10

Optimized Airborne Collision Avoidance

Mykel J. Kochenderfer

Developing robust aircraft collision avoidance logic that reliably prevents midair collisions without excessive alerting is challenging because of sensor error and uncertainty in the future paths of the aircraft. Over the past few years, research has focused on the use of a computational method known as dynamic programming for producing an optimized decision logic for airborne collision avoidance. This chapter describes recent research in modeling the collision avoidance problem as a partially observable Markov decision process and solving for the optimal strategy using dynamic programming. The performance of the system is evaluated using a statistical airspace model represented as a dynamic Bayesian network learned from recorded radar data.

10.1 Airborne Collision Avoidance Systems

Because the sky is so big and aircraft are so small, there were very few midair collisions during the early years of aviation. By the 1950s, air travel had become commonplace, and the skies became more crowded. In 1956, a midair collision over the Grand Canyon resulted in 128 fatalities. At the time, this collision was the worst commercial air disaster in history. The collision caused a press frenzy and congressional hearings, ultimately leading to the establishment of the Federal Aviation Administration (FAA) in 1958.

The establishment of the FAA led to major improvements in both airspace design and air traffic control. The airspace was designed to keep aircraft separated. For example, depending on whether aircraft were flying west or east, they were expected to fly at different altitudes. Air traffic controllers relied on ground-based radars, keeping aircraft safely separated by calling out traffic to pilots and by vectoring aircraft.

The enhancements to airspace design and air traffic control significantly improved the safety of the airspace. However, midair collisions still occurred. A midair collision involving a commercial airliner over San Diego, California, in 1978 resulted in 144 fatalities, and another commercial airliner collision over Cerritos, California, in 1986 resulted in 82 fatalities. These two collisions, in particular, convinced Congress that an additional layer of collision protection was needed in the form of an onboard system. This system would provide an independent safety net to protect against human error, both by air traffic controllers and pilots, and against the failures and limitations of visual see and avoid.

10.1.1 Traffic Alert and Collision Avoidance System

Development of an onboard capability started shortly after the midair collision over the Grand Canyon. Early concepts focused on primary radar surveillance that sends out energy pulses and measures the timing of the echo to infer distance. This approach did not work well for a variety of reasons, including the inability to accurately estimate the altitude of the intruder. The focus shifted to beacon-based systems that made use of the transponders already on board most aircraft. An aircraft would send out an interrogation over the radio data link and measure the amount of time required for the aircraft to reply. Information about altitude and intended maneuvers could also be shared across this radio data link. The initial system, called Beacon Collision Avoidance System (BCAS), was designed to operate in low-density airspace. The collision over San Diego spurred the development of the Traffic Alert and Collision Avoidance System (TCAS), which was based on BCAS but with enhancements that enabled its use in high-density airspace. The development spanned several decades. The collision over Cerritos led to Congress's mandating the use of TCAS in the United States, and now TCAS is required on all large passenger and cargo aircraft worldwide.

TCAS conducts airborne surveillance, computes advisories using a safety logic, and relays those advisories to the pilot through audio and a display. If another airborne aircraft is a potential threat, TCAS issues a traffic advisory (TA), which gives the pilots an audio announcement "Traffic, Traffic" and highlights the intruder on a traffic display. The TA is intended to help pilots achieve visual acquisition of other aircraft and prepare the pilots for a potential avoidance maneuver. If a maneuver becomes necessary, the system will issue a resolution advisory (RA), instructing the pilots to climb or descend to maintain a safe distance. There is an audio announcement of the required vertical maneuver, and the range of acceptable vertical rates is shown on the vertical speed indicator. On some aircraft, additional pitch guidance is provided to pilots.

TCAS may issue a variety of different vertical advisories, including:

- do not climb or descend,
- limit climb or descend to 500, 1000, or 2000 ft/min,
- level off,
- climb or descend at 1500 ft/min,
- increase climb or descend to 2500 ft/min, or
- maintain current vertical rate.

Depending on how the encounter evolves, TCAS may strengthen, weaken, or reverse the direction of the advisory. An RA provides vertical guidance only; TCAS does not issue horizontal maneuvers such as heading changes or turns. After the encounter has been resolved, TCAS declares "Clear of Conflict."

The logic for specifying when to alert and what advisory to issue is represented as a large collection of rules. The TCAS logic uses straight-line extrapolation to estimate the time to closest approach and the projected miss distance. If both are small, then the logic determines that an alert is necessary. If an alert is necessary, the logic will model standard climb and descend maneuvers assuming a 5-second pilot response delay, followed by a 0.25 g acceleration. It chooses the direction that provides the greatest separation from the intruder. It then models a set of different advisory rates that are consistent with the chosen direction. TCAS chooses the lowest rate that provides a required amount of separation.

Although the general steps TCAS uses to select advisories are relatively straightforward, the details of the logic are very complex. Embedded in the TCAS logic specification are many heuristic rules and parameter settings designed to compensate for sensor noise and error as well as for variability in the pilot response. There are also rules that govern when to strengthen, weaken, and reverse advisories and how to handle encounters with multiple simultaneous intruder aircraft.

10.1.2 Limitations of Existing System

TCAS has been very successful in preventing midair collisions over the years, but the way in which the logic was designed limits its robustness. Fundamental to TCAS design is the use of a deterministic model. However, recorded radar data show that pilots do not always behave as assumed by the logic. Not anticipating the spectrum of responses limits TCAS's robustness, as demonstrated by the collision of two aircraft in 2002 over Überlingen, Germany. TCAS instructed one aircraft to climb and one to descend. However, the pilot who received the TCAS climb instruction instead descended in accordance with the air traffic controller's instructions, leading to a collision with the second aircraft, which was descending according to its TCAS instruction. If TCAS recognized the noncompliance of one of the aircraft and reversed the advisory of the compliant aircraft from a descend to a climb, the collision could have been prevented. A modification was later developed to address this specific situation, but improving the overall robustness of the logic requires a fundamental design change.

Just as the airspace has evolved since the 1950s, it will continue to evolve over the next decade. Significant change will occur with the introduction of the next-generation air traffic management system, which will be based on satellite navigation. This improved surveillance will allow aircraft to fly closer together to support traffic growth. In addition, there are user classes, such as general aviation and unmanned aircraft, that would benefit from a collision avoidance capability. Unfortunately, the current version of TCAS cannot support the safety and operational requirements of this new airspace. With aircraft flying closer together, TCAS will alert pilots too frequently to be useful.

To meet these requirements, a major overhaul of the TCAS logic and surveillance system is needed. TCAS is currently limited to large aircraft capable of supporting its hardware and power requirements. The aircraft must also have sufficient performance to achieve the required vertical rates of climb or descent that the advisories currently demand. Although a collision avoidance system for small aircraft might help improve safety within general aviation, TCAS cannot be adapted for small aircraft without a costly redesign.

10.1.3 Unmanned Aircraft Sense and Avoid

Unmanned aircraft have great potential for scientific, law enforcement, and commercial applications, but they currently cannot fly in civil airspace without special authorization. Flexible airspace access requires the capability to sense and avoid other aircraft effectively. An automated airborne collision avoidance system that meets the strict safety requirements of civil aviation authorities would greatly expand the utility of unmanned aircraft.

TCAS cannot be used directly for sense and avoid for several reasons. The legacy system assumes that TCAS-equipped aircraft can achieve rates of at least 1500 ft/min to comply with initial climb and descend advisories and 2500 ft/min for strengthened advisories. Many unmanned aircraft platforms can only achieve around 500 ft/min to 1000 ft/min. These vertical constraints may make it beneficial to maneuver horizontally rather than vertically. TCAS, however, was not designed to provide heading change or airspeed guidance.

The TCAS logic was designed assuming a beacon-based surveillance system, but unmanned aircraft cannot rely solely on such a surveillance system. Because there is not a pilot on board the aircraft who can physically see and avoid other aircraft, unmanned aircraft surveillance systems must avoid all categories of aircraft—including those not equipped with beacon transponders. Sensor systems that can detect aircraft without transponders include radar, electro-optical, and infrared sensors. These different sensors have very different error characteristics and size, weight, and power constraints.

