REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*
03-05-2011

2. REPORT TYPE
Article

3. DATES COVERED *(From - To)*
APR 2011 - MAY 2011

4. TITLE AND SUBTITLE
Decomposition Methods for Optimized Collision Avoidance with Multiple Threats

5a. CONTRACT NUMBER
FA8720-05-C-0002

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
James P. Chryssanthacopoulos and Mykel J. Kochenderfer

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
MIT Lincoln Laboratory
244 Wood Street
Lexington, MA 02420

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Federal Aviation Administration
800 Independence Avenue, SW
ATO-P
Washington, DC 20591

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

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT
DISTRIBUTION STATEMENT A. Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
Aircraft collision avoidance systems assist in the resolution of collision threats from nearby aircraft by issuing avoidance maneuvers to pilots. Encounters where more than one aircraft poses a threat, though rare, can be difficult to resolve because a maneuver that might resolve a conflict with one aircraft might induce conflicts with others. Recent efforts to develop robust collision avoidance systems for single-threat encounters have involved modeling the problem as a Markov decision process, discretizing the model, and applying dynamic programming to solve for the optimal avoidance strategy. Because the direct application of this methodology does not scale well to multiple threats, this paper evaluates a variety of decomposition methods that leverage the optimal avoidance strategy for single-threat encounters.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF: U			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON Zach Sweet
a. REPORT U	b. ABSTRACT U	c. THIS PAGE U	SAR	21	19b. TELEPHONE NUMBER *(include area code)* 781-981-5997

Decomposition Methods for Optimized Collision Avoidance with Multiple Threats

James P. Chryssanthacopoulos[1] and Mykel J. Kochenderfer[2]
Lincoln Laboratory, Massachusetts Institute of Technology, Lexington, Massachusetts 02420

Aircraft collision avoidance systems assist in the resolution of collision threats from nearby aircraft by issuing avoidance maneuvers to pilots. Encounters where more than one aircraft poses a threat, though rare, can be difficult to resolve because a maneuver that might resolve a conflict with one aircraft might induce conflicts with others. Recent efforts to develop robust collision avoidance systems for single-threat encounters have involved modeling the problem as a Markov decision process, discretizing the model, and applying dynamic programming to solve for the optimal avoidance strategy. Because the direct application of this methodology does not scale well to multiple threats, this paper evaluates a variety of decomposition methods that leverage the optimal avoidance strategy for single-threat encounters.

I. Introduction

Aircraft collision avoidance systems attempt to detect and resolve collision threats from nearby aircraft. Typically no more than one aircraft poses a threat at any given time in today's airspace, but if airspace densities continue to grow as expected, the ability to resolve multiple collision threats becomes increasingly important. Deciding the appropriate avoidance maneuver to issue in a multiple-

---

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

[1] Assistant Staff, Lincoln Laboratory, 244 Wood Street; chryssanthacopoulos@ll.mit.edu. Member AIAA.
[2] Technical Staff, Lincoln Laboratory, 244 Wood Street; mykelk@ll.mit.edu. Senior Member AIAA.

threat situation is more difficult than in a single-threat situation because attempts to resolve a conflict with one aircraft might induce conflicts with others.

Conflicts involving multiple threats can be resolved in either a pairwise or global manner. The pairwise method generates avoidance maneuvers to avoid each threat in isolation and issues the avoidance maneuver that achieves a compromise between them. The global method takes into account all aircraft simultaneously when choosing an avoidance maneuver. Pairwise methods can lead to suboptimal solutions, but they are generally less demanding computationally and can permit richer probabilistic models of aircraft behavior. The Traffic Alert and Collision Avoidance System (TCAS), the system currently mandated on all large transport aircraft, resolves multiple-threat encounters pairwise but makes modifications to the pairwise solution if necessary to avoid conflicts with other aircraft [1]. Other collision avoidance systems that use pairwise and global strategies are surveyed in [2].

Recent efforts to develop robust collision avoidance systems have involved modeling the problem as a Markov decision process (MDP) [3–5]. After discretizing the model, dynamic programming was used to solve for the optimal avoidance strategy that minimizes a cost metric. Past work has been limited to single-threat encounters. Solving for the globally optimal solution for multiple-threat encounters would require adding additional state variables to the model for each additional intruder. Because the number of discrete states grows exponentially with the number of variables in the model, solving for the optimal strategy in this way is infeasible.

This paper discusses computationally tractable methods for approximately solving the MDP for multiple-threat encounters through pairwise decomposition. One method is to use a command arbitration strategy, similar to TCAS, that selects between maneuvers optimized to avoid each threat in isolation [6]. Another method is to fuse the utilities of the various avoidance maneuvers associated with avoiding different threats [7, 8]. Various command arbitration and utility fusion methods are compared in simulation against the existing TCAS logic and a baseline system that employs a global method.

The organization of this paper is as follows. Section II reviews the single-threat collision avoidance problem and solution. The multiple-threat problem and various solutions are presented in Sec.

III. Section IV summarizes the results of the simulation study. Section V concludes the paper and outlines areas of future work.

II. Single-threat Collision Avoidance

Previous work has shown how to model the single-threat collision avoidance problem as a Markov decision process (MDP) [9, 10]. The own aircraft, equipped with a collision avoidance system, must avoid a single unequipped intruder. The collision avoidance system alerts pilots to potential threats by issuing resolution advisories instructing the pilots how to adjust their vertical rate to avoid conflict.

An MDP is defined by the tuple $(\mathcal{S}, \mathcal{A}, R, T)$. The sets $\mathcal{S}$ and $\mathcal{A}$ are a finite set of states and a finite set of actions, respectively. The reward function $R(s, a)$ is the immediate reward when taking action a in state s. The state-transition function $T(s, a, s')$ is the probability of transitioning from state s to state s' after taking action a.

