic model of the environment [12].

Previous work looked at approximating the problem as a multi-agent MDP (MMDP). An MMDP extends an MDP by allowing for a finite set of agents, each with a finite set of available actions. The joint action space is the Cartesian product of the action spaces of the individual agents. The state space is the set of all joint states, and the transition and cost functions are defined over these joint state and action spaces. In an MMDP, each of the agents can fully identify the true state through the use of its own observations. The benefit of the added complexity of the MMDP was small and was outperformed by simpler schemes [2].

Other approaches to coordination in multi-agent systems involve modeling the problem as an interactive partially observable Markov decision process (I-POMDP) [13, 14] or as a decentralized partially observable Markov decision process (Dec-POMDP) [15, 16, 17]. However, solutions to these categories of problems are computationally intractable in general.

3.2 FORCED COOPERATION

TCAS uses a forced cooperation scheme, which involves restricting the choice of advisories to those compatible with the advisory issued by the other aircraft. Two actions are considered compatible if the target vertical rate rate is is opposite directions. Extending forced cooperation to environments with more than two agents will be discussed in Section 4. The remainder of this section assumes that the own aircraft is equipped with a CAS and there is only one intruder that is also equipped with a CAS.

If both aircraft simultaneously choose to issue incompatible advisories, then the tie needs to be broken. TCAS breaks ties based on the 24 bit International Civil Aviation Organization (ICAO) Mode S address of the aircraft involved in the encounter. The aircraft with the lower Mode S address is designated the master and the other aircraft is the slave. The slave is forced to reverse its advisory to be compatible with the master.

The forced cooperation approach is suboptimal in general, but is suitable for ACAS X for several reasons. This scheme has been used by TCAS for many years, provides robustness against intruder aircraft not following their resolution advisories, and allows the offline development to be focused on encounters with unequipped intruders. Forced cooperation can be implemented entirely online (Section 2.7). The offline optimization would still occur as with an unequipped intruder. Therefore the expected costs are computed assuming that future actions are unimpeded by cooperation restrictions dictated by another aircraft. Although the expected costs may be inaccurate, performance is not expected to be significantly impaired.

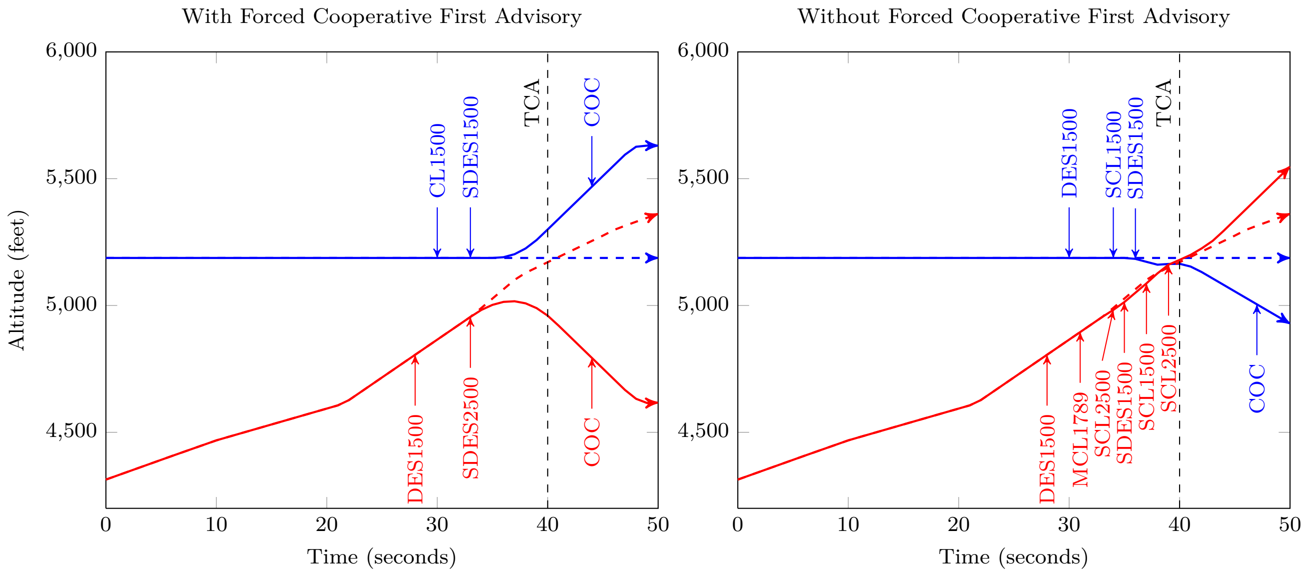
3.3 ONLINE COSTS

Online costs can be added to improve the general forced cooperation scheme. The experiments in this report show that forcing a cooperative first advisory and prohibiting reversals in certain situations significantly increase safety.

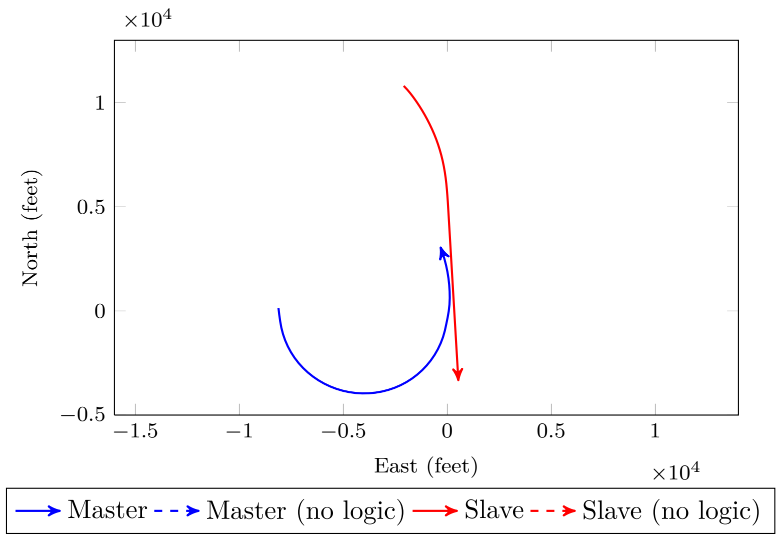
3.3.1 Cooperative First Advisory

Performance can be improved if the master is forced to be compatible with the slave when the slave issues an advisory first. Figure 6 illustrates the benefit of this restriction implemented as an online cost. The left vertical profile plot shows the encounter with the online cost and the right vertical profile plot shows the encounter without the cost. Without the cost, the master's first advisory is the same sense as the slave's, forcing the slave to reverse its advisory at 31 s. The master then reverses after the intruder strengthens its advisory. The intruder is then required to reverse again. The final vertical separation is 15 ft. With the online cost, the master aircraft is forced to be cooperative with the slave since the slave issued an advisory 2 s prior to the master. Both aircraft strengthen their advisories at 33 s and successfully resolve the encounter with a vertical separation of 340 ft.

Forcing a cooperative first advisory restricts the available actions. The difference in policies can be seen in Figure 7 where the own aircraft is flying level directly at the intruder. The intruder aircraft starts at 6800 ft and is climbing at 1000 ft/min. TCA occurs at 40 s. The polices shown are for the master aircraft. One major difference is the reduction in the size of the DES1500 region. When the slave alerted first, the master is then restricted to cooperative advisories. In the region where the DES1500 turned into a COC or CL1500, the intruder alerted first with a down sense advisory, thus forcing a cooperative advisory.



(a) Vertical profile.



(b) Horizontal profile.

Figure 6. Example coordinated encounter where the first advisory is forced to be cooperative.

