

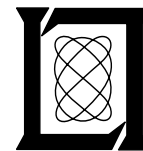
**Project Report
ATC-344**

**Correlated Encounter Model
for Cooperative Aircraft in the
National Airspace System
Version 1.0**

**M.J. Kochenderfer
L.P. Espindle
J.K. Kuchar
J.D. Griffith**

24 October 2008

Lincoln Laboratory
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LEXINGTON, MASSACHUSETTS



Prepared for the Federal Aviation Administration,
Washington, D.C. 20591

This document is available to the public through
the National Technical Information Service,
Springfield, Virginia 22161

This document is disseminated under the sponsorship of the Department of Transportation, Federal Aviation Administration, in the interest of information exchange. The United States Government assumes no liability for its contents or use thereof.

1. Report No. ATC-344	2. Government Accession No.	3. Recipient's Catalog No.	
4. Title and Subtitle Correlated Encounter Model for Cooperative Aircraft in the National Airspace System, Version 1.0		5. Report Date 24 October 2008	
		6. Performing Organization Code	
7. Author(s) Mykel J. Kochenderfer, Leo P. Espindle, James K. Kuchar, and J. Daniel Griffith		8. Performing Organization Report No. ATC-344	
9. Performing Organization Name and Address MIT Lincoln Laboratory 244 Wood Street Lexington, MA 02420-9108		10. Work Unit No. (TRAVIS)	
		11. Contract or Grant No. FA8721-05-C-0002	
12. Sponsoring Agency Name and Address Department of Transportation Federal Aviation Administration 800 Independence Ave., S.W. Washington, DC 20591		13. Type of Report and Period Covered Project Report	
		14. Sponsoring Agency Code	
15. Supplementary Notes This report is based on studies performed at Lincoln Laboratory, a center for research operated by Massachusetts Institute of Technology, under Air Force Contract FA8721-05-C-0002.			
16. Abstract This document describes a new cooperative aircraft encounter model for the National Airspace System (NAS). The model is used to generate random close encounters between transponder-equipped (cooperative) aircraft in fast-time Monte Carlo simulations to evaluate collision avoidance system concepts. An extensive set of radar data from across the United States, including more than 120 sensors and collected over a period of nine months, was used to build the statistical relationships in the model to ensure that the encounters that are generated are representative of actual events in the airspace.			
17. Key Words		18. Distribution Statement This document is available to the public through the National Technical Information Service, Springfield, VA 22161.	
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 126	22. Price

ABSTRACT

This document describes a new cooperative aircraft encounter model for the National Airspace System (NAS). The model is used to generate random close encounters between transponder-equipped (cooperative) aircraft in fast-time Monte Carlo simulations to evaluate collision avoidance system concepts. An extensive set of radar data from across the United States, including more than 120 sensors and collected over a period of nine months, was used to build the statistical relationships in the model to ensure that the encounters that are generated are representative of actual events in the airspace.

Several prior encounter models have been developed to model U.S. and European airspaces for the purpose of evaluating collision avoidance systems. Due to changes in the structure of the airspace and traffic characteristics, and due to the need to evaluate more complex situations, it has become necessary to completely re-define a new and more comprehensive encounter model based on recent radar data. This new encounter model extends prior models in several ways, most notably in its accommodation of multiple vertical and horizontal accelerations during an encounter. The model represents probability distributions over a range of encounter variables including approach angle, airspeed, and horizontal and vertical miss distance. Dynamic Bayesian networks were used to capture the variation of vertical rate and turn rate over the course of an encounter scenario. Bayesian model selection techniques were used to obtain the best network structure that represents the correlations observed between variables in the radar data that were collected.

This document describes and justifies the theoretical framework used to model cooperative encounters between aircraft. The report explains how various encounter parameters were extracted from radar data and Bayesian statistical methods were used to estimate a probability distribution over encounters. Finally, the document explains the sampling method used to generate encounter trajectories for Monte Carlo analyses to provide accurate estimates of collision risk.

The model described in this report is one of three developed by Lincoln Laboratory. This model has been developed to represent situations in which it is likely that there would be air traffic control intervention prior to a close encounter, and applies to cases in which both aircraft are cooperative and at least one is using a discrete (non-1200) transponder Mode A code. Other models have been developed for an encounter with an intruder that does not have a transponder, or between two aircraft using a Mode A code of 1200 (VFR).

Separate electronic files are available from Lincoln Laboratory that contain the statistical data required to generate and validate encounter trajectories. Details on how to interpret the data files and an example of how to randomly construct trajectories are provided in Appendices A and B. A software package is also available to generate random encounter trajectories based on the data tables.

A byproduct of the encounter modeling effort was the development of aircraft track and traffic density databases. Example plots of encounter density data are provided in this report, but the complete track and density databases are not provided in electronic form due to their size and the complexity of processing specific locations, altitudes, and times.

ACKNOWLEDGMENTS

This report is the result of research and development sponsored by the TCAS Program Office at the FAA. The authors greatly appreciate the support and assistance provided by Paul Fontaine, Manager of Future Surveillance, and Neal Suchy, TCAS Program Manager.

The authors greatly appreciate the support for access to radar data from Hill Air Force Base, including Jeff Richardson, Steven Schimmelpfennig, Richard Whitlock, Lt. Tanuxay Keooudom, Lt. Han Saydam, TSgt. Christopher Cosper, and James Evans.

The authors wish to thank Josh Silbermann, Hui Men, and Zipora Sidel (Johns Hopkins Applied Physics Laboratory); Ken Carpenter (QinetiQ); Thierry Arino (Directorate of Air Navigation Services, France); John Law and Garfield Dean (Eurocontrol); and Andy Zeitlin and Michael McLaughlin (MITRE CAASD) for their valuable technical input.

Finally, the authors would also like to thank Lincoln staff members Vito Cavallo, James Chryssanthacopoulos, Ann Drumm, Jeffrey Gertz, Robert Grappel, Garrett Harris, Jessica Olszta, John O'Rourke, Angela Sharer, and Steven Thompson for their contributions to the airspace encounter model effort.

This page intentionally left blank.

TABLE OF CONTENTS

	Page
Abstract	iii
Acknowledgments	v
List of Illustrations	xi
List of Tables	xiii
1. INTRODUCTION	1
1.1 Encounter Types	2
1.2 Radar Data	4
1.3 Process Overview	6
2. APPROACH	9
2.1 Encounter Definition	9
2.2 Model Variables	10
2.3 Initial Distribution	13
2.4 Transition Distribution	14
3. ESTIMATION	19
3.1 Encounter Filtering	19
3.2 Outlier Removal	21
3.3 Track Smoothing	21
3.4 Interpolation	24
3.5 Feature Extraction	24
3.6 Statistics Extraction	27
4. SAMPLING	33
4.1 Discrete Sampling	33
4.2 Continuous Sampling	34
4.3 Model Validation	35
4.4 Sampling When One Trajectory is Specified	37

4.5	Sampling from Alternative Distributions	38
5.	ENCOUNTER INITIALIZATION	41
5.1	Initial Aircraft Positions	41
5.2	Slow Closing Encounters	44
6.	SAFETY EVALUATION TECHNIQUES	49
6.1	Safety Metrics	49
6.2	Importance Sampling	50
7.	SUMMARY	53
A.	MODEL PARAMETERS	55
B.	ENCOUNTER GENERATION	59
B.1	Initial Network Sampling	59
B.2	Transition Network Sampling	62
C.	VALIDATION	67
D.	ENCOUNTER FILTER	69
D.1	Proximity Test	69
D.2	TCAS-like Test	69
D.3	Justification	69
D.4	Validation	71
E.	PARALLEL RUNWAY OPERATION AND FORMATION FLIGHT IDENTIFICATION	73
E.1	Parallel Runway Operation Test	73
E.2	Formation Flight Test	74

F.	TRACKING AND FUSION	75
F.1	Adding Local Sensor Tracks to Global Tracks	76
F.2	Determining Track Airspeed and Heading	78
F.3	Cooperative Encounter Identification	81
G.	SENSOR LIST	83
H.	ENCOUNTER DENSITY	87
I.	FEATURE HISTOGRAMS	95
J.	BAYESIAN NETWORKS	97
J.1	Definition	97
J.2	Sampling	97
J.3	Parameter Learning	97
J.4	Structure Learning	99
K.	NETWORK CANDIDATES	101
K.1	Initial Network Candidates	101
K.2	Transition Network Candidates	105
	References	109

This page intentionally left blank.

LIST OF ILLUSTRATIONS

Figure No.		Page
1	Radar coverage map.	4
2	Cumulative flight hours.	5
3	Modeling and simulation process overview.	7
4	Airspace layers.	12
5	Approach angle (β) and bearing (χ) definition.	12
6	Bayesian network representing the variable dependency structure for the initial distribution.	15
7	Bayesian network representing the variable dependency structure for the transition distribution.	16
8	Estimation process flow.	20
9	Filtering processes flow and results.	22
10	Encounter density over all altitude layers.	23
11	Preprocessing.	24
12	Piecewise-cubic Hermite interpolation on a smoothed track.	25
13	Feature histograms of observed radar data.	28
14	A plot of extracted features over time.	29
15	Convergence curve for Bayesian network conditional probabilities.	30
16	Sampling process flow.	33
17	A plot of sampled features over time.	36
18	An encounter generated by sampling from the initial and transition distributions.	36
19	Encounter generation flow.	42
20	Encounter geometry at TCA.	43
21	Example steps to initialize aircraft.	45
22	An example slow closing encounter generated from the encounter model.	47
23	Joint probability distributions for hmd and vmd .	52

B-1	A graphical representation of the initial network.	60
B-2	A graphical representation of the transition network.	65
H-1	Encounters over all altitude layers.	88
H-2	Encounters between 1000 and 3000 ft AGL.	89
H-3	Encounters between 3000 ft AGL and 10,000 ft MSL.	90
H-4	Encounters between 10,000 ft MSL and FL180.	91
H-5	Encounters between FL180 and FL290.	92
H-6	Encounters above FL290.	93
I-1	Feature histograms of observed encounters.	95
I-2	Feature histograms of observed encounters.	96

LIST OF TABLES

Table No.		Page
1	Encounter model types.	3
2	Cut points used for feature quantization.	27
3	Sufficient statistics for airspace class given altitude layer.	28
4	Sampling boundaries.	35
B-1	Sufficient statistics for bearing.	61
B-2	Sufficient statistics for altitude layer.	61
B-3	Sufficient statistics for airspace class given altitude layer.	62
B-4	Sufficient statistics for vertical rate in dynamic Bayesian network.	63
D-1	TCAS-like test parameters.	70
F-1	Smoothing values depending on the current and previous turn states.	80
G-1	Beacon hours.	83

This page intentionally left blank.

1. INTRODUCTION

This document describes a new cooperative encounter model for the National Airspace System (NAS) sponsored by the Federal Aviation Administration (FAA).

Collision avoidance systems are important components contributing to the safety of the NAS. 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. Each encounter generated from the model specifies the initial positions and orientations of two aircraft and the nominal dynamic maneuvers (not affected by a collision avoidance system) that may occur leading up to the closest point of approach. What is produced, then, is the geometric, dynamic situation a collision avoidance system may be expected to resolve safely. Identical encounter situations are typically simulated with and without a collision avoidance system so that relative system benefits may be quantified. Knowledge of the rates at which encounter situations occur in the NAS can also be used to estimate the rate of near mid-air collisions per flight hour and thus arrive at an overall risk metric.

In these models, encounter situations are abstracted in the sense that there is no consideration of an explicit location or local airspace structure (e.g., airways, metering fixes, approach paths). Rather, the encounters represent what may be statistically expected to occur over the lifetime of a given system. If desired, a particular altitude layer or airspace class (e.g., Class B) can be specified. Additionally, the flight path of one aircraft can be constrained to focus, for instance, on a particular departure profile or flight condition. It should also be noted that the models cover approximately 1 minute before closest point of approach and so are not appropriate for large-scale air traffic impact studies (e.g., to examine sector loading or conflict rates); the focus here is on events in which loss of separation has already occurred between two aircraft and collision avoidance becomes paramount.

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 United States. The model structure is based, in part, on prior efforts of Eurocontrol. However, that modeling approach has been extended in several ways, most notably to accommodate multiple vertical and horizontal accelerations during an encounter and to use a Bayesian network to capture important correlations between variables. The advantage of applying a Bayesian framework is that it optimally leverages available radar data to produce a model that is the most representative of actual operations.

A byproduct of the encounter modeling effort was the development of aircraft track and encounter density databases. Example plots of encounter density data are provided in this report, but the complete track and density databases are not provided in electronic form due to their size and the complexity of processing specific locations, altitudes, and times.

1.1 ENCOUNTER TYPES

The encounters covered by this model are those involving aircraft in the final stages before a collision. 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 other separation aspects such as ATC communications or coordination.

Because they are by far the most likely to occur, only pairwise (two-aircraft) encounters are included in this model. If required, the probability of multiple aircraft encounters can be determined from the encounter density database. Future extensions to the model will likely enable the generation and simulation of random multi-aircraft encounters.

There are two fundamental types of 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 (i.e., not using a transponder) 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 is that the two aircraft blunder into close proximity.

This report focuses on correlated encounter modeling and so is appropriate for situations involving two transponder-equipped aircraft with at least one flying with a discrete code. Other, related models for uncorrelated encounters have also been developed by Lincoln Laboratory.

TABLE 1**Encounter model types.**

C = correlated, U = uncorrelated conventional, X = uncorrelated unconventional

Intruder Aircraft	Aircraft of Interest	
	Discrete	VFR
Discrete	C	C
VFR	C	U
Noncooperative conventional (fixed-wing powered aircraft)	U	U
Noncooperative unconventional (balloon, glider)	X	X

Table 1 shows which encounter model is appropriate to use depending on the types of aircraft involved in the encounter. There are three types of encounter models shown in Table 1: correlated (C), uncorrelated conventional (U), and uncorrelated unconventional (X). In terms of the aircraft types shown, Discrete indicates a cooperative aircraft using a non-1200 Mode A transponder code, and VFR denotes a cooperative aircraft using the 1200 Mode A transponder code. Only the correlated model (bold C) is discussed in this report.

It must be emphasized that this encounter model only represents the characteristics of almost exclusively manned aircraft flying in the NAS during the period of data collection between December 1, 2007 and August 31, 2008. There is interest in applying this encounter model to simulating encounters with unmanned aircraft systems (UAS) to aid in the development and certification of sense and avoid systems. Because the encounter model is based on recorded encounters that involved few, if any, unmanned aircraft, special care must be taken when extrapolating the use of this model for unmanned aircraft studies. ATC involvement and procedures may or may not induce similar encounter geometries on unmanned aircraft encounters as were captured in the current model. If UAS are treated identically to manned aircraft in terms of flight plans and receiving and dynamically responding to ATC separation services, then it is expected that this encounter model would be appropriate in terms of representing the types of close encounters that may occur. If the UAS is handled or responds differently from manned aircraft, then additional adjustments to the model will be required. Other modifications to the model would also be required to obtain estimates of UAS traffic densities and corresponding expected encounter rates. Finally, flight characteristics such as turn rates and vertical rates for unmanned aircraft are likely to be different from those observed. Methods for addressing some of these issues have been developed and are discussed later in this document (Sections 4.4 and 4.5).

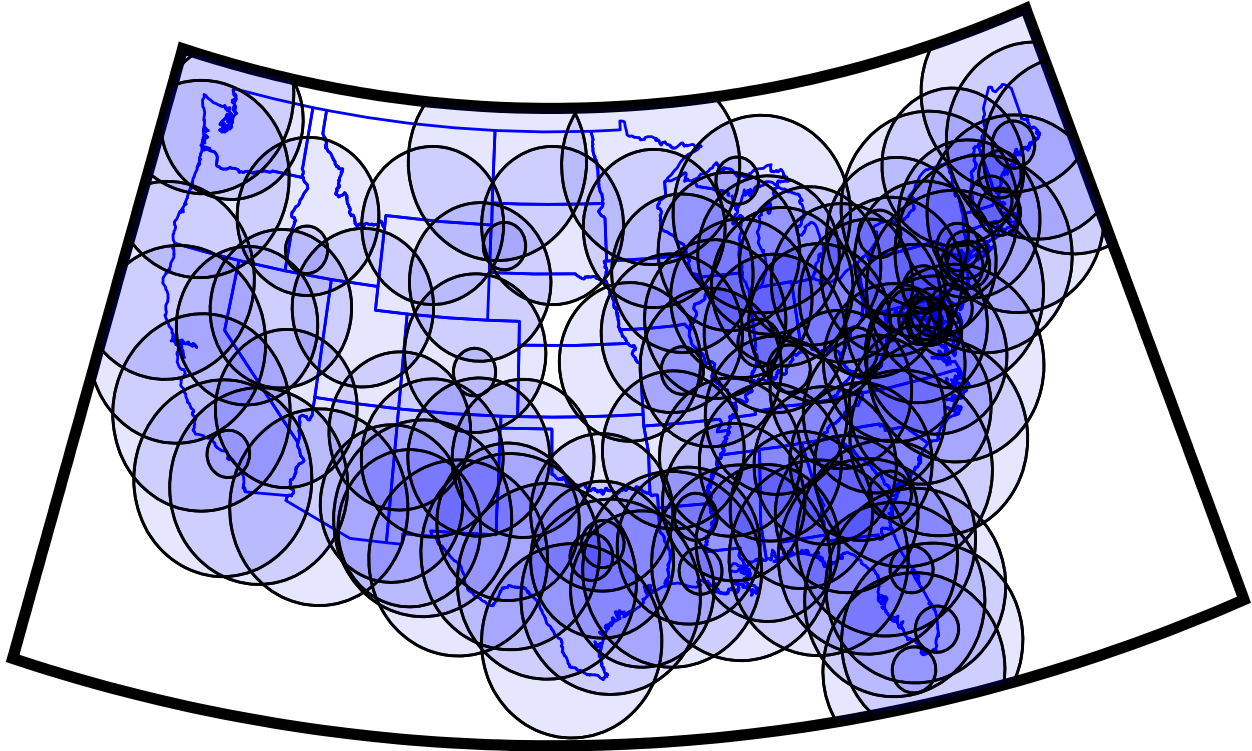
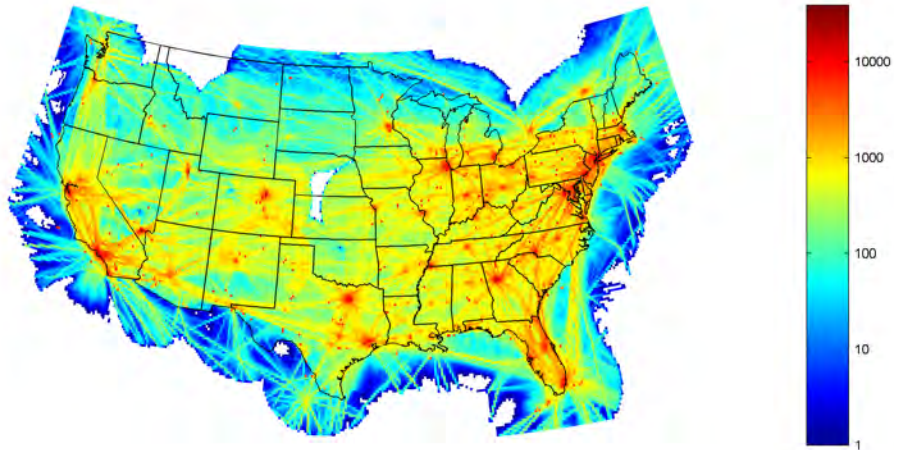


Figure 1. Radar coverage map.

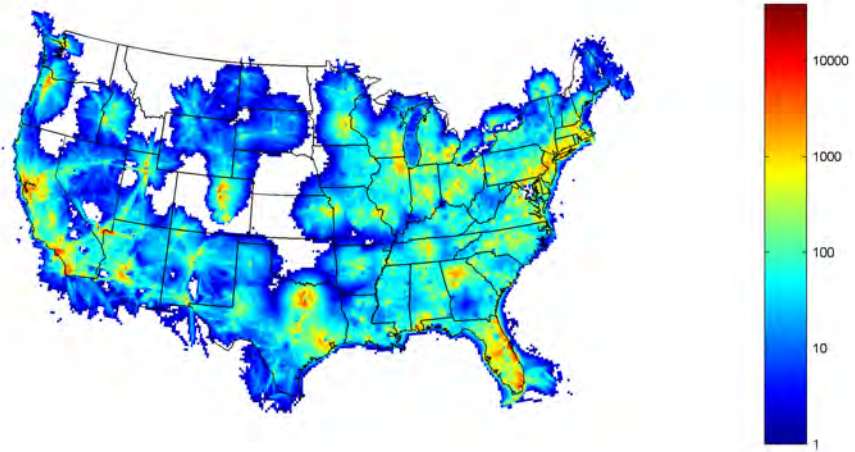
1.2 RADAR DATA

The radar data used to build the model 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 1 shows the coverage by the more than 120 sensors that were used to construct the model. Discrete-code and VFR (1200-code) beacon reports were collected between December 1, 2007 and August 31, 2008, representing 2,902,000 VFR flight hours and 21,153,000 discrete-code flight hours. Figure 2a shows the cumulative flight hours of observed discrete-code traffic, and Figure 2b shows the cumulative flight hours of observed VFR (1200-code) traffic, both covering the period 1 December 2007 to 31 August 2008. Appendix G summarizes the volume of data received from each sensor.

Note that there are a number of advantages to the 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 IFR flight plans and provides updates once per minute showing aircraft position after processing by air traffic control automation. In contrast, the RADES data feed streams continuously directly from the radar, including primary-only radar returns as well as cooperative transponder returns (whether on a flight plan or not), providing track updates



(a) Discrete code aircraft.



(b) 1200-code aircraft.

Figure 2. Cumulative flight hours per $1/6 \times 1/6$ degree cell from 1 December 2007 to 31 August 2008. Cells with less than 1 flight hour of observed traffic are uncolored.

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.

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

1.3 PROCESS OVERVIEW

Figure 3 diagrams the steps involved in processing radar data to build the encounter model and generate encounters for simulation. 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, sufficient statistics are collected to describe the distribution over maneuvers and other properties of trajectories. Bayesian model selection methods are used to search for the best network structure that represents the observed data (Section 3). The best network structure is then selected and the associated sufficient statistics are obtained to generate new trajectories that are representative of those observed by radar (Section 4). Finally, the trajectories are used in a dynamic simulation to evaluate encounters between aircraft with or without a collision avoidance system (Section 5). Section 6 discusses using the encounter model for large-scale safety analysis and collision risk estimation.

A series of Appendices is also included to provide additional detail and background data. Briefly: Appendix A describes the parameters and data formats used to represent encounters in the model; Appendix B describes how to use the data file and model parameters to generate encounter trajectories; Appendix C describes the process to be followed to validate encounter trajectories generated from the encounter model; Appendix D defines the filter used to extract encounters from the radar data; Appendix E documents the process used to identify parallel approach operations and formation flight; Appendix F provides more detail on the radar data processing and track fusion process; Appendix G lists the radars included in the RADES data feed; Appendix H shows encounter density maps for the U.S. at varying altitude levels; Appendix I shows summary histograms of features such as vertical and horizontal miss distance, altitude of encounter, or approach angle; Appendix J provides a brief review of Bayesian networks; and finally, Appendix K provides an archive of the different Bayesian networks that were tested as candidates for the final model.

A digital representation of the sufficient statistics and model structure described in this report are available on request from Lincoln Laboratory. An example of using the data in that file to construct a random trajectory is provided in Appendix B. Additionally, a Matlab software package is available to generate random trajectories using the data tables.

Throughout this report we use the standard units used in aviation. Altitudes are in feet (ft), positions are in nautical mile (NM) coordinates, vertical rates are in feet per minute (ft/min), turn rates are in degrees per second (deg/s), airspeeds are in knots (kt), and accelerations are in knots per second (kt/s). Time is reported in seconds (s).