Given that it took many years to develop and certify TCAS, it is anticipated that it will be especially challenging to develop and certify different collision avoidance systems for all the various combinations of sensor systems and aircraft platforms. Efforts in the unmanned aircraft industry have largely focused on proprietary solutions for specific platforms and sensors, but a common system that could accommodate different sensor configurations and flight characteristics would significantly reduce the cost of development and certification.

10.1.4 Airborne Collision Avoidance System X

Research over the past several years has focused on formulating the problem of collision avoidance as a partially observable Markov decision process and on using dynamic programming to optimize the collision avoidance system. Simulation studies with recorded radar data have confirmed that such an approach leads to a significant improvement to safety and operational performance. The FAA has formed a team of organizations to mature the technology into a new collision avoidance system called Airborne Collision Avoidance System X (ACAS X). The system is intended to become the next international standard for collision avoidance for both manned and unmanned aircraft.

ACAS X will bring major enhancements to both surveillance and the advisory logic. The system will move from the beacon-only surveillance of TCAS to a plug-and-play surveillance architecture that supports surveillance based on global positioning system (GPS) data and that accommodates new sensor modalities. The new surveillance capabilities will also enable collision avoidance protection for new user classes, including small, general-aviation aircraft that are not currently equipped with TCAS. ACAS X represents a major revolution in how the advisory logic is generated and represented. Instead of using ad hoc rule-based specification, ACAS X represents much of the logic by using a numeric table that has been optimized with respect to models of the airspace. This new approach improves robustness, supports new requirements, and reduces unnecessary alerts. The process adopted by ACAS X greatly simplifies development and is anticipated to significantly lower the implementation and maintenance costs.

10.2 Collision Avoidance Problem Formulation

The collision avoidance problem can be formulated as a partially observable Markov decision process (POMDP) in a variety of different ways [1]–[4]. This section describes a formulation of the collision avoidance problem used in an early prototype of ACAS X designated for large transport and cargo aircraft.

10.2.1 Resolution Advisories

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

Table 10.1 Advisory Set.

	Vertical Rate Range		
Name	**Min (ft/min)**	**Max (ft/min)**	**Aural Annunciation**
COC	$-\infty$	∞	Clear of Conflict (or nothing)
DNC2000	$-\infty$	2000	Monitor Vertical Speed
DND2000	−2000	∞	Monitor Vertical Speed
DNC1000	$-\infty$	1000	Monitor Vertical Speed
DND1000	−1000	∞	Monitor Vertical Speed
DNC500	$-\infty$	500	Monitor Vertical Speed
DND500	−500	∞	Monitor Vertical Speed
DNC	$-\infty$	0	Level-off, Level-off (or Monitor Vertical Speed)
DND	0	∞	Level-off, Level-off (or Monitor Vertical Speed)
MDES	$-\infty$	current	Maintain Vertical Speed, Maintain
MCL	current	∞	Maintain Vertical Speed, Maintain
DES1500	$-\infty$	−1500	Descend, Descend
CL1500	1500	∞	Climb, Climb
SDES1500	$-\infty$	−1500	Descend, Descend NOW, Descend, Descend NOW
SCL1500	1500	∞	Climb, Climb NOW, Climb, Climb NOW
SDES2500	$-\infty$	−2500	Increase Descent, Increase Descent
SCL2500	2500	∞	Increase Climb, Increase Climb

The advisories available to ACAS X are the same as those available to TCAS and are summarized in Table 10.1. In the POMDP problem formulation, the discrete actions correspond to the various advisories—except that MCL and MDES are merged into a single “maintain” action to reduce the size of the action space. Whether the maintain action is an MCL or an MDES is determined during execution on the basis of the current vertical rate.

If the current vertical rate is within the prescribed range of the advisory when it is issued, the advisory is called *preventive*. Otherwise, the advisory is called *corrective*. In the table, DES1500, CL1500, SDES1500, SCL1500, SDES2500, and SCL2500 are always corrective. The DNC and DND can be either corrective or preventive. If DNC or DND is issued as a corrective, it is annunciated as “Level-off, Level-off” (LOLO); otherwise, it is annunicated as “Monitor Vertical Speed” (MVS). All other MVS advisories are preventive.

The availability of these 16 discrete actions depends upon the current advisory. For example, SDES2500 can only be issued after an MDES or DES1500. The constraints on transitions between advisories are carried over from the original TCAS design so as to reduce the amount of new pilot training requirements.

The *sense* of an advisory is either up or down (or neither in the case of COC). An up advisory instructs the pilots to climb or to not descend. A down sense advisory instructs the pilots to descend or to not climb. Transitioning from one sense to another is called a *reversal*. Transitioning between two advisories of the same sense is either a *strengthening* or *weakening*, depending on whether the new advisory instructs a faster vertical rate. Distinguishing these categories of transitions is important in modeling the pilot response.

10.2.2 Dynamic Model

The ACAS X prototype uses a relatively simple dynamic model of how aircraft behave. For several reasons, it is important to keep the model as simple as possible while capturing the important properties of aircraft behavior. Simpler models are easier for designers to validate and are less likely to be tailored too tightly to the idiosyncrasies of a particular airspace. In addition, because simpler models require fewer state variables, computing optimal advisories using dynamic programming is more tractable.

There are six state variables in the POMDP formulation:

- h, the altitude of the intruder relative to the own aircraft,
- $\dot{h}_0$, the vertical rate of the own aircraft,
- $\dot{h}_1$, the vertical rate of the intruder aircraft,
- τ, the time to potential collision,
- s_{adv}, the current advisory, and
- s_{res}, whether the pilot is responding to the advisory.

In past publications, s_{adv} and s_{res} were combined together in a single discrete variable called s_{RA}, but keeping these variables separate simplifies the explanation.

The discrete-time dynamic model assumes a time step of $\Delta t = 1$ s, which corresponds to the decision frequency of TCAS. The advisory response at the next time step s'_{res} is determined stochastically based on the current advisory s_{adv}, response s_{res}, and new advisory a according to

$$P(s'_{\text{res}} = \text{true} \mid s_{\text{adv}}, s_{\text{res}}, a) = \begin{cases} 1 & \text{if } a = \text{COC} \\ 1 & \text{if } s_{\text{res}} = \text{true and } s_{\text{adv}} = a \\ 1/(1+5) & \text{if } s_{\text{adv}} = \text{COC and } a \neq \text{COC} \\ 1/(1+5) & \text{if } s_{\text{adv}} \text{ and } a \text{ are opposite sense} \\ 1/(1+3) & \text{if } s_{\text{adv}} \text{ and } a \text{ are same sense} \end{cases} . \quad (10.1)$$

Because the advisory response is determined according to a Bernoulli process, the delay until response follows a geometric distribution. If the response probability at each step in the process is $(1/1 + k)$, then the mean time until response is k. Hence,

- the pilot always responds to a COC,
- once the pilot responds, the pilot continues to respond for the duration of the advisory,
- the average response delay for initial advisories is 5 s,
- the average response delay for reversals is 5 s, and
- the average response delay for a strengthening or weakening is 3 s.

A more complex pilot response model has been explored, but it brought little benefit and added significantly to the size of the state space [5].

The own acceleration $\ddot{h}_0$ is determined stochastically from s'_{res} and a. If the pilot is not responding to an advisory, then $\ddot{h}_0$ is sampled from a zero-mean Gaussian with a standard deviation of $3\,\text{ft/s}^2$. If the pilot is responding, then $\ddot{h}_0$ is chosen to bring the vertical rate to within the range required by the advisory. The magnitude of the acceleration is drawn from a Gaussian distribution with a standard deviation of $1\,\text{ft/s}^2$. Generally, the mean of the Gaussian distribution is $8.33\,\text{ft/s}^2$, but if an advisory has been reversed or strengthened to a climb or descend, then a stronger acceleration of $10.7\,\text{ft/s}^2$ is used instead. These accelerations were chosen to match those used by TCAS.

