

**Project Report
CASSATT-1**

**Preliminary Uncorrelated Encounter
Model of the National Airspace System**

**M.J. Kochenderfer
J.K. Kuchar
L.P. Espindle
J.L. Gertz**

26 June 2008

MIT LINCOLN LABORATORY

0031



G18D2625987

Lincoln Laboratory
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LEXINGTON, MASSACHUSETTS



Prepared for the Department of the Air Force under Contract FA8721-05-C-0002.

Approved for public release; distribution is unlimited.

20080703 096

This report is based on studies performed at Lincoln Laboratory, a center for research operated by Massachusetts Institute of Technology. This work was sponsored by the Department of the Air Force under Contract FA8721-05-C-0002.

This report may be reproduced to satisfy needs of U.S. Government agencies.

The ESC Public Affairs Office has reviewed this report, and it is releasable to the National Technical Information Service, where it will be available to the general public, including foreign nationals.

This technical report has been reviewed and is approved for publication.

FOR THE COMMANDER


Gary Tutangian
Administrative Contracting Officer
Plans and Programs Directorate
Contracted Support Management

Non-Lincoln Recipients

PLEASE DO NOT RETURN

Permission is given to destroy this document
when it is no longer needed.

Massachusetts Institute of Technology
Lincoln Laboratory

Preliminary Uncorrelated Encounter Model
of the National Airspace System

*M.J. Kochenderfer
J.K. Kuchar
L.P. Espindle
J.L. Gertz
Group 42*

Project Report CASSATT-1

26 June 2008

Approved for public release; distribution is unlimited.

Lexington

Massachusetts

ABSTRACT

Airspace encounter models, covering close encounter situations that may occur after standard separation assurance has been lost, are a critical component in the safety assessment of aviation procedures and collision avoidance systems. Of particular relevance to Unmanned Aircraft Systems is the potential for encountering general aviation aircraft that are flying under Visual Flight Rules (VFR) and which may not be in contact with air traffic control. In response to the need to develop a model of these types of encounters, Lincoln Laboratory undertook an extensive radar data collection and modeling effort. This report describes the structure and content of that encounter model. Bayesian networks are used to represent relationships between dynamic variables and to construct random aircraft trajectories that are statistically similar to those observed in the radar data.

The model described in this report is one of three that are currently under development by Lincoln Laboratory. An encounter with an intruder that does not have a transponder, or between two aircraft using a Mode A code of 1200 (VFR), is uncorrelated in the sense that it is unlikely that there would be prior intervention by air traffic control. The uncorrelated model described in this report is based on transponder-equipped aircraft using a 1200 (VFR) Mode A code observed by radars across the U.S. In addition to representing VFR-on-VFR encounters, this model is representative of encounters between a cooperative aircraft and conventional non-cooperative aircraft similar to those that use a VFR transponder code. A second uncorrelated model is also being developed based on true primary-only (non-cooperative) aircraft tracks, though the quality of such tracks is inferior to that available from the cooperative 1200-code tracks used here. Finally, a correlated encounter model is also under development to represent situations in which it is likely that there would be air traffic control intervention prior to a close encounter. The correlated model will apply to aircraft that are using a discrete (non-1200) mode A code.

A separate electronic file is provided by Lincoln Laboratory that contains the statistical data required to generate encounter trajectories. Details on how to interpret the data file and an example of randomly constructing trajectories are provided in Appendices A and B, respectively.

ACKNOWLEDGMENTS

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), and the Department of Defense Unmanned Aircraft System Airspace Integration Joint Integrated Product Team.

The authors greatly appreciate the support and assistance provided by Kevin “Spanky” Kirsch, Merv Leavitt, Joseph Kielman, and David Thomasson from the Department of Homeland Security, and Major Luke Cropsey, Lt. Col. Kent Tiffany, and Mr. Tony Salmonson from the Air Force.

The authors appreciate the support from Hill Air Force Base, including Jeff Richardson, Steven H. Schimmelpfennig, Richard C. Whitlock, Lt. Han Saydam, Lt. Tanuxay Keooudom, TSgt. Christopher B. Cosper, and James Evans.

Finally, the authors would also like to thank Vito D. Cavallo, Ann C. Drumm, Robert D. Grappel, J. Daniel Griffith, Garrett S. Harris, Jessica E. Olszta, John O’Rourke, Angela Sharer, and Steven D. Thompson for their contributions to the airspace encounter model effort.

TABLE OF CONTENTS

	Page
Abstract	iii
Acknowledgments	v
List of Illustrations	ix
List of Tables	xi
1. INTRODUCTION	1
1.1 Encounter Types	1
1.2 Process Overview	3
2. MODEL	5
2.1 Model Variables	5
2.2 Initial Distribution	6
2.3 Transition Distribution	7
3. ESTIMATION	9
3.1 Radar Tracks	9
3.2 Outlier Removal	12
3.3 Track Smoothing	12
3.4 Interpolation	12
3.5 Feature Extraction	13
3.6 Feature Smoothing	14
3.7 Quantization	14
3.8 Statistics Extraction	14
4. SAMPLING	21
4.1 Discrete Sampling	21
4.2 Continuous Sampling	22
5. SIMULATION	25
5.1 Encounter Cylinder Dimensions	25
5.2 Initial Conditions	26
5.3 Trajectory Construction	27

5.4	Multiple Encounters	28
6.	SAFETY EVALUATION	29
7.	SUMMARY AND FURTHER WORK	31
	APPENDIX A: MODEL PARAMETERS	33
	APPENDIX B: TRAJECTORY GENERATION	37
	B.1 Initial Network Sampling	37
	B.2 Transition Network Sampling	40
	APPENDIX C: BAYESIAN NETWORKS	45
	C.1 Definition	45
	C.2 Sampling	45
	C.3 Parameter Learning	45
	C.4 Structure Learning	47
	APPENDIX D: DENSITY PLOTS	49
	APPENDIX E: SENSORS	57
	APPENDIX F: TRACKING AND FUSION	61
	F.1 Adding Local Sensor Tracks to Global Tracks	62
	F.2 Determining Track Airspeed and Heading	64
	APPENDIX G: NETWORK CANDIDATES	69
	G.1 Initial Network Candidates	69
	G.2 Transition Network Candidates	73
	References	77

LIST OF ILLUSTRATIONS

Figure No.		Page
1	Modeling and simulation process overview.	4
2	Bayesian networks representing the variable dependency structure for the initial and transition distributions.	8
3	Estimation process flow.	10
4	Radar coverage map.	11
5	VFR density between 500 ft AGL and FL180 in $\log_{10}$ aircraft per NM^3 .	11
6	Preprocessing.	16
7	Piecewise-cubic Hermite interpolation.	16
8	A plot of extracted features over time.	17
9	Feature histograms of observed radar data based on 97 million samples.	18
10	Convergence curve for Bayesian network conditional probabilities.	19
11	Sampling process flow.	21
12	A plot of sampled features over time.	23
13	A track generated by sampling from the initial and transition distributions.	24
14	Criteria for accepting a trajectory for simulation.	26
15	Variables relevant to rejection sampling in the horizontal plane.	27
16	Initial bearing distributions produced by rejection sampling for two randomly-generated VFR aircraft.	28
B-1	A graphical representation of the initial network.	38
B-2	A graphical representation of the transition network.	41
D-1	VFR density over all altitude layers up to FL180 in $\log_{10}$ aircraft per NM^3 .	50

D-2	VFR density between 500 and 1200 ft AGL in $\log_{10}$ aircraft per NM^3 .	51
D-3	VFR density between 1200 and 3000 ft AGL in $\log_{10}$ aircraft per NM^3 .	52
D-4	VFR density between 3000 and 5000 ft AGL in $\log_{10}$ aircraft per NM^3 .	53
D-5	VFR density between 5000 ft AGL and 10,000 ft MSL in $\log_{10}$ aircraft per NM^3 .	54
D-6	VFR density between 10,000 ft MSL and FL180 in $\log_{10}$ aircraft per NM^3 .	55

LIST OF TABLES

Table No.		Page
1	Encounter model types.	3
2	Cut points used for feature quantization.	14
3	Sampling boundaries.	22
E-1	Beacon hours	57
F-1	Smoothing values depending on the current and previous turn states.	66

1. INTRODUCTION

One of the main challenges to integrating unmanned aircraft into the National Airspace System is the development of systems that are able to sense and avoid local air traffic. If designed properly, these collision avoidance systems could provide an additional layer of protection that maintains or even enhances 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 larger transport aircraft. More recently, Eurocontrol and the International Civil Aviation Organization (ICAO) performed similar sets of simulation studies for European and worldwide TCAS mandates [5, 6].

The design of a collision avoidance system represents a careful balance to enhance safety while ensuring a low rate of unnecessary maneuvers. 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 based on radar data [1, 3, 7–10]. The primary contribution of this report is to introduce a new approach to encounter modeling that is based on a Bayesian statistical framework and which uses recent radar data collected across the U.S. The advantage of applying a Bayesian framework is that it allows us to optimally leverage available radar data to produce a model that is representative of actual operations.

1.1 ENCOUNTER TYPES

The encounters covered by this model are those involving aircraft in the final stages before a collision over a period of time on the order of one minute. It is assumed that prior safety layers (e.g., airspace structure, Air Traffic Control (ATC) advisories or vectors) have failed to maintain standard separation distances between aircraft. The model is therefore useful in describing the types of situations that need to be addressed by a collision avoidance system, but will not address earlier aspects such as ATC communications or coordination.

There are two fundamental types of close 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 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 we term “uncorrelated” and involves at least one noncooperative 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 (or some other collision avoidance system) 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 close proximity; 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.

This report focuses on uncorrelated encounter modeling. In this report, we base our uncorrelated model on beacon reports from aircraft squawking 1200, not radar returns from noncooperative traffic. Radar surveillance of noncooperative targets is typically complicated due to clutter and missed detections, making identification of real tracks difficult. The lack of a transponder means that only position information, and no identity code or altitude, is available.¹ Hence, it is difficult to infer vertical rates, an important component of an encounter model.

Beacon-equipped aircraft can transmit either a discrete Mode A code or the non-discrete code of 1200. Aircraft using code 1200 tend to be general aviation typically flying VFR. They generally fly at low altitudes and make significantly more maneuvers than transport aircraft, both horizontally and vertically. Thus to a large degree they resemble noncooperative aircraft. Due to the poor quality of noncooperative data, we use 1200 tracks here as surrogates for primary-only tracks for the construction of this model. Other work is underway toward processing true non-cooperative tracks and will be described in a future document.

The 1200 tracks are a suitable surrogate for much of the noncooperative traffic in the National Airspace System, but there are certain categories of noncooperative targets for which they are not suitable. For example, most balloons, ultralights, and gliders do not fly like transponder-equipped aircraft squawking 1200. The challenge in developing models for unconventional aircraft such as balloons, ultralights, and gliders is the lack of high-quality data. As these data become available, the same processes outlined in this report will be used to estimate a noncooperative model for unconventional aircraft. Additionally, where such data are not available, a model can be manually constructed based on knowledge of typical flight trajectories and performance limits. Table 1 shows which encounter model to use depending on the types of aircraft involved in the encounter.

¹Some military radars have height-finding capability, although the accuracy of the altitude generated is significantly inferior to the transponder Mode C altitude.

TABLE 1. Encounter model types. There are three types of encounter models: correlated (C), uncorrelated conventional VFR (V), and uncorrelated unconventional VFR (U). This table shows which model to use depending on the types of aircraft involved in the encounter. In terms of aircraft types, Discrete indicates a cooperative aircraft using a non-1200 Mode A transponder code; VFR denotes a cooperative aircraft using the 1200 Mode A transponder code; Non-coop. Conv. indicates a non-cooperative but conventional fixed-wing aircraft; Non-coop. Unconv. indicates a non-cooperative and unconventional aircraft (e.g., balloon, ultralight). Only the uncorrelated conventional VFR model (in bold) is discussed in this report; the other models are under development.

Intruder Aircraft	Aircraft of Interest	
	Discrete	VFR
Discrete	C	C
VFR	C	V
Non-coop. Conv.	V	V
Non-coop. Unconv.	U	U

1.2 PROCESS OVERVIEW

Figure 1 diagrams the steps involved in processing radar data to build the encounter model. The high-level approach is to model nominal aircraft trajectories using Markov models represented by Bayesian networks (Section 2). The first step in constructing a Markov model involves extracting features, such as turn rate and vertical rate, from the radar data. From these features, we collect sufficient statistics that describe the distribution over maneuvers and other properties of trajectories. We use Bayesian model selection methods to search for the best network structure for representing the observed data. (Section 3). We use the best network structure and the associated sufficient statistics to generate new trajectories that are representative of those observed by radar (Section 4). The trajectories are then used to simulate encounters between aircraft (Section 5). A Monte Carlo safety analysis involves running a large collection of encounters and collecting data about near mid-air collision (NMAC) frequency (Section 6). The final section of this report summarizes and discusses further work.

The main outputs of the work described in this report are the sufficient statistics and model structure. A digital representation of the sufficient statistics and model structure are publicly available from Lincoln Laboratory (Appendix A). An example of using the data in that file to construct a random trajectory is provided in Appendix B.

Throughout this report we use the standard units used in aviation. In particular, altitudes are in feet, positions are in nautical mile coordinates, vertical rates are in feet per minute, turn rates are in degrees per second, airspeeds are in knots (true airspeed), and accelerations are in knots per second. Time is reported in seconds.

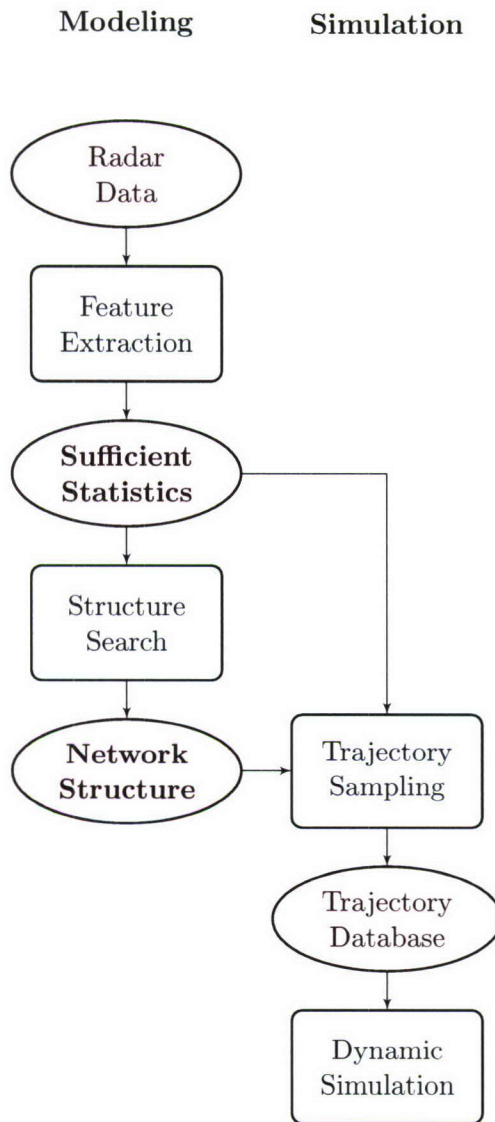


FIGURE 1. Modeling and simulation process overview. The sufficient statistics and network structure (in bold) are the main elements provided as part of this work.

2. 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 , vertical rate $\dot{h}$, altitude layer L , and airspace class A , we can infer from our model how the aircraft trajectory evolves over time.

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. Appendix A provides the necessary background on Bayesian networks. The remainder of this section outlines our approach.

2.1 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.

- **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 (Appendix A) was used to evaluate different Bayesian network structures and arrive at a structure that optimally represents the observed radar data.

2.2 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 2a 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 (Appendix C) to determine which structure is more likely given the data. We chose the network structure in Figure 2a because it had the highest score among the selection of candidate networks we considered (see Appendix G).

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 2a, 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. An example of working through this process is provided in Appendix B. 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.

2.3 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 2b. 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 [11, 12]. Parameter and structure learning in dynamic Bayesian networks is similar to regular Bayesian networks (Appendix C). Again, we chose the highest-scoring network structure among our candidate network structures (see Appendix G). 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.

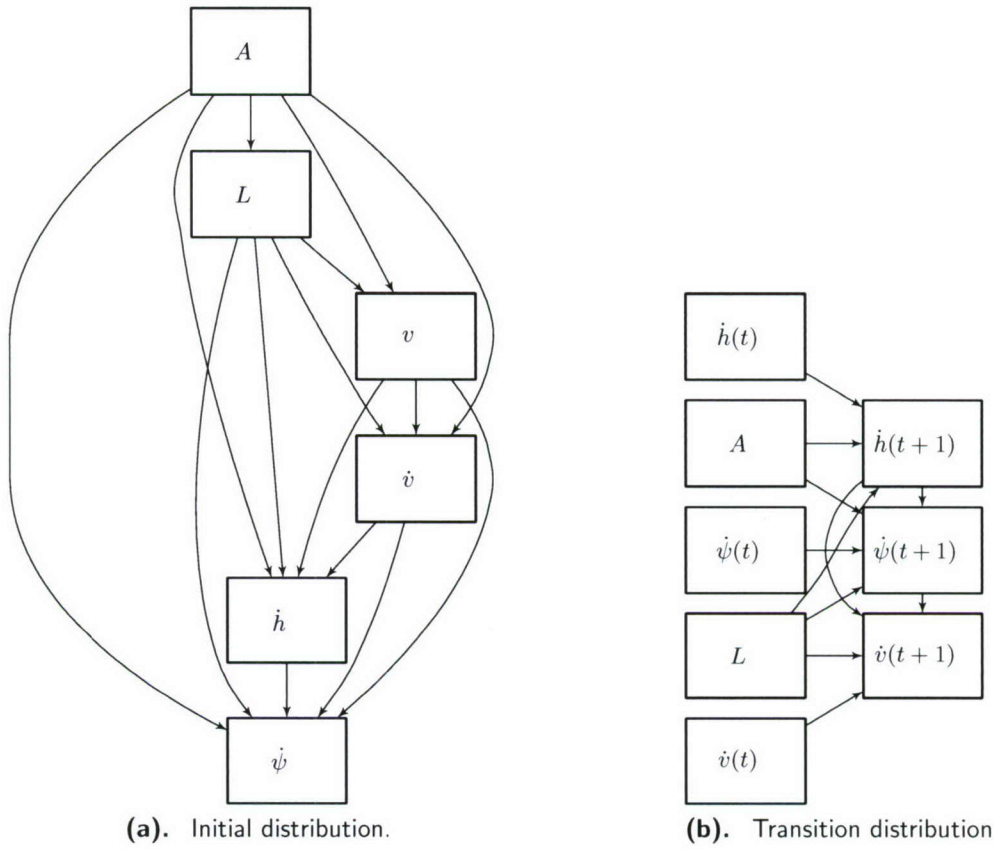


FIGURE 2. Bayesian networks representing the variable dependency structure for the initial and transition distributions.

3. ESTIMATION

This section describes the processing required to transform radar tracks into sufficient statistics that may be used to model uncorrelated encounters. Figure 3 outlines this process.