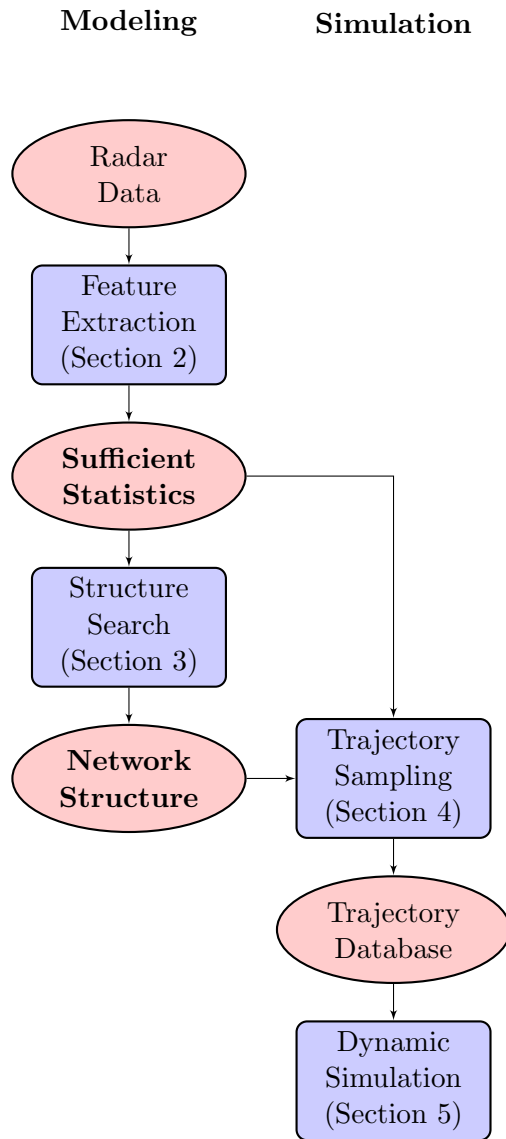


Figure 3. Modeling and simulation process overview. The sufficient statistics and network structure (in bold) are the main elements provided as part of this work.

This page intentionally left blank.

2. APPROACH

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 the model specify how the position, altitude, and airspeed of each aircraft involved in an encounter change over time. In particular, each state specifies a vertical rate \dot{h} and turn rate $\dot{\psi}$ for each aircraft. Given an initial airspeed v , airspeed acceleration \dot{v} , and vertical rate \dot{h} , as well as the altitude layer L , airspace class A , relative bearing χ , approach angle β , and horizontal and vertical miss distance of the aircraft at the time of closest approach, one can determine from the model how the aircraft trajectories evolve over the course of the encounter.

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 (that is, independent probabilities) 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 J provides the necessary background on Bayesian networks. The remainder of this section defines an encounter in further detail, introduces the variables used to describe an encounter, and presents the modeling structure.

2.1 ENCOUNTER DEFINITION

As stated previously, we define an encounter as a situation where two aircraft have lost standard separation and whose trajectories result in significant collision risk. We identify encounters from radar data using a conservative approximation of the threat declaration logic contained in TCAS version 6.04a (see Appendix D). The conditions that qualify a pair of trajectories as an encounter occur well in advance of a collision to allow a collision avoidance system time to track and avoid the threat. Note that only a small number of encounters that are simulated without a collision avoidance system would result in an NMAC. However, it is important to generate and evaluate non-NMAC trajectories as well due to the potential for a collision avoidance system to induce a collision.

The encounter model is based on encounters observed over nine months from a national radar data feed. Since we only want to model unintentional encounters where a collision avoidance system would potentially need to intervene, we filter out intentional encounters that may be military exercises, parallel runway approaches, or formation flight (see Appendix E). Many of the encounters we capture in the dataset may have still been safe since the pilots on board one or both aircraft may have visually acquired the other aircraft and assumed responsibility for avoidance.

The encounters in the dataset may include the effects of TCAS intervention. Removing these effects is challenging. First, from the available radar data it was impossible to determine whether an aircraft was responding to a TCAS Resolution Advisory (RA), in part because the aircraft equipage is unknown. Second, it is difficult to remove the effects of an RA, even if one was known to have occurred. Previous studies attempted to remove the effects of TCAS manually [8], but manual removal for our model would have been impractical given the hundreds of thousands of encounters in our dataset and the correctness of the manual removal process is difficult to validate. Ignoring the effects of TCAS is not anticipated to introduce a significant systematic bias into the model, given the relatively low response rate observed in the U.S. and the likelihood of avoidance maneuvering due to pilot visual acquisition [11]. Because new TCAS monitoring data are starting to be collected nation-wide by the FAA, the potential for improved RA removal will be studied further and may form an update to this model at a later date.

As with previous encounter models [9, 10], the variables in the model define what occurs during the 40s before and 10s after the Time of Closest Approach (TCA) for the two aircraft in the encounter. TCA is defined as the moment at which the horizontal separation between aircraft is at a minimum (equivalent to the Closest Point of Approach [CPA]). An encounter in the model typically starts 40s before TCA to give TCAS or some other collision avoidance system a chance to track threats; we allow more time prior to TCA for simulation of slow-closure encounters where aircraft are in close proximity for an extended duration prior to TCA. An encounter continues for 10s after nominal TCA in case the collision avoidance system modifies the situation enough to change the timing at which CPA occurs.

Correlated encounters are constructed by first randomly determining the situation at TCA. Then, the dynamic motion of each of the two aircraft is computed that leads to the situation defined at TCA. Finally, the two aircraft are initialized in the proper positions such that they will arrive at the desired situation at TCA in the absence of collision avoidance intervention.

2.2 MODEL VARIABLES

The aircraft at the higher altitude at TCA is called AC1. The other aircraft is called AC2. The remainder of this section explains how we model the relationship between the behavior of these two aircraft. We model the following variables to describe each encounter:

- **Vertical Miss Distance vmd :** Vertical miss distance is defined as the difference in altitude between the two aircraft at the point of closest approach (point of minimum horizontal miss distance).
- **Horizontal Miss Distance hmd :** Horizontal miss distance is defined as the horizontal range between the two aircraft at the point of closest approach (point of minimum horizontal miss distance).
- **Airspace class A :** This variable may take on one of four values: B, C, D, and O, indicating which class of airspace the encounter is in. The values B, C, and D correspond to the controlled airspace classes defined by the FAA. The value O represents “other airspace,” which includes

Class A, E, and G airspace. The airspace class variable was incorporated into the model to account for the variation in how aircraft fly in different airspace classes. Note that Class A can be distinguished from Classes E and G by referring to the next variable: altitude layer.

- **Altitude layer L :** Airspace is divided into five altitude layers, similar to prior encounter models developed by Eurocontrol (see Figure 4). The first layer spans from 1000 to 3000 ft Above Ground Level (AGL) to capture encounters in the traffic pattern. Encounters that occur below 1000 ft AGL are filtered out; TCAS, for example, will not issue resolution advisories for encounters occurring below 1000 ft AGL. The second layer spans from 3000 ft AGL to 10,000 ft Mean Sea Level (MSL), the upper limit for aircraft without transponders and the 250 kt airspeed restriction. The third layer spans from 10,000 ft MSL to FL180, the upper limit for VFR traffic and the beginning of Class A airspace. The fourth layer spans from FL180 to FL290, the beginning of the Reduced Vertical Separation Minimum (RVSM). The fifth layer includes all airspace above FL290. The altitude layer for an encounter is determined by the altitude of AC1 at TCA.
- **Approach Angle β :** Approach angle is the heading of AC2 relative to AC1 at TCA. Figure 5 shows how β is calculated. The European encounter models also used this definition.
- **Bearing χ :** Figure 5 shows how the bearing of AC2 relative to AC1 is calculated at TCA. Given β , hmd , and χ , we can uniquely identify the lateral position and orientation of AC2 relative to AC1 at TCA (Section 5).
- **Category C_1 and C_2 :** We currently divide aircraft into two categories: 1200-code aircraft and discrete-code aircraft. Compared to discrete-code aircraft, aircraft squawking 1200 tend to accelerate more frequently, fly at lower altitudes and at lower speeds, and are often smaller aircraft.
- **Initial Airspeed v_1 and v_2 :** We model initial airspeeds of the two aircraft. We assume zero wind since aircraft close enough to be in an encounter situation are most likely within the same air mass and experiencing approximately the same windfield.
- **Acceleration \dot{v}_1 and \dot{v}_2 :** The model assumes constant airspeed acceleration for the duration of the encounter as was the case in the prior European encounter models. This is a reasonable approximation given the short 50 s duration of each encounter.
- **Turn rate $\dot{\psi}_1$ and $\dot{\psi}_2$:** Turn rate is permitted to change every second in the model. The prior European and ICAO models allowed only a single turn during an encounter.
- **Vertical rate \dot{h}_1 and \dot{h}_2 :** The vertical rate is permitted to change at every second. All prior cooperative models allowed only a single vertical acceleration period during an encounter.

Including vmd and hmd , there is a total of sixteen variables describing an encounter. Four of the variables, $\dot{\psi}_1$, $\dot{\psi}_2$, \dot{h}_1 , and \dot{h}_2 , are dynamic variables that are allowed to change over the course of an encounter.

Because many of the variables are closely related due to physical constraints and aircraft characteristics (e.g., turn rate and vertical rate) it is important to properly represent correlations

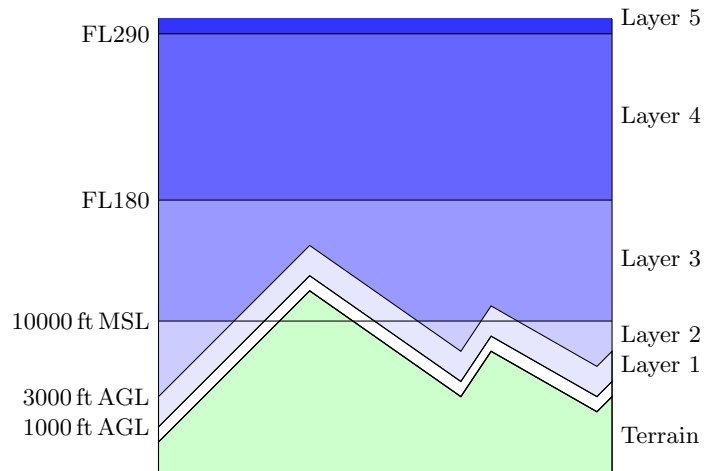


Figure 4. Airspace layers.

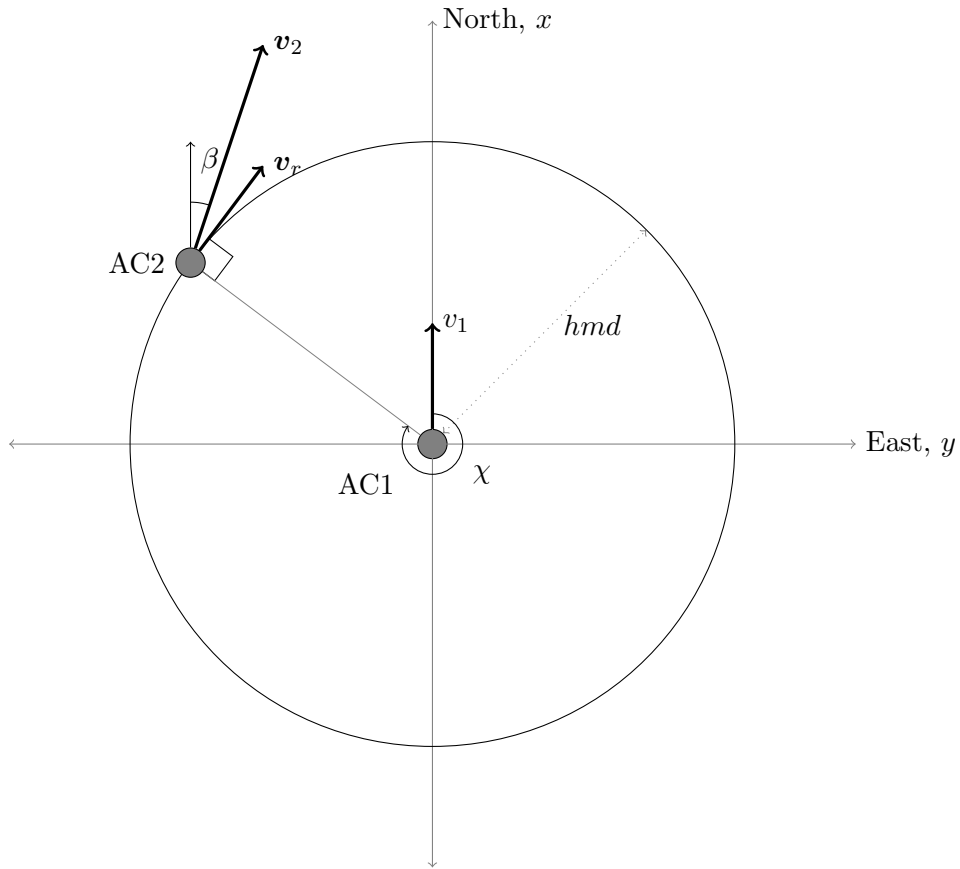


Figure 5. Approach angle (β) and bearing (χ) definition.

in the model. Independently sampling from distributions for turn rate and vertical rate would miss these important relationships. The remainder of this section explains how to model joint probability distributions over these variables. To generate an encounter, we first randomly sample from the joint distribution over the encounter variables to define the encounter geometry and initial conditions. Second, we use a Markov model to determine how the dynamic variables, i.e., turn rate and vertical rate, evolve during the course of the encounter. 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 dynamic variables specifying the trajectories of the aircraft evolve over time.

2.3 INITIAL DISTRIBUTION

The aircraft encounter model represents the distribution over the initial values of \dot{h}_1 , \dot{h}_2 , ψ_1 , ψ_2 , \dot{v}_1 , \dot{v}_2 , v_1 , and v_2 . It also models the distribution over the values of C_1 , C_2 , β , χ , L , A , vmd , and hmd at TCA. Probabilistic relationships between these variables are represented using a Bayesian network. An example of such a relationship is the one between turn rate and vertical rate. Without properly capturing this dependency and other important dependencies, unrealistic situations may be generated, e.g., involving aircraft with simultaneously high climb rates and high turn rates.

A Bayesian network (e.g., Figure 6) includes a series of nodes represented by rectangles and directed arcs represented by arrows. Each node corresponds to a particular variable that may take on one of several discrete values with associated probabilities. Certain variables, such as airspace class or altitude layer, are naturally quantized into a few discrete values (e.g., B, C, D, and O for airspace class). Continuous variables, such as approach angle, vertical rate, or miss distance, are described by a series of discrete bins from which a continuous value is later selected. Within each node, then, is a description of the possible values a variable can take and the probability that each value will occur. The directed arcs describe how the probabilities of one variable may depend on other variables. Arrows leading into a particular node denote which parent nodes must first be evaluated in order to select the value of the given node. For example, referring to Figure 6, the probabilities for node A depend on the value of node L ; the probabilities for node hmd depend on the values of nodes vmd and L . In the latter case, for instance there would be a conditional probability table describing the probability of each possible value of hmd conditioned on the values of vmd and L : $P(hmd | vmd, L)$.

Because there are many possible ways variables can be connected in the model, it is necessary to use a quantifiable scoring process to evaluate the quality of each candidate network. For this model, we used a Bayesian scoring process (Appendix J) to evaluate different Bayesian network structures and choose a structure that optimally represents the encounters we observed. The space of possible Bayesian network graphical structures is intractably large, and so we have to rely upon heuristic search and imposed node connectivity constraints. To guide our search, we used the Greedy Thick Thinning algorithm as implemented in the GeNIe modeling environment developed by the Decision Systems Laboratory of the University of Pittsburgh (<http://dsl.sis.pitt.edu>). We restricted the maximum number of parents per node in the network to five, and imposed temporal layer constraints. Temporal layers enforce causality beliefs by ensuring that variables can

only be parents of other variables within their own temporal layer or temporal layers below it. We used four temporal layers for our network. The first temporal layer contains A and L , which identify the type of airspace in which the encounter occurs. The second layer contains C_1 and C_2 , which identify the types of aircraft involved in the encounter, with the mixture of aircraft types depending on the airspace. The third layer, which depends on both the airspace and the types of aircraft involved, describes the control variables of the encounter trajectories, and includes \dot{h}_1 , \dot{h}_2 , $\dot{\psi}_1$, $\dot{\psi}_2$, \dot{v}_1 , \dot{v}_2 , v_1 , and v_2 . The variables in the final layer, including β , χ , vmd , and hmd , describe the situation at TCA and may be functions of all the variables in the previous temporal layers. Figure 6 shows the optimal structure for the initial distribution. Appendix K shows other candidate network structures and their scores.

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 nodes and directed arcs used in the structural diagram show the order in which this sampling occurs. For example, to determine the initial state of the aircraft based on the structure in Figure 6, we first randomly determine an altitude layer L . Once the altitude layer has been determined, an airspace class A is selected. The probabilities associated with each airspace class depend on which altitude layer was chosen earlier. Once A and L have been selected, we then randomly select the category of AC1, C_1 , and so on. An example of working through this process is provided in Appendix B. Alternately, we could specify a particular value for the altitude layer (and possibly the airspace class as well) and then randomly select values for the remaining variables.

2.4 TRANSITION DISTRIBUTION

We use a separate Bayesian network to model how the variables \dot{h}_1 , \dot{h}_2 , $\dot{\psi}_1$, and $\dot{\psi}_2$ evolve over time. In this network, we have a first layer that represents the state of the system at the present time step and a second layer that represents the state of the system at the next time step. There may be dependencies between layers and within the second layer. Such a two-layer temporal Bayesian network is known as a dynamic Bayesian network [12, 13]. Parameter and structure learning in dynamic Bayesian networks is similar to regular Bayesian networks (Appendix J).

Figure 7 shows the structure used for the transition distribution. Again, we chose the highest-scoring network structure among our candidate network structures (see Appendix K). Given a structure, sufficient statistics extracted from data, and a prior, we can then sample from the Bayesian network to determine the next vertical rate and turn rate 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 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 and are also appropriate given typical timescales of dynamic maneuvers.

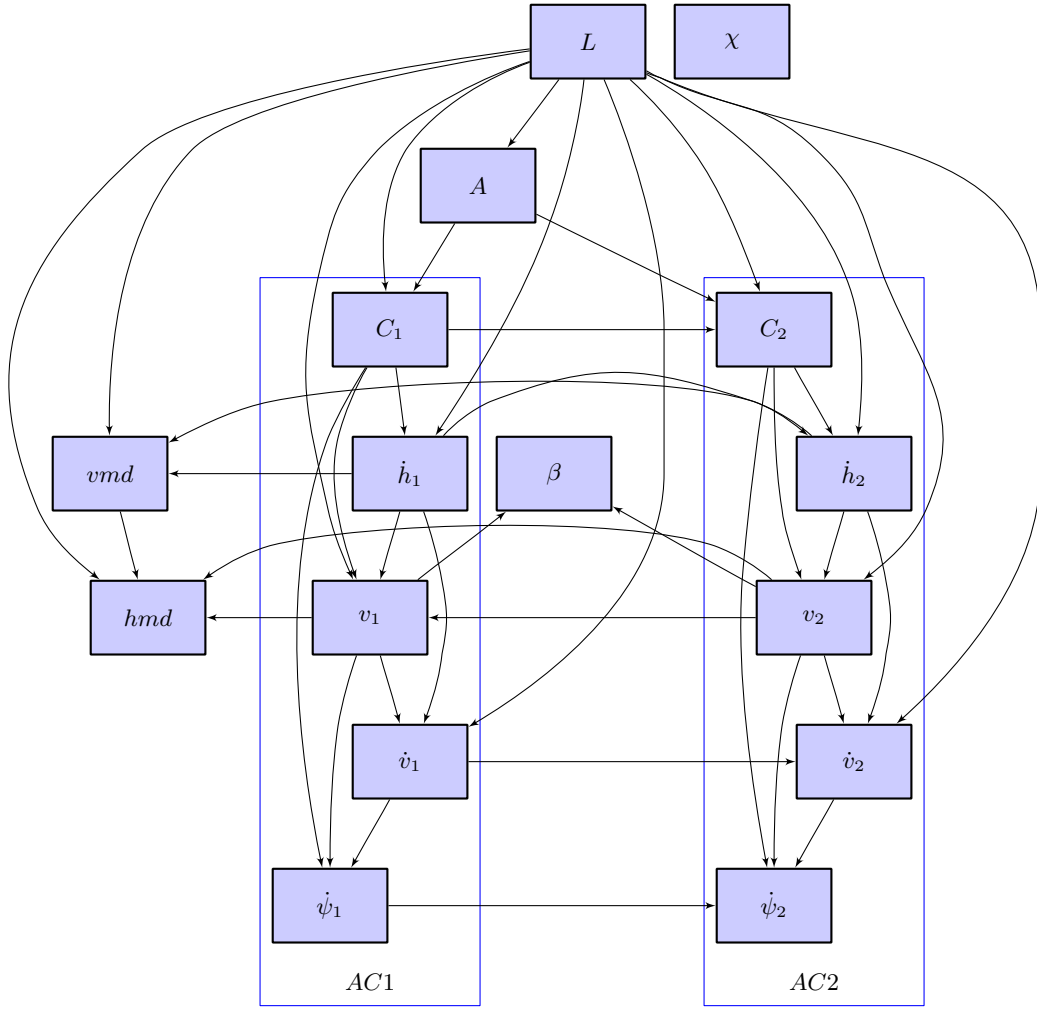


Figure 6. Bayesian network representing the variable dependency structure for the initial distribution.

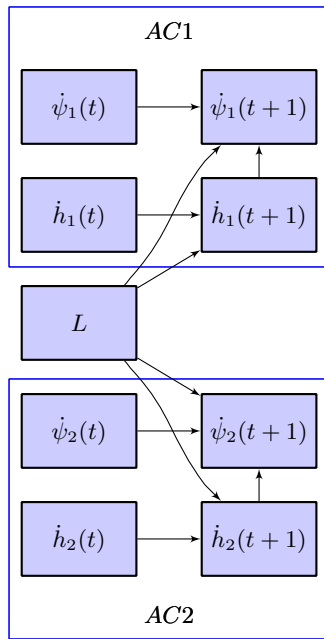


Figure 7. Bayesian network representing the variable dependency structure for the transition distribution.

A complete trajectory is constructed by updating the aircraft state in 1 s intervals. Within each interval, the four control variables \dot{h}_1 , \dot{h}_2 , $\dot{\psi}_1$, and $\dot{\psi}_2$ are treated as target values and held constant. We assume constant airspeed acceleration for each aircraft over the course of the encounter, and \dot{v}_1 and \dot{v}_2 are chosen from the initial network. A dynamic model is used to compute and update the aircraft state at each time step based on these piecewise-constant target values. The dynamic model is independent of this encounter model and not discussed in this report, but a process for validating an encounter model against the one used at Lincoln Laboratory is described in Appendix C.

This page intentionally left blank.

3. ESTIMATION

At a high level, the modeling process involves taking in a large volume of radar data (spanning nine months from more than 120 sensors) and carefully filtering and processing that data to extract features of close encounters between aircraft. Features include static variables that specify an encounter (such as vertical or horizontal miss distance, approach angle, and altitude layer) and multiple, dynamic variables that describe aircraft motion leading up to and through the closest point of approach (such as turn rate and vertical rate every second). To aid in data processing, each feature was quantized into several bins and counts were taken of the frequency with which each bin was occupied by radar data. Based on these counts (termed sufficient statistics), probability tables were then constructed so that each feature can be randomly generated such that the overall geometries and dynamics are representative of the actual events observed in the radar data.

Accordingly, the inputs to this process are raw radar reports (range, azimuth, altitude, time) and the outputs are probability tables that specify the likelihood that a given feature will take on a value within a bin corresponding to each table cell. This section describes the processing required to transform radar tracks into sufficient statistics that may be used to model cooperative encounters. Figure 8 outlines the multiple-stage feature extraction process.

3.1 ENCOUNTER FILTERING

The raw radar data are first processed using a tracking algorithm developed at Lincoln Laboratory [14]. A fusion algorithm, also developed at Lincoln Laboratory [15], then fuses tracks from multiple sensors to give one global view of all the tracks in U.S. airspace (see Appendix F). Next, encounter events are identified (filtered) from the available tracks.

There are two stages in encounter filtering. The first stage uses only the raw fused tracks. After tracks are formed by associating radar scans, filtering software searches for cases where two tracks satisfy criteria defining an encounter by searching each track, scan by scan, and determining if any other track in the dataset satisfies the criteria for an encounter at the time of that particular radar scan (see Appendix F). An encounter is declared on the basis of two tests: a proximity test and a TCAS-like test. The details of the tests are included in Appendix D. Although one or both of the aircraft may be detected by multiple sensors, tracker output for both aircraft during the encounter is only from a single sensor.

After the raw tracks have been filtered and assigned to an encounter pair, we remove outlier radar scans, which is described in the following section. Before removing outlier data points, however, we eliminate encounters with tracks that had fewer than ten scans. We found that approximately ten scans are required to accurately estimate the various maneuver rates. After outlier removal, tracks are then smoothed and interpolated to 1 Hz, according to the procedures described in Section 3.4.