The model assumes that the intruder simply applies random accelerations, drawn independently once per second. The intruder acceleration $\ddot{h}_1$ is simply drawn from a zero-mean Gaussian with $3\,\text{ft/s}^2$ standard deviation. More sophisticated models explored in the past have captured the influence of the intruder's own collision avoidance system, but analysis has shown relatively little benefit [6].

Given a, s'_{res}, $\ddot{h}_0$, and $\ddot{h}_1$, the state is updated as follows:

$$\begin{bmatrix} h \\ \dot{h}_0 \\ \dot{h}_1 \\ \tau \\ s_{\text{adv}} \\ s_{\text{res}} \end{bmatrix} \leftarrow \begin{bmatrix} h + \dot{h}_1(\Delta t) + \frac{1}{2}\ddot{h}_1(\Delta t)^2 - \dot{h}_0(\Delta t) - \frac{1}{2}\ddot{h}_0(\Delta t)^2 \\ \dot{h}_0 + \ddot{h}_0(\Delta t) \\ \dot{h}_1 + \ddot{h}_1(\Delta t) \\ \tau - 1 \\ a \\ s'_{\text{res}} \end{bmatrix}. \tag{10.2}$$

10.2.3 Reward Function

The reward function is intended to capture the safety and operational considerations. In order to easily apply dynamic programming, the reward function can only depend upon the six state variables and the current action. Early in the development of the system, the reward function had costs assigned to near collision, issuing an initial advisory, strengthening, and reversing. Analysis of the performance of the system in simulation revealed that a variety of additional cost parameters were necessary to make the advisories more suitable and effective [7].

Table 10.2 Event rewards.

Reward	Separation (ft)	Closure (ft/min)	Event
-1	≤ 175		$\tau \leq 0$
-1			Maintain advisory with $\dot{h}_0 < 1500$ ft/min
-1			Prohibited advisory transitions
-1			Preventive crossing advisory
-0.1	> 650	< 2000	Corrective advisory
-3×10^{-2}	> 1000	< 4000	Corrective advisory
-1×10^{-2}	> 650	< 2000	Preventive advisory
-1×10^{-2}	> 500		Crossing advisory
-8×10^{-3}			Reversal
-5×10^{-3}			Strengthening
-1×10^{-3}			Weakening
-1.5×10^{-3}		> 3000	Non-MVS/LOLO
-2.3×10^{-3}		< 3000	Any advisory
-5×10^{-4}		> 3000	MVS/LOLO
$-4 \times 10^{-4} \times \Delta\dot{h}$			Crossing advisory when $\lvert\dot{h}_0\rvert > 500$ ft/s and $\dot{h}_0$ is in opposite direction of advisory
-4×10^{-4}			Maintain
-1×10^{-4}			MVS/LOLO
$-3 \times 10^{-5} \times \Delta\dot{h}$			Any advisory
-1×10^{-5}			Corrective advisory
1×10^{-9}			COC

Table 10.2 outlines the various event rewards. For a reward in the table to apply, the specified separation, closure, and event must all hold. Given a state and action, all events that apply in the table contribute to the immediate reward. All the rewards in the table are negative except for issuing COC. Some rewards depend on the vertical separation (i.e., $|h|$) and vertical closure rate (i.e., $|\dot{h}_1 - \dot{h}_0|$). Others depend on whether the advisory is a *crossing*, which is defined to occur when either (1) a down sense is issued when the intruder is below or (2) an up sense is issued when the intruder is above. Crossing advisories can require the aircraft to cross each other in altitude and are avoided when possible by TCAS.

The table contains two rows that depend on a variable $\Delta\dot{h}$. This variable is defined to be the magnitude of the difference between the minimum required change in vertical rate to comply with the advisory. In other words, if the advisory has a required range of $[\dot{h}_{\text{min}}, \dot{h}_{\text{max}}]$, then $\Delta\dot{h} = \min(|\dot{h}_{\text{min}} - \dot{h}_0|, |\dot{h}_{\text{max}} - \dot{h}_0|)$.

As discussed later, an important part of the development process involves determining how to set these reward parameters. These parameters can be adjusted to make trades between different safety and operational considerations. For example, the corrective advisory cost can be decreased to increase the rate of corrective advisories, leading to increased safety.

10.2.4 Dynamic Programming

Dynamic programming (Section 4.2) is used to compute the value function U^* assuming full observability. The dynamics, as described in Section 10.2.2, have the distributions over the accelerations $\ddot{h}_0$ and $\ddot{h}_1$ specified as continuous probability densities—and, therefore, the next state distribution T is a density. Hence, the Bellman equation involves integration instead of summation:

$$U^*(s) = \max_a \left(R(s,a) + \gamma \int T(s' \mid s,a) U^*(s') \, ds' \right). \tag{10.3}$$

Evaluating the integral analytically is not feasible, but any of the standard numerical integration methods can be used. The approach taken in the ACAS X prototype involves generating a set of weighted values for $\ddot{h}_0$ and $\ddot{h}_1$ using sigma-point sampling [8]. With the set of next states given s and a now finite, the Belman equation becomes

$$U^*(s) = \max_a \left(R(s,a) + \gamma \sum_{s'} T(s' \mid s,a) U^*(s') \right). \tag{10.4}$$

Since the state space is continuous, the local approximation value iteration algorithm (Section 4.5.1) is used to compute U^*. In particular, ACAS X uses multilinear interpolation over a grid-based discretization of the state space. The h variable is discretized into 33 points over the range ±4000 ft with finer discretization near 0 ft. The vertical rate variables $\dot{h}_0$ and $\dot{h}_1$ are discretized into 25 points each between ±10,000 ft/min with finer discretization near 0 ft/min. The time to potential collision variable τ is discretized at 1 s from 0 to 40 s.

The structure of the problem is such that only a single sweep of Gauss-Seidel value iteration (Section 4.2.6) is required. Since states with $\tau = k$ depend only on the states with $\tau = k - 1$, ordering the sweep of the states by increasing τ value results in the optimal value function. Although there are more than 26 million vertices, the process requires only a few minutes on a modern workstation. The state-action values $Q^*(s,a)$ produced through dynamic programming are saved in a lookup table. Recent research has explored methods for reducing the size of the table for use on airborne equipment with limited memory capacity [9].

10.3 State Estimation

The previous section explained how to compute the lookup tables under the assumption of full observability. During flight, there is uncertainty in the state variables that needs to be accommodated in real time. The system estimates a belief distribution b and uses it to select the best action:

$$\pi^*(b) = \arg\max_a \int b(s)Q^*(s,a)ds, \tag{10.5}$$

where $Q^*(s,a)$ comes from using multilinear interpolation on the lookup table computed offline. This method for approximating the solution to a POMDP was introduced in Section 6.4.1 as the QMDP technique. To improve the efficiency of computing Equation (10.5), instead of representing the belief b as a probability density function, the system uses a set of samples $s^{(1)},\dots,s^{(n)}$ with associated weights $w^{(1)},\dots,w^{(n)}$. Hence, Equation (10.5) becomes

$$\pi^*(b) = \arg\max_a \sum_i w^{(i)}Q^*(s^{(i)},a). \tag{10.6}$$

The belief state b can be factored as follows:

$$b(s) = b(h,\dot{h}_0,\dot{h}_1,\tau,s_{\text{adv}},s_{\text{res}}) = b(h,\dot{h}_0,\dot{h}_1)b(\tau)b(s_{\text{adv}},s_{\text{res}}), \tag{10.7}$$

where the three components are disambiguated in the text here by their arguments. This section explains how to estimate these three component distributions and how to incorporate additional operational considerations in real time through the introduction of online costs.

10.3.1 Sensor Error

Aircraft estimate their altitude using an altimeter that measures atmospheric pressure. These estimates include some error resulting from variability in pressure gradients and calibration. When an aircraft sends its altitude information to another aircraft across a radio data link, it is quantized to either 25 or 100 ft, depending on the transponder.

TCAS uses a simple alpha-beta filter to estimate vertical rate when the altitude is quantized at 25 ft [10], but it uses a much more complex, nonlinear filter when the altitude is quantized at 100 ft [11]. The TCAS filters only provide single estimates of the altitude and vertical rates.