3.1 RADAR TRACKS

Our radar data stream 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 4 shows the coverage by the more than 120 sensors whose data was used to construct our model. Figure 5 shows the density of VFR traffic, and Appendix D plots flight hour densities at different altitude layers. Appendix E summarizes the volume of data received from each sensor.

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.

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.

To build our model, we collected VFR (1200-code) beacon reports between December 1, 2007 and December 7, 2007, amounting to 30,000,000 reports representing 78,000 flight hours. The raw radar data is first processed using a tracking algorithm developed at Lincoln Laboratory [13]. A fusion algorithm, also developed at Lincoln Laboratory [14], then fuses tracks from multiple sensors to give one global view of all the tracks in U.S. airspace (see Appendix F). 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).

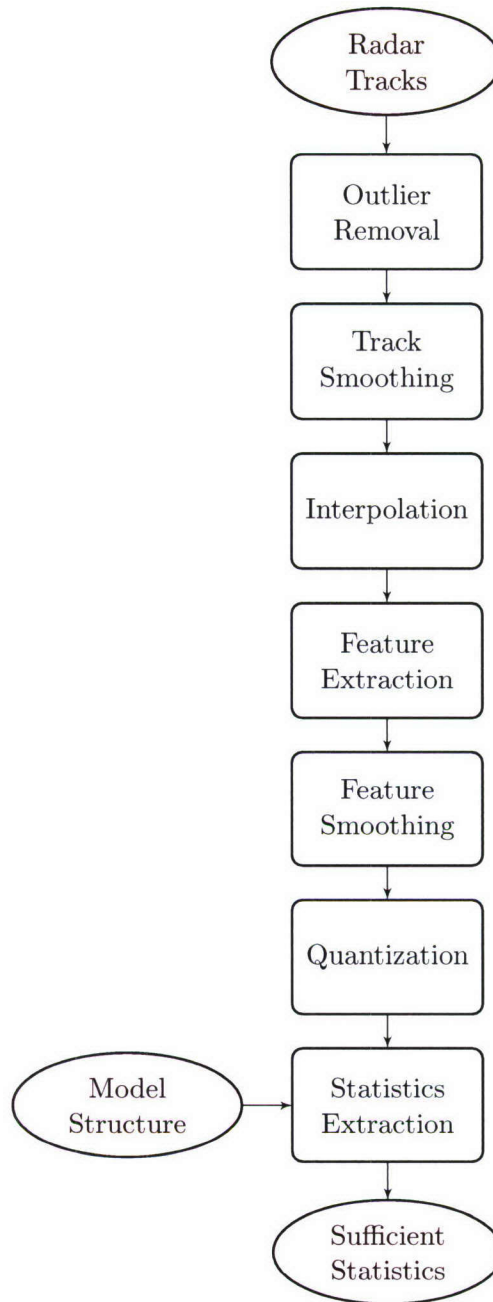


FIGURE 3. Estimation process flow.

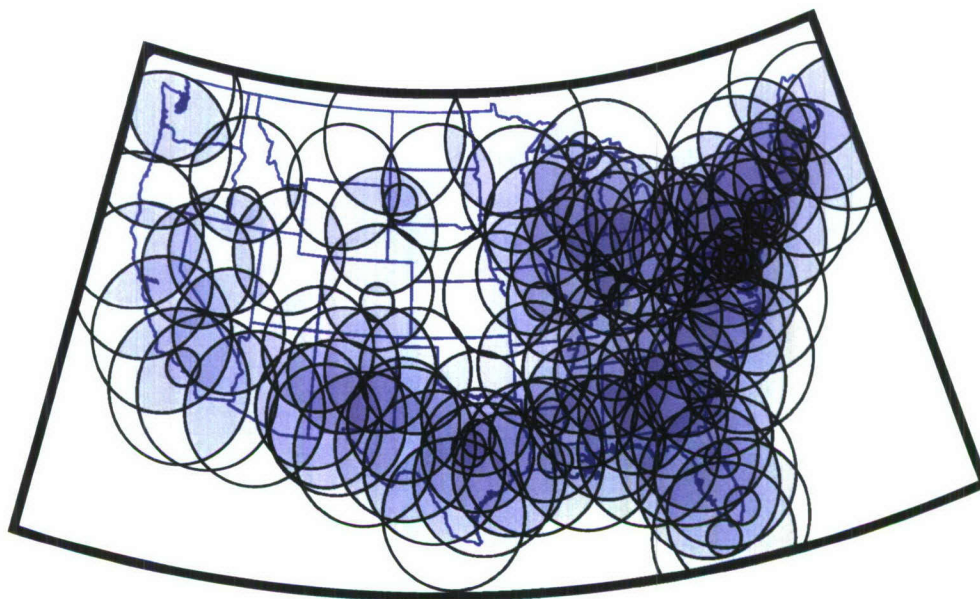


FIGURE 4. Radar coverage map.

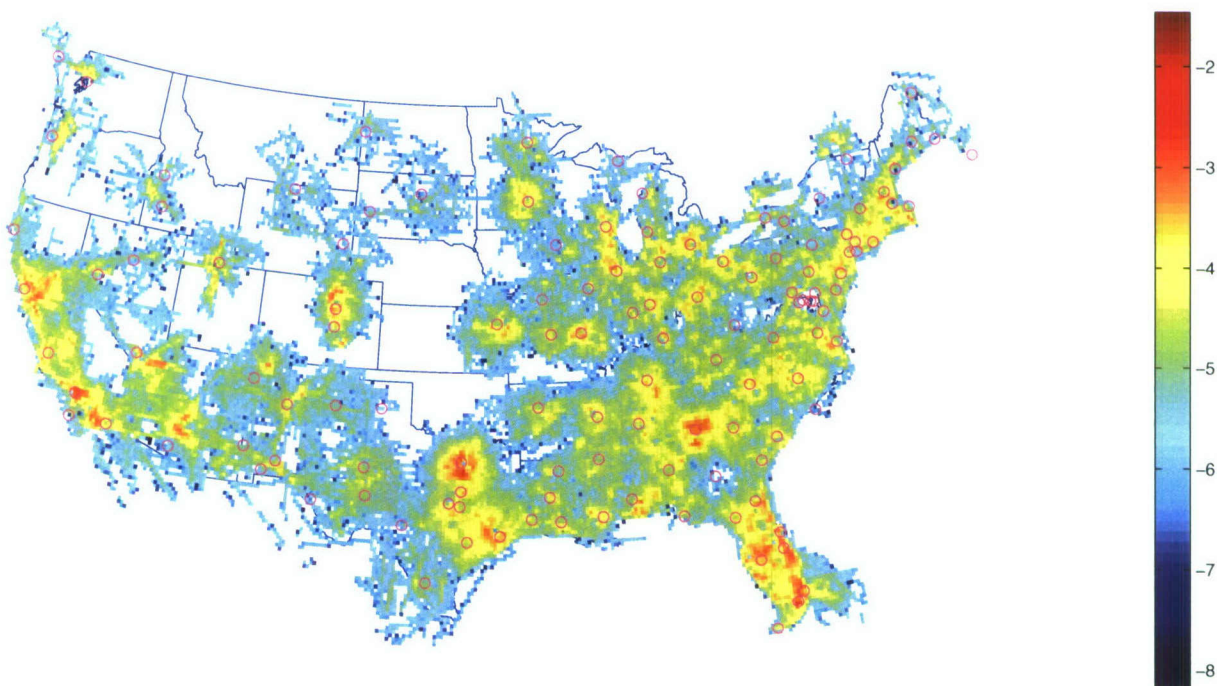


FIGURE 5. VFR density between 500 ft AGL and FL180 in $\log_{10}$ aircraft per NM^3 . The locations of the sensors are indicated by magenta circles.

3.2 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.

3.3 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 $(t_1, \mathbf{x}_1), \dots, (t_n, \mathbf{x}_n)$ to a smoothed trajectory $\mathbf{y}_1, \dots, \mathbf{y}_n$.

$$\mathbf{y}_i = \frac{\sum_j w(t_i, t_j) \mathbf{x}_j}{\sum_j w(t_i, t_j)}, \quad (1)$$

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 ,

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

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 on horizontal and vertical profiles in our data set; the chosen values preserve the underlying tracks while removing noise.

3.4 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 seconds, and en route (long range) radars scan aircraft every 10 to 12 seconds. 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 3.2). 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 [15].

Figure 6 shows the result of outlier detection, smoothing, and interpolation on a track from our data set. Figure 7 shows the result of piecewise-cubic Hermite interpolation on a smoothed track.

3.5 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 [16]. 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 8 shows the result of feature extraction on the same track shown in Figure 6.

TABLE 2. 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

3.6 FEATURE SMOOTHING

We then smooth the extracted features using the same smoothing scheme we used for tracks (Section 3.3). 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 in this step because the altitudes are already smoothed (Section 3.3).

3.7 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 $c_1, \dots, c_n$. 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 i th bin. The cut points we used for quantization are listed in Table 2. The cut points were chosen to capture the variation of the features as shown in the histograms in Figure 9.

3.8 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 C).² Appendix A describes the sufficient statistics extracted from our data.

Figure 10 illustrates the convergence of the Bayesian network parameters. The horizontal axis represents the number of samples used to estimate the parameters of the Bayesian network, and the vertical axis represents the maximum difference between elements in the conditional probability tables with the addition of more data. As the figure shows, the change in the probabilities in the conditional probabilities converges to zero.

²The counts are called *sufficient statistics* because they provide a summarization of the data that is sufficient to compute the posterior distribution from the prior. For an introduction to Bayesian statistics, see [17].

This section has described how to construct a model of nominal VFR flight based on radar data. The next section describes how to use this model to produce new trajectories through sampling.

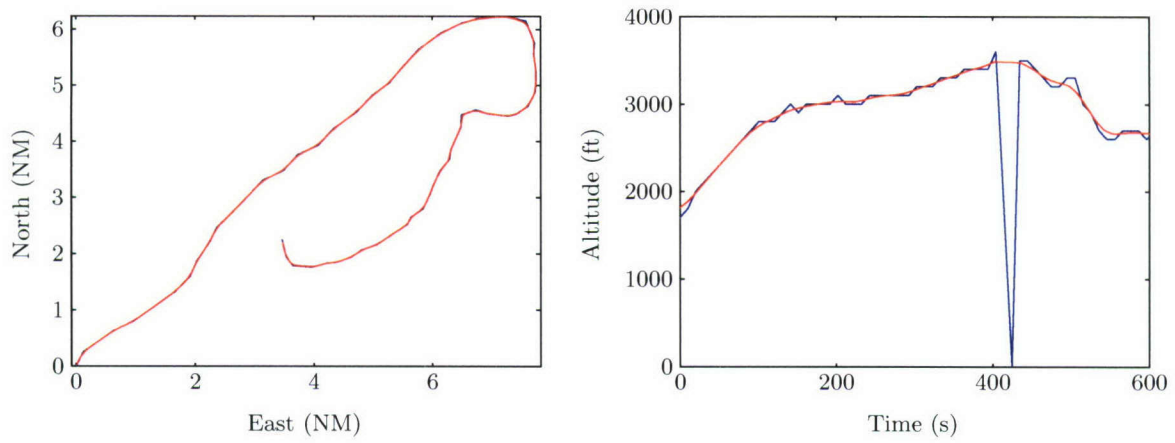


FIGURE 6. Preprocessing. Blue lines show an example raw track. Red lines show the track after outlier removal, smoothing, and interpolation.

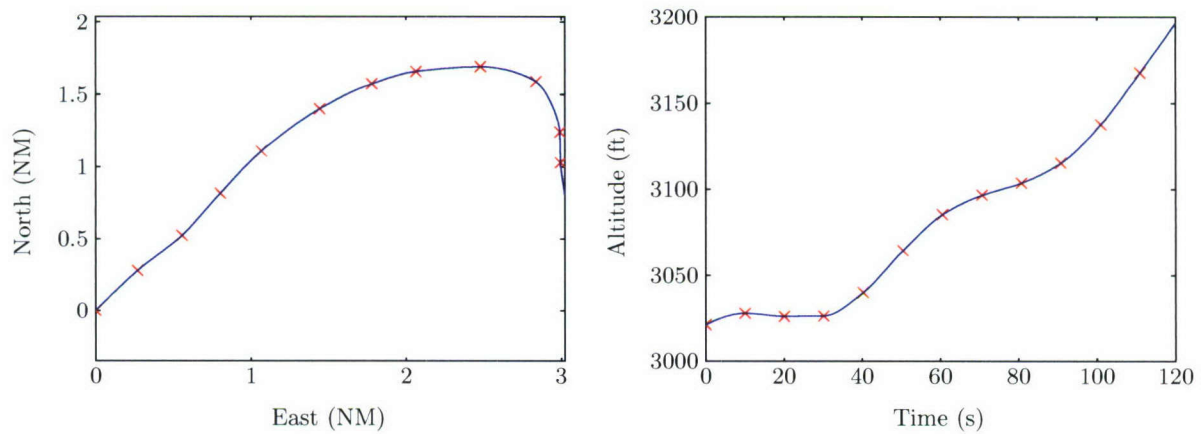


FIGURE 7. Piecewise-cubic Hermite interpolation on a smoothed track. Red crosses indicate smoothed data points, and the blue curve shows the interpolation.

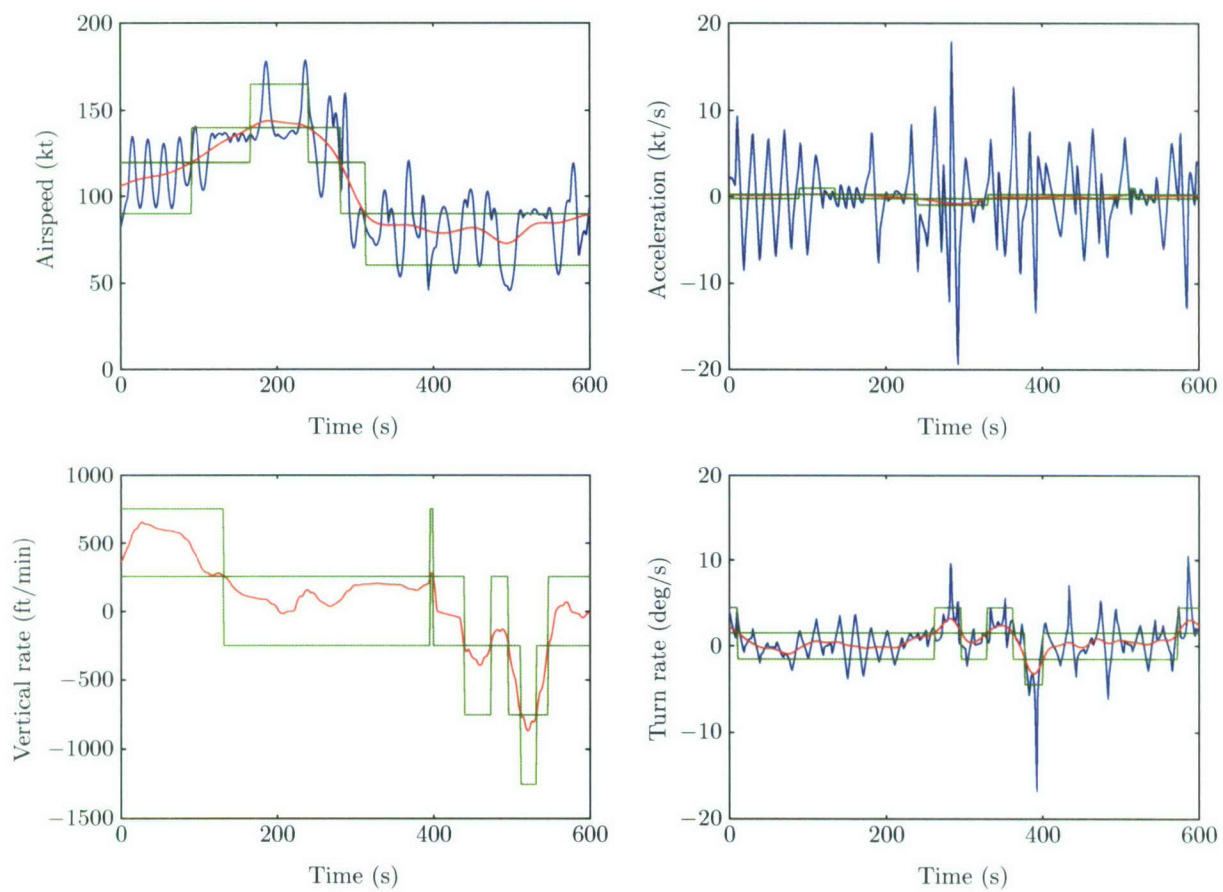


FIGURE 8. A plot of extracted features over time. Blue lines show features before smoothing, and red lines show features after smoothing. The green blocks show the outlines of the bins to which the features belong.

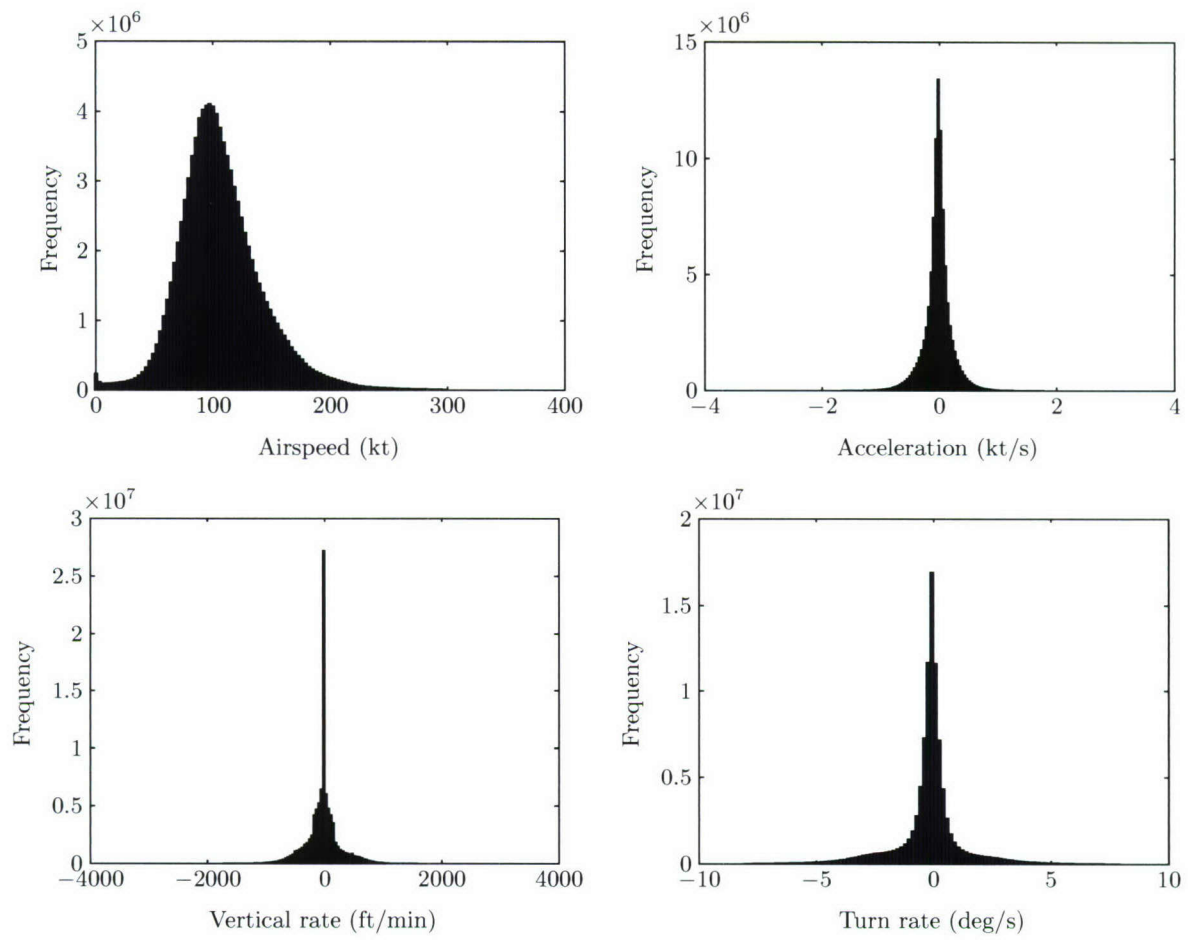


FIGURE 9. Feature histograms of observed radar data based on 97 million samples.