The second stage of filtering uses the interpolated tracks. First, if either interpolated track does not have at least 45 data points after interpolation, the encounter is rejected. Encounters with tracks of sufficient length and altitude at TCA are then re-subjected to the TCAS-like test described

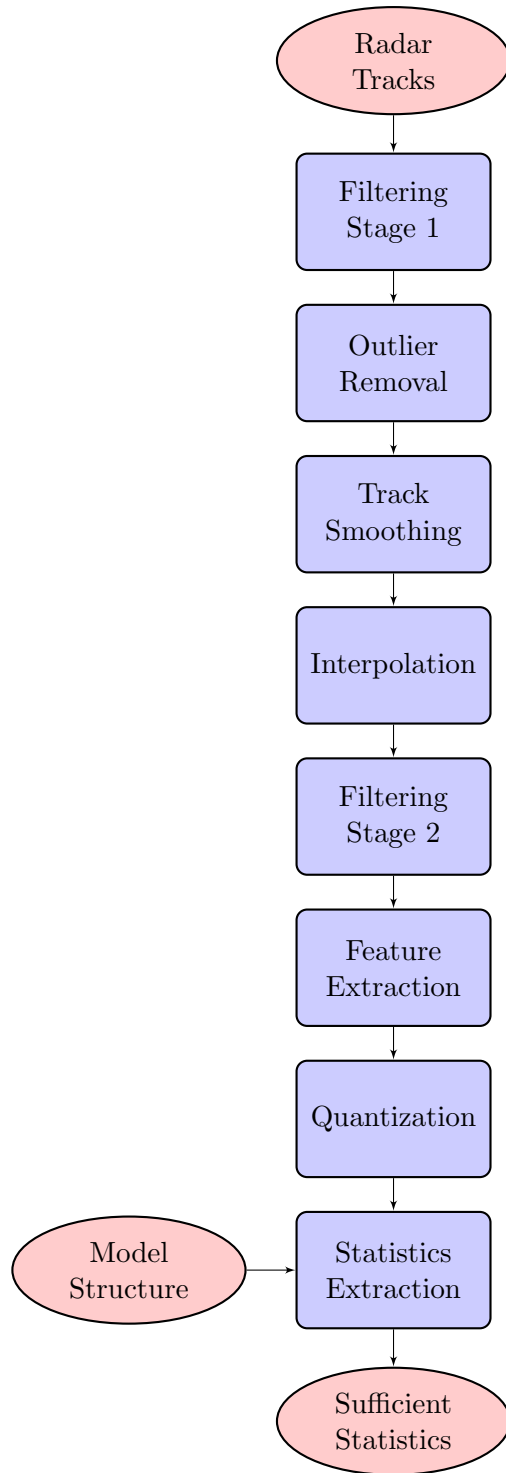


Figure 8. Estimation process flow.

in Appendix D. At this point, TCA is re-calculated using interpolated data points. Finally, there must be at least 60% of the expected number of Mode C reports (given the scan rate for the sensor) for both aircraft in the 40 s before and the 10 s after TCA.

Additionally, there is a test to identify parallel runway operations and formation flight. Details of these tests can be found in Appendix E. Encounters determined to be part of those operations are not included in the model.

Figure 9 shows the results of the various filtering operations from capturing encounters with the coarse filter to the final set of encounters used to build the model. Figure 10 shows the spatial density of the remaining encounters, and Appendix H shows the density of encounters at different altitude layers.

3.2 OUTLIER REMOVAL

The first step in processing the raw radar tracks is to detect and remove outlier data points. In the horizontal plane, we remove jumps with ground speeds above 800 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 800 kt are stored in a list of candidates for removal. We iterate through the list of candidates and remove the one that, because of its removal, minimizes the total sum of speeds above the set threshold for the remaining segments. The process repeats until there are no longer any segments with speeds above the threshold. We then use the same process to remove points that have a turn rate magnitude greater than 10 deg/s, and an airspeed acceleration magnitude greater than 6 kt/s.

In the vertical plane, we remove missing Mode C altitude reports. Then we remove outliers with vertical rate magnitudes greater 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

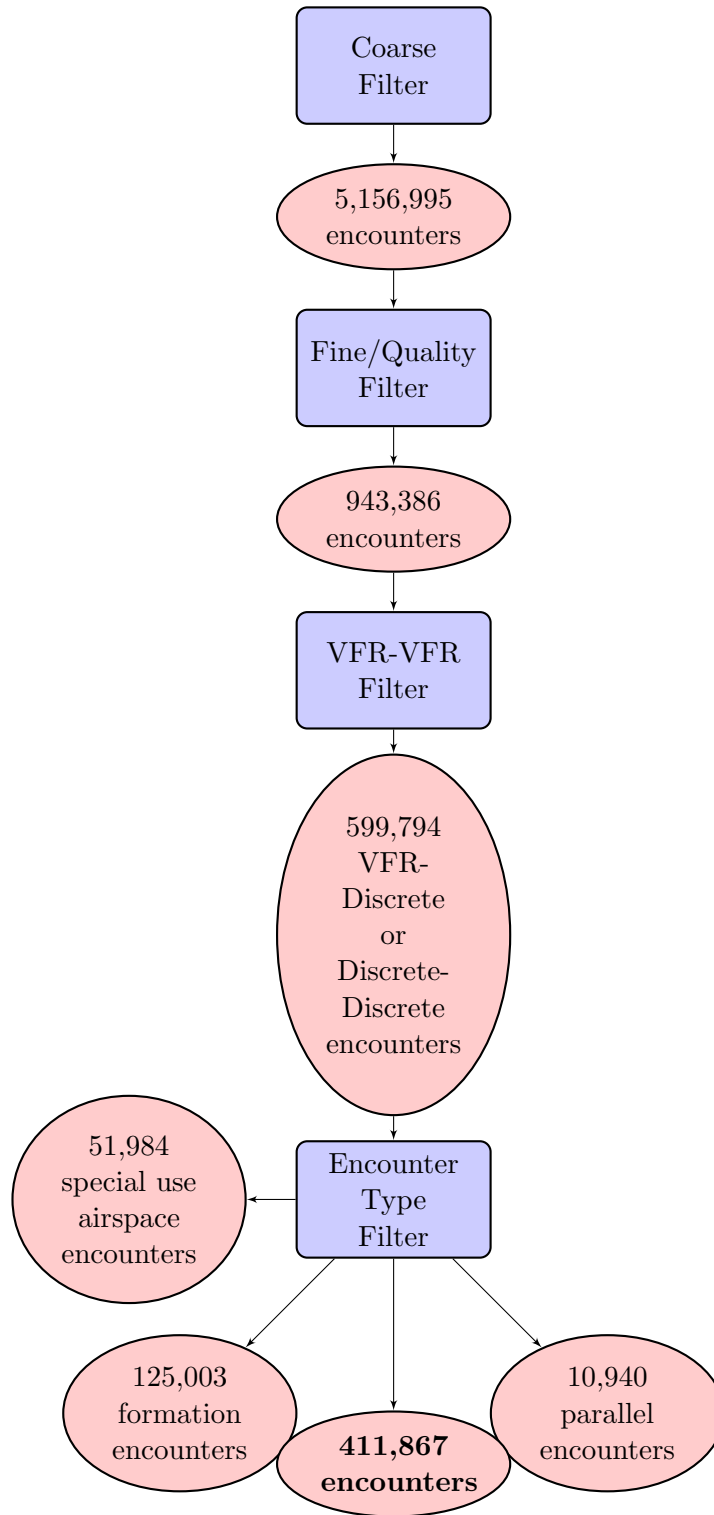


Figure 9. Filtering processes flow and results.

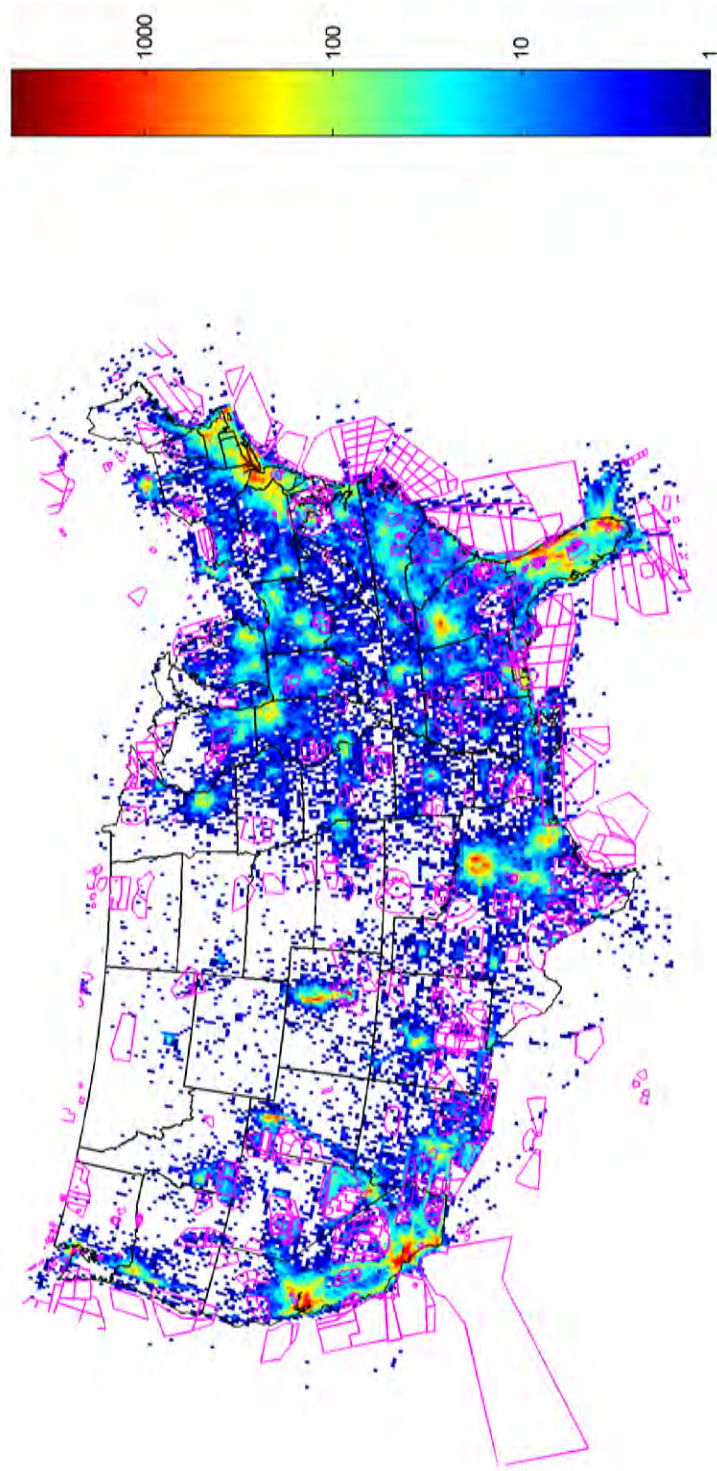


Figure 10. Encounter density over all altitude layers in encounters per cell (cell size is between 64 NM^2 and 92 NM^2) from 1 December 2007 to 31 August 2008. Special use airspace is outlined in magenta.

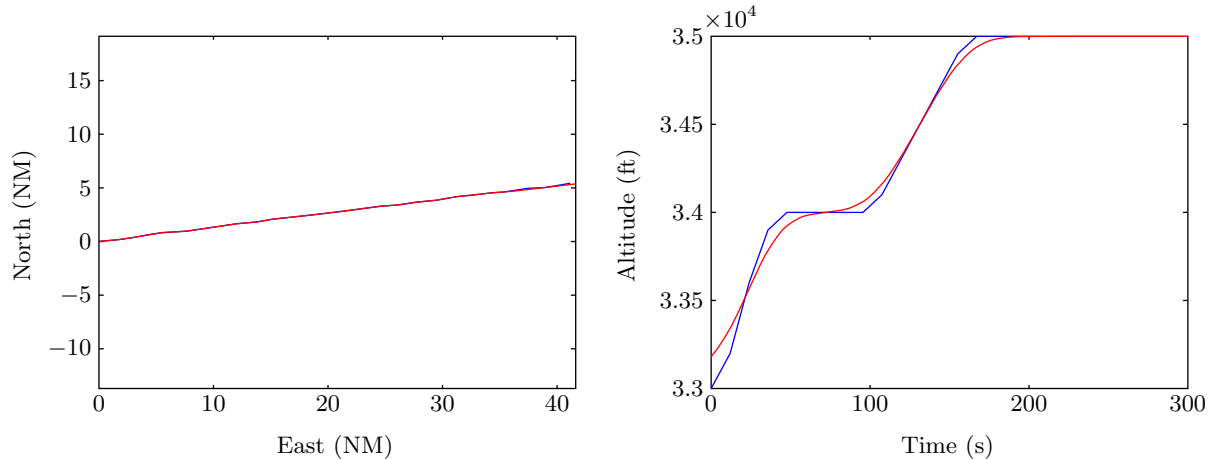


Figure 11. Preprocessing. Blue lines show the raw track. Red lines show the track after outlier removal, smoothing, and interpolation.

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 [16].

Figure 11 shows the result of outlier detection, smoothing, and interpolation on an example track from the radar data set. Figure 12 shows the result of piecewise-cubic Hermite interpolation on the same smoothed track.

3.5 FEATURE EXTRACTION

Feature extraction involves converting both interpolated tracks involved in an encounter into sequences of quantized vertical rates and turn rates. In addition, feature extraction involves charac-

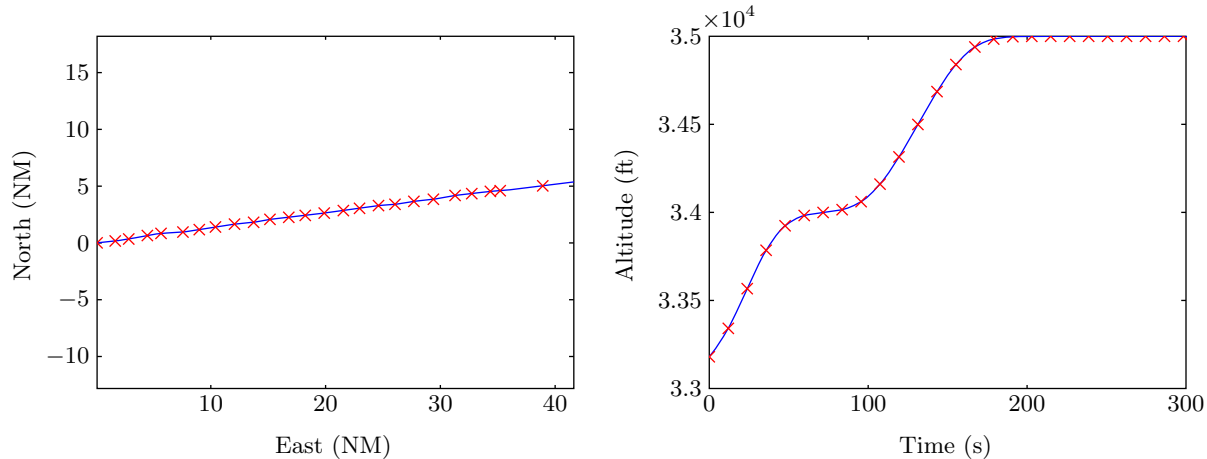


Figure 12. Piecewise-cubic Hermite interpolation on a smoothed track. Red crosses indicate smoothed data points, and the blue curve shows the interpolation.

terizing certain static features such as aircraft category, initial airspeed, and acceleration as well as features defined at TCA (including airspace class, altitude layer, approach angle, bearing, horizontal and vertical miss distance). After smoothing, each feature is then converted into appropriate quantized bins.

- Airspace class A:** We estimate latitude and longitude of radar returns using an algorithm developed at Lincoln [17]. From altitude, latitude, and longitude estimates, we determine the class of airspace of AC1 at TCA by searching through the National Airspace System Resources (NASR) database provided by the FAA. Since altitude estimates are based on Mode C reports of pressure altitude, uncorrected for barometric variation, it is possible that the airspace associated with some encounters is identified incorrectly. We expect that this limited inaccuracy in airspace class identification due to barometric variation has a negligible impact on the model. We also use the altitude, latitude and longitude of the radar returns to determine whether an encounter occurred in special use airspace, as defined by the Digital Aeronautical Flight Information File (DAFIF) provided by the National Geospatial-Intelligence Agency (NGA). Most special use airspace can be turned on or off according to time, but we did not include the time dimension when filtering encounters for occurring within special use airspace. Since most air-routes tend to avoid special use airspace in general, it is expected that not including time in filtering for special use airspace encounters will introduce negligible bias in the model.
- Altitude layer L:** Determining the altitude layer requires computing altitude above ground level (AGL) and above Mean Sea Level (MSL). MSL altitude is given by the smoothed Mode C altitude reports. We estimate AGL altitude by subtracting an estimate of ground elevation from MSL pressure altitude. Our estimates of ground elevation comes from Digital Terrain Elevation Data (DTED) provided by the NGA. We use DTED Level 0, which has post spacing of 30 arcseconds (approximately 900 meters).

- **Approach angle β :** The instantaneous heading of an aircraft is determined by the arc tangent of the ratio between the change in y and x position of the aircraft at each point in time. All angular values are reported in degrees where 0 deg corresponds to true north. The approach angle β is the difference in headings between AC1 and AC2. An approach angle near 0 deg indicates a parallel encounter, and an approach angle near 180 deg indicates a head-on encounter.
- **Bearing χ :** The bearing of AC2 relative to AC1 at TCA is calculated by taking the arc tangent of the ratio of the difference between the two x - y aircraft positions at TCA.
- **Category C_1 and C_2 :** If the majority of Mode A returns for a track is 1200, the aircraft is labeled as 1200-code. Otherwise, the aircraft is labeled as discrete-code.
- **Airspeed v_1 and v_2 :** The true airspeed at time t is given by

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

- **Acceleration \dot{v}_1 and \dot{v}_2 :** Acceleration is estimated by subtracting the initial airspeed from the final airspeed and dividing by the duration of the encounter (50 s).
- **Turn rate $\dot{\psi}_1$ and $\dot{\psi}_2$:** 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.
- **Vertical rate \dot{h}_1 and \dot{h}_2 :** 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 $\dot{h}(t) = h(t+1) - h(t)$.

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).

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. For example, referring to Table 2, all airspeed values less than 100 kt are placed into one bin; airspeeds between 100–200 kt are placed in a different bin, etc. The cut points were chosen to capture the variation of the features as shown in the histograms in Figure 13. Additional histograms are provided in Appendix I.

Figure 14 shows an example of feature extraction on the same track as was shown in Figure 11. Looking at vertical rate, for example, the track begins in the $[400, 1000)$ bin, then transitions into

TABLE 2

Cut points used for feature quantization.

	Cut Points
v	100, 200, . . . , 500
\dot{v}	-2, -0.25, 0.25, 2
\dot{h}	-3000, -2000, -1000, -400, 400, 1000, 2000, 3000
$\dot{\psi}$	-6, -3.5, -1, -0.25, 0.25, 1, 3.5, 6
β	30, 60, . . . , 330
χ	90, 270 (angles between 270 deg and 90 deg belong to the same bin)
hmd	0.1, 0.5, 1
vmd	100, 200, . . . , 900

the [1000, 2000) bin, back to [400, 1000), then to [-400, 400), etc. The duration spent in each bin contributes to the likelihood of that particular bin being chosen when sampling and generating a trajectory. Similarly, the rate at which transitions between bins occurs affects the dynamics with which turn rate and vertical rate will change during an encounter.

3.6 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 the model. For the two Bayesian networks, the sufficient statistics are simply the counts N_{ijk} of the various features (see Appendix J).¹

The counts N_{ijk} are then compiled into a separate table for each feature. For example, the table for airspace class A (which is itself dependent on altitude layer L) is shown in Table 3. Each cell entry in the table represents the counts N_{ijk} for that combination of parent bin for altitude layer L (1, 2, 3, 4, 5) and bin of airspace class A (B, C, D, O). An electronic file available from Lincoln Laboratory contains all of the data required to construct these tables.

Figure 15 illustrates the convergence of the Bayesian network parameters (probabilities) as additional data are added. 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 any elements in the conditional probability tables with the addition of more data. As the figure shows, the change in the probabilities in the conditional probability tables converged to less than 0.0001 given the amount of radar data collected over the span of nine months.

To summarize, this section describes the process used to construct a model of correlated encounters based on radar data. Raw radar data were processed through multiple filtering and

¹The 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 [18].

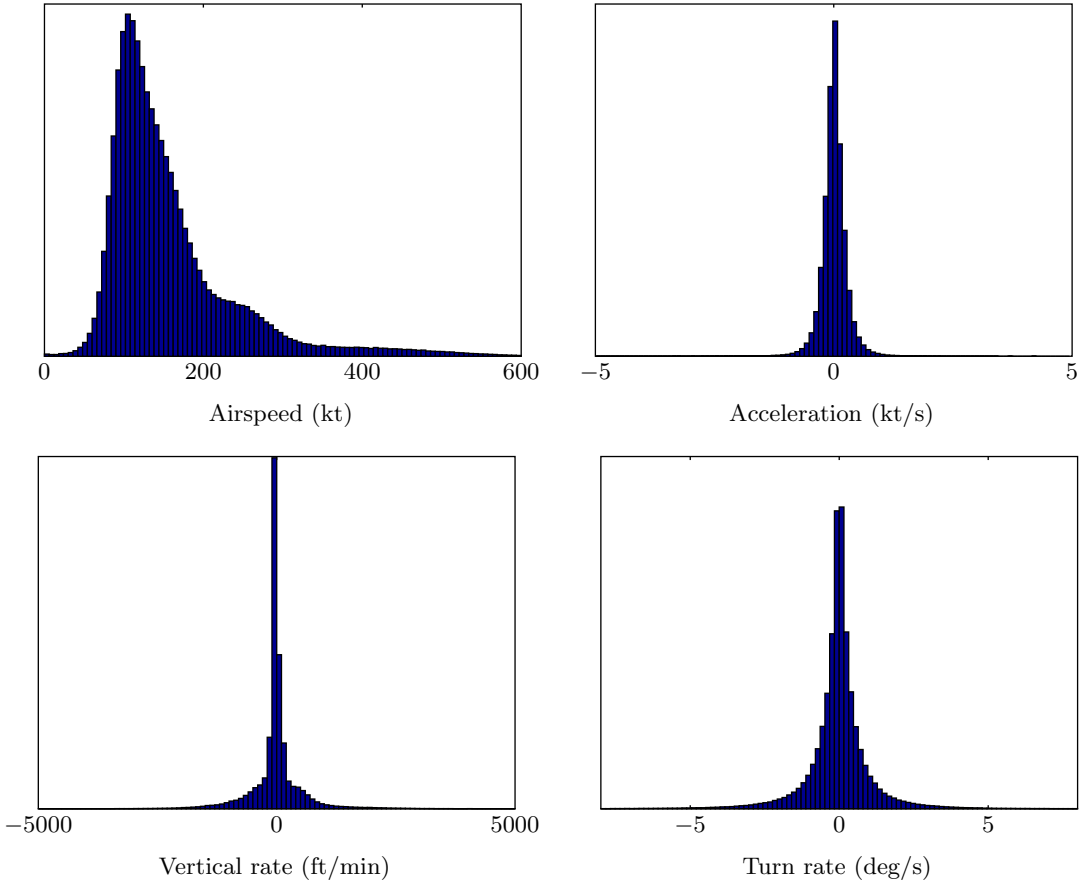


Figure 13. Feature histograms of observed radar data based on 411,867 encounters.

