JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 34, No. 4, July–August 2011

Accounting for State Uncertainty in Collision Avoidance

James P. Chryssanthacopoulos* and Mykel J. Kochenderfer†
Lincoln Laboratory, Massachusetts Institute of Technology, Lexington, Massachusetts 02420

DOI: 10.2514/1.53172

An important consideration in the development of aircraft collision avoidance systems is how to account for state uncertainty due to sensor limitations and noise. However, many collision avoidance systems simply use point estimates of the state instead of leveraging the full posterior state distribution. Recently, there has been work on applying decision-theoretic methods to collision avoidance, but the importance of accommodating state uncertainty has not yet been well studied. This paper presents a computationally efficient framework for accounting for state uncertainty based on dynamic programming. Examination of characteristic encounters and Monte Carlo simulations demonstrates that properly handling state uncertainty rather than simply using point estimates can significantly enhance safety and improve robustness to sensor error.

I. Introduction

COLLISION avoidance systems for aircraft, manned or autonomous, rely upon imperfect sensor measurements to avoid traffic. Sensor limitations result in state uncertainty and can make deciding when and how to maneuver difficult. Because collision avoidance systems are safety critical, taking into account state uncertainty can be very important. However, relatively little attention has been paid to principled techniques for accommodating state uncertainty in collision avoidance. Most systems largely rely upon point estimates of the state to make decisions [1–3].

Decision making with state uncertainty has been well studied in the artificial intelligence and operations research communities [4], but the theory and techniques have only recently been applied to aircraft collision avoidance system development. The standard way to frame decision problems in stochastic environments with state uncertainty is as a partially observable Markov decision process (POMDP) [5–7]. In a POMDP, the decision maker tries to maximize its expected utility with limited knowledge of the world, encoded in a belief state. Although solving a POMDP is a rigorous way of accounting for state uncertainty when making decisions, it can be difficult to scale to large problems.

Both offline and online POMDP solution methods have been applied to aircraft collision avoidance. Offline methods perform the bulk of the computation before execution, saving the solution for the full belief state space in memory, whereas online methods perform all computation during execution for specific belief states. Recently, a generic offline solver was used to develop collision avoidance systems for unmanned aircraft with different sensor modalities [8]. Because the computational complexity scales exponentially with the dimensionality of the state space, the problem was made more tractable by only modeling the vertical motion of the aircraft. Online approaches can scale better to higher-dimensional state spaces, but they require much more computation during execution. Other work pursued an online approach for aircraft collision avoidance that relies on Monte Carlo sampling to search the belief space [9]. Because the computational requirements are exponential in the depth of the search, sparse sampling of the belief space at only a few decision points of an encounter was required.

Other approaches to collision avoidance involve computing collision probabilities and maneuvering when they exceed some threshold [3]. Several conflict probability estimation methods have been suggested that account for state uncertainty using analytic [10,11], numerical [12,13], and Monte Carlo techniques [14–16]. Carpenter and Kuchar used the probability of conflict for conflict detection in closely spaced parallel approaches [17], and Yang and Kuchar used a multistage threshold approach for conflict detection and resolution for free flight [18]. Karlsson et al. used a statistical hypothesis test to detect conflicts in automotive collision avoidance [19]. Building on this, Jansson and Gustafsson developed a framework for collision avoidance based on statistical decision making and stochastic numerical integration that accounts for state uncertainty in both conflict detection and resolution [20]. While these approaches may perform well, they are not optimal in general [21].

Although previous research has made significant progress in developing collision avoidance algorithms that are robust to state uncertainty, the importance of accounting for state uncertainty in collision avoidance has not been well studied. This paper introduces a new framework for accounting for state uncertainty in collision avoidance and uses it to explore the effect of properly handling state uncertainty on system performance. Similar to [8,9], the approach involves formulating the collision avoidance problem as a POMDP. Instead of solving the POMDP entirely offline or online, a hybrid solution is proposed that is more scalable yet still robust to state uncertainty. The expected utility of being in a particular belief state and taking a particular action is approximated as a function of the belief state, computed online, and the state-action utilities, computed offline. The collision avoidance system issues alerts that maximize expected utility.

This expected utility approach has a number of advantages over existing methods. First, unlike [8], the approach is more scalable, allowing more realistic systems to be modeled. Second, the approach is independent of the particular sensor system deployed on the aircraft. Unlike [8], the offline solution remains the same, regardless of the sensor system, although the online filtering process is sensor specific. Third, the approach is flexible, permitting different methods for approximating the state-action utilities. Fourth, this method directly accounts for state uncertainty using probabilistic models without having to rely upon specialized heuristics.

Two prototype collision avoidance systems developed using this framework are analyzed: one that accounts for the current state uncertainty in making decisions and another that neglects state uncertainty by only using a point estimate of the current state. Qualitative and quantitative comparisons of these two systems reveal the importance of adequately accounting for state uncertainty in collision avoidance. In addition to comparing these systems to each other, this paper also compares them to the Traffic Alert and Collision Avoidance System (TCAS), a system currently mandated on all large transport aircraft [1,22].

Received 18 November 2010; revision received 7 January 2011; accepted for publication 5 February 2011.

*Assistant Staff, 244 Wood Street; chryssanthacopoulos@ll.mit.edu. Member AIAA.

†Technical Staff, 244 Wood Street; mykelk@ll.mit.edu. Senior Member AIAA.

II. Partially Observable Markov Decision Processes

A Markov decision process (MDP) models problems in which an agent must act in a stochastic environment [23]. An MDP is defined by the tuple $(\mathcal{S}, \mathcal{A}, T, R)$. The set $\mathcal{S}$ contains all environment states. The set $\mathcal{A}$ contains all available actions. The transition function $T(s, a, s')$ represents the probability of transitioning to state s' after taking action a in state s. The reward function $R(s, a)$ is the reward for taking action a in state s. A policy π is a mapping from states to actions; the action to be executed from state s is denoted $\pi(s)$. An optimal policy is one that, if followed, maximizes the expected sum of rewards. If $\mathcal{S}$ and $\mathcal{A}$ are discrete, dynamic programming can be used to efficiently compute an optimal policy [24–26].

A POMDP extends the MDP model to problems where the state is not perfectly known [6]. A POMDP is defined by the tuple $(\mathcal{S}, \mathcal{A}, T, R, \mathcal{Z}, O)$, where $\mathcal{Z}$ is the set of all possible observations the agent can make and $O(s, z)$ gives the probability of observing z when in state s. The states transition according to T, as in an MDP, but the agent only has knowledge of the observations. Based on the sequence of past actions and observations, the agent must decide which action to execute to maximize its expected sum of rewards.

In a POMDP, all relevant information regarding the past actions and observations of the agent can be summarized by a probability distribution over the state space $\mathcal{S}$, called a belief state. The initial belief state is denoted b_0. The belief state at time t, denoted b_t, can be computed using the previous belief state, the previous action, and the current observation. According to Bayes' rule,

$$b_t(s') = \kappa O(s', z_t) \sum_{s \in \mathcal{S}} T(s, a_{t-1}, s') b_{t-1}(s) \tag{1}$$

for all states s', where κ is a normalization constant such that the distribution sums to one. This new belief state corresponds to the posterior state distribution. In systems that involve state variables that are naturally continuous, this computation may be done by a Kalman filter or approximated by one of its nonlinear variants [27–29].

A policy in a POMDP maps belief states to actions. The action to take from belief state b according to policy π is denoted $\pi(b)$. The expected sum of rewards, starting in belief state b, following action a for one time step, and following the actions prescribed by policy π thereafter, is called the expected belief-action utility, denoted $U^\pi(b, a)$. It is defined recursively as

$$U^\pi(b, a) = \sum_{s \in \mathcal{S}} b(s) R(s, a) + \sum_{z \in \mathcal{Z}} \Pr(z|b, a) U^\pi[f(b, a, z)] \tag{2}$$

where $U^\pi(b) = U^\pi[b, \pi(b)]$, and $f(b, a, z)$ is the updated belief state after executing a and observing z. The solution to a POMDP is an optimal policy π^*. The expected belief-action utility for an optimal policy π^* is $U^*(b, a)$. An optimal policy always selects the action that maximizes the expected utility:

$$\pi^*(b) = \arg\max_{a \in \mathcal{A}} U^*(b, a) \tag{3}$$

A POMDP can be solved optimally using dynamic programming [6], but exact solutions are computationally intractable in general [30]. A number of different approximations have been proposed in the literature that use point-based representations of the belief state to compute policies offline [31–34]. Additionally, limited planning over the belief space can be done online by expanding the tree of belief states reachable from the current belief state up to some horizon [35].