A policy is a mapping from states to actions that defines what action to execute from each state. The solution to an MDP is a policy π^* that, if followed, maximizes the expected sum of immediate rewards, or expected utility, from any given state. The optimal policy is closely related to the optimal state-action utility function $U^*(s, a)$, which is the expected utility when starting in state s, taking action a for one time step, and then continuing with the actions prescribed by π^*. It obeys the following recursion:

$$U^*(s, a) = R(s, a) + \sum_{s' \in \mathcal{S}} T(s, a, s')U^*(s'), \tag{1}$$

where $U^*(s) = \max_{a \in \mathcal{A}} U^*(s, a)$. The state-action utility function can be computed using a dynamic programming algorithm known as value iteration. Value iteration starts with an initial estimate of U^* and updates the estimate by repeated application of Eq. (1) until the estimate converges. The optimal action from each state s is given by

$$\pi^*(s) = \arg\max_{a \in \mathcal{A}} U^*(s, a). \tag{2}$$

The remainder of this section discusses how to formulate the single-threat collision avoidance problem as an MDP.

A. Resolution Advisories

In the single-threat problem, the system can issue one of three different initial advisories: climb at least 1500 ft/min, descend at least 1500 ft/min, or level-off with a vertical rate between ± 100 ft/min. Following the initial advisory, the system can either terminate, strengthen, reverse, or level-off. A strengthening increases the minimum target vertical rate to 2500 ft/min, and a reversal changes the minimum target vertical rate to 1500 ft/min in the opposite direction. The advisories, as well as the decision to not alert, constitute the action set $\mathcal{A}$.

B. Dynamic Model

The state of the system is described by the following variables: the altitude of the intruder relative to the own aircraft, the vertical rate of the own aircraft, the vertical rate of the intruder, the state of the resolution advisory, and the east and north positions and velocities of the aircraft. The state of the resolution advisory is a discrete variable that allows the system to track which advisory is currently active, if any, and whether the pilot is responding to it.

The pilot responds immediately to the first resolution advisory issued with probability 1/6 and remains unresponsive for one time step otherwise. The pilot responds to an initial advisory by applying a 1/4 g acceleration to meet the target minimum vertical rate. Should the initial advisory remain in effect at the next time step, the pilot responds with probability 1/6 if he has not responded already. For a given advisory, therefore, the response delay follows a geometric distribution where the pilot responds in 5 s on average. When the pilot receives a subsequent advisory, such as a strengthening or reversal, he responds to it with probability 1/4 and neglects all advisories otherwise, regardless of whether he was responding to the previous advisory. The response to a subsequent advisory is a 1/3 g maneuver to reach the target minimum vertical rate. When the system stops alerting, the pilot stops responding immediately. Further details regarding the pilot response model can be found in [11].

When the pilot is not responding to an advisory, the vertical acceleration of the aircraft is modeled as a zero-mean Gaussian with a standard deviation of $3\,\mathrm{ft/s^2}$. The aircraft also experience

random horizontal accelerations selected independently from a zero-mean Gaussian with a standard deviation of $8\,\text{ft/s}^2$.

Because several of the variables in the collision avoidance problem are continuous, discretization is required to generate the set of discrete states S and the discrete state-transition function $T(s, a, s')$. The experiments in this paper used the scheme from [5] to discretize the state space and estimate the discrete transition probabilities.

C. Reward Function

The reward function R penalizes conflicts and alerting. Unit cost (negative reward) is incurred when the aircraft come into conflict, defined to be when the intruder comes within 1000 ft horizontally and 100 ft vertically of the own aircraft. To reduce unnecessary alerts, a cost of 0.001 is incurred when an alert is first issued. Costs of 0.009 and 0.01 are also incurred any time an advisory is strengthened or reversed, respectively.

D. Optimal Policy

The policy π^* specifies the action (no alert or issue one of the various advisories) to execute from every state. However, computing the optimal policy even for the simple single-threat problem is challenging. Because the single-threat model is high dimensional, discretizing the model at a suitable resolution results in an exorbitant amount of discrete states (approximately 1.15×10^{11}), making value iteration an impractical solution method. To reduce the computational complexity, this paper uses the solution method introduced in [4] to approximately solve for the optimal policy.

The approximation method decomposes the full problem into controlled and uncontrolled subproblems that are solved independently using dynamic programming. The controlled subproblem is an MDP that models the relative vertical motion of the aircraft controllable by the collision avoidance system. The uncontrolled subproblem corresponds to the relative horizontal motion that is assumed to be unaffected by resolution advisories. Discretization results in only 6.45 million controlled states and 730,000 uncontrolled states. Solving the controlled and uncontrolled subproblems offline requires approximately 4 min on a single 3 GHz Intel Xeon core.

Figure 1 shows the approximately optimal policy, optimized to an alert cost of 0.01, for two particular encounter scenarios. In Fig. 1(a), the aircraft start 8000 ft apart horizontally and begin flying head-on with ground speeds of 100 ft/s. Both aircraft are flying level, the intruder constantly at 43,000 ft. The position of the intruder is shown on the right. No resolution advisory has yet been issued. The plot indicates the action that would be executed some time into the encounter at a particular altitude. For instance, when 25 s has elapsed since the beginning of the encounter and the own aircraft is flying at 43,200 ft, the optimal action is to issue a climb advisory. The own aircraft achieves minimal horizontal separation with the intruder 40 s into the encounter. In Fig. 1(b), the encounter scenario is identical except the intruder is descending at 1500 ft/min. The alerting region is pushed down as the system must alert at lower altitudes to prevent the intruder from descending into the own aircraft from above.