TABLE 3

$$N(A | L)$$

L	A			
	B	C	D	O
1	22504	14720	19628	141108
2	16473	4804	17	150972
3	899	12	0	27293
4	0	0	0	7718
5	0	0	0	5719

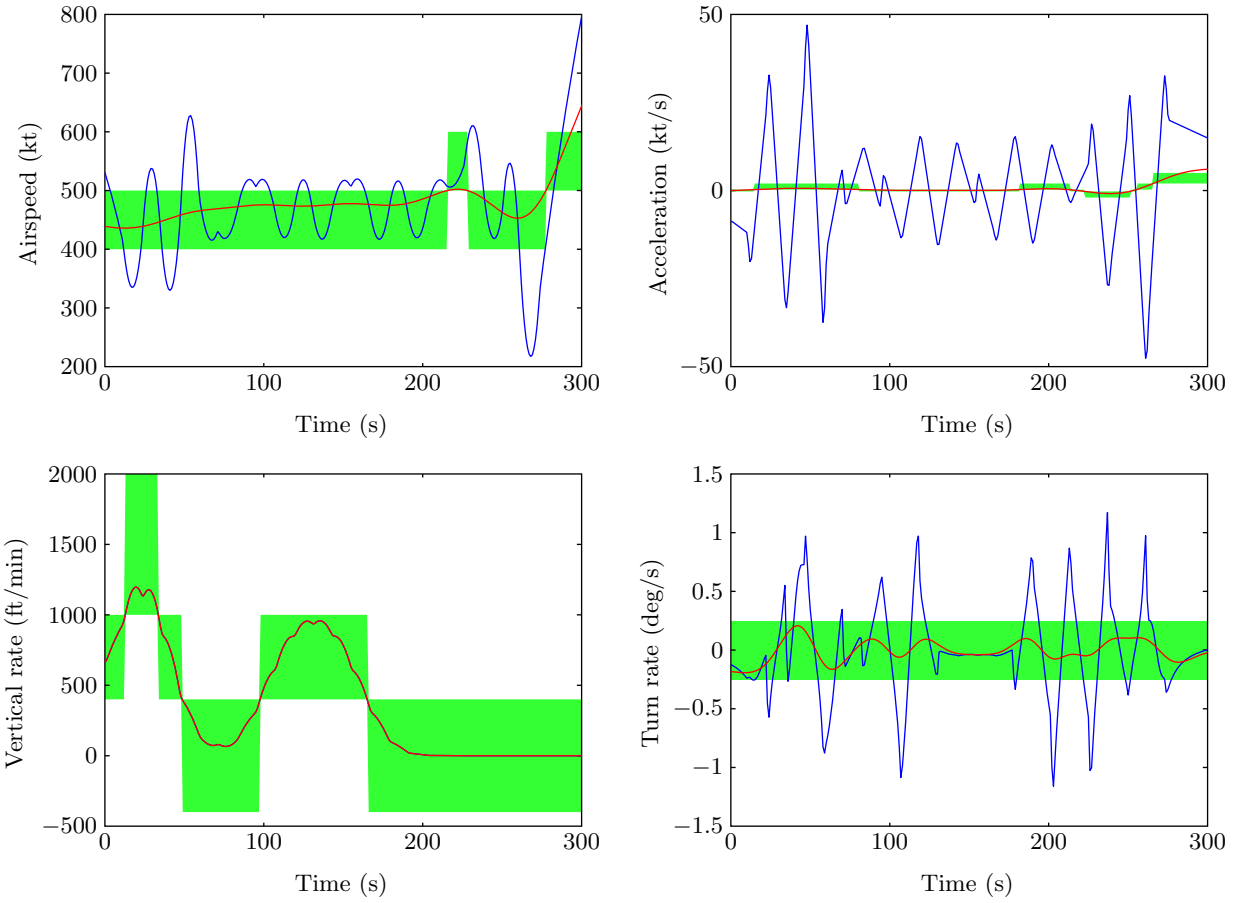


Figure 14. A plot of extracted features over time. Blue lines show features before smoothing, and red lines show features after smoothing. The green blocks show the quantized bins into which the features are then assigned.

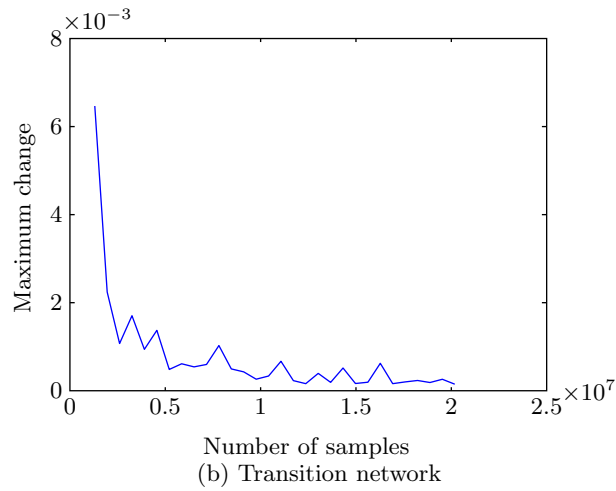
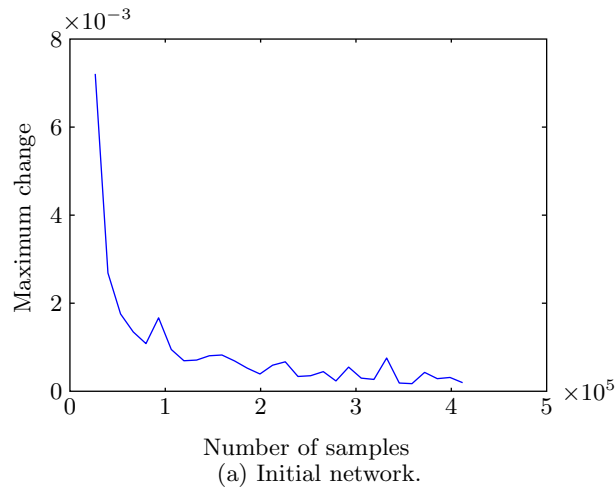


Figure 15. Convergence curve for Bayesian network conditional probabilities. A total of 411,867 encounters were available for use in building the initial network. A total of $411,867 \times 49 = 20,181,483$ data points were available to build the transition network (based on 49 time steps following the initial variable assignment for each encounter).

tracking stages and then digested into a series of probability tables organized within Bayesian networks. Each table cell represents the probability of a particular feature taking on a value within a certain quantized bin. The next section describes how to use these tables to generate random, but statistically representative, encounters through sampling. An example of producing encounters using the encounter model is given in Appendix B.

This page intentionally left blank.

4. SAMPLING

Once the data have been processed as described in the previous section, one can use the model structure and sufficient statistics to produce new encounters that are representative of the ones observed by radar. The first step involves sampling from the discrete Bayesian network tables representing the initial and transition distributions. This provides a series of bins that represent coarse values to be used for each feature. The second step involves converting the coarse, discrete samples into fine, continuous samples by sampling within bins. Figure 16 illustrates this process.

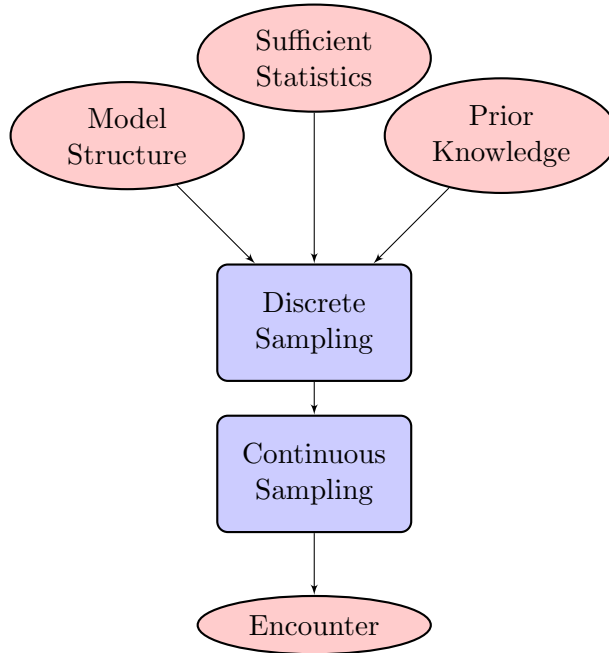


Figure 16. Sampling process flow.

4.1 DISCRETE SAMPLING

We begin by sampling from the Bayesian network representing the initial state distribution. Referring back to Figure 6, first one would randomly sample from the table describing $N(L)$ to determine the bin to use for altitude layer L . Given that bin for altitude layer, one would next sample from the table $N(A | L)$ to determine the bin for airspace class A and so on. See Appendix B for a more detailed description of this process.

Formally, consider sampling from a table describing the i th variable X_i (e.g., where $X_i =$ airspace class). As was shown in Table 3, each table contains a series of bins k (e.g., B, C, D, O) for each parent bin j (e.g., 1, 2, 3, 4, 5).

The probability assigned to bin k is then given by

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

where

- j is the instantiation of the parents of X_i in the Bayesian network,
- N_{ijk} is the value provided in the data tables, equal to the number of times $X_i = k$ 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 sufficient statistics tables and sampling according to the resulting relative frequencies. This ensures that there are no transitions with zero probability in the Markov model.

For instance, referring to Table 3, if altitude layer L had previously been selected to be 1, then the probability of selecting airspace class $A = B$ would be $P(A = \text{Class B}) = (1 + 22504)/(1 + 22504 + 1 + 14720 + 1 + 19628 + 1 + 141108) = 0.1137$.

The set of samples from the initial-state Bayesian network defines which bins airspeed, vertical rate, turn rate, and acceleration fall. Once the initial state of the trajectory is defined, one samples from the dynamic Bayesian network representing how the state changes over time using a similar process. First, the values for $\dot{h}(t)$ and $\dot{\psi}(t)$ are set to their initial values and the standard Bayesian network sampling scheme is used to determine $\dot{h}(t+1)$ and $\dot{\psi}(t+1)$. This process may be indefinitely repeated — typically a span of 50 s is used for each trajectory in the encounter.

4.2 CONTINUOUS SAMPLING

To produce a fine, continuous sample given a coarse discrete sample from the initial distribution, a random sample is selected uniformly within each bin. For example, if we determine that the initial airspeed is within the bin $[100, 200)$, we simply sample from a uniform distribution over the half-open interval $[100, 200)$. Because the first and last bins associated with each interval are unbounded, it is necessary to impose some bounds. Table 4 shows the boundaries used in the quantization scheme.

With regard to the transition network, instead of sampling within bins for turn rate and vertical rate at every time step, we only produce a new continuous sample with a fixed probability per time step that has been estimated from the radar data. The probabilities of resampling at each time step are 0.0521451, 0.0542133, 0.0796733, and 0.0829421 for \dot{h}_1 , \dot{h}_2 , $\dot{\psi}_1$, and $\dot{\psi}_2$, respectively.

TABLE 4
Sampling boundaries.

	Boundaries
v	50, 100, 200, . . . , 600
\dot{v}	-5, -2, -0.25, 0.25, 2, 5
\dot{h}	-5000, -3000, -2000, -1000, -400, 400, 1000, 2000, 3000, 5000
$\dot{\psi}$	-8, -6, -3.5, -1, -0.25, 0.25, 1, 3.5, 6, 8
β	0, 30, . . . , 360
χ	90, 270
hmd	0, 0.1, 0.5, 1, 3
vmd	0, 100, . . . , 900, 6000

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, the value of zero is assigned instead of sampling uniformly in order to prevent very small vertical rates and turn rates.

Figure 17 shows an example of vertical rates and turn rates generated by sampling from the Bayesian networks. First, coarse bins are selected randomly and shown as green bars. Next, fine values within each bin are selected and shown in blue. Note that vertical rate, for example, is held to precisely zero when the bin spans zero, and otherwise the vertical rate may be reselected several times within non-zero bins at the rate described above.

Figure 18 shows a complete sample two-aircraft encounter. Generating a complete encounter requires translation of features into tracks by running a simulation, described in Section 5.

4.3 MODEL VALIDATION

It is important to have confidence that sampling from the model will produce a representative distribution of encounters in the national airspace. A model that does not properly capture the true level of variation in the observed encounters may not provide a representative mix of possible encounter situations in which a given collision avoidance system is expected to operate. On the other hand, a model that over-fits the complexity of the observed data may not be flexible enough to generate reasonable encounter situations that, while not observed, are certainly plausible.

There are three key aspects of validation relevant to this model. First, the features extracted from the radar data should be shown to be a valid representation of actual close encounters between

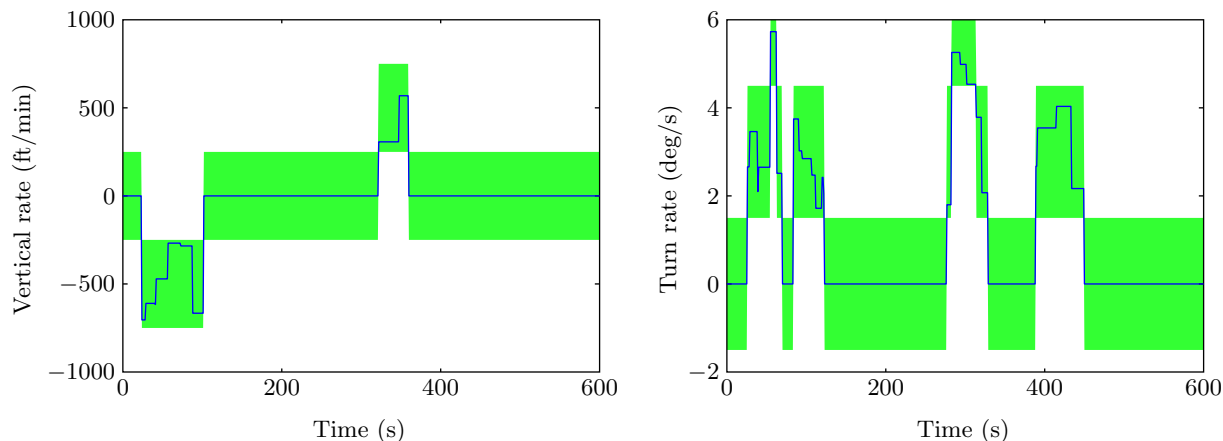


Figure 17. A plot of sampled features over time. Green blocks show the coarse discrete bins and blue lines show the resulting fine samples within those bins.

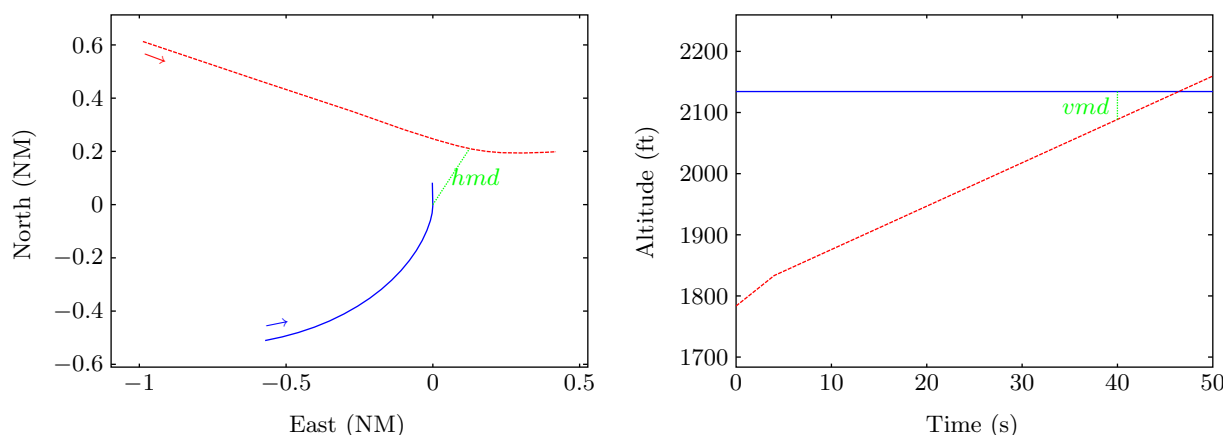


Figure 18. An encounter generated by sampling from the initial and transition distributions (note: not the same as shown in Figure 17).

aircraft. Second, when implemented, the features that are randomly-selected from the data tables should occur with probabilities appropriate to those tables (that is, the feature-selection process should be implemented properly). Third, given a series of features defining an encounter, the actual aircraft trajectories generated in simulation should be implemented accurately.

Regarding the first case, validation has been performed by comparing a set of sample encounters drawn from the model against the observed encounters using features not included in the model. For example, unlike the Eurocontrol model, the model presented here did not explicitly define (using a variable) whether an encounter would involve a crossing in altitudes of the two aircraft involved (a so-called crossing encounter). Since TCAS, for instance, works differently depending on whether aircraft are projected to be crossing in altitude over the course of the encounter, it is important to ensure the model produces a reasonable rate of crossing encounters. We observed 12.46% crossing encounters in our radar dataset. Out of a sample of 10,000 randomly-generated

encounters, the model produced 11.42% crossing encounters. Similarly, we also examined the rate of slow closure encounters, defined as encounters that begin already at close range (within Z and $DMOD$, see Appendix D) 40 s prior to TCA. These types of encounters are also known to degrade the effectiveness of TCAS and are important to model at a realistic rate. Approximately 0.34% of encounters we observed in radar data are slow closure encounters, and out of 10,000 sample encounters generated randomly from the model, we found 0.56% slow closure encounters. These results suggest that the model will indeed produce a set of encounters that realistically reflect what is occurring in actual airspace.

Regarding the second validation case, software for generating samples from the model is available from Lincoln Laboratory. If one wishes to use their own sampling software, samples may first be validated by comparison against the histograms in Figure 13 and Appendix I. With as few as 1,000 samples, it should be clear whether the generated samples come from the distribution shown in our histograms. Although the histograms do not explicitly capture the many relationships between variables, it is unlikely that an incorrect implementation of the sampling algorithm will produce matching marginal distributions because the marginal distributions are an indirect byproduct of the relationships between variables.

Finally, to ensure the third validation case that trajectories are constructed properly given a set of features, Appendix C describes a process to be followed using a set of sample data available from Lincoln Laboratory.

4.4 SAMPLING WHEN ONE TRAJECTORY IS SPECIFIED

Directly sampling from the Bayesian networks as described in Section 4 will produce encounters that are representative of how two cooperative aircraft may be involved in close encounters over their operational lifetime. For some safety analyses, however, one may wish to constrain one of the aircraft to follow some fixed trajectory. For instance, it may be of interest to examine collision avoidance performance in only climb-out flight conditions, or along a specific known departure flight profile. For this discussion, assume that aircraft AC1 is given a specified trajectory to follow; specifying the trajectory of AC2 would involve a similar process.

The first step is to extract the features of the prescribed trajectory for AC1, including L , A , C_1 , \dot{h}_1 , v_1 , \dot{v}_1 , and $\dot{\psi}_1$. We shall denote this set of these known variables \mathcal{A} . We then need to sample from the Bayesian network representing the initial distribution to obtain values for the unknown variables, $\mathcal{B} = \{\chi, \beta, vmd, hmd, C_2, \dot{h}_2, v_2, \dot{v}_2, \dot{\psi}_2\}$. Unfortunately, we cannot use a direct sampling scheme (Section 4.1) to obtain values for \mathcal{B} . The directed edge from v_2 to v_1 complicates the sampling process. The instantiated value for v_2 changes the probability distribution over v_1 , and consequently some of the other variables in \mathcal{B} . For example, suppose that the model predicts that an aircraft is more likely to encounter another aircraft flying at a similar airspeed. If v_2 is instantiated to a high airspeed, then v_1 is more likely to also be a high airspeed. If we were to sample v_2 without taking into account the instantiation of v_1 , then the samples would not reflect the distribution observed in the airspace.

One way to ensure that the distribution of the remaining variables reflects the encounter model distribution is to use *logic sampling* [19]. To obtain a sample, generate as many samples from all of the variables using the direct sampling scheme (Section 4.1) until the sampled values for the variables in \mathcal{A} match the fixed values extracted from the prescribed trajectory. Given the graphical structure shown in Figure 6, we may use this process:

1. Fix $\mathcal{A} = \{L, A, C_1, \dot{h}_1, v_1, \dot{v}_1, \dot{\psi}_1\}$ to the values extracted from the prescribed trajectory.
2. Sample χ .
3. Sample C_2 given A and C_1 .
4. Sample \dot{h}_2 given L, C_2 , and \dot{h}_1 .
5. Sample v_2 given C_2, L , and \dot{h}_2 .
6. Sample v_1 given L, C_1, \dot{h}_1 and v_2 .
7. If the bin randomly assigned to v_1 matches the bin extracted from the prescribed trajectory, go to Step 2.
8. Continue sampling the remaining variables.

One potential disadvantage of logic sampling is that one might have to loop back to Step 2 many times before an acceptable sample is generated. Section 4.5 discusses other possible sampling methods, including arc reversal. By manipulating the graphical structure of the Bayesian network, one can make all of the known variables come earlier in the topological sort, thereby eliminating the need to reject any samples.

Once the initial network is sampled, the control sequence is generated from the dynamic Bayesian network as usual.

4.5 SAMPLING FROM ALTERNATIVE DISTRIBUTIONS

The previous subsection explained how to generate an encounter assuming that the trajectory of one of the aircraft is fixed. This subsection generalizes the discussion to the problem of generating encounters when some of the variables are selected from distributions that are different from those represented by the model. Sampling from an alternative distribution may be useful, for example, when modeling encounters where one of the aircraft has an airspeed distribution that is different from the modeled population. If we sample a subset of the encounter variables $\mathcal{A} \subset \mathcal{X}$ from a distribution $P_{\text{alt}}(\mathcal{A})$ that is different from the encounter model distribution $P(\mathcal{X})$, special care must be taken when sampling the remaining variables so that the relationships in the Bayesian networks are preserved, as observed in Section 4.4. In other words, after sampling the variables in \mathcal{A} we want to sample the remaining variables $\mathcal{B} = \mathcal{X} \setminus \mathcal{A}$ according to $P(\mathcal{B} \mid \mathcal{A})$ as inferred from our Bayesian network.

Simply assigning variables in \mathcal{A} to values according to $P_{\text{alt}}(\mathcal{A})$ is permissible so long as the remaining variables \mathcal{B} come later in a topological sort of \mathcal{X} . For example, if one wanted to produce samples where the category of AC1 is 1200-code, one could not simply assign C_1 to 1200-code and then assign the remaining variables using forward sampling. The reason for this is that the assignment of C_1 indirectly affects the distribution over the other variables. For example, setting C_1 to 1200-code makes the probability that the altitude layer variable gets assigned to a layer above FL180 much lower than it would be otherwise.

As mentioned in Section 4.4, one way to ensure that the distribution of the remaining variables reflects the encounter model distribution is to use logic sampling [19]. First, sample the variables in \mathcal{A} according to $P_{\text{alt}}(\mathcal{A})$. Then, sample from the encounter model Bayesian network until a sample is obtained that matches the bins that were assigned from the alternative distribution. For example, if we wish to sample v_1 such that the first two bins are equally likely and the remaining bins have probability zero, we begin by randomly assigning v_1 according to that distribution. Let us say that we assign v_1 to the first bin. We then produce a sample from the Bayesian network over all the variables. On our first try, let us say we assign v_1 to the third bin. We reject this sample completely and resample from the Bayesian network. After several tries, we finally get a sample where v_1 is assigned to the first bin, matching the assignment made by our alternative distribution. The probabilistic logic sampling approach can require the generation of many samples before acceptance. In fact, the number of samples required before acceptance grows exponentially with the size of \mathcal{A} in the worst case. If there are relatively few variables in \mathcal{A} , then probabilistic logic sampling may be appropriate.

Alternately, one can manipulate the graphical structure of the encounter model Bayesian network to make the variables in \mathcal{A} appear earlier than the variables in \mathcal{B} in a topological sort. Then, the variables in \mathcal{A} can be assigned directly according to $P_{\text{alt}}(\mathcal{A})$ and then the remaining variables \mathcal{B} can be assigned using forward sampling. Special care must be taken when changing the direction of edges (also called arcs) in the Bayesian network to preserve the relationships between variables. If one wishes to reverse the direction of an edge between two nodes X_i and X_j in a Bayesian network, one must add the parents of X_j to X_i and vice versa. This *arc reversal* operation preserves the original distribution over \mathcal{X} [20,21]. A downside to arc reversal is that the resulting network can become very complex.

Another approach called *likelihood weighting* [22] involves simply generating a collection of samples from the Bayesian network and then overwriting the variables in \mathcal{A} with samples from $P_{\text{alt}}(\mathcal{A})$. The results of each sample, i.e., the NMAC probability, must be weighted according to the likelihood of the assignment to the variables in \mathcal{B} given the assignment of the variables in \mathcal{A} .

Lincoln Laboratory is currently developing a Global-Hawk-specific cooperative encounter model. The approach being pursued is manipulating the graphical structure of the Bayesian network such that the parameters of one aircraft (Global Hawk) can be completely specified from distributions appropriate to that aircraft. The intruder aircraft's parameters are then organized at a lower topological level in the Bayesian network such that they can be sampled once the Global Hawk parameters have been selected.

This page intentionally left blank.

5. ENCOUNTER INITIALIZATION

Sampling from the Bayesian networks tells us how the aircraft will maneuver during an encounter and their nominal relative positioning at TCA in the absence of collision avoidance maneuvering. However, the variables sampled from the networks do not describe where to initialize the aircraft at the beginning of an encounter. This section describes the process for initializing the location of the aircraft in simulation so that the encounter is constructed correctly. An encounter is implemented correctly if hmd , vmd , β , and χ that occur during simulation match the desired sampled values from the initial Bayesian network.

