AIAA Guidance, Navigation and Control Conference and Exhibit
18 - 21 August 2008, Honolulu, Hawaii

AIAA 2008-6629

A Bayesian Approach to Aircraft Encounter Modeling*

Mykel J. Kochenderfer†, Leo P. Espindle‡,
James K. Kuchar§, and J. Daniel Griffith¶
Lincoln Laboratory, Massachusetts Institute of Technology, Lexington, MA 02420

Aircraft encounter models can be used in a variety of analyses, including collision avoidance system safety assessment, sensor design trade studies, and visual acquisition analysis. This paper presents an approach to airspace encounter model construction based on Markov models estimated from radar data. We use Bayesian networks to represent the distribution over initial states and dynamic Bayesian networks to represent transition probabilities. We apply Bayesian statistical techniques to identify the relationships between the variables in the model to best leverage a large volume of raw aircraft track data obtained from more than 130 radars across the United States.

I. Introduction

One of the main challenges to integrating unmanned aircraft into the National Airspace System is the development of systems that sense and avoid local air traffic. If designed properly, these collision avoidance systems could provide an additional layer of protection that maintains the current exceptional level of aviation safety. However, due to their safety-critical nature, rigorous assessment is required before sufficient confidence can exist to certify collision avoidance systems for operational use. Evaluations typically include flight tests, operational impact studies, and simulation of millions of traffic encounters with the goal of exploring the robustness of the collision avoidance system. Key to these simulations are so-called encounter models that describe the statistical makeup of the encounters in a way that represents what actually occurs in the airspace.

One system that has been rigorously tested in this manner is the Traffic alert and Collision Avoidance System (TCAS). As part of the TCAS certification process in the 1980s and 1990s, several organizations tested the system across millions of simulated close encounters and evaluated the risk of a near mid-air collision (NMAC, defined as separation less than 500 ft horizontally and 100 ft vertically).[1–4] This analysis ultimately led to the certification and U.S. mandate for TCAS equipage on large transport aircraft. More recently, Eurocontrol and ICAO performed similar sets of simulation studies to support European and worldwide TCAS mandates.[5,6]

The design of a collision avoidance system represents a careful balance between preventing collision and not maneuvering unnecessarily. This balance is strongly affected by the types of encounter situations to which the system is exposed. It is therefore important that simulated encounters are representative of those that occur in the airspace. Hence, tremendous effort has been made by various institutions since the early 1980s to develop encounter models.[1,3,7–10] The primary contribution of this paper is to introduce a new approach to encounter modeling that is based on a Bayesian statistical framework. The advantage of such a theoretical framework is that it allows us to optimally leverage available radar data to produce a model that is representative of reality.

There are two fundamental types of close traffic encounters. In the first, both aircraft involved are cooperative (i.e., have a transponder) and at least one is in contact with air traffic control. It is then likely that at least one aircraft will receive some notification about the traffic conflict and begin to take

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

†Technical Staff, Surveillance Systems, 244 Wood Street.
‡Assistant Staff, Surveillance Systems, 244 Wood Street.
§Assistant Group Leader, Surveillance Systems, 244 Wood Street. Senior Member AIAA.
¶Associate Staff, Surveillance Systems, 244 Wood Street.

action before a collision avoidance system gets involved. We term this type of encounter correlated because the trajectories of each aircraft may involve maneuvers that are correlated to some degree due to this prior intervention. The second type of encounter involves at least one non-cooperative aircraft or two aircraft flying under Visual Flight Rules (VFR) without flight following (i.e., using a transponder Mode A code of 1200). In these encounters, it is unlikely that air traffic control would become involved prior to the close encounter; rather the two aircraft must rely solely on visual acquisition at close range to remain separated. Such encounters tend to be uncorrelated since there is no coordinated intervention prior to the close encounter. The assumption behind uncorrelated encounters is that the two aircraft blunder into each other; it is up to the collision avoidance system—not the encounter model—to describe how the aircraft respond to the encounter. A complete evaluation of unmanned systems will require analysis using both correlated and uncorrelated models.

There are two factors leading to the need to update prior encounter models. Encounter models developed for TCAS involved only correlated encounters because TCAS is only able to track cooperative aircraft and is typically used on aircraft receiving air traffic control services. In addition, airspace has changed in the intervening 20 years due to new aircraft types (e.g., regional jets) and new procedures (e.g., the use of Reduced Vertical Separation Minimum at higher flight levels). Accordingly, our effort has been focused on both completely redesigning and updating the correlated encounter model and on developing an entirely new uncorrelated encounter model. This paper focuses on the latter uncorrelated model; correlated encounter models are also being developed using the same framework presented in this paper.

To demonstrate the approach, we describe a model of nominal VFR flight estimated from secondary radar returns from aircraft squawking 1200. The model of nominal VFR flight can be used to simulate the behavior of realistic intruders in uncorrelated encounters. Typically, VFR aircraft have more frequent accelerations, both vertically and horizontally, than previous models allowed.[3,7–11] Our model permits maneuvers every second.

The next section explains how we model the dynamics of the variables that describe the behavior of aircraft in encounters. Section III describes how to estimate the parameters of the model from radar data. Section IV shows how to sample from the model to produce simulated trajectories. Section V explains how to combine models of nominal flight to produce pairwise encounters. Section VII demonstrates how our encounter model can be applied to collision avoidance safety evaluation and visual acquisition analysis. The final section concludes and discusses further work. An appendix provides a brief review of Bayesian networks.

II. Model

We model nominal flight, i.e. flight without avoidance maneuvering, using a Markov process represented by a dynamic Bayesian network. A Markov process is a stochastic process where the probability distribution over future states is conditionally independent of past states given the present state. In other words, one only needs to know the present state to predict the next state.

The states in our model specify how the position, altitude, and airspeed change over time. In particular, each state specifies a vertical rate $\dot{h}$, turn rate $\dot{\psi}$, and airspeed acceleration $\dot{v}$. Given an initial airspeed v, horizontal coordinates (x, y), heading ψ, altitude layer L, and airspace class A, we can infer from our model how the aircraft trajectory evolves over time. We use Bayesian networks to model the probability distribution over initial states and to model how states change probabilistically over time. We chose to use Bayesian networks in our modeling effort because they compactly represent multivariate probability distributions and because there is well-developed theory on how to use them to efficiently learn distributions from data. The appendix provides some background on Bayesian networks. The remainder of this section outlines our approach.

II.A. Model Variables

There are six variables in the uncorrelated encounter model:

- **Airspace class** A**:** This variable may take on one of four values: B, C, D, and O, indicating which class of airspace the aircraft is in. The values B, C, and D correspond to the controlled airspace classes defined by the FAA. The value O represents "other airspace," that is airspace, such as Class A, E, G, that is not B, C, or D. The airspace class variable was incorporated into our model to account for the variation in how aircraft fly in different airspace classes. Note that there should be no VFR aircraft

in Class A due to the requirement that aircraft in that Class of airspace fly under Instrument Flight Rules.

- **Altitude layer L:** Airspace is also divided into four altitude layers, in a process similar to prior encounter models developed by Eurocontrol. The first layer spans from 500 to 1200 ft Above Ground Level (AGL) to capture aircraft in the traffic pattern or performing low-level maneuvers. The second layer spans a transition zone from 1200 to 3000 ft AGL, the cruise altitude where the hemispheric rule begins. The third layer spans from 3000 ft AGL to 5000 ft AGL covering a mix of low-altitude enroute and maneuvering aircraft. The fourth layer includes airspace above 5000 ft AGL and would cover most enroute VFR traffic.

- **Airspeed v:** We model true airspeed and allow it to vary during flight in the absence of wind.

- **Acceleration $\dot{v}$:** Unlike previous encounter models, we allow airspeed acceleration to vary at every second.

- **Turn rate $\dot{\psi}$:** Turn rate is permitted to change every second in our model. The prior European and ICAO cooperative models allowed only a single turn during an encounter.

- **Vertical rate $\dot{h}$:** The vertical rate is permitted to change at every second. All prior cooperative models allowed only a single vertical acceleration period during an encounter.

The remainder of this section explains how we model the joint probability distribution over these variables. A core concept of the model is that first, parameters are randomly selected to describe the situation at the beginning of an encounter simulation. Second, certain parameters may vary dynamically during an encounter so that realistic turning, climbing, or accelerating trajectories can be generated. Thus there are two separate probability distributions in the model: an initial distribution to set up an encounter situation, and a transition distribution to describe how the trajectory evolves over time.

The probability distributions are represented using a Bayesian network. The network describes which parameters have dependencies on other parameters. For example, it would be expected that the turn rate of an aircraft would be dependent on its vertical rate. Without properly capturing this dependency, unrealistic situations may be generated, e.g., involving aircraft with simultaneously high climb rates and high turn rates. A Bayesian scoring process (see appendix) was used to evaluate different Bayesian network structures and arrive at a structure that optimally represents the observed radar data.

II.B. Initial Distribution

In our model of aircraft flight, we are concerned with modeling the distribution over the initial values of $\dot{h}$, $\dot{\psi}$, $\dot{v}$, v, L, and A. Figure 1(a) shows the structure we use for our modeling effort. Other structures are certainly plausible, and it is possible to compare different structures using Bayesian scoring (see appendix) to determine which structure is more likely given the data. We chose the network structure in Figure 1(a) because it had the highest score among the selection of candidate networks we considered.

Given a structure, sufficient statistics extracted from data, and a Bayesian prior, we then sample from the Bayesian network to produce initial airspace classes, altitude layers, vertical rates, turn rates, airspeeds, and accelerations that are representative of those found in the data. The boxes and arrows used in the structural diagram show the order in which this sampling occurs. For example, based on the structure in Figure 1(a), to determine the initial state of the aircraft we first randomly determine an airspace class A. Once the airspace class has been determined, an altitude layer L is selected. The probabilities associated with each altitude layer depend on which airspace class was chosen earlier. Once A and L have been selected, we then randomly select airspeed v, and so on. Alternately, we could assign outright an airspace class and/or altitude layer for a particular study and then randomly select values for the remaining variables.