Early in the development of ACAS X, it was found that performance could be improved by explicitly taking into account the uncertainty of the estimates of h, $\dot{h}_0$, and $\dot{h}_1$ [12]. Further work resulted in a flexible filter based on the Kalman filter (Section 6.2.2) modified to better accommodate quantization error [13]. The filter outputs a set of weighted state samples.

10.3.2 Pilot Response

The distribution over whether the pilot is responding to the previously issued advisory is updated over time according to Bayes' rule [5]. The update is a function of the distribution over own vertical rate $\dot{h}_0$, the advisory issued, and the pilot response model specified in Equation (10.1).

10.3.3 Time to Potential Collision

The time to potential collision τ cannot be known exactly because it depends upon the future trajectories of the aircraft. There are many different models of how the lateral positions of aircraft evolve over time [14], but one of the simplest models assumes white-noise acceleration vectors. Using this model, ACAS X estimates the distribution over the time until another aircraft comes within some fixed lateral distance.

There are different ways to estimate the time to potential collision. For example, one could use Monte Carlo sampling. Given the initial positions and velocities of the aircraft, the accelerations can be sampled and the paths simulated forward in time. With a sufficient number of sampled trajectories, a histogram over the time to collision can be built. One disadvantage of a sampling approach is that it can be too computationally demanding to be done in real time. In addition, certifying a safety-critical system that relies upon random-number generation can be difficult.

The approach taken in ACAS X is to compute the distributions offline and store them in a lookup table. The distributions can be computed efficiently by using an iterative process. Let $D_k(s)$ represent the probability that the other aircraft comes within a fixed lateral distance in k seconds. Computing $D_0(s)$ is straightforward, since it is simply 1 if s is a state in which the other aircraft is within the fixed lateral distance and 0 otherwise. The probability $D_k(s)$ can be computed from D_{k-1} as follows:

$$D_k(s) = \sum_{s'} T(s, s') D_{k-1}(s'), \tag{10.8}$$

where $T(s, s')$ is the probability of transitioning from one horizontal state s to another horizontal state s'. The equation above can be applied repeatedly up to some desired horizon. For the early prototype of ACAS X, this horizon is 40 s, chosen to provide adequate alert time prior to potential collision. The probability that $\tau \geq 40$ given state s is simply $1 - \sum_{k=0}^{39} D_k(s)$.

By taking advantage of symmetry, it is possible to represent the horizontal state space using three variables (see [15]). In order to store the distribution in a table, the horizontal state space must be discretized. Since there are only three variables, the discretization can be made relatively fine. To determine the distribution for an arbitrary horiziontal state not corresponding to a discretization vertex, ACAS X uses multilinear interpolation.

The horizontal state, of course, cannot be known exactly, but a distribution can be inferred from sensor measurements. Specialized state-estimation algorithms based on the unscented Kalman filter [8] have been developed for different surveillance systems. These algorithms output weighted horizontal state samples, and their associated distributions are combined together.

In cases when the aircraft are close to each other horizontally, it can be beneficial to base $b(\tau)$ on h, $\dot{h}_0$, and $\dot{h}_1$. The vertical distribution tables can be computed using the same process specified in Equation (10.8). The vertical dynamic model also assumes white-noise acceleration and can be represented using three state variables (i.e., h, $\dot{h}_0$, and $\dot{h}_1$).

10.4 Real-Time Execution

The real-time execution of ACAS X consists primarily of estimating the belief state and computing the associated state-action values by interpolating entries in the lookup table. However, there are certain situations in which the state-action values are modified online (i.e., during the execution of the logic). This section outlines these online costs and discusses the handling of multiple threats and traffic alerts.

10.4.1 Online Costs

As discussed in Section 10.2.2, the lookup table is a function of only six variables. However, there are several other variables (e.g., altitude above ground) that need to be taken into account during execution that are not part of the offline optimization. Of course, these other variables could be added to the optimization, but it would be at the expense of larger table sizes.

Since early prototypes of ACAS X were constrained by table size, research explored alternatives to increasing the number of state variables in the offline optimization. Experiments showed that simply adding costs to the action values online can be very effective. One of the first online costs explored in the development of ACAS X was for altitude inhibits. Legacy TCAS has rules that inhibit advisories on the basis of altitude to prevent disruption during landing. It was desired that these rules would be preserved in ACAS X, and so if these rules trigger an advisory inhibit during flight, infinite cost is added online to the appropriate actions.

Another example of an online cost is related to the coordination mechanism adopted from TCAS. When an aircraft issues an advisory against an intruder, it sends the intruder the sense of its advisory over the radio datalink. The intruder adjusts its advisory, if any, to be compatible (i.e., in opposite direction). In cases of simultaneous sense selection, ties are broken based on unique transponder identification numbers. If table size is not a concern, the intruder sense could be incorporated into the offline optimization [16].

The offline optimization could account for the fact that the intruder will likely maneuver on the basis of the sense information and also account for the fact that the system cannot select incompatible senses in the future. Although an online cost approach is suboptimal in general, experiments have shown that this method is effective in practice [16].

The set of online costs incorporated in a prototype of ACAS X include

- *Altitude inhibit* prevents issuing advisories below certain altitudes.
- *Advisory switch or restart* penalizes advisory changes or restarts within a certain amount of time.
- *Initialization* prevents advisories from being issued during the first few seconds of the system's starting.
- *Multiple reversals* prevents multiple reversals (unless required by coordination).
- *Bad transitions* prevents advisory sequences not allowed by TCAS.
- *No-response vertical chase* penalizes continuing an advisory in a vertical chase scenario when the pilot is not responding.
- *Compatibility* prevents issuing advisories that are not compatible with the advisories issued by other aircraft.
- *Crossing* forces an advisory when an intruder issues an advisory that requires passing through the altitude of the own aircraft.

Some of these costs are effectively infinite (e.g., altitude inhibit), but others are relatively small (e.g., advisory switch and restart). The rules governing these online costs are implemented in code and can be arbitrarily complex without impacting memory requirements.

10.4.2 Multiple Threats

The MDP in Section 10.2 assumes a single intruder. It would be relatively straightforward to add additional state variables for each additional intruder, but the table size would grow exponentially with the number of intruders. Legacy TCAS determines the best advisory for each intruder in isolation and then relies on a relatively complex set of rules to combine these individual advisories to produce a single advisory to provide to the pilot.

ACAS X, in contrast with TCAS, fuses the state-action costs associated with different intruders [17]. Experimental results have shown that this method is effective and can scale to a large number of intruders [16]. In certain situations, the system will issue a multi-threat level off (MTLO) advisory, which does not belong in the action set available to the MDP. An MTLO can be issued if two different intruders indicate they are following different senses. Since an MTLO is neither an up sense or down sense, it can remain compatible with both intruders. MTLOs are useful in "sandwich" encounters in which an aircraft must fly between two other aircraft.

Unlike TCAS, ACAS X has the capability to provide different protection modes against different aircraft. Such a capability is especially useful in situations such as closely spaced parallel operations [18]. The system may want to adopt a less conservative alerting behavior against another aircraft that is known to be on a simultaneous approach, but still provide the standard alerting behavior against all other aircraft. Early prototypes of ACAS X that implement this functionality use different lookup tables and state-estimation parameters for different protection modes and then fuse the state-action values together.

10.4.3 Traffic Alerts

The majority of the development effort on ACAS X has focused on the generation of RAs, but TAs play an important role in visual acquisition and preparing the pilots to respond to an RA. Several different approaches can be taken to produce optimized TAs with respect to the following objectives:

- Allow for appropriate lead time. The ideal time between a TA and RA is around 10 to 15 s. A lead time of less than 6 s is called a surprise RA and is unlikely to be sufficient.
- Avoid nuisance alerts. TAs without RAs should be avoided. Because of variability in pilot response, it is not possible to perfectly predict whether an RA will be issued in the future. Therefore some TAs without RAs must be tolerated. Limiting nuisance TAs must be balanced with limiting surprise RAs.
- Avoid split alerts. The system should try to prevent multiple TA segments during a single encounter.