Figure 19 illustrates the general process for initializing an encounter. The process first requires simulating the aircraft independently of each other. The correct relative geometry at TCA is then obtained by rotating and translating the independently simulated tracks to create the correct geometry at TCA. Section 5.1 describes this process.

5.1 INITIAL AIRCRAFT POSITIONS

5.1.1 Horizontal Plane

To begin, AC1 is temporarily initialized such that it is located at the origin and flying due North at TCA ($x_{1,tca} = 0$, $y_{1,tca} = 0$ and $\psi_{1,tca} = 0$). The correct horizontal location of AC2 relative to AC1 at TCA ($x_{2,tca}$, $y_{2,tca}$) depends on the variables hmd , β , and χ from the initial Bayesian network, in addition to the velocity and heading of the two aircraft at TCA as shown in Figure 20.

The encounter model requires that $\psi_{2,tca} - \psi_{1,tca} = \beta$. Since $\psi_{1,tca} = 0$ in Figure 20, the correct value for $\psi_{2,tca}$ is β . The encounter definition also states that the horizontal separation between the aircraft is minimal at TCA. In order for this to be true the relative velocity vector $v_{r,tca}$, when originating from the location of AC2, must be tangent to the circle about AC1 with radius hmd . The relative velocity vector between the aircraft is defined as

$$\mathbf{v}_{r,tca} = \begin{bmatrix} v_{2,tca} \cos \beta - v_{1,tca} \\ v_{2,tca} \sin \beta \end{bmatrix}. \quad (4)$$

If $\mathbf{v}_{r,tca}$ is not tangent to the circle about AC1, then the minimum horizontal separation between the aircraft will not equal the value sampled from the initial Bayesian network and will not occur at the desired TCA.

As demonstrated in Figure 20, there are two locations for AC2 that fit the criteria for both β and hmd . The $(x_{2,tca}, y_{2,tca})$ coordinates of these two locations are

$$\begin{bmatrix} x_a \\ y_a \end{bmatrix} = \frac{hmd}{\|\mathbf{v}_r\|} \begin{bmatrix} -v_{r,y} \\ v_{r,x} \end{bmatrix} \quad \begin{bmatrix} x_b \\ y_b \end{bmatrix} = \frac{hmd}{\|\mathbf{v}_r\|} \begin{bmatrix} v_{r,y} \\ -v_{r,x} \end{bmatrix}. \quad (5)$$

The variable χ determines which of these locations to use in an encounter. If $\chi = [270, 90)$ (front), then we choose the (x, y) coordinates where $x_{2,tca} \geq 0$; if $\chi = [90, 270)$ (back), then we choose the coordinates where $x_{2,tca} < 0$.

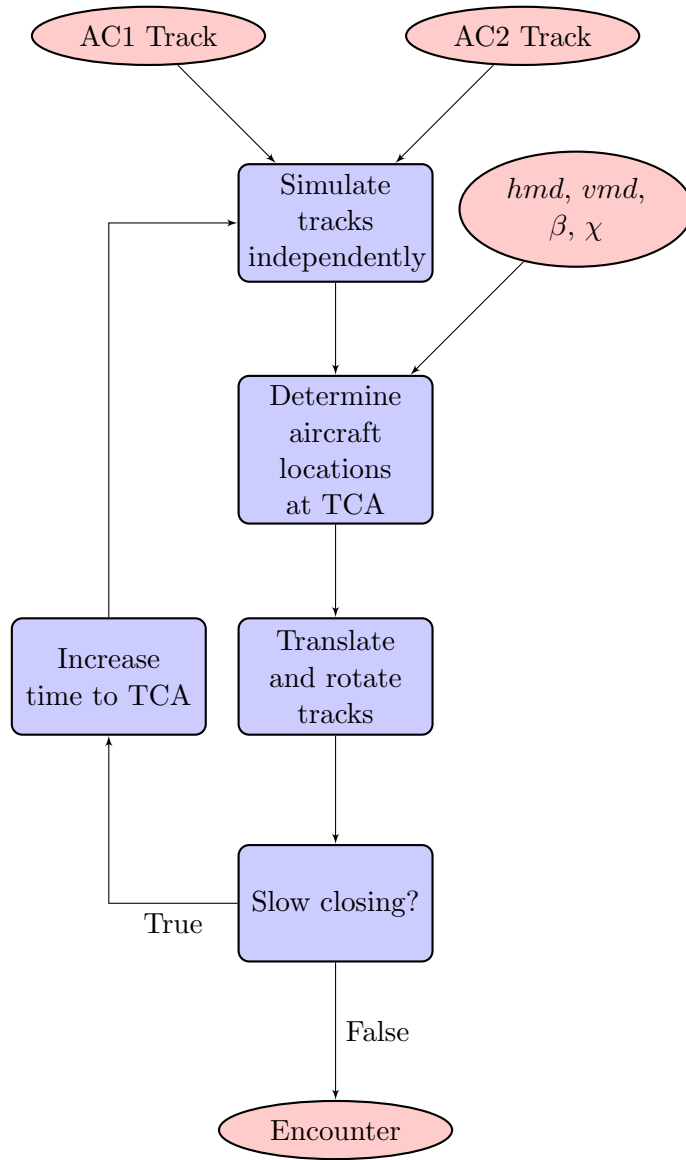


Figure 19. Encounter generation flow.

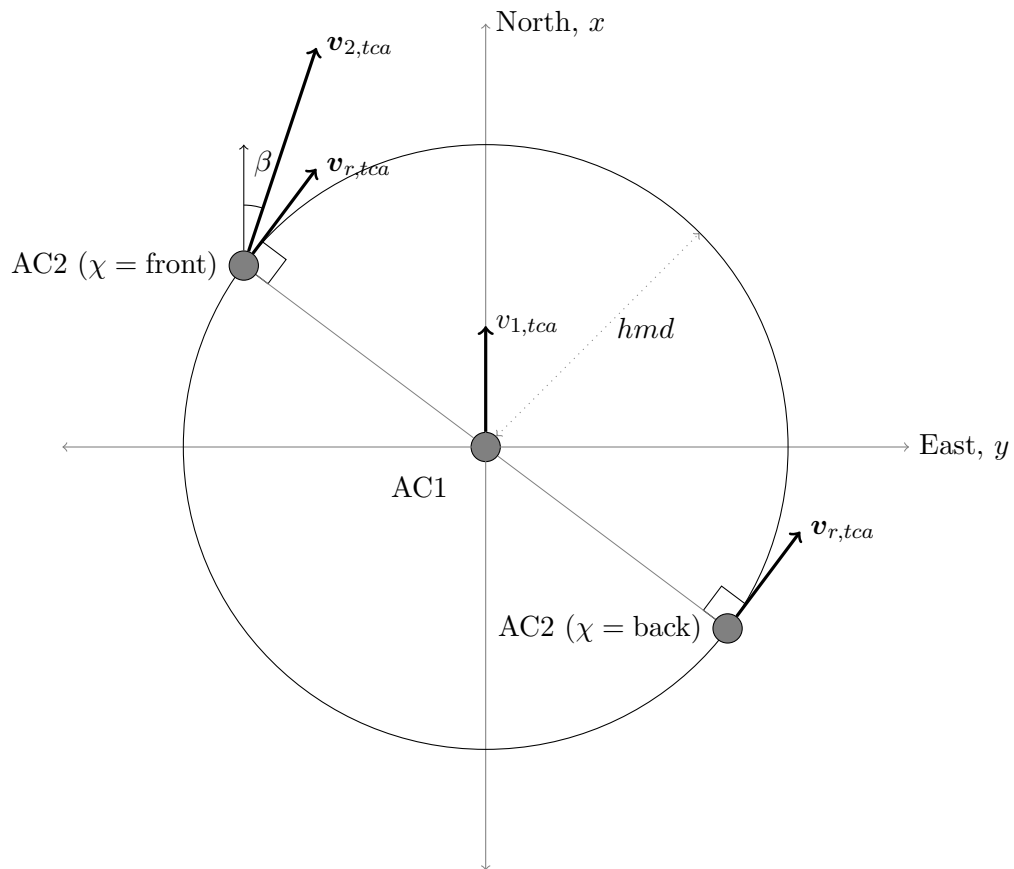


Figure 20. Encounter geometry at TCA.

Once we know the desired location and heading of the two aircraft at TCA, we need to rotate and translate the initial positions of the two independently simulated tracks in order to achieve the correct relative horizontal geometry between the two aircraft at TCA. The process for computing the initial horizontal-plane states for each aircraft involves four steps, illustrated in Figure 21:

1. Start with the aircraft at the (x, y) origin $(0, 0)$, heading north. Move the aircraft forward according to the planned behavior as defined by the encounter model. This includes any turns, vertical maneuvers, or changes in airspeed. Call the aircraft's resulting position after step 1 (x_1, y_1) .
2. Shift the initial position of the aircraft by $(-x_1, -y_1)$. The aircraft will then be located at the origin after performing its maneuver.
3. Rotate the aircraft's position around the origin such that its final heading is equal to the desired approach angle at TCA. This depends on the desired approach angle, ψ_{tca} , and the heading change achieved during the maneuver up to TCA, $\Delta\psi$. Rotation of the aircraft's position (and in fact the entire aircraft trajectory) about the origin is achieved using the direction cosine matrix multiplication below. Call the aircraft's resulting position (x_2, y_2) .

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} \cos(\psi_{tca} - \Delta\psi) & \sin(\psi_{tca} - \Delta\psi) \\ -\sin(\psi_{tca} - \Delta\psi) & \cos(\psi_{tca} - \Delta\psi) \end{bmatrix} \begin{bmatrix} -x_1 \\ -x_2 \end{bmatrix} \quad (6)$$

The aircraft's initial heading is then given by: $\psi_0 = \psi_{tca} - \Delta\psi$.

4. Finally, the aircraft position is shifted such that it is located at the desired position at TCA. This involves a shift by (x_{tca}, y_{tca}) . The resulting position $(x_2 + x_{tca}, y_2 + y_{tca})$ is the appropriate starting position for the aircraft such that it will arrive at the desired position and heading at TCA.

5.1.2 Vertical Plane

Vertically, the encounter model requires that $h_{1,tca} - h_{2,tca} = vmd$. Note that we cannot require that the vertical separation is also minimal at TCA—doing so may over-define the encounter making the encounter impossible to construct. We choose the altitude of AC1 at TCA by uniformly sampling within the altitude layer L ; AC2's altitude is determined relative to AC1 based on the value of vmd . We can then shift the initial altitudes, so that both aircraft are at the correct altitudes at TCA.

5.2 SLOW CLOSING ENCOUNTERS

Due to the nearly-parallel geometry of slow-closure-rate encounters, the two aircraft may still be at very close range 40 s prior to TCA when the simulation begins. This does not provide a realistic assessment of the performance of a collision avoidance system since that system would have likely taken some action prior to the start of the simulation. To accommodate this issue, this section describes a test for identifying slow closing encounters and a process for choosing how far back to begin an encounter before TCA.

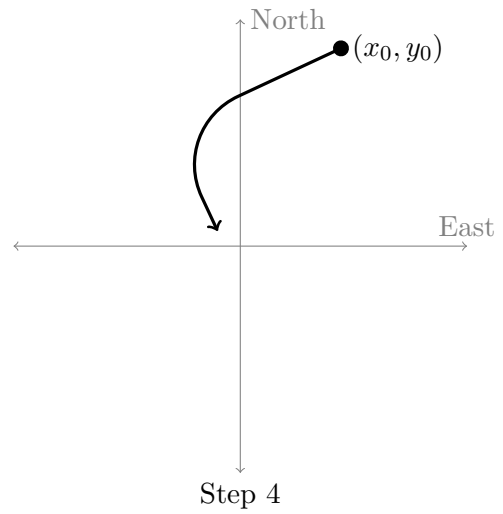
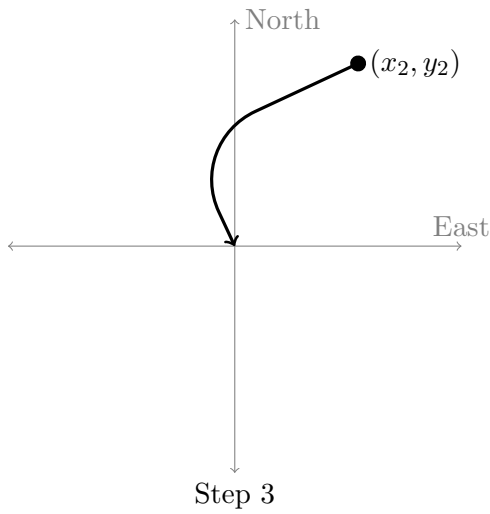
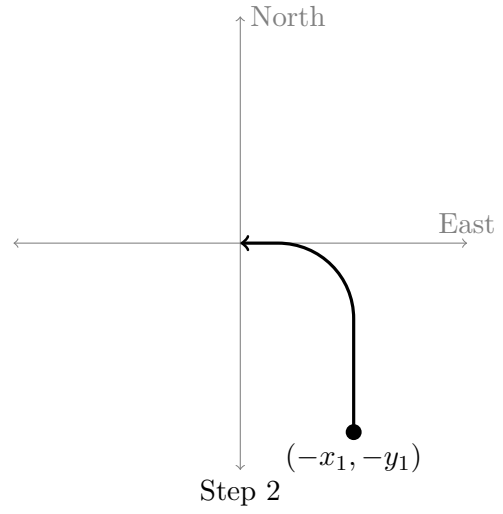
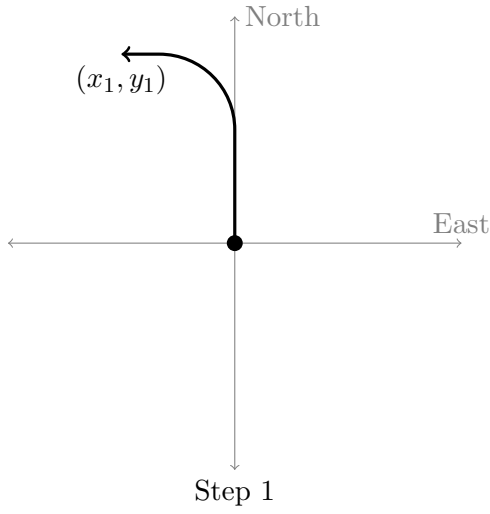


Figure 21. Example steps to initialize aircraft.

5.2.1 Identifying Slow Closing Encounters

The encounter model also requires that the aircraft are sufficiently far enough away from each other at the beginning of simulation so that the collision avoidance system under study is given sufficient time to track and issue an avoidance command prior to TCA. Encounters where the initial separation is not sufficient are termed slow-closing encounters: a slow-closing encounter is an encounter between two aircraft where the range and range rate between the aircraft are small for a long period of time. Slow closing encounters typically occur when the headings of the two aircraft are nearly equal. We cannot simply discard these encounters, because they frequently occur in the airspace and, thus, need to be included in the model. This section describes a test for identifying slow closing encounters and how to correctly re-initialize those encounters. Essentially, after rotating and translating the tracks into the initial positions, if the encounter is determined to be a slow closing encounter, we then extrapolate the initial positions backward in time so that the aircraft are sufficiently separated at the beginning of the simulation.

The tests described in Appendix D are used to determine whether or not an encounter is a slow closing encounter. First, we initialize the encounter using the standard 40s simulation time prior to TCA. We then look at the horizontal and vertical separation between the two aircraft at the beginning of the encounter. If the initial vertical separation of the two aircraft is less than Z and the initial horizontal separation is less than D_m , then we declare the encounter a slow closing encounter.

5.2.2 Increasing Time to TCA

We require that the aircraft in an encounter are initialized such that either their vertical separation is greater than Z or their horizontal separation is greater than D_m at the beginning of the simulation. When this is not the case, we increase time to TCA.

The variables sampled from the Bayesian networks do not define how the aircraft would have been maneuvering more than 40s prior to TCA. In the encounter model we assume the airspeed acceleration and turn rate for the two aircraft are both zero more than 40s prior to TCA. Additionally, we assume that the aircraft maintain the same vertical rate as is modeled at 40s prior to TCA.

Since we know the relative positions of the two aircraft and their horizontal and vertical velocities at 40s prior to TCA, we can propagate backwards to determine how much more time is required prior to TCA so that the aircraft are either vertically separated by more than Z or horizontally separated by more than D_m (not shown here). Once we determine how much additional time is required to start before TCA, we re-initialize the encounter and check to ensure that the aircraft are sufficiently separated at the beginning of the encounter.

Figure 22 shows an example of extending the start time to accommodate a slow-closing encounter in our model.

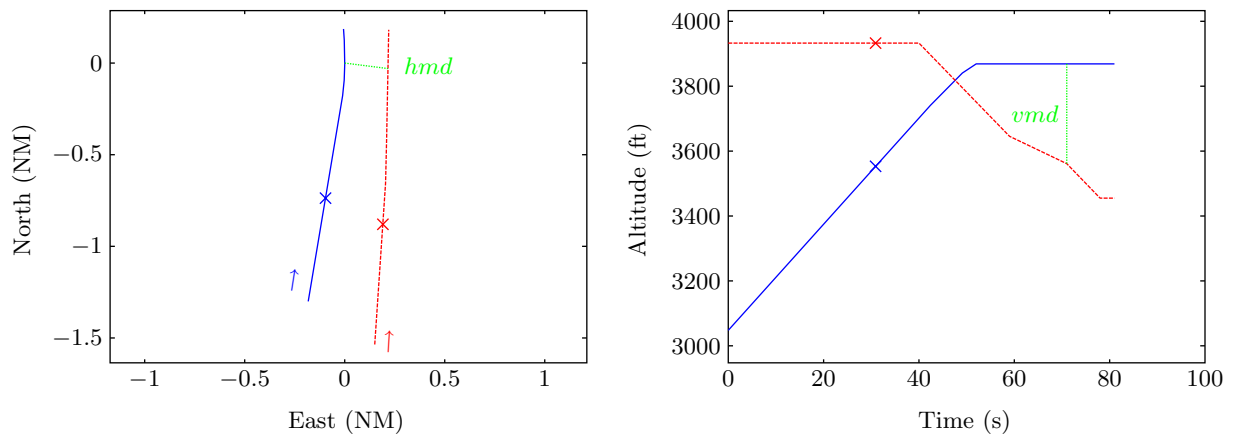


Figure 22. An example slow closing encounter generated from the encounter model. The crosses denote the location of the aircraft 40 s prior to TCA.

This page intentionally left blank.

6. SAFETY EVALUATION TECHNIQUES

This report describes how to randomly generate a set of encounter situations that reflect those that were observed in the national airspace for use in Monte Carlo simulation. Often, a safety analysis involves evaluating the effect a proposed risk mitigation measure, such as TCAS equipage, has on overall safety. Running millions of randomly generated encounter situations allows us to estimate the probability that an encounter will result in an NMAC, or $P(\text{nmac} \mid \text{enc})$.

6.1 SAFETY METRICS

Given $P(\text{nmac} \mid \text{enc})$ and the rate at which encounters occur, λ_{enc} , one can then estimate the rate at which NMACs will occur in the airspace using

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

The rate λ_{enc} depends on the regions and types of operations that aircraft fly, and a value for discrete-code aircraft will differ from that for 1200-code aircraft. Collision avoidance studies will typically be focused on new technologies for discrete-code aircraft, and so we are concerned here with λ_{enc} for that type of aircraft. Taking the NAS as a whole, we estimated λ_{enc} empirically by dividing the number of encounters identified from the radar data by the total flight hours accumulated by discrete-code aircraft. Excluding encounters and flight hours flown in or over special use airspace, we found this estimate to be $406728/1.82 \times 10^7 = 0.021$ encounters per discrete-code flight hour. Simply put, then, one could consider each simulated encounter to occur every 48 flight hours. Many of these encounters are benign and would not result in an NMAC or even in collision avoidance maneuvers, but it is important to simulate them to ensure that the collision avoidance system does not induce problems. It must also be emphasized that this is a national average over all altitudes, airspaces, and times. A more focused study may need to target such an estimate at specific regions, airspaces, or times. Alternatively, a similar estimate could be made for 1200-code aircraft using the appropriate encounter and flight hour counts.

Another common metric that incorporates $P(\text{nmac} \mid \text{enc})$ is the risk ratio, which compares the $P(\text{nmac} \mid \text{enc})$ resulting from equipping one or more aircraft during the encounter with TCAS versus the nominal encounter condition where neither plane is equipped with TCAS [23]. A typical safety study consists of one or more Monte Carlo simulations that permits an unbiased estimate of $P(\text{nmac} \mid \text{enc}, \text{equip})$. The variable “equip” signifies the equipage on the two aircraft involved in the simulation. For instance, consider three equipage combinations for the two aircraft involved in the encounter: Unequipped/Unequipped (UU), TCAS/Unequipped (TU), and TCAS/TCAS (TT). The equipage option “unequipped” signifies that the aircraft has an altitude reporting transponder but no TCAS, and the option “TCAS” signifies the aircraft is equipped with TCAS Version 7. Using samples from our encounter model and our simulation environment, we can calculate the risk ratios including

$$RR_1 = \frac{P(\text{nmac} \mid \text{enc}, \text{TU})}{P(\text{nmac} \mid \text{enc}, \text{UU})} \quad \text{and} \quad RR_2 = \frac{P(\text{nmac} \mid \text{enc}, \text{TT})}{P(\text{nmac} \mid \text{enc}, \text{UU})}.$$

If we assume our intruder is unequipped but is carrying an altitude-reporting transponder, equipping ourselves with TCAS will reduce NMAC risk by $(1 - RR_1) \cdot 100$ percent. If two unequipped aircraft in an encounter had been equipped with TCAS, the NMAC risk would have been reduced by $(1 - RR_2) \cdot 100$ percent.

It is also possible to vary other encounter parameters, including transponder models, pilot response models, visual acquisition models, and collision avoidance systems in simulation, and compare effects on risk ratio. Other safety metrics have also been used successfully [23], as well as metrics designed to evaluate alerting efficiency [24, 25].

6.2 IMPORTANCE SAMPLING

Directly sampling from the Bayesian network (Section 4) provides an unbiased estimate of the probability that an encounter results in an NMAC:

$$P(\text{nmac} \mid \text{enc}) \approx \frac{1}{N} \sum_i P(\text{nmac} \mid \mathbf{x}^{(i)}, \text{enc}),$$

where N is the number of encounters, $\mathbf{x}^{(i)}$ represents the i th sample, and $P(\text{nmac} \mid \mathbf{x}^{(i)}, \text{enc})$ is the probability that the i th encounter situation resulted in an NMAC as evaluated in simulation. The problem with this direct sampling approach is that the vast majority of samples will involve large horizontal and vertical miss distances, where an NMAC is unlikely to result with or without a collision avoidance system. Simulating encounters that are extremely unlikely to result in an encounter wastes computation.