Several algorithms approximate the expected belief-action utility using state-action utilities:

$$U^*(b, a) \approx \sum_{s \in \mathcal{S}} U(s, a) b(s) \tag{4}$$

That is, the expected belief-action utility is approximately equal to the expectation over the belief state of the state-action utilities. The state-action utilities indicate how desirable it is to take a certain action from a certain state. They can be computed using a variety of methods. For example, they can be computed offline using a dynamic programming algorithm called value iteration on the underlying MDP (the POMDP without state uncertainty). Another way of computing the state-action utilities is to use Monte Carlo sampling, offline or online, to propagate a collection of state trajectories forward using different avoidance maneuvers.

Using the state-action utilities of the underlying MDP together with the complete belief state is called the QMDP method [36]. Although the offline planning to compute the state-action utility function neglects the partial observability of the problem, the state uncertainty is accounted for online using the current belief state. By only considering the complete belief state at the current time when making decisions, the QMDP method chooses actions under the assumption that the state will become fully observable at the next time step. The QMDP method has been shown to perform well on many problems but tends to fail in problems where taking particular actions results in a significant reduction in state uncertainty [35–38].

Instead of using the complete belief state in Eq. (4), the most likely state (MLS) can be used. In general, the MLS method is less robust than the QMDP method, especially when state uncertainty is high. The MLS method, when applied to collision avoidance, can result in missing potential threats. Figure 1 shows the space of relative future trajectories for different samples from the posterior state distribution. From one of the samples, there is some nonzero probability that the two aircraft will collide in the future, and so the potential threat is taken into account when using QMDP. If only the MLS is used, the space of relative future trajectories is restricted to the dark gray region, resulting in the potential threat being missed. Although the threat may be detected later in the encounter, it may be too late to prevent collision. The experiments later in this paper empirically show how much better a collision avoidance system is expected to perform when accounting for the full posterior state distribution.

III. Collision Avoidance Model

This section discusses how to model the collision avoidance problem as a POMDP. This paper considers a simplified version of the collision avoidance problem in which one aircraft equipped with a collision avoidance system, called the own aircraft, encounters an aircraft not equipped with a collision avoidance system, called the intruder aircraft. Similar to TCAS, the collision avoidance system on the own aircraft can issue resolution advisories that indicate how to adjust the vertical rate to avoid conflict. Conflict in this paper occurs when an aircraft comes within 500 ft horizontally and 100 ft vertically of the own aircraft. This has been termed a near midair collision (NMAC) in prior TCAS studies [39]. It has been used as the primary metric for safety evaluations of TCAS. Additional details

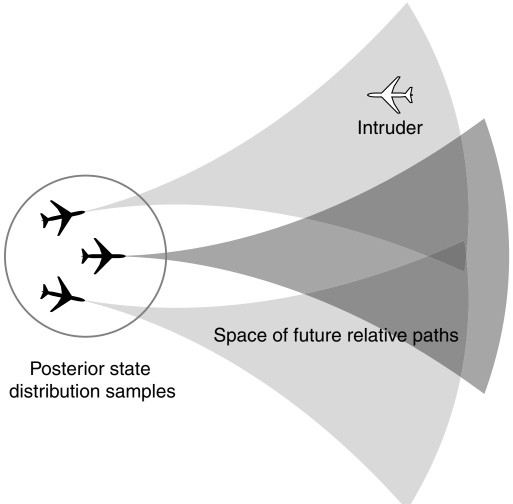

Intruder
Space of future relative paths
Posterior state distribution samples

Fig. 1 Notional diagram showing relative future trajectories from posterior state distribution.

can be found in [40]. The remainder of this section describes the various components of the POMDP.

A. Resolution Advisories

The collision avoidance system can issue one of two different initial advisories: climb at least 1500 ft/ min or descend at least 1500 ft/ min. The response to the advisory is modeled as a $1/4$ g acceleration to meet the target rate, if necessary. Following the initial advisory, the system can either terminate, reverse, or strengthen the advisory. The response to the reversal consists of applying a $1/3$ g acceleration to reach a 1500 ft/ min vertical rate in the direction opposite the original advisory. The response to a strengthening similarly consists of a $1/3$ g maneuver to achieve a vertical rate of 2500 ft/ min in the direction of the previous advisory. After an advisory has been strengthened, it can be weakened to reduce the required vertical rate to 1500 ft/ min. It is responded to with a $1/3$ g acceleration. The action space of the POMDP is composed of these advisories as well as the decision to issue no advisory.

B. Aircraft Dynamic Model

The aircraft trajectories are perturbed by zero-mean white Gaussian accelerations, horizontally and vertically [41]. The intruder aircraft experiences vertical accelerations with $3\ \text{ft/s}^2$ standard deviation. The own aircraft, moreover, undergoes the same vertical process, except when a resolution advisory is actively followed, in which case the own aircraft accelerates according to the response model of Sec. III.A. In the horizontal plane, both aircraft move with east and north accelerations with $3\ \text{ft/s}^2$ standard deviation.

The dynamic state of the aircraft is expressed using the following seven-variable state representation: altitude of the intruder aircraft relative to the own aircraft h, vertical rate of the own aircraft $\dot{h}_0$, vertical rate of the intruder aircraft $\dot{h}_1$, state of the resolution advisory s_{RA}, horizontal range to the intruder r, relative horizontal speed r_v, and difference in the direction of the relative horizontal velocity and the bearing of the intruder θ_v.

The advisory state variable s_{RA} is used to model the response to the advisories. There is a delay between the time the advisories are displayed and when they are executed. All initial advisories are responded to in exactly 5 s, and all subsequent advisories are responded to in exactly 3 s. The advisory state variable indicates the active advisory, if any, and the time remaining until response. This deterministic response model consists of 23 distinct states (one for no advisory, five for each initial advisory, and three for each reversal and strengthening).

To produce a discrete state-space model, the continuous state variables are discretized according to a standard grid-based scheme that uses cut points along each of the dimensions of the state space. The transition probabilities $T(s, a, s')$ of the discrete model are estimated using the continuous dynamic model together with sigma-pointing sampling and multilinear interpolation [21,40].

C. Sensor Model

The collision avoidance system senses the environment using an onboard beacon radar similar to the one currently employed by TCAS. The sensor measures the slant range and bearing of the intruder aircraft. 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 sensor also receives measurements of the intruder altitude. Although there is no jitter in the intruder altitude measurements, they are quantized at 25 ft intervals. The own altitude, vertical rate, and heading are provided by the onboard avionics. The noise in those quantities is negligible. This sensor model defines $O(s, z)$.

D. Reward Function

Unit cost (negative reward) is accrued when the intruder comes within 1000 ft horizontally and 100 ft vertically of the own aircraft. The horizontal conflict radius was chosen to be 1000 ft as opposed to 500 ft, as in the NMAC definition, to help compensate for sensor noise. Additionally, to reduce the number of false alerts and the course deviation introduced by the system, a small cost of 0.001 is incurred any time an alert is initially issued. Costs of 0.009 and of 0.01 are accumulated when the system strengthens and reverses, respectively, in order to address other operational concerns. This cost scheme defines $R(s, a)$.