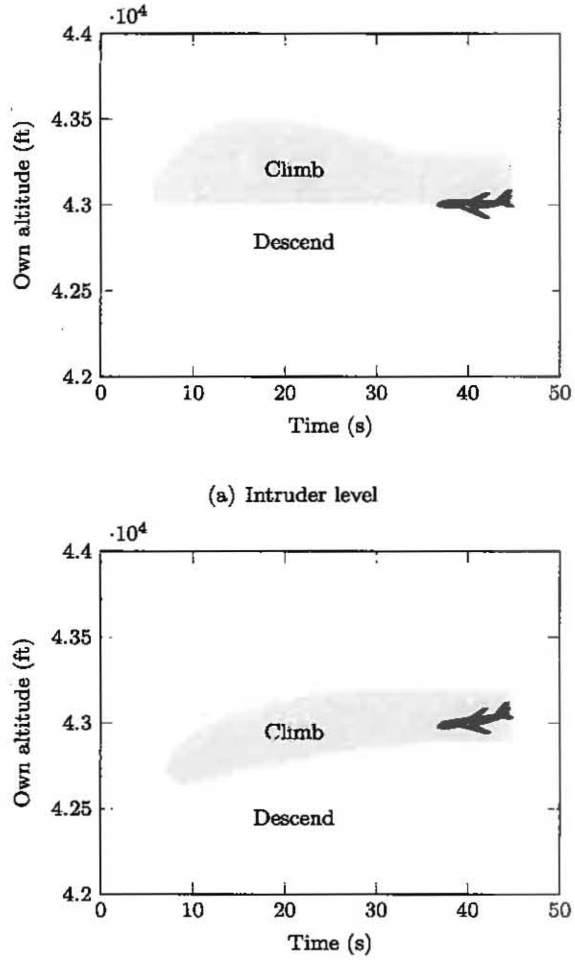
III. Multiple-Threat Collision Avoidance

Extending the MDP model of the previous section to incorporate more than one intruder is straightforward. Adding an additional intruder requires introducing new variables to capture the relative altitude, vertical rate, and horizontal position and velocity. Adding only one additional intruder increases the number of controlled states from 6.45×10^6 states to 1.17×10^{11} states. A third intruder would require 2.11×10^{15} states. Scaling the MDP to multiple intruders in this way is currently computationally infeasible. Approximate solutions can be found, however, using decompositions methods such as command arbitration and utility fusion. This section also discusses a global method against which the decomposition methods can be compared.

A. Command Arbitration

Command arbitration computes, for each intruder i, the optimal action to take $\pi^*(s_i)$ assuming that intruder i is the only threat. Here s_i denotes the component of state s that describes the motion of the own aircraft and intruder i only. This information is used to choose actions.

This paper investigates two command arbitration methods. In the first, the action of the closest intruder (in slant range) is executed. Because the closest intruder often is the most immediate threat, prioritizing its action in this way seems sensible. Resolving conflicts sequentially may be acceptable



(a) Intruder level

(b) Intruder descending at 1500 ft/min

Fig. 1 Single-threat policy plots for two encounter scenarios.

much of the time, but it is easy to generate situations in which this approach fails (as shown in Sec. IV).

The second command arbitration method chooses between the various actions using an arbitration strategy similar to TCAS. TCAS computes provisional resolution advisories for each intruder in isolation using its single-threat logic. If only one intruder results in a resolution advisory, that advisory is executed. If there are multiple advisories with the same sense (i.e., upward or downward), TCAS simply selects the individual advisory commanding the greatest vertical rate. When the senses disagree, TCAS uses a set of rules to identify either a single sense appropriate against all intruders or whether it should issue a level-off advisory. The TCAS-like arbitration method in this work does not emulate this set of rules exactly, but captures the important properties.

Figure 2 shows the policies for the command arbitration methods. The encounter scenario is similar to the one presented in Fig. 1(a), except now two intruders, separated 400 ft in altitude, are approaching the own aircraft head-on. Their positions are shown on the right. Figure 2(a) shows the policy for the closest arbitration method. The own aircraft switches between the single-threat policies depending on which intruder is closer without consideration of how it will impact the other intruder. When the own aircraft is between the intruders in altitude but closer to the top intruder, the recommended action is to issue a descend advisory. If following the descend advisory leads to conflict with the bottom intruder, the advisory may be reversed later.

Figure 2(b) shows the policy for the TCAS-like arbitration method. Unlike closest arbitration, the policy may recommend leveling off when the own aircraft is flying between the intruders. When the own aircraft is flying at 43,100 ft 15 s into the encounter, the policy says to level-off instead of descend as in closest arbitration because it knows that if it climbs or descends there may be insufficient separation with one of the intruders.

B. Utility Fusion

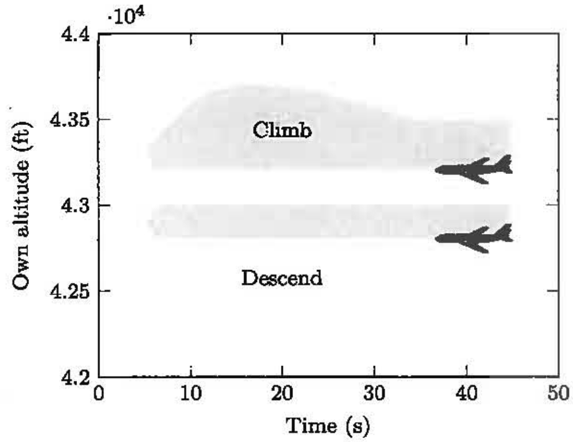
Utility fusion computes, for each intruder i, the optimal state-action utilities $U^*(s_i, a)$ for all actions a, again assuming that intruder i is the only threat. The utility $U^*(s_i, a)$ is a measure of how effective action a is in resolving a conflict with intruder i alone, assuming the optimal policy for that intruder is followed in the future. The state-action utilities from multiple intruders are fused to arrive at the optimal state-action utility function $U^*(s, a)$. Fusing the utilities requires defining a function f that combines utilities associated with multiple intruders. That is,

$$U^*(s, a) = f(U^*(s_1, a), \ldots, U^*(s_N, a)), \tag{3}$$

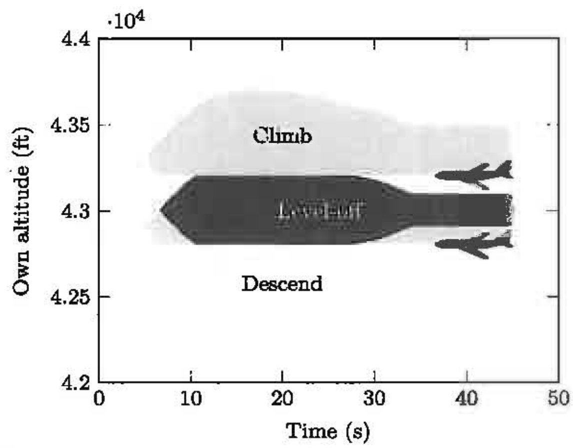
where N is the number of intruders.