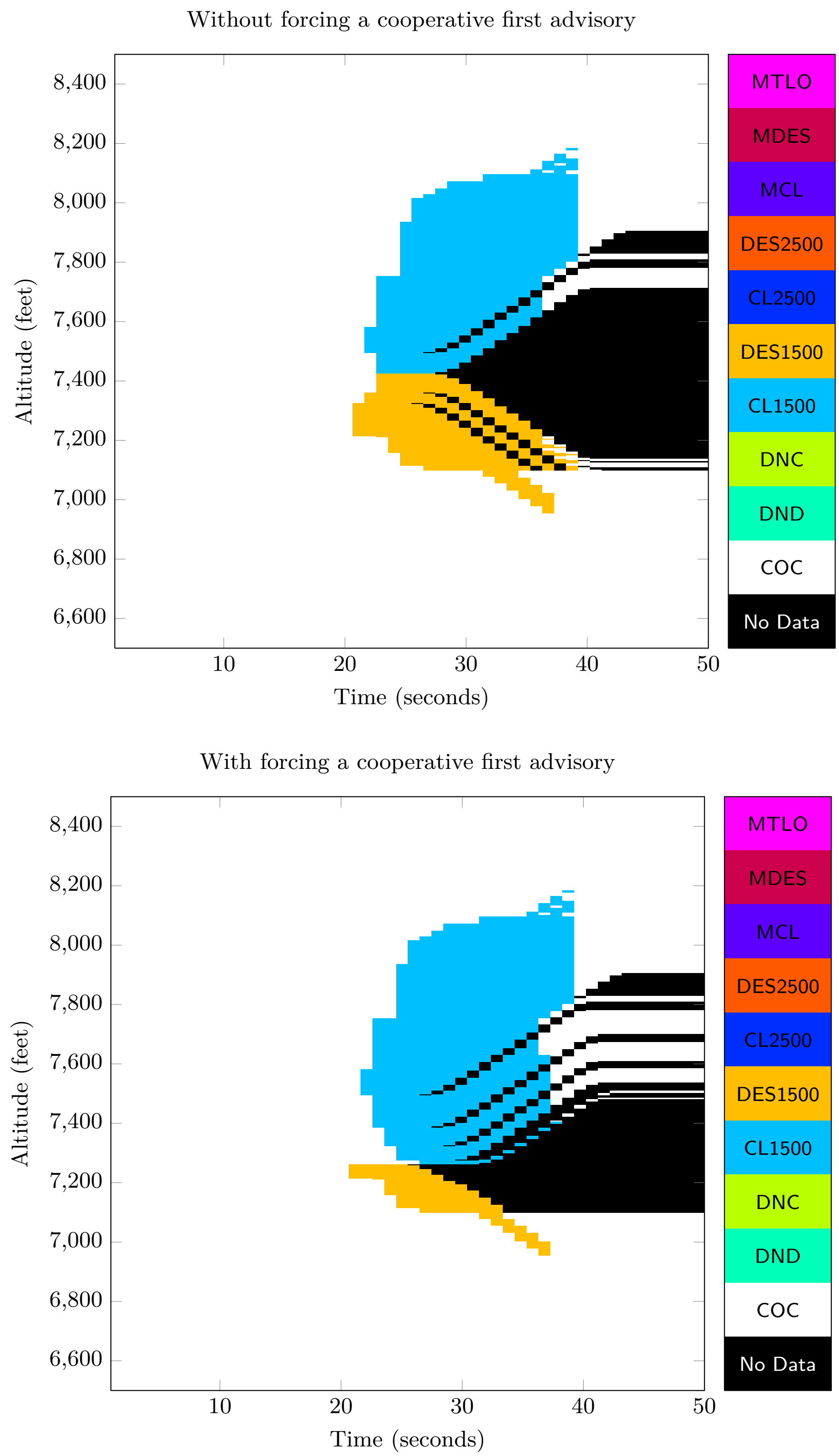


Figure 7. Policy plot of the master aircraft during a cooperative encounter. The intruder starts at 6800 ft with a vertical rate of 1000 ft/min. The horizontal TCA occurs at 40 s. Both aircraft are equipped with ACAS X.

3.3.2 Prohibited Reversals

The offline dynamics assume an unequipped intruder. Not considering the dynamics of an equipped intruder can affect encounters where the relative altitude between aircraft is small. Multiple reversals can result with low separation. An example is shown on the right vertical plot of Figure 8. Prohibiting a reversal for a certain time after a differing received VRC allows the vertical tracker to better detect intruder maneuvering.

In the example with no prohibition to reversals (right side of Figure 8), both aircraft issue advisories at 26 s. The slave aircraft starts the maneuver which can be seen by the leveling off of the track around 33 s. However, the offline dynamics do not consider a coordinated response by the intruder and the tracker does not have enough time to detect the maneuver, resulting in the master reversing. The intruder was maneuvering due to forced cooperation and the negative vertical rate was not as great as the master believed, making the crossing more difficult. The master then reverses again back to its original advisory but the encounter results in an NMAC with a vertical separation of 35 ft.

The encounter with the cost implemented can be seen on the left side of Figure 8. The cost prohibits reversals and COC if an advisory was issued for 10 s after a conflicting coordination message and the intruder is within 200 ft vertically. The master aircraft is not allowed to reverse at 32 s and issues a strengthening instead. The encounter resolves with a vertical separation of 385 ft.

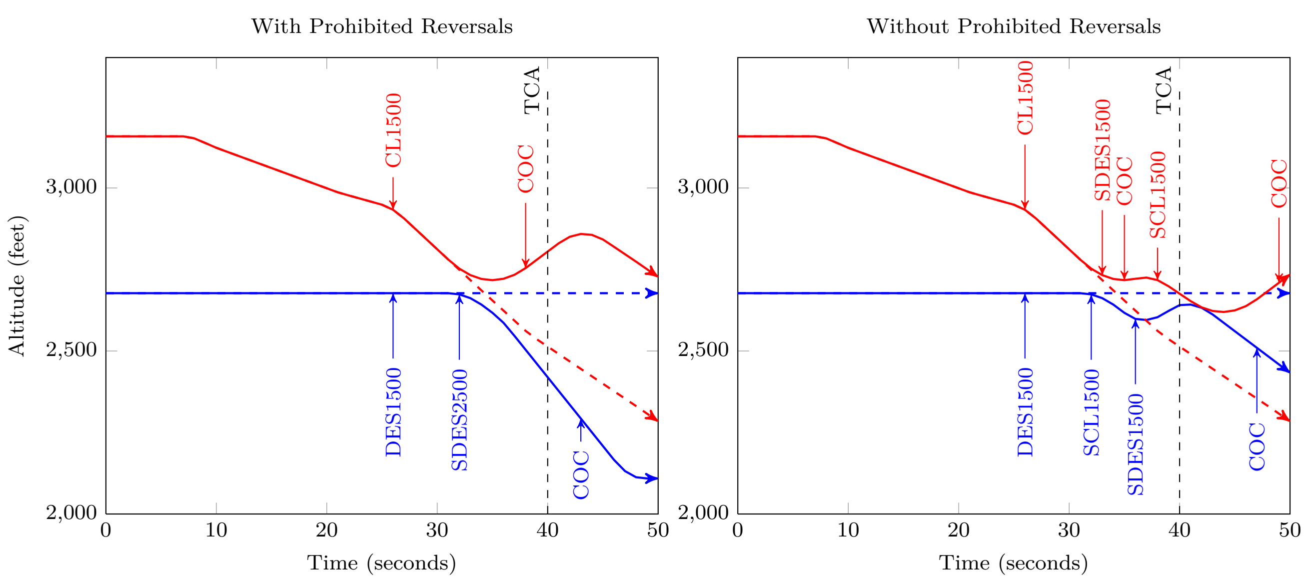
This online cost provides a significant improvement to safety. However, it removes some robustness to non-compliant intruders since it delays the ability to reverse an advisory. A robustness analysis is presented in Section 3.5.

3.3.3 Forced Reversal

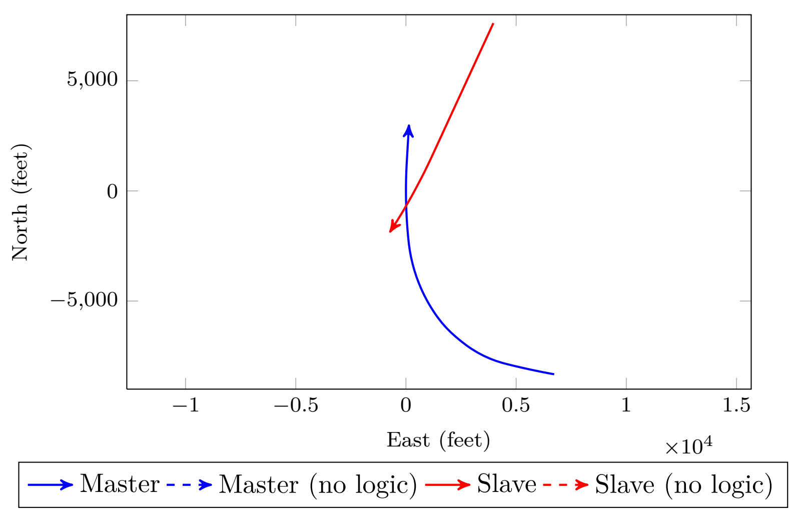
Forced cooperation can result in a situation where a slave issues an advisory and is then forced to reverse. Since the offline optimization does not take into consideration that certain advisories are prohibited, clear of conflict might have a lower expected cost than the reversed advisory. The cost of the reversed advisory is also increased by the online switching cost and could be penalized more than switching to clear of conflict (Section 2.7). The result can be a premature switch to clear of conflict rather than a reversal. This cost penalizes switching to clear of conflict when a reversal is the result of forced cooperation.

3.4 SIMULATION

Simulations were conducted using 5×10^5 encounters generated from a high-fidelity encounter model [18]. The same encounters were simulated twice, varying which aircraft was the master. Both aircraft were equipped with a CAS. The pilot response to an advisory is modeled as a 1/4 g acceleration applied 5 s after the advisory until the minimum commanded vertical rate is achieved. Subsequent advisories are modeled with a 1/3 g acceleration applied 3 s after the advisory is issued.



(a) Vertical profile.



(b) Horizontal profile.

Figure 8. Example coordinated encounter where basing the allowance of reversals on a received VRC and relative altitude significantly affects the outcome. The left vertical plot is with reversals prohibited for 10 s after a received VRC and when the intruder is within 200 ft vertically. The right plot is without the cost.

TABLE 5

Performance Evaluation of Coordination Strategies with Standard TCAS Sensor Noise

Metrics	No Strategy	Forced Coop	FC with Coop 1st Ad	ACAS X	TCAS
Risk Ratio	4.92×10^{-2}	1.07×10^{-2}	5.91×10^{-3}	3.26×10^{-3}	8.81×10^{-3}
Pr(NMAC)	1.45×10^{-4}	3.15×10^{-5}	1.74×10^{-5}	9.56×10^{-6}	2.59×10^{-5}
Pr(Induced NMAC)	8.63×10^{-5}	1.97×10^{-5}	1.33×10^{-5}	6.21×10^{-6}	1.82×10^{-5}
Pr(Unresolved NMAC)	5.82×10^{-5}	1.18×10^{-5}	4.05×10^{-6}	3.36×10^{-6}	7.72×10^{-6}
Pr(Alert)	2.84×10^{-1}	2.84×10^{-1}	2.84×10^{-1}	2.84×10^{-1}	5.32×10^{-1}
Pr(Strengthening)	8.45×10^{-3}	4.16×10^{-3}	4.15×10^{-3}	4.70×10^{-3}	1.36×10^{-2}
Pr(Reversal)	9.88×10^{-3}	7.54×10^{-3}	2.67×10^{-3}	2.39×10^{-3}	5.55×10^{-3}
Pr(Restart)	1.11×10^{-3}	9.58×10^{-4}	1.30×10^{-3}	6.31×10^{-4}	9.95×10^{-3}

3.4.1 ACAS X vs. ACAS X

Various simulations were performed on the high-fidelity encounter model. Table 5 summarizes the results. In the table "No Strategy" means the simulation was conducted with two equipped aircraft, but there was no coordination strategy implemented. The simulation "Forced Coop" implemented the basic forced cooperation strategy with no online costs. "FC with Coop 1st Ad" refers to forced cooperation with the forced cooperative first advisory online cost (Section 3.3.1), and "ACAS X" is forced cooperation with all online costs.