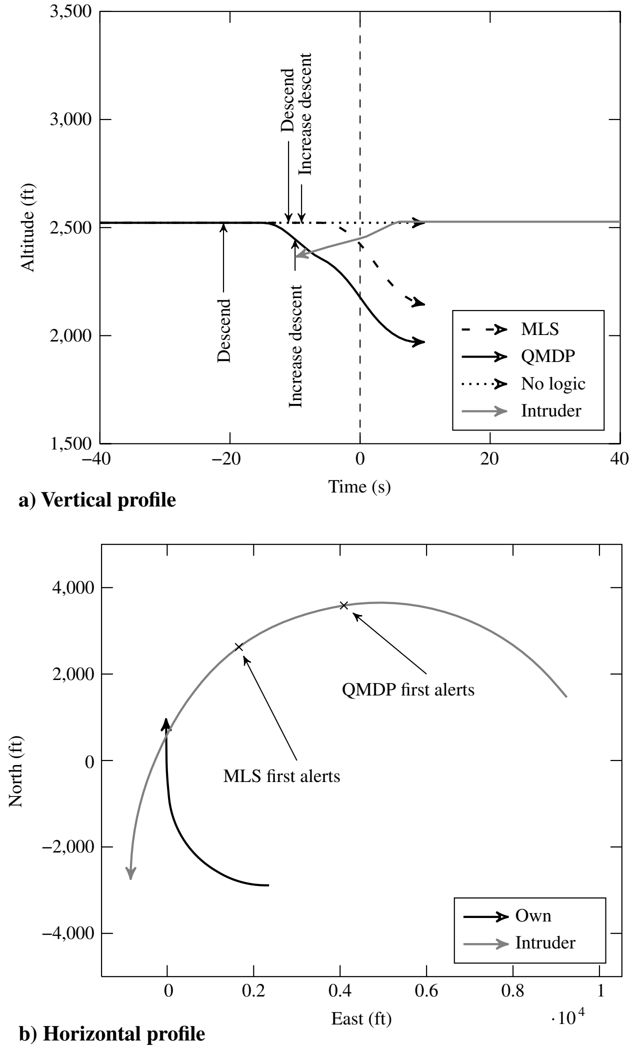
IV. Approximate Solution to the Collision Avoidance Problem

This section solves the collision avoidance problem using the approximation in Eq. (4). The belief state is recursively computed online as new observations from the sensor are received. The state-action utilities are computed offline by approximately solving the underlying MDP. The result is collision avoidance logic that, given the state of the system, returns the advisory to issue.

A. Belief State Construction

The belief state is constructed by separately estimating the horizontal and vertical state distributions and combining the results. The horizontal and vertical state estimators each produce a collection of samples with associated probabilities. The belief state is represented by the Cartesian product of the horizontal and vertical samples. Because each sample may not be in the discrete state space $\mathcal{S}$, multilinear interpolation of the samples is used to obtain a belief state over $\mathcal{S}$.

Horizontal state estimation involves using the measurements and the prior belief state to produce horizontal state samples and



a) Vertical profile

b) Horizontal profile

Fig. 2 First example encounter.

probabilities. The horizontal state estimation uses measurements of the slant range, bearing, and altitude of the intruder, as well as measurements of the own altitude and heading. The horizontal state estimator is an unscented Kalman filter [42] that returns a Gaussian posterior distribution over ΔE, $\Delta\dot{E}$, ΔN, and $\Delta\dot{N}$, the relative east and north positions and velocities. Sigma-point samples from this distribution are drawn with associated weights that are interpreted as probabilities. Each sample is converted into the three-dimensional representation as follows:

$$\begin{cases} r = \sqrt{(\Delta E)^2 + (\Delta N)^2}, \\ r_v = \sqrt{(\Delta\dot{E})^2 + (\Delta\dot{N})^2}, \\ \theta_v = \operatorname{atan2}(\Delta\dot{N}, \Delta\dot{E}) - \operatorname{atan2}(\Delta N, \Delta E) \end{cases} \tag{5}$$

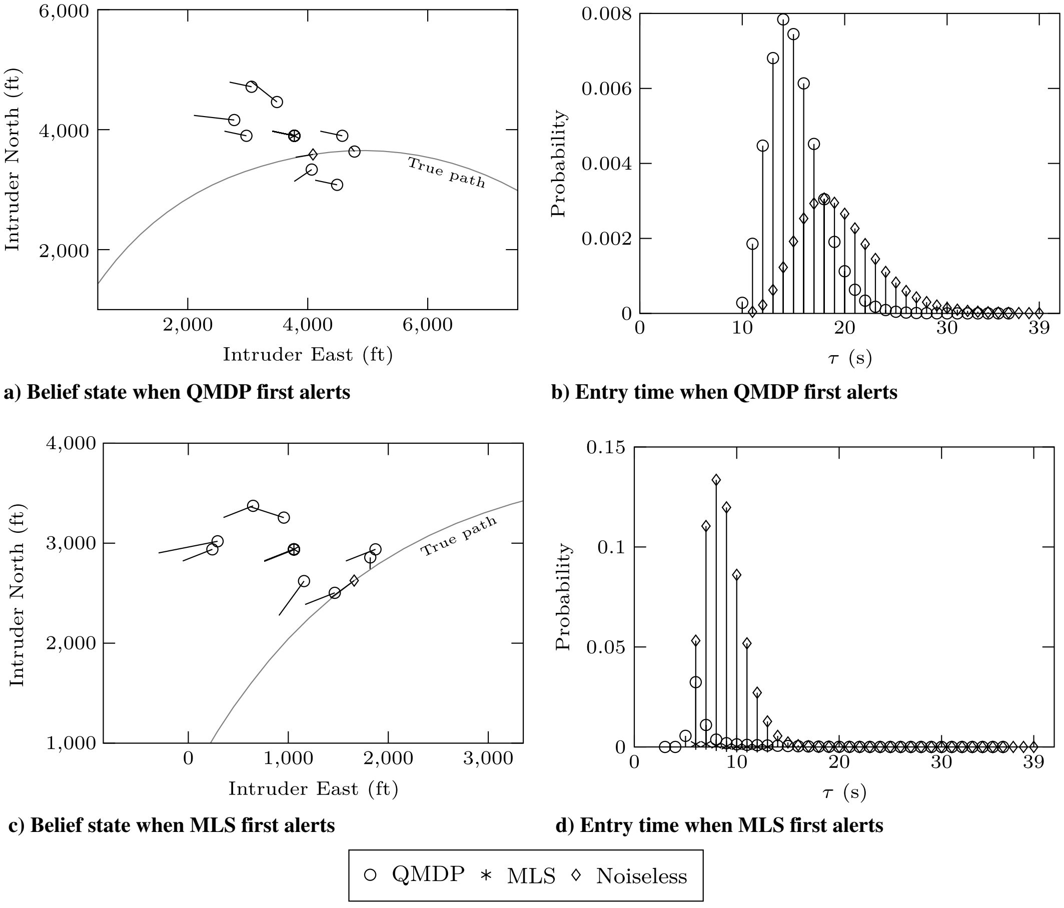
Vertical state estimation involves using the measurements and the prior belief state to produce vertical state samples and probabilities. A Kalman filter is used to estimate the Gaussian posterior of the intruder altitude and vertical rate using measurements of the intruder altitude. Because the noise in the own altitude and vertical rate is relatively insignificant, no estimation is required for those quantities. The relative altitude h is estimated by subtracting the own altitude measurement from the maximum a posteriori estimate of the intruder altitude. The intruder vertical rate $\dot{h}_1$ is estimated by the maximum a posteriori estimate of the intruder vertical rate. Because uncertainty in the intruder altitude and vertical rate is not expected to be significant, the maximum a posteriori estimates are used instead of the entire distribution to reduce computation time.

B. State-Action Utility Function

The state-action utility function is computed by solving the underlying MDP. This paper uses an approximation method for solving MDPs that was originally introduced by the authors in [40,43]. The approximation is applicable to a certain class of MDPs, where only a subset of the state variables is controlled by the agent. This is true of the present collision avoidance problem in which the system only affects the vertical motion of the aircraft.