This paper investigates two utility fusion methods. The first method, the max-sum strategy, defines f to be a summation:

$$f = \sum_i U^*(s_i, a). \tag{4}$$



(a) Closest arbitration



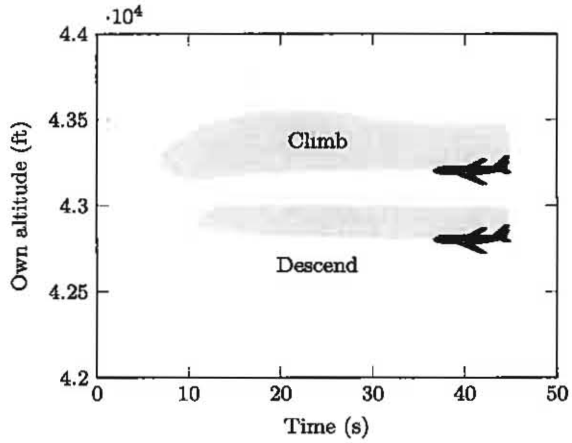
(b) TCAS-like arbitration

Fig. 2 Multiple-threat policy plots using command arbitration.

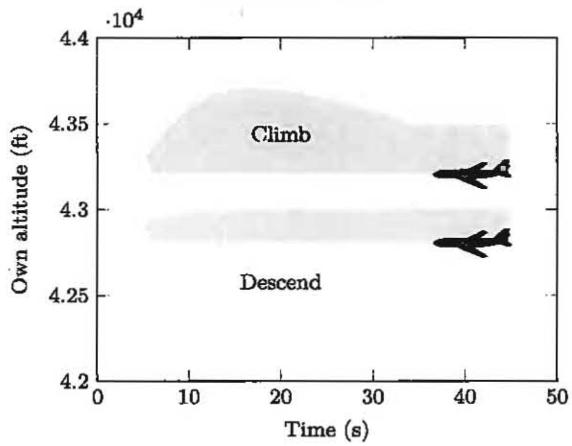
Defining f in this way leads to counting alert costs multiple times. The cost of alerting, for example, would be reflected in the state-action utilities for each intruder. Adding these utilities together amounts to incurring the alert cost multiple times, though in reality the collision avoidance system can only alert once at any given time. This may cause the system to delay issuing the alert. Waiting a long time to issue an alert is undesirable because as more time elapses the own aircraft has fewer available options to successfully resolve the conflict. When more intruders are present, the importance of alerting earlier is magnified.

The second method, the max-min strategy, avoids accumulating the cost of alerting for each intruder by defining f to be the minimum state-action utility over all intruders:

$$f = \min_i U^*(s_i, a). \tag{5}$$



(a) Max-sum fusion



(b) Max-min fusion

Fig. 3 Multiple-threat policy plots using utility fusion.

Figure 3(a) and Fig. 3(b) show the policies for the max-sum and max-min fusion methods, respectively. As expected, counting alert costs multiple times makes the alerting region for the max-sum method smaller. The alerting region for the max-min method is similar to closest arbitration. The max-min method delays alerting a little longer when the own aircraft is exactly between the intruders.

Table 1 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 utility for each intruder and for each action. The max-sum method issues the climb advisory because it is very effective in preventing conflict with the second intruder, even though following the climb may lead to conflict with the first intruder. The max-min method selects

Table 1 **Utilities for a simple two-intruder example**

intruder	no alert	climb	descend
1	-3	-1	3
2	-5	10	2
sum	-8	**9**	5
min	-5	-1	**2**

the descend action because the lower utility for executing the descend is 2 while the lower utility for executing the climb is -1.

One important property of the decomposition methods is that they do not begin alerting 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 is confirmed by observing that the multiple-threat policy plots of Fig. 2 and Fig. 3 do not extend any further to the left than their single-threat counterpart shown in Fig. 1. This may be an undesirable feature because in multiple-threat encounters it may be necessary to alert a little earlier in order to pass above or below all intruders.

C. Global Method

This paper compares the decomposition methods to a collision avoidance system that employs a global resolution method. Unlike decomposition methods, which compute the actions or utilities optimized for each intruder in isolation and then combine the information, global methods optimize for all intruders simultaneously. As mentioned earlier, global methods typically cannot accommodate the rich probabilistic models pairwise methods are able to use. This paper uses a deterministic aircraft model to attempt to find a sequence of advisories that results in a path that does not violate the protected zones around the other aircraft. Several different methods can be used to determine conflict-free paths, including mixed-integer linear programming [12] or geometric optimization [13]. The experiments in this paper use an extension of the method discussed in [14]. The system issues advisories to barely miss the protected zones of the intruders. If the protected zones can be evaded by alerting later on, the alert is delayed.

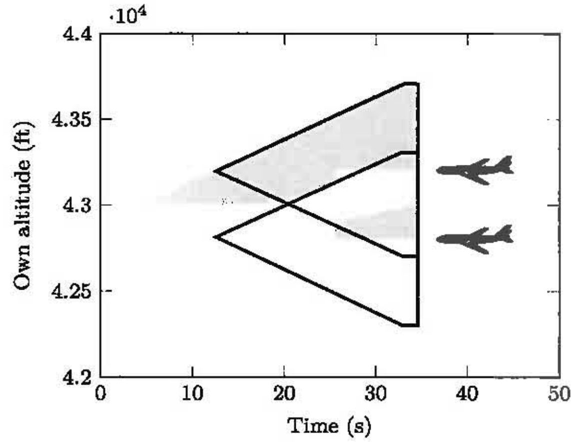


Fig. 4 Policy plot using a global approach to multiple-threat collision avoidance.