The risk ratio for no strategy is over twice that of TCAS. Once a TCAS-like strategy is used (FC with Coop 1st Ad), the risk ratio drops below that of TCAS with a much lower alert rate. The final coordination strategy for ACAS X outperforms TCAS in every metric and drops the risk ratio by 63 % while alerting 46.6 % less often.

3.4.2 ACAS X vs. TCAS 7.1

To test the interoperability of ACAS X with TCAS, two different interoperability tests were conducted. Both simulations involved one ACAS X equipped aircraft and one TCAS equipped aircraft. The Mode S addresses were varied to make the ACAS X equipped aircraft the master and vice versa. The results are summarized in Table 6.

Since the coordination strategy implemented on ACAS X is very similar to that of TCAS, the interoperability of the two logics was not a concern. The simulation results support that initial hypothesis. The addition of ACAS X to the encounter significantly improves safety. The alert rate for the mixed scenarios is higher than the alert rate for TCAS. The alert rate for the mixed scenarios include cases where ACAS X would alert and TCAS would not and vice versa. The mixed scenario where TCAS is the master aircraft is actually safer than the scenario with two ACAS X equipped aircraft; however, the safety comes at a much higher alert rate.

TABLE 6

Performance Evaluation of Interoperability with Standard TCAS Sensor Noise

Metrics	ACAS X Master	TCAS Master	Both ACAS	Both TCAS
Risk Ratio	3.71×10^{-3}	2.90×10^{-3}	3.26×10^{-3}	8.81×10^{-3}
Pr(NMAC)	1.09×10^{-5}	8.51×10^{-6}	9.56×10^{-6}	2.59×10^{-5}
Pr(Induced NMAC)	6.86×10^{-6}	4.87×10^{-6}	6.21×10^{-6}	1.82×10^{-5}
Pr(Unresolved NMAC)	4.05×10^{-6}	3.64×10^{-6}	3.36×10^{-6}	7.72×10^{-6}
Pr(Alert)	5.34×10^{-1}	5.34×10^{-1}	2.84×10^{-1}	5.32×10^{-1}
Pr(Strengthening)	8.64×10^{-3}	8.33×10^{-3}	4.70×10^{-3}	1.36×10^{-2}
Pr(Reversal)	6.16×10^{-4}	1.42×10^{-3}	2.39×10^{-3}	5.55×10^{-3}
Pr(Restart)	8.73×10^{-3}	8.72×10^{-3}	6.31×10^{-4}	9.95×10^{-3}

3.5 ROBUSTNESS ANALYSIS

Coordinated encounters where an intruder does not respond to an advisory is an area of concern, especially after the Überlingen mid-air collision in 2002. If the own aircraft is the master, then forced cooperation should not be affected by a non-compliant intruder. However, the cost discussed in Section 3.3.2 does consider the received VRC message if the own aircraft is the master. Also, if the intruder is the master, forced cooperation limits when the own aircraft is allowed to reverse. The robustness of the forced cooperation scheme with the added online costs is analyzed in this section and compared to TCAS.

The simulation setup is the same as in Section 3.4 except the pilot response time was varied. The pilot response was varied for either the master or the slave, but not at the same time. The normal pilot response of an initial delay of 5 s and all subsequent delays of 3 s was used unless specified. In Table 7 the pilot response column describes the pilot response varied for that simulation. "Normal" refers to a standard simulation. "None" means the pilot response was turned off for that particular aircraft and "7 – 5" means the initial delay was set to 7 s and the subsequent delay was set to 5 s for either the master or slave. The simulations were conducted using TCAS, and ACAS X, both with and without the prohibited reverals online cost discussed in Section 3.3.2. The logic with the online costs is labeled "ACAS X" and the logic without the costs is labeled "Modified ACAS X." The results are summarized in Table 7.

In Table 7, the metric "Induced RR" refers to the induced NMAC risk ratio. It was calculated by dividing the probability of an induced NMAC by the probability of NMAC without a collision avoidance system:

$$\text{Induced RR} = \frac{\text{Pr(Induced NMAC)}}{\text{Pr(NMAC without a CAS)}}. \tag{5}$$

In every scenario, ACAS X with all of the added costs outperformed both TCAS and the modified version of ACAS X. When the slave was the aircraft not responding to advisories, the results between the modified version of ACAS X and ACAS X were very similar, as expected. The percentage increase in the metrics from the normal scenario was greater for ACAS X than for the other two logics; however, ACAS X was still safer. When the master aircraft or the slave aircraft still responded to its advisories but with a delayed response, ACAS X saw a significant increase in risk ratio compared to TCAS (around 130 % for ACAS X and around 35 % for TCAS). Yet, ACAS X was safer than TCAS in the same scenarios and was even safer than when the TCAS intruder responded normally.

TABLE 7

Performance Evaluation of Coordination Robustness with Standard TCAS Sensor Noise

	TCAS		Modified ACAS X		ACAS X	
Pilot Response	Risk Ratio	Induced RR	Risk Ratio	Induced RR	Risk Ratio	Induced RR
Normal	8.81×10^{-3}	6.18×10^{-3}	5.91×10^{-3}	4.53×10^{-3}	3.26×10^{-3}	2.11×10^{-3}
7–5 Master	1.25×10^{-2}	6.49×10^{-3}	1.28×10^{-2}	1.06×10^{-2}	7.43×10^{-3}	5.63×10^{-3}
None Master	7.43×10^{-2}	3.07×10^{-2}	3.92×10^{-2}	2.48×10^{-2}	3.80×10^{-2}	2.44×10^{-2}
7–5 Slave	1.15×10^{-2}	6.83×10^{-3}	1.35×10^{-2}	1.12×10^{-2}	7.95×10^{-3}	6.19×10^{-3}
None Slave	5.14×10^{-2}	2.67×10^{-2}	2.49×10^{-2}	1.23×10^{-2}	2.89×10^{-2}	1.62×10^{-2}

3.6 DISCUSSION

This section discussed different coordination schemes for ACAS X. Offline schemes could be implemented into ACAS X; however, each scheme would require an increase in complexity of the model. A simple coordination scheme similar to that of TCAS was applied online and was shown to outperform TCAS in all evaluated metrics. ACAS X and TCAS are not significantly affected during mixed logic encounters. The similar coordination schemes allowed for well behaved cooperation between the two logics. Introducing ACAS X into an encounter with TCAS significantly increases safety over two TCAS equipped aircraft. The coordination scheme is implemented online for ACAS X. Therefore, as the offline logic improves for uncoordinated encounters, an increase in performance for coordinated scenarios is expected as well. Also, as long as the coordination scheme remains similar, an increase in interoperability performance is expected with continued improvements in the logic.

The robustness to delayed pilot responses was analyzed and ACAS X outperformed TCAS in all scenarios. However, since the ACAS X risk ratio was sensitive to pilot response delay, further work can explore different pilot response models during the offline optimization. The offline table used for this section was generated with a probabilistic pilot response that had an initial response delay that averages to 5 s and subsequent response delays that average to 3 s. Using higher delays or a different model could increase robustness while still maintaining safety.

This page intentionally left blank.

4 MULTIPLE THREAT ENCOUNTERS

Multithreat encounters are rare, but they will increase in frequency as the airspace becomes more dense. During the development of ACAS X, there has been some research into unequipped multithreat scenarios [2, 19]. For multithreat scenarios involving equipped intruders, the coordination of advisories is important. This section discusses the challenges with both unequipped and equipped intruders.

4.1 POTENTIAL SOLUTION METHODS

Ideally, for multiple intruder encounters, a solution would be obtained from an MDP modeling an arbitrary number of aircraft. However, this approach involves introducing at least three new variables for each additional intruder. The processing time to compute the optimal cost table and the storage requirements become intractable even with a modest number of intruders. If the intruders were equipped, it would involve solving a Dec-POMDP or I-POMDP. As previously discussed, the computational requirements for Dec-POMDPs and I-POMDPs are intractable, especially when considering multithreat encounters [15, 17, 14]. A global geometric approach could also be considered, but would not take advantage of the single threat progress attained with the optimized logic.

TCAS treats each threat individually, with the same threat detection, initial sense selection, and initial strength selection logic that would be used with a single intruder. The multithreat portion of the logic attempts to reconcile the senses and strengths associated with each intruder before displaying a composite advisory to the pilots. When all threats have the same sense, the logic simply uses the individual advisory with the greatest strength. When the senses of the individual advisories differ, TCAS uses a set of rules to either (1) identify a single sense for all threats or (2) issue a "dual-negative advisory" that places speed limits in both directions.

A command arbitration approach like that of TCAS could be used by ACAS X, but past studies have shown that a cost fusion approach can result in better performance [2, 19]. Cost fusion is implemented online using tables computed offline for single intruders.

4.2 COST FUSION METHODS

Cost fusion computes, for each intruder i, the optimal state-action costs $C^*(s_i, a)$ for all actions a, assuming that intruder i is the only threat. The state-action costs from multiple intruders are fused to arrive at the optimal state-action cost function $C^*(s, a)$. Fusing the costs requires defining a function f that combines costs associated with multiple intruders. That is,

$$C^*(s, a) = f(C^*(s_1, a), \ldots, C^*(s_N, a)), \tag{6}$$

where N is the number of intruders. After fusing the utilities, the optimal action is computed using

$$\pi^*(s) = \arg\min_a C^*(s, a). \tag{7}$$

TABLE 8

Costs for a Two-Intruder Example

Intruder	No Alert	Climb	Descend
1	13	11	7
2	15	0	8
Sum	28	**11**	15
Max	15	11	**8**

The fusion of the costs occurs after the online and offline costs are combined. Therefore, $C^*(s_i, a)$ represents the sum of the offline and online costs of taking action a for intruder i alone.