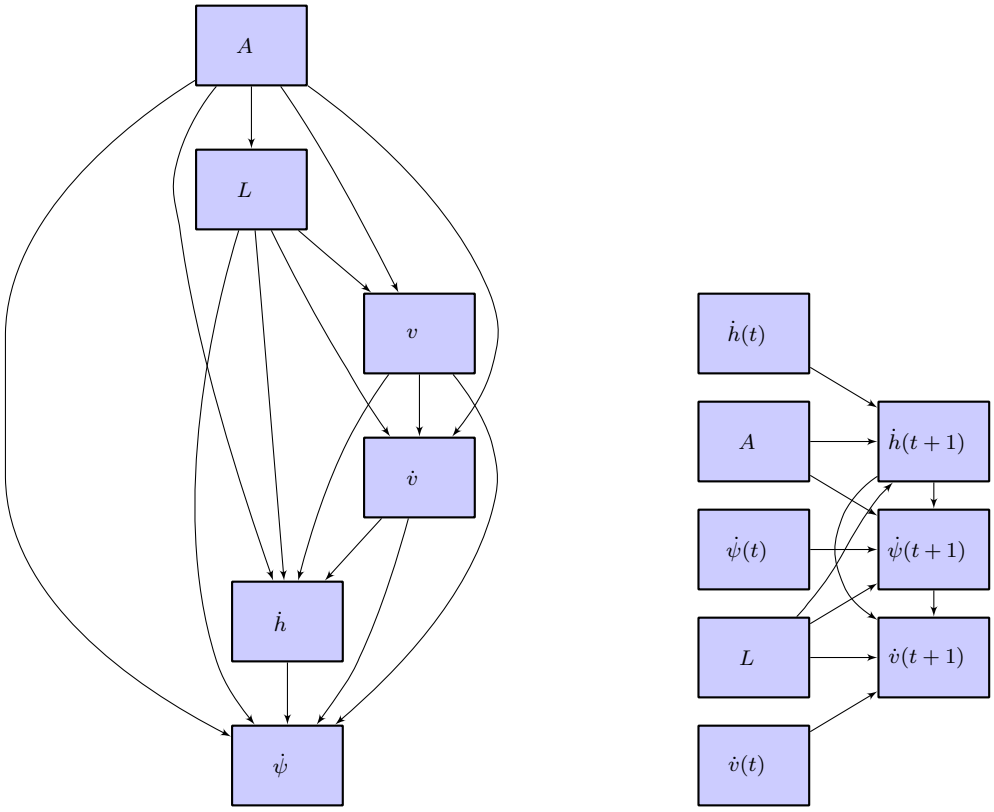
II.C. Transition Distribution

We use a separate Bayesian network to model how the variables $\dot{h}$, $\dot{\psi}$, and $\dot{v}$ evolve over time. The structure we use is shown in Figure 1(b). In this network, we have a first layer that represents the state of the system at the present time step and a second layer that represents the state of the system at the next time step. There may be dependencies between layers and within the second layer. Such a two-layer temporal Bayesian

network is known as a dynamic Bayesian network.[12,13] Parameter and structure learning in dynamic Bayesian networks is similar to regular Bayesian networks (see appendix). Again, we chose the highest-scoring network structure among our candidate network structures. Given a structure, sufficient statistics extracted from data, and a prior, we then sample from the Bayesian network to determine the next vertical rate, turn rate, and acceleration command that are representative of what was observed in the data.

In general, time steps in dynamic Bayesian networks may be of any duration, but for our modeling effort we chose steps of 1 s. Shorter time steps allow for more frequent variations in airspeed, vertical rate, and turn rate, but they require more computation per unit of simulation time. Time steps of 1 s balance maneuver complexity with computation.

A complete trajectory is constructed by updating the aircraft state in 1 s intervals. Within each interval, the three derivative variables $\dot{h}$, $\dot{\psi}$, and $\dot{v}$ are treated as target values and held constant. A dynamic model (which is beyond the scope of this report) is used to compute and update the aircraft state at each time step based on these piecewise-constant target values.



(a) Initial distribution. (b) Transition distribution

Figure 1. Bayesian networks representing the variable dependency structure for the initial and transition distributions.

III. Estimation

This section describes the processing required to transform radar tracks into sufficient statistics that may be used to model uncorrelated encounters. Figure 2 outlines this process.

III.A. Radar Tracks

Our radar data comes from the 84th Radar Evaluation Squadron (RADES) at Hill AFB, Utah. RADES receives radar data from FAA and Department of Defense sites throughout the United States. They maintain continuous real-time feeds from a network of sensors, including long-range ARSR-4 radars around the perimeter of the United States and short-range ASR-8, ASR-9, and ASR-11 radars in the interior. Radar ranges vary from 60 to 250 NM. Figure 3 shows the coverage by the 134 sensors.

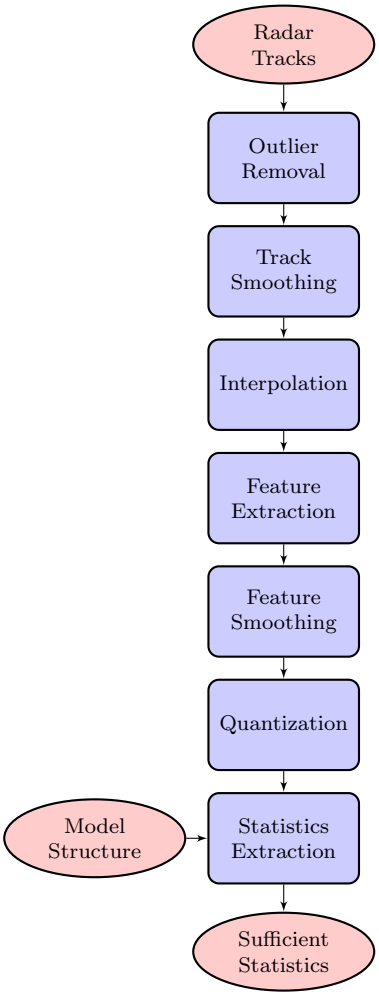


Figure 2. Estimation process flow.

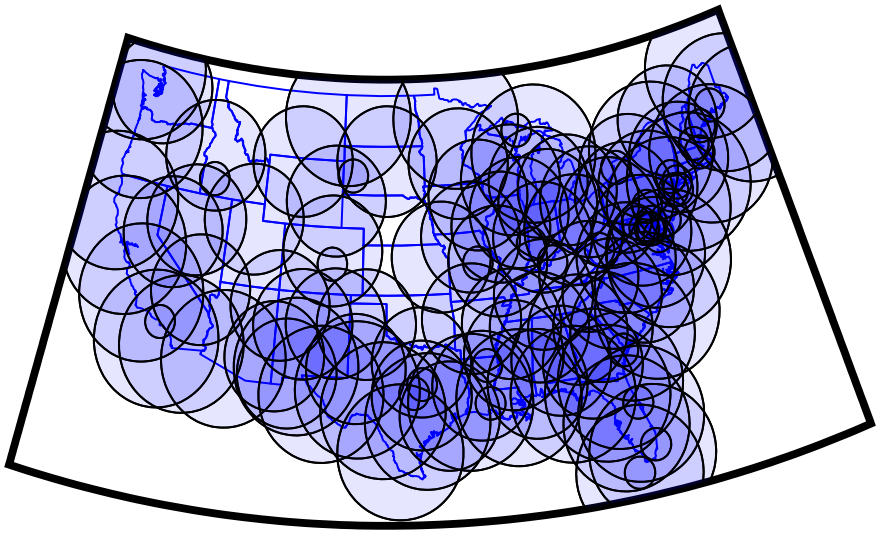


Figure 3. Radar coverage map.

National Offload Program (NOP) data is another source that we could have used to construct our model. An advantage of NOP data is the inclusion of flight-plan and aircraft-type information. However, NOP data is post automation, like ETMS. NOP does not include data from Department of Defense sensors and does not have as comprehensive coverage as our RADES feed.

Note that there are a number of advantages to our RADES data feed compared to the Enhanced Traffic Management System (ETMS) data often used in airspace analyses. ETMS data include only cooperative aircraft on filed Instrument Flight Rules flight plans and provides updates once per minute showing aircraft position after processing by air traffic control automation. In contrast, RADES data is continuously streaming directly from the radar, includes primary-only radar returns as well as all cooperative transponder returns (whether on a flight plan or not), providing track updates every 5 or 12 seconds without being affected by automation systems. This ensures that our filters and trackers have the best raw data with which to begin processing.

To build our model, we collected VFR (1200-code) beacon reports from December 1, 2007 to December 7, 2007 and June 1, 2008 to June 7, 2008, amounting to 74,000 flight hours after fusion. The raw radar data is first processed using a tracking algorithm developed at Lincoln Laboratory.[14] A fusion algorithm, also developed at Lincoln Laboratory,[15] then fuses tracks from multiple sensors to give one global view of all the tracks in U.S. airspace. We eliminated tracks that had fewer than ten scans. We found that approximately ten scans are required to accurately estimate the various maneuver rates. We also eliminated tracks if any of their associated reports were inside special use airspace whose boundaries are defined in the Digital Aeronautical Flight Information File (DAFIF), 8th Edition, managed by the National Geospatial-Intelligence Agency (NGA).

III.B. Outlier Removal

The first step in processing the raw radar tracks is to detect and remove outliers. In the horizontal plane, we remove jumps with ground speeds above 600 kt using the following algorithm. We begin by estimating the speed between each sample point by dividing the distance between samples by the time interval between samples. Samples on either side of segments where the speed is above the threshold of 600 kt are stored in a list of candidates for removal. We iterate through the list of candidates and remove the one that minimizes the sum of speeds above the set threshold. The process repeats until there are no longer any segments with speeds above the threshold.

In the vertical plane, we remove missing Mode C altitude reports from consideration. Then, using the same process as used for the horizontal plane, we remove outliers with vertical rates greater than 5000 ft/min or less than -5000 ft/min. We remove altitude reports that come before the first position report or after the last position report to prevent extrapolation. We also remove position reports that come before the first altitude report or after the last altitude report, also to prevent extrapolation. After outlier removal, we discard tracks with fewer than ten valid scans.

III.C. Track Smoothing

After removing any outliers from a track, we smooth the remaining data points, first horizontally and then vertically. We use the same smoothing scheme for both horizontal and vertical smoothing. We use the following general formula to transform a raw trajectory $(t_1, \boldsymbol{x}_1), \ldots, (t_n, \boldsymbol{x}_n)$ to a smoothed trajectory $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n$.

$$\boldsymbol{y}_i = \frac{\sum_j w(t_i, t_j)\boldsymbol{x}_i}{\sum_j w(t_i, t_j)}, \tag{1}$$

where $w(t_i, t_j)$ is a weighting function that monotonically decreases as the difference between t_i and t_j increases. For the weighting function, we use the following definition based on a Gaussian kernel with standard deviation σ,

$$w(t_i, t_j) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(t_i - t_j)^2}{2\sigma^2}\right). \tag{2}$$

When smoothing horizontally, we use $\sigma = 5$ s. When smoothing vertically, we use $\sigma = 15$ s. A larger σ is required for vertical smoothing because of 100 ft Mode C quantization. We chose these values for σ after trying different standard deviations on a sampling of horizontal and vertical profiles in our data set; the chosen values preserve the underlying tracks while removing noise.

III.D. Interpolation

The time interval between radar scans in our data is much longer than the 1 s time step of our dynamic Bayesian network. Terminal (short range) radars scan aircraft approximately every 5 s, and en route (long range) radars scan aircraft every 10 to 12 s. Additionally, it is common for sensors to skip one or more consecutive scans of a target and some scans produce outliers that we remove (Section III.B). Hence, we need to use interpolation to estimate the parameters in our dynamic Bayesian network. We chose a piecewise-cubic Hermite interpolation scheme that preserves monotonicity and shape.[16]

Figure 4 shows the result of outlier detection, smoothing, and interpolation on a track from our data set. Figure 5 shows the result of piecewise-cubic Hermite interpolation on a smoothed track.