The approach taken in ACAS X does not require enlarging or modifying the current lookup table. TAs are generated based on the value of the "no alert" action. The no-alert value is primarily influenced by the cost of collision, providing a measure of threat. The more likely a collision is, the lower the value of no alert. Comparing this value against a single threshold can cause chatter (i.e., excessive switching of the TA between on and off) when the value is near the threshold. To help prevent chatter in the original design of TCAS, TAs were required to stay on for at least 8 s. This requirement is carried over to ACAS X. To further prevent chatter and improve operational performance, ACAS X uses the following three thresholds:

- The *on threshold* is a constant value threshold. When the no-alert value descends below the on threshold, it is determined costly enough to allow a TA to be issued against the intruder.
- The *squeeze threshold* is a constant offset from the no-alert value. To help prevent excessive lead times or unnecessary TAs, this threshold allows the system to suppress TAs when no advisory values are close to the no-alert cost. If an advisory is less expensive than the no-alert cost offset by the squeeze threshold and the on threshold is also met, a TA is issued against the intruder.

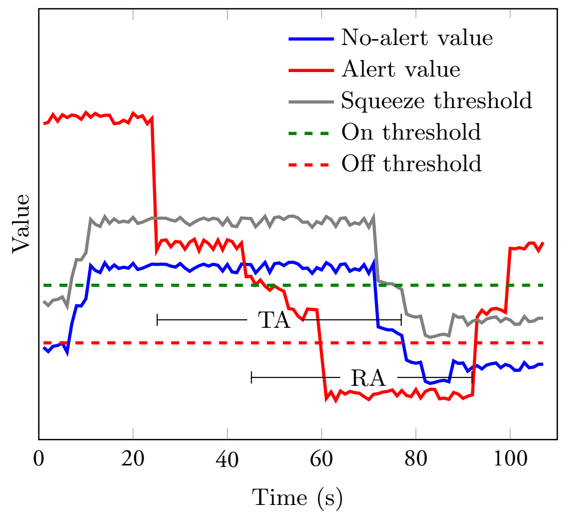


Figure 10.1 Notional TA behavior using on, off, and squeeze thresholds.

- The *off threshold* is a constant value threshold similar to the on threshold. If a TA has remained active for at least 8 s and the no-alert cost drops below the off threshold, then the TA will be discontinued.

These thresholds were chosen by running simulations of the system on radar data [19].

Figure 10.1 is a notional plot comparing the cost of the no-alert value (blue) and the highest-value advisory (alert value) (red) over the course of a single encounter. First, the no-alert cost rises above the on threshold, but the highest-value advisory is above the squeeze threshold, so a TA is not issued. Later, when the highest-value advisory descends below the squeeze threshold, a TA is issued. When the value of the highest-value advisory exceeds the value of no alert, an RA is issued [19].

10.5 Evaluation

ACAS X must accommodate many operational goals and constraints while meeting the established safety requirements. It is important that the system provide effective collision protection without unnecessarily disrupting pilots and the air traffic control system. In addition to producing as few alerts as possible, it must issue advisories that resolve encounters in a manner deemed suitable and acceptable by pilots and the operational community. This section discusses the process of safety and operational performance analysis, tuning of the logic, and flight tests of an ACAS X prototype.

10.5.1 Safety Analysis

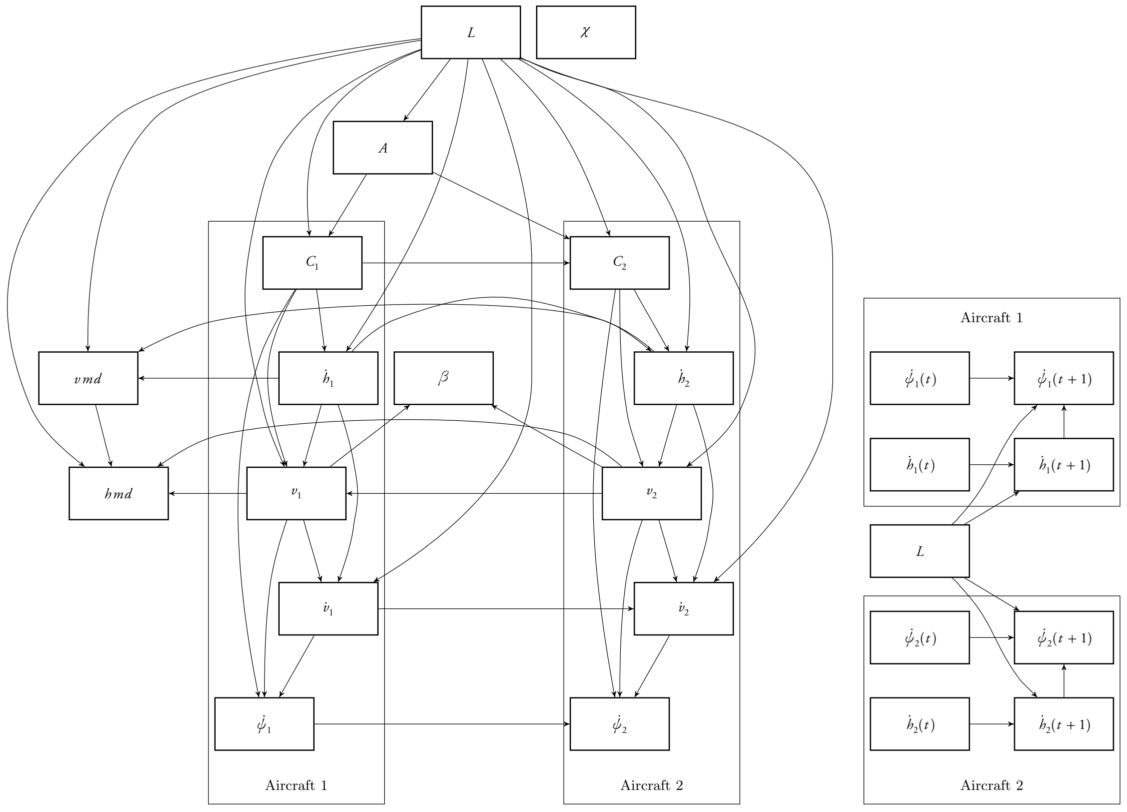
Collision risk is estimated using an encounter model, which can be used to generate encounters that are representative of the airspace. Sampling a large collection of situations from an encounter model and running them in simulation both with and without a collision avoidance system provides an estimate of the differential in collision risk. Estimated risk depends strongly on the distribution of the encounters represented by the model. Hence, it is important that the encounter geometries and aircraft behavior represented by the model be as representative of the actual airspace as possible; otherwise, the risk associated with a collision avoidance system could be significantly over- or underestimated. To ensure a representative model, a large collection of recorded surveillance data is typically used to extract state probabilities over various encounter variables.

The encounter model used in the assessment of ACAS X is based on a U.S. radar data stream from 130 short- and long-range radars with 4.5 and 12 s update rates, amounting to approximately 10 GB of data per day. The raw reports provide range, azimuth, altitude, and transponder code. Those reports are converted to latitude and longitude coordinates and tracks are fused from multiple sensors [20]. A database of tracks ranging from 1 December 2007 to 31 August 2008 was used to identify encounters that meet certain distance and time criteria, chosen to be less restrictive than the alerting criteria used by TCAS.

Associated with each of the 393,077 encounters identified are a set of static features, such as the horizontal miss distance and the initial vertical rates of the aircraft, along with a set of dynamic features, such as the turn rate and airspeed acceleration [21]. Bayesian network structure learning (Section 2.4) resulted in the topology of the initial and transition Bayesian networks shown in Figure 10.2. The distribution parameters were learned using the process outlined in Section 2.3.2.

Sampling from the model will produce encounters that are representative of the airspace. The collision avoidance system can then be run in simulation on these encounters to estimate collision risk. Safety studies typically involve simulating the aircraft as point masses and then estimating the probability of near midair collision (NMAC), defined to be when two aircraft come within 500 ft horizontally and 100 ft vertically. One of the most important metrics in safety analyses is *risk ratio*, the probability of NMAC with the collision avoidance system divided by the probability of NMAC without the collision avoidance system. If the rate of actual midair collision is to be estimated, then one must simulate aircraft wire frames and estimate distributions over aircraft types [22].