Two utility fusion methods were studied for the multithreat logic. The first method, the min-sum strategy, defines f to be a summation:

$$C^*(s, a) = \sum_i C^*(s_i, a). \tag{8}$$

Defining f in this way leads to counting penalties such as alert costs and reversal costs multiple times. The cost of alerting, for example, would be reflected in the state-action costs for each intruder. Adding these utilities together amounts to incurring the alert cost multiple times, though in reality the collision avoidance system can only alert once at any given time. This may cause the system to delay issuing the alert. Delaying an alert can be undesirable because there may be fewer available options to resolve the conflict further into the encounter. When more intruders are present, the importance of alerting earlier is magnified.

The second method, the min-max strategy, avoids accumulating penalties for each intruder by defining f as follows:

$$C^*(s, a) = \max_i C^*(s_i, a). \tag{9}$$

Table 8 is an example contrived to illustrate the difference between the two methods. There are two intruders and three actions (no alert, climb, and descend) from which to select at the current time. The table shows the cost for each intruder and for each action. The min-sum method issues the climb advisory because it is effective in preventing conflict with the second intruder (hence the low cost), even though following the climb may lead to conflict with the first intruder. The min-max method selects the descend action because the higher cost for executing the descend is 8 while the highest cost for executing the climb is 11.

One property of the cost fusion methods is that they do not alert any earlier than the single-threat policy on which they are built. It can be shown that if the optimal action for each intruder $\pi^*(s_1), \ldots, \pi^*(s_N)$ is to not alert, then the decomposition methods will not alert as well. This may be undesirable because it may be necessary to alert a little earlier to pass above or below all intruders.

Figure 9 shows the policies for the min-max and min-sum fusion methods. The intruders are at 7150 ft and 7850 ft and fly level. The horizontal geometry is head-on and TCA for both intruders is at 40 s. The own aircraft is initially level. As expected, double-counting costs makes the alerting region for the min-sum method smaller. The alerting region for the min-max method is similar to the individual policies, but still varies. The min-max method delays alerting a little longer when compared to the individual policies, especially when the own aircraft is between the intruders.

One problem with cost fusion techniques is the influence of non-critical agents on decisions. For example, suppose a second intruder enters an encounter, but is far away and does not pose a threat. The third aircraft is not a concern; however, fusing the costs of that intruder, despite it not being a factor, affects the action selection process. To mitigate the problem of non-threatening aircraft impacting the alert behavior in ACAS X, a form of arbitration is incorporated in which the costs of an intruder are only considered if the own aircraft would alert against that intruder in isolation. This process allows the cost fusion to alert at the same time as it would in an individual encounter.

Figure 10 shows an example of the ACAS X policy with this type of arbitration and compares the policy to TCAS. Once arbitration is introduced, the policies between the two fusion schemes become very similar. The geometry of the encounter is the same as presented in Figure 9. The min-max cost fusion is used for ACAS X, but the min-sum policy looks similar. Because the multithreat logic is only used when both aircraft are threats, there is a delay when the own aircraft is in the middle compared to TCAS.

4.3 MULTITHREAT LEVEL-OFFS

The single threat advisory set does not contain any dual-negative advisories that place speed limits in both senses. TCAS uses a variety of these advisories to resolve multithreat encounters. An example of a dual-negative advisory is a mulithreat level-off (MTLO). An MTLO advisory limits the up sense to a maximum vertical rate of 250 ft/min and limits the down sense to a minimum vertical rate of −250 ft/min. An MTLO is often appropriate in sandwich encounters, where there are threats both above and below. A sandwich encounter often results in an up sense being issued against one intruder and a down sense against another. Often a single sense advisory is inadequate, and so an MTLO is issued, allowing the aircraft to pass between the two intruders.

In ACAS X, the cost of an MTLO is based on the offline costs of the single-threat actions. Let a^i denote the best action with respect to intruder i. Also, let a^i_{cor} and a^i_{prev} denote the best corrective and preventive actions with respect to intruder i that is consistent with the sense of a^i. The cost of an MTLO is given by

$$\min\{f(C^*(s_1, a^1_{\mathrm{cor}}), \ldots, C^*(s_n, a^n_{\mathrm{cor}})), f(C^*(s_1, a^1_{\mathrm{prev}}), \ldots, C^*(s_n, a^n_{\mathrm{prev}}))\}, \tag{10}$$

where f is the multithreat fusion function, such as min or sum.

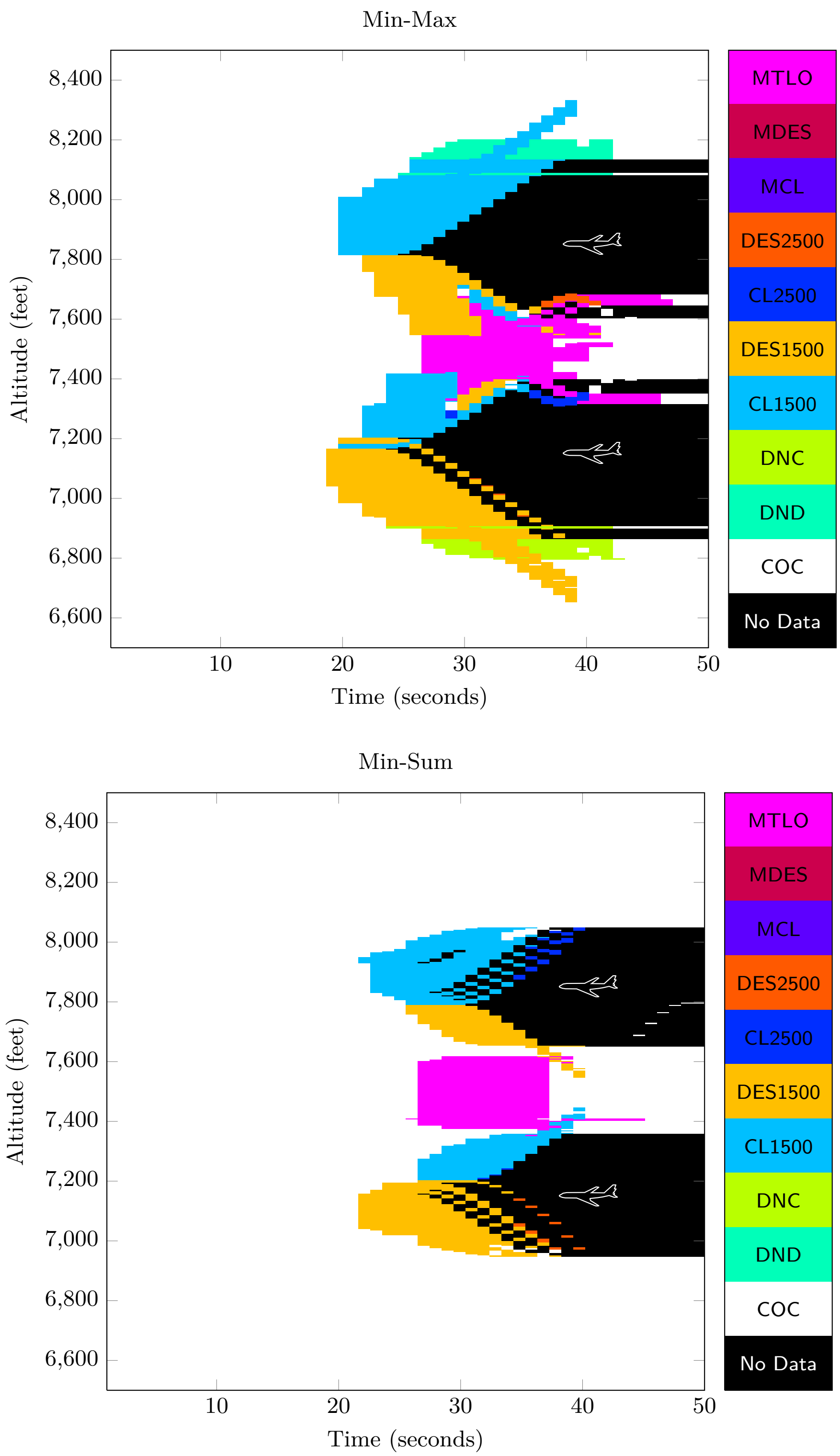


Figure 9. Policy plot using the min-sum and min-max cost fusion schemes. The intruders start at 7150 ft and 7850 ft and fly level. The horizontal geometry is head-on and both intruders' horizontal TCA to the own aircraft occur at 40 s. The own aircraft is initially in level flight.