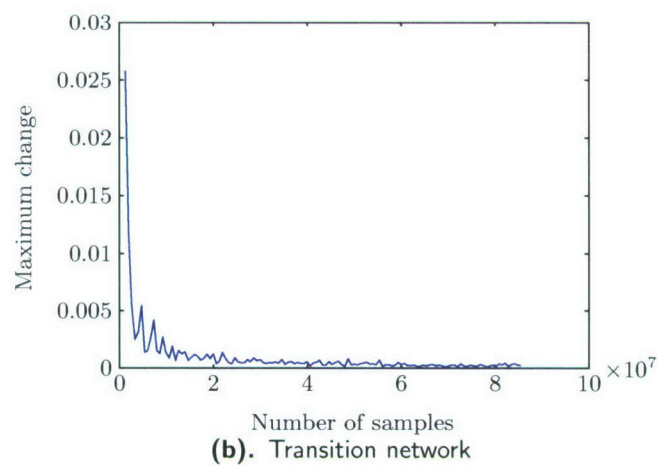
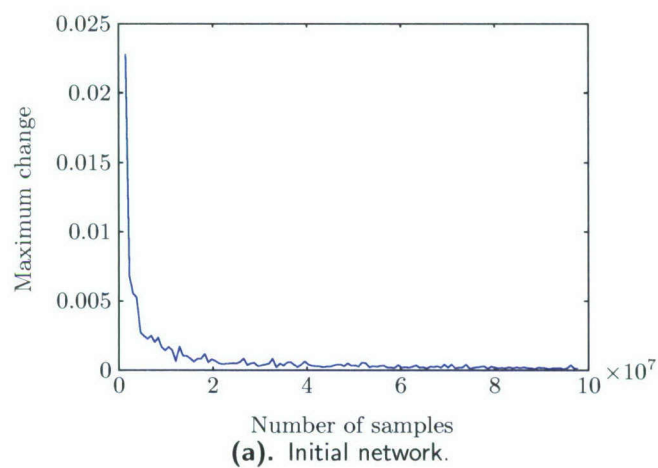


FIGURE 10. Convergence curve for Bayesian network conditional probabilities.

4. 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 11 illustrates this process. An example of working through this process using the data provided for this encounter model is given in Appendix B.

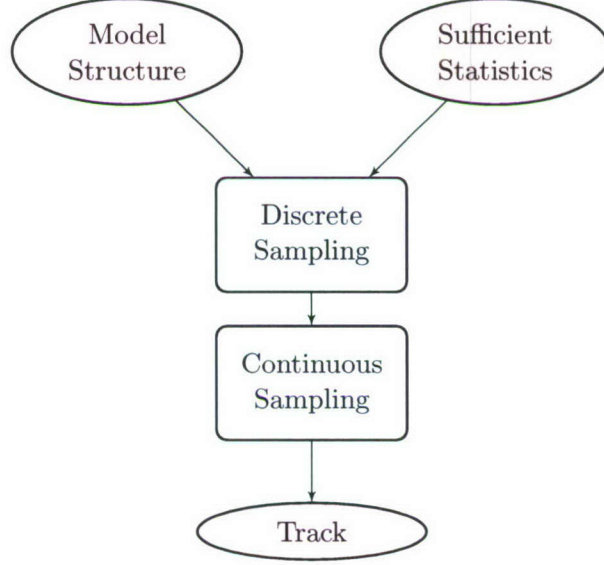


FIGURE 11. Sampling process flow.

4.1 DISCRETE SAMPLING

We begin by sampling from the Bayesian network representing the initial distribution. As outlined in Appendix C and shown in more detail in Appendix B, we randomly assign value k to variable X_i with probability

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

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. Sampling from the posterior distribution with $\alpha_{ijk} = 1$ is equivalent to adding 1 to all the counts in the tables in Appendix A and sampling according to the resulting relative frequencies. This ensures that there are no transitions with zero probability in our Markov model.

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 4.2.

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.

4.2 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 2, 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 3 shows the boundaries in our quantization based on the limits observed in the radar data.

TABLE 3. Sampling boundaries.

	Boundaries
h	500, 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 0.0211495, 0.0412835, and 0.0885336 changes per second for acceleration, vertical rate, and turn rate respectively. We estimated these rates from the data by introducing 3 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.

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 12 plots an example of vertical rates, turn rates, accelerations, and airspeeds generated by sampling from the Bayesian networks. Figure 13 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.

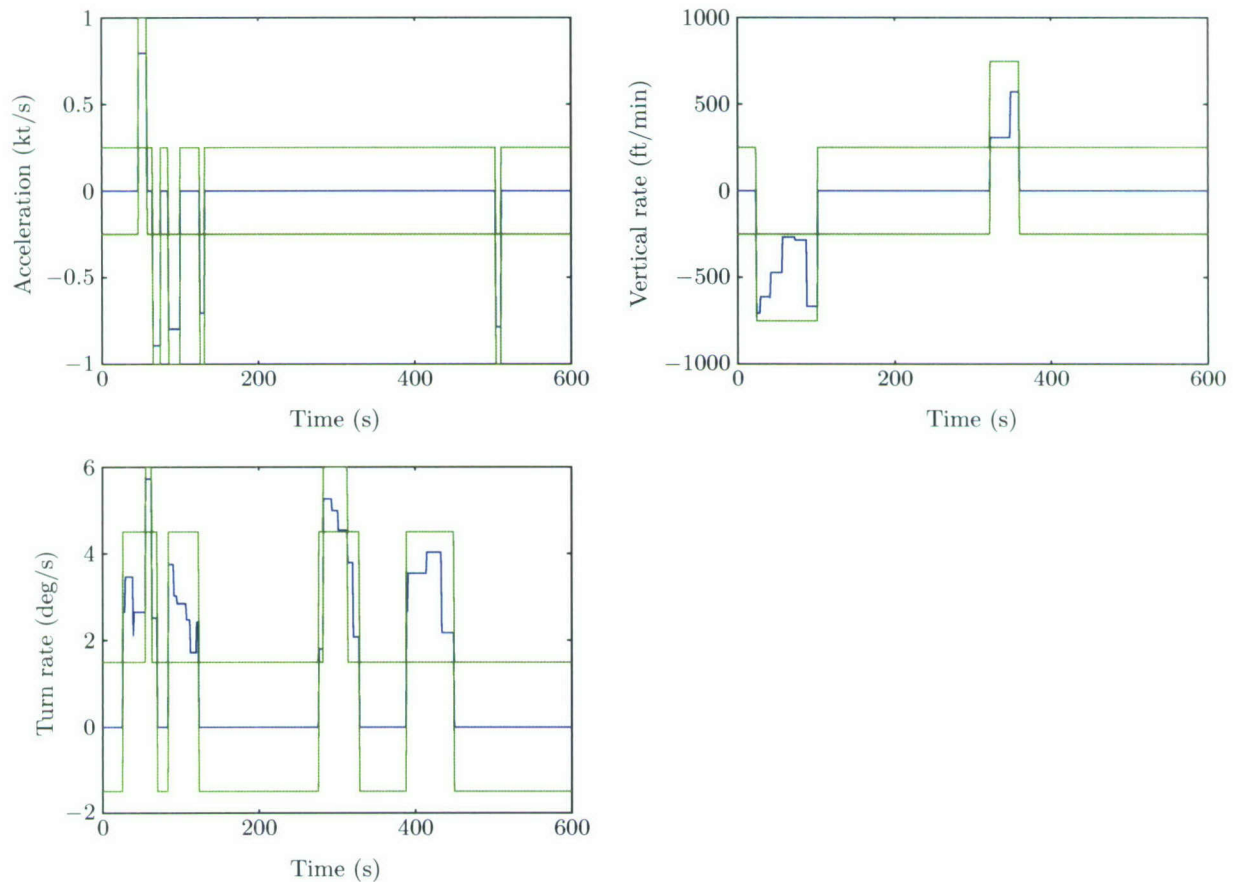


FIGURE 12. A plot of sampled features over time. Blue lines show continuous samples, and green blocks show the outlines of the discrete samples.

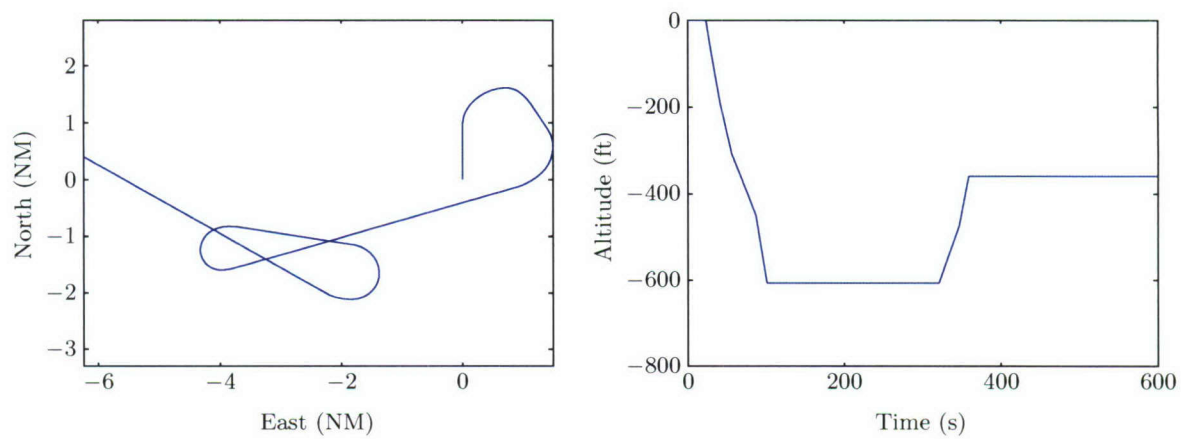


FIGURE 13. A track generated by sampling from the initial and transition distributions. Positions and altitudes are relative to the initial position and altitude.

5. 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 or AC2 suggests an avoidance maneuver, it overrides the nominal rates commanded by the encounter model. If AC2 enters an NMAC cylinder or exits the encounter cylinder, the encounter run is terminated. The NMAC cylinder has radius r_{nmac} and height $2h_{\text{nmac}}$. We use $r_{\text{nmac}} = 500$ ft and $h_{\text{nmac}} = 100$ ft.

Running millions of randomly generated encounters allows us to estimate the probability that an aircraft that enters the encounter cylinder penetrates the NMAC cylinder before exiting the encounter cylinder. We use $P(\text{nmac} \mid \text{enc})$ to represent this probability. Section 6 explains how to use this probability to estimate the mean time between NMACs.

5.1 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.

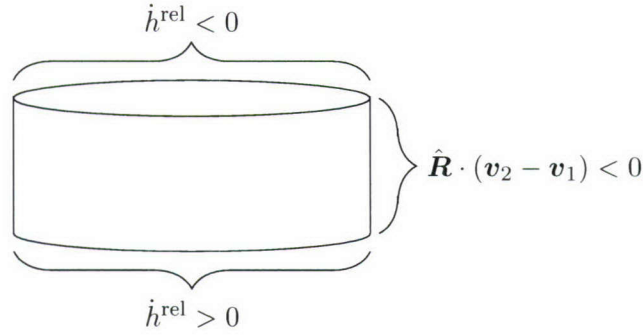


FIGURE 14. Depending on where on the encounter cylinder the intruder is initialized, different criteria must be met in order to accept a given trajectory for further simulation.

5.2 INITIAL CONDITIONS

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 14). 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.
2. Randomly 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 probability of being initialized on the top, bottom, or side is proportional to its surface area. 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 $\dot{h}^{rel}$, is negative. This ensures that AC2 is penetrating the encounter cylinder in the first time step.
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 $\dot{h}^{rel}$, is positive. This ensures that AC2 is penetrating the encounter cylinder in the first time step.
6. If AC2 was initialized on the side of the encounter cylinder, accept the sample if $\hat{\mathbf{R}} \cdot (\mathbf{v}_2 - \mathbf{v}_1)$ is negative, where $\hat{\mathbf{R}} = (\sin \chi, \cos \chi)$ is the unit vector from AC1 to AC2. The vectors $\mathbf{v}_1$ and $\mathbf{v}_2$ are the velocities of AC1 and AC2 respectively. When $\hat{\mathbf{R}} \cdot (\mathbf{v}_2 - \mathbf{v}_1)$ is negative, the relative velocity of AC2 is into the encounter cylinder, and therefore should be accepted. See Figure 15.

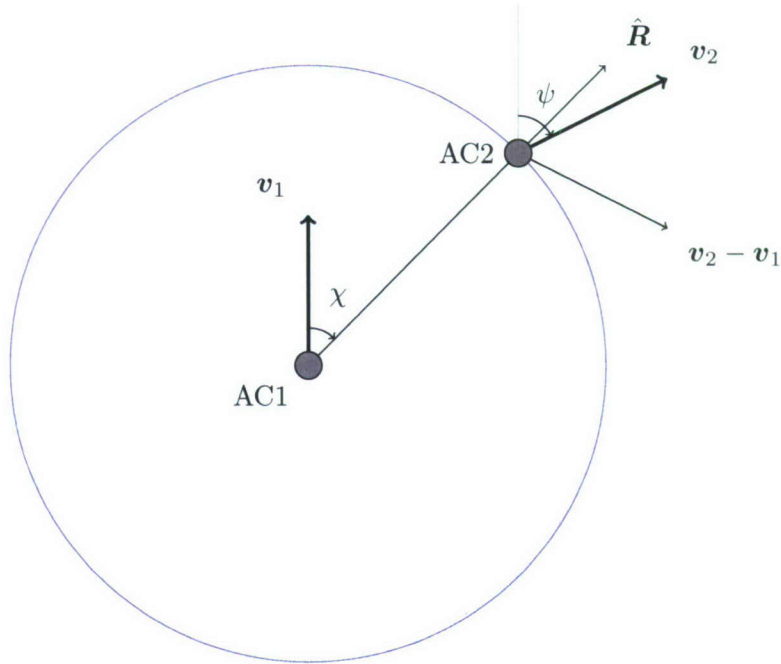


FIGURE 15. Variables relevant to rejection sampling in the horizontal plane.

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 16 shows bearing distributions of VFR/VFR encounters.

5.3 TRAJECTORY CONSTRUCTION

Once the initial conditions are selected, the dynamic models of AC1 and AC2 are used to update their trajectories over time.

Given the initial state of an aircraft (including at a minimum: position, altitude, heading, climb rate, turn rate, and velocity) and its control variables ($\dot{h}$, $\dot{\psi}$, and $\dot{v}$) the aircraft state is updated in 1 s time steps using a dynamic model. Due to the wide variety of possible dynamic model implementations, details for computing this state update are not provided here. A basic approach would be to apply simple point-mass kinematics to update the aircraft state without considering what the aircraft may actually be doing in terms of bank angle, pitch rate, etc. More complex implementations could include 6 degree-of-freedom dynamic models in which the control variables are treated as target states provided to an autoflight control system which then applies the necessary control deflections to attain those targets. Lincoln Laboratory's Collision Avoidance System Safety Assessment Tool (CASSATT) typically uses a 4 degree-of-freedom model to update aircraft state by applying the necessary airspeed acceleration, roll rate, and pitch rate to achieve

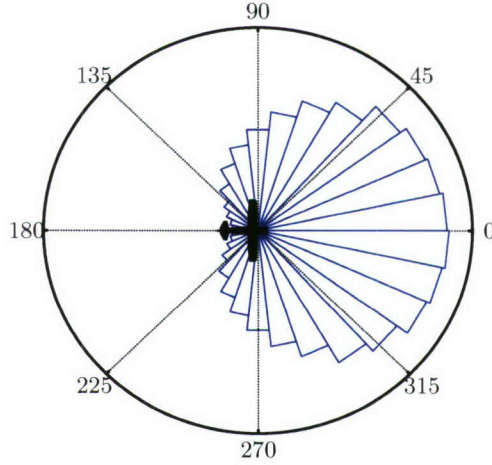


FIGURE 16. Initial bearing distributions produced by rejection sampling for two randomly-generated VFR aircraft.

the target values for $\dot{h}$, $\dot{\psi}$, and $\dot{v}$ assuming curvilinear motion with a zero-sideslip constraint. The zero-sideslip constraint can be relaxed, for instance, if it is necessary to model transient dynamics with higher fidelity.

A simulation run terminates when the intruder either exits the encounter cylinder or penetrates the NMAC cylinder. After running many simulations, we estimate $P(\text{nmac} \mid \text{enc})$ by dividing the number of runs that resulted in an NMAC by the total number of runs.

5.4 MULTIPLE ENCOUNTERS

The approach discussed in Section 5.2 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

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

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.

6. SAFETY EVALUATION

The previous section described how to compute $P(\text{nmac} \mid \text{enc})$, the probability that an aircraft that enters the encounter cylinder penetrates the NMAC cylinder before exiting the encounter cylinder. This section explains how to compute the rate at which aircraft enter the encounter cylinder, λ_{enc} . Once we know $P(\text{nmac} \mid \text{enc})$ and λ_{enc} , we may estimate the NMAC rate

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

The mean time between NMACs is simply $\lambda_{\text{nmac}}^{-1}$.

We begin with the assumption that the density of traffic outside the encounter cylinder is fixed and known. This density, ρ , has units $(\text{NM}^2 \times \text{ft})^{-1}$. The rate at which aircraft enter the encounter cylinder depends on ρ and the average volume of new airspace the encounter cylinder sweeps through per unit time.