Directly sampling from the model and computing the average number of NMACs will provide an unbiased estimate of the probability that an encounter results in an NMAC. However, because of the rarity of NMAC events in the airspace, direct sampling



(a) Initial distribution.

(b) Transition distribution.

Figure 10.2 Airspace encounter model structure.

from the encounter distribution will result in the generation of relatively few NMACs. Simulating encounters that are unlikely to result in an NMAC is inefficient. Instead, it is better to produce encounters that have mostly low vertical and horizontal separation at the time of closest approach and then weight the encounters appropriately [21]. Such an approach is known as *importance sampling* and has been widely studied in the literature as a variance-reduction technique in estimation [23]. Even with importance sampling, generally on the order of hundreds of thousands of encounters are required in order to arrive at a risk ratio estimate with a relatively narrow confidence interval.

Assuming standard models for altimetry bias, active surveillance, and pilot response, the risk ratio for the current ACAS X prototype is less than 40% that of TCAS. Although the overall risk ratio for ACAS X is very encouraging, ongoing work involves identifying and categorizing areas where ACAS X can be further improved. One area of study involves analyzing the safety of the system in European airspace where the encounter distribution is known to be different because of different air traffic control procedures. Besides analyzing failure cases in encounters models that are designed to be representative of an actual airspace, the development team studied logic performance on *stress testing* models. These stress testing models probe the limits of the system on exhaustive variations of certain classes of encounters [24].

10.5.2 Operational Suitability and Acceptability

The operational performance of ACAS X is evaluated in simulation using real TCAS encounters collected under the FAA TCAS monitoring program by the TCAS Resolution Advisory Monitoring System (TRAMS). The data comprise more than 100,000 encounters that occurred during normal operations in 21 high-density terminal areas [25]. These encounters reflect all airspace classes, altitudes, domestic and foreign air carrier and business jet operations, enroute and terminal air traffic separation and procedures, airport arrival and departure routes, and a variety of intruder aircraft types and encounter geometries.

In addition to the TRAMS data, procedure-specific mini-models are also used to comprehensively assess current and future procedures of interest across a wide range of encounter dynamics [26]. These mini-models include procedures such as 500 ft and 1000 ft vertical separation encounters, closely spaced parallel approaches, and 3 nmi enroute separation procedures. As future air traffic procedures mature, additional models will be created to assess the safety and operational compatibility of ACAS X.

One of the key operational suitability metrics is the overall alert rate. The overall alert rate, as estimated from the TRAMS dataset, is 30% lower than that of TCAS. The reason why the alert rate is significantly lower than for TCAS can be seen in Figure 10.3. In these plots, both aircraft are level and are approaching head on. ACAS X has a much smaller alerting region than TCAS for this geometry. Except for the strengthening transition to increase climb or descend, all the alerts occur later than TCAS.

Ensuring that pilots will trust the ACAS X alerts is an important goal of logic tuning. During initial TCAS development, pilots identified that reversal and altitude crossing alerts warrant extra consideration because of their impact on flight crews. ACAS X performance was specifically evaluated in these areas to ensure equal or better performance than TCAS. Reversal performance was assessed with the TRAMS data, and numerous encounters were manually examined to ensure that they were acceptable. Overall, the current ACAS X prototype reduces reversals by 22%.

Due to the size of the TRAMS dataset, it is impossible to manually inspect every encounter, however, hundreds were examined over the nine logic iterations. Assessment of individual encounters is an important step in verifying that the logic tuning is effective in encouraging desired behavior. Figure 10.4 shows an example level-off encounter. The horizontal geometry (not shown) is a 90-degree crossing, common to many encounters.

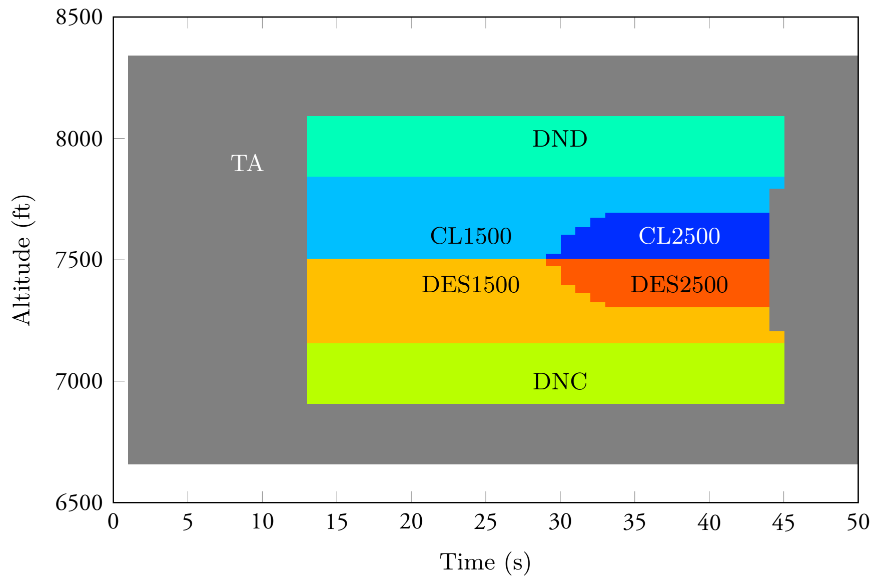
The aircraft with a collision avoidance system initially descends, and the threat climbs and then levels off. TCAS issues an initial crossing descend alert, then reverses to a climb. After the climb alert, TCAS issues a "weakening" level-off, which is intended to minimize the altitude change when sufficient vertical separation has been achieved. Finally, the clear-of-conflict notification occurs well after the closest horizontal approach.

ACAS X, in contrast, waits a little longer than TCAS and issues a level-off. The clear of conflict then occurs shortly after the closest approach. In this example, ACAS X resolves the encounter without a crossing or reversal alert. The single level-off alert did not cause a deviation from the pilots' intentions of leveling off, resulting in an acceptable resolution, while still providing safe vertical guidance.

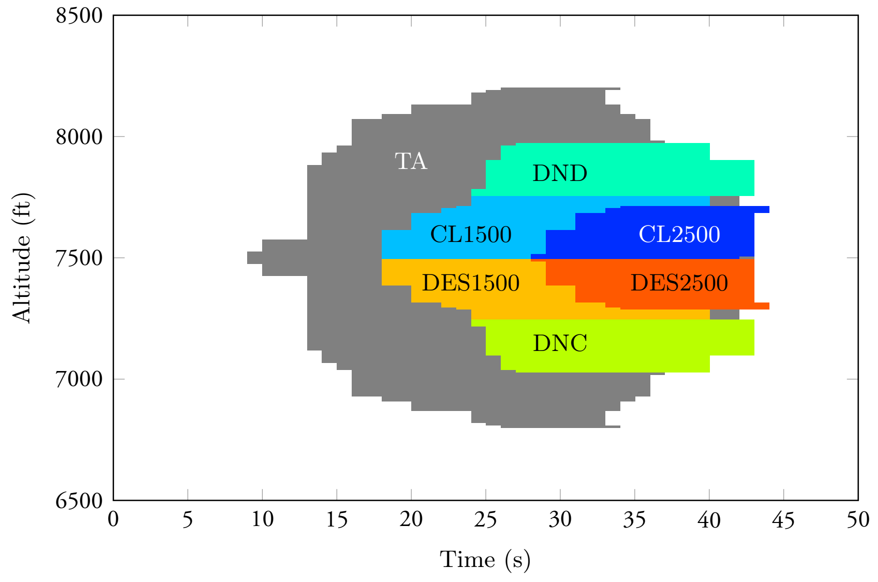
10.5.3 Parameter Tuning

A designer can modify many *design parameters* in ACAS X to achieve the goals of the system. Examples of design parameters include the offline cost of alert, the online cost of restarting an advisory, and the white-noise acceleration parameter used in the MDP. These design parameters can be adjusted to trade performance on different metrics. For example, the cost of alert could be increased to reduce the number of alerts, but at the expense of decreased safety.