Figure 4 shows the policy for the global method. The policy was computed using a deterministic pilot response model in which the pilot responds to all initial advisories in exactly 5 s and all subsequent advisories in exactly 3 s. The protected zones around the intruders were cylinders with heights of 1000 ft and diameters of 5000 ft. These protected zones usually need to be large to compensate for the fact that the deterministic models do not capture the uncertainty in the future trajectories of the aircraft. Outlined in black are the single-threat alerting regions when each intruder is considered independently. Unlike the pairwise decomposition methods, the alerting region extends further out than both of the individual alerting regions, allowing the own aircraft sufficient time to pass above or below the intruders even when initially between them in altitude.

IV. Results

The various decomposition methods discussed in the previous section were evaluated in simulation to assess their performance. In simulation, the collision avoidance system is equipped with imperfect sensors, which introduce uncertainty in the current state of the environment. For example, due to imperfections in the sensors, measurements of the range and bearing to the intruder may be corrupted with noise, leading to uncertainty in the intruder position. Uncertainty may also arise due to an inherent limitation in the sensors. For example, even with perfect sensing of the own aircraft vertical rate, there is still uncertainty in the response of the pilot to resolution advisories. The pilot may be descending, for example, because he is responding to a descend advisory or simply

due to random perturbations. When the state is not fully observable, recursive Bayesian estimation can be used to infer a probability distribution over the state space, called a belief state, from the sequence of observations. In multiple-threat encounters, separate belief states must be maintained for each intruder.

An MDP model that incorporates state uncertainty is called a partially observable MDP (POMDP). As the state is no longer fully observable, the policy becomes a mapping from belief states to actions. Analogous to the MDP case, the optimal policy is one that maximizes belief-action utility from every belief state. Finding the exact optimal policy is difficult in general, but a number of different methods may be used to arrive at an approximate solution [15–17]. The QMDP method, for example, approximates the optimal belief-action utilities as a weighted sum of optimal state-action utilities assuming full observability [18]. The belief-action utility for intruder i is, according to the QMDP method, approximately

$$U^*(b_i, a) \approx \sum_{s_i} b_i(s_i) U^*(s_i, a), \tag{6}$$

where b_i is the belief state for intruder i. The approximately optimal action $\pi^*(b_i)$ for intruder i is therefore $\arg\max_{a \in \mathcal{A}} U^*(b_i, a)$. The QMDP method accounts for the present uncertainty in the state as encoded by the belief state, but fails to account for future state uncertainty. This amounts to assuming that at the next time step the world becomes fully observable. The QMDP method has been shown to work well on single-threat collision avoidance [11, 19]. In the experiments in this paper, the QMDP method is used to compute $\pi^*(b_i)$ and $U^*(b_i, a)$, which are in turn used by the decomposition methods to select actions.

A. Performance Statistics

The decomposition methods were evaluated against a set of 500,000 encounters randomly generated from an encounter model. The positions and velocities of the aircraft were available to the methods without error. The encounter model, inferred from recorded radar data, is statistically representative of encounters between three aircraft observed in the U.S. airspace [20]. Importance sampling was used to generate the encounters from the model so that approximately half of the encounters result in near collision without collision avoidance.

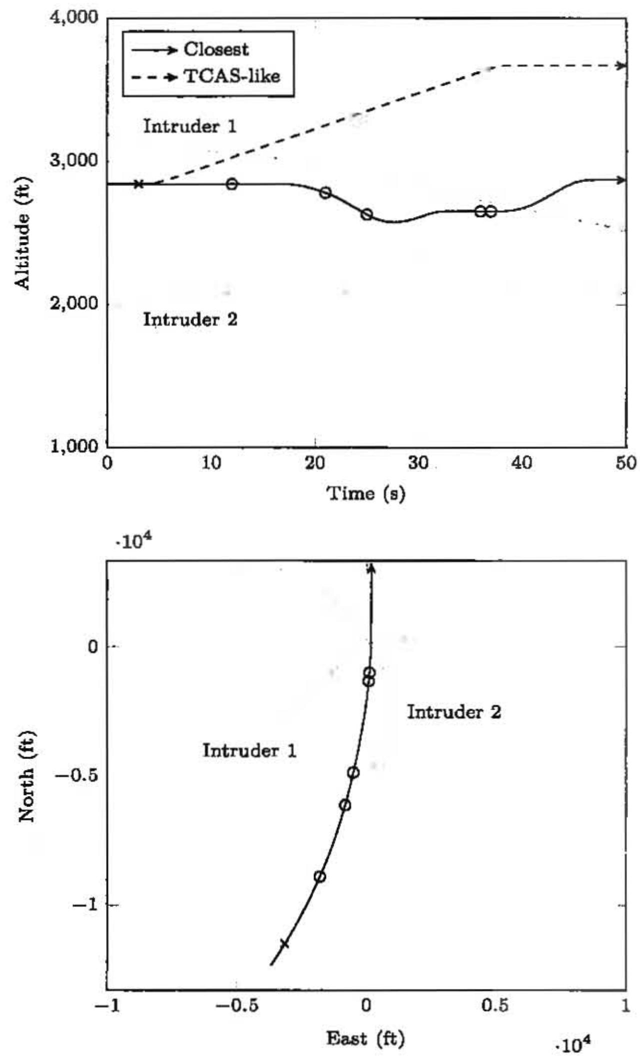
Table 2 Performance statistics

	Command Arbitration		Utility Fusion			
	Closest	TCAS-like	Max-sum	Max-min	Global	TCAS
Pr(NMAC)	$8.354 \cdot 10^{-3}$	$2.916 \cdot 10^{-3}$	$1.326 \cdot 10^{-3}$	$1.152 \cdot 10^{-3}$	$3.638 \cdot 10^{-3}$	$7.852 \cdot 10^{-3}$
Pr(Alert)	0.648	0.690	0.593	0.690	0.562	0.753
Pr(Strengthening)	0.138	$8.806 \cdot 10^{-2}$	$6.552 \cdot 10^{-2}$	$8.896 \cdot 10^{-2}$	0.425	$5.460 \cdot 10^{-2}$
Pr(Reversal)	$4.912 \cdot 10^{-3}$	$5.616 \cdot 10^{-3}$	$6.422 \cdot 10^{-3}$	$7.310 \cdot 10^{-3}$	$4.968 \cdot 10^{-3}$	$6.872 \cdot 10^{-3}$