A variance-reduction technique known as importance sampling [26] allows us to focus on the encounters that are likely to result in an NMAC by adjusting the sampling distribution and weighting the samples appropriately. We begin by generating a collection of N samples from the Bayesian network as before. We then discard the values for hmd and vmd that were generated by the Bayesian network and replace them with new values sampled from different distributions. The probability that an encounter results in an NMAC may then be estimated as follows:

$$P(\text{nmac} \mid \text{enc}) \approx \frac{1}{N} \sum_i P(\text{nmac} \mid hmd^{(i)}, vmd^{(i)}, \mathbf{y}^{(i)}, \text{enc}) \frac{p(hmd^{(i)}, vmd^{(i)} \mid \mathbf{y}^{(i)})}{f(hmd^{(i)})g(vmd^{(i)})}, \quad (7)$$

where $\mathbf{y}^{(i)}$ represents the values of all the encounter variables except hmd and vmd in the i th sample, $f(hmd^{(i)})$ is the sampling density over hmd , and $g(vmd^{(i)})$ is the sampling density over vmd . The probability density $p(hmd^{(i)}, vmd^{(i)} \mid \mathbf{y}^{(i)})$ comes from the Bayesian network. In the Bayesian network used in our encounter model (Figure 6), this density may be computed as follows (see Equation J-1):

$$p(hmd^{(i)}, vmd^{(i)} \mid \mathbf{y}^{(i)}) = p(vmd^{(i)} \mid L^{(i)}, \dot{h}_1^{(i)}, \dot{h}_2^{(i)})p(hmd^{(i)} \mid L^{(i)}, v_1^{(i)}, v_2^{(i)}).$$

So long as the weighting is used as specified in Equation 7, any sampling density that is nonzero over the positive real domain can work for generating hmd and vmd . Ideally, $f(hmd)g(vmd)$

approximates $P(\text{nmac} \mid hmd, vmd, \text{enc})$ to maximally reduce the variance of the estimate [26]. An example of a sampling distribution for hmd that may be appropriate for a TCAS safety study is

$$hmd \sim \left\{ \begin{array}{ll} U(0, 0.08229) & \text{if } X < 0.95 \\ U(0.08229, 3) & \text{otherwise} \end{array} \right\}$$

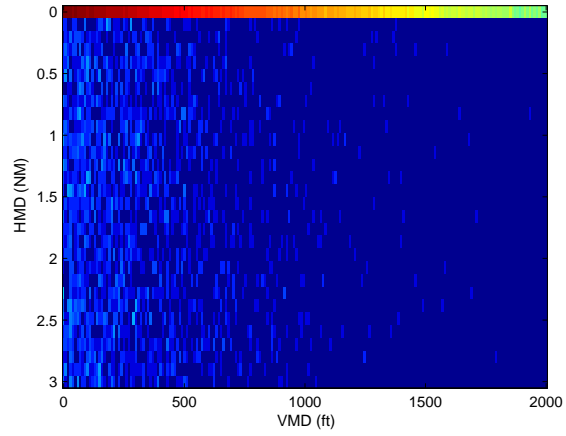
where $U(a, b)$ represents the uniform distribution between a and b , and $X \sim U(0, 1)$. This corresponds to a distribution where 95% of the time a random sample will have hmd below 0.08229 NM (500 ft). An example of a sampling distribution for vmd is

$$vmd \sim \text{Exp}(500),$$

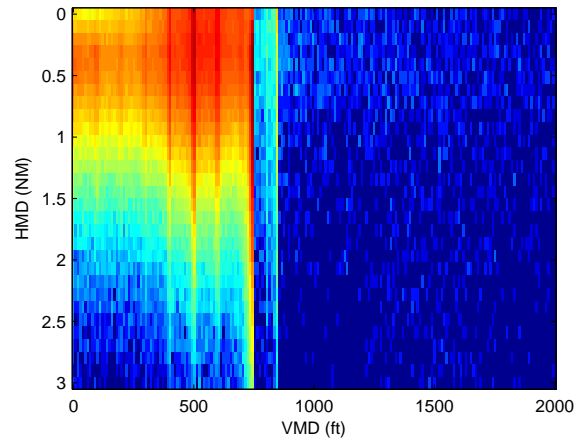
where $\text{Exp}(\lambda)$ represents the exponential distribution with parameter λ . Using the exponential distribution with λ set to 500 ft, approximately 18% of sampled vmd values are under 100 ft. By combining the independent distributions of hmd and vmd , approximately 17% of samples will be NMAC encounters in the absence of avoidance maneuvering.

The resulting joint probability distribution is shown in Figure 23a, while Figure 23b shows the actually-observed joint distribution for hmd and vmd from the radar data. These figures demonstrate that the vast majority of synthetically-generated encounters have hmd smaller than 500 ft. While this distribution is very different than the observed distribution, it more closely approximates the distribution of encounters that a collision avoidance system will be expected to resolve. The vertical bands in Figure 23b are likely a byproduct of 100 ft altitude quantization and air traffic control.

The technique described above corrects for the fact that more samples are generated with low hmd and vmd values compared to observed events. The result is a more efficient simulation (by focusing on low separation events that probe collision avoidance performance) that will still provide an accurate estimate of overall collision risk matching that observed in the radar data.



(a) Sampled hmd and vmd (log scale).



(b) Observed hmd and vmd (log scale).

Figure 23. Joint probability distributions for hmd and vmd .

7. SUMMARY

This document 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 states based on a Markov model where the probability of the next state depends only upon the current state. By using dynamic Bayesian networks, we can leverage conditional independence between variables to greatly reduce the number of parameters that would otherwise need to be estimated in a complete Markov state transition matrix. We can learn the structure of the dynamic Bayesian network by maximizing the posterior probability of the network structure given the radar data that were collected.

We processed orders of magnitude more radar data, spanning nine months from more than 120 sensors across the continental United States, than previous encounter modeling efforts. The volume of data used for the model ensures a broad and representative variety of encounter situations, including those in high density terminal areas as well as en route regions of the national airspace. We chose the best model network structure using Bayesian model selection to optimally balance model complexity and the amount of data we had available. We sampled from the model to produce encounter situations to validate the quality of the model. Finally, we described how to set up an encounter in simulation and provided guidance on techniques for conducting safety studies using the model.

The correlated encounter model presented in this report is intended to be used to evaluate potential revisions to TCAS. In addition, the correlated encounter model, along with an uncorrelated encounter model developed in parallel to model noncooperative aircraft, will be used to generate encounter situations for use in Monte Carlo safety analyses of future collision avoidance systems for manned and unmanned aircraft.

This page intentionally left blank.

APPENDIX A MODEL PARAMETERS

This appendix describes the sufficient statistics, N_{ijk} (see Appendix J), used to estimate the conditional probabilities associated with the initial and transition distributions. These sufficient statistics are based on the 411,867 encounters captured from 1 December 2007 to 31 August 2008. 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, contains the following model parameters:

- **Variable labels:** A quoted, comma-delimited list specifies the variable labels as would be used by \LaTeX . There are different variable labels for the initial network and the transition network. The ordering of the variables in this list determines the ordering of the variables in the other tables. Note that the ordering of the variable labels does not necessarily correspond to the order in which they are sampled; a topological sort may be necessary before sampling as discussed in Appendix B.
- **Graphical structure:** A binary matrix is used to represent the graphical structure of the Bayesian network. 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 ordering of the variables are as defined in the variable labels section of the file. 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 here formally by the variable $G(i, j)$.
- **Variable instantiations:** For each network, a list of integers specifies the number of instantiations (bins) 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 between the variable bins are specified by a row of numbers. The variables A , L , χ , C_1 , and C_2 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 . Again, the variable ordering is as defined in the variable labels section of the file. One way to load the sufficient statistics into memory is to allocate an array of pointers to 2-dimensional matrices. There would be 16 matrices for the initial network and 20 matrices for the transition network. The dimensions of each matrix is $r_i \times q_i$, or the number of instantiations of the variable by the number of instantiations of the

parents of the variable (see Appendix J).² The counts may be read directly into the matrices from the file, starting with the first column of the first variable to the last column of the last variable.

Instead of reading the sufficient statistics into an array of matrices stored in memory, one can reference the elements in the parameters file directly. 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 J.³

It is important to clarify the ordering of the parental instantiations. If the variables X_1, \dots, X_n are instantiated to bins b_1, \dots, b_n , the parental instantiation of variable X_i is given by

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

For example, suppose that a variable has three parents. The first parental instantiation will assign all parents to their first bin. The second parental instantiation will assign the first parent (as defined by the ordering in the variable labels portion of the file) to its second bin and the other two parents to their first bin. The sequence continues until all of the parents are instantiated to their last bins.

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", "\chi", "\beta", "C_1", "C_2", "v_1", "v_2", "\dot v_1", "\dot v_2",
"\dot h_1", "\dot h_2", "\dot \psi_1", "\dot \psi_2", "hmd", "vmd"
# G_initial
0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0
1 0 0 0 1 1 1 1 1 1 1 1 0 0 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 1 0 0 0 1 0 1 0 0 0
0 0 0 0 0 0 0 1 0 0 0 1 0 1 0 0
0 0 0 1 0 0 0 0 1 0 0 0 1 0 1 0
0 0 0 1 0 0 1 0 0 1 0 0 0 1 1 0
0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
0 0 0 0 0 0 1 0 1 0 0 1 0 0 0 1
```

²For the transition network, note that the matrices for the variables that are not associated with time $t + 1$ are empty.

³In the transition network, q_i is as defined in Appendix J for the nodes representing variables at time $t + 1$. For the other nodes, i.e., the static variables and the variables at time t , q_i is set to zero.


```

0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 1
0 0 0 0 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 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
# r_initial
4 5 2 12 2 2 6 6 5 5 9 9 9 9 4 10
# N_initial
22504 14720 19628 141108 16473 4804 17 ...
# labels_transition
"A", "L", "\chi", "\beta", "C_1", "C_2", "v_1", "v_2", "\dot v_1", "\dot v_2",
"\dot h_1(t)", "\dot h_2(t)", "\dot \psi_1(t)", "\dot \psi_2(t)", "hmd", "vmd",
"\dot h_1(t+1)", "\dot h_2(t+1)", "\dot \psi_1(t+1)", "\dot \psi_2(t+1)"
# G_transition
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 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 0 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 0 0 0 0 0 0 1 0
0 0 0 0 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 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 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 0 0 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 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
# r_transition
4 5 2 12 2 2 6 6 5 5 9 9 9 9 4 10 9 9 9 9
# N_transition
6006 300 0 0 0 0 0 0 0 18142 566 0 0 0 0 0 0 13397 435 ...
# boundaries
*
*
*
0 30 60 90 120 150 180 210 240 270 300 330 360
*
```

```

*
50 100 200 300 400 500 600
50 100 200 300 400 500 600
-5 -2 -0.25 0.25 2 5
-5 -2 -0.25 0.25 2 5
-5000 -3000 -2000 -1000 -400 400 1000 2000 3000 5000
-5000 -3000 -2000 -1000 -400 400 1000 2000 3000 5000
-8 -6 -3.5 -1 -0.25 0.25 1 3.5 6 8
-8 -6 -3.5 -1 -0.25 0.25 1 3.5 6 8
0 0.1 0.5 1 3
0 100 200 300 400 500 600 700 800 900 6000
# resample_rates
0 0 0 0 0 0 0 0 0 0.0521451 0.0542133 0.0796733 0.0829421 0 0

```

Software is available from Lincoln Laboratory that parses this parameter information and generates random encounter features.

APPENDIX B ENCOUNTER GENERATION

This appendix explains how to generate an encounter from the model. Software for parsing the parameters file (Appendix A) and generating trajectories is available from Lincoln Laboratory.

B.1 INITIAL NETWORK SAMPLING

The first step in generating a random trajectory using the model is to sample from the Bayesian network representing the initial state distribution. To sample from a Bayesian network, as explained in Appendix J, one must first 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 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0
1 0 0 0 1 1 1 1 1 1 1 1 0 0 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 1 0 0 0 1 0 1 0 0 0
0 0 0 0 0 0 0 1 0 0 0 1 0 1 0 0
0 0 0 1 0 0 0 0 1 0 0 0 1 0 1 0
0 0 0 1 0 0 1 0 0 1 0 0 0 1 1 0
0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
0 0 0 0 0 0 1 0 1 0 0 1 0 0 0 1
0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 1
0 0 0 0 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 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0

```

One possible topological sort is

$$X_3, X_2, X_1, X_5, X_{11}, X_6, X_{12}, X_{16}, X_8, X_7, X_{15}, X_9, X_{13}, X_{10}, X_{14}, X_4.$$

corresponding to variables (as specified in the parameters file):

$$\chi, L, A, C_1, \dot{h}_1, C_2, \dot{h}_2, vmd, v_2, v_1, hmd, \dot{v}_1, \dot{\psi}_1, \dot{v}_2, \dot{\psi}_2, \beta.$$

With the nodes topologically sorted, we begin by sampling the first variable, X_3 . As specified in the parameters file, the third variable X_3 corresponds to bearing χ . Equation J-5, reproduced below, shows how to produce a random sample from X_3 :

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

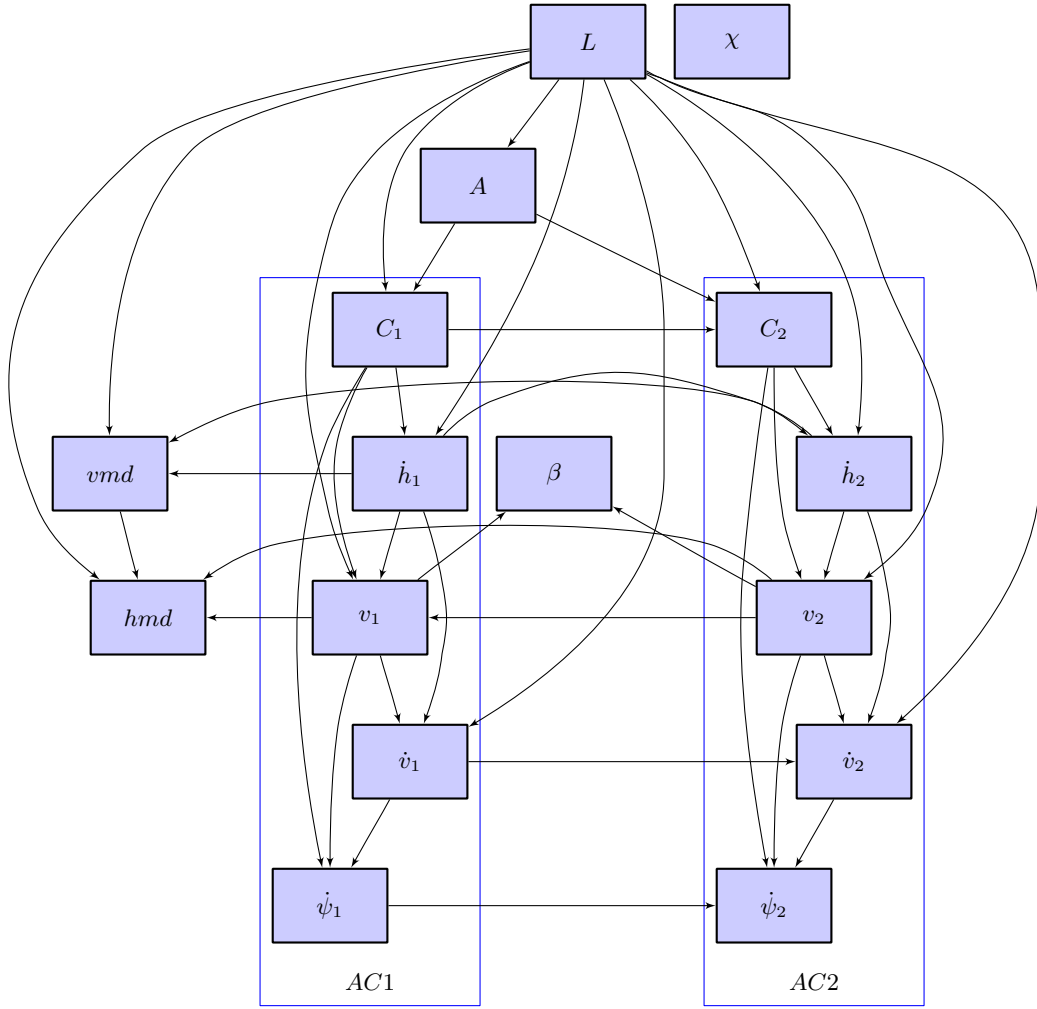


Figure B-1. A graphical representation of the initial network.

In other words, the probability of selecting a particular bearing is proportional to the prior (α_{ijk}) plus the frequency that bearing appeared in the data (N_{ijk}). Following Cooper and Herskovits [27], 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.

TABLE B-1

$N(\chi)$	
χ	
[270, 90)	[90, 270)
206529	205338

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

- $P(\chi = [270, 90)) = (206529 + 1)/(206529 + 1 + 205338 + 1) = 0.5014$
- $P(\chi = [90, 270)) = (205338 + 1)/(206529 + 1 + 205338 + 1) = 0.4986$

We use a random number generator to choose the bin for χ . For this example, let us choose [270, 90), meaning that the intruder approached from the front.

The next step is to instantiate the second variable in our topological sort, L , which is altitude layer. Choosing a random instantiation for L works just like χ . The table for L is as follows:

TABLE B-2

$N(L)$				
L				
1	2	3	4	5
208401	175683	28710	8657	5916

Suppose that we randomly choose $L = 3$. The next variable to select in our topological sort is X_1 , which corresponds to airspace class, A . Choosing a random bin for A is a little more

complicated because it depends upon the value we chose for L . We select A using the row in the table (highlighted) that is consistent with the value we selected for L .

TABLE B-3

$N(A L)$				
L	A			
	B	C	D	O
1	22504	14720	19628	141108
2	16473	4804	17	150972
3	899	12	0	27293
4	0	0	0	7718
5	0	0	0	5719

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. 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, 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:

$$A, L, \chi, \beta, C_1, C_2, v_1, v_2, \dot{v}_1, \dot{v}_2, \dot{h}_1(t), \dot{h}_2(t), \dot{\psi}_1(t), \dot{\psi}_2(t), hmd, vmd, \dot{h}_1(t+1), \dot{h}_2(t+1), \dot{\psi}_1(t+1), \dot{\psi}_2(t+1).$$