These design parameters should be distinguished from the *system parameters*. System parameters are the parameters that govern the behavior of the system but are not necessarily adjusted directly by the designer. In ACAS X, many of the design parameters are also system parameters, but the system parameters also include all the millions of values stored in the lookup table. Hence, there are many more system parameters than design parameters. The large number of system parameters allows the alerting behavior (as illustrated in Figure 10.3) to be more finely adjusted to achieve better performance.

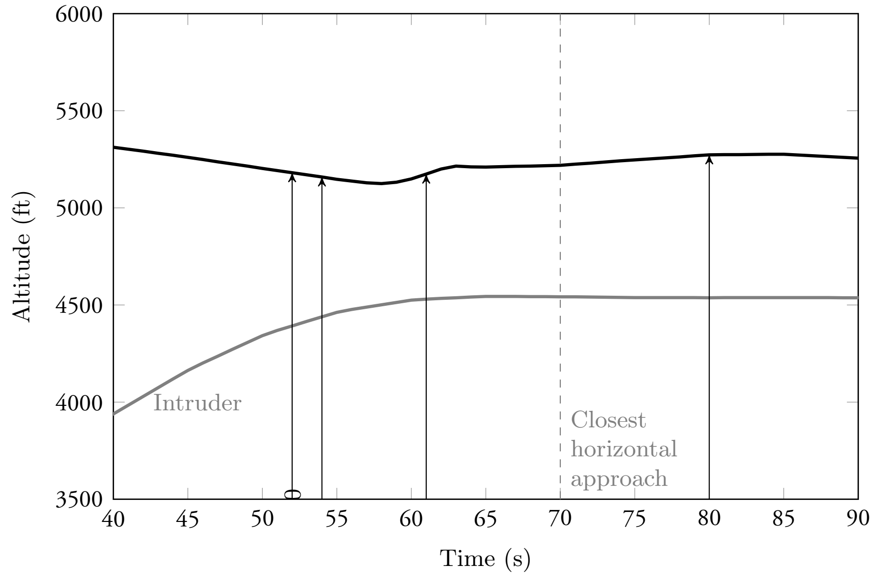


(a) TCAS.

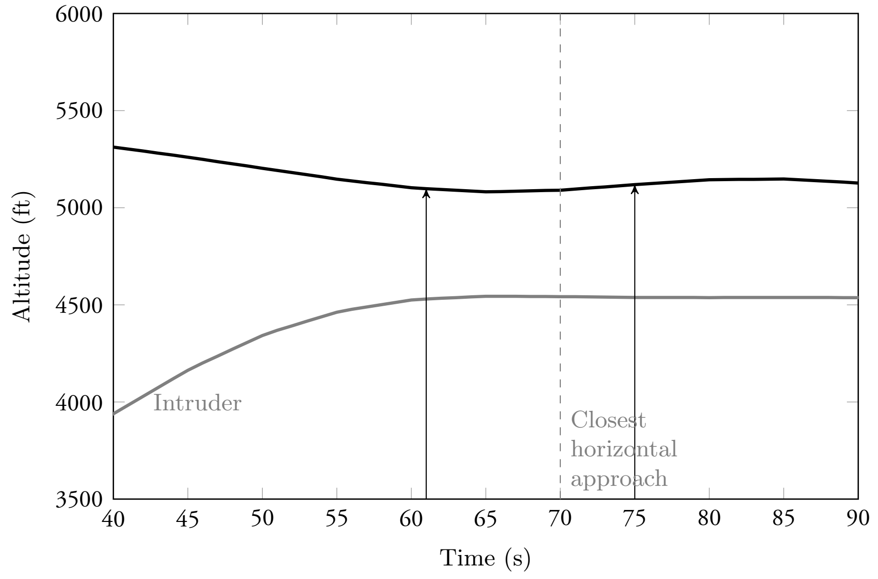


(b) ACAS X.

Figure 10.3 Logic plots.



(a) TCAS.



(b) ACAS X.

Figure 10.4 Example level-off encounter.

One of the advantages of the design approach taken in ACAS X comes from the fact that the number of design parameters is small compared to the number of system parameters. In general, the complexity of the design process can grow exponentially with the number of design parameters. Each evaluation of a design point requires millions of simulations using a variety of different models. Even with a high-performance compute cluster with 64 nodes, evaluating a single design point requires an hour. In addition, given the results of a single evaluation, it can be challenging to predict which design point to try next. Hence, during the early development of ACAS X, there was tremendous interest in automating the process of tuning the design parameters.

In order to automate the process, there needs to be some scalar utility function u defined over the design space. If there are two design points θ_1 and θ_2 and $u(\theta_1) > u(\theta_2)$, then θ_1 is preferred to θ_2. For ACAS X, u is based on the results of millions of different simulations generated by a variety of models. Since u is a scalar function, the various metrics must be weighted together appropriately. Although future work will explore the use of formal utility elicitation techniques to determine the weights (Section 3.1.4), the weights in the most recent tuning process were chosen by the consensus of an ad hoc committee of experts.

The optimization process is outlined in Figure 10.5. The first step involves *screening* the design parameters. Searching the design space is made easier by screening out the parameters that have little impact on the performance of the system. The elementary effects of the parameters were estimated by varying the parameters individually from their nominal value. The parameters that did not have a significant effect on the metrics were not included in the design search [18].

Once the important design parameters are identified, *surrogate model optimization* searches for the point in the design space that maximizes utility. Surrogate model optimization involves using past evaluations of design points to build a surrogate model of the utility function. Each step in the optimization process involves searching for the point in the design space that provides the maximum expected improvement according to the surrogate model. That point is then evaluated in simulation, and the model is updated using Bayes' rule. This process is repeated until a selection of high-performing design points are found. A similar surrogate optimization process has been used in a variety of other applications, including airfoil design [27].

10.5.4 Flight Test

ACAS X development takes advantage of modeling and simulation, but flight testing is still critical. Flight tests are required to validate simulation results and update models. They also expose the system to real environment challenges and are important in collecting pilot acceptability feedback.

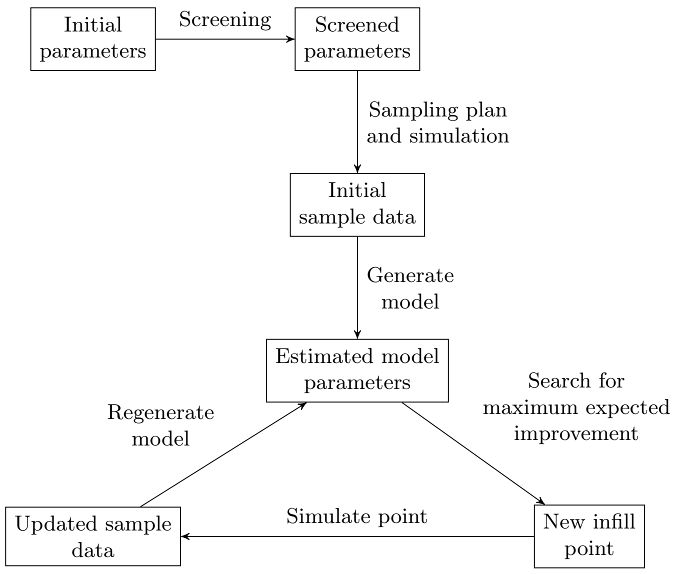


Figure 10.5 Overview of surrogate model optimization process.

The first flight test of ACAS X was conducted in August 2013 at the FAA William J. Hughes Technical Center in Atlantic City, New Jersey. MIT Lincoln Laboratory provided the algorithm specifications and lookup tables. Johns Hopkins University Applied Physics Laboratory implemented the algorithms on modified legacy hardware provided by Honeywell, Inc. The ACAS X box was flown on a Convair 580 using TCAS beacon surveillance. Encounters were scripted with a Beechcraft King Air equipped with a Mode S transponder and another Convair 580 with TCAS.