The approach involves decomposing the full problem into two subproblems: one involving the uncontrolled state space and the other involving the controlled state space. Solving the uncontrolled subproblem involves estimating the probability distribution over the entry time: the time τ until the uncontrolled state enters a certain subspace of the uncontrolled state space. The distribution is estimated up to a certain horizon, and the remaining probability mass represents the probability that τ is outside the horizon. Dynamic programming is used offline to estimate this distribution for every discrete uncontrolled state. The controlled subproblem is treated as a finite-horizon MDP, where τ indicates the number of steps until termination. The solution of the controlled subproblem is a sequence of state-action utility functions for the controlled state space for every possible value of τ up to the horizon. They are computed using value iteration. The solutions to the uncontrolled and controlled subproblems are stored in tables in memory, and they are combined to form an approximation to the joint optimal state-action utility function of the MDP. Further details regarding the algorithm, such as the assumptions under which it is applicable, can be found in [40,43].

In the collision avoidance problem, the uncontrolled state space corresponds to the horizontal motion of the aircraft. It comprises all discrete states of the form (r, r_v, θ_v). With the discretization granularity used in this paper, there are approximately 730,000 discrete uncontrolled states. The variable τ is the time until a horizontal conflict (when the aircraft are horizontally separated by less than 1000 ft). The controlled state space corresponds to the vertical motion of the aircraft plus the advisory state variable that keeps track of the current resolution advisory. The controlled state space comprises all discrete states of the form $(h, \dot{h}_0, \dot{h}_1, s_{\mathrm{RA}})$. There are approximately 213,000 discrete controlled states.



a) Belief state when QMDP first alerts

b) Entry time when QMDP first alerts

c) Belief state when MLS first alerts

d) Entry time when MLS first alerts

Fig. 3 Belief state samples and entry time distribution slices for first example encounter.

The MDP was solved using this method with a horizon of 40 s. Solving the uncontrolled subproblem took 92 s on a single 3 GHz Intel Xeon core and required 228 MB of storage using a 64-bit floating point representation. Solving the controlled subproblem took less than 2 min and required 263 MB of storage.

V. Example Encounters

This section compares the performance of QMDP and MLS on two characteristic example encounters. Because they involve simple table lookups and interpolation, QMDP and MLS run orders of magnitude faster than real time in simulation. The computation time of QMDP, however, is K times greater than MLS, where K is the number of belief state samples. The encounters presented in this section highlight some of the deficiencies of the MLS method for collision avoidance and demonstrate how QMDP can lead to significantly improved performance that approaches that of the system without sensor error. The encounters were generated using a high-fidelity encounter model that is reflective of the statistical properties of how aircraft encounter each other in the current airspace. The parameters and structure of the encounter model were derived from nine months of national radar data [44,45].

In the first encounter, MLS delays the alert too long and is unable to prevent NMAC. QMDP, which accounts for the probability that the intruder may be heading toward the own aircraft, alerts earlier and safely separates the aircraft. In the second encounter, both MLS and QMDP initially alert together. However, MLS, for which the estimates can be very erratic, discontinues the alert prematurely, leading to NMAC. QMDP, correctly estimating the danger, keeps the advisory on display and successfully resolves the encounter. The following discussion analyzes the encounters in further detail.

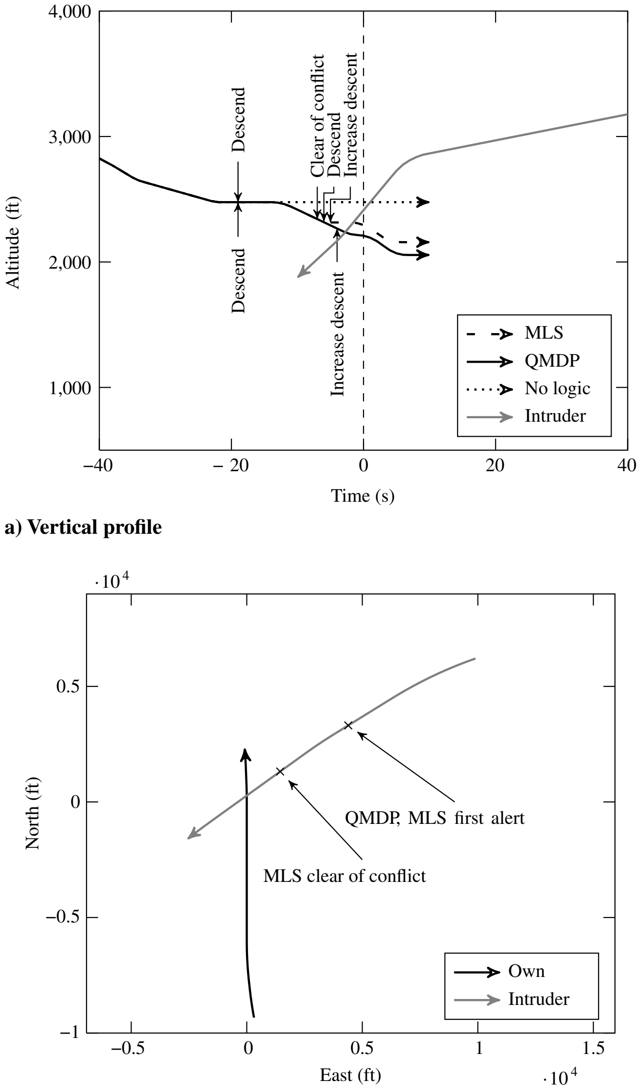
In the first encounter, shown in Fig. 2, the own aircraft and the intruder are initially nearly coaltitude and are slowly turning into each other. QMDP issues an initial descend alert 20 s into the encounter, just as the intruder begins to turn into the own aircraft. The expected utility of issuing a descend advisory is -0.01386, which is greater than that of issuing a climb advisory (-0.01394) or of not issuing an advisory (-0.01542). MLS estimates the expected utilities of issuing a descend, a climb, and no advisory at this time as -0.001, -0.001, and 0, respectively, which are simply the immediate rewards of each action because, according to MLS, collision risk is predicted to be negligible from this state. MLS postpones issuing the descend advisory until 29 s into the encounter. Both QMDP and MLS strengthen the descend in the future as the intruder begins descending. MLS strengthens only 2 s after initially alerting. QMDP successfully prevents an NMAC, while MLS does not. The performance of the logic with noiseless sensors and perfect estimation, not shown in the figure, performs comparably to QMDP, issuing a descend advisory 21 s into the encounter and strengthening it 7 s later.

Figure 3 shows the belief state samples of the intruder horizontal position and velocity when QMDP first alerts (Fig. 3a) and when MLS first alerts (Fig. 3c). The samples are represented as markers indicating the position and lines indicating the magnitude and direction of the velocity. The samples used by QMDP are indicated by circles. The MLS sample, which coincides with one of the samples used by QMDP, is indicated by an asterisk. The true intruder trajectory is shown in gray, and the current true position is marked by a diamond. Figures 3b and 3d show the entry time distributions for QMDP and MLS when QMDP first alerts and when MLS first alerts, respectively. Also shown are the entry time distributions during those alert times computed using noiseless sensors and perfect estimation. Only the probability mass of the entry time within the time horizon of 40 s is shown.

When QMDP first alerts, several samples are closer to the true position and velocity than the MLS estimate. Indeed, the entry time distribution used by QMDP better approximates the noiseless entry time distribution than MLS. In fact, MLS assigns all probability to the entry time being outside the horizon (and thus is absent in Fig. 3b), indicating the own aircraft is not expected to come into conflict in the near future; QMDP, however, does correctly predict that the aircraft can come into conflict within 10 to 20 s. Only as the intruder continues turning and begins to approach the own aircraft more directly does MLS predict horizontal conflict and issue an alert. The alert is issued too late, however, to prevent NMAC.

The performance of TCAS version 7.1, not shown, was also evaluated on this encounter. The alerting behavior of TCAS is similar to that of MLS. It initially issues a descend advisory 27 s into the encounter. TCAS strengthens 3 s later but fails to prevent an NMAC. It should be noted that, although TCAS relies mostly on point estimates of the state, it does maintain some measure of confidence in the state estimates to determine whether to alert. For example, the horizontal miss distance filter, which suppresses advisories for intruders for which the estimated horizontal miss distance exceeds some threshold, uses several trackers that are initialized only after consecutive valid range and bearing measurements are received. The range tracker maintains an estimate of the range measurement error, which is used to compute the range acceleration. If the range acceleration exceeds some threshold, resolution advisories are suppressed [46].