Table 2 summarizes the results of the simulation. It reports the probability that an encounter results in a near mid-air collision (NMAC) and the probabilities that the methods alert, strengthen, and reverse in an encounter. An NMAC occurs when either intruder comes within 500 ft horizontally and 100 ft vertically of the own aircraft [21]. The probability of NMAC without collision avoidance is 0.0982. The table also shows the statistics for the global method and the current version of TCAS (Version 7.1). The standard errors associated with each of the estimates were also calculated and were found to be small in relation to the actual estimates. The standard error was between 0.1% and 4% the size of the estimate.

TCAS-like arbitration is almost three times safer than closest arbitration. Figure 5 shows an example encounter where closest arbitration fails to prevent NMAC. The own aircraft is flying between the two intruders in altitude and, because intruder 1 is initially closer in range, receives a descend advisory, abbreviated DES1500. Some seconds later the system strengthens the advisory (SDES2500). As descending may cause a conflict with intruder 2, the descend advisory is reversed to a climb (SCL1500). While the climb advisory is being executed, the advisory is terminated, but later a climb advisory is reissued and strengthened to prevent conflict with intruder 1. This tendency to reverse and strengthen the advisory multiple times, though rare, may be operationally unacceptable. Also shown in Fig. 5 is the behavior of TCAS-like arbitration. It initially alerts earlier than closest arbitration, issuing a climb to safely pass above both intruders. It is interesting to note that the TCAS logic behaves similarly on this encounter, issuing a climb advisory followed by a "Do Not Descend" advisory to successfully resolve the encounter.

The utility fusion methods are over twice as safe as the command arbitration methods while alerting at a lower or comparable rate. The max-min method is safer than the max-sum method but



	Closest		TCAS-like
o	DES1500	×	CL1500
o	SDES2500		
o	SCL1500		
o	CL1500		
o	SCL2500		

Fig. 5 Example encounter using command arbitration.

it alerts more often and generally earlier, requiring it to strengthen and reverse more. By cutting the alert cost in half, the max-sum method achieves an NMAC probability of 1.132×10^{-3} with an alert probability of 0.6511. Though the global method alerts less frequently than the utility fusion methods, it has a much higher NMAC probability and also tends to strengthen the advisory much more frequently. All the decomposition methods, with the exception of closest arbitration, result

in greater safety than TCAS with a lower alert rate and comparable strengthening and reversal rates. Using the max-sum method over TCAS, for example, reduces the NMAC probability by 83%, the alert probability by 21%, and the reversal probability by 6.5% while only increasing the strengthening probability by 20%.

B. Stress Test

Although encounters between more than three aircraft are very rare, before a collision avoidance system can be adopted for use in actual aircraft operations, it must be shown to handle encounters with a potentially large number of intruders. Figure 6 shows how the max-min method resolves an encounter with four intruders. The intruders are initially evenly distributed (with some variation) around the own aircraft so that, on average, all aircraft will converge near the center in about 40 s. The accelerations of the aircraft are white Gaussian noise sampled every second. Although this simple model may not be a realistic representation of how encounters with many intruders evolve in the airspace, it does provide a way to stress test the systems to ensure that they do not behave unusually when faced with more intruders.

Figure 7 illustrates the performance of the decomposition methods as the number of intruders is increased. The performance of TCAS and of the global method are also shown as baselines. Each point on the curves was estimated from 100,000 simulations. All decomposition methods alert approximately 30% more often when the number of intruders is increased from 2 to 9. The percent increase in the probability of NMAC, however, is lower for the closest arbitration and max-min methods than it is for the TCAS-like arbitration and max-sum methods. The max-sum method nonetheless achieves a similar level of safety as the max-min method with lower alert, strengthening, and reversal rates. In terms of safety and alert rates, the utility fusion methods are consistently better than TCAS for this simple white-noise model. The black dashed line indicates the probability of NMAC without collision avoidance. Even in the presence of many intruders, the methods can still improve safety.