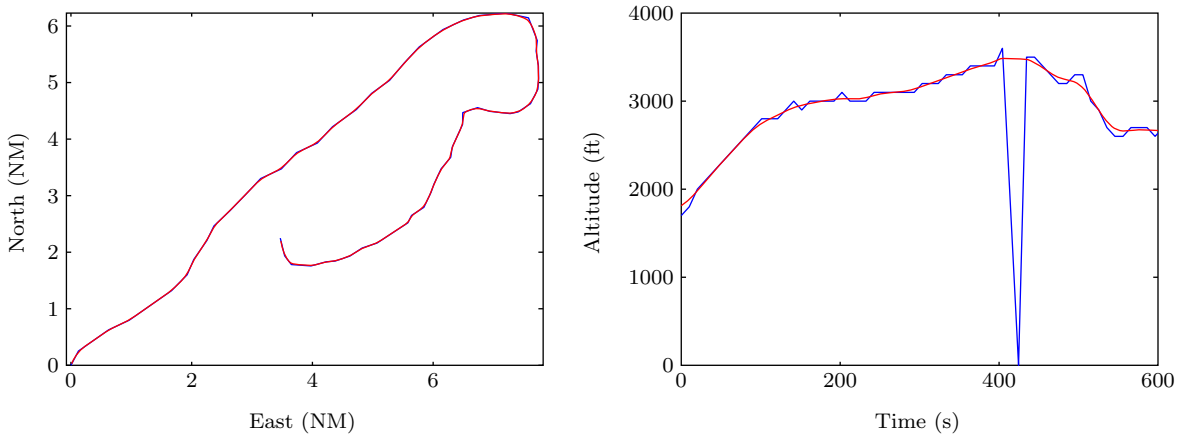


Figure 4. Preprocessing. Blue lines show the raw track. Red lines show the track after outlier removal, smoothing, and interpolation.

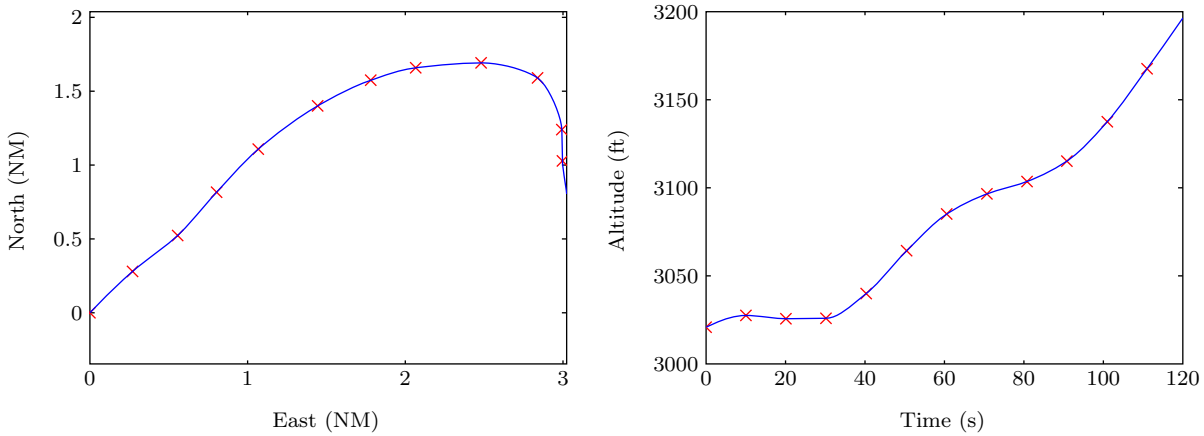


Figure 5. Piecewise-cubic Hermite interpolation on a smoothed track. Red crosses indicate smoothed data points, and the blue curve shows the interpolation.

III.E. Feature Extraction

Feature extraction involves converting an interpolated track into sequences of airspace classes, altitude layers, airspeeds, vertical rates, turn rates, and accelerations.

- **Airspace class:** We estimate latitude and longitude of radar returns using an algorithm developed at Lincoln.[17] From altitude estimates and latitude and longitude estimates, we determine the class of airspace by searching through the National Airspace System Resources (NASR) database provided by the FAA. Since the altitude estimates are based on Mode C reports of pressure altitude, uncorrected for barometric variation, it is possible that the airspace of some tracks are identified incorrectly. We expect that some inaccuracy in airspace class identification due to barometric variation has a negligible impact on our model.

- **Altitude layer:** Altitude above ground level (AGL) determines the altitude layer in our model. We estimate altitude AGL by subtracting an estimate of ground elevation from pressure altitude. Our estimates of ground elevation come from Digital Terrain Elevation Data (DTED) provided by the National Geospatial-Intelligence Agency (NGA). We use DTED Level 0, which has post spacing of 30 arcseconds (approximately 900 meters).
- **Airspeed:** The true airspeed at time t is given by

$$v(t) = \sqrt{(x(t+1) - x(t))^2 + (y(t+1) - y(t))^2 + (h(t+1) - h(t))^2}\,.$$

- **Vertical rate:** The vertical rate is estimated from the smoothed and interpolated altitudes estimated from Mode C reports. The vertical rate at time t is given by $\dot{h}(t) = h(t+1) - h(t)$.
- **Turn rate:** We first compute the heading along the interpolated track. The heading at time t is given by $\psi(t)$ and corresponds to the direction from $(x(t), y(t))$ to $(x(t+1), y(t+1))$. To compute the turn rate at time t, we find the acute change in heading between $\psi(t)$ and $\psi(t+1)$. Turns to the right have positive turn rates, and turns to the left have negative turn rates.
- **Acceleration:** To find the acceleration at a particular point, we average the change in airspeed per unit time looking forward one time step and looking back one time step.

Figure 6 shows the result of feature extraction on the same track shown in Figure 4.

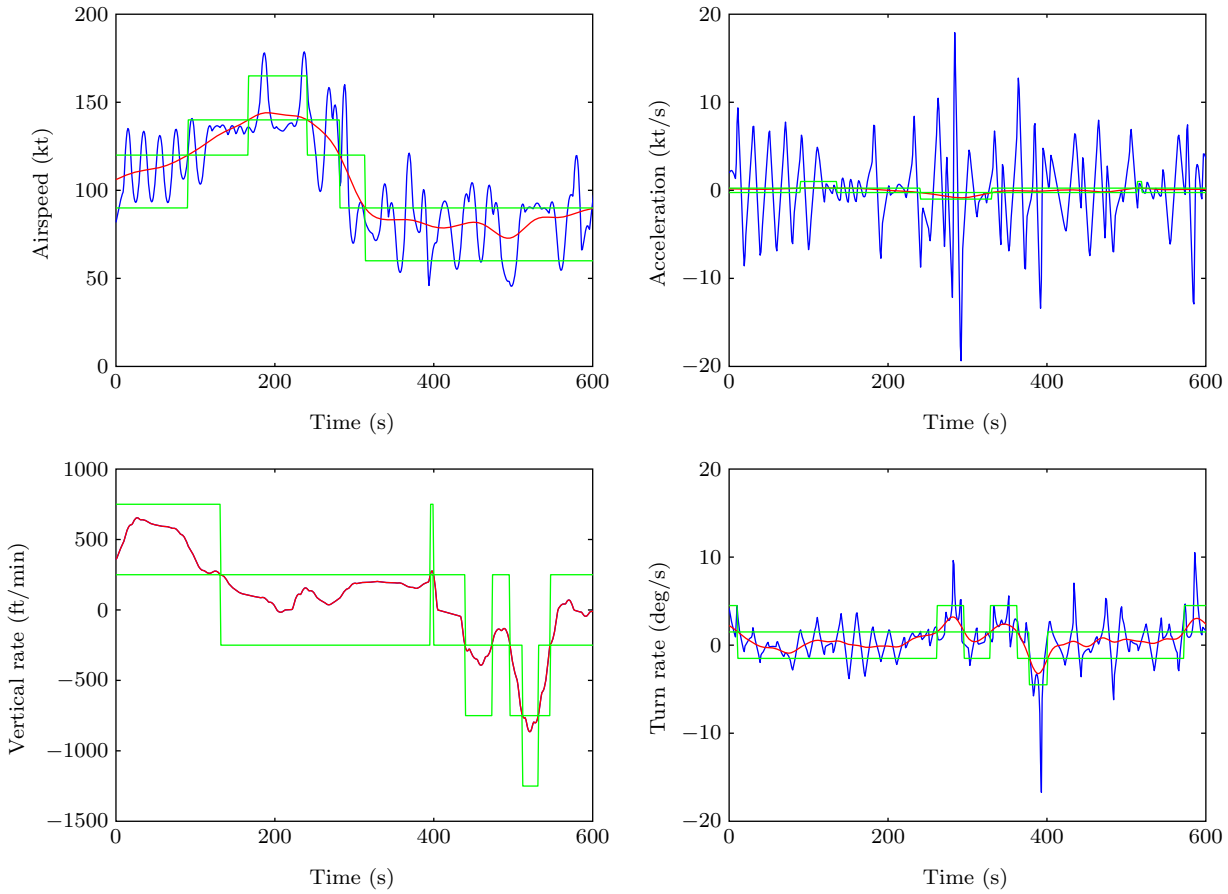


Figure 6. A plot of extracted features over time. Blue lines show features before smoothing, and red lines show features after smoothing. The green blocks show the outlines of the bins to which the features belong.

III.F. Feature Smoothing

We then smooth the extracted features using the same smoothing scheme we used for tracks (Section III.C). For turn rate, airspeed, and acceleration, we set σ to 10 s, 20 s, and 20 s respectively. We choose these

numbers large enough so that noise is removed from the measurements but low enough so that the underlying properties of the maneuvers are not lost. We do not smooth vertical rates because the altitudes are already smoothed (Section III.C).

III.G. Quantization

In order to be modeled by a discrete Bayesian network, it is necessary to quantize the features. We quantize continuous values by defining a sequence of cut points $c_1, \ldots, c_n$. Values less than c_1 are in the first bin, values greater than c_n are in the $(n+1)$th bin, and values in the half-open interval $[c_{i-1}, c_i)$ are in the ith bin. The cut points we used for quantization are listed in table 1. The cut points were chosen to capture the variation of the features as shown in the histograms in figure 7.

Table 1. Cut points used for feature quantization.