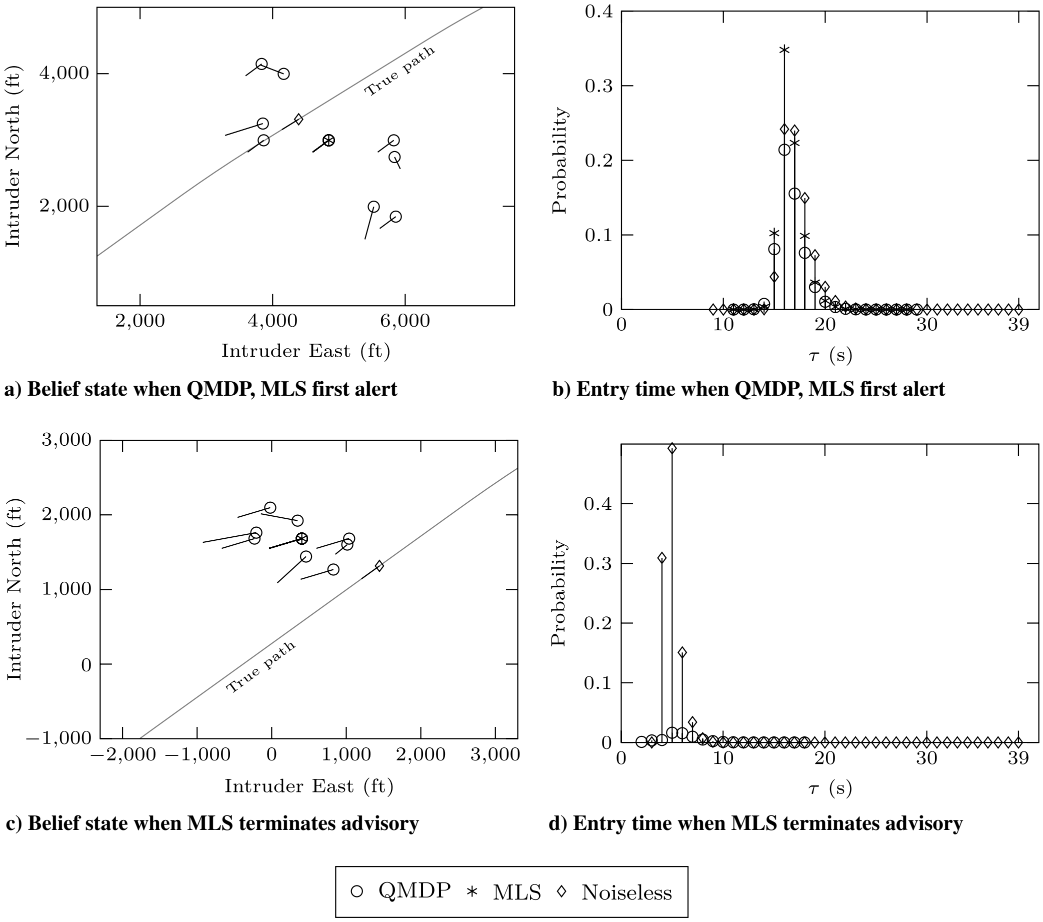
In the second example encounter, shown in Fig. 4, both QMDP and MLS initially issue descend advisories 21 s into the encounter. Figures 5a and 5b show the belief state samples and entry time distributions at this time, respectively. The entry time distributions of both methods closely approximate the noiseless distribution. Twelve seconds later, MLS terminates the advisory. Figures 5c and 5d show the belief state samples and entry time distributions. MLS assigns probability one to τ outside the time horizon. QMDP,



a) Vertical profile

b) Horizontal profile

Fig. 4 Second example encounter.



a) Belief state when QMDP, MLS first alert

b) Entry time when QMDP, MLS first alert

c) Belief state when MLS terminates advisory

d) Entry time when MLS terminates advisory

Fig. 5 Belief state samples and entry time distribution slices for second example encounter.

accounting for the full state distribution, maintains the advisory, estimating the expected utility of continuing the descent as -4.966×10^{-5}, which is greater than those of reversing (−0.01018), of strengthening (−0.009), or of terminating the advisory (-1.5067×10^{-4}). The entry time distribution inferred using the full posterior state distribution, like the noiseless entry time distribution, has probability mass between 0 and 10 s. MLS activates the advisory again at the next time step and must quickly strengthen it, but an NMAC results several seconds later. QMDP strengthens 15 s after it first alerts, and the aircraft safely pass each other. As before, the performance of the logic without sensor noise is similar to that of QMDP: the system issues a descend at $t = 18$ s and strengthens at $t = 36$ s.

TCAS issues the descend advisory after 19 s. It weakens the advisory to "do not climb" 11 s later, but after the own aircraft levels off, it must revert to issuing the descend advisory again. Similar to MLS, the aircraft come into conflict shortly thereafter.

VI. Simulation Results

While examining example encounters may be useful in qualitatively comparing the different methods, large-scale Monte Carlo simulations are necessary to quantify the overall system performance. This section presents the results of Monte Carlo evaluations. All simulations used encounters generated from the same encounter model used to generate the example encounters in the previous section.

A. Performance Evaluation Statistics

Table 1 summarizes the results of evaluating QMDP, MLS, and TCAS. In addition to summarizing the probability (denoted Pr) of certain events of interest, such as an NMAC, an alert, a strengthening, and a reversal, the table also reports the number of times the resolution advisory is changed (RA changes) and how long the advisories remain active (RA duration). Also reported are the standard deviations associated with the estimates. The metrics for QMDP and MLS were estimated using 500,000 simulated encounters. The encounter set included encounters that required alerts and encounters that did not require alerts. The probability of NMAC without any mitigation is 2.9236×10^{-3}. One million encounters were used for estimating the metrics for TCAS due to their higher variance. As the table indicates, although both QMDP and MLS are safer than TCAS, the QMDP method is twice as safe as

Table 1 Performance evaluation statistics

	QMDP		MLS		TCAS	
	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.
Pr (NMAC)	$4.23 \cdot 10^{-5}$	$3.66 \cdot 10^{-6}$	$8.82 \cdot 10^{-5}$	$3.96 \cdot 10^{-6}$	$1.43 \cdot 10^{-4}$	$1.81 \cdot 10^{-5}$
Pr (alert)	$4.33 \cdot 10^{-1}$	$1.10 \cdot 10^{-2}$	$4.49 \cdot 10^{-1}$	$1.29 \cdot 10^{-2}$	$5.03 \cdot 10^{-1}$	$1.02 \cdot 10^{-2}$
Pr (strengthening)	$2.29 \cdot 10^{-2}$	$1.38 \cdot 10^{-3}$	$5.54 \cdot 10^{-2}$	$2.76 \cdot 10^{-3}$	$1.18 \cdot 10^{-2}$	$5.98 \cdot 10^{-4}$
Pr (reversal)	$6.75 \cdot 10^{-4}$	$1.65 \cdot 10^{-4}$	$5.53 \cdot 10^{-4}$	$1.16 \cdot 10^{-4}$	$3.26 \cdot 10^{-3}$	$5.44 \cdot 10^{-4}$
RA changes	$4.58 \cdot 10^{-1}$	$1.16 \cdot 10^{-2}$	$5.85 \cdot 10^{-1}$	$1.69 \cdot 10^{-2}$	$6.56 \cdot 10^{-1}$	$1.28 \cdot 10^{-2}$
RA duration	$9.21 \cdot 10^{0}$	$2.39 \cdot 10^{-1}$	$6.36 \cdot 10^{0}$	$1.87 \cdot 10^{-1}$	$1.10 \cdot 10^{1}$	$2.45 \cdot 10^{-1}$

Table 2 Outcome categories

Outcome category	Alert necessary?	Alert issued?	Conflict occurred?
Correct rejection			
Correct detection	×	×	
False alarm		×	
Missed detection	×		×
Induced conflict		×	×
Late alert	×	×	×

Table 4 Probabilities of four encounter categories

		QMDP	
		Safe	NMAC
MLS	Safe	0.99	$1.46 \cdot 10^{-5}$
	NMAC	$6.05 \cdot 10^{-5}$	$2.77 \cdot 10^{-5}$

MLS while alerting, strengthening, and reversing just as often. QMDP, on average, changes the advisory less and, because it does not terminate the advisory as frequently as MLS, keeps the advisories active for longer. QMDP is 3.4 times safer than TCAS, with 13% fewer alerts.