The prototype unit operated successfully for over 21 hours on 11 flights without any hardware or software failures. There were no unit resets necessary in flight. Because of memory constraints with the legacy hardware, the lookup tables were compressed using algorithms that did not significantly impact the timing of real-time lookups [9]. To ensure processing completed within each surveillance cycle, the logic processed only the four closest intruders. Research is currently under way to investigate ways of accommodating more intruders each processing cycle.

The logic was tested on a wide variety of different encounter scenarios, including those discussed in Section 10.5.2. In total, 127 test cards were flown. These encounters involved both traffic advisories and the various categories of resolutions advisories provided by TCAS. The encounters exercised the coordination mechanism with legacy TCAS. Encounters were flown both with and without a pilot response to the advisories.

ACAS X performed as desired in most of the test scenarios. However, there were a few areas in which the performance could be improved. Some undesirable alerts were issued in certain level-off encounters with 500 ft separation. Future rounds of optimization will aim to remove as many of these alerts as possible while still providing the safety required in blunder scenarios. Alerts were also observed during non-blunder parallel approaches. Improvements to surveillance anticipated in the final ACAS X system will help remove these alerts. The encounters also revealed known issues with the older version of the logic used for the test flight, such as desired reversals not being issued under certain circumstances. Some differences between the flight test and simulations were observed, resulting in modifications to the next iteration of analysis and optimization.

10.6 Summary

This chapter showed how the problem of aircraft collision avoidance can be modeled as a partially observable Markov decision process and solved using dynamic programming. Modeling and simulation reveal that such an approach can lead to a system that is less disruptive than TCAS while improving upon the level of safety TCAS provides. This research has led to the establishment of the ACAS X, which is aimed to become the next international standard for collision avoidance. As with TCAS, the regulatory effort required for both U.S. and international acceptance is intensive. Following the flight test, the standardization process commenced with the federal advisory committee, the Radio Technical Commission for Aeronautics (RTCA). The system will play an important role in the next generation of commercial aviation, as well as support the safe introduction of unmanned aircraft in civil airspace.

References

1. S. Temizer, M.J. Kochenderfer, L.P. Kaelbling, T. Lozano-Pérez, and J.K. Kuchar, "Collision Avoidance for Unmanned Aircraft using Markov Decision Processes," in *AIAA Guidance, Navigation, and Control Conference (GNC)*, 2010.
2. 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 (ITSC)*, 2010.
3. T.B. Wolf and M.J. Kochenderfer, "Aircraft Collision Avoidance using Monte Carlo Real-Time Belief Space Search," *Journal of Intelligent and Robotic Systems*, vol. 64, no. 2, pp. 277–298, 2011. DOI: 10.1007/s10846-010-9532-6.
4. H. Bai, D. Hsu, M.J. Kochenderfer, and W.S. Lee, "Unmanned Aircraft Collision Avoidance using Continuous-State POMDPs," in *Robotics: Science and Systems*, 2011.

5. J.P. Chryssanthacopoulos and M.J. Kochenderfer, "Collision Avoidance System Optimization with Probabilistic Pilot Response Models," in *American Control Conference (ACC)*, 2011.
6. M.J. Kochenderfer and J.P. Chryssanthacopoulos, "Robust Airborne Collision Avoidance Through Dynamic Programming," Massachusetts Institute of Technology, Lincoln Laboratory, Project Report ATC-371, 2011.
7. J.E. Holland, M.J. Kochenderfer, and W.A. Olson, "Optimizing the Next Generation Collision Avoidance System for Safe, Suitable, and Acceptable Operational Performance," *Air Traffic Control Quarterly*, 2013.
8. S. Julier and J. Uhlmann, "Unscented Filtering and Nonlinear Estimation," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–422, 2004. DOI: 10.1109/JPROC.2003.823141.
9. M.J. Kochenderfer and N. Monath, "Compression of Optimal Value Functions for Markov Decision Processes," in *Data Compression Conference*, 2013.
10. RTCA, *Minimum Operational Performance Standards for Traffic Alert and Collision Avoidance System II (TCAS II)*, DO-185B, 2008.
11. J.W. Andrews, "An Improved Technique for Altitude Tracking of Aircraft," Massachusetts Institute of Technology, Lincoln Laboratory, Project Report ATC-105, 1981.
12. J.P. Chryssanthacopoulos and M.J. Kochenderfer, "Accounting for State Uncertainty in Collision Avoidance," *AIAA Journal on Guidance, Control, and Dynamics*, vol. 34, no. 4, pp. 951–960, 2011. DOI: 10.2514/1.53172.
13. D.M. Asmar, M.J. Kochenderfer, and J.P. Chryssanthacopoulos, "Vertical State Estimation for Aircraft Collision Avoidance with Quantized Measurements," *AIAA Journal on Guidance, Control, and Dynamics*, vol. 36, no. 6, pp. 1797–1802, 2013. DOI: 10.2514/1.58938.
14. K.V. Ramachandra, *Kalman Filtering Techniques for Radar Tracking*. New York: Marcel Dekker, 2000.
15. M.J. Kochenderfer and J.P. Chryssanthacopoulos, "Partially-Controlled Markov Decision Processes for Collision Avoidance Systems," in *International Conference on Agents and Artificial Intelligence (ICAART)*, 2011.
16. D.M. Asmar, "Airborne Collision Avoidance in Mixed Equipage Environments," Master's thesis, Massachusetts Institute of Technology, 2011.
17. J.P. Chryssanthacopoulos and M.J. Kochenderfer, "Decomposition Methods for Optimized Collision Avoidance with Multiple Threats," *AIAA Journal on Guidance, Control, and Dynamics*, vol. 35, no. 2, pp. 398–405, 2012. DOI: 10.2514/1.54805.

18. K. Smith, M.J. Kochenderfer, W. Olson, and A. Vela, "Collision Avoidance System Optimization for Closely Spaced Parallel Operations Through Surrogate Modeling," in *AIAA Guidance, Navigation, and Control Conference (GNC)*, 2013.
19. B. Puntin and M.J. Kochenderfer, "Traffic Alert Optimization for Airborne Collision Avoidance Systems," in *AIAA Guidance, Navigation, and Control Conference (GNC)*, 2013.
20. M.J. Kochenderfer, L.P. Espindle, J.K. Kuchar, and J.D. Griffith, "Correlated Encounter Model for Cooperative Aircraft in the National Airspace System," Massachusetts Institute of Technology, Lincoln Laboratory, Project Report ATC-344, 2008.
21. M.J. Kochenderfer, M.W.M. Edwards, L.P. Espindle, J.K. Kuchar, and J.D. Griffith, "Airspace Encounter Models for Estimating Collision Risk," *AIAA Journal on Guidance, Control, and Dynamics*, vol. 33, no. 2, pp. 487–499, 2010. DOI: 10.2514/1.44867.
22. M.J. Kochenderfer, J.D. Griffith, and J.E. Olszta, "On Estimating Mid-Air Collision Risk," in *AIAA Aviation Technology, Integration, and Operations Conference (ATIO)*, 2010.
23. R. Srinivasan, *Importance Sampling: Applications in Communications and Detection*. Berlin: Springer, 2002.
24. B.J. Chludzinski, "Evaluation of TCASII Version 7.1 Using the FAA Fast-Time Encounter Generator Model," Massachusetts Institute of Technology, Lincoln Laboratory, Project Report ATC-346, 2009.
25. W.A. Olson and J.E. Olszta, "TCAS Operational Performance Assessment in the U.S. National Airspace," in *Digital Avionics Systems Conference (DASC)*, 2010.
26. J.E. Holland, M.J. Kochenderfer, and W.A. Olson, "Optimizing the Next Generation Collision Avoidance System for Safe, Suitable, and Acceptable Operational Performance," in *USA/Europe Air Traffic Management Research and Development Seminar*, 2013.
27. A.I.J. Forrester, A.J. Keane, and N.W. Bressloff, "Design and Analysis of Noisy' Computer Experiments," *AIAA Journal*, vol. 44, no. 10, pp. 2331–2339, 2006. DOI: 10.2514/1.20068.