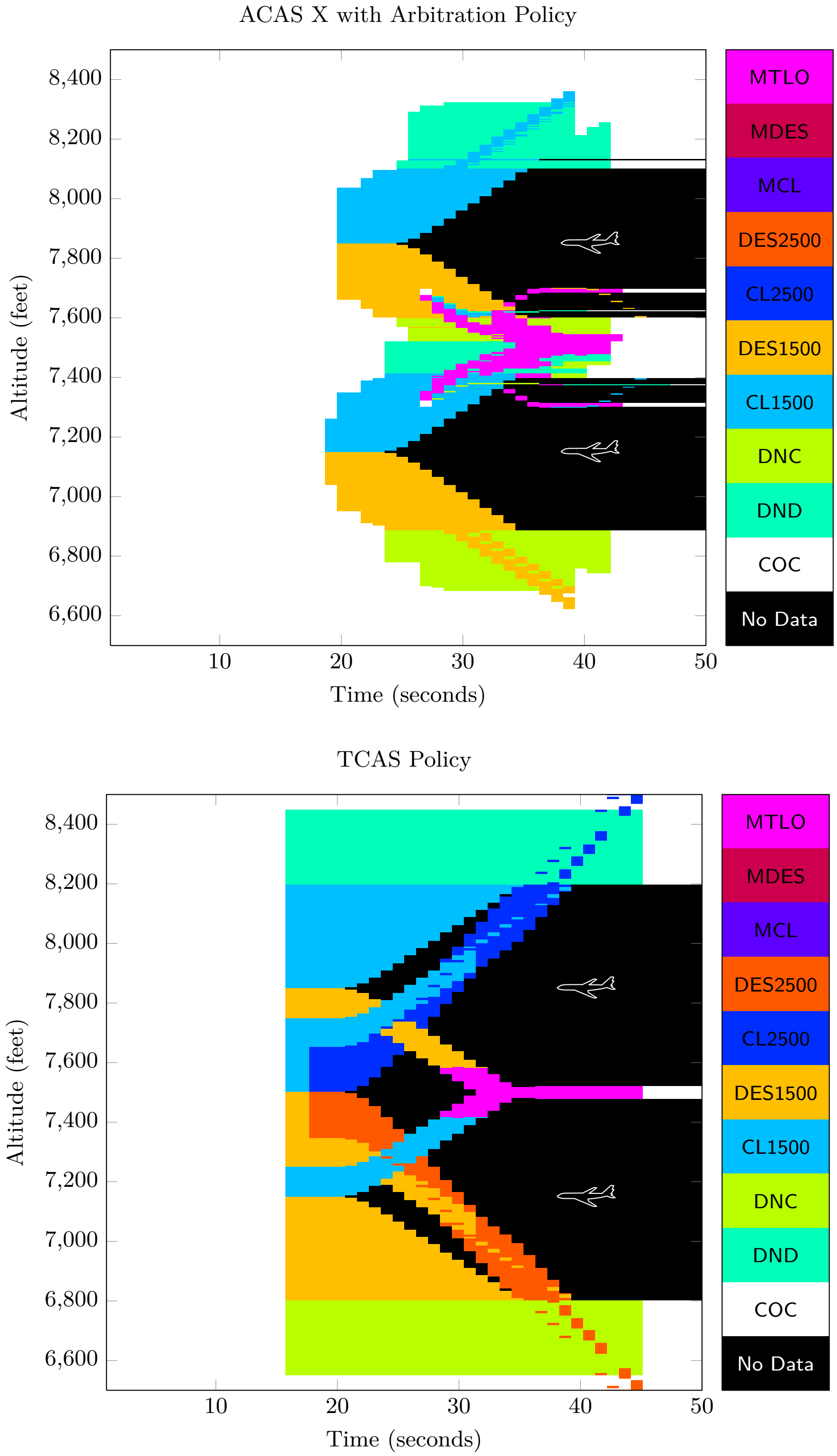


Figure 10. Policy plot of ACAS X and TCAS. The ACAS X plot uses the min-max cost fusion scheme with arbitration. The intruders start at 7150 ft and 7850 ft and fly level. The horizontal geometry is head-on and both intruders' horizontal TCA to the own aircraft occur at 40 s. The own aircraft is initially in level flight.

4.4 RESOLUTION ADVISORY STATE

The offline costs are read from the look-up table based on the belief state of the intruder and the own aircraft's state. Therefore, determining the alert to issue at each time step requires knowledge of the state of the resolution advisory, s_{RA}. There is uncertainty as to whether the pilot is responding to an advisory, thus making s_{RA} not fully observable. Instead, a probability distribution, or belief state, over possible values of s_{RA} must be maintained to summarize the beliefs regarding the response of the pilot to advisories. As new observations of the aircraft state are made each time step, the belief state is recursively updated using standard model-based filtering techniques [2].

The belief distribution over s_{RA} must always be updated based on the global action. Maintaining different beliefs over s_{RA} for each intruder could result in selecting inappropriate advisories. All s_{RA} updates occur with the global actions except when a MTLO is issued. If a MTLO is issued, the s_{RA} is updated with a DNC if the own vertical rate is greater than or equal to zero and DND if the own vertical rate is less than zero.

Treating s_{RA} globally has some important consequences in multithreat situations. Suppose, for example, there are two intruders. The own aircraft alerts due to the first intruder. The second intruder is close, but is not considered a threat. Since s_{RA} is global, there is no longer a cost for alerting against the second intruder, which can result in premature alerting against the second intruder. The multithreat COC incentive cost (Section 4.5.2) helps mitigate this problem.

4.5 ONLINE COSTS

Incorporating online costs can help improve performance in multithreat scenarios. These online costs are summed with the offline costs during execution. Various online costs depend upon the previous advisory issued. In multithreat situations, each intruder is treated as if it were in isolation. If the intruder is not a threat, then the online costs are updated as if a clear of conflict was issued. If the best action with respect to the intruder is a MTLO, then the online costs associated with that intruder are updated with either a DNC or DND. Otherwise, they are updated with the global action.

4.5.1 Incentive to MTLO

In the absence of any online costs, the individual cost of an MTLO is exactly equal to one of the other actions. To provide an incentive to issue an MTLO, negative costs are added online to the MTLO action. Initiating an MTLO receives an incentive of 0.005 and continuing an MTLO receives an incentive of 0.001.

4.5.2 Multithreat COC Incentive

As discussed in Section 4.4, the s_{RA} is common when referencing the offline table and can result in premature alerting toward secondary intruders. This online cost helps mitigate that affect. One part of the cost function used in the offline optimization is the alert cost. If the own aircraft believes that it is responding to an advisory, then the costs associated with that state are not affected by the alert cost. If the own aircraft has alerted against intruder A and then intruder B enters the encounter, the initial alert cost should still be applied to B.

Instead of raising the cost of all alerts, a negative online cost can be added to the clear of conflict advisory for secondary intruders. An example of a problem that can occur without this negative online cost is shown on the right side of the vertical profile in Figure 11. Aircraft A, B, and C are ordered by decreasing Mode S address. At 5 s, B issues a climb due to the threat from C. Since the advisory states with respect to all the intruders are identical, there is no longer an alert cost for issuing an advisory against A. Therefore, B issues a climb with respect to A at 6 s and sends a do not climb VRC to A. Since A is forced to be compatible with B, it delays alerting and then issues a descend, forcing B to maintain its climb, ultimately resulting in an NMAC. If a COC incentive was applied (left side of the vertical profile), then B would not issue a climb with respect to C. B delays until 23 s, at which it issues a descend toward A and results in a MTLO.

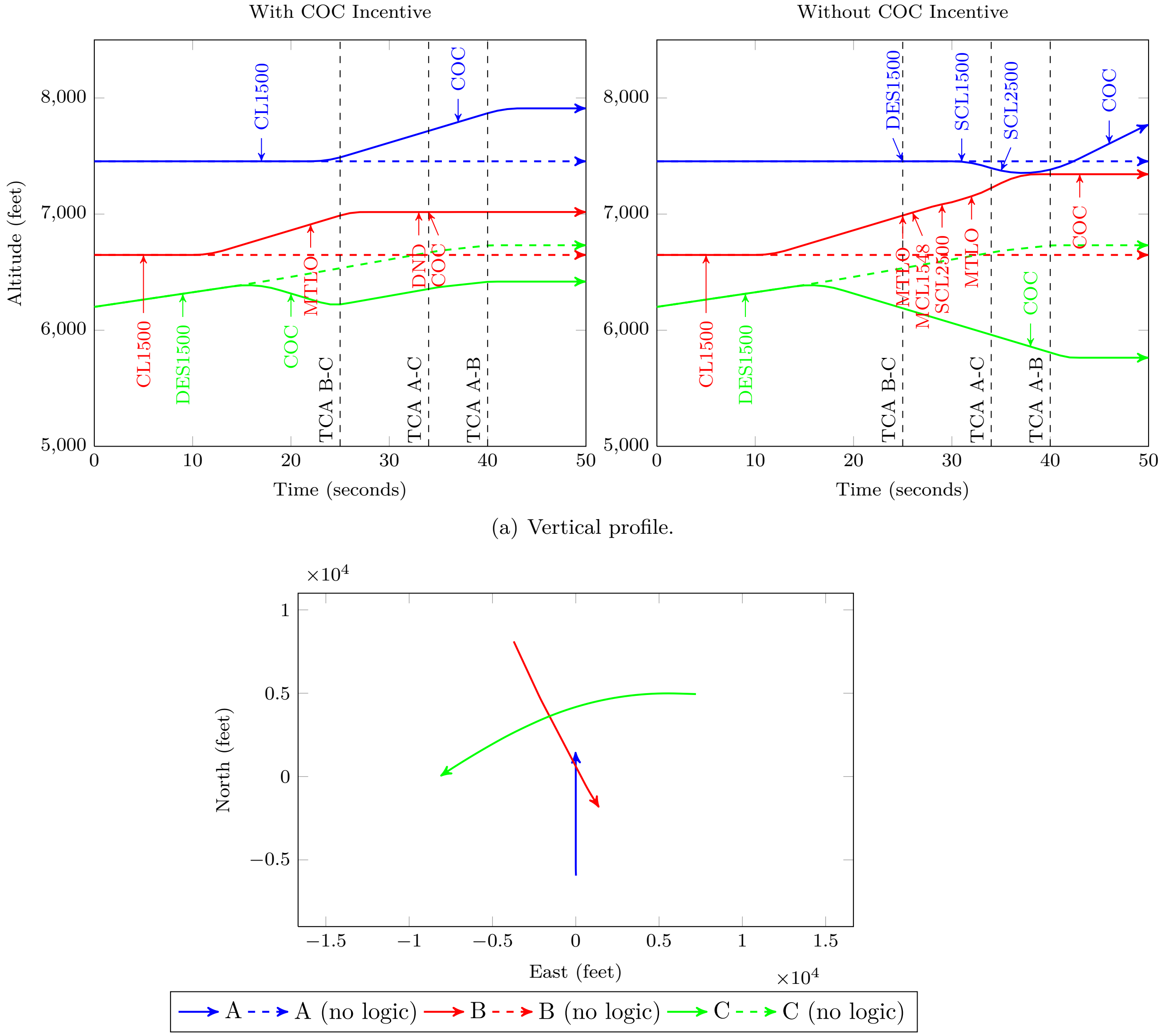
4.5.3 Restriction of MTLOs

MTLOs are needed to resolve some multithreat encounters, but there are situations in which issuing a MTLO is problematic. Checking for these situations when computing the online cost can improve performance.