System performance can also be measured in terms of other alerting system outcomes, shown in Table 2 [47]. The definition of each outcome category depends on three criteria: 1) whether an alert was necessary, 2) whether an alert was issued, and 3) whether a conflict occurred. Determining whether an alert is necessary in a particular situation is not exactly straightforward, because the aircraft dynamics are nondeterministic. The standard approach is to run simulations, both with and without the alerting system, with the same random seed. Because the seeds are the same, the trajectories with and without the system will be exactly the same until after an alert is issued. Once the pilot responds to the alert, the trajectory with the alerting system will diverge from the trajectory without the alerting system, called the nominal trajectory. An alert is defined to be necessary if the nominal trajectory results in conflict.

Table 3 reports the probability of each outcome category for QMDP, MLS, and TCAS. QMDP has more or comparable correct rejections and correct detections as MLS and TCAS and fewer or comparable false alarms, missed detections, induced NMACs, and late alerts.

In addition to comparing the effectiveness of QMDP and MLS in preventing conflict, it is also interesting to compare how often one method resolves a conflict the other fails to resolve. Table 4 lists the probabilities of the following encounter categories: QMDP resolves a conflict that MLS does not, MLS resolves a conflict that QMDP does not, both resolve the conflict, and neither resolves the conflict. Although both methods are able to resolve most encounters, the probability that QMDP succeeds where MLS fails is four times greater than the probability that QMDP fails where MLS succeeds.

B. Sensor Noise Robustness

Because the sensor error may deviate significantly from the sensor error model assumed in Sec. III.C, it is important to study the robustness of the logic to different degrees of state uncertainty. Figures 6a and 6b illustrate the effect of varying the bearing noise on the probability of NMAC and of alert for QMDP, MLS, and TCAS. The range noise was kept fixed at its nominal value of 50 ft. Figures 6c and 6d similarly illustrate the effect of varying the range noise on the probability of NMAC and of alert for the three systems. The bearing noise was kept fixed at its nominal value of 10°. Each point on the QMDP and MLS curves was estimated using 500,000 simulated encounters. Each point on the TCAS curves was estimated using 1 million encounters. The error bars indicate the standard deviation of the estimates.

When the bearing noise is small, QMDP and MLS alert at the same rate and are equally successful in preventing NMAC. When the bearing noise is increased, the ability of MLS to prevent NMACs significantly degrades. The performance of QMDP is relatively insensitive to the change in bearing noise, indicating a large degree of robustness to sensor error. Increasing the bearing noise also increases the alert probability, although the rate of increase diminishes as bearing noise increases. Although MLS alerts 64% more when the bearing noise is 20° than when it is 0°, the probability of NMAC is increased by a factor of nearly eight. QMDP increases its alert rate by a comparable amount (59%), but the probability of NMAC is only increased by a factor of 1.6, representing a significant improvement over MLS. TCAS is relatively insensitive to the increase in bearing noise, because bearing measurements are only used to disable alerting in situations where the projected miss distance is large and not directly for threat resolution purposes.

Similar trends are seen when the range noise is increased, although the difference between QMDP and MLS is less significant. As the range noise is increased from 0 to 1000 ft, the probability of NMAC is increased by 50% when using MLS, while the probability of NMAC is increased by only 25% when using QMDP. The performance of the two methods does not meet when the range noise is zero because of the large amount of bearing noise. The response of TCAS to changes in range noise is somewhat erratic. The probability of NMAC for TCAS is lowered at the expense of alerting 48% more frequently. QMDP and MLS, however, do not alert considerably more.

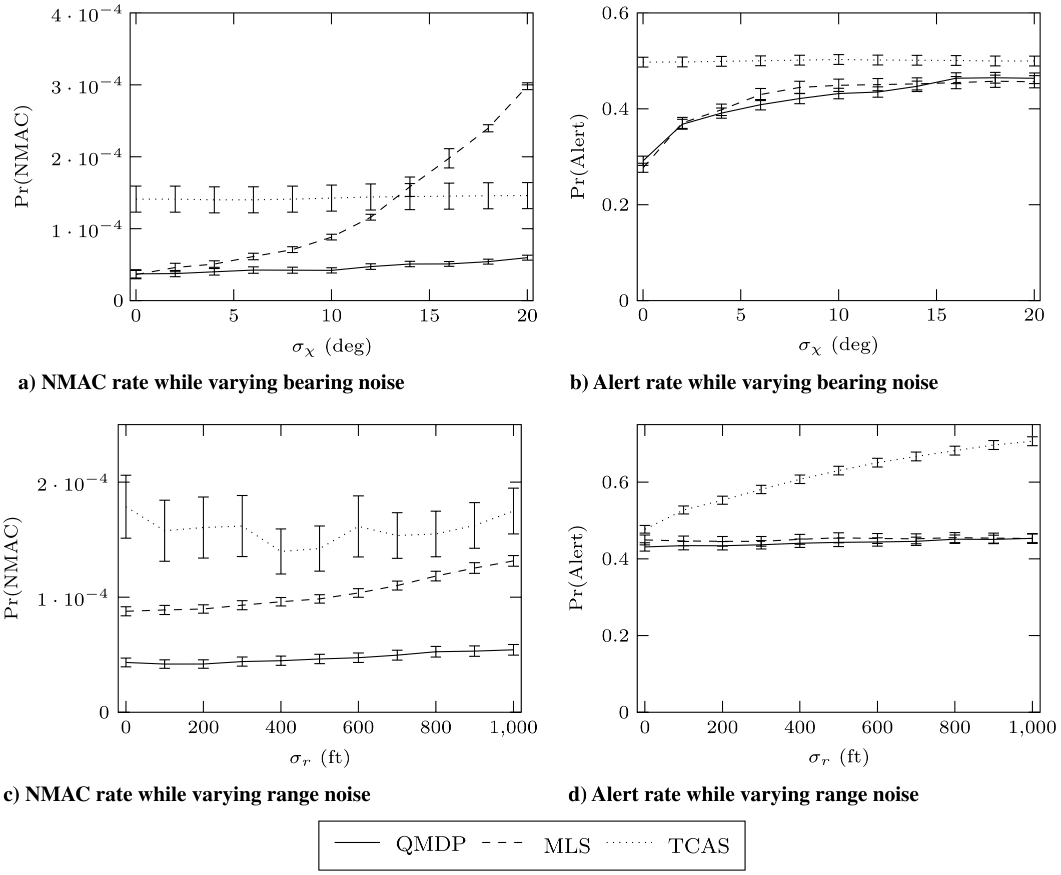
C. Safety Curves

The results of Secs. VI.A and VI.B considered the performance of the logic optimized using the fixed reward function of Sec. III.D. A safety curve is one way to visualize how varying a system parameter affects the safety and alert rates [48]. It is a plot (see Fig. 7) of the probability of safety versus the probability of alert as a system parameter is varied. Each point on the curve is called an operating point. Operating points near the top left represent good system performance; the system prevents conflicts reliably with few alerts. Figure 7 shows the safety curves for QMDP and MLS swept out as the cost of alerting is varied from zero to one while keeping the cost of other events, such as the cost of conflict, fixed. Each operating point was estimated using 100,000 simulated encounters. The lower left regions of the plots correspond to costs of alerting near one, and the upper right regions correspond to costs near zero. The performance of TCAS is shown as a single operating point.

The QMDP curve dominates the MLS curve. That is, QMDP achieves the same safety rate as MLS with a lower alert rate or, alternatively, achieves a higher safety rate than MLS with the same alert rate. Both methods dominate TCAS.