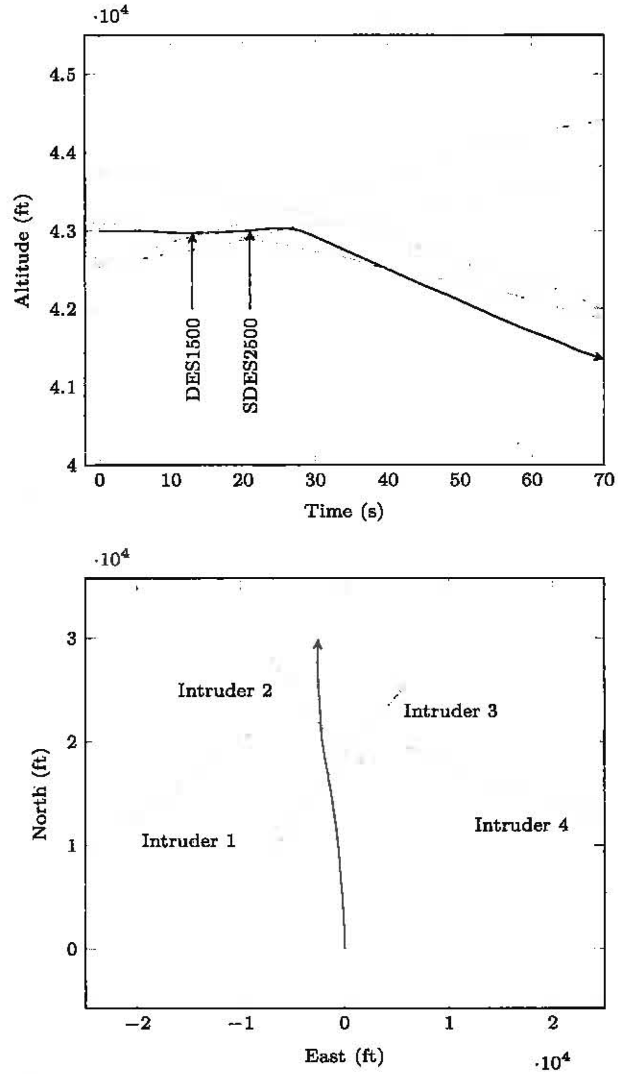


Fig. 6 Example encounter between five aircraft.

C. State Uncertainty

In the previous experiments, the collision avoidance system had perfect state information regarding the positions and velocities of the aircraft. Table 3 shows how the various methods perform when the own aircraft is equipped with noisy sensors. The collision avoidance system receives measurements of the intruders using a beacon radar similar to the one currently employed by TCAS. The radar measures the slant range, bearing, and altitude of all intruders. The slant range error is modeled as a zero-mean Gaussian with 50 ft standard deviation. The bearing error is modeled as a zero-mean Gaussian with 10° standard deviation. The intruder altitude is quantized to 25 ft increments. The own aircraft altitude, vertical rate, and heading are assumed to be available through

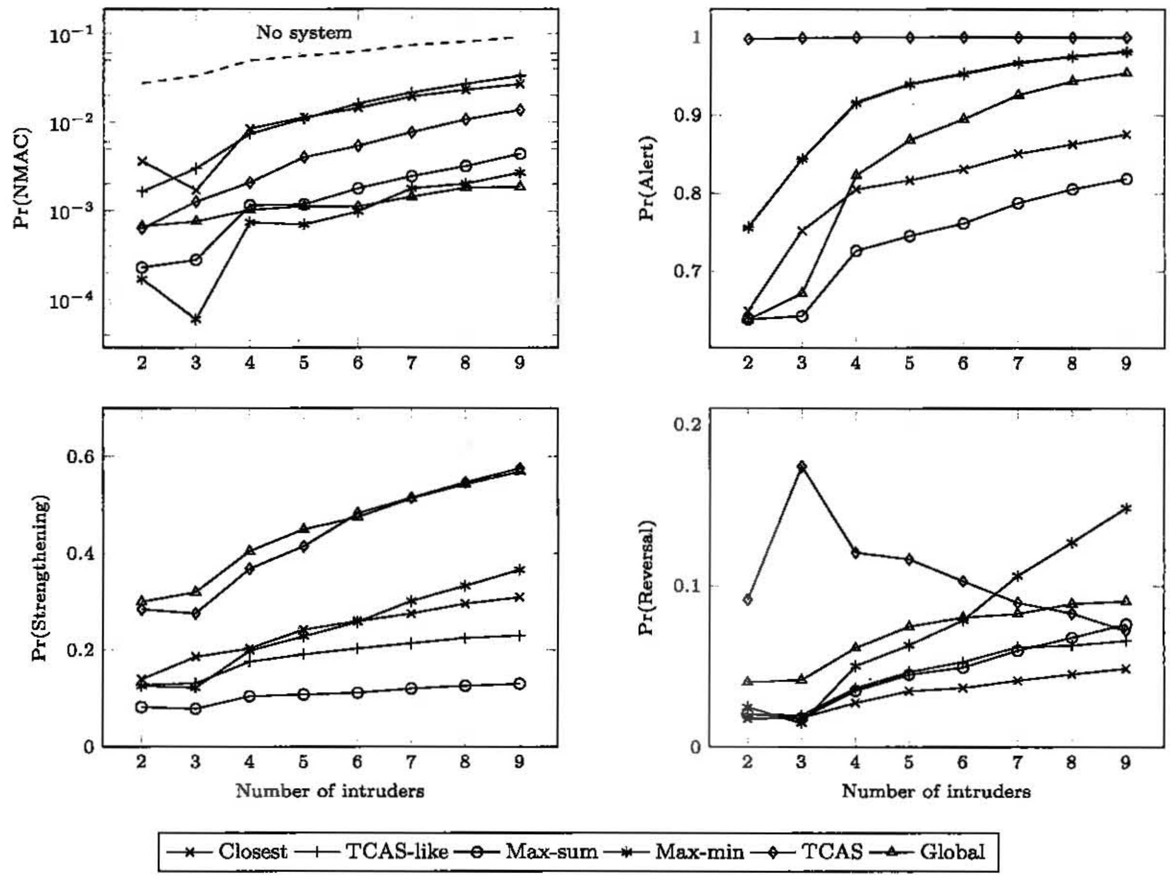


Fig. 7 Probability of NMAC, alert, strengthening, and reversal as the number of intruders increases.

Table 3 Performance statistics with a TCAS-like sensor

	Command Arbitration		Utility Fusion			
	Closest	TCAS-like	Max-sum	Max-min	Global	TCAS
Pr(NMAC)	$7.750 \cdot 10^{-3}$	$5.856 \cdot 10^{-3}$	$2.964 \cdot 10^{-3}$	$2.418 \cdot 10^{-3}$	$7.074 \cdot 10^{-3}$	$7.520 \cdot 10^{-3}$
Pr(Alert)	0.699	0.752	0.641	0.752	0.580	0.764
Pr(Strengthening)	0.129	0.102	$8.018 \cdot 10^{-2}$	0.106	0.493	$5.276 \cdot 10^{-2}$
Pr(Reversal)	$6.898 \cdot 10^{-3}$	$9.406 \cdot 10^{-3}$	$9.370 \cdot 10^{-3}$	$1.205 \cdot 10^{-2}$	$6.046 \cdot 10^{-3}$	$7.844 \cdot 10^{-3}$

the onboard avionics. Horizontal and vertical trackers are used to infer the belief state. Further detail can be found in [19].