The parameters file also defines the following graphical structure for the network:

```

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 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 0 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 0 0 0 0 0 1 0
0 0 0 0 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 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 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 0 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 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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 already topologically sorted.

We are only interested in assigning new values to the dynamic variables, namely $\dot{h}_1(t+1)$, $\dot{\psi}_1(t+1)$, $\dot{h}_2(t+1)$, and $\dot{\psi}_2(t+1)$. The process is similar to the process used to sample from the initial network. First, we consult the table of sufficient statistics for $\dot{h}_1(t+1)$. This table of conditional counts is shown below.

TABLE B-4

$$N(\dot{h}_1(t+1) \mid L, \dot{h}_1(t))$$

L	$\dot{h}_1(t)$	$\dot{h}_1(t+1)$								
		$[-5k, -3k)$	$[-3k, -2k)$	$[-2k, -1k)$	$[-1k, -0.4k)$	$[-0.4k, 0.4k)$	$[0.4k, 1k)$	$[1k, 2k)$	$[2k, 3k)$	$[3k, 5k]$
1	$[-5k, -3k)$	6006	300	0	0	0	0	0	0	0
2	$[-5k, -3k)$	18142	566	0	0	0	0	0	0	0
3	$[-5k, -3k)$	13397	435	0	0	0	0	0	0	0
4	$[-5k, -3k)$	16458	530	0	0	0	0	0	0	0
5	$[-5k, -3k)$	12807	483	0	0	0	0	0	0	0
1	$[-3k, -2k)$	189	10525	749	0	0	0	0	0	0
2	$[-3k, -2k)$	615	26782	1293	0	0	0	0	0	0
3	$[-3k, -2k)$	424	20445	862	0	0	0	0	0	0
4	$[-3k, -2k)$	668	9104	655	0	0	0	0	0	0
5	$[-3k, -2k)$	561	8013	519	0	0	0	0	0	0
1	$[-2k, -1k)$	0	398	165657	5551	0	0	0	0	0
2	$[-2k, -1k)$	0	1048	248667	7717	0	0	0	0	0
3	$[-2k, -1k)$	0	639	82260	2349	0	0	0	0	0
4	$[-2k, -1k)$	0	817	21416	894	0	0	0	0	0
5	$[-2k, -1k)$	0	578	11286	348	0	0	0	0	0
1	$[-1k, -0.4k)$	0	0	3639	801708	15122	0	0	0	0

Continued on next page...

TABLE B-4. Continued

L	$\dot{h}_1(t)$	$\dot{h}_1(t+1)$								
		[-5k, -3k]	[-3k, -2k]	[-2k, -1k]	[-1k, -0.4k]	[-0.4k, 0.4k]	[0.4k, 1k]	[1k, 2k]	[2k, 3k]	[3k, 5k]
2	[-1k, -0.4k)	0	0	4072	657284	13994	0	0	0	0
3	[-1k, -0.4k)	0	0	1252	106552	2938	0	0	0	0
4	[-1k, -0.4k)	0	0	1056	32504	1233	0	0	0	0
5	[-1k, -0.4k)	0	0	489	12855	1016	0	0	0	0
1	[-0.4k, 0.4k)	0	0	0	15341	7698759	10800	0	0	0
2	[-0.4k, 0.4k)	0	0	0	10930	6557002	9246	0	0	0
3	[-0.4k, 0.4k)	0	0	0	1937	943084	1658	1	0	0
4	[-0.4k, 0.4k)	0	0	1	1432	207205	1256	0	0	0
5	[-0.4k, 0.4k)	0	0	0	1191	193913	530	0	0	0
1	[0.4k, 1k)	0	0	0	0	13424	797774	3517	0	0
2	[0.4k, 1k)	0	0	0	0	9187	605396	3802	0	0
3	[0.4k, 1k)	0	0	0	0	1377	75409	1231	0	0
4	[0.4k, 1k)	0	0	0	0	1367	34763	914	0	0
5	[0.4k, 1k)	0	0	0	0	541	17814	344	0	0
1	[1k, 2k)	0	0	0	0	0	2846	126712	1082	0
2	[1k, 2k)	0	0	0	0	0	2942	174736	2059	0
3	[1k, 2k)	0	0	0	0	0	944	73743	1017	0
4	[1k, 2k)	0	0	0	0	0	1011	25676	565	0
5	[1k, 2k)	0	0	0	0	1	418	10509	189	0
1	[2k, 3k)	0	0	0	0	0	0	543	13817	396
2	[2k, 3k)	0	0	0	0	0	0	1161	56998	1189
3	[2k, 3k)	0	0	0	0	0	0	685	34622	573
4	[2k, 3k)	0	0	0	0	0	0	592	8970	346
5	[2k, 3k)	0	0	0	0	0	0	204	2357	132
1	[3k, 5k]	0	0	0	0	0	0	0	218	4967
2	[3k, 5k]	0	0	0	0	0	0	0	565	25641
3	[3k, 5k]	0	0	0	0	0	0	0	327	13835
4	[3k, 5k]	0	0	0	0	0	0	0	356	8393
5	[3k, 5k]	0	0	0	0	0	0	0	136	2997

We are only interested in the row that is consistent with the variable assignment made so far. After randomly assigning a value to $\dot{h}_1(t+1)$ given an assignment to L and $\dot{h}_1(t)$, we continue assigning values to the other variables $\dot{\psi}_1(t+1)$, $\dot{h}_2(t+1)$, and $\dot{\psi}_2(t+1)$. The process is repeated for each time step. The length of the trajectory is 50 s.

Sampling from the Bayesian networks produces a sequence of assignments of variables to bins. As Section 4 describes, 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 mean rates specified in the parameters file:

0 0 0 0 0 0 0 0 0 0.0521451 0.0542133 0.0796733 0.0829421 0 0

Most of the variables have zero as their rates because they do not change during the course of the trajectory. The non-zero elements specify the probabilities with which \dot{h}_1 , \dot{h}_2 , $\dot{\psi}_1$, and $\dot{\psi}_2$ change within bins at each time step. 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 probabilities.

Once the initial conditions and a series of control variables (\dot{h}_1 , \dot{h}_2 , $\dot{\psi}_1$, and $\dot{\psi}_2$) have been selected, the aircraft trajectories are constructed or simulated using an appropriate dynamic model.

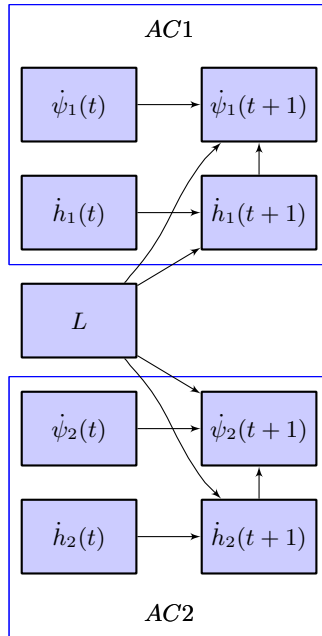


Figure B-2. A graphical representation of the transition network.

The control variables are assumed to be held constant across each 1 s time step.

This page intentionally left blank.

APPENDIX C VALIDATION

This appendix describes a set of sample encounters that may be used as validation for those wishing to implement their own dynamic simulation models. Several text files, named `cor_enc.txt`, `cor_ac1.txt`, and `cor_ac2.txt`, are available from Lincoln Laboratory and contain data for 50 sample encounters. Each encounter is 50 s in length and was generated by first sampling model parameters from the Bayesian network and then initializing and simulating the aircraft trajectories using Lincoln Laboratory’s dynamic simulation model. Note that minimum horizontal separation for the sample encounters always occurs at 40 s simulation time—none of the provided encounters required that the encounter start more than 40 s prior to TCA.

The first text file, `cor_enc.txt`, is tab-delimited and contains variables for 50 encounters sampled from the initial Bayesian network. The first column corresponds to a unique `id` number. The following 16 columns correspond to the values sampled from the initial Bayesian network for the variables: A , L , β , χ , C_1 , C_2 , v_1 , v_2 , \dot{v}_1 , \dot{v}_2 , $\dot{\psi}_1$, $\dot{\psi}_2$, \dot{h}_1 , \dot{h}_2 , hmd , and vmd . The final column corresponds to the altitude of AC1 at TCA—typically this value is determined by randomly sampling within the altitude layer of the encounter, but is provided here so that a direct comparison may be made.

The other text files (`cor_ac1.txt` and `cor_ac2.txt`) are also tab-delimited and provide dynamic variables sampled from the dynamic Bayesian network and position time histories for each aircraft in the encounter constructed and simulated by Lincoln Laboratory. Each file has the following ten columns:

- **ID:** Each row begins with an `id` number. All rows with the same `id` number correspond to a single track in the encounter with the same `id` number.
- **Time t :** Time values are reported once per second in the text file.
- **Vertical rate \dot{h} :** The vertical rate is reported in ft/min. This value is sampled from the dynamic Bayesian network.
- **Airspeed acceleration \dot{v} :** The airspeed acceleration is reported in kt/s and is also sampled from the dynamic Bayesian network.
- **Turn rate $\dot{\psi}$:** Turn rate is given in deg/s and is also a variable from the dynamic Bayesian network.
- **Airspeed v :** The initial airspeed (kt) is defined by sampling from the initial Bayesian network. The remaining values for $t > 0$ are outputs from Lincoln Laboratory’s dynamic simulation.
- **North position x :** Each aircraft’s north position is reported in ft. The north location of AC1 at minimum horizontal separation is $x = 0$ in the sample encounters.
- **East position y :** Each aircraft’s east position is reported in ft. The south location of AC1 at minimum horizontal separation is $y = 0$ in the sample encounters.

- **Altitude h :** The altitude of all the sample aircraft trajectories are in ft.
- **Heading ψ :** Aircraft heading is reported in degrees. In the sample encounters, AC1’s heading is zero at TCA.

The dynamic variables (\dot{h} and $\dot{\psi}$) are inputs to an aircraft dynamic model that describe how the aircraft maneuvers throughout the simulation. The variables v , x , y , and h can be used to compare outputs of a simulation. The sample encounters are constructed such that AC1 is located at the origin ($x = 0$ and $y = 0$), heading due north, and with an altitude defined in `cor_enc.txt` at minimum horizontal separation, which occurs at 40s simulation time. The location of AC2 at minimum horizontal separation is defined using the variables in the initial Bayesian network.

The most important features to match with the sample encounters are the relative location and heading of the two aircraft at 40s simulation time. Additionally, the remainder of the aircraft tracks should approximate the sample tracks in both the vertical and horizontal plane. Exactly matching simulation outputs with values in the text files is extremely unlikely due to the variety of acceptable aircraft models in simulation (e.g., degrees-of-freedom, transient dynamics, simulation step size, etc). Note that small differences in heading, resulting from a turn close to minimum horizontal separation, may result in large positional deviations at the beginning of the simulation—horizontal errors resulting from small differences in aircraft headings are acceptable.

Since there are numerous aircraft models that may be correctly implemented in simulation, this appendix does not define thresholds or minimum errors for validating a simulation. Instead, we only require that simulated aircraft must track steady-state inputs. For example, if sampling from the dynamic Bayesian network results in an aircraft flying straight (i.e., $\dot{\psi} = 0$ deg/s) that later transitions to $\dot{\psi} = 2$ deg/s, then the aircraft must eventually turn clockwise at 2 deg/s—the transient between $\dot{\psi} = 0$ and $\dot{\psi} = 2$ can differ from than the sample trajectories.

APPENDIX D ENCOUNTER FILTER

We define an encounter as a situation where two aircraft have lost standard separation and whose trajectories, if extrapolated into the future without pilot intervention, result in a significant chance of collision. We rely on the existing logic within TCAS to guide the way in which we extrapolate in order to identify situations in which aircraft are at significant risk of collision. Specifically, we approximate the threat detection logic found in TCAS v6.04a.

D.1 PROXIMITY TEST

The main objective of the proximity test is to reduce processing time by setting a very coarse filter that will exclude encounters for which there is little prospect of passing the TCAS-like test. If the proximity test passes, then the two tracks are subject to a TCAS-like test. The TCAS-like test includes both a closing time component and a volume component in the horizontal and vertical plane as described below. If the TCAS-like test passes, then the conflict is passed on for further processing.

If r is the horizontal range between the two aircraft, the proximity test passes if $-r/\dot{r} \leq 51$ s and $r \leq 2.5$ NM.

D.2 TCAS-LIKE TEST

The TCAS-like test passes if both the following conditions are simultaneously satisfied:

$$\begin{aligned} -\frac{r-D_m}{\dot{r}} \leq T \quad \text{or} \quad r \leq D_m \\ \text{AND} \\ -\frac{a}{\dot{a}} \leq T \quad \text{or} \quad a \leq Z . \end{aligned}$$

where Z , D_m , and T are as defined in Table D-1, a is the vertical separation between aircraft (always positive), and \dot{r} and \dot{a} must be negative if the aircraft are closing in range and altitude, respectively.

D.3 JUSTIFICATION

The TCAS-like test is intended to approximate the TCAS threat-detection logic, and most of the parameter values correspond to values in the Minimum Operational Performance Standard (MOPS) for TCAS v6.04a. A notable exception is the value for Z above FL295, which has been reduced to 850 ft from 950 ft in the MOPS, due to nuisance RAs caused by the introduction of RVSM. The TCAS-like test is constructed to ensure that, given equivalent data, the filter will always produce a threat condition when a threat condition is produced by the actual TCAS logic given the same data.

TABLE D-1

TCAS-like test parameters.

	Altitude of lower aircraft						
	AGL 1000 -FL020	FL021 -FL044	FL045 -FL094	FL095 -FL194	FL195 -FL244	FL245 -FL294	FL295 -FL500
Z	750 ft	750 ft	750 ft	750 ft	850 ft	850 ft	850 ft
D_m	0.35 NM	0.45 NM	0.55 NM	0.80 NM	0.95 NM	1.10 NM	1.50 NM
T	20 s	23 s	25 s	30 s	33 s	35 s	40 s

The threat detection logic for TCAS v6.04a is described by the pseudo-code for the function `Hit_or_miss_test` [28]. This function calls two functions in succession, `Range_test` and `Altitude_test`. In order to be declared a threat, an intruder must pass both of those tests.

`Range_test` has two components: a “tau” calculation (or closing time test) and an “incremental protection volume” calculation (or volume test). If the intruder and own aircraft are diverging, then according to the MOPS, there can still be a threat condition in range if the two aircraft are sufficiently close in range and the “range \times range-rate product is sufficiently large.” In our range test, we remove the condition on the “range \times range-rate product,” but keep the volume test. If the intruder and own aircraft are converging, then the “tau” calculation is performed. In the MOPS, the “tau” calculation is $-(r - D_m^2/r)/\dot{r} \leq T$. The subtracted term in the numerator is designed to account for slow-closure conditions where the aircraft may be close and accelerating in a turn; essentially, when range is small, it increases the likelihood that the tau test will pass. The formula in our filter is similar, except that we simply subtract D_m as the subtraction term in the numerator. For values of r greater than or equal to D_m , our filter will be larger than the filter specified in the MOPS. For values of r less than D_m , we will capture the encounter because of the volume filter, whereas with the MOPS, certain other conditions must be met (i.e., the range \times range-rate product) for the encounter to be captured.

`Altitude_test` also has a “tau” component and a volume component. In the MOPS, if the current difference in altitude is less than the “alarm volume” and if the projected vertical miss distance is less than the “alarm volume” then the intruder is declared a threat. In our filter, we remove the condition on the projected vertical miss distance. In the MOPS, if the current difference in altitude is outside the “alarm volume” a “tau” test is performed. If the vertical tau test passes, and several other conditions are met then the intruder is declared a threat. In our filter, we remove the conditions; if the vertical tau test passes, the intruder is declared a threat in the vertical plane.

It should also be noted that the Eurocontrol encounter model study had a very similar threat-declaration filter, except with much larger parameter values. While both the Eurocontrol study and our study capture many more encounters than would have been declared by the actual TCAS threat declaration algorithm, that result is acceptable for two reasons. First, although unlikely, there may be encounter geometries that induce unwanted TCAS involvement that have not been considered by the designers of the logic. Second, by designing our encounter filter in this manner,

we guarantee we keep all encounters where there is a chance of TCAS intervening, given surveillance data from the encounter. If we filter out encounters where TCAS would intervene due to a filter that was too rigid, we run the risk of not faithfully modeling the full encounter environment in which TCAS operates.

D.4 VALIDATION

We used data from the MIT Lincoln Laboratory Mode S Experimental Facility (MODSEF) sensor to validate that the filtering process was working correctly in terms of capturing TCAS RAs. MODSEF has an approximately 60 NM range and captures downlinks from TCAS-equipped aircraft within its range, as well as standard beacon replies (Mode S and Mode A/C) from transponder-equipped aircraft. We processed 52 days of data from this sensor from 1 April 2007 to 22 May 2007, and received 27 downlinks for which there were Mode S addresses for both aircraft. Only those downlinks for which there were two Mode S replies were used because those are the only encounters for which we can be sure to identify both own and intruding aircraft in the general airspace from a tracking algorithm.

Of the 27 downlinks, four were encounters for which we did not have position reports for both aircraft, which may be due to limitations of the beacon sensor or the tracking software. Of the 23 encounters for which we had both position reports and downlinks for both aircraft, we captured 20 in our encounter database using our filter. Of the 3 missed encounters,

- one occurred at the edge of coverage with insufficient reports to determine a threat condition;
- one had a 5 NM horizontal separation at TCA, perhaps a borderline “tau” threat, that did not pass even the coarse proximity filter; and
- one involved an aircraft that turned off its transponder near TCA, which did not allow our filter to establish a threat (the aircraft in question was entering formation flight).

Given this validation data, we can be confident that in areas we have coverage, we capture actual TCAS RAs.

This page intentionally left blank.

APPENDIX E

PARALLEL RUNWAY OPERATION AND FORMATION FLIGHT IDENTIFICATION

Certain frequent operations, such as parallel runway operations and formation flight, often result in “nuisance” RAs, so-called because the pilot has complete situational awareness and assumes responsibility for collision avoidance, but TCAS issues an RA. Currently, it is impossible for TCAS to determine whether or not the pilot is aware of a potential threat. However, since we observe entire tracks, we can identify benign encounters and exclude them from the model. There are two reasons why we exclude these operations:

- Previous encounter models excluded these operations when calculating the encounter rate. By including these encounters in our model, our encounter rate (and NMAC rate) would be higher than what it would be under the generally accepted definition [5,6].