If the average relative horizontal speed is $\bar{s}_{\text{horiz}}^{\text{rel}}$ and the average relative vertical speed is $\bar{s}_{\text{vert}}^{\text{rel}}$, then the average volume of new airspace the encounter cylinder sweeps through per unit time is

$$4r_{\text{enc}}h_{\text{enc}}\bar{s}_{\text{horiz}}^{\text{rel}} + \pi r_{\text{enc}}^2 \bar{s}_{\text{vert}}^{\text{rel}}. \quad (5)$$

The average relative vertical speed is given by

$$\bar{s}_{\text{vert}}^{\text{rel}} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} p(\dot{h}_1)p(\dot{h}_2) \left| \dot{h}_1 - \dot{h}_2 \right| d\dot{h}_1 d\dot{h}_2, \quad (6)$$

where $\dot{h}_1$ is the vertical rate of the unmanned aircraft and $\dot{h}_2$ is the vertical rate of the intruder. The average relative horizontal speed is given by

$$\bar{s}_{\text{horiz}}^{\text{rel}} = \frac{1}{\pi} \int_0^\pi \int_0^\infty \int_0^\infty p(v_1^g)p(v_2^g) \sqrt{(v_1^g + v_2^g \cos \alpha)^2 + (v_2^g \sin \alpha)^2} dv_1^g dv_2^g d\alpha, \quad (7)$$

where v_1^g is the groundspeed of the unmanned aircraft and v_2^g is the ground speed of the intruder. Both Equations 6 and 7 may be estimated through sampling. We estimate $\bar{s}_{\text{horiz}}^{\text{rel}}$ to be 147 kt for VFR/VFR encounters. We estimate $\bar{s}_{\text{vert}}^{\text{rel}}$ to be 260 ft/min for VFR/VFR encounters.

If we make the following two 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,

then the encounter rate is

$$\lambda_{\text{enc}} = \rho \left(4r_{\text{enc}}h_{\text{enc}}\bar{s}_{\text{horiz}}^{\text{rel}} + \pi r_{\text{enc}}^2 \bar{s}_{\text{vert}}^{\text{rel}} \right). \quad (8)$$

The NMAC rate is simply the product of λ_{enc} and $P(\text{nmac} \mid \text{enc})$.

7. SUMMARY AND FURTHER WORK

This report 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 report 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. We are also developing a noncooperative model of unconventional aircraft such as balloons, ultralights, and gliders.

The uncorrelated encounter model presented in this report 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.

The model presented in this report will be reevaluated over Spring and Summer 2008, and a final version will be published in September 2008.

APPENDIX A

MODEL PARAMETERS

This appendix describes the sufficient statistics, N_{ijk} (see Appendix C), used to estimate the conditional probabilities associated with the initial and transition distributions. These sufficient statistics are based on beacon reports from aircraft squawking VFR (Mode A code 1200) between December 1, 2007 and December 7, 2007. This amounts to 30,109,411 reports representing 78,000 flight hours from across the United States. Other parameters that are relevant to generating new encounters from the model are also described in this section.

A text file, available electronically from Lincoln Laboratory, describes the following model parameters:

- **Variable labels:** A quoted, comma-delimited list specifies the variable labels, e.g. `\dot \psi`, as would be used by $\text{\LaTeX}$. There are different variable labels for the initial network and the transition network.
- **Graphical structure:** A binary matrix is used to represent graphical structure. A 1 in the i th row and j th column means that there is a directed edge from the i th variable to the j th variable in the Bayesian network. The text file specifies two graphical structures; one for the initial network and the other for the transition network. The element in the i th row and j th column is represented by $G(i, j)$.
- **Variable instantiations:** For each network, a list of integers specifies the number of instantiations that exist for each variable.
- **Sufficient statistics:** For each network, a list of integers specifies the sufficient statistics. We explain how to interpret the array of integers below.
- **Boundaries:** The boundaries of the variable bins are specified by a row of numbers. The variables A and L are not quantized because they are already discrete, and so boundaries do not exist. A $*$ is used for these variables.
- **Resampling rates:** A list of numbers specify the resampling rates (Section 4.2).

The list of numbers describing the sufficient statistics, N_{ijk} , requires explanation. The array is ordered first by increasing k , then increasing j , and then increasing i . For some specified variable i , parental instantiation j , and variable instantiation k , the value N_{ijk} is given by the following element in the list

$$k + r_i(j - 1) + \sum_{i'=1}^{i-1} q_{i'} r_{i'}, \quad (\text{A-1})$$

where q and r are as specified in Appendix C. It is important to clarify the ordering of the parental instantiations. If the variables are instantiated to bins $b_1, \dots, b_n$, the parental instantiation of

variable i is given by

$$j = 1 + \sum_{i'=1}^n G(i', i) (b_{i'} - 1) \prod_{i''=1}^{i'-1} r_{i''}^{G(i'', i)}. \quad (\text{A-2})$$

The following is a fragment of the parameter file. The lines that describe the sufficient statistics are truncated due to length.

```
# labels_initial
"A", "L", "v", "\dot v", "\dot h", "\dot \psi"
# G_initial
0 1 1 1 1 1
0 0 1 1 1 1
0 0 0 1 1 1
0 0 0 0 1 1
0 0 0 0 0 1
0 0 0 0 0 0
# r_initial
4 4 8 5 7 7
# N_initial
487885 517728 10285433 86339819 249426 130407 78279 29773 302812 182518 ...
# labels_transition
"A", "L", "v", "\dot v(t)", "\dot h(t)", "\dot \psi(t)",
"\dot v(t+1)", "\dot h(t+1)", "\dot \psi(t+1)"
# G_transition
0 0 0 0 0 0 0 1 1
0 0 0 0 0 0 1 1 1
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 0 0
0 0 0 0 0 0 0 1 0
0 0 0 0 0 0 0 0 1
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 0 1
0 0 0 0 0 0 1 0 0
# r_transition
4 4 8 5 7 7 5 7 7
# N_transition
135 1 0 0 0 168 1 0 0 0 39 0 0 0 0 59 0 0 0 0 3 921 46 0 0 3 902 20 0 0 ...
# boundaries
*
*
0 30 60 90 120 140 165 250 300
-2 -1 -0 0 1 2
-2000 -1250 -750 -250 250 750 1250 2000
```

```
-8 -6 -5 -2 2 5 6 8
# resample_rates
0 0 0 0.0211495 0.0412835 0.0885336
```

This document and the parameters file is available for download from <ftp://duke.atc.ll.mit.edu> as a ZIP file located at [/pub/outgoing/encounter_model/uncor-20080303.zip](ftp://duke.atc.ll.mit.edu/pub/outgoing/encounter_model/uncor-20080303.zip). This FTP site allows for anonymous access. Sign in with the username anonymous and your email address as your password.

APPENDIX B

TRAJECTORY GENERATION

This appendix explains how to generate a trajectory from our model.

B.1 INITIAL NETWORK SAMPLING

The first step in generating a random trajectory using our model is to sample from the Bayesian network representing the initial state distribution. To sample from a Bayesian network, as explained in Appendix C, is to produce a topological sort of the nodes in the network. A topological sort orders the nodes of the network so that parents come before their descendants. The following is the graphical structure of the initial network as specified in the parameters file (see Appendix A) and shown in Figure B-1:

```

0 1 1 1 1 1
0 0 1 1 1 1
0 0 0 1 1 1
0 0 0 0 1 1
0 0 0 0 0 1
0 0 0 0 0 0

```

As can be seen, this ordering of the nodes is already topologically sorted: the first node (airspace class) is connected to all other nodes. The second node (altitude layer) is connected to all following nodes, and so on. The final node (ψ) is not connected to any other nodes. We will see later that the transition network, in contrast, requires a topological sort.

With the nodes being topologically sorted, we begin by sampling the first variable. As specified in the parameters file, the first variable is A , airspace class. Equation C-5, reproduced below, shows how to produce a random sample:

$$P(X_i = k \mid \pi_{ij}, D, G) = \frac{\alpha_{ijk} + N_{ijk}}{\sum_{k'=1}^{r_i} (\alpha_{ijk'} + N_{ijk'})}.$$

In other words, the probability of selecting a particular airspace class is proportional to the prior (α_{ijk}) plus the frequency that airspace class appeared in the data (N_{ijk}). Following Cooper and Herskovits [18], we use an objective prior and set α_{ijk} to 1. To determine the values for N_{ijk} , we need to look at the sufficient statistics recorded in the parameters file. The sufficient statistics for the initial network is recorded as a long series of numbers. We use the scheme described in Appendix A to determine the actual values. These values turn out to be the first four numbers in the sufficient statistics sequence and are shown in the following table.

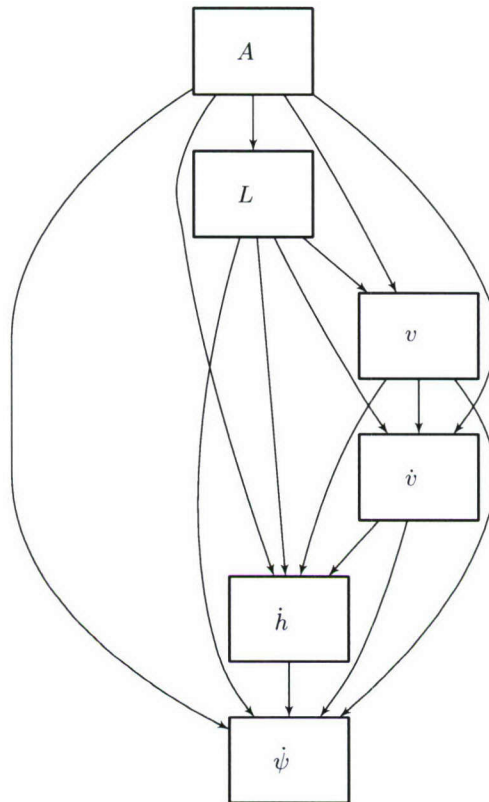


FIGURE B-1. A graphical representation of the initial network.

TABLE B-1. $N(A)$

A			
B	C	D	O
487885	517728	10285433	86339819

Now, we may compute the probability of selecting the different values of A using Equation C-5:

- $P(A = B) = (487885 + 1)/(487885 + 1 + 517728 + 1 + 10285433 + 1 + 86339819 + 1) = 487886/97630869 = 0.00499725143284344$
- $P(A = C) = (517728 + 1)/(487885 + 1 + 517728 + 1 + 10285433 + 1 + 86339819 + 1) = 517729/97630869 = 0.00530292319737521$
- $P(A = D) = (10285433 + 1)/(487885 + 1 + 517728 + 1 + 10285433 + 1 + 86339819 + 1) = 10285434/97630869 = 0.105350224835139$
- $P(A = O) = (86339819 + 1)/(487885 + 1 + 517728 + 1 + 10285433 + 1 + 86339819 + 1) = 86339820/97630869 = 0.884349600534642$

We use a random number generator to choose an airspace. For this example, let us choose O, the fourth instantiation of A .

The next step is to instantiate the second variable, L , which is altitude layer. Choosing a random instantiation for L is not quite as easy as for A because L depends upon other variables, namely A . We need to compute the conditional probability distribution $P(L | A = O)$, i.e. the distribution over the values of L given that A is O. We consult the sufficient statistics table $N(L | A)$ which is extracted from the parameters file (using the process explained in Appendix A) and is displayed in the table below.

TABLE B-2. $N(L | A)$

A	L			
	[500, 1200)	[1200, 3000)	[3000, 5000)	[5000, ∞]
B	249426	130407	78279	29773
C	302812	182518	32339	59
D	8383276	1901858	299	0
O	27280449	39197911	13704707	6156752

Since we are only interested in the counts associated with $A = O$, we consider only the last row in the table. To translate the counts into probabilities, we add 1 to each element in the last row

and then divide by the total resulting sum of that row. Again, we use a random number generator to select an airspace layer according to these probabilities. This process of randomly assigning values to each of the variables conditional on the values of their parents continues until all of the variables have been assigned. For this example, assume that the following assignments have been made:

- $A = O$
- $L = [3000, 5000)$
- $v = [120, 140)$
- $\dot{v} = [-0.25, 0.25)$
- $\dot{h} = [250, 750)$
- $\dot{\psi} = [-1.5, 1.5)$

Specific values within each bin are then determined based on a different uniform random number for each. In the case where a bin spans 0 (as it does in this example for $\dot{v}$, and $\dot{\psi}$), the value 0 itself is assigned.

B.2 TRANSITION NETWORK SAMPLING

We have described how to sample from the initial Bayesian network to generate a random initial state. The next step is to use the dynamic Bayesian network representing the transition distribution to generate the state at the next time step based on the initial state. The parameters file defines the following sequence of variables in the dynamic Bayesian network: $A, L, v, \dot{v}(t), \dot{h}(t), \dot{\psi}(t), \dot{v}(t+1), \dot{h}(t+1), \dot{\psi}(t+1)$. The parameters file also defines the following graphical structure for the network:

```

0 0 0 0 0 0 0 1 1
0 0 0 0 0 0 1 1 1
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 0 0
0 0 0 0 0 0 0 1 0
0 0 0 0 0 0 0 0 1
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 0 1
0 0 0 0 0 0 1 0 0

```

The graphical representation of this matrix is shown in Figure B-2. As can be seen, the ordering in the parameters file is not topologically sorted because there are arrows from $\dot{h}(t+1)$ and $\dot{\psi}(t+1)$ to $\dot{v}(t+1)$ but $\dot{v}(t+1)$ comes before both $\dot{h}(t+1)$ and $\dot{\psi}(t+1)$ in the list. We need to

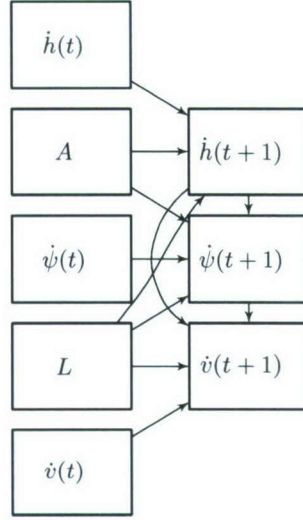


FIGURE B-2. A graphical representation of the transition network.

topologically sort the list of variables to make sure that we assign values to variables in the right order. One possible topological sort involves simply moving $\dot{v}(t+1)$ to the end of the list.

We are only interested in assigning new values to the dynamic variables, namely $\dot{h}(t+1)$, $\dot{\psi}(t+1)$ and $\dot{v}(t+1)$. The process is similar to the process used to sample from the initial network. First, we consult the table of sufficient statistics for $\dot{h}(t+1)$. This table of conditional counts is shown below. For sampling purposes, we are only interested in the row that represents the current variable assignment, i.e. $A = O$, $L = [3000, 5000)$, and $\dot{h} = [250, 750)$. The relevant line highlighted in the table:

TABLE B-3. $N(\dot{h}(t+1) \mid A, L, \dot{h}(t))$

$h(t+1)$									
A	L	$\dot{h}(t)$	$[-2000, -1250)$	$[-1250, -750)$	$[-750, -250)$	$[-250, 250)$	$[250, 750)$	$[750, 1250)$	$[1250, 2000]$
B	[500, 1200)	$[-2000, -1250)$	437	13	0	0	0	0	0
C	[500, 1200)	$[-2000, -1250)$	1436	23	0	0	0	0	0
D	[500, 1200)	$[-2000, -1250)$	32319	1059	0	0	0	0	0
O	[500, 1200)	$[-2000, -1250)$	44854	1503	2	0	0	0	0
B	[1200, 3000)	$[-2000, -1250)$	576	14	0	0	0	0	0
C	[1200, 3000)	$[-2000, -1250)$	1202	45	0	0	0	0	0
D	[1200, 3000)	$[-2000, -1250)$	14412	450	0	0	0	0	0
O	[1200, 3000)	$[-2000, -1250)$	73685	2484	2	0	0	0	0
B	[3000, 5000)	$[-2000, -1250)$	1137	34	0	0	0	0	0
C	[3000, 5000)	$[-2000, -1250)$	551	16	0	0	0	0	0
D	[3000, 5000)	$[-2000, -1250)$	32	1	0	0	0	0	0
O	[3000, 5000)	$[-2000, -1250)$	49962	1567	2	0	0	0	0
B	[5000, ∞]	$[-2000, -1250)$	1452	16	0	0	0	0	0
C	[5000, ∞]	$[-2000, -1250)$	0	0	0	0	0	0	0
D	[5000, ∞]	$[-2000, -1250)$	0	0	0	0	0	0	0
O	[5000, ∞]	$[-2000, -1250)$	75599	1325	0	0	0	0	0
B	[500, 1200)	$[-1250, -750)$	23	2229	66	0	0	0	0
C	[500, 1200)	$[-1250, -750)$	48	3693	81	0	0	0	0
D	[500, 1200)	$[-1250, -750)$	1318	176114	5157	0	0	0	0
O	[500, 1200)	$[-1250, -750)$	1807	259491	8025	0	0	0	0
B	[1200, 3000)	$[-1250, -750)$	18	1490	55	0	0	0	0
C	[1200, 3000)	$[-1250, -750)$	52	5107	187	0	0	0	0

Continued on next page...