TCAS-like arbitration, max-sum, and max-min methods have a greater NMAC rate with sensor noise than without. Beyond slightly increasing the alert rate, closest arbitration is mostly unaffected. Similarly, the range and bearing noise have little effect on TCAS performance. Nonetheless, the

utility fusion methods are still able to achieve a greater level of safety than TCAS with a lower alert rate.

V. Conclusions and Future Work

This paper discussed decomposition methods for aircraft collision avoidance with multiple threats. Like the single-threat problem, the multiple-threat collision avoidance problem can be framed as a Markov decision process (MDP). Unfortunately, the solution method for the single-threat problem does not scale well to the multiple-threat problem, which requires many more variables to be modeled. This paper presented decomposition methods for solving the MDP that leverage the solution to the single-threat problem.

The results showed that decomposition methods, though suboptimal, can be effective. In realistic three-aircraft simulations, the decomposition methods were able to outperform the current version of the Traffic Alert and Collision Avoidance System, the system in use on aircraft today. Utility fusion methods performed better than command arbitration methods by fusing utilities associated with different intruders instead of simply selecting between candidate actions. The utility fusion methods can be as safe as a baseline global method while strengthening far less often.

In the collision avoidance problem presented in this paper, only one aircraft was equipped with a collision avoidance system. In encounters between two or more aircraft equipped with collision avoidance systems, the maneuvers must be carefully coordinated so that, for example, two aircraft do not both receive climb advisories and induce collision. In general, proper coordination can greatly enhance safety. Future work will show several ways to extend the MDP model to handle coordination and will explore the impact of coordination in multiple-threat encounters where some or all of the intruders are equipped with collision avoidance systems.

VI. Acknowledgements

This work is the result of research sponsored by the TCAS Program Office at the Federal Aviation Administration. The authors appreciate the support provided by the TCAS Program Manager, Neal Suchy. This work has benefited from conversations with J. Daniel Griffith, Leslie P. Kaelbling, and Tomás Lozano-Pérez.

References

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

[2] J. K. Kuchar and L. C. Yang, "A review of conflict detection and resolution modeling methods," *IEEE Transactions on Intelligent Transportation Systems*, vol. 1, no. 4, pp. 179–189, Dec. 2000. doi: 10.1109/6979.898217.

[3] M. J. Kochenderfer and J. P. Chryssanthacopoulos, "A decision-theoretic approach to developing robust collision avoidance logic," in *IEEE International Conference on Intelligent Transportation Systems*, Sep. 2010, pp. 1837–1842. doi: 10.1109/ITSC.2010.5625063.

[4] M. J. Kochenderfer and J. P. Chryssanthacopoulos, "Partially-controlled Markov decision processes for collision avoidance systems," in *International Conference on Agents and Artificial Intelligence*, Rome, Italy, 2011, pp. 62–70.

[5] M. J. Kochenderfer and J. P. Chryssanthacopoulos, "Robust airborne collision avoidance through dynamic programming," Massachusetts Institute of Technology, Lincoln Laboratory, Project Report ATC-371, Jan. 2011.

[6] R. A. Brooks, "Achieving artificial intelligence through building robots," Massachusetts Institute of Technology, A. I. Memo 899, 1986.

[7] J. K. Rosenblatt, "Optimal selection of uncertain actions by maximizing expected utility," *Autonomous Robots*, vol. 9, pp. 17–25, 2000.

[8] S. J. Russell and A. Zimdars, "Q-decomposition for reinforcement learning agents," in *International Conference on Machine Learning*, T. Fawcett and N. Mishra, Eds., Washington DC, Aug. 2003, pp. 656–663.

[9] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, ser. Wiley series in probability and mathematical statistics. New York: Wiley, 1994, pp. 17–25.

[10] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, 3rd ed. Belmont, MA: Athena Scientific, 2005, vol. 1, pp. 19–56.

[11] J. P. Chryssanthacopoulos and M. J. Kochenderfer, "Collision avoidance system optimization with probabilistic pilot response models," in *American Control Conference*, 2011, to appear.

[12] A. Richards and J. P. How, "Aircraft trajectory planning with collision avoidance using mixed integer linear programming," in *American Control Conference*, vol. 3, May 2002, pp. 1936–1941. doi: 10.1109/ACC.2002.1023918.

[13] K. D. Bilimoria, "A geometric optimization approach to aircraft conflict resolution," in *AIAA Guidance,*

Navigation, and Control Conference and Exhibit, Aug. 2000, AIAA Paper 2000-4265.

[14] 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, under review.

[15] M. T. J. Spaan and N. Vlassis, "A point-based POMDP algorithm for robot planning," in *IEEE International Conference on Robotics and Automation*, vol. 3, no. 2399–2404, 2004. doi: 10.1109/ROBOT.2004.1307420.

[16] T. Smith and R. G. Simmons, "Point-based POMDP algorithms: Improved analysis and implementation," in *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence*. Arlington, VA: AUAI Press, 2005, pp. 542–547.

[17] H. Kurniawati, D. Hsu, and W. Lee, "SARSOP: Efficient point-based POMDP planning by approximating optimally reachable belief spaces," in *Robotics: Science and Systems IV*. Cambridge, MA: MIT Press, 2008, pp. 65–72.

[18] M. L. Littman, A. R. Cassandra, and L. P. Kaelbling, "Learning policies for partially observable environments: Scaling up," in *Proceedings of the Twelfth International Conference on Machine Learning*. San Francisco, CA: Morgan Kaufmann, 1995, pp. 362–370.

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

[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, Oct. 2009.

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