One situation where a MTLO is not desired is when the individual action toward one intruder would result in crossing its altitude and the action toward the other intruder is a preventive advisory (DNC or DND). If a crossing is selected along with a preventive, then the crossing is more pertinent and a MTLO is not allowed. An example of a situation in which this online cost helps resolve an encounter is shown in Figure 12. In this encounter, only the blue and red aircraft are equipped with ACAS X and the green aircraft is unequipped. At 22 s the master aircraft wants to issue a crossing descend with respect to the unequipped intruder and a DND with respect to the slave aircraft. The conflicting sense advisories results in a MTLO and then an NMAC (right side). If the MTLO was prevented until either both advisories became a corrective or the encounter was no longer a crossing, then the NMAC can be prevented (left side).

Another situation where a MTLO is undesirable involves unequipped intruders. If an MTLO is issued when an unequipped intruder is within the vertical separation for an NMAC, then the own aircraft will fly level into a conflict. Prohibiting a MTLO within a vertical threshold of an unequipped intruder ensures that a maneuver will occur to exit the NMAC region before a MTLO is issued. Figure 13 shows an encounter where this online cost is helpful. In this example, a MTLO is prohibited if the aircraft is within 150 ft vertically of an unequipped intruder.

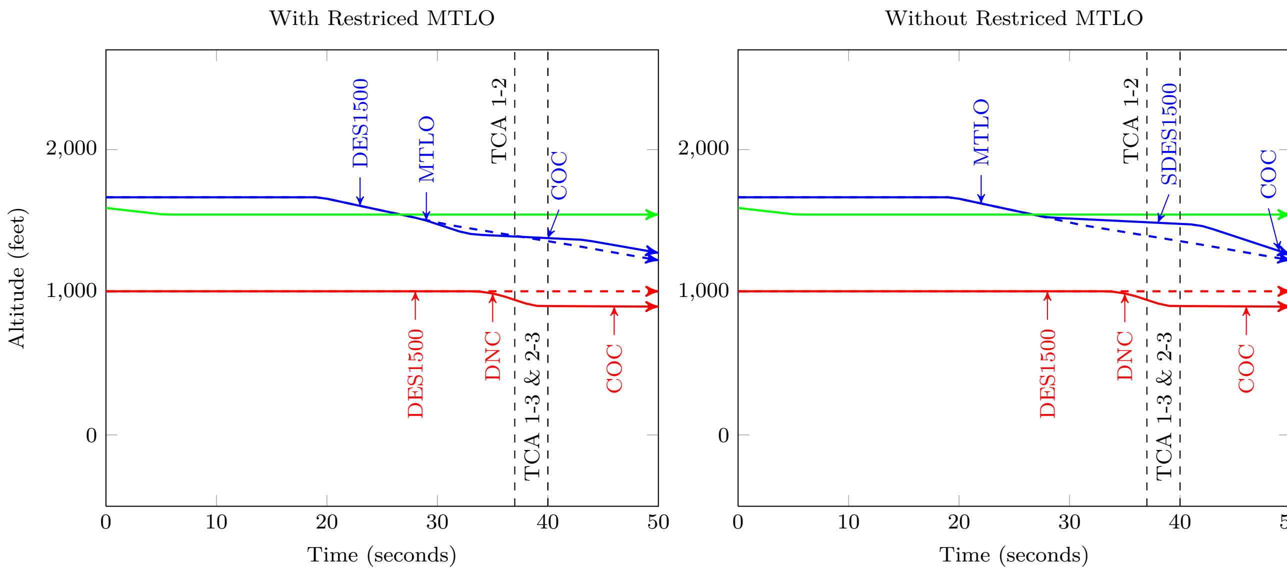
In Figure 13, the intruders are only separated by 170 ft. There is not enough separation to allow the own aircraft to split the two intruders. However, if the intruders were equipped with a CAS, then a MTLO could be issued because intruder 1 would be expected to climb while intruder 2 would be expected to descend. However, since neither aircraft is equipped, then a MTLO should not be allowed.



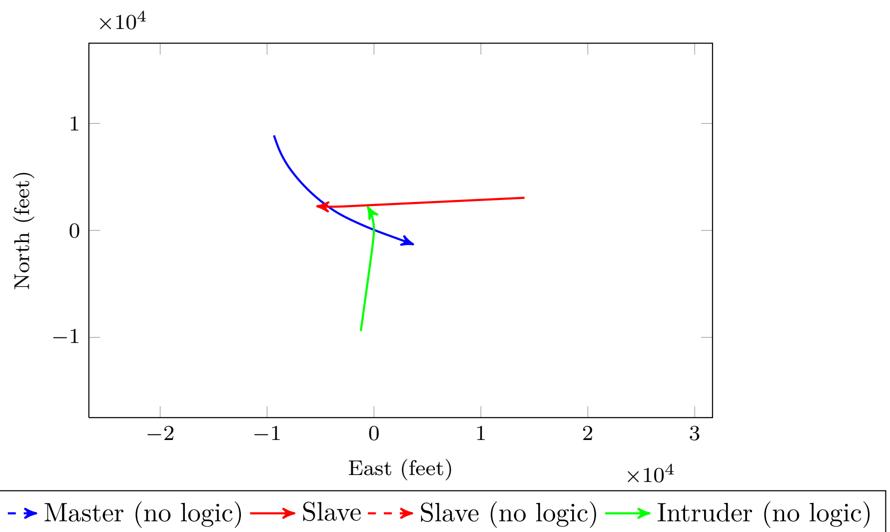
(a) Vertical profile.

(b) Horizontal profile.

Figure 11. Example multithreat encounter where the multithreat COC incentive plays a critical role.

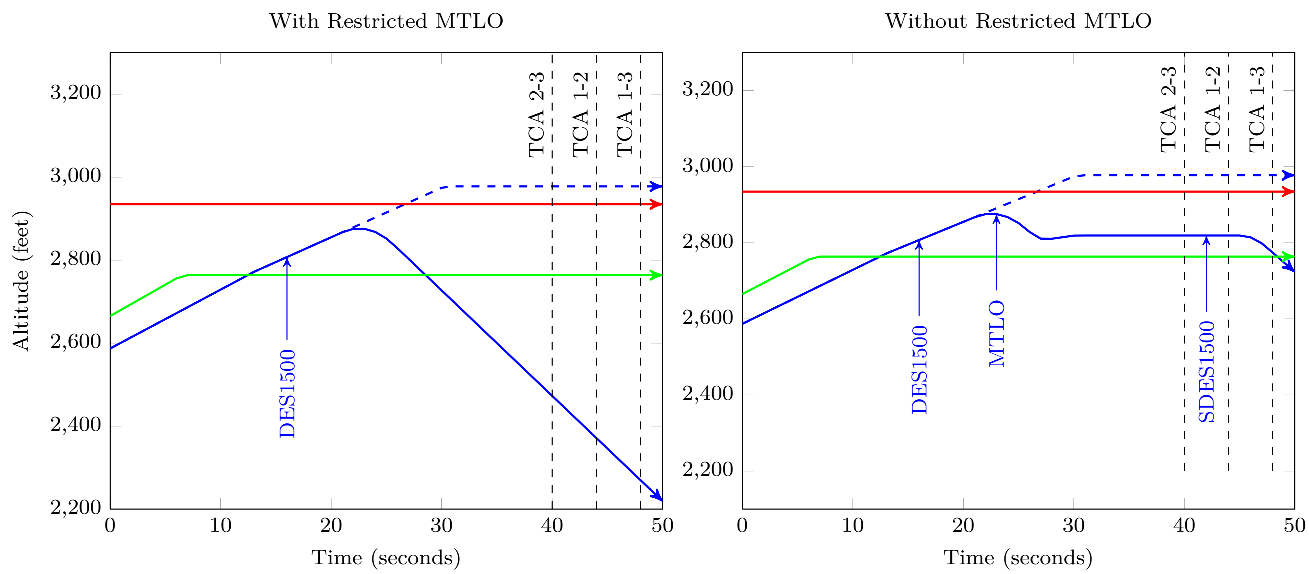


(a) Vertical profile.

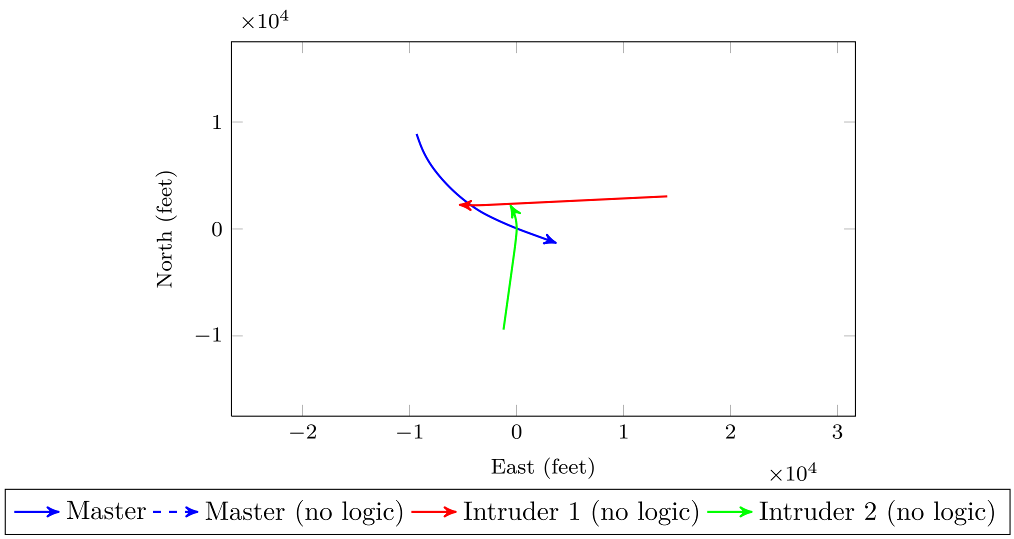


(b) Horizontal profile.