TABLE B-3. Continued

A	L	$h(t)$	$h(t+1)$						
			[-2000, -1250]	[-1250, -750]	[-750, -250]	[-250, 250]	[250, 750]	[750, 1250]	[1250, 2000]
D	[1200, 3000]	[-1250, -750]	494	58181	1782	0	0	0	0
O	[1200, 3000]	[-1250, -750]	2726	314132	10417	1	0	0	0
B	[3000, 5000]	[-1250, -750]	42	1445	82	0	0	0	0
C	[3000, 5000]	[-1250, -750]	15	865	39	0	0	0	0
D	[3000, 5000]	[-1250, -750]	1	0	1	0	0	0	0
O	[3000, 5000]	[-1250, -750]	1620	167085	4740	0	0	0	0
B	[5000, ∞]	[-1250, -750]	22	428	16	0	0	0	0
C	[5000, ∞]	[-1250, -750]	0	0	0	0	0	0	0
D	[5000, ∞]	[-1250, -750]	0	0	0	0	0	0	0
O	[5000, ∞]	[-1250, -750]	1345	111865	2811	1	0	0	0
B	[500, 1200]	[-750, -250]	0	100	15290	186	0	0	0
C	[500, 1200]	[-750, -250]	0	160	37490	500	0	0	0
D	[500, 1200]	[-750, -250]	0	7037	1712037	23271	0	0	0
O	[500, 1200]	[-750, -250]	2	10383	2919815	41529	0	0	0
B	[1200, 3000]	[-750, -250]	0	61	11823	217	0	0	0
C	[1200, 3000]	[-750, -250]	0	209	45446	639	0	0	0
D	[1200, 3000]	[-750, -250]	2	1997	388905	5752	0	0	0
O	[1200, 3000]	[-750, -250]	1	11662	2796178	41758	0	0	0
B	[3000, 5000]	[-750, -250]	0	91	9007	165	0	0	0
C	[3000, 5000]	[-750, -250]	0	41	4714	103	0	0	0
D	[3000, 5000]	[-750, -250]	0	1	30	1	0	0	0
O	[3000, 5000]	[-750, -250]	1	5082	1001108	12656	0	0	0
B	[5000, ∞]	[-750, -250]	0	23	1193	28	0	0	0
C	[5000, ∞]	[-750, -250]	0	0	0	0	0	0	0
D	[5000, ∞]	[-750, -250]	0	0	0	0	0	0	0
O	[5000, ∞]	[-750, -250]	2	2937	490952	4873	0	0	0
B	[500, 1200]	[-250, 250]	0	0	380	128943	265	0	0
C	[500, 1200]	[-250, 250]	0	0	1021	300731	632	0	0
D	[500, 1200]	[-250, 250]	0	0	39831	11486322	20842	0	0
O	[500, 1200]	[-250, 250]	0	0	66410	22520261	36293	1	0
B	[1200, 3000]	[-250, 250]	0	0	247	79669	179	0	0
C	[1200, 3000]	[-250, 250]	0	0	840	267771	482	0	0
D	[1200, 3000]	[-250, 250]	0	0	7390	2275618	4651	0	0
O	[1200, 3000]	[-250, 250]	0	0	52430	18977463	29588	1	0
B	[3000, 5000]	[-250, 250]	0	0	187	61715	161	0	0
C	[3000, 5000]	[-250, 250]	0	0	111	18816	85	0	0
D	[3000, 5000]	[-250, 250]	0	0	1	152	1	0	0
O	[3000, 5000]	[-250, 250]	0	1	14992	5695017	8068	0	0
B	[5000, ∞]	[-250, 250]	0	0	33	12535	36	0	0
C	[5000, ∞]	[-250, 250]	0	0	0	0	0	0	0
D	[5000, ∞]	[-250, 250]	0	0	0	0	0	0	0
O	[5000, ∞]	[-250, 250]	0	0	5706	2621051	2624	2	0
B	[500, 1200]	[250, 750]	0	0	0	479	23748	96	0
C	[500, 1200]	[250, 750]	0	0	0	1153	70327	147	0
D	[500, 1200]	[250, 750]	0	0	0	38448	1993067	8258	0
O	[500, 1200]	[250, 750]	0	0	0	59198	3089682	11222	0
B	[1200, 3000]	[250, 750]	0	0	0	202	12523	39	0
C	[1200, 3000]	[250, 750]	0	0	0	580	39857	177	0
D	[1200, 3000]	[250, 750]	0	0	0	6300	345205	1905	0
O	[1200, 3000]	[250, 750]	0	0	0	36232	1845196	8232	2
B	[3000, 5000]	[250, 750]	0	0	0	185	8620	63	0
C	[3000, 5000]	[250, 750]	0	0	0	96	2607	23	0
D	[3000, 5000]	[250, 750]	0	0	0	1	37	1	0
O	[3000, 5000]	[250, 750]	0	0	0	9269	462258	2288	2
B	[5000, ∞]	[250, 750]	0	0	0	42	2273	13	0
C	[5000, ∞]	[250, 750]	0	0	0	0	0	0	0
D	[5000, ∞]	[250, 750]	0	0	0	0	0	0	0
O	[5000, ∞]	[250, 750]	0	0	0	3037	137408	988	2
B	[500, 1200]	[750, 1250]	0	0	0	0	162	3778	25
C	[500, 1200]	[750, 1250]	0	0	0	0	229	7429	31
D	[500, 1200]	[750, 1250]	0	0	0	0	10921	304726	1171
O	[500, 1200]	[750, 1250]	0	0	0	1	13859	388637	1555
B	[1200, 3000]	[750, 1250]	0	0	0	0	40	1464	12
C	[1200, 3000]	[750, 1250]	0	0	0	0	188	6213	38
D	[1200, 3000]	[750, 1250]	0	0	0	0	2059	68013	462
O	[1200, 3000]	[750, 1250]	0	0	0	1	8705	256042	1900
B	[3000, 5000]	[750, 1250]	0	0	0	0	68	1059	26
C	[3000, 5000]	[750, 1250]	0	0	0	0	25	360	10
D	[3000, 5000]	[750, 1250]	0	0	0	0	1	9	1

Continued on next page...

TABLE B-3. Continued

A	L	$\dot{h}(t)$	$\dot{h}(t+1)$						
			[-2000, -1250)	[-1250, -750)	[-750, -250)	[-250, 250)	[250, 750)	[750, 1250)	[1250, 2000]
O	[3000, 5000)	[750, 1250)	0	0	0	0	2423	68127	717
B	[5000, ∞)	[750, 1250)	0	0	0	0	16	258	5
C	[5000, ∞)	[750, 1250)	0	0	0	0	0	0	0
D	[5000, ∞)	[750, 1250)	0	0	0	0	0	0	0
O	[5000, ∞)	[750, 1250)	0	0	0	1	1105	31246	544
B	[500, 1200)	[1250, 2000]	0	0	0	0	0	44	2282
C	[500, 1200)	[1250, 2000]	0	0	0	0	0	73	2046
D	[500, 1200)	[1250, 2000]	0	0	0	0	3	1694	42489
O	[500, 1200)	[1250, 2000]	0	0	0	0	3	2151	57621
B	[1200, 3000)	[1250, 2000]	0	0	0	0	0	12	556
C	[1200, 3000)	[1250, 2000]	0	0	0	0	0	36	1177
D	[1200, 3000)	[1250, 2000]	0	0	0	0	0	503	15598
O	[1200, 3000)	[1250, 2000]	0	0	0	0	2	1980	63193
B	[3000, 5000)	[1250, 2000]	0	0	0	0	0	29	986
C	[3000, 5000)	[1250, 2000]	0	0	0	0	0	10	482
D	[3000, 5000)	[1250, 2000]	0	0	0	0	0	1	15
O	[3000, 5000)	[1250, 2000]	0	0	0	0	1	733	27729
B	[5000, ∞)	[1250, 2000]	0	0	0	0	0	8	252
C	[5000, ∞)	[1250, 2000]	0	0	0	0	0	0	0
D	[5000, ∞)	[1250, 2000]	0	0	0	0	0	0	0
O	[5000, ∞)	[1250, 2000]	0	0	0	0	0	577	24900

We look at the row of interest and randomly assign a value to $\dot{h}(t+1)$ with probability proportional to the corresponding elements plus 1. We then assign $\dot{\psi}(t+1)$ to a random value conditional on A , $\dot{\psi}(t)$, L , and $\dot{h}(t+1)$. Finally, we assign $\dot{v}(t+1)$ based on the assignments of L , $\dot{v}(t)$, $\dot{h}(t+1)$, and $\dot{\psi}(t+1)$. The process is repeated for each time step. The length of the trajectory may be made as long as desired.

Sampling from the Bayesian networks produces a sequence of assignments of variables to bins, e.g. [250, 750). As described above, the next step is to sample uniformly within the bins to produce real values (with the exception of a bin spanning 0 in which case 0 itself is selected). If we simply sample within each bin at each time step, there would be excessive variability in the vertical rates, turn rates, and acceleration. We therefore only resample within bins at rates specified in the parameters file:

0 0 0 0.0211495 0.0412835 0.0885336

The first three variables have zero as their rates because they do not change during the course of the trajectory. The last three elements specify the rates with which $\dot{v}$, $\dot{h}$, and $\dot{\psi}$ change within bins. As the trajectory evolves, whenever a variable switches to a new bin we must sample within the bin. Typically, the values of variables continue in the same bin at the next time step. If the bin remains the same, we either continue with the same value or resample within the bin according to the specified rates.

Once the initial conditions and a series of control variables ($\dot{v}$, $\dot{h}$, and $\dot{\psi}$) have been selected, the aircraft trajectory is constructed or simulated using an appropriate dynamic model. The control variables are assumed to be held constant across each 1 s time step.

APPENDIX C

BAYESIAN NETWORKS

This appendix briefly reviews Bayesian networks. Further discussion of Bayesian networks may be found elsewhere [19–21].

C.1 DEFINITION

A Bayesian network is a graphical representation of a multivariate probability distribution over variables $\mathbf{X} = X_1, \dots, X_n$. 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 $P(x_i \mid \pi_i)$, where π_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:

$$P(\mathbf{x}) = P(x_1, \dots, x_n) = \prod_{i=1}^n P(x_i \mid \pi_i). \quad (\text{C-1})$$

C.2 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 [22].

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.

C.3 PARAMETER LEARNING

The parameters θ 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 j th instantiation of the parents of X_i is denoted π_{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

$$p(\boldsymbol{\theta} \mid D, G) = \frac{P(D \mid \boldsymbol{\theta}, G)p(\boldsymbol{\theta} \mid G)}{P(D \mid G)} = \frac{P(D \mid \boldsymbol{\theta}, G)p(\boldsymbol{\theta} \mid G)}{\int P(D \mid \boldsymbol{\theta}, G)p(\boldsymbol{\theta} \mid G) d\boldsymbol{\theta}}. \quad (\text{C-2})$$

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

$$P(D \mid \boldsymbol{\theta}) = \prod_{i=1}^n \prod_{j=1}^{q_i} \prod_{k=1}^{r_i} \theta_{ijk}^{N_{ijk}}. \quad (\text{C-3})$$

Let $\boldsymbol{\theta}_{ij} = (\theta_{ij1}, \dots, \theta_{ijr_i})$. 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

$$p(\boldsymbol{\theta} \mid G) = \prod_{i=1}^n \prod_{j=1}^{q_i} p(\boldsymbol{\theta}_{ij} \mid G). \quad (\text{C-4})$$

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 (see [21], Section 6.2.3). Hence,

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

where $\alpha_{ij1}, \dots, \alpha_{ijr_i}$ 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 [18] use $\alpha_{ijk} = 1$. Heckerman, Geiger, and Chickering [23] 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 $\alpha_{ijk} + N_{ijk}, \dots, \alpha_{ijk} + N_{ijk}$. Hence,

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

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

$$P(X_i = k \mid \boldsymbol{\pi}_{ij}, D, G) = \int \theta_{ijk} p(\boldsymbol{\theta}_{ij} \mid D, G) d\boldsymbol{\theta}_{ij} = \frac{\alpha_{ijk} + N_{ijk}}{\sum_{k'=1}^{r_i} (\alpha_{ijk'} + N_{ijk'})}. \quad (\text{C-5})$$

C.4 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,

$$P(G \mid D) \propto P(G)P(D \mid G) = P(G) \int P(D \mid \boldsymbol{\theta}, G)p(\boldsymbol{\theta} \mid G) d\boldsymbol{\theta}. \quad (\text{C-6})$$

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 [18] show how to evaluate the integral above, resulting in

$$P(G \mid D) = P(G) \prod_{i=1}^n \prod_{j=1}^{q_i} \frac{\Gamma(\alpha_{ij0})}{\Gamma(\alpha_{ij0} + N_{ij})} \prod_{k=1}^{r_i} \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})}, \quad (\text{C-7})$$

where $N_{ij} = \sum_{k=1}^r N_{ijk}$. Heckerman, Geiger, and Chickering [23] 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.

APPENDIX D

DENSITY PLOTS

The figures in this appendix show the density of aircraft squawking VFR in the radar data used to construct our model. Density is measured in number of aircraft per NM^3 . Shown in the plots are the base-10 logarithm of the density. The logarithm of the density was used because of the extreme concentrations in some areas of the country, particularly Florida, in and around Dallas and Houston, Texas, and southern California. Each pixel measures $1/6$ of a degree on a side in latitude and in longitude. Depending on latitude, the square mileage of a pixel may be between 64 NM^2 at the northernmost latitude to 92 NM^2 at the southernmost latitude. The highest density pixel is in Miami-Dade County and has a density of approximately $10^{-2.6}$ aircraft per NM^3 over all altitude layers between 500 ft AGL and FL180, with the highest density layer between 500 ft AGL and 1200 ft AGL.

It is important to emphasize that this density map is the observed density from our data set. Factors such as terrain and curvature of the earth prevent the observation of aircraft, particularly low flying aircraft such as those flying VFR. The locations of the sensors are indicated on the maps by magenta circles. It is no coincidence that the measured density in the lowest altitude layer, for example, tend to be concentrated around the locations of the sensors. In particular, coverage at low altitudes tends to be sparser in the western portion of the country. Coverage above 5000 ft AGL, however, is fairly good across the country.

Certain areas on the maps are worth attention. Due to the protected area around Washington DC, there are virtually no 1200 squawking aircraft at any altitude layer in that region. By contrast, there are is a large concentration in Florida and southern Arizona, especially near the two locations of the Embry-Riddle flight school in Daytona Beach, Florida and Prescott, Arizona. In addition, the effect of the “upside down wedding-cake” airspace class structure around major airports can be seen if one compares the heavy concentration near the pixel representing Atlanta and Dallas at lower altitude layers against the empty airspace in the same pixel at the highest altitude layer. Finally, it can be seen that the density near Denver at higher altitudes AGL is quite low; this may be attributed to the fact that an additional 5,000 ft above ground in that region puts the aircraft above 10,000 ft MSL.

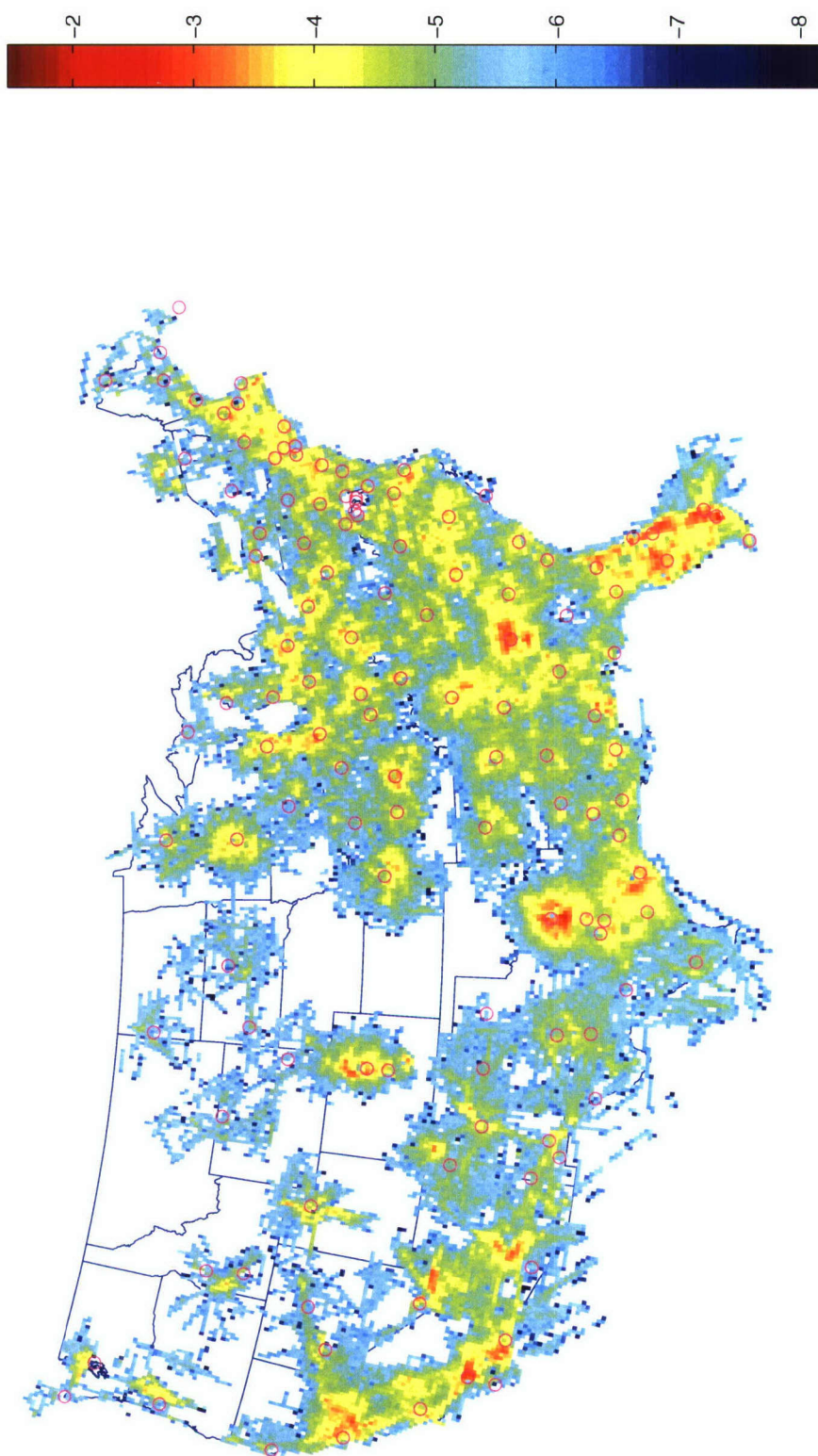


FIGURE D-1. VFR density over all altitude layers up to FL180 in $\log_{10}$ aircraft per NM^3 .

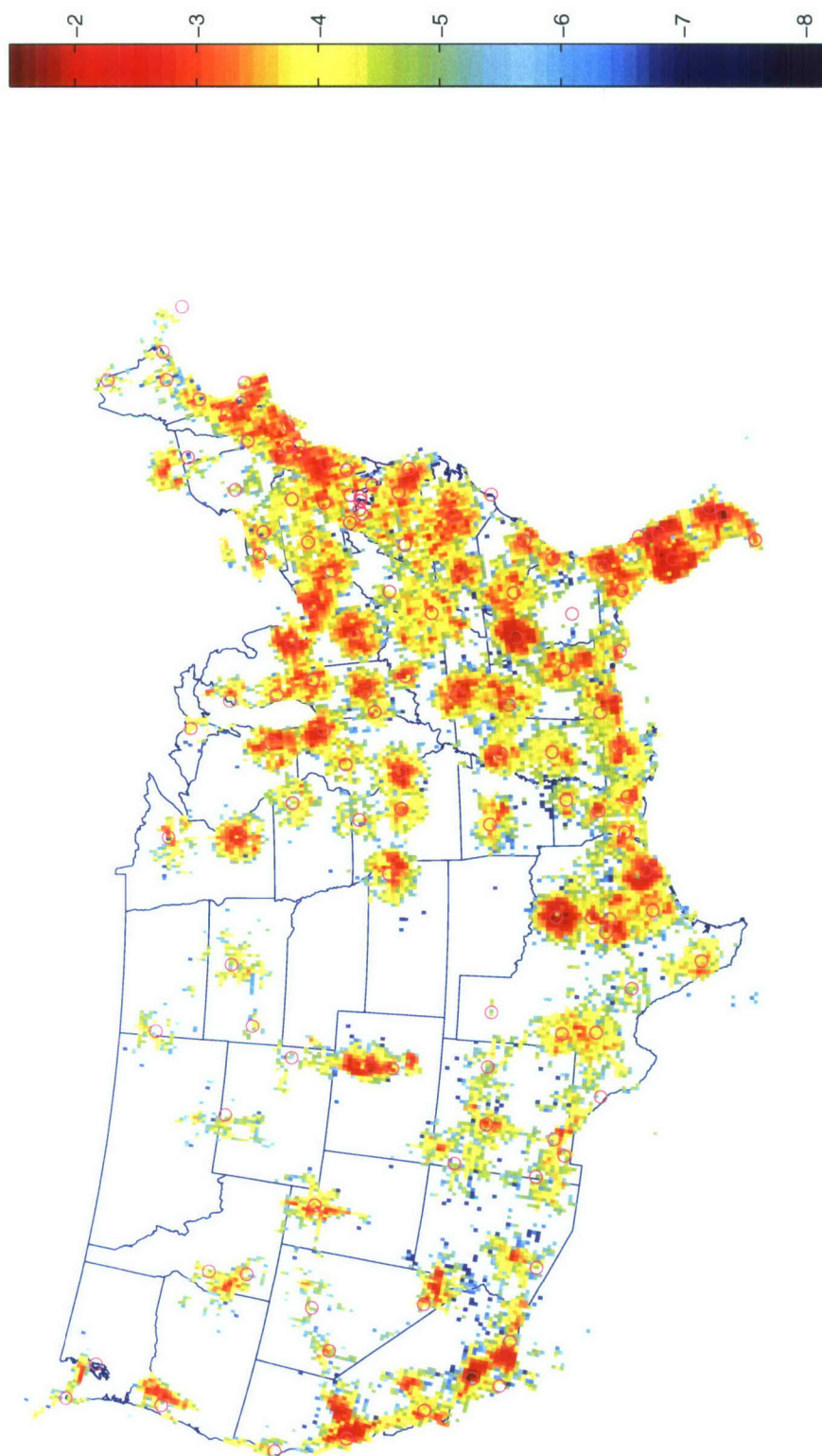


FIGURE D-2. VFR density between 500 and 1200 ft AGL in $\log_{10}$ aircraft per NM^3 .