	Cut Points
A	B, C, D, O
L	$1200, 3000, 5000$
v	$30, 60, 90, 120, 140, 165, 250$
$\dot{v}$	$-1, -0.25, 0.25, 1$
$\dot{h}$	$-1250, -750, -250, 250, 750, 1250$
$\dot{\psi}$	$-6, -4.5, -1.5, 1.5, 4.5, 6$

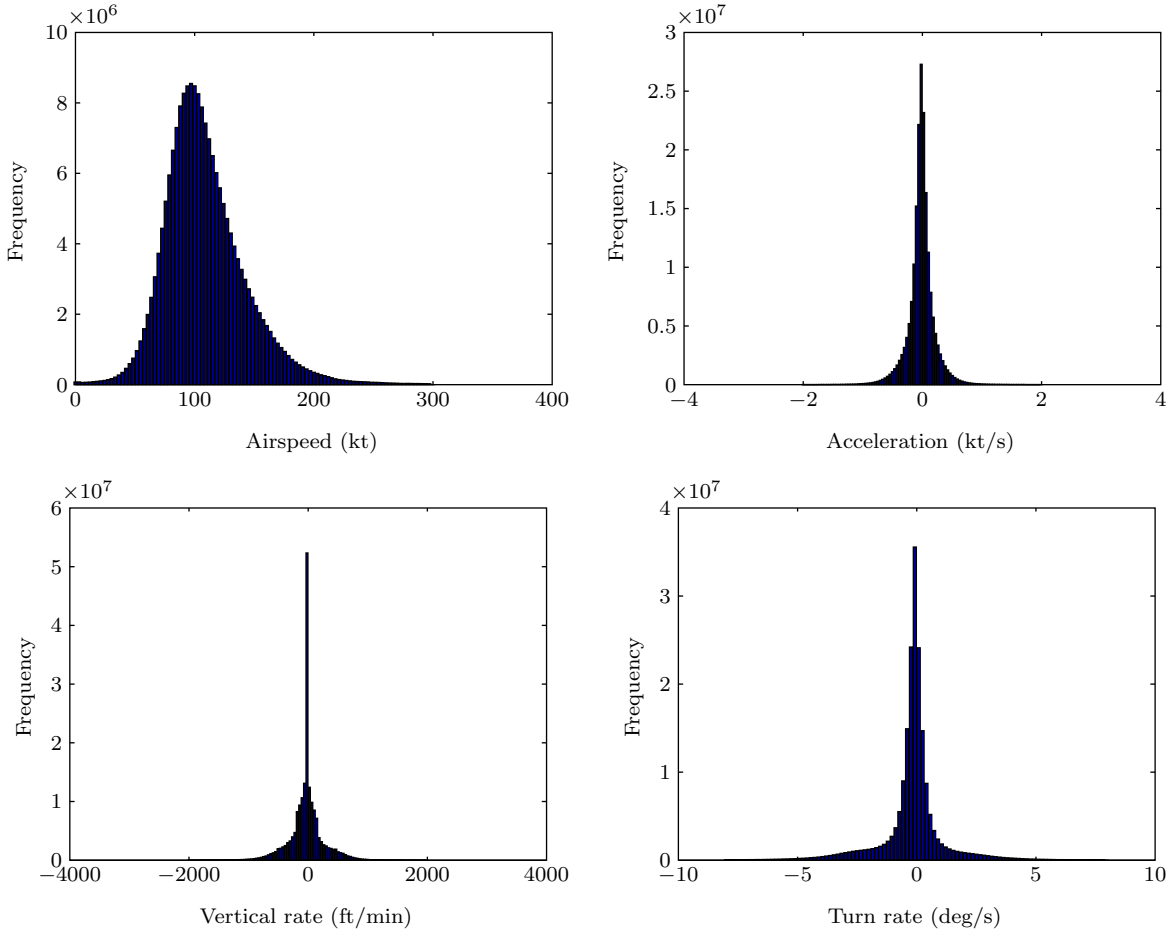


Figure 7. Feature histograms of observed radar data.

III.H. Statistics Extraction

With structures for the initial and transition distributions and the quantized features from a set of tracks, we are able to collect the sufficient statistics to estimate the parameters for our model. For the two Bayesian networks, the sufficient statistics are simply the counts N_{ijk} of the various features (see appendix).

IV. Sampling

Once we process the data as described in the previous section, we can use the model structure and sufficient statistics to produce new tracks that are representative of the ones observed by radar. The first step involves sampling from the discrete Bayesian networks representing the initial and transition distributions. The second step involves converting the discrete sample into a continuous sample by sampling within bins. Figure 8 illustrates this process.

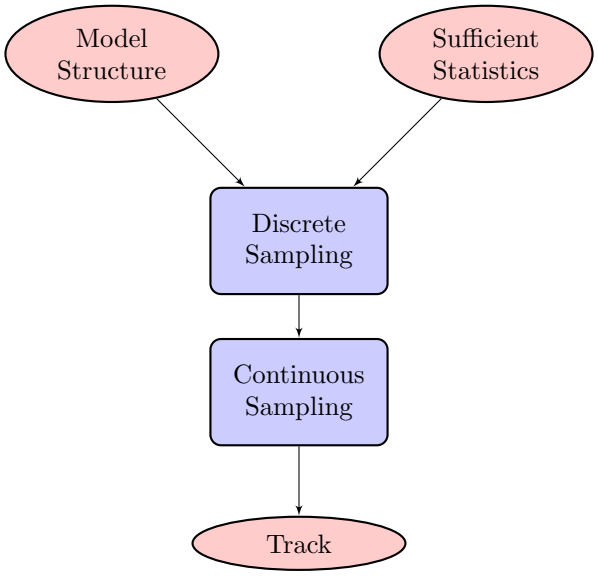


Figure 8. Sampling process flow.

IV.A. Discrete Sampling

We begin by sampling from the Bayesian network representing the initial distribution. As outlined in the appendix, we randomly assign value k to variable X_i with probability

$$\frac{\alpha_{ijk} + N_{ijk}}{\sum_{k'=1}^{r_i}(\alpha_{ijk'} + N_{ijk'})}, \tag{3}$$

where

- j is the instantiation of the parents of X_i in the Bayesian network,
- N_{ijk} is the number of times X_i was equal to k when its parents were instantiated to j in the data,
- α_{ijk} is a Dirichlet prior parameter, and
- r_i is the number of ways to instantiate the parents of X_i.

We use $\alpha_{ijk} = 1$, which corresponds to the prior assumption that all combinations of relative frequencies for k are equally probable.

Our sample from the Bayesian network tells us into which bins airspeed, vertical rate, turn rate, and acceleration fall. We then sample within the bins as discussed in Section IV.B.

Once we have the initial state of the trajectory, we can sample from our dynamic Bayesian network representing how the state changes. We fix $\dot{h}(t)$, $\dot{\psi}(t)$, and $\dot{v}(t)$, and then use the standard Bayesian network

sampling scheme to determine $\dot{h}(t+1)$, $\dot{\psi}(t+1)$, and $\dot{v}(t+1)$. The process may be repeated for as long as we wish to run the trajectory.

IV.B. Continuous Sampling

To produce a continuous sample from the discrete sample from the initial distribution, we simply sample uniformly within the bins. For example, if we determine that the initial airspeed is within the bin $[60, 90)$, i.e. the second bin according to the quantization scheme in table 1, we simply sample from the uniform distribution over the half-open interval $[60, 90)$. Because the first and last bins associated with each interval are unbounded and uniform sampling on an unbounded interval is undefined, it is necessary to impose some bounds. Table 2 shows the boundaries in our quantization scheme.

Table 2. Sampling boundaries.

	Boundaries
h	$0, 1200, 3000, 5000, 12500$
v	$0, 30, 60, 90, 120, 140, 165, 250, 300$
$\dot{v}$	$-2, -1, -0.25, 0.25, 1, 2$
$\dot{h}$	$-2000, -1250, -750, -250, 250, 750, 1250, 2000$
$\dot{\psi}$	$-8, -6, -4.5, -1.5, 1.5, 4.5, 6, 8$

To produce a continuous sample of the trajectory produced by the transition network, we iterate through the arrays of climb rates, turn rates, and accelerations. Instead of sampling within bins at every time step, we only produce a new continuous sample at some fixed rate estimated from the data. The rates we used are 0.0212, 0.0413, and 0.0885 changes per second for acceleration, vertical rate, and turn rate respectively. We estimated these rates from the data by introducing three smaller bins within each bin and computing the relative frequency that tracks stay within a single smaller bin (versus moving to another small bin within the same coarse bin). A similar strategy was used by Eurocontrol for their cooperative encounter model.[9,10]

When producing continuous samples from bins that include zero in their range, we simply produce zero instead of sampling uniformly in order to prevent very small vertical rates, turn rates, and acceleration.

Figure 9 plots an example of vertical rates, turn rates, accelerations, and airspeeds generated by sampling from the Bayesian networks. Figure 10 shows the resulting track produced by the sampled features shown in the previous figure. Translation of features into tracks involves running a discrete-time simulation, described in the next section.

V. Simulation

Earlier sections explained how to build a dynamic probabilistic model of aircraft and how to sample from the model to produce nominal trajectories that are representative of what we observed in our radar data. The same modeling techniques discussed earlier may be used to build a model of the manned or unmanned aircraft with the collision avoidance system to be evaluated. This section explains how to combine a model of the aircraft with the collision avoidance system to be evaluated, which we call AC1, with a model of an intruder, which we call AC2, into a simulated encounter.

The trajectory for AC1 may be specified by the analyst (e.g., to focus on a particular phase of flight), based on actual flight paths from mission planning or radar data, or randomly generated using a statistical model representative of that aircraft's typical flight profiles. In a study of VFR-on-VFR encounters, for example, trajectories for both AC1 and AC2 could be generated using the uncorrelated encounter model described here.