Figure 12. Example multithreat encounter where delaying a MTLO prevents an NMAC.



(a) Vertical profile.



(b) Horizontal profile.

Figure 13. Example multithreat encounter where preventing a MTLO due to unequipped intruders prevents an NMAC.

4.6 MULTITHREAT COORDINATION

The communication between aircraft in a multithreat encounter is still limited to the VRC, VSB, CVC, and the Mode S address. Therefore, coordination must be done pairwise. A non-zero VRC will only be sent to an intruder if the individual advisory is an alert. For example, if the own aircraft issues a climb with respect to intruder A and a COC with respect to B, then a do not climb VRC would be sent to A and no message would be sent to B.

If a MTLO is issued by the own aircraft, then different coordination messages are sent to the intruders. The coordination message sent to each intruder is based on the individual advisories toward those intruders considered in isolation. All intruders in which an up sense was desired would be sent a do not climb and all intruders in which a down sense was desired would be sent a do not descend VRC. If there was no advisory toward a particular intruder, then no message would be sent.

From a high level perspective, ACAS X does not differ greatly from TCAS in how it handles multithreat encounters. Both handle coordination in a pairwise manner and use single threat logic to resolve encounters individually, fusing those results to determine a global action. The fundamental differences are in the fusion process. However, those differences do not affect interoperability. Section 4.7.5 will discuss interoperability simulation results.

4.7 SIMULATION

Simulations were conducted using 5×10^5 three-aircraft encounters generated from a high-fidelity model [20]. Unless otherwise stated, all results include all permutations of the encounter model encounters, representing a total of 3×10^6 encounters. The pilot response to an advisory is modeled as a 1/4 g acceleration applied 5 s after the advisory until the minimum commanded vertical rate is achieved. Subsequent advisories are modeled with a 1/3 g acceleration applied 3 s after the advisory is issued.

If the encounter only involves one equipped aircraft, then the risk ratio is calculated with that aircraft. Since a CAS cannot control the other two aircraft, only the equipped aircraft is considered when computing the NMAC rate. For ease of presenting the results a three letter sequence will be used to signify the equipage of the aircraft involved. The aircraft listed first would have the lowest Mode S address and the aircraft listed last would have the highest Mode S address. For example, the sequence "XTS" signifies a three-aircraft encounter of a master ACAS X aircraft, a slave TCAS aircraft and an unequipped aircraft with a Mode S transponder.

4.7.1 Online Cost Evaluation

The effectiveness of the online costs were evaluated on 5×10^5 encounters. The min-max fusion technique was used for the ACAS X simulations. Table 9 summarizes the results. The metrics presented are risk ratios for the respective systems. Therefore, comparing an XXX system to an XSS system is not possible because the XSS system only considers NMACs with the equipped aircraft while the XXX system considers NMACs for all of the aircraft.

In the table "Crossing-preventive" refers to the cost that prohibits a MTLO when a crossing and a preventive advisory are the two opposing senses that prompted the check for a MTLO. "Unequipped MTLO" refers to the cost that prohibits a MTLO when the intruders that the own aircraft are alerting against are unequipped. Overall, the online costs provide a benefit to the multithreat scenarios. The COC incentive and the unequipped MTLO costs provide the largest improvements.

TABLE 9

Performance Evaluation of the Multithreat Online Costs Using Risk Ratios

System	XXX	XXS	XSS
All online costs	7.19×10^{-2}	1.09×10^{-1}	4.07×10^{-2}
No COC incentive cost	7.77×10^{-2}	1.10×10^{-1}	3.95×10^{-2}
No crossing-preventive cost	7.17×10^{-2}	1.13×10^{-1}	4.09×10^{-2}
No unequipped MTLO cost	7.19×10^{-2}	1.09×10^{-1}	7.21×10^{-2}

4.7.2 ACAS X vs. Two Unequipped Intruders

Table 10 summarizes the simulation results for scenarios where there were two unequipped intruders. These results are for all acceptable permuted cases of the encounters from the encounter model (1.5×10^6 total encounters). "XSS Max" refers to the min-max fusion technique, while "XSS Sum" refers to the min-sum method. The min-sum method outperforms the min-max method, but has a higher induced NMAC rate. Both methods significantly outperform TCAS. The ACAS X min-max method reduces the risk ratio by 20 % while alerting 28 % less.

TABLE 10

Performance Evaluation with Two Unequipped Intruders

Metrics	XSS Max	XSS Sum	TSS
Risk Ratio	1.04×10^{-1}	1.03×10^{-1}	1.30×10^{-1}
Pr(NMAC)	1.16×10^{-3}	1.15×10^{-3}	1.45×10^{-3}
Pr(Induced NMAC)	1.43×10^{-4}	1.54×10^{-4}	2.02×10^{-4}
Pr(Alert)	4.60×10^{-1}	4.60×10^{-1}	6.42×10^{-1}
Pr(Strengthening)	2.53×10^{-2}	2.07×10^{-2}	6.09×10^{-2}
Pr(Reversal)	1.02×10^{-2}	5.87×10^{-3}	1.16×10^{-2}

4.7.3 Two Equipped Aircraft and One Unequipped

Table 11 summarizes the simulation results for scenarios where two aircraft were equipped with ACAS X and there was a Mode S unequipped intruder. These results are for all permuted cases of the encounters from the encounter model (3×10^6 total encounters). ACAS X outperforms

TCAS in every category except for the Pr(Induced NMAC). A large portion of the induced NMACs occur when an advisory is issued due to the other equipped aircraft and causes a maneuver toward the unequipped aircraft. Often a MTLO is issued which would resolve the encounter if the aircraft was equipped, but because it is not, the aircraft continue on their paths and result in an NMAC.

ACAS X results in a significant improvement to the unresolved NMACs, thus resulting in an increase in overall safety. The safety improvement was capable with a 21 % reduction in alerts, and for the min-max method, a 78 % reduction in reversals compared to TCAS.

TABLE 11

Performance Evaluation with Two Equipped Aircraft and One Unequipped Intruder

Metrics	XXS Max	XXS Sum	TTS
Risk Ratio	1.06×10^{-1}	1.09×10^{-1}	1.17×10^{-1}
Pr(NMAC)	1.77×10^{-3}	1.82×10^{-3}	1.95×10^{-3}
Pr(Induced NMAC)	3.11×10^{-4}	3.33×10^{-4}	2.17×10^{-4}
Pr(Alert)	6.20×10^{-1}	6.20×10^{-1}	7.85×10^{-1}
Pr(Strengthening)	3.51×10^{-2}	3.23×10^{-2}	8.40×10^{-2}
Pr(Reversal)	7.26×10^{-3}	6.91×10^{-3}	3.32×10^{-2}

4.7.4 Three Equipped Aircraft

Table 12 summarizes the simulation results for scenarios with two ACAS X aircraft and one Mode S aircraft. These results are for all permuted cases of the encounters from the encounter model (3×10^6 total encounters). ACAS X outperforms TCAS in every metric and the min-max fusion method outperforms the min-sum method in all metrics but Pr(Strengthening) and Pr(Reversal). The min-max fusion method resulted in a reduction in Risk Ratio over TCAS by 14 % and a reduction in induced NMACs by 31 %. The improvements to the operational metrics by ACAS X were more significant with a reduction in alerts by 19 % and a reduction in reversals by 73 % compared to TCAS.

TABLE 12

Performance Evaluation with Three Equipped Aircraft

Metrics	XXX Max	XXX Sum	TTT
Risk Ratio	7.23×10^{-2}	7.29×10^{-2}	8.40×10^{-2}
Pr(NMAC)	1.20×10^{-3}	1.21×10^{-3}	1.40×10^{-3}
Pr(Induced NMAC)	4.44×10^{-5}	5.09×10^{-5}	6.49×10^{-5}
Pr(Alert)	6.45×10^{-1}	6.45×10^{-1}	8.02×10^{-1}
Pr(Strengthening)	4.05×10^{-2}	3.78×10^{-2}	6.61×10^{-2}
Pr(Reversal)	1.49×10^{-2}	1.45×10^{-2}	5.45×10^{-2}

4.7.5 Interoperability

Interoperability was tested in both two and three aircraft scenarios, varying the Mode S addresses. Since the systems are fundamentally different, multiple scenarios were run varying the Mode S addresses. The results for the two equipped aircraft scenarios are summarized in Table 13, and the three equipped aircraft scenario results are presented in Table 14. All simulations used the min-max fusion method on ACAS X.

Similar to the results of Section 4.7.3, introducing an ACAS X equipped aircraft increases safety and outperforms TCAS in all operational metrics. However, it increases the Pr(Induced NMAC) slightly over the TTS scenario, but reduces it compared to the XXS scenario. Examining the Pr(NMAC) for each equipped aircraft in the individual scenarios shows that the ACAS X aircraft is always safer than a TCAS equipped aircraft.

For the three equipped interoperability scenarios, the introduction of an ACAS X aircraft improves the overall safety. Safety is improved the most when the ACAS X aircraft is the master and can control the reversals. The more ACAS X involvement in the scenarios, the safer the outcome, as can be seen in Figure 14.