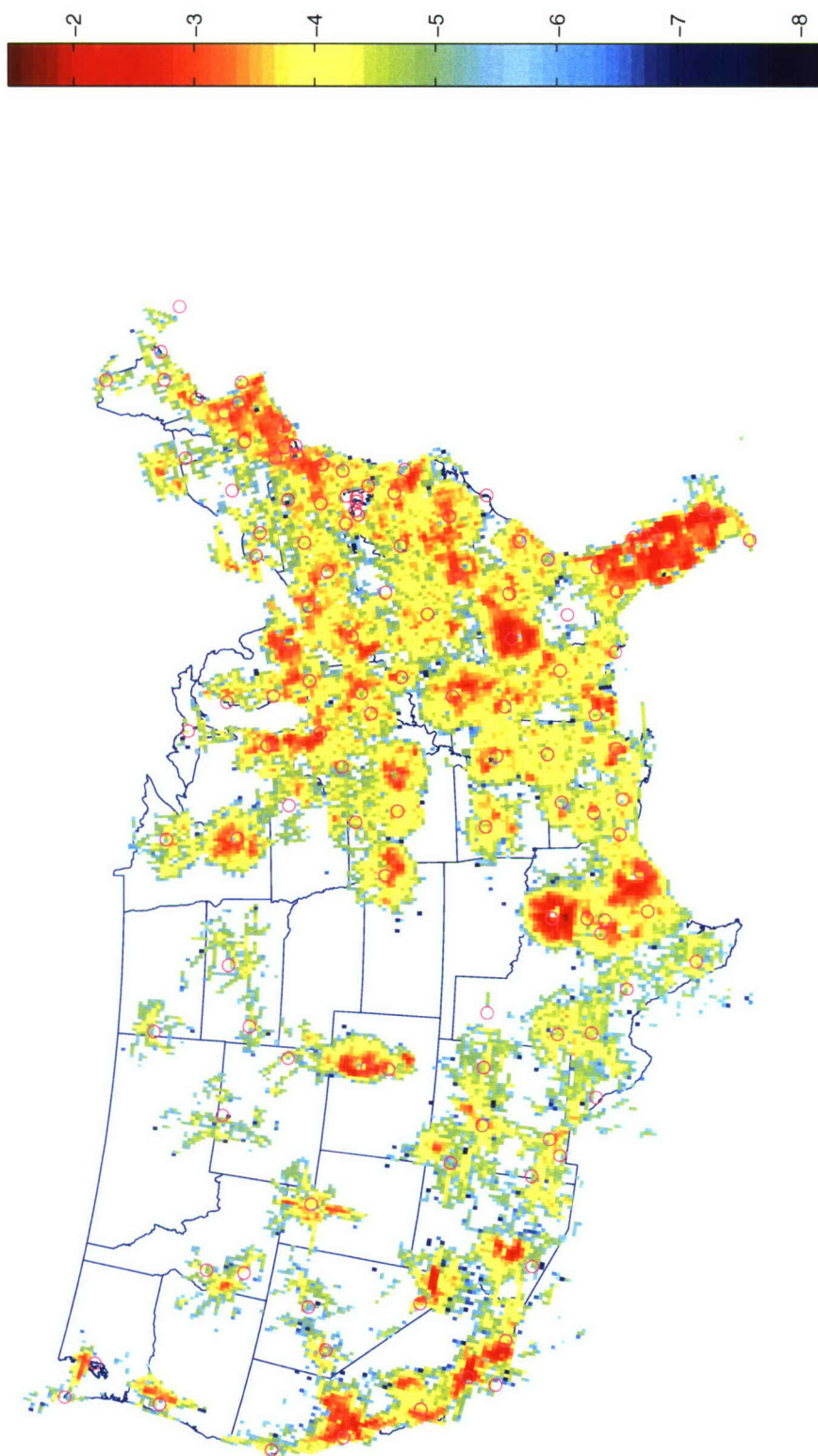


FIGURE D-3. VFR density between 1200 and 3000 ft AGL in log₁₀ aircraft per NM³.

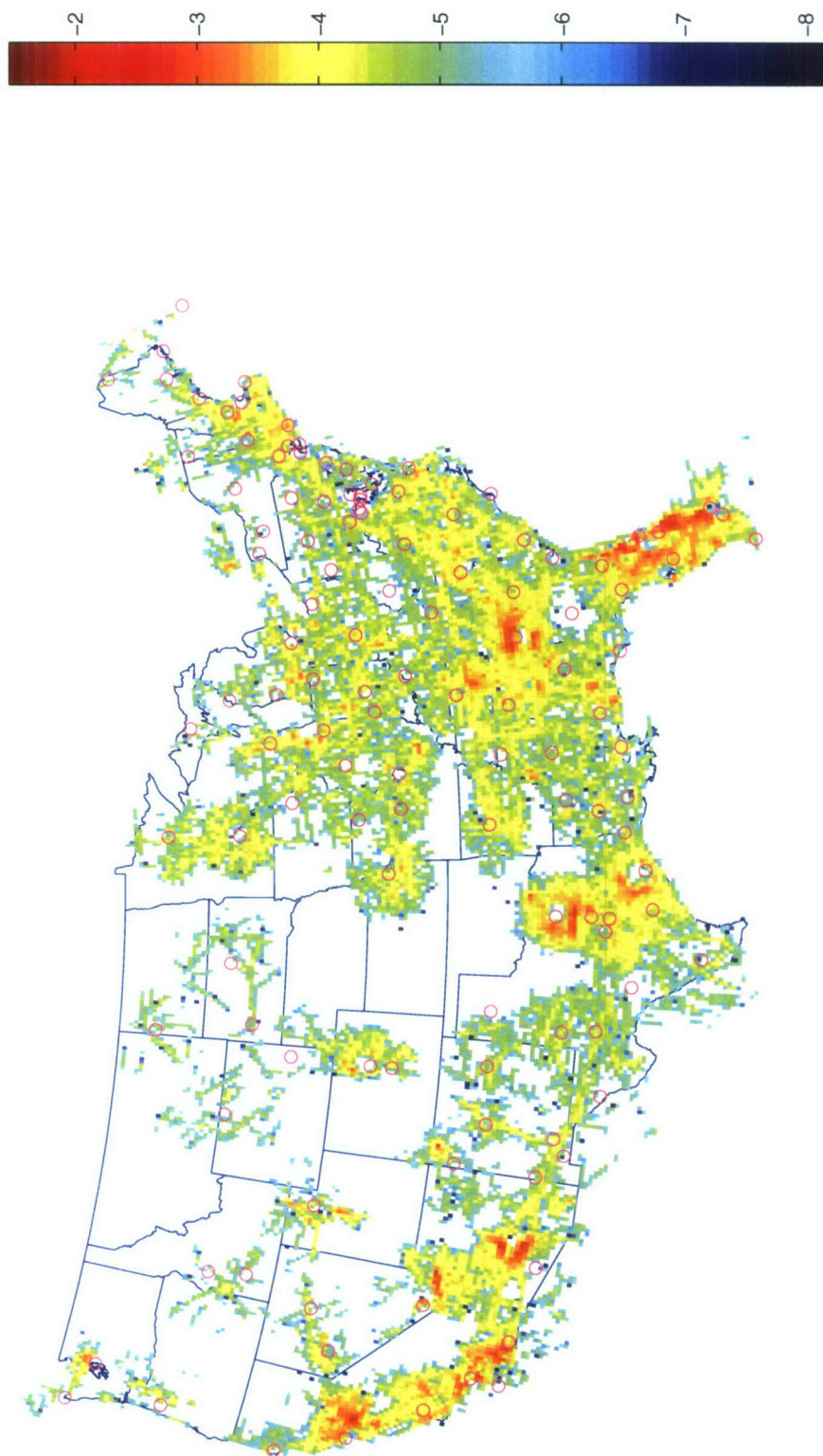


FIGURE D-4. VFR density between 3000 and 5000 ft AGL in $\log_{10}$ aircraft per NM^3 .

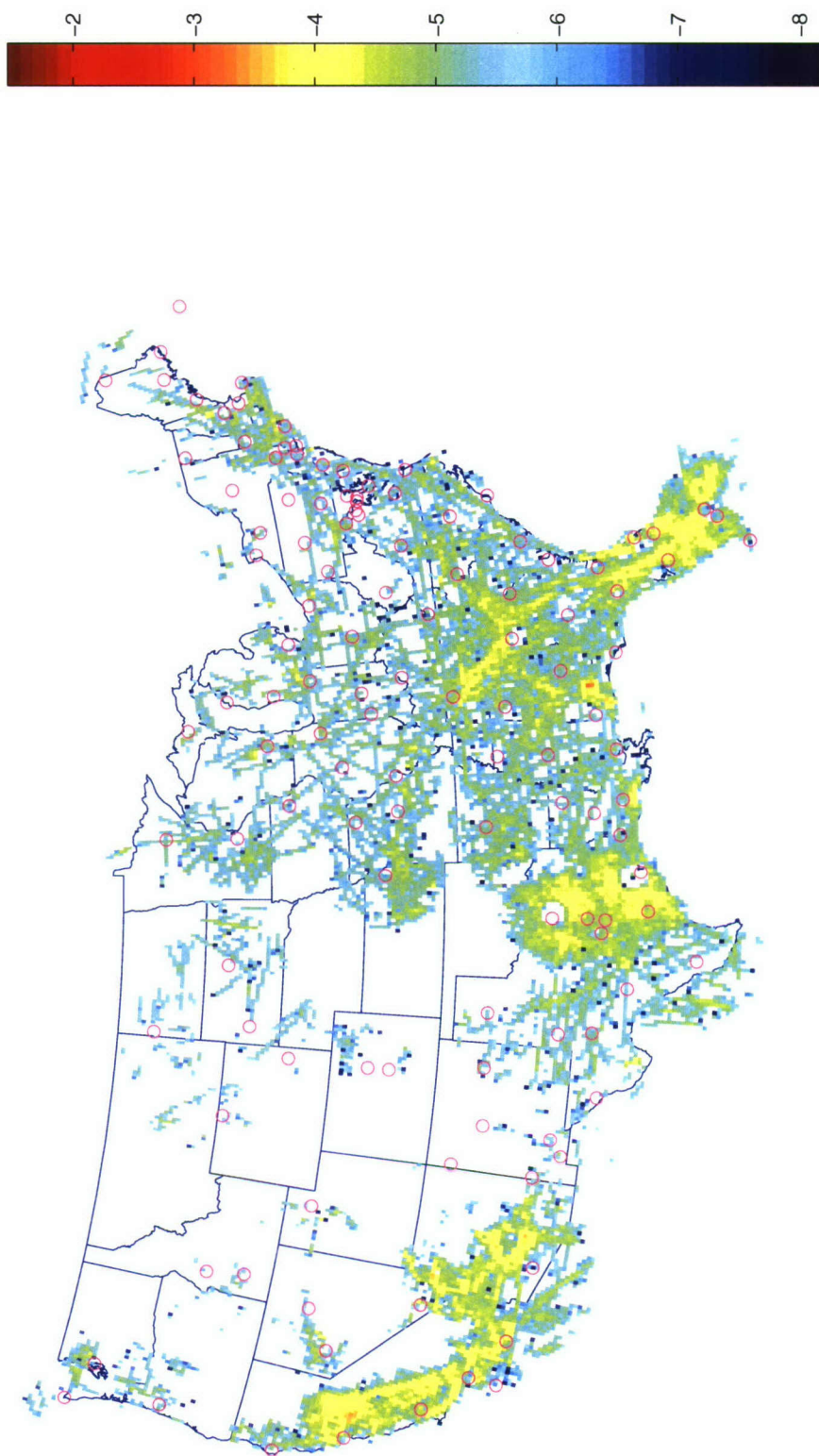


FIGURE D-5. VFR density between 5000 ft AGL and 10,000 ft MSL in $\log_{10}$ aircraft per NM^3 .

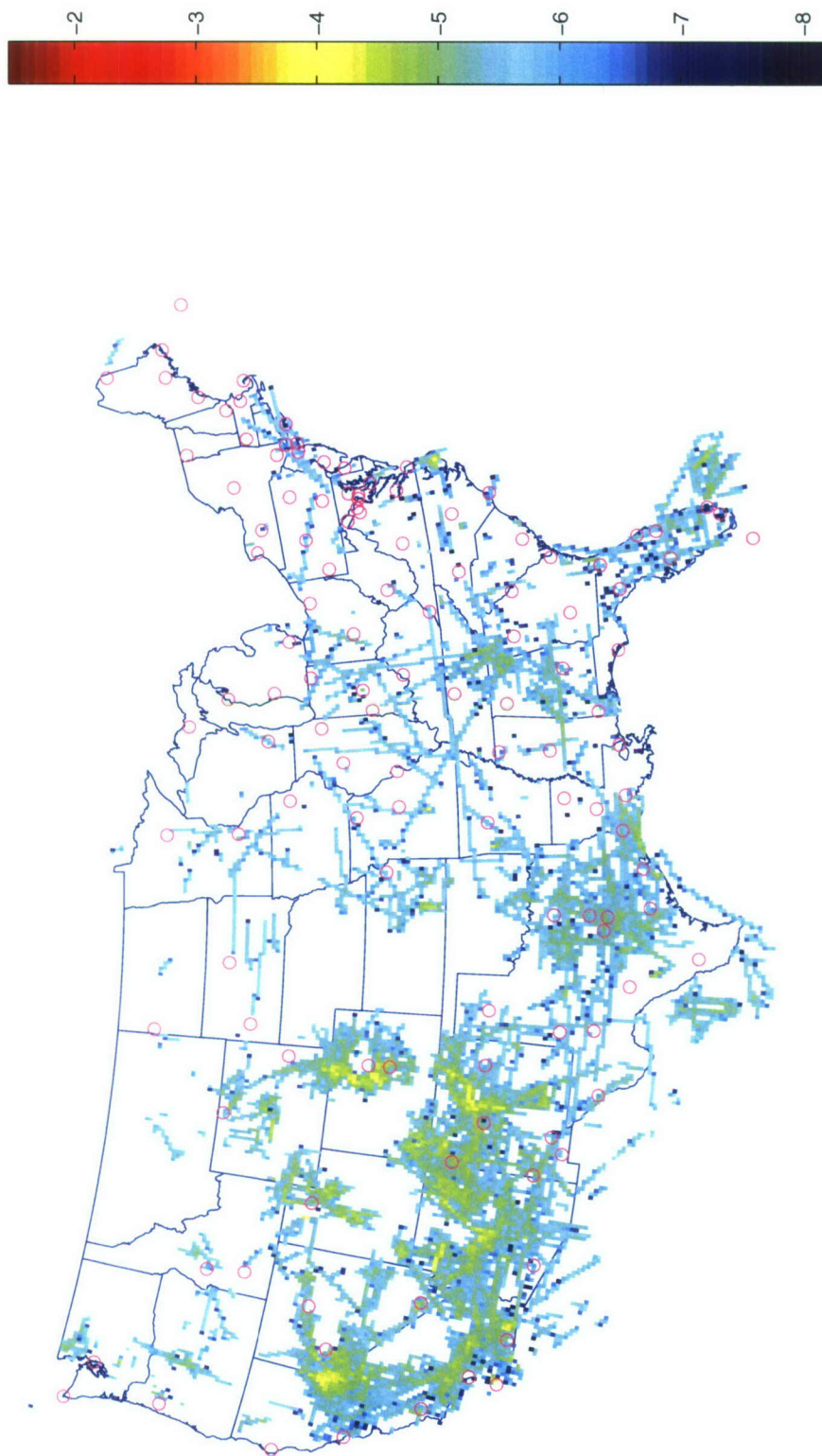


FIGURE D-6. VFR density between 10,000 ft MSL and FL180 in $\log_{10}$ aircraft per NM³.

APPENDIX E

SENSORS

The following table lists the RADES sensors that contributed to our model. Shown are the number of hours of flight time by transponder-equipped aircraft squawking 1200 captured by each sensor between December 1, 2007 and December 7, 2007.

TABLE E-1. Beacon Hours

60 NM range sensors	
628.19	ACT Waco, TX, USA (ASR-11)
82.64	ADW Camp Springs (Andrews AFB), MD, USA (ASR-9)
26.23	BGR Bangor, ME, USA (ASR-8)
426.58	BOS Logan Intl (Boston), MA, USA (ASR-9)
96.52	BUF Buffalo Intl, NY, USA (ASR-9)
291.51	BWI Baltimore (BWI), MD, USA (ASR-9)
295.06	COS Colorado Springs, CO, USA (ASR-11)
165.55	COU Columbia MO, MO, USA (ASR-11)
56.54	CRW Charleston, WV, USA (ASR-8)
72.57	DCA Washington National, DC, USA (ASR-9)
322.48	DOV Dover AFB, DE, USA (GPN-20)
795.56	EWR Newark, NJ, USA (ASR-9)
461.82	GRK Ft Hood, TX, USA (ASR-9)
977.69	HPN White Plains (Westchester Co), NY, USA (ASR-9)
218.21	HUF Terre Haute, IN, USA (ASR-8)
236.84	IAD Washington Dulles (Chantilly), DC, USA (ASR-9)
913.5	JFK New York (JFK), NY, USA (ASR-9)
1994.23	LAXB Los Angeles Intl - North, CA, USA (ASR-9)
247.43	LFT Lafayette, LA, USA (ASR-11)
454.24	MDT Harrisburg, PA, USA (ASR-9)
684.51	MHT Manchester, NH, USA (ASR-9)
183.77	MLU Monroe LA, LA, USA (ASR-8)
201.24	MRB Martinsburg, WV, USA (ASR-9)
67.78	MUO Mountain Home AFB, ID, USA (GPN-20)
67.23	NQXA Key West NAS, FL, USA (GPN-27)
206.52	PWM Portland (Cumberland), ME, USA (ASR-9)
14.02	RCA Ellsworth AFB (Rapid City), SD, USA (GPN-20)
224.52	SAV Savannah, GA, USA (ASR-8)
23.99	SAW Marquette (KI Sawyer), MI, USA (ASR-7F)
201.91	SDF Louisville, KY, USA (ASR-9)

Continued on next page...