Table 3 Probability of each alerting outcome

	QMDP		MLS		TCAS	
	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.
Correct rejection	$5.60 \cdot 10^{-1}$	$1.73 \cdot 10^{-2}$	$5.44 \cdot 10^{-1}$	$1.59 \cdot 10^{-2}$	$4.83 \cdot 10^{-1}$	$1.43 \cdot 10^{-2}$
Correct detection	$2.91 \cdot 10^{-3}$	$5.44 \cdot 10^{-5}$	$2.86 \cdot 10^{-3}$	$5.44 \cdot 10^{-5}$	$2.87 \cdot 10^{-3}$	$5.44 \cdot 10^{-5}$
False alarm	$4.30 \cdot 10^{-1}$	$1.10 \cdot 10^{-2}$	$4.46 \cdot 10^{-1}$	$1.29 \cdot 10^{-2}$	$5.07 \cdot 10^{-1}$	$1.46 \cdot 10^{-2}$
Missed detection	$0.00 \cdot 10^{0}$	$0.00 \cdot 10^{0}$	$2.49 \cdot 10^{-7}$	$1.65 \cdot 10^{-7}$	$0.00 \cdot 10^{0}$	$0.00 \cdot 10^{0}$
Induced conflict	$2.79 \cdot 10^{-5}$	$3.52 \cdot 10^{-6}$	$2.77 \cdot 10^{-5}$	$3.62 \cdot 10^{-6}$	$8.67 \cdot 10^{-5}$	$3.60 \cdot 10^{-5}$
Late alert	$1.44 \cdot 10^{-5}$	$1.01 \cdot 10^{-6}$	$6.03 \cdot 10^{-5}$	$1.61 \cdot 10^{-6}$	$5.73 \cdot 10^{-5}$	$1.20 \cdot 10^{-6}$



a) NMAC rate while varying bearing noise

b) Alert rate while varying bearing noise

c) NMAC rate while varying range noise

d) Alert rate while varying range noise

Fig. 6 Sensor noise robustness. Error bars indicate standard deviation of estimates.

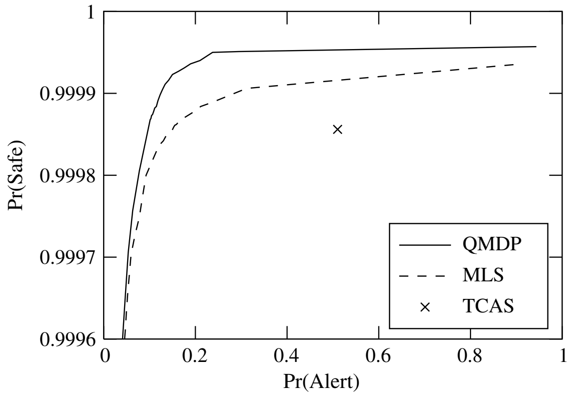


Fig. 7 Safety curves.

VII. Conclusions

This paper introduces a framework for accounting for state uncertainty in the development of collision avoidance systems. The problem was cast as a POMDP and solved approximately. The expected utility of being in a particular belief state and taking a particular action was approximated as a function of the belief state, computed online, and the state-action utilities, computed offline. The approach was applied to the problem of collision avoidance using a sensor similar to that employed by TCAS. The belief state was computed using several Kalman filters, and the state-action utilities were computed by solving the underlying MDP using dynamic programming. This approach was used to study the importance of accounting for state uncertainty in collision avoidance.

Although many collision avoidance systems in the literature simply use point estimates of the state, the experiments in this paper show that performance can be significantly enhanced by using information from the full posterior state distribution. A Monte Carlo simulation of 1000s of realistic encounter scenarios showed that adequately accounting for state uncertainty can cut the probability of conflict in half without increasing the alert rate. Using the full posterior state distribution also led to greater robustness to increased sensor noise. The experiments showed that using point estimates leads to severely degraded performance as sensor noise is increased, but performance when using the full posterior distribution only degrades modestly. Regardless of whether point estimates or state distributions are used, the logic that results from solving the MDP significantly enhances safety beyond that provided by TCAS with a lower alert rate.

Although the experiments focus on a system for an aircraft equipped with a TCAS sensor, the general approach can be extended to systems that use a variety of different surveillance systems: both ground-based and airborne. The TCAS sensor can only detect aircraft that are equipped with transponders. Other sensors that rely upon passive radar surveillance or electro-optical/infrared technology can detect aircraft without transponders, but they are generally noisier and result in much more state uncertainty than the TCAS sensor. Adequately accounting for state uncertainty when using such sensors becomes even more important, especially for unmanned aircraft systems where there is no pilot in the cockpit to validate the recommended maneuver.

Acknowledgments

This work is sponsored by the U.S. Air Force under U.S. Air Force Contract FA8721-05-C-0002. Opinions, interpretations, conclusions, and recommendations are those of the authors and are not necessarily endorsed by the U.S. Government. The authors appreciate the support provided by the Traffic Alert and Collision Avoidance System Program Manager, Neal Suchy. This work has benefited from conversations with Matthew W. M. Edwards, J. Daniel Griffith, and David A. Spencer of Lincoln Laboratory, Massachusetts Institute of Technology. The authors gratefully acknowledge the helpful comments received from anonymous reviewers.

References

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

[2] Chamlou, R., "Design Principles and Algorithm Development for Two Types of NextGen Airborne Conflict Detection and Collision Avoidance," *Integrated Communications Navigation and Surveillance Conference*, Herndon, VA, IEEE Publ., Piscataway, NJ, May 2010, pp. N7-1–N7-12.
doi:10.1109/ICNSURV.2010.5503327

[3] Kuchar, J. K., and Yang, L. C., "A Review of Conflict Detection and Resolution Modeling Methods," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 1, No. 4, Dec. 2000, pp. 179–189.
doi:10.1109/6979.898217

[4] Russell, S., and Norvig, P., *Artificial Intelligence: A Modern Approach*, 3rd ed., Prentice–Hall, Upper Saddle River, NJ, 2010.

[5] Sondik, E. J., "The Optimal Control of Partially Observable Markov Processes," Ph.D. Dissertation, Stanford Univ., Stanford, CA, 1971.

[6] Kaelbling, L. P., Littman, M. L., and Cassandra, A. R., "Planning and Acting in Partially Observable Stochastic Domains," *Artificial Intelligence*, Vol. 101, Nos. 1–2, May 1998, pp. 99–134.
doi:10.1016/S0004-3702(98)00023-X

[7] Thrun, S., Burgard, W., and Fox, D., *Probabilistic Robotics*, MIT Press, Cambridge, MA, 2006.

[8] Temizer, S., Kochenderfer, M. J., Kaelbling, L. P., Lozano-Pérez, T., and Kuchar, J. K., "Collision Avoidance for Unmanned Aircraft Using Markov Decision Processes," AIAA Guidance, Navigation, and Control Conference, Toronto, AIAA Paper 2010-8040, 2010.

[9] Wolf, T. B., and Kochenderfer, M. J., "Aircraft Collision Avoidance Using Monte Carlo Real-Time Belief Space Search," *Journal of Intelligent and Robotic Systems* [online], 2011.
doi:10.1007/s10846-010-9532-6

[10] Paielli, R. A., and Erzberger, H., "Conflict probability estimation for free flight," *Journal of Guidance, Control, and Dynamics*, Vol. 20, No. 3, May–June 1997, pp. 588–596.
doi:10.2514/2.4081

[11] Chan, F. K., *Spacecraft Collision Probability*, Aerospace Press, El Segundo, CA, 2008.

[12] Jones, T., "Tractable Conflict Risk Accumulation in Quadratic Space for Autonomous Vehicles," *Journal of Guidance, Control, and Dynamics*, Vol. 29, No. 1, Jan.–Feb. 2006, pp. 39–48.
doi:10.2514/1.10515

[13] van Daalen, C. E., and Jones, T., "Fast Conflict Detection Using Probability Flow," *Automatica*, Vol. 45, No. 8, Aug. 2009, pp. 1903–1909.
doi:10.1016/j.automatica.2009.04.010

[14] Yang, L., Yang, J. H., Kuchar, J., and Feron, E., "A Real-Time Monte Carlo Implementation for Computing Probability of Conflict," AIAA Guidance, Navigation, and Control Conference and Exhibit, Providence, RI, AIAA Paper 2004-4876, Aug. 2004.