TABLE 13

Performance Evaluation of Interoperability with Two Equipped Aircraft and One Unequipped Intruder

Metrics	XTS	TXS	XXS	TTS
Risk Ratio	1.08×10^{-1}	1.11×10^{-1}	1.06×10^{-1}	1.17×10^{-1}
Pr(NMAC AC1)	1.08×10^{-3}	1.18×10^{-3}	1.08×10^{-3}	1.20×10^{-3}
Pr(NMAC AC2)	1.15×10^{-3}	1.11×10^{-3}	1.09×10^{-3}	1.20×10^{-3}
Pr(Induced NMAC)	2.52×10^{-4}	2.73×10^{-4}	3.11×10^{-4}	2.17×10^{-4}
Pr(Alert)	7.49×10^{-1}	7.50×10^{-1}	6.20×10^{-1}	7.85×10^{-1}
Pr(Strengthening)	6.19×10^{-2}	6.15×10^{-2}	3.51×10^{-2}	8.40×10^{-2}
Pr(Reversal)	1.19×10^{-2}	1.53×10^{-2}	7.26×10^{-3}	3.32×10^{-2}

TABLE 14

Performance Evaluation of Interoperability with Three Equipped Aircraft

Metrics	XXT	XTX	TXX	XTT	TXT	TTX	XXX	TTT
Risk Ratio	7.44×10^{-2}	7.57×10^{-2}	7.74×10^{-2}	7.81×10^{-2}	7.92×10^{-2}	8.17×10^{-2}	7.23×10^{-2}	8.40×10^{-2}
Pr(NMAC AC1)	8.25×10^{-4}	8.25×10^{-4}	8.88×10^{-4}	8.45×10^{-4}	9.06×10^{-4}	9.02×10^{-4}	8.01×10^{-4}	9.29×10^{-4}
Pr(NMAC AC2)	8.15×10^{-4}	8.62×10^{-4}	8.50×10^{-4}	8.80×10^{-4}	8.56×10^{-4}	9.16×10^{-4}	8.07×10^{-4}	9.28×10^{-4}
Pr(NMAC AC3)	8.39×10^{-4}	8.38×10^{-4}	8.44×10^{-4}	8.80×10^{-4}	8.78×10^{-4}	9.08×10^{-4}	8.02×10^{-4}	9.42×10^{-4}
Pr(Induced NMAC)	4.86×10^{-5}	7.77×10^{-5}	8.91×10^{-5}	5.33×10^{-5}	6.99×10^{-5}	1.16×10^{-4}	4.44×10^{-5}	6.49×10^{-5}
Pr(Alert)	7.73×10^{-1}	7.73×10^{-1}	7.73×10^{-1}	8.13×10^{-1}	8.13×10^{-1}	8.13×10^{-1}	6.45×10^{-1}	8.02×10^{-1}
Pr(Strengthening)	5.31×10^{-2}	5.36×10^{-2}	5.37×10^{-2}	6.14×10^{-2}	6.22×10^{-2}	6.28×10^{-2}	4.05×10^{-2}	6.61×10^{-2}
Pr(Reversal)	1.18×10^{-2}	1.48×10^{-2}	1.89×10^{-2}	2.50×10^{-2}	2.88×10^{-2}	3.27×10^{-2}	1.49×10^{-2}	5.45×10^{-2}

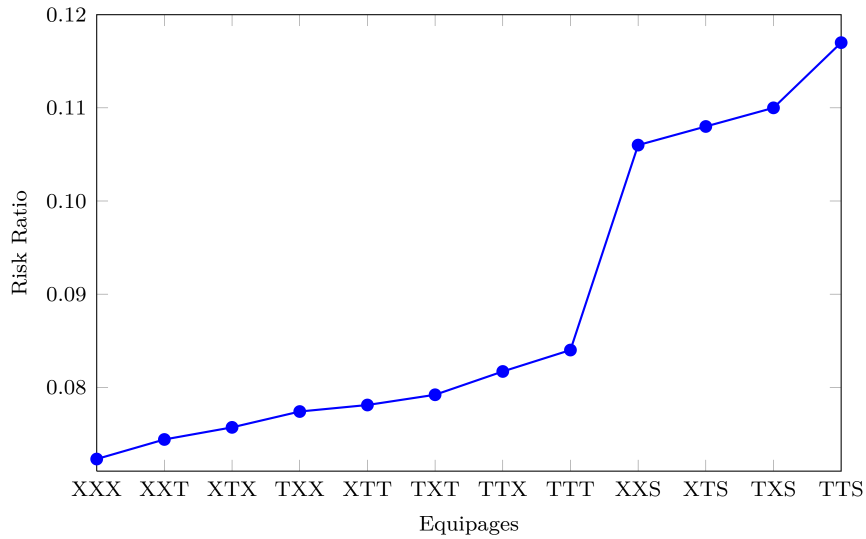


Figure 14. Summary of multithreat interoperability risk ratios.

4.8 DISCUSSION

This section explored options for expanding the ACAS X logic to multiple intruders. The result was improved performance over TCAS in almost all metrics evaluated. In the all equipped and single equipped scenarios, ACAS X outperformed TCAS in all metrics. However despite an overall gain in safety, improvements can be made to the two equipped scenario, focusing on induced NMACs.

Sandwich encounters are particularly challenging to resolve. Due to the shape of the individual policies, the costs cannot be fused to gain an earlier alert in multithreat situations. A potential solution to the current cost fusion approach could be to provide a negative cost to alerting once multiple intruders reach a certain threshold. However, the current method still outperforms TCAS on realistic scenarios derived from an encounter model.

The multithreat logic was designed to take advantage of improvements to the single threat logic. An initial look at stress testing the multihreat logic identified some areas needed for improvement. The focus of this report was on realistic three aircraft multithreat encounters. Further work should investigate how the logic behaves with additional intruders.

This page intentionally left blank.

5 CONCLUSIONS

This report has explored improving a collision avoidance system developed using dynamic programming. Prior to this report, research on ACAS X was focused on single unequipped intruders. Because expanding the same optimization approach to multiple intruders was intractable, approximate solutions were examined.

A simple coordination scheme was applied to the logic online. The forced cooperation scheme is based on the scheme used by TCAS. Simulations in this report demonstrate that ACAS X with the forced cooperation scheme can further reduce the risk of near mid-air collision beyond what is currently provided by TCAS while significantly reducing the alert rate. Interoperability between ACAS X and TCAS was demonstrated to be successful. Encounters between ACAS X and TCAS were shown to be significantly safer than TCAS against TCAS encounters. The robustness of the logic to slow responding and non-compliant aircraft was also analyzed and ACAS X outperformed TCAS.

Decomposition methods for multiple intruders were discussed and implemented in ACAS X. The idea of treating each intruder independently and fusing the individual actions together was derived from TCAS. However, complex rules were avoided by leveraging the expected costs associated with individual intruders. The approach does not affect performance in single intruder encounters. Multiple-intruder simulations demonstrated improvements over TCAS both in safety and operational performance.

The fusion technique does not allow ACAS X to alert earlier in multithreat environments. This property can cause alerts later than what is ideal in some sandwich encounters involving unequipped intruders. Further investigation into resolving sandwich encounters with unequipped intruders might be needed if the current performance is deemed unacceptable.

This report addressed the primary research questions left open from prior reports, including extending the optimized logic to coordinated encounters, multiple threat situations, and testing interoperability scenarios. Further improvements to the logic on specific encounter scenarios will be conducted in addition to the accommodation of different surveillance technologies and specific operational procedures.

This page intentionally left blank.

REFERENCES

[1] T.B. Billingsley, M.J. Kochenderfer, and J.P. Chryssathacopoulos, "Collision avoidance for general aviation," *IEEE Aerospace and Electronic Systems Magazine* 27, 4–12 (2012).

[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).

[3] R. Bellman, *Dynamic Programming*, Princeton: Princeton University Press (1957).

[4] M.L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, New York: Wiley (1994).

[5] D.P. Bertsekas, *Dynamic Programming and Optimal Control*, vol. 1, Belmont, MA: Athena Scientific, 3rd ed. (2005).

[6] O. Sigaud and O. Buffet (eds.), *Markov Decision Processes in Artificial Intelligence*, Hoboken, NJ: Wiley (2010).

[7] International Civil Aviation Organization, "Surveillance, radar and collision avoidance," in *International Standards and Recommended Practices*, vol. IV, annex 10, 4th ed. (2007).

[8] M.J. Kochenderfer and J.P. Chryssanthacopoulos, "Partially-controlled Markov decision processes for collision avoidance systems," in *International Conference on Agents and Artificial Intelligence* (2011).

[9] M.J. Kochenderfer, "Algorithm description document for the airborne collision avoidance system X threat resolution module," Federal Aviation Administration, TCAS Program Office (2012).

[10] A. Richards and J.P. How, "Aircraft trajectory planning with collision avoidance using mixed integer linear programming," in *American Control Conference* (2002).

[11] K.D. Bilimoria, "A geometric optimization approach to aircraft conflict resolution," in *AIAA Guidance, Navigation, and Control Conference and Exhibit* (2000).

[12] J.P. Chryssanthacopoulos and M.J. Kochenderfer, "Analysis of open-loop and closed-loop planning for aircraft collision avoidance," in *IEEE International Conference on Intelligent Transportation Systems* (2011).

[13] P.J. Gmytrasiewicz and P. Doshi, "A framework for sequential planning in multi-agent settings," *Journal of Artificial Intelligence Research* 24, 49–79 (2005).

[14] P. Doshi and P. Gmytrasiewicz, "Monto Carlo sampling methods for approximating interactive POMDPs," *Journal of Artificial Intelligence Research* 34, 297–337 (2009).

[15] C.V. Goldman and S. Zilberstein, "Optimizing information exchange in cooperative multi-agent systems," in *International Joint Conference on Autonomous Agents and Multiagent Systems* (2003).

[16] C.V. Goldman and S. Zilberstein, "Decentralized control of cooperative systems: Categorization and complexity analysis," *Journal of Artificial Intelligence Research* 22, 143–174 (2004).

[17] F.A. Oliehoek, *Value-based planning for teams of agents in stochastic partially observable environments:*, Ph.D. thesis, University of Amsterdam (2010).

[18] 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* 33(2), 487–499 (2010).

[19] J.P. Chryssanthacopoulos and M.J. Kochenderfer, "Decomposition methods for optimized collision avoidance with multiple threats," *Journal of Guidance, Control, and Dynamics* 35(2), 398–405 (2012).

[20] T.B. Billingsley, L.P. Espindle, and J.D. Griffith, "TCAS multiple threat encounter analysis," Massachusetts Institute of Technology, Lincoln Laboratory, Project Report ATC-359 (2009).