We avoid simulating encounters that are extremely unlikely to result in NMACs by focusing our computational effort on encounters that occur in an encounter cylinder centered on AC1. AC2 is initialized on the surface of the encounter cylinder and the dynamic models are used to update the states of AC1 and AC2 over time. If the collision avoidance system on AC1 suggests an avoidance maneuver, it overrides the nominal rates suggested by the dynamic model. If AC2 enters an NMAC cylinder or exits the encounter

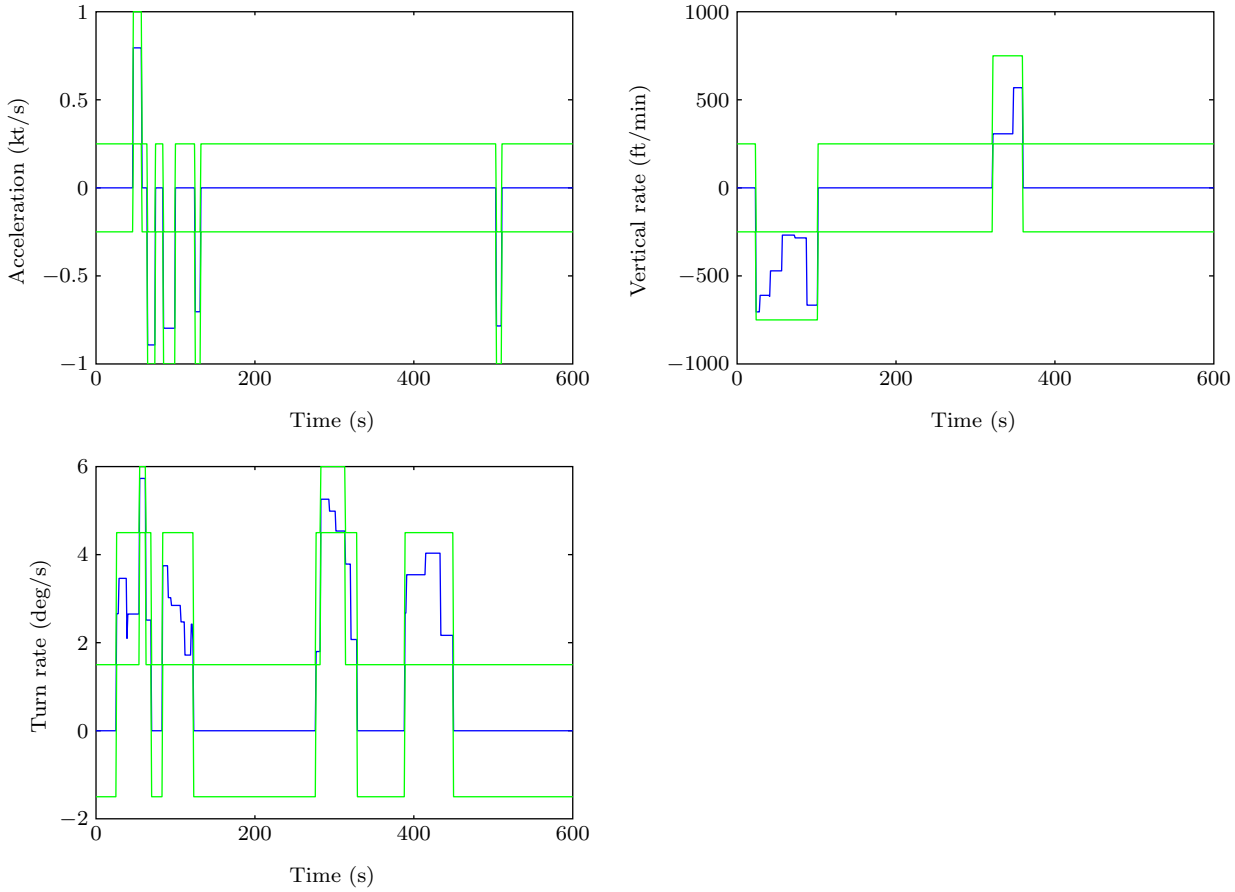


Figure 9. A plot of sampled features over time. Blue lines show continuous samples, and green blocks show the outlines of the discrete samples.

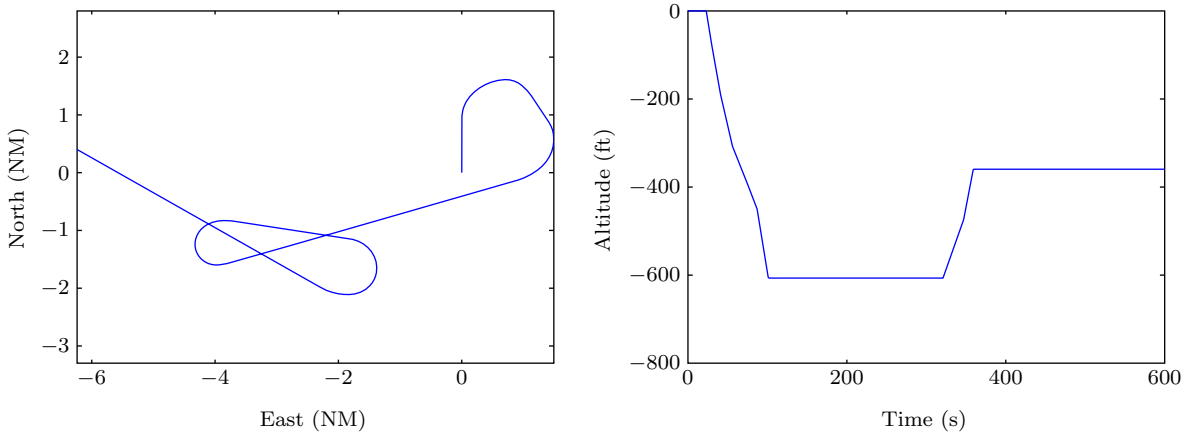


Figure 10. A track generated by sampling from the initial and transition distributions. Positions and altitudes are relative to the initial position and altitude.

cylinder, the encounter run is terminated. The NMAC cylinder has radius r_{nmac} and height $2h_{\text{nmac}}$. We use $r_{\text{nmac}} = 500\,\text{ft}$ and $h_{\text{nmac}} = 100\,\text{ft}$.

V.A. Encounter Cylinder Dimensions

The encounter cylinder has radius r_{enc} and height $2h_{\text{enc}}$. The appropriate dimensions of the encounter cylinder depend on the aircraft dynamics and collision avoidance system. If the encounter cylinder is too small, the collision avoidance system will not have enough opportunity to be fully exercised before a collision. If the encounter cylinder is too large, then computation is wasted.

An upper bound for r_{enc} is the amount of time required by the collision avoidance system to detect and track a target multiplied by the sum of the maximal airspeeds of AC1 and AC2. An upper bound for h_{enc} is the amount of time required by the collision avoidance system to detect and track a target multiplied by the sum of the maximal vertical rate magnitudes of AC1 and AC2.

V.B. Initial Conditions

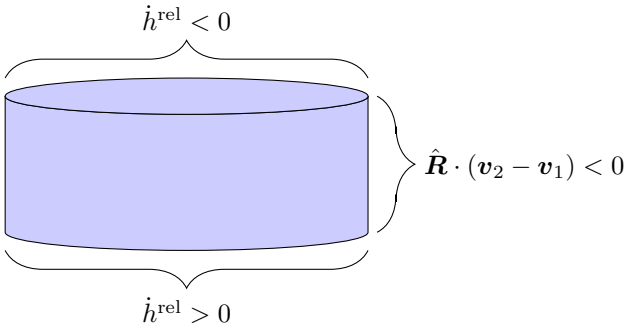


Figure 11. Depending on where on the encounter cylinder the intruder is initialized, different criteria must be met.

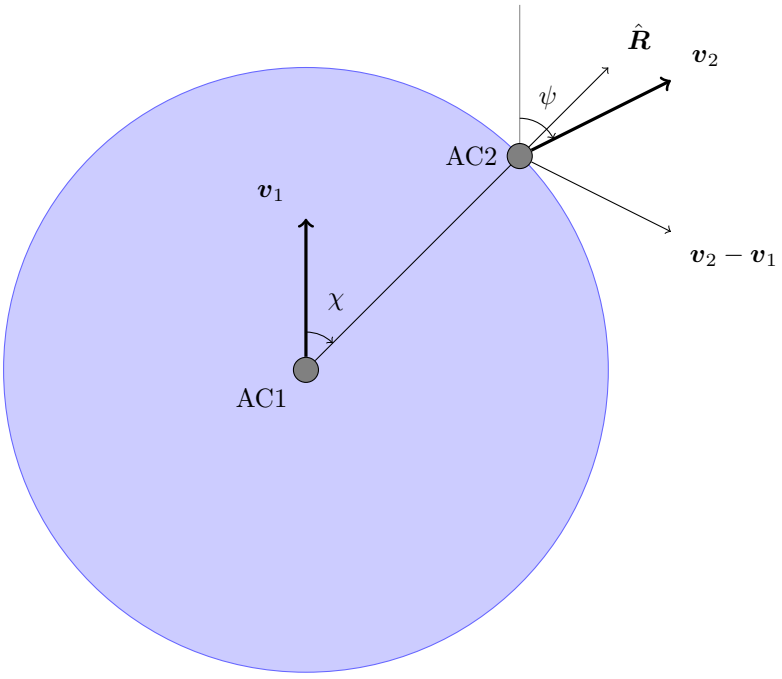


Figure 12. Variables relevant to rejection sampling in the horizontal plane.

We use rejection sampling to generate the initial conditions of an encounter. Rejection sampling involves proposing a series of candidate samples from a random distribution until choosing one that meets a set of

criteria (summarized in figure 11). The process we use for generating initial conditions for encounters is as follows:

1. Generate airspeeds, vertical rates, turn rates, and accelerations for AC1 and AC2 according to their models such that they belong to the same airspace class and altitude layer. Forcing this constraint can be done using rejection sampling. Simply generate AC1 and AC2 independently and reject both if they have a different airspace class or altitude layer.

2. Initialize the position of AC2 on the surface of the encounter cylinder centered on AC1. AC2 is not restricted to being on the side of the cylinder; it may be initialized on the top or bottom. The bearing of AC2 relative to AC1 is denoted χ.

3. The heading of AC1 is set to 0. The heading of AC2, denoted ψ, is randomly selected from a uniform distribution over $[0, 2\pi)$.

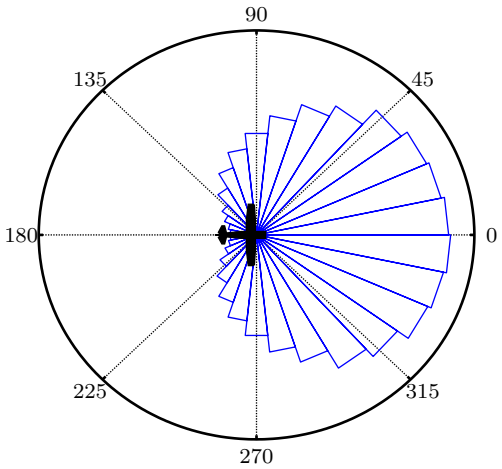
4. If AC2 was initialized on the top of the encounter cylinder, accept the sample if the vertical rate of $AC2$ relative to $AC1$, denoted $\dot{h}^{\text{rel}}$, is negative. This ensures that AC2 is penetrating the encounter cylinder for the first time.

5. If AC2 was initialized on the bottom of the encounter cylinder, accept the sample if the vertical rate of $AC2$ relative to $AC1$, denoted $\dot{h}^{\text{rel}}$, is positive. This ensures that AC2 is penetrating the encounter cylinder for the first time.

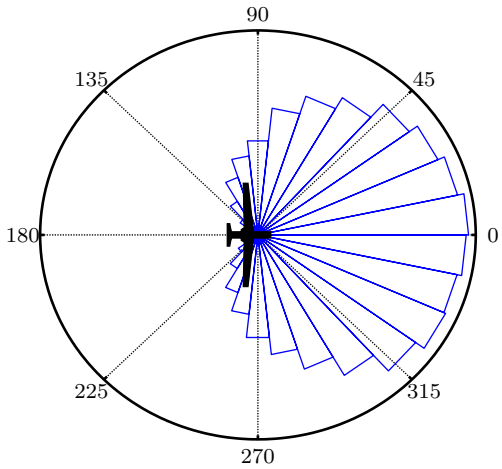
6. If AC2 was initialized on the side of the encounter cylinder, accept the sample if $\hat{\boldsymbol{R}} \cdot (\boldsymbol{v}_2 - \boldsymbol{v}_1)$ is negative, where $\hat{\boldsymbol{R}} = (\sin\chi, \cos\chi)$ is the unit vector from AC1 to AC2. The vectors $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ are the ground velocities of AC1 and AC2 respectively. When $\hat{\boldsymbol{R}} \cdot (\boldsymbol{v}_2 - \boldsymbol{v}_1)$ is negative, the relative velocity of AC2 is into the encounter cylinder, and therefore should be accepted. See figure 12.

The process is repeated until a candidate initialization is accepted.

A byproduct of rejection sampling is that the bearing distribution is nonuniform. As one would expect, more encounters occur head on than from the side. Figure 13 shows bearing distributions of VFR/VFR encounters and Global Hawk/VFR encounters. The VFR trajectories were generated from our encounter model, and the Global Hawk trajectories were based on radar data from a Global Hawk flight from Beale Air Force Base on October 4, 2007. Because Global Hawk flies faster than the average VFR aircraft, Global Hawk typically encounters aircraft that are closing from ahead. As the bearing distributions in figure 13 show, it is very rare for intruders to come from behind.



(a) Bearing distribution of 604,637 intruders initialized on the side of the encounter cylinder for the VFR pair encounter set.



(b) Bearing distribution of 177,702 intruders initialized on the side of the encounter cylinder for the Global Hawk/VFR encounter set.

Figure 13. Bearing distributions produced by rejection sampling.

Once the initial conditions are selected, the dynamic models of AC1 and AC2 are used to update their trajectories over time. A simulation run terminates when the intruder either exits the encounter cylinder or penetrates the NMAC cylinder.

V.C. Multiple Encounters

Our simulation currently only handles pairwise encounters. The probability of an intruder penetrating the encounter cylinder while another intruder is within the encounter cylinder is likely to be very small. One may compute this probability using

$$\int_0^\infty p(t)[1 - e^{\lambda_{\text{enc}} t}]\mathrm{d}t = 1 - \int_0^\infty p(t) e^{\lambda_{\text{enc}} t}\mathrm{d}t\,, \tag{4}$$

where $p(t)$ is the distribution over the amount of time intruders spend in the encounter cylinder and λ_{enc} is the rate at which new intruders penetrate the encounter cylinder. In the above equation, $1-e^{\lambda_{\text{enc}} t}$ comes from the cumulative density function for an exponential distribution. The possibility of simultaneous multiple intruders needs to be examined and will be an area of future work.

VI. Safety Evaluation

This section explains how to estimate the NMAC rate, denoted λ_{nmac}, based on a large number of simulations. We begin by observing that

$$\lambda_{\text{nmac}} = P(\text{nmac} \mid \text{enc})\lambda_{\text{enc}}\,,$$

where $P(\text{nmac} \mid \text{enc})$ is the probability that an aircraft that enters the encounter cylinder penetrates the NMAC cylinder before exiting the encounter cylinder and λ_{enc} is the rate at which aircraft penetrate the encounter cylinder. The mean time between NMACs is simply $\lambda_{\text{nmac}}^{-1}$.

This section makes the following assumptions:

1. the density of air traffic outside the encounter cylinder is uniform, and
2. the trajectories of aircraft outside of the encounter cylinder are independent of the trajectories of aircraft within the encounter cylinder,

From these two assumptions, we explain how to compute $P(\text{nmac} \mid \text{enc})$ and λ_{enc}.

VI.A. Estimating NMAC Probability

Section V.B explained how to construct an encounter from two independent trajectories sampled from the distribution represented by our Bayesian networks. By generating a large collection of encounters and determining which encounters lead to NMACs, we can estimate $P(\text{nmac} \mid \text{enc})$. Unfortunately, we cannot simply divide the number of sampled encounters that lead to NMACs by the total number of sampled encounters to estimate $P(\text{nmac} \mid \text{enc})$ due to the fact that our sampling scheme does not produce encounters from the same distribution that would occur in the airspace. In particular, we generate encounters with aircraft velocities distributed identically to the aircraft population at large, despite the fact that in reality the distribution of aircraft velocities given that an encounter is occurring favors high-speed aircraft. Although we sample from a distribution that is different from the true distribution when constructing encounters, we can still use the samples to estimate $P(\text{nmac} \mid \text{enc})$ so long as we weight their results properly using an approach known as importance sampling.[18] We will begin by stating the weighting scheme and then prove that it is correct.

Section IV explained how to generate the trajectories for AC1 and AC2, which we call $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$, by sampling from the Bayesian networks with the requirement that both aircraft come from the same airspace class and altitude layer. Section V.B explained how to randomly select the position of AC2 relative to AC1, which we call $\boldsymbol{x}_r$, and the heading of AC2 relative to AC1, which we call ψ. Importance sampling allows us to make the following approximation based on N samples

$$P(\text{nmac} \mid \text{enc}) \approx \frac{1}{N}\sum_i P(\text{nmac} \mid \boldsymbol{z}_1^{(i)}, \boldsymbol{z}_2^{(i)}, \psi^{(i)}, \boldsymbol{x}_r^{(i)}, \text{enc}) \frac{V(\boldsymbol{z}_1^{(i)}, \boldsymbol{z}_2^{(i)})}{\bar{V}}\,.$$

The weight $V(\boldsymbol{z}_1^{(i)}, \boldsymbol{z}_2^{(i)})/\bar{V}$ corrects for the fact that our sampling distribution does not match the true distribution of encounter situations. The function $V(\boldsymbol{z}_1^{(i)}, \boldsymbol{z}_2^{(i)})$ is the average volume the encounter cylinder

sweeps out per unit time when AC1 follows $\boldsymbol{z}_1^{(i)}$ and the airspace consists exclusively of aircraft following $\boldsymbol{z}_2^{(i)}$. In particular,

$$V(\boldsymbol{z}_1^{(i)}, \boldsymbol{z}_2^{(i)}) = 4 r_{\text{enc}} h_{\text{enc}} \underbrace{\frac{1}{\pi} \int_0^{\pi} \sqrt{(v_1^g + v_2^g \cos\psi)^2 + (v_2^g \sin\psi)^2}\, \mathrm{d}\psi}_{\substack{\text{average} \\ \text{relative} \\ \text{horizontal} \\ \text{speed}}} + \pi r_{\text{enc}}^2 \underbrace{\left| \dot{h}_1 - \dot{h}_2 \right|}_{\substack{\text{average} \\ \text{relative} \\ \text{vertical} \\ \text{speed}}},$$

where v_1^g and v_2^g are the initial ground speeds and $\dot{h}_1$ and $\dot{h}_2$ are the initial vertical rates of $\boldsymbol{z}_1^{(i)}$ and $\boldsymbol{z}_2^{(i)}$. The constant $\bar{V}$ is the average volume the encounter cylinder sweeps out per unit time

$$\bar{V} = \iint p(\boldsymbol{z}_2) p(\boldsymbol{z}_2 \mid \boldsymbol{z}_1) V(\boldsymbol{z}_1, \boldsymbol{z}_2)\, \mathrm{d}\boldsymbol{z}_1 \mathrm{d}\boldsymbol{z}_2 \,.$$

Note that the distribution over $\boldsymbol{z}_2$ is conditional on $\boldsymbol{z}_1$ due to the constraint that AC1 and AC2 must belong to the same airspace class and altitude layer. The constant $\bar{V}$ can be estimated through sampling.

Now that we have defined our weighting scheme, we will now prove that it is correct. From the laws of probability,

$$\begin{aligned} P(\text{nmac} \mid \text{enc}) &= \iint P(\text{nmac} \mid \boldsymbol{z}_1, \boldsymbol{z}_2, \text{enc}) p(\boldsymbol{z}_1, \boldsymbol{z}_2 \mid \text{enc})\, \mathrm{d}\boldsymbol{z}_1 \mathrm{d}\boldsymbol{z}_2 \\ &= \iiiint P(\text{nmac} \mid \boldsymbol{z}_1, \boldsymbol{z}_2, \psi, \boldsymbol{x}_r, \text{enc}) p(\psi, \boldsymbol{x}_r \mid \boldsymbol{z}_1, \boldsymbol{z}_2, \text{enc}) p(\boldsymbol{z}_1, \boldsymbol{z}_2 \mid \text{enc})\, \mathrm{d}\boldsymbol{z}_1 \mathrm{d}\boldsymbol{z}_2 \mathrm{d}\psi \mathrm{d}\boldsymbol{x}_r \,. \end{aligned}$$

We may approximate $P(\text{nmac} \mid \text{enc})$ using Monte Carlo sampling. Since it is difficult to sample from $p(\boldsymbol{z}_1, \boldsymbol{z}_2 \mid \text{enc})$ directly, we sample $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$ from the distribution represented by our Bayesian network subject to the constraint that both aircraft come from the same airspace class and altitude layer, and weight the samples appropriately:

$$P(\text{nmac} \mid \text{enc}) \approx \frac{1}{N} \sum_i P(\text{nmac} \mid \boldsymbol{z}_1^{(i)}, \boldsymbol{z}_2^{(i)}, \psi^{(i)}, \boldsymbol{x}_r^{(i)}, \text{enc}) \frac{p(\boldsymbol{z}_1^{(i)}, \boldsymbol{z}_2^{(i)} \mid \text{enc})}{p(\boldsymbol{z}_1^{(i)}) p(\boldsymbol{z}_2^{(i)} \mid \boldsymbol{z}_1^{(i)})} \,.$$

We know that

$$\begin{aligned} p(\boldsymbol{z}_1^{(i)}, \boldsymbol{z}_2^{(i)} \mid \text{enc}) &= \frac{p(\boldsymbol{z}_1^{(i)}) p(\boldsymbol{z}_2^{(i)} \mid \boldsymbol{z}_1^{(i)})}{\lambda_{\text{enc}}} \lambda_{\text{enc} \mid \boldsymbol{z}_1^{(i)}, \boldsymbol{z}_2^{(i)}} \\ &\propto p(\boldsymbol{z}_1^{(i)}) p(\boldsymbol{z}_2^{(i)} \mid \boldsymbol{z}_1^{(i)}) \lambda_{\text{enc} \mid \boldsymbol{z}_1^{(i)}, \boldsymbol{z}_2^{(i)}} \\ &\propto p(\boldsymbol{z}_1^{(i)}) p(\boldsymbol{z}_2^{(i)} \mid \boldsymbol{z}_1^{(i)}) V(\boldsymbol{z}_1^{(i)}, \boldsymbol{z}_2^{(i)}) \,. \end{aligned}$$

We may normalize to obtain

$$\begin{aligned} p(\boldsymbol{z}_1^{(i)}, \boldsymbol{z}_2^{(i)} \mid \text{enc}) &= p(\boldsymbol{z}_1^{(i)}) p(\boldsymbol{z}_2^{(i)} \mid \boldsymbol{z}_1^{(i)}) V(\boldsymbol{z}_1^{(i)}, \boldsymbol{z}_2^{(i)}) / \iint p(\boldsymbol{z}_1) p(\boldsymbol{z}_2 \mid \boldsymbol{z}_1) V(\boldsymbol{z}_1, \boldsymbol{z}_2)\, \mathrm{d}\boldsymbol{z}_1 \mathrm{d}\boldsymbol{z}_2 \\ &= p(\boldsymbol{z}_1^{(i)}) p(\boldsymbol{z}_2^{(i)} \mid \boldsymbol{z}_1^{(i)}) V(\boldsymbol{z}_1^{(i)}, \boldsymbol{z}_2^{(i)}) / \bar{V} \,. \end{aligned}$$

We may substitute and simplify to obtain

$$\begin{aligned} P(\text{nmac} \mid \text{enc}) &\approx \frac{1}{N} \sum_i P(\text{nmac} \mid \boldsymbol{z}_1^{(i)}, \boldsymbol{z}_2^{(i)}, \psi^{(i)}, \boldsymbol{x}_r^{(i)}, \text{enc}) \frac{p(\boldsymbol{z}_1^{(i)}) p(\boldsymbol{z}_2^{(i)} \mid \boldsymbol{z}_1^{(i)}) V(\boldsymbol{z}_1^{(i)}, \boldsymbol{z}_2^{(i)}) / \bar{V}}{p(\boldsymbol{z}_1^{(i)}) p(\boldsymbol{z}_2^{(i)} \mid \boldsymbol{z}_1^{(i)})} \\ &\approx \frac{1}{N} \sum_i P(\text{nmac} \mid \boldsymbol{z}_1^{(i)}, \boldsymbol{z}_2^{(i)}, \psi^{(i)}, \boldsymbol{x}_r^{(i)}, \text{enc}) \frac{V(\boldsymbol{z}_1^{(i)}, \boldsymbol{z}_2^{(i)})}{\bar{V}} \,, \end{aligned}$$

which corresponds to the weighting scheme we defined.

VI.B. Estimating Encounter Rate

Estimating the encounter rate requires knowing the density of traffic outside the encounter cylinder. This density, ρ, has units $(\text{NM}^2 \times \text{ft})^{-1}$. The rate at which aircraft enter the encounter cylinder is the product of ρ and the average volume of new airspace the encounter cylinder sweeps through per unit time, $\bar{V}$, which was discussed in Section VI.A:

$$\lambda_{\text{enc}} = \rho \bar{V} \, .$$

Hence, we may compute λ_{nmac} as follows:

$$\begin{aligned}
\lambda_{\text{nmac}} &= P(\text{nmac} \mid \text{enc}) \lambda_{\text{enc}} \\
&= \rho \bar{V} P(\text{nmac} \mid \text{enc}) \\
&\approx \frac{\rho \bar{V}}{N} \sum_i P(\text{nmac} \mid \boldsymbol{z}_1^{(i)}, \boldsymbol{z}_2^{(i)}, \psi^{(i)}, \boldsymbol{x}_r^{(i)}, \text{enc}) \frac{V(\boldsymbol{z}_1^{(i)}, \boldsymbol{z}_2^{(i)})}{\bar{V}} \\
&= \frac{\rho}{N} \sum_i P(\text{nmac} \mid \boldsymbol{z}_1^{(i)}, \boldsymbol{z}_2^{(i)}, \psi^{(i)}, \boldsymbol{x}_r^{(i)}, \text{enc}) V(\boldsymbol{z}_1^{(i)}, \boldsymbol{z}_2^{(i)}) \, .
\end{aligned}$$

VII. Applications

Our encounter models are useful for a wide variety of analyses for both manned and unmanned aircraft. A few applications include:

TCAS safety assessment: Encounter models have been developed since the early 1980s by various organizations for the safety evaluation of TCAS.[3,7–10] Millions of encounters were generated to test TCAS in a wide variety of situations. The results of these simulation studies played an important role in supporting a worldwide mandate for TCAS equipage for transport aircraft. Over the years, various changes have been proposed for TCAS, requiring further safety analysis. Changes to the national airspace structure and traffic densities will also require further analysis of TCAS behavior.

Evaluation of sense and avoid algorithms: The FAA requires that all unmanned aircraft flying in civil airspace be capable of sensing and avoiding other aircraft. Several sense and avoid algorithms have been developed, but the FAA and other civil aviation organizations have yet to certify such algorithms. Monte Carlo safety assessment studies will play an important role in certification as they have in the past with TCAS.

Sensor trade studies: Before unmanned aircraft are permitted to fly routinely in civil airspace, sensor performance standards need to be established. Lincoln Laboratory has used the encounter model described in this paper to provide a parametric analysis of the sense and avoid capability of an electro-optical system. This includes an assessment of the exchange between sensor field-of-view shape and detection range with the probability of intruder detection prior to near miss.[19]

Hazard alerting design: It has been proposed that line-of-sight rates, as measured by an inexpensive electro-optical sensor, can be used to predict whether an intruder aircraft is likely to be a hazard.[20] Lincoln Laboratory has used an encounter model to demonstrate that simply alerting whenever line-of-sight rate falls below a certain threshold leads to frequent false alerts. A more sophisticated approach using partially observable Markov decision processes (POMDPs) leads to a significant decrease in false alerts.[21]

Advanced collision avoidance algorithm development: We are collaborating with the Computer Science and Artificial Intelligence Laboratory (CSAIL) at MIT to develop advanced collision avoidance algorithms using POMDPs. The encounter model is used to estimate the parameters of the state transition model. Given the state transition model and a sensor model, one may solve for an approximately optimal policy that robustly balances aircraft separation, flight plan deviation, and maneuver coordination.

VIII. Conclusions and Further Work

This paper has presented a new approach to encounter modeling that allows for the generation of more realistic encounters than previous models. The approach involves modeling the dynamics of aircraft state based on a Markov model where the probability of the next state depends only upon the current state. One way to represent a Markov model is with an exhaustive state-transition matrix that specifies the probability of transitioning between all pairs of states. Unfortunately, the number of independent parameters required to define the matrix grows super exponentially with the number of variables defining the model. The more independent parameters there are in the model, the more data one needs to properly estimate their values. However, by using dynamic Bayesian networks, we can leverage conditional independence between some variables to greatly reduce the number of parameters. We can learn the structure of the dynamic Bayesian network by maximizing the posterior probability of the network structure given the data.

The work presented in this paper has focused on a model where the trajectories of the aircraft involved in the encounter are independent of each other prior to intervention by a collision avoidance system, human or automated. This model assumes that aircraft blunder into close proximity without prior intervention. We are currently developing a separate model for aircraft under air traffic control, where intervention may impact the way in which aircraft encounter each other. Such a correlated encounter model is similar to the uncorrelated encounter model described here except that the variables defining the state of the aircraft in the encounter are tied to each other in the dynamic Bayesian network.

The uncorrelated encounter model presented in this paper along with a correlated encounter model under development will be used to generate encounter situations for use in Monte Carlo safety analyses of collision avoidance systems for manned and unmanned aircraft. The results of these robustness studies will inform the development and certification of new systems.

Appendix

This appendix briefly reviews Bayesian networks. Further discussion of Bayesian networks may be found elsewhere.[22–24]

VIII.A. Definition

A Bayesian network is a graphical representation of a multivariate probability distribution over variables $\boldsymbol{X} = X_1, \ldots, X_n$. In particular, a Bayesian network is a directed acyclic graph G whose nodes correspond to variables and edges correspond to probabilistic dependencies between them. Associated with each variable X_i is a conditional probability distribution $P(x_i \mid \boldsymbol{\pi}_i)$, where $\boldsymbol{\pi}_i$ denotes an instantiation of the parents of X_i in the graph. The probability of an instantiation of the variables is specified directly by the conditional probability distributions in the Bayesian network:

$$P(\boldsymbol{x}) = P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i \mid \boldsymbol{\pi}_i)\,. \tag{5}$$

VIII.B. Sampling

It is rather straightforward to sample from a multivariate distribution represented by a Bayesian network. The first step is to produce a topological sort of the nodes in the network. A topological sort orders the nodes in a Bayesian network such that if a node X_i comes before X_j there does not exist a directed path from X_j to X_i. Every Bayesian network has at least one topological sort, but there may be many. Efficient algorithms exist for finding a valid topological sort.[25]

To produce a sample from the joint distribution represented by a Bayesian network, we simply iterate through a topologically sorted sequence of the variables and sample from their conditional probability distributions. The topological sort ensures that when sampling from each conditional probability distribution the necessary parents have been instantiated.

VIII.C. Parameter Learning

The parameters $\boldsymbol{\theta}$ of a Bayesian network determine the associated conditional probability distributions. Given some fixed network structure G, we can learn these parameters from data. In this appendix, we

assume that the variables are discrete.

Before discussing how to learn the parameters of a Bayesian network, it is necessary to introduce some notation. Let r_i represent the number of instantiations of X_i and q_i represent the number of instantiations of the parents of X_i. If X_i has no parents, then $q_i = 1$. The jth instantiation of the parents of X_i is denoted $\boldsymbol{\pi}_{ij}$.

There are $\sum_{i=1}^{n} r_i q_i$ parameters in a Bayesian network. Each parameter is written θ_{ijk} and determines $P(X_i = k \mid \boldsymbol{\pi}_{ij})$, i.e.,

$$P(X_i = k \mid \boldsymbol{\pi}_{ij}) = \theta_{ijk}\,.$$

Although there are $\sum_{i=1}^{n} r_i q_i$ parameters, only $\sum_{i=1}^{n}(r_i - 1)q_i$ are independent.

Computing the posterior $p(\boldsymbol{\theta} \mid D, G)$ involves specifying a prior $p(\boldsymbol{\theta} \mid G)$ and applying Bayes' rule

$$p(\boldsymbol{\theta} \mid D, G) = \frac{P(D \mid \boldsymbol{\theta}, G)p(\boldsymbol{\theta} \mid G)}{P(D \mid G)} = \frac{P(D \mid \boldsymbol{\theta}, G)p(\boldsymbol{\theta} \mid G)}{\int P(D \mid \boldsymbol{\theta}, G)p(\boldsymbol{\theta} \mid G)\,\mathrm{d}\boldsymbol{\theta}}\,. \tag{6}$$

If N_{ijk} is the count of $X_i = k$ given $\boldsymbol{\pi}_{ij}$ in the data D, then the probability of the data given the parameters $\boldsymbol{\theta}$ is

$$P(D \mid \boldsymbol{\theta}) = \prod_{i=1}^{n}\prod_{j=1}^{q_i}\prod_{k=1}^{r_i}\theta_{ijk}^{N_{ijk}}\,. \tag{7}$$

Let $\boldsymbol{\theta}_{ij} = (\theta_{ij1}, \ldots, \theta_{ijr_i})$. Since $\boldsymbol{\theta}_{ij}$ is independent of $\boldsymbol{\theta}_{i'j'}$ when $ij \neq i'j'$, the prior probability of the parameters assuming a fixed structure G is

$$p(\boldsymbol{\theta} \mid G) = \prod_{i=1}^{n}\prod_{j=1}^{q_i} p(\boldsymbol{\theta}_{ij} \mid G)\,. \tag{8}$$

The density $p(\boldsymbol{\theta}_{ij} \mid G)$ is a distribution over relative frequencies. Under some very weak assumptions, it is possible to prove that $p(\boldsymbol{\theta}_{ij} \mid G)$ is Dirichlet.[24] Hence,

$$p(\boldsymbol{\theta}_{ij} \mid G) = \begin{cases} \frac{\Gamma(\alpha_{ij0})}{\prod_{k=1}^{r}\Gamma(\alpha_{ijk})}\prod_{k=1}^{r_i}\theta_{ijk}^{\alpha_{ijk}-1} & \text{if } 0 \leq \theta_{ijk} \leq 1 \text{ and } \sum_{k=1}^{r_i}\theta_{ijk} = 1 \\ 0 & \text{otherwise} \end{cases},$$

where $\alpha_{ij1}, \ldots, \alpha_{ijr_i}$ are the parameters of the Dirichlet distribution and $\alpha_{ij0} = \sum_{k=1}^{r_i}\alpha_{ijk}$. For the prior to be objective (or noninformative), the parameters α_{ijk} must be identical for all k. Different objective priors have been used in the literature. Cooper and Herskovits[26] use $\alpha_{ijk} = 1$. Heckerman, Geiger, and Chickering[27] use and justify $\alpha_{ijk} = 1/(r_i q_i)$.

It is possible to show that $p(\boldsymbol{\theta}_{ij} \mid D, G)$ is Dirichlet with parameters $\alpha_{ijk} + N_{ijk}, \ldots, \alpha_{ijk} + N_{ijk}$. Hence,

$$p(\boldsymbol{\theta}_{ij} \mid D, G) = \begin{cases} \frac{\Gamma(\alpha_{ij0}+N_{ij})}{\prod_{k=1}^{r}\Gamma(\alpha_{ijk}+N_{ijk})}\prod_{k=1}^{r_i}\theta_{ijk}^{\alpha_{ijk}+N_{ijk}-1} & \text{if } 0 \leq \theta_{ijk} \leq 1 \text{ and } \sum_{k=1}^{r_i}\theta_{ijk} = 1 \\ 0 & \text{otherwise} \end{cases},$$

where $N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$.

Sampling from a Bayesian network with G known, $\boldsymbol{\theta}$ unknown, and D observed involves assigning k to X_i with probability

$$P(X_i = k \mid D, G) = \int \theta_{ijk}\,p(\theta_{ijk} \mid D, G)\,\mathrm{d}\theta_{ijk} = \frac{\alpha_{ijk} + N_{ijk}}{\sum_{k'=1}^{r_i}(\alpha_{ijk'} + N_{ijk'})}\,. \tag{9}$$

VIII.D. Structure Learning

Finding the most likely structure G that generated a set of data D. The objective is to find the most likely graph given data. By Bayes' rule,

$$P(G \mid D) \propto P(G)P(D \mid G) = P(G)\int P(D \mid \boldsymbol{\theta}, G)p(\boldsymbol{\theta} \mid G)\,\mathrm{d}\boldsymbol{\theta}\,. \tag{10}$$

The previous section explains how to compute the likelihood $P(D \mid \boldsymbol{\theta}, G)$ and the prior $p(\boldsymbol{\theta} \mid G)$. Cooper and Herskovits[26] show how to evaluate the integral above, resulting in

$$P(G \mid D) = P(G) \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(\alpha_{ij0})}{\Gamma(\alpha_{ij0} + N_{ij})} \prod_{k=1}^{r_i} \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})}, \tag{11}$$

where $N_{ij} = \sum_{k=1}^{r} N_{ijk}$. Heckerman, Geiger, and Chickering[27] suggest priors over graphs, but it is not uncommon in the literature to assume a uniform prior. For numerical convenience, most Bayesian network learning packages calculate and report $\log P(G \mid D) + K$, where K is a constant independent of G. This quantity is often called the *Bayesian score* and may be used for structure comparison and search.

Acknowledgements

This report is the result of research and development sponsored by the United States Department of Homeland Security Science and Technology Directorate, the United States Air Force 303rd Aeronautical Systems Wing (303 AESW/XRX), the Department of Defense Unmanned Aircraft System Airspace Integration Joint Integrated Product Team, and the Federal Aviation Administration.

The authors greatly appreciate the support and assistance provided by Kevin "Spanky" Kirsch, John Appleby, Merv Leavitt, Joseph Kielman, and David Thomasson from the Department of Homeland Security, along with Maj. Luke Cropsey, Lt. Col. Kent Tiffany, and Tony Salmonson from the Air Force and Paul Fontaine and Neal Suchy from the FAA. The authors appreciate the support from Hill Air Force Base, including Jeff Richardson, Steven H. Schimmelpfennig, Richard C. Whitlock, Lt. Tanuxay Keooudom, Lt. Han Saydam, TSgt. Christopher B. Cosper, and James Evans. Finally, the authors would also like to thank Vito D. Cavallo, Ann C. Drumm, Jeffrey Gertz, Robert D. Grappel, Garrett S. Harris, Jessica E. Olszta, John O'Rourke, Angela Sharer, Steven D. Thompson, and Richard Williams for their contributions to the airspace encounter model effort.

References

[1]MITRE, "System Safety Study of Minimum TCAS II," Tech. Rep. MTR-83W241, MITRE, 1983.

[2]Drumm, A., "Lincoln Laboratory Evaluation of TCAS II Logic Version 6.04a," Tech. Rep. MIT-LIN-ATC-240, MIT Lincoln Laboratory, 1996.

[3]McLaughlin, M. P., "Safety Study of the Traffic Alert and Collision Avoidance System (TCAS II)," Tech. Rep. MTR 97W32, MITRE Corporation, June 1997.

[4]Chludzinski, B., "Lincoln Laboratory Evaluation of TCAS II Logic Version 7," Tech. Rep. MIT-LIN-ATC-268, MIT Lincoln Laboratory, 1999.

[5]Arino, T., Carpenter, K., Chabert, S., Hutchinson, H., Miquel, T., Raynaud, B., Rigotti, K., and Vallauri, E., "Studies on the Safety of ACAS II in Europe," Tech. Rep. ACASA/WP-1.8/210D, Eurocontrol, 2002.

[6]ICAO, "ACAS Manual," Tech. Rep. SCRSP/1-WP/53, Eurocontrol, 2004.

[7]Choyce, T. A. and Ciaramella, K. M., "Test and Evaluation of TCAS II Logic Version 7," Tech. rep., Federal Aviation Administration, Oct. 2000.

[8]ICAO, "Surveillance, Radar and Collision Avoidance," *ICAO Standards and Recommended Practices*, Vol. IV, annex 10, 1998.

[9]Miquel, T. and Rigotti, K., "European Encounter Model," Tech. Rep. ACASA/WP1/186/D, CENA/Sofréavia and QinetiQ, March 2002.

[10]Chabert, S., "Safety Encounter Model Focused on Issue SA01a," Tech. Rep. SIRE/WP2/21/D, CENA/Sofréavia and QinetiQ, April 2005.

[11]Asmat, J., Rhodes, B., Umansky, J., Villavicencio, C., Yunas, A., Donohue, G., and Lacher, A., "UAS Safety: Unmanned Aerial Collision Avoidance System (UCAS)," *Proceedings of the System and Information Engineering Design Symposium*, edited by M. D. DeVore, 2006, pp. 43–49.

[12]Dean, T. and Kanazawa, K., "A model for reasoning about persistence and causation," *Computational Intelligence*, Vol. 5, No. 3, Aug. 1989, pp. 142–150.

[13]Murphy, K., *Dynamic Bayesian Networks: Representation, Inference and Learning*, Ph.D. thesis, University of California, Berkeley, 2002.

[14]Grappel, R. D., "ASR-9 Processor Augmentation Card (9-PAC) Phase II Scan-Scan Correlator Algorithms," Tech. Rep. MIT-LIN-ATC-298, MIT Lincoln Laboratory, April 2001.

[15]Gertz, J. L., "Mode S Surveillance Netting," Tech. Rep. MIT-LIN-ATC-120, MIT Lincoln Laboratory, Nov. 1983.

[16]Fritsch, F. N. and Carlson, R. E., "Monotone Piecewise Cubic Interpolation," *SIAM Journal of Numerical Analysis*, Vol. 17, No. 2, April 1980, pp. 238–246.

[17]Shank, E. M., "A Coordinate Conversion Algorithm for Multisensor Data Processing," Tech. Rep. MIT-LIN-ATC-139, MIT Lincoln Laboratory, Aug. 1986.

[18]Srinivasan, R., *Importance Sampling*, Springer, 2002.

[19]Griffith, J. D., Kochenderfer, M. J., and Kuchar, J. K., "Electro-Optical System Analysis for Sense and Avoid," *Proceedings of the AIAA Guidance, Navigation and Control Conference*, 2008.

[20]McCalmont, J., Utt, J., Deschenes, M., and Taylor, L. M., "Sense and Avoid, Phase I (Man-in-the-Loop) Advanced Technology Demonstration," *Infotech@Aerospace*, 2005, Arlington, VA, September 2005.

[21]Kochenderfer, M. J., Griffith, J. D., and Kuchar, J. K., "Electro-Optical Hazard Alerting Using Line-of-Sight Rate," *Proceedings of the AIAA Guidance, Navigation and Control Conference*, 2008.

[22]Pearl, J., *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann, San Francisco, CA, 1988.

[23]Jensen, F. V., *Bayesian Networks and Decision Graphs*, Springer Verlag, 2001.

[24]Neapolitan, R. E., *Learning Bayesian Networks*, Prentice Hall, Upper Saddle River, NJ, 2004.

[25]Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C., *Introduction to Algorithms*, MIT Press, 2nd ed., 2001.

[26]Cooper, G. F. and Herskovits, E., "A Bayesian Method for the Induction of Probabilistic Networks from Data," *Machine Learning*, Vol. 9, No. 4, Oct. 1992, pp. 309–347.

[27]Heckerman, D., Geiger, D., and Chickering, D. M., "Learning Bayesian Networks: The Combination of Knowledge and Statistical Data," *Machine Learning*, Vol. 20, No. 3, Sept. 1995, pp. 197–243.