- We did not want certain specialized features of these encounters, such as approach angles close to 0 deg, to be over-represented in the general encounter model.

E.1 PARALLEL RUNWAY OPERATION TEST

Parallel runway operations occur frequently at many airports in the NAS. Often, the closing time test in our encounter filter becomes satisfied when two aircraft are converging onto the final approach courses for parallel runways. Encounters occurring under these circumstances do not necessarily constitute an unintentional conflict provided the appropriate clearances have been given by air traffic control and are followed by flight crews. Due to the intentional and controlled nature of these operations, identifying parallel runway operations and excluding them from the encounter model is important.

Because manually identifying parallel runway operations can be tedious with a large data set, we developed an automated parallel runway operations classifier. Our classifier uses a database of Class B airports with parallel runways that was compiled for a NASA study [29].

Our automated parallel runway operation classifier identifies an encounter as a parallel runway operation if and only if all of the following criteria are met:

- **Airport proximity:** Both aircraft involved in the encounter must be within a 5 NM radial distance of an airport with parallel runways at TCA.
- **Encounter duration:** The two aircraft must be within 5 NM of an airport for at least 15 s.
- **Track intersection:** The two aircraft tracks involved in the encounter may not intersect in the horizontal plane.
- **Average heading difference:** The average heading between the two aircraft must not exceed 20 deg. If the aircraft are on reciprocal headings, one must be within 20 deg of the other’s reciprocal heading.

- **Average horizontal distance:** The average horizontal distance between the encounter aircraft must be between 450 and 12,000 ft. The minimum spacing of 450 ft was chosen because that was the minimum spacing between centerlines for all of the runways in the referenced NASA airport list. The upper bound of 12,000 ft was chosen since most observed encounters resembling parallel runway operations averaged a horizontal spacing of 6,000 ft and none were greater than 12,000 ft.
- **Maximum horizontal distance:** If the maximum distance between tracks is greater than 15,000 ft, the encounter will not be categorized as a parallel runway operation.
- **Groundspeed:** Aircraft tracks with groundspeed under 30 kt will not be considered a parallel runway operation. This parameter was set to eliminate hovering aircraft from the parallel approach filter.

E.2 FORMATION FLIGHT TEST

A conflict which passes the proximity test as well as the TCAS-like test must fail the formation flight test before being declared an encounter to be passed onto further filtering and processing. The formation flight test is designed to detect encounters during civil or military formation flight or refueling [30], air refueling operations, and any other operation where there is a very high chance of aircraft intentionally staying close to each other. A conflict can be considered formation flight in a two different ways:

- If the two aircraft are within 0.5 NM horizontally and 1000 ft vertically for 45 consecutive seconds.
- If the two aircraft are both going above 250 kt, are within 2 NM horizontally and 1000 ft vertically and their headings are within 20 deg of each other for 2 continuous minutes.

The second test was necessary because there are many high speed aircraft flying in this manner cross country. The test occurs after the encounter has already been tested for parallel runway operations.

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 [14], 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 project 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 [31]. 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 (9 bins total). 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; three 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 $\Delta r_{\max} = 20$ NM for a discrete code match and $\Delta r_{\max} = 1$ 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 $\Delta h_{\max} = 600$ ft and $\Delta \dot{h}_{\max} = 100$ 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 $\Delta s_{\max} = 100$ 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:

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

where we use $\Delta\psi_{\text{min}} = 3^\circ$ 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	Small Left	No Turn (0)	Small Right	Large Right
	Turn (-2)	Turn (-1)		Turn (+1)	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.

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

where we use $\Delta s_{\text{min}} = 18\text{kt}$ 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:

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

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.

F.3 COOPERATIVE ENCOUNTER IDENTIFICATION

Encounter identification is facilitated by having all global tracks sorted into 20×20 NM geographical bins. In addition, determining which sensor to employ for an encounter is guided by an ordered list of the sensor priorities associated with each bin; terminal sensor are preferred to enroute ones due to the more frequent update rate, and closer sensors are preferred to more distant ones because report accuracy is a function of range.

Two criteria are used to declare an encounter in progress:

1. **Proximity Condition:** Both tracks are currently within a sufficiently small horizontal and vertical box relative to each other.
2. **Closure Condition:** Both tracks, when projected to the point of closest horizontal approach, reach positions within a similar box, and the time to closest horizontal approach is within a sufficiently short time.

Encounter detection is performed for each global track update, at which time the track is compared to all other tracks in the appropriate bins. Proximity testing is done by predicting the bin track to the time of the updated track, using the track velocity. Closure testing predicts each track to the

time of closest approach. The equation for the time of closest approach given the current positions and velocities of two tracks is

$$t_{\text{closest}} = -\frac{(x_1 - x_2)(\dot{x}_1 - \dot{x}_2) + (y_1 - y_2)(\dot{y}_1 - \dot{y}_2)}{(\dot{x}_1 - \dot{x}_2)^2 + (\dot{y}_1 - \dot{y}_2)^2}.$$

In order to initiate an encounter, several additional tests must be satisfied:

1. The sensor of the current update must be the highest priority sensor having surveillance on both tracks; otherwise encounter initiation awaits the report from that sensor.
2. Both tracks must be at least 3 scans old to prevent spurious or split tracks from being used; because the proximity filter is quite large, this delay will not prevent the finding of true encounters.
3. The bin track must have been updated within the last half scan; this rule prevents a track number change from creating an encounter, as both reports initiating the encounter must be on the same scan.
4. The tracks must be at a minimum altitude above ground to filter out ground traffic or aircraft on parallel approach paths.

Once an encounter is initiated, only the sensor forming the encounter can be used to maintain the encounter, thereby preventing inter-sensor biases from contaminating the track trajectories.

APPENDIX G SENSOR LIST

The following table lists the RADES sensors that contributed to the model. As an example of the relative rates at which aircraft are observed by each sensor, also shown are the number of hours of flight time by transponder-equipped aircraft captured by each sensor for the month 1–31 August 2008.

TABLE G-1

Beacon hours.

60 NM range sensors

16217.4	■	ACT Waco, TX, USA (ASR-11)
40604.3	■	ADW Camp Springs (Andrews AFB), MD, USA (ASR-9)
28952.8	■	BOS Logan Intl (Boston), MA, USA (ASR-9)
24907.2	■	BUF Buffalo Intl, NY, USA (ASR-9)
47364.2	■	BWI Baltimore (BWI), MD, USA (ASR-9)
15579.7	■	COS Colorado Springs, CO, USA (ASR-11)
14123.9	■	COU Columbia MO, MO, USA (ASR-11)
16403.4	■	CRW Charleston, WV, USA (ASR-8)
40879.9	■	DCA Washington National, DC, USA (ASR-9)
38714.5	■	DOV Dover AFB, DE, USA (GPN-20)
67808.9	■	EWR Newark, NJ, USA (ASR-9)
32673.2	■	FLL Ft Lauderdale (Hollywood Intl), FL, USA (ASR-9)
15250.1	■	GRK Ft Hood, TX, USA (ASR-9)
67155	■	HPN White Plains (Westchester Co), NY, USA (ASR-9)
17830.9	■	HUF Terre Haute, IN, USA (ASR-8)
42623.4	■	IAD Washington Dulles (Chantilly), DC, USA (ASR-9)
72213.2	■	JFK New York (JFK), NY, USA (ASR-9)
53451.4	■	LAXB Los Angeles Intl - North, CA, USA (ASR-9)
12005.8	■	LFT Lafayette, LA, USA (ASR-11)
36461.6	■	MDT Harrisburg, PA, USA (ASR-9)
29431.9	■	MHT Manchester, NH, USA (ASR-9)
8997.32	■	MLU Monroe LA, LA, USA (ASR-8)
30288.1	■	MRB Martinsburg, WV, USA (ASR-9)
6422.51	■	MUO Mountain Home AFB, ID, USA (GPN-20)
3393.62	■	NQXA Key West NAS, FL, USA (GPN-27)
11674.7	■	PWM Portland (Cumberland), ME, USA (ASR-9)
3840.23	■	RCA Ellsworth AFB (Rapid City), SD, USA (GPN-20)
14636.2	■	SAV Savannah, GA, USA (ASR-8)











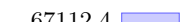











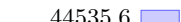


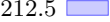












Continued on next page...

TABLE G-1. Continued

19534	SAW Marquette (KI Sawyer), MI, USA (ASR-7F)
40449.9	SCK Stockton, CA, USA (ASR-11)
19698.8	SDF Louisville, KY, USA (ASR-9)
13303.7	SJU San Juan, PR, USA (ASR-8)
13838.1	STT St. Thomas, PR, USA (ASR-8)
38673.8	SWF Newburgh (Stewart), NY, USA (ASR-9)
56754.4	XMR Cape Canaveral, FL, USA (GPN-30)
120 NM range sensors	
90734.8	NHK Patuxent River NAS, MD, USA (ASR-11)
200 NM range sensors	
68249.9	AEX Alexandria, LA, USA (AN/FPS-20A)
59934.7	AMA Amarillo, TX, USA (AN/FPS-67B)
129487	ATL Atlanta (Marietta), GA, USA (ARSR-1E)
47633.3	BAM Battle Mountain, NV, USA (ARSR-2)
79479.1	CDC Cedar City, UT, USA (ARSR-2)
128961	CPV Coopersville, MI, USA (AN/FPS-66A)
125151	DSV Buffalo (Dansville), NY, USA (ARSR-1E)
39595.4	FLX Fallon, NV, USA (AN/FPS-66A)
62429.5	FPK Salt Lake City (Francis Peak), UT, USA (ARSR-1E)
105220	FTW Ft Worth (Keller), TX, USA (ARSR-1E)
72901	GUP Gallup (Farmington), NM, USA (ARSR-2)
76080.2	HOU Houston (Ellington), TX, USA (ARSR-1E)
146558	IND Indianapolis, IN, USA (ARSR-1E)
105386	IRK Kirksville, MO, USA (ARSR-3)
165747	JOL Elwood (Joliet), IL, USA (ARSR-3)
49554	LSK Lusk, WY, USA (ARSR-2)
87451.5	MGM Montgomery, AL, USA (ARSR-1E)
164175	PIT Pittsburgh (Oakdale), PA, USA (AN/FPS-67B)
92096.9	QAS Las Vegas (Angel Peak), NV, USA (AN/FPS-20E)
170350	QBE Roanoke (Bedford), VA, USA (ARSR-3)
142799	QBN Binns Hall, VA, USA (ARSR-3)
93569.8	QBZ Oskaloosa, KS, USA (ARSR-2)
166856	QCF Du Bois (Clearfield), PA, USA (ARSR-3)
35849.4	QCK Boise (Cascade), ID, USA (ARSR-2)
164525	QDB Cleveland (Brecksville), OH, USA (ARSR-1E)
134944	QDT Detroit (Canton), MI, USA (ARSR-1E)
132408	QHA Cummington, MA, USA (AN/FPS-67B)
64543.5	QHB St. Albans, VT, USA (AN/FPS-67B)









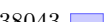
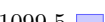




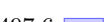







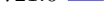
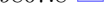
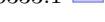
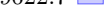
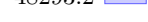
Continued on next page...

TABLE G-1. Continued

122637		QHZ Horicon, WI, USA (ARSR-2)
29792.4		QJB Gettysburg, SD, USA (AN/FPS-67B)
70133.4		QJE Minneapolis (Apple Valley), MN, USA (ARSR-1E)
102560		QJO Arlington, IA, USA (ARSR-3)
17794.2		QJQ San Juan (Pico del Este), PR, USA (AN/FPS-20E)
132673		QNK Lincolnton, GA, USA (ARSR-3)
87630.8		QNM Newport, MS, USA (ARSR-3)
115906		QOJ Nashville (Joelton), TN, USA (ARSR-1E)
111650		QPC Haleyville, AL, USA (AN/FPS-67B)
90423.3		QPK Denver (Parker), CO, USA (ARSR-1E)
209130		QPL The Plains, VA, USA (ARSR-3)
67112.4		QRB Citronelle, AL, USA (ARSR-2)
183623		QRC Benton, PA, USA (AN/FPS-67B)
148731		QRI Lynch, KY, USA (ARSR-2)
116607		QRL Raleigh (Benson), NC, USA (ARSR-1E)
128782		QRM Charlotte (Maiden), NC, USA (ARSR-1E)
55493.1		QSA Albuquerque (West Mesa), NM, USA (AN/FPS-66A)
30485.7		QSI Lovell, WY, USA (ARSR-2)
87120.5		QSR Boron, CA, USA (AN/FPS-67B)
155815		QTZ La Grange, IN, USA (ARSR-1E)
135298		QUZ Hanna City, IL, USA (AN/FPS-67B)
56402.3		QWC Mesa Rica, NM, USA (ARSR-1E)
163599		QWO London, OH, USA (ARSR-1E)
85600.5		QXR Russellville, AR, USA (AN/FPS-64A)
44535.6		QXS Odessa (Andrews), TX, USA (ARSR-1E)
94609.2		QYB Byhalia (Memphis), MS, USA (ARSR-1E)
94265.8		QYS Rogers, TX, USA (ARSR-1E)
48212.5		SEA Seattle (Ft Lawton), WA, USA (ARSR-1E)
116767		STL St. Louis, MO, USA (ARSR-1E)
45191.8		SVC Silver City, NM, USA (ARSR-2)
250 NM range sensors		
90581.8		AJO Ajo, AZ, USA (ARSR-4)
18121.8		BAR Barrington, NS, Canada (AN/FPS-117)
88287.7		CTY Cross City, FL, USA (ARSR-4)
40782.9		DMN Deming (Magdalena Peak), NM, USA (ARSR-4)
83085.1		LCH Lake Charles, LA, USA (ARSR-4)
103691		MLB Melbourne, FL, USA (ARSR-4)
93241.5		NEN Jacksonville (White House Fl), FL, USA (ARSR-4)
55439.7		NEW New Orleans (Slidell), LA, USA (ARSR-4)

Continued on next page...

TABLE G-1. Continued

41529.2		NQX Key West, FL, USA (ARSR-4)
103126		NSD San Clemente Island, CA, USA (ARSR-4)
80857.4		PAM Panama City (Tyndall), FL, USA (ARSR-4)
86499.7		PRB Paso Robles, CA, USA (ARSR-4)
101238		QEA North Truro, MA, USA (ARSR-4)
97525		QFN Ft Green, FL, USA (ARSR-4)
223922		QIE Gibbsboro, NJ, USA (ARSR-4)
74284.5		QJA Empire, MI, USA (ARSR-4)
38043		QJD Nashwauk, MN, USA (ARSR-4)
31099.5		QKW Makah, WA, USA (ARSR-4)
75389.8		QM8 Tamiami, FL, USA (ARSR-4)
109673		QMV Mill Valley, CA, USA (ARSR-4)
70600.4		QNA Morales, TX, USA (ARSR-4)
35167.4		QNW Eagle Peak, TX, USA (ARSR-4)
45497.6		QOM King Mountain, TX, USA (ARSR-4)
108370		QRJ Charleston (Jedburg), SC, USA (ARSR-4)
99078		QRW Mt. Laguna, CA, USA (ARSR-4)
188484		QVH Riverhead, NY, USA (ARSR-4)
116153		QVR Oceana, VA, USA (ARSR-4)
25634.5		QWA Watford City, ND, USA (ARSR-4)
120107		QXU Utica (Remsen), NY, USA (ARSR-4)
40721.6		QYA Bucks Harbor, ME, USA (ARSR-4)
29807.8		QYD Caribou, ME, USA (ARSR-4)
35333.1		QZA Oilton, TX, USA (ARSR-4)
29622.7		QZZ Rainbow Ridge, CA, USA (ARSR-4)
48293.2		RSG Rocksprings, TX, USA (ARSR-4)
54919.4		SLE Salem (Laurel Mountain), OR, USA (ARSR-4)

APPENDIX H ENCOUNTER DENSITY

The figures in this appendix show the density of encounters in the radar data used to construct the model. Density is measured in total number of encounters per pixel for the period 1 December 2007 to 31 August 2008. 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 NM² at the northernmost latitude to 92 NM² at the southernmost latitude. The highest density pixel encompasses Newark, NJ and New York, NY and had almost 6000 encounters over all altitude layers over the course of the data collection time, with the highest density altitude layer between 1000 ft AGL and 3000 ft AGL.

It is important to emphasize that this density map is the observed density from the radar data set. Factors such as terrain and curvature of the earth prevent the observation of some aircraft, particularly low flying aircraft such as those flying VFR. 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.

Not surprisingly, high encounter density occurs in areas of high traffic density, such as in the northeast corridor, Florida, and near busy airports around the country. Certain areas on the maps are worth further attention; in particular, the relationship between Special Use Airspace (SUA) and encounter location is especially complex. Encounters are concentrated in certain corridors between SUAs. The presence of SUAs may increase traffic density in areas which would otherwise have sparse traffic, and encounter rate is likely a function of airspace density. In addition, in upper altitude layers, such as between FL180 and FL290 and above FL290, the majority of encounters occur within the geographic boundaries of SUAs, but above their altitude limits. Since it is impossible to determine from our data whether an aircraft is military, it is unknown whether the majority of these encounters represent cases of military exercises breaching the boundaries of the SUA (which are undesired encounters for our model), or whether the conflicts are due to military aircraft coming into conflict with aircraft passing overhead (which are relevant encounters for our model).

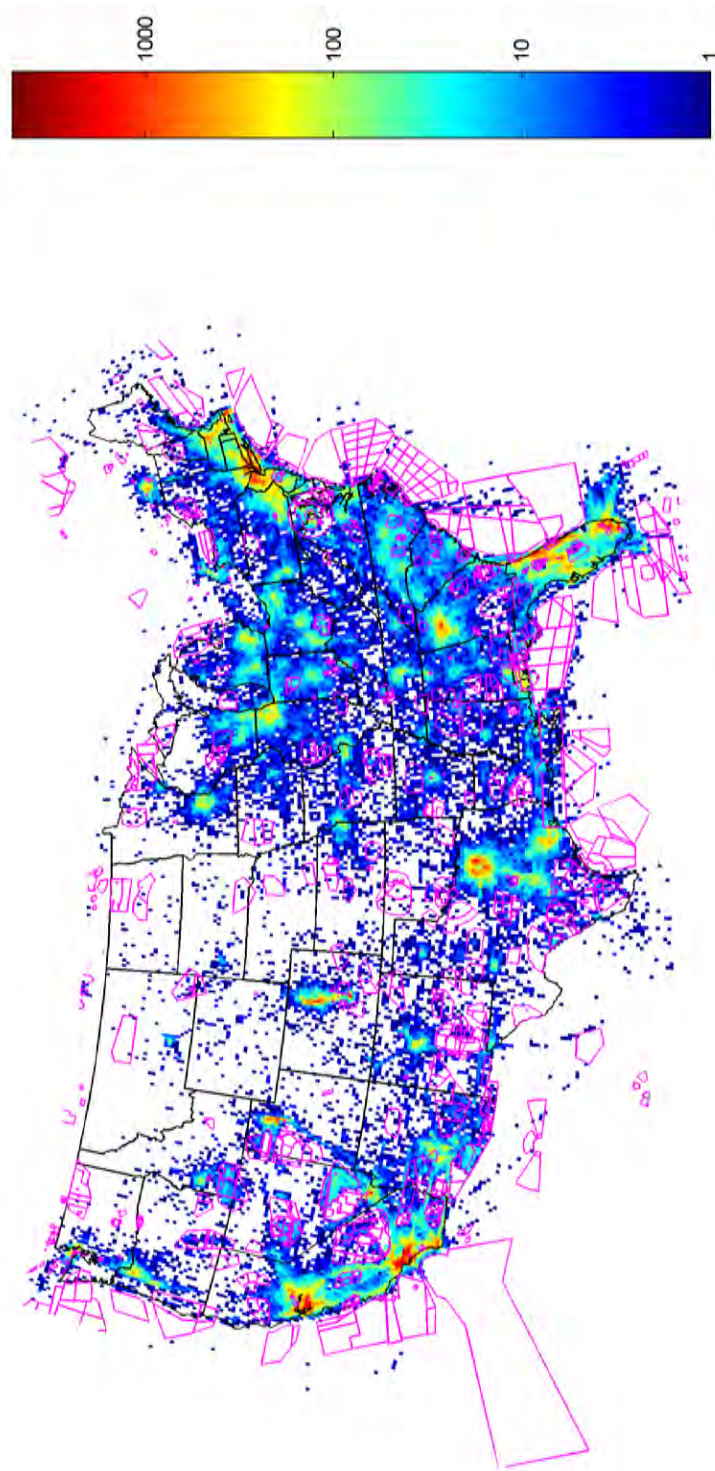


Figure H-1. Cumulative number of encounters per cell over all altitude layers, 1 December 2007 to 31 August 2008.

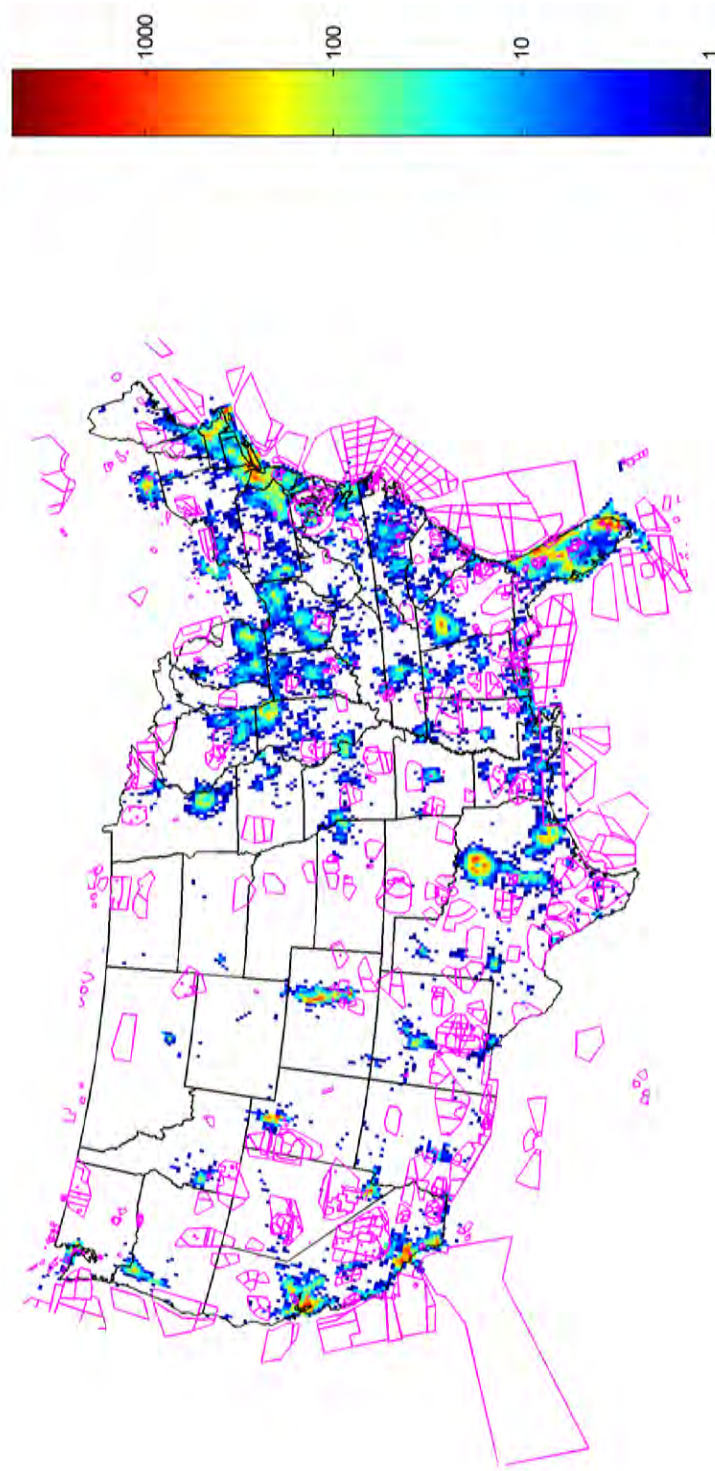


Figure H-2. Cumulative number of encounters per cell between 1000 and 3000 ft AGL, 1 December 2007 to 31 August 2008.

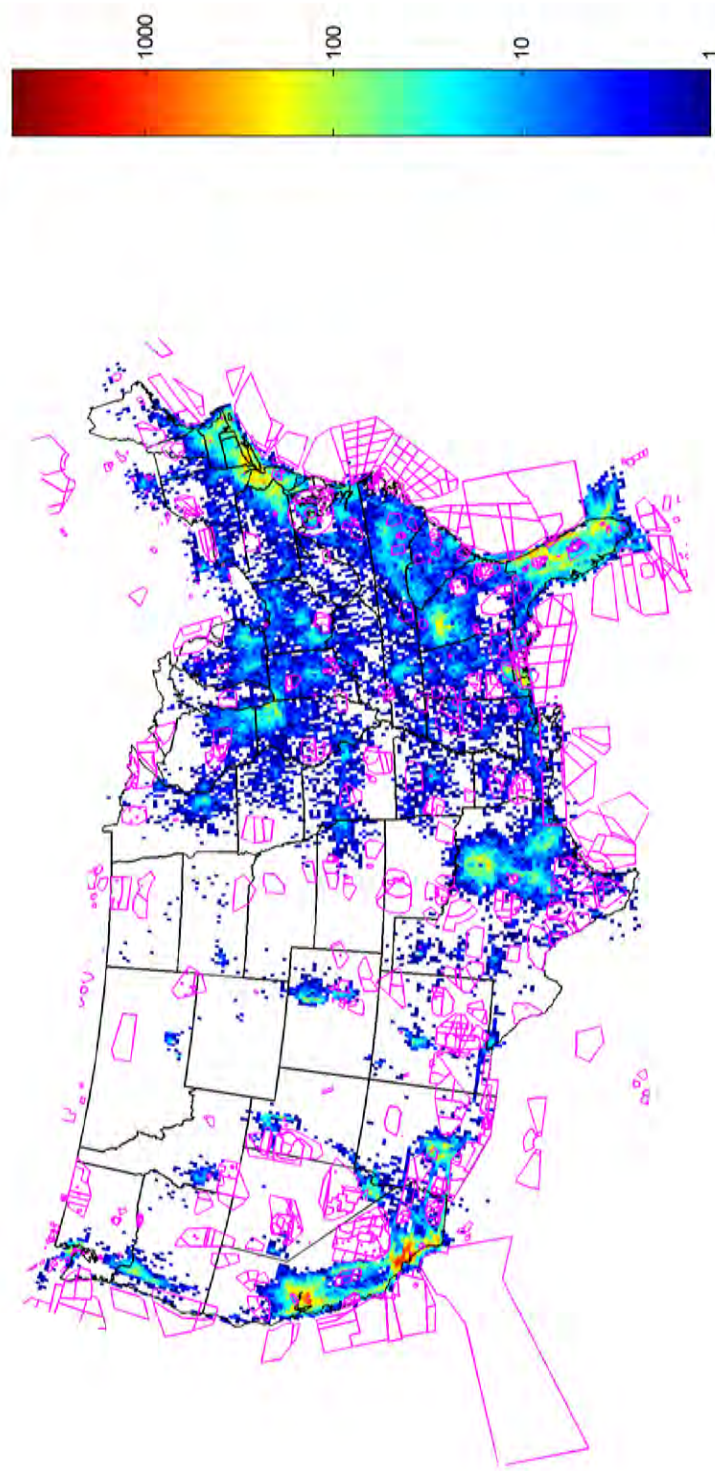


Figure H-3. Cumulative number of encounters per cell between 3000 ft AGL and 10,000 ft MSL, 1 December 2007 to 31 August 2008.

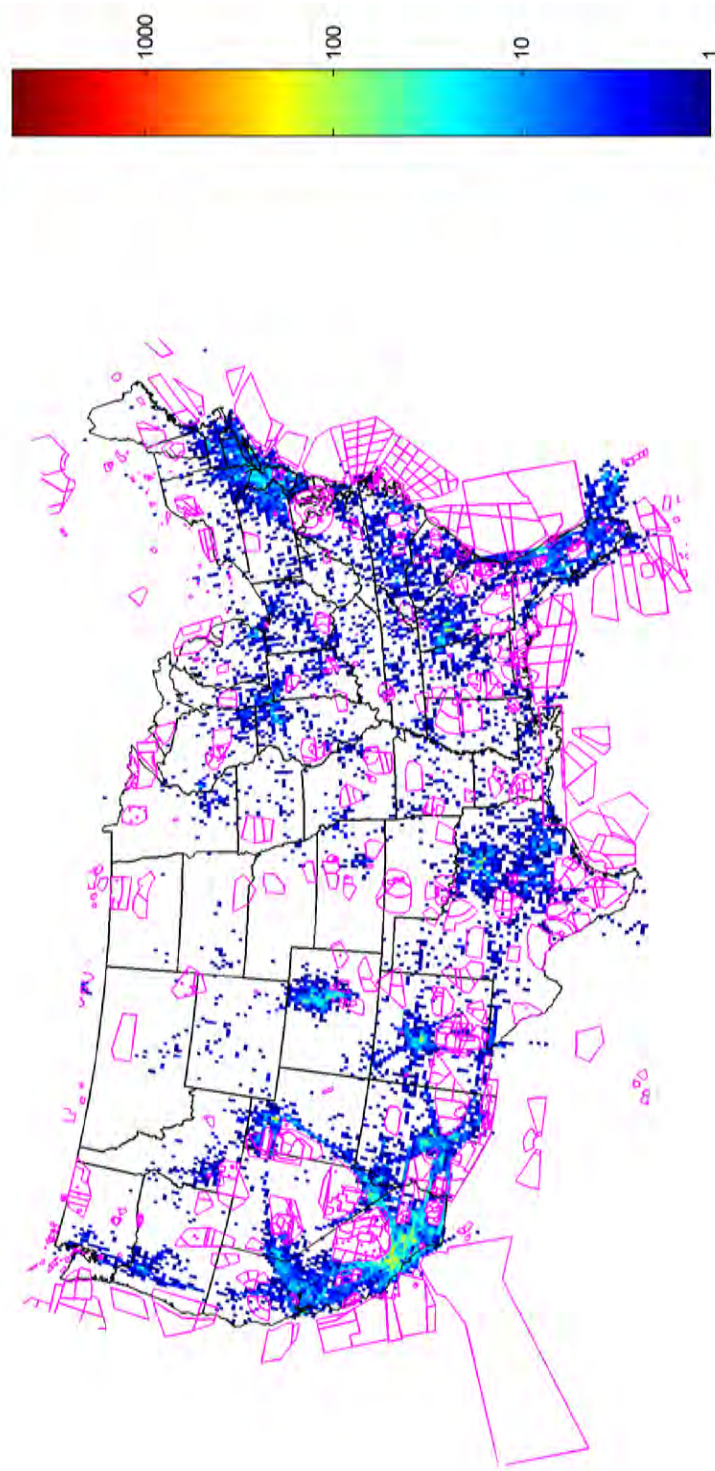


Figure H-4. Cumulative number of encounters per cell between 10,000 ft MSL and FL180, 1 December 2007 to 31 August 2008.

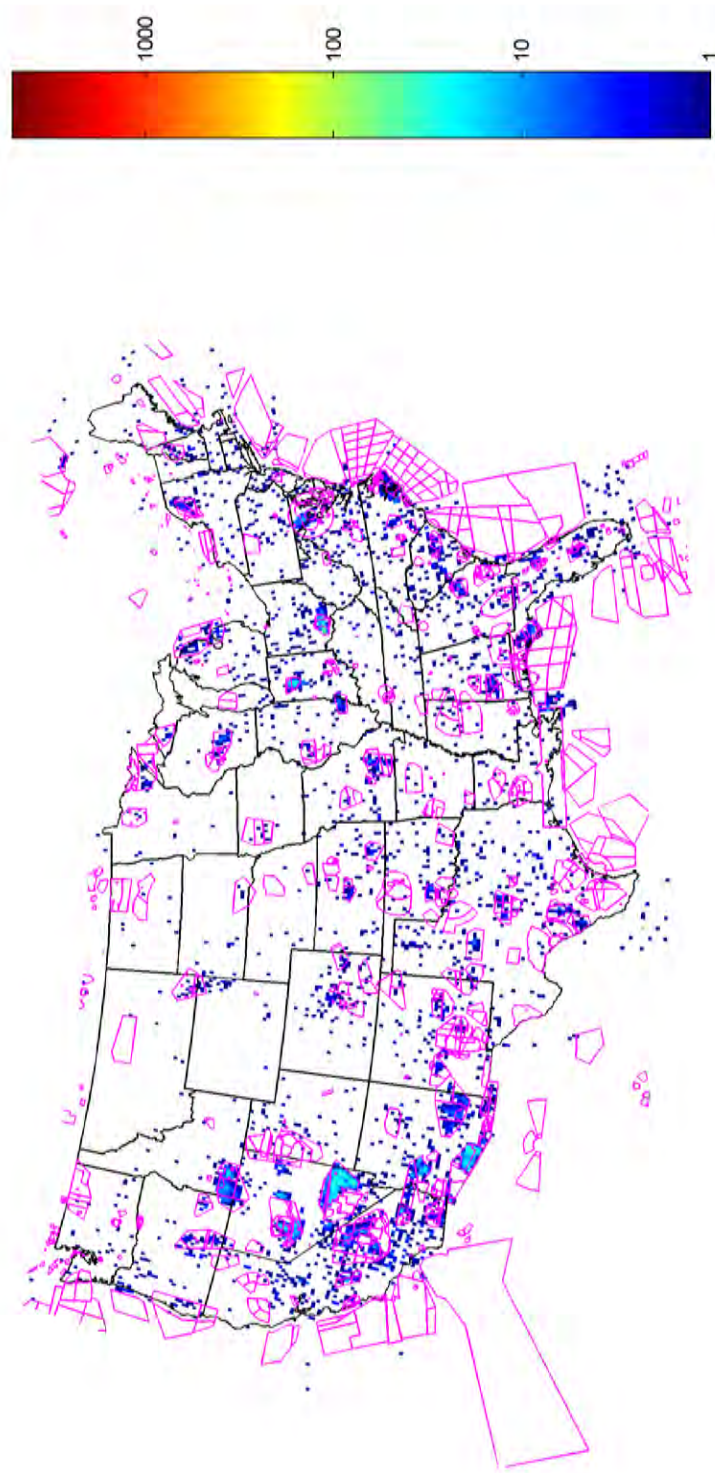


Figure H-5. Cumulative number of encounters per cell between FL180 and FL290, 1 December 2007 to 31 August 2008.

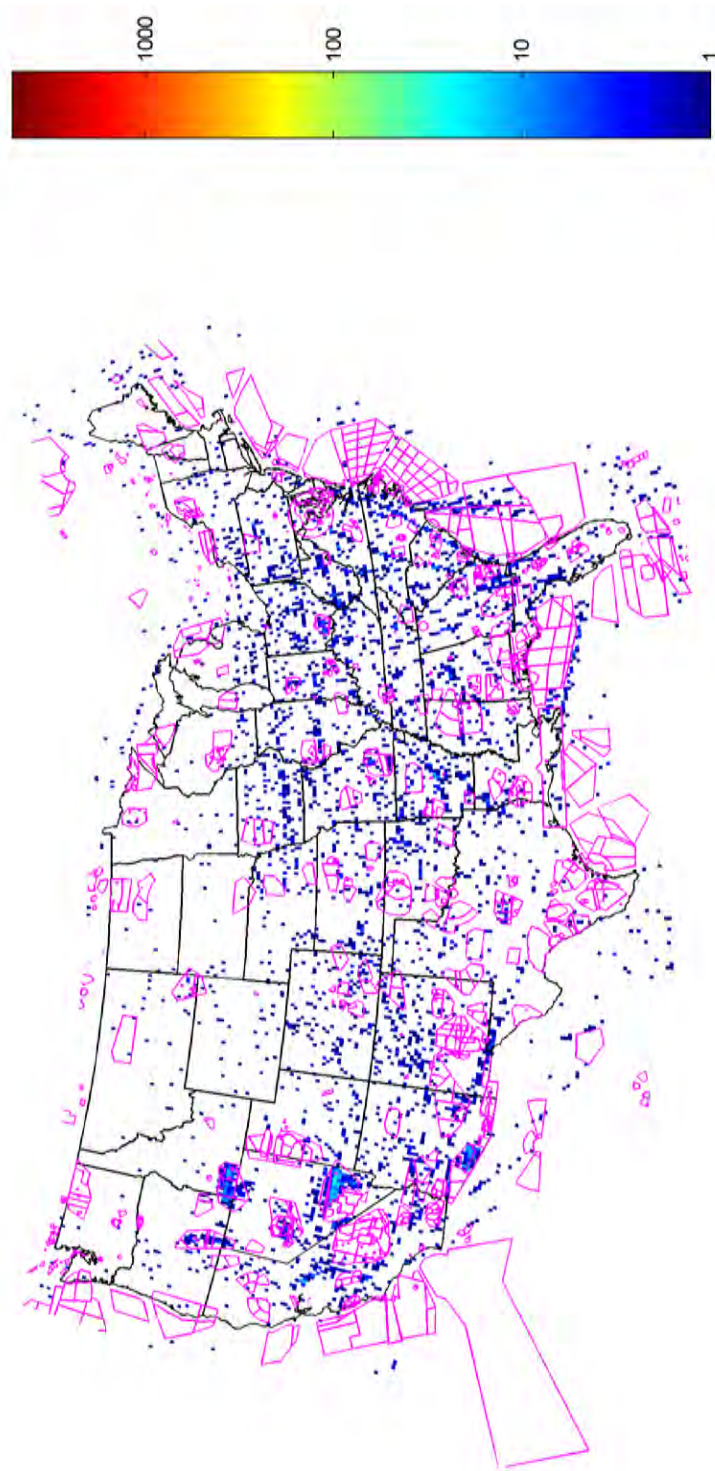


Figure H-6. Cumulative number of encounters per cell above FL290, 1 December 2007 to 31 August 2008.

APPENDIX I FEATURE HISTOGRAMS

Figure I-1 and Figure I-2 present histograms of the features not shown in Section 3. These features are only drawn from encounter situations; they do not represent the vast majority of normal separations observed between aircraft. The three histograms plotted in Figure I-1 show where encounters occur in the NAS in terms of airspace and altitude layer, as well as the types of aircraft involved. Note that the vast majority of encounters occur in airspace other than classes B, C, or D, and occur below 10,000 ft MSL.