[15] Yang, L. C., "Aircraft Conflict Analysis and Real-Time Conflict Probing Using Probabilistic Trajectory Modeling," Ph.D. Dissertation, Massachusetts Inst. of Technology, Lexington, MA, 2000.

[16] Blom, H. A. P., Krystul, J., Bakker, G. J., Klompstra, M. B., and Obbink, B. K., "Free Flight Collision Risk Estimation by Sequential MC Simulation," *Stochastic Hybrid Systems*, edited by C. G. Cassandras, and J. Lygeros, CRC/Taylor and Francis, Boca Raton, FL, 2007, pp. 249–281.

[17] Carpenter, B. D., and Kuchar, J. K., "Probability-Based Collision Alerting Logic for Closely-Spaced Parallel Approach," AIAA 35th Aerospace Sciences Meeting, Reno, NV, AIAA Paper 1997-0222, Jan. 1997.

[18] Yang, L. C., and Kuchar, J. K., "Prototype conflict alerting system for free flight," *Journal of Guidance, Control, and Dynamics*, Vol. 20, No. 4, July–Aug. 1997, pp. 768–773.
doi:10.2514/2.4111

[19] Karlsson, R., Jansson, J., and Gustafsson, F., "Model-Based Statistical Tracking and Decision Making for Collision Avoidance Application," *American Control Conference*, Vol. 4, IEEE Publ., Piscataway, NJ, 2004, pp. 3435–3440.

[20] Jansson, J., and Gustafsson, F., "A Framework and Automotive Application of Collision Avoidance Decision Making," *Automatica*, Vol. 44, 2008, No. 9, pp. 2347–2351.
doi:10.1016/j.automatica.2008.01.016

[21] Kochenderfer, M. J., Chryssanthacopoulos, J. P., Kaelbling, L. P., and Lozano-Perez, T., "Model-Based Optimization of Airborne Collision Avoidance Logic," Lincoln Lab., Massachusetts Inst. of Technology, Project Rept. ATC-360, Lexington, MA, 2010.

[22] Kuchar, J. K., and Drumm, A. C., "The Traffic Alert and Collision Avoidance System," *Lincoln Laboratory Journal*, Vol. 16, No. 2, 2007, pp. 277–296.

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

[24] Bellman, R., *Dynamic Programming*, Princeton: Princeton Univ. Press, Princeton, NJ, 1957.

[25] Bertsekas, D. P., *Dynamic Programming: Deterministic and Stochastic Models*, 1st ed., Prentice–Hall, Englewood Cliffs, NJ, 1987.

[26] Bertsekas, D. P., *Dynamic Programming and Optimal Control*, Vol. 1, 3rd ed., Athena Scientific, Belmont, MA, 2005.

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

[28] Mohinder, S. G., and Andrews, A. P., *Kalman Filtering: Theory and Practice Using MATLAB*, Wiley, New York, 2001.

[29] Minkler, G., and Minkler, J., *Theory and Applications of Kalman Filtering*, Magellan Book Co., Tuscon, AZ, 1993.

[30] Papadimitriou, C. H., and Tsitsiklis, J. N., "The Complexity of Markov Decision Processes," *Mathematics of Operations Research*, Vol. 12, No. 3, Aug. 1987, pp. 441–450.
doi:10.1287/moor.12.3.441

[31] Smith, T., and Simmons, R. G., "Point-Based POMDP Algorithms: Improved Analysis and Implementation," *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence*, AUAI Press, Arlington, VA, 2005, pp. 542–549.

[32] Kurniawati, H., Hsu, D., and Lee, W., "SARSOP: Efficient Point-Based POMDP Planning by Approximating Optimally Reachable Belief Spaces," *Robotics: Science and Systems IV*, MIT Press, Cambridge, MA, 2009, pp. 65–72.

[33] Roy, N., Gordon, G., and Thrun, S., "Finding Approximate POMDP Solutions Through Belief Compression," *Journal of Artificial Intelligence Research*, Vol. 23, 2005, pp. 1–40.
doi:10.1613/jair.1496

[34] Spaan, M. T. J., and Vlassis, N., "A Point-Based POMDP Algorithm for Robot Planning," *IEEE International Conference on Robotics and Automation*, Vol. 3, IEEE Publ., Piscataway, NJ, 2004, pp. 2399–2404.
doi:10.1109/ROBOT.2004.1307420

[35] Ross, S., Pineau, J., Paquet, S., and Chaib-Draa, B., "Online Planning Algorithms for POMDPs," *Journal of Artificial Intelligence Research*, Vol. 32, 2008, pp. 663–704.
doi:10.1613/jair.2567

[36] Littman, M. L., Cassandra, A. R., and Kaelbling, L. P., "Learning Policies for Partially Observable Environments: Scaling Up," *Proceedings of the Twelfth International Conference on Machine Learning*, Morgan Kaufmann, San Francisco, CA, 1998, pp. 362–370

[37] Hauskrecht, M., "Value-Function Approximations for Partially Observable Markov Decision Processes," *Journal of Artificial Intelligence Research*, Vol. 13, 2000, pp. 33–94.
doi:10.1613/jair.678

[38] Fernández, J. L., Sanz, R., Simmons, R. G., and Diéguez, A. R., "Heuristic Anytime Approaches to Stochastic Decision Processes," *Journal of Heuristics*, Vol. 12, No. 3, May 2006, pp. 181–209.
doi:10.1007/s10732-006-4834-3

[39] "Safety Analysis of Proposed Change to TCAS RA Reversal Logic," RTCA, DO-298, Washington, D.C., Nov. 2005.

[40] Kochenderfer, M. J., and Chryssanthacopoulos, J. P., "Robust Airborne Collision Avoidance Through Dynamic Programming," Lincoln Lab., Massachusetts Inst. of Technology, Project Rept. ATC-371, Lexington, MA, 2011.

[41] Ramachandra, K. V., *Kalman Filtering Techniques for Radar Tracking*, Marcel Dekker, New York, 2000.

[42] Julier, S. J., and Uhlmann, J. K., "Unscented Filtering and Nonlinear Estimation," *Proceedings of the IEEE*, Vol. 92, No. 3, March 2004, pp. 401–422.
doi:10.1109/JPROC.2003.823141

[43] Kochenderfer, M. J., and Chryssanthacopoulos, J. P., "Partially-Controlled Markov Decision Processes for Collision Avoidance Systems," *International Conference on Agents and Artificial Intelligence*, Rome, Jan. 2011, pp. 62–70.

[44] Kochenderfer, M. J., Espindle, L. P., Kuchar, J. K., and Griffith, J. D., "Correlated Encounter Model for Cooperative Aircraft in the National Airspace System," Lincoln Lab., Massachusetts Inst. of Technology, Project Rept. ATC-344, Lexington, MA, 2008.

[45] Kochenderfer, M. J., Edwards, M. W. M., Espindle, L. P., Kuchar, J. K., and Griffith, J. D., "Airspace Encounter Models for Estimating Collision Risk," *Journal of Guidance, Control, and Dynamics*, Vol. 33, No. 2, 2010, pp. 487–499.
doi:10.2514/1.44867

[46] Hammer, J., "Performance Analysis of a Horizontal Miss Distance Filter for TCAS II," *Air Traffic Control Quarterly*, Vol. 6, No. 3, 1998, pp. 205–225.

[47] Winder, L. F., and Kuchar, J. K., "Evaluation of Collision Avoidance Maneuvers for Parallel Approach," *Journal of Guidance, Control, and Dynamics*, Vol. 22, No. 6, Nov.–Dec. 1999, pp. 801–807.
doi:10.2514/2.4481

[48] Kuchar, J. K., "Methodology for Alerting-System Performance Evaluation," *Journal of Guidance, Control, and Dynamics*, Vol. 19, No. 2, March–April 1996, pp. 438–444.
doi:10.2514/3.21637