TABLE E-1. Continued

229.89	■	SJU San Juan, PR, USA (ASR-8)
250.74	■	STT St. Thomas, PR, USA (ASR-8)
414.31	■	SWF Newburgh (Stewart), NY, USA (ASR-9)
200 NM range sensors		
289.27	■	AEX Alexandria, LA, USA (AN/FPS-20A)
2274.9	■	ATL Atlanta (Marietta), GA, USA (ARSR-1E)
92.82	■	BAM Battle Mountain, NV, USA (ARSR-2)
267.37	■	CPV Coopersville, MI, USA (AN/FPS-66A)
133.24	■	DSV Buffalo (Dansville), NY, USA (ARSR-1E)
113.89	■	FLX Fallon, NV, USA (AN/FPS-66A)
703.96	■	FPK Salt Lake City (Francis Peak), UT, USA (ARSR-1E)
2584.5	■	FTW Ft Worth (Keller), TX, USA (ARSR-1E)
403.56	■	GUP Gallup (Farmington), NM, USA (ARSR-2)
1335.22	■	HOU Houston (Ellington), TX, USA (ARSR-1E)
479.19	■	IND Indianapolis, IN, USA (ARSR-1E)
121.43	■	IRK Kirksville, MO, USA (ARSR-3)
744.49	■	JOL Elwood (Joliet), IL, USA (ARSR-3)
108.98	■	LSK Lusk, WY, USA (ARSR-2)
744.27	■	MGM Montgomery, AL, USA (ARSR-1E)
418.95	■	PIT Pittsburgh (Oakdale), PA, USA (AN/FPS-67B)
1605.12	■	QAS Las Vegas (Angel Peak), NV, USA (AN/FPS-20E)
1282.42	■	QBE Roanoke (Bedford), VA, USA (ARSR-3)
831.23	■	QBN Binns Hall, VA, USA (ARSR-3)
503.78	■	QBZ Oskaloosa, KS, USA (ARSR-2)
273.38	■	QCF Du Bois (Clearfield), PA, USA (ARSR-3)
199.97	■	QCK Boise (Cascade), ID, USA (ARSR-2)
605.39	■	QDB Cleveland (Brecksville), OH, USA (ARSR-1E)
595.5	■	QDT Detroit (Canton), MI, USA (ARSR-1E)
893.41	■	QHA Cummington, MA, USA (AN/FPS-67B)
201.11	■	QHB St. Albans, VT, USA (AN/FPS-67B)
511.27	■	QHZ Horicon, WI, USA (ARSR-2)
82.66	■	QJB Gettysburg, SD, USA (AN/FPS-67B)
501.35	■	QJE Minneapolis (Apple Valley), MN, USA (ARSR-1E)
154.15	■	QJO Arlington, IA, USA (ARSR-3)
461.99	■	QJQ San Juan (Pico del Este), PR, USA (AN/FPS-20E)
880.29	■	QNK Lincolnton, GA, USA (ARSR-3)
468.43	■	QNM Newport, MS, USA (ARSR-3)
751.66	■	QOJ Nashville (Joelton), TN, USA (ARSR-1E)
769.26	■	QPC Haleyville, AL, USA (AN/FPS-67B)

Continued on next page. . .

TABLE E-1. Continued

1119.91		QPK Denver (Parker), CO, USA (ARSR-1E)
903.24		QPL The Plains, VA, USA (ARSR-3)
1547.86		QRB Citronelle, AL, USA (ARSR-2)
346.34		QRC Benton, PA, USA (AN/FPS-67B)
681.5		QRI Lynch, KY, USA (ARSR-2)
823.63		QRL Raleigh (Benson), NC, USA (ARSR-1E)
548.93		QRM Charlotte (Maiden), NC, USA (ARSR-1E)
388.89		QSA Albuquerque (West Mesa), NM, USA (AN/FPS-66A)
97.39		QSI Lovell, WY, USA (ARSR-2)
510.95		QTZ La Grange, IN, USA (ARSR-1E)
221.78		QUZ Hanna City, IL, USA (AN/FPS-67B)
146.17		QWC Mesa Rica, NM, USA (ARSR-1E)
613.66		QWO London, OH, USA (ARSR-1E)
430.32		QXR Russellville, AR, USA (AN/FPS-64A)
334.63		QXS Odessa (Andrews), TX, USA (ARSR-1E)
373.29		QYB Byhalia (Memphis), MS, USA (ARSR-1E)
1401.67		QYS Rogers, TX, USA (ARSR-1E)
767.89		SEA Seattle (Ft Lawton), WA, USA (ARSR-1E)
735.82		STL St. Louis, MO, USA (ARSR-1E)
265.31		SVC Silver City, NM, USA (ARSR-2)
250 NM range sensors		
1320.68		AJO Ajo, AZ, USA (ARSR-4)
5.27		BAR Barrington, NS, Canada (AN/FPS-117)
1220.64		CTY Cross City, FL, USA (ARSR-4)
350.53		DMN Deming (Magdalena Peak), NM, USA (ARSR-4)
317.16		LCH Lake Charles, LA, USA (ARSR-4)
3833.04		MLB Melbourne, FL, USA (ARSR-4)
1560.6		NEN Jacksonville (White House Fl), FL, USA (ARSR-4)
566.36		NEW New Orleans (Slidell), LA, USA (ARSR-4)
432.44		NQX Key West, FL, USA (ARSR-4)
2632.79		NSD San Clemente Island, CA, USA (ARSR-4)
662.4		PAM Panama City (Tyndall), FL, USA (ARSR-4)
942.28		PRB Paso Robles, CA, USA (ARSR-4)
689.27		QEA North Truro, MA, USA (ARSR-4)
4235.19		QFN Ft Green, FL, USA (ARSR-4)
1321.28		QIE Gibbsboro, NJ, USA (ARSR-4)
112.12		QJA Empire, MI, USA (ARSR-4)
125.28		QJD Nashwauk, MN, USA (ARSR-4)
159.75		QKW Makah, WA, USA (ARSR-4)

Continued on next page...

TABLE E-1. Continued

3158.42		QM8 Tamiami, FL, USA (ARSR-4)
2649.29		QMV Mill Valley, CA, USA (ARSR-4)
692.15		QNA Morales, TX, USA (ARSR-4)
221.85		QNW Eagle Peak, TX, USA (ARSR-4)
457.99		QOM King Mountain, TX, USA (ARSR-4)
328.46		QRJ Charleston (Jedburg), SC, USA (ARSR-4)
1129.22		QRW Mt. Laguna, CA, USA (ARSR-4)
1173.37		QVH Riverhead, NY, USA (ARSR-4)
541.18		QVR Oceana, VA, USA (ARSR-4)
41.6		QWA Watford City, ND, USA (ARSR-4)
54.44		QXU Utica (Remsen), NY, USA (ARSR-4)
34.93		QYA Bucks Harbor, ME, USA (ARSR-4)
20.77		QYD Caribou, ME, USA (ARSR-4)
225.25		QZA Oilton, TX, USA (ARSR-4)
66.18		QZZ Rainbow Ridge, CA, USA (ARSR-4)
198.62		RSG Rocksprings, TX, USA (ARSR-4)
337.83		SLE Salem (Laurel Mountain), OR, USA (ARSR-4)

APPENDIX F

TRACKING AND FUSION

Converting raw radar reports into tracks that are usable for our model development is a two-stage process. The first stage involves forming local tracks from the reports associated with each sensor. The second stage involves fusing the local tracks from multiple sensors to form global tracks. This appendix provides a brief overview of tracking and fusion.

We use the tracking algorithms from the Mode S and ASR-9 systems [13], the two most modern sensors in the Air Traffic Control System. The beacon correlation algorithms come from Mode S and the primary radar correlation algorithms come from ASR-9 Processor Augmentation Card (9-PAC). Both systems are integrated to provide a consistent track, although for the purpose of this report we ignore primary only reports.

After the reports of each sensor have been correlated into local tracks, we can fuse the local tracks to provide a global picture of the airspace. Fusion performs the following functions:

1. Merge tracks from multiple sensors that correspond to the same aircraft into a single global track.
2. Compute the speed and heading of each global track to permit trajectory predictions.
3. Correct sensor tracking errors that would have led to split global tracks and thus false encounters.

We use a *track-to-track* fusion method, meaning that we track each sensor's individual reports and then we merge all of the tracks [24]. The main advantages of *track-to-track* fusion over *report-to-track* fusion, in which all reports are correlated directly to global tracks, are:

1. Bias independence for velocity determination and maneuver detection.
2. Removal of short update interval velocity anomalies.
3. Reduced likelihood of forming clutter tracks.
4. Reduced likelihood of introducing incorrect data points into the track due to correlation errors.

Track merging employs position, velocity, Mode A code, and altitude as matching attributes over the entire track. Track merging for tracks with discrete codes, which are unique within an area, employs large correlation boxes for each of the matching attributes. Other tracks (1200 code or radar-only) must pass more stringent position tests and velocity tests in order to be merged together.

Our fusion method works forward in time. Thus, there is often doubt in whether or not tracks from multiple radars are indeed the same track with just a few data points. In cases of

doubt, tentative matches are remembered that can be upgraded to a merge after more scans of information are obtained. Merges are checked each scan and can be undone if later found to be unsatisfactory. Aircraft code changes are also accommodated, although they must be verified by other sensors in the merge set of the global track before being accepted. If only one sensor reports a code change, it is assumed that the sensor had a track swap, and the merge situation is altered accordingly.

The remainder of this section explains how we add local sensor tracks to global tracks, how to estimate track velocity as part of the fusion process, and how to filter for encounters.

F.1 ADDING LOCAL SENSOR TRACKS TO GLOBAL TRACKS

In order to facilitate fusing of tracks from multiple sensors into global tracks, we break the continental United States into 20 NM by 20 NM bins. Every track is associated to a geographic bin. Whenever the fusion process receives a new local sensor track, or a later report for an as yet unfused local track, an attempt is made to fuse this track to an existing global track. This process is performed by comparing the new track to all neighboring global tracks. The neighboring global tracks for a local track include all global tracks in the same geographical bin as the local track plus global tracks in the surrounding bins (totalling 9 bins). Several tests must be passed for the successful fusion of the single track to a global track:

1. The global track must not already be fused to another local track from the new track's sensor.
2. The tracks must agree on Mode A code (primary-only tracks automatically pass).
3. If the code agreement was on a discrete code, a very coarse horizontal positional test must be passed.
4. If the code agreement was on 1200 code, or no codes, a tighter positional test must be passed, as well as altitude and velocity tests. However, if only the velocity test fails, a potential fusion is declared; 3 successive potential fusions with the same global track results in a successful fusion.

If more than one possible global track satisfies the fusion tests, the one with the highest matching score is chosen. Existing fusion matches are checked each time a new report is received. If the tests fail for three scans in a row, the fusion is ended, and a new global track is sought for the local sensor track.

F.1.1 Code Matching Test

Normally, the code of the new track is the same as the code of the global track. However, code mismatching can complicate the fusion process. Code declaration errors due to data corruption, missing codes, and code changes due to controller action are all common. For this reason, associated with each global track is an established code and an alternate code, which is the code of the most recent report. Usually these two codes are the same. When an alternate code is different from

the established code for three successive reports, however, the established code is updated to the alternative code. Reports with no beacon code are ignored in this process. However, if a track has never had been associated with a beacon code report, we consider this track to be a primary-only and give the track an established code of 0.

If both the local track the global track have a beacon code, then a successful code match is declared if any of the following statements are true:

1. The established codes match.
2. The alternate codes match.
3. One of the established codes and the other alternate code match.

However, failure is declared if the match is on code 0, and both tracks have a beacon code in the other code slot that do not match. Lastly, if one track is radar only, and the other track has a beacon code, failure is declared.

To handle local track code changes due to a track swap in the single sensor tracker, confirmation of the code change by the global track is required. If at the time of the local track code change, the global track has had an update by a different sensor's track with the old beacon code, a track swap is declared, and the local track is removed from fusion with the global track. The local track then undergoes a new fusion process.

F.1.2 Horizontal Position Matching Test

Horizontal positional matching requires agreement between the global track's most recent horizontal position x_g, y_g and the new local track's horizontal position x_l, y_l projected back to the time of the global track. This test is simple if both track's reports contained altitude. If a radar report contains altitude h , range ρ , azimuth θ , then the track's horizontal position x, y can be empirically determined.

If, however, the altitude of a track is unknown, then the tracks' altitude has to be assumed (or guessed) in order to derive the track's horizontal position. Simply guessing an altitude may produce a erroneous x, y position. In order to use a reasonable altitude value, we employ the following algorithm:

1. If only one track has known altitude, then we convert the other track's stored ρ, θ position to x, y using the first track's altitude.
2. If neither track has known altitude (which is always true for a primary-only match), we consider all altitudes from 0NM to 7NM at 1NM steps. We then use the altitude that produces the closest positional match between the two tracks. While a smaller step size may produce more accurate estimates, we found that 1NM is sufficient for fusing two tracks.

Horizontal positional agreement is declared if the horizontal distance between the two tracks

is less than an acceptable value:

$$\sqrt{(x_g - x_l)^2 + (y_g - y_l)^2} \leq \Delta r_{\max} + 3\sigma_{az}\rho_g + 3\sigma_{az}\rho_l, \quad (\text{F-1})$$

where we use $\Delta r_{\max} = 20 \text{ NM}$ for a discrete code match and $\Delta r_{\max} = 1 \text{ NM}$ for a 1200 code or radar match. The standard deviation for horizontal position error terms σ_{az} account for positional errors due to azimuth noise in radar measurements (which is the dominant source of horizontal position error). We model the standard deviation of the azimuth noise as 3 milliradians for the data format of our radar feed.

F.1.3 Altitude Matching Test

Altitude matching requires agreement between the local and global track altitudes when both are known. Two comparisons are tested; the success of either test results in a match. The comparison test is:

$$\begin{aligned} \Delta t_l &= t_g - t_l \\ \Delta h_l &= |h_g - h_l| \\ \Delta h_l &\leq \Delta h_{\max} \\ \frac{\Delta h_l}{\Delta t_l} &\leq \Delta \dot{h}_{\max} \end{aligned}$$

where we use $\Delta h_{\max} = 600 \text{ ft}$ and $\Delta \dot{h}_{\max} = 100 \text{ ft/s}$. Since the altitude of a track can significantly change between sequential reports, we test both the most recent and previous local track altitudes with the most recent global track altitude update. Only one of the local track altitudes is required to pass the test.

F.1.4 Velocity Matching Test

Velocity matching requires agreement between the two track headings ψ and speeds s according to the following tests:

$$\begin{aligned} |\psi_g - \psi_l| &\leq \Delta \psi_{\max} \\ |s_g - s_l| &\leq \Delta s_{\max} \\ \frac{1}{2} &\leq \frac{s_g}{s_l} \leq 2 \end{aligned}$$

where we use $\Delta \psi_{\max} = 45^\circ$ and $\Delta s_{\max} = 100 \text{ kt}$. The last test is needed for slow aircraft and clutter tracks, to prevent, for example, speeds of 20 and 110 kt from agreeing.

F.2 DETERMINING TRACK AIRSPEED AND HEADING

Determining a global track's airspeed and heading is a two step process. First, the individual sensor tracks are smoothed. Second, the individual track are averaged using relative weights that

account for sensor update times and the quality of each sensor's measurement. We apply both alpha smoothing and curve fitting to determine airspeed and heading, depending upon the track situation. We have implemented various maneuver detection algorithms, and tracking is dependent upon the current turn rate and acceleration states of the track.

F.2.1 Local Track Smoothing

First, we require that the track has moved a minimum distance for it to be considered. If the track never moves more than 1 NM, then the track is thrown out. After the movement test is satisfied, the track's airspeed and heading are calculated from the new and previous positions. We then update the local track's airspeed and heading estimates using alpha smoothing.

First, the current heading estimate $\psi^{(j)}$, and its difference from the previous estimate $\psi^{(j-1)}$, are given by

$$\begin{aligned}\psi^{(j)} &= \text{atan2}\left((x^{(j)} - x^{(j-1)}), (y^{(j)} - y^{(j-1)})\right) \\ \Delta\psi^{(j)} &= \psi^{(j)} - \psi^{(j-1)}\end{aligned}$$

Next, we determine the current turn rate state S_ψ of the track:

$$S_\psi^{(n)} = \begin{cases} 2 & \text{if } \Delta\psi^{(j)} > \sigma_{\text{heading}} \\ 1 & \text{if } \Delta\psi^{(j)} > \Delta\psi_{\min} \\ -2 & \text{if } \Delta\psi^{(j)} < -\sigma_{\text{heading}} \\ -1 & \text{if } \Delta\psi^{(j)} < -\Delta\psi_{\min} \\ 0 & \text{otherwise} \end{cases} \quad (\text{F-2})$$

where we use $\Delta\psi_{\min} = 3^\circ$ and σ_{heading} is the standard deviation of the heading noise, which is calculated from the standard deviations for range and azimuth noise of the sensors. Note that a positive $\Delta\psi$ value corresponds to a right turn, while a negative value corresponds to a left turn. We then use $S_\psi^{(j)}$ to determine the smoothing value alpha α in Table F-1 of the individual tracks that will be used to calculate the heading of the global track at the current time.

The new track heading is finally given by:

$$\psi^{(j)} = \psi^{(j-1)} + \alpha \times \Delta\psi^{(j)}$$

and we iterate through this process for the entire track.

The process to estimate airspeed s is similar, with one important difference. If successive positions are simply connected, then the airspeed estimates will always be too high, since the aircraft will appear to “zig-zag” along the track. Thus, only the projection of the velocity vector

TABLE F-1. Smoothing values depending on the current and previous turn states.

Previous State	Current Turn State				
	Large Left Turn (-2)	Small Left Turn (-1)	No Turn (0)	Small Right Turn (+1)	Large Right Turn (+2)
Large Left (-2)	0.7	0.7	0.4	0.5	0.5
Small Left (-1)	0.7	0.4	0.4	0.5	0.5
No Turn (0)	0.4	0.4	0.3	0.4	0.4
Small Right (+1)	0.5	0.5	0.4	0.4	0.7
Large Right (+2)	0.5	0.5	0.4	0.7	0.7

onto the track's heading vector is used to determine the track's airspeed:

$$s^{(j)} = \cos\left(\frac{\Delta\psi^{(j)}}{2}\right) \times \sqrt{\frac{(x^{(j)} - x^{(j-1)})^2 + (y^{(j)} - y^{(j-1)})^2}{t^{(j)} - t^{(j-1)}}}$$

$$\Delta s^{(j)} = s^{(j)} - s^{(j-1)}$$

We then determine the current airspeed acceleration state S_s of the track using a similar technique as we did for turn rate.

$$S_s^{(j)} = \begin{cases} 2 & \text{if } \Delta s^{(j)} > \sigma_{\text{speed}} \\ 1 & \text{if } \Delta s^{(j)} > \Delta s_{\min} \\ -2 & \text{if } \Delta s^{(j)} < -\sigma_{\text{heading}} \\ -1 & \text{if } \Delta s^{(j)} < -\Delta s_{\min} \\ 0 & \text{otherwise} \end{cases}$$

where we use $\Delta s_{\min} = 18\text{kt}$ and σ_{speed} is the standard deviation of airspeed error due to noise in range and azimuth measurements from the radar sensors. The speed smoothing and the speed alpha table rules are the same as for the heading case.

F.2.2 Global Track Smoothing

In order to determine a global track's airspeed and heading at each measurement, we use a weighted least squares estimation approach. In this section we describe in detail the approach for determining the tracks heading.

First, each sensor's heading estimate is assigned a weight w_i at the current time $t^{(c)}$ as follows:

$$w_i^{(c)} = \sigma_{\text{heading}}^{-1} \times \frac{t_{\max} - \frac{t_i^{(j)} + t_i^{(j-1)}}{2} - t^{(c)}}{t_{\max}}$$

where σ_{heading} is the standard deviation of the heading noise and $t_{\text{max}} = 18\text{ s}$ is a discounting factor that takes into account the time difference between the measurement from the sensor being considered and the time for when we are determining heading. The time $t_i^{(j)}$ corresponds to the time of the next closest measurement for sensor i with respect to the current time that we are trying to determine the track's heading.

Next, we determine the total turn state score for the track

$$\left| \sum_{i=1}^N S_i \right|,$$

where N is the number of sensors supporting the track and S_i is the current turn state value for the i th sensor defined in Equation F-2. If the turn rate score is less than N , then we consider the track to be non-turning and the current global heading is simply the weighted average of the N sensor heading estimates:

$$\psi_{\text{global}}^{(c)} = \frac{\sum_{i=1}^N \psi_i^{(j)} \times w_i^{(j)}}{\sum_{i=1}^N w_i^{(c)}}$$

Otherwise, if the turn rate score is greater than or equal to N , then we consider the track to be in a turn. In this case, we utilize weighted least squares estimated to determine a first-order relationship between time and heading.

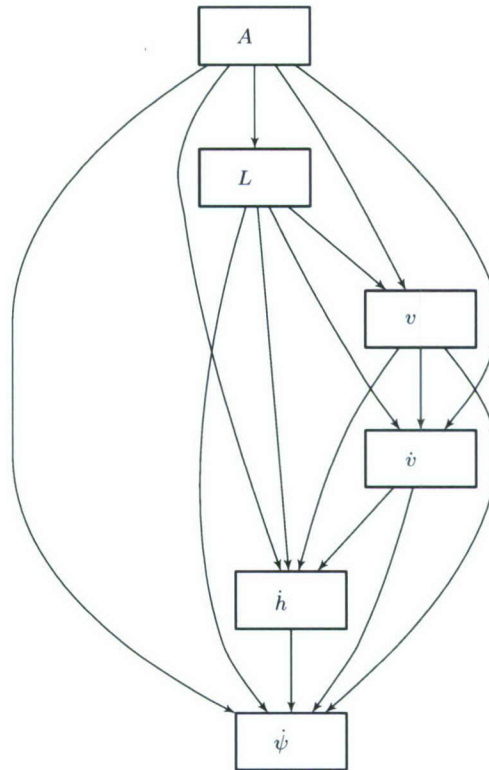
The global track speed calculation is identical in form to the global track heading calculation.

APPENDIX G

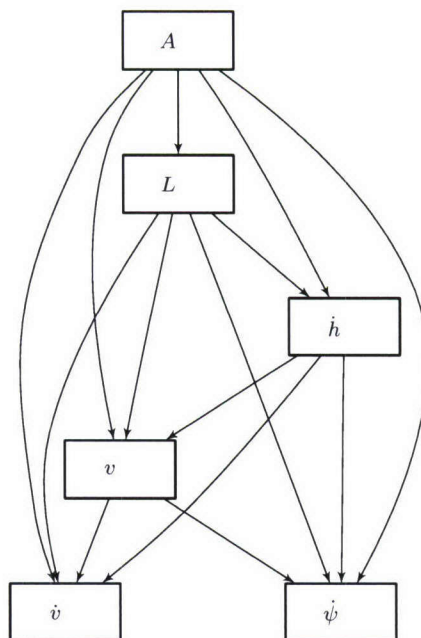
NETWORK CANDIDATES

This section displays a selection of various network structures along with their log-Bayesian score, number of edges, and number of parameters. Bayesian model selection balances the complexity of the model according to the amount of data available.

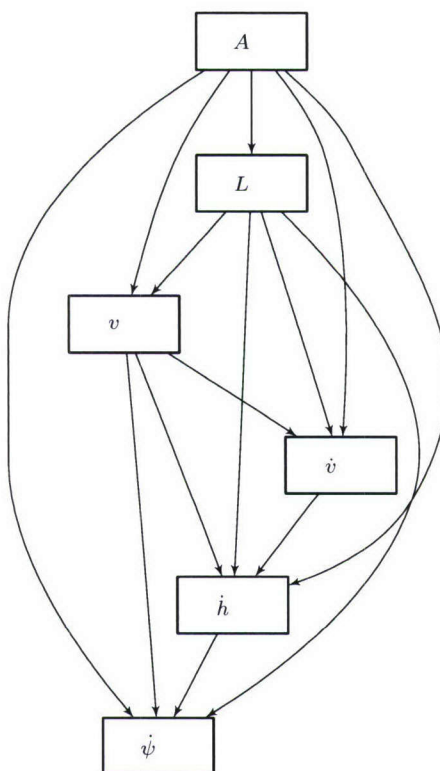
G.1 INITIAL NETWORK CANDIDATES



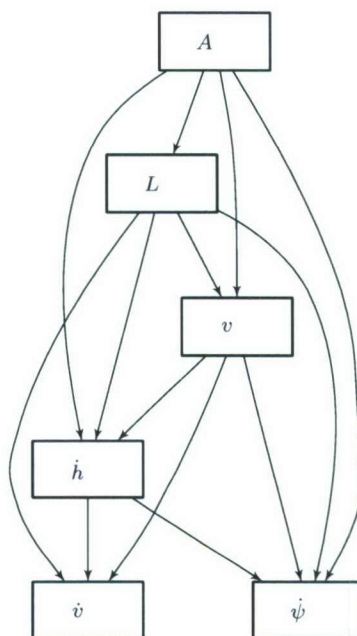
Score = -500006790 (best), Edges = 15, Parameters = 31359



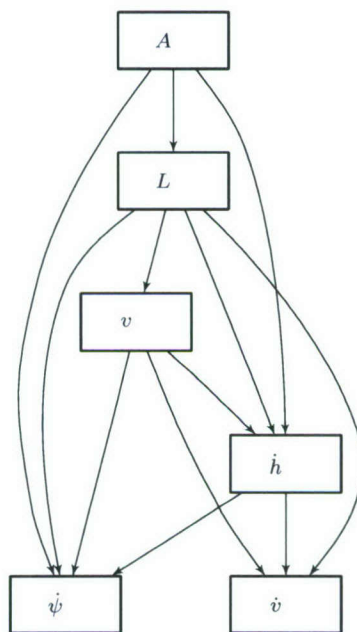
Score = -502636693, Edges = 14, Parameters = 9855



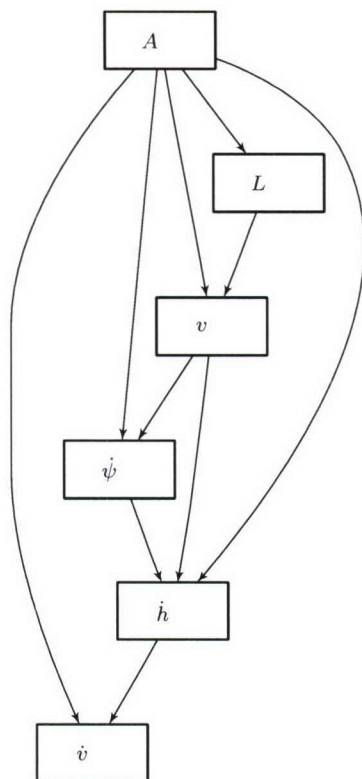
Score = -502638225, Edges = 14, Parameters = 9855



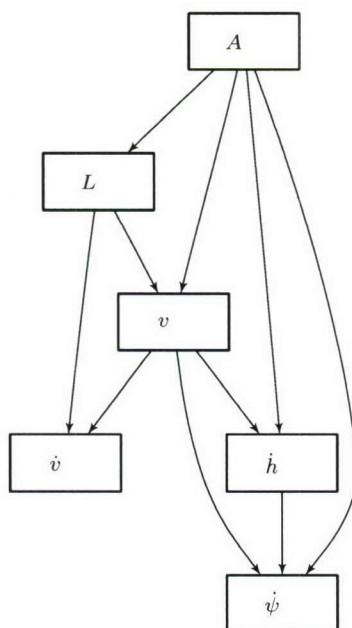
Score = -503170515, Edges = 13, Parameters = 7167



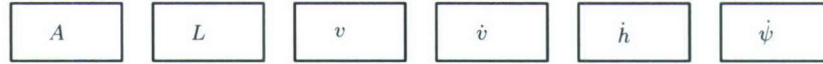
Score = -503711030, Edges = 12, Parameters = 7083



Score = -507340429, Edges = 10, Parameters = 1775

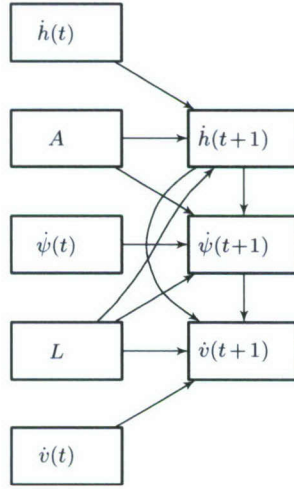


Score = -507958719, Edges = 10, Parameters = 1791

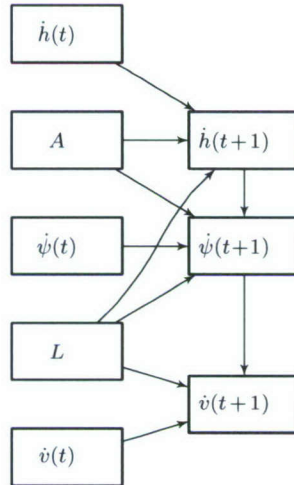


Score = -531540309, Edges = 0, Parameters = 29

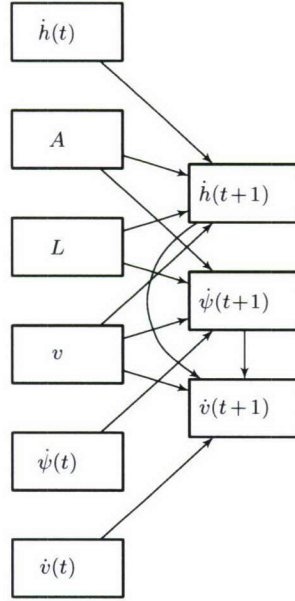
G.2 TRANSITION NETWORK CANDIDATES



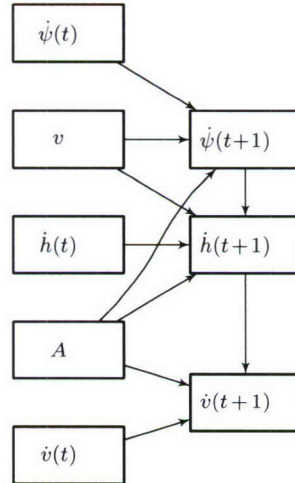
Score = -13243212 (best), Edges = 11, Parameters = 9296



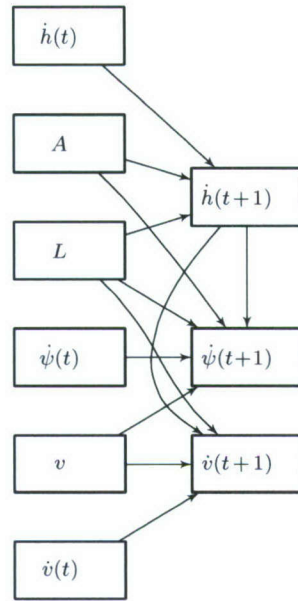
Score = -13266356, Edges = 10, Parameters = 5936



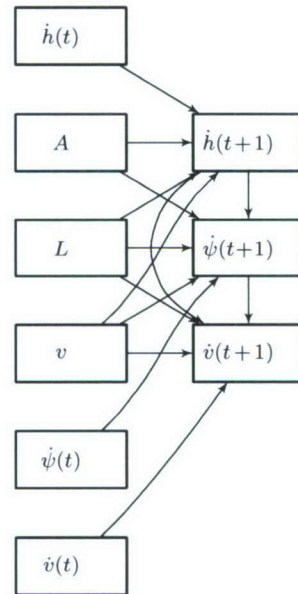
Score = -13276011, Edges = 12, Parameters = 18592



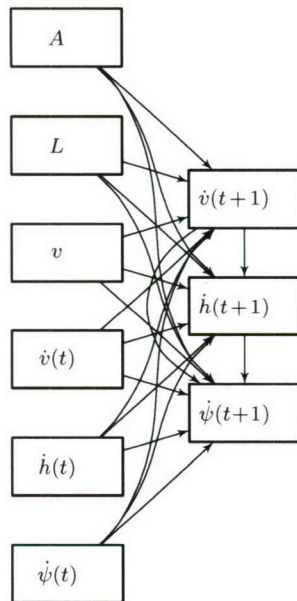
Score = -13277700, Edges = 10, Parameters = 11312



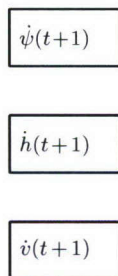
Score = -13340477, Edges = 12, Parameters = 42784



Score = -13380276, Edges = 14, Parameters = 74368



Score = -13942906, Edges = 21, Parameters = 7651840



Score = -180576482, Edges = 0, Parameters = 16

REFERENCES

- [1] MITRE, “System safety study of minimum TCAS II,” MITRE, Technical Rep. MTR-83W241 (1983).
- [2] A. Drumm, “Lincoln Laboratory evaluation of TCAS II Logic Version 6.04a,” MIT Lincoln Laboratory, Project Report ATC-240 (1996).
- [3] M.P. McLaughlin, “Safety study of the Traffic Alert and Collision Avoidance System (TCAS II),” MITRE Corporation, Technical Rep. MTR 97W32 (1997).
- [4] B. Chludzinski, “Lincoln Laboratory evaluation of TCAS II logic version 7,” MIT Lincoln Laboratory, Project Report ATC-268 (1999).
- [5] T. Arino, K. Carpenter, S. Chabert, H. Hutchinson, T. Miquel, B. Raynaud, K. Rigotti, and E. Vallauri, “Studies on the safety of ACAS II in Europe,” Eurocontrol, Technical Rep. ACASA/WP-1.8/210D (2002).
- [6] ICAO, “ACAS manual,” Eurocontrol, Technical Rep. SCRSP/1-WP/53 (2004).
- [7] T.A. Choyce and K.M. Ciaramella, “Test and evaluation of TCAS II logic version 7,” Federal Aviation Administration, Technical rep. (2000).
- [8] ICAO, “Surveillance, radar and collision avoidance,” in *ICAO Standards and Recommended Practices*, vol. IV, annex 10 (1998).
- [9] T. Miquel and K. Rigotti, “European encounter model,” CENA/Sofréavia and QinetiQ, Technical Rep. ACASA/WP1/186/D (2002).
- [10] S. Chabert, “Safety encounter model focused on issue SA01a,” CENA/Sofréavia and QinetiQ, Technical Rep. SIRE/WP2/21/D (2005).
- [11] T. Dean and K. Kanazawa, “A model for reasoning about persistence and causation,” *Computational Intelligence* 5(3), 142–150 (1989).
- [12] K. Murphy, *Dynamic Bayesian Networks: Representation, Inference and Learning*, Ph.D. thesis, University of California, Berkeley (2002).
- [13] R.D. Grappel, “ASR-9 Processor Augmentation Card (9-PAC) phase II scan-scan correlator algorithms,” MIT Lincoln Laboratory, Project Report ATC-298 (2001).
- [14] J.L. Gertz, “Mode S surveillance netting,” MIT Lincoln Laboratory, Project Report ATC-120 (1983).
- [15] F.N. Fritsch and R.E. Carlson, “Monotone piecewise cubic interpolation,” *SIAM Journal of Numerical Analysis* 17(2), 238–246 (1980).

- [16] E.M. Shank, “A coordinate conversion algorithm for multisensor data processing,” MIT Lincoln Laboratory, Project Report ATC-139 (1986).
- [17] W.M. Bolstad, *Introduction to Bayesian Statistics*, Wiley, 2nd ed. (2007).
- [18] G.F. Cooper and E. Herskovits, “A Bayesian method for the induction of probabilistic networks from data,” *Machine Learning* 9(4), 309–347 (1992).
- [19] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, San Francisco, CA: Morgan Kaufmann (1988).
- [20] F.V. Jensen, *Bayesian Networks and Decision Graphs*, Springer Verlag (2001).
- [21] R.E. Neapolitan, *Learning Bayesian Networks*, Upper Saddle River, NJ: Prentice Hall (2004).
- [22] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein, *Introduction to Algorithms*, MIT Press, 2nd ed. (2001).
- [23] D. Heckerman, D. Geiger, and D.M. Chickering, “Learning Bayesian networks: The combination of knowledge and statistical data,” *Machine Learning* 20(3), 197–243 (1995).
- [24] J.L. Gertz and R.D. Grappel, “Surveillance improvement algorithms for Airport Surface Detection Equipment Model X (ASDE-X) at Dallas-Fort Worth Airport,” MIT Lincoln Laboratory, Project Report ATC-333 (2007).

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.					
1. REPORT DATE 26 June 2008		2. REPORT TYPE Project Report		3. DATES COVERED (From - To)	
4. TITLE AND SUBTITLE Preliminary Uncorrelated Encounter Model of the National Airspace System				5a. CONTRACT NUMBER FA8721-05-C-0002	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) M. J. Kochenderfer, J. K. Kuchar, L. P. Espindle, and J. L. Gertz				5d. PROJECT NUMBER 105	
				5e. TASK NUMBER 6	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) MIT Lincoln Laboratory 244 Wood Street Lexington, MA 02420-9108				8. PERFORMING ORGANIZATION REPORT NUMBER PR-CASSATT-1	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Lt. Col. Kent Tiffany Wright-Patterson Air Force Base 2640 Loop Road West Wright-Patterson AFB, OH 45433-7106				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S) ESC-TR-2007-056	
12. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT Airspace encounter models, covering close encounter situations that may occur after standard separation assurance has been lost, are a critical component in the safety assessment of aviation procedures and collision avoidance systems. Of particular relevance to Unmanned Aircraft Systems is the potential for encountering general aviation aircraft that are flying under Visual Flight Rules (VFR) and which may not be in contact with air traffic control. In response to the need to develop a model of these types of encounters, Lincoln Laboratory undertook an extensive radar data collection and modeling effort. This report describes the structure and content of that encounter model.					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT Same as report	18. NUMBER OF PAGES 92	19a. NAME OF RESPONSIBLE PERSON
a. REPORT Unclassified	b. ABSTRACT Unclassified	c. THIS PAGE Unclassified			19b. TELEPHONE NUMBER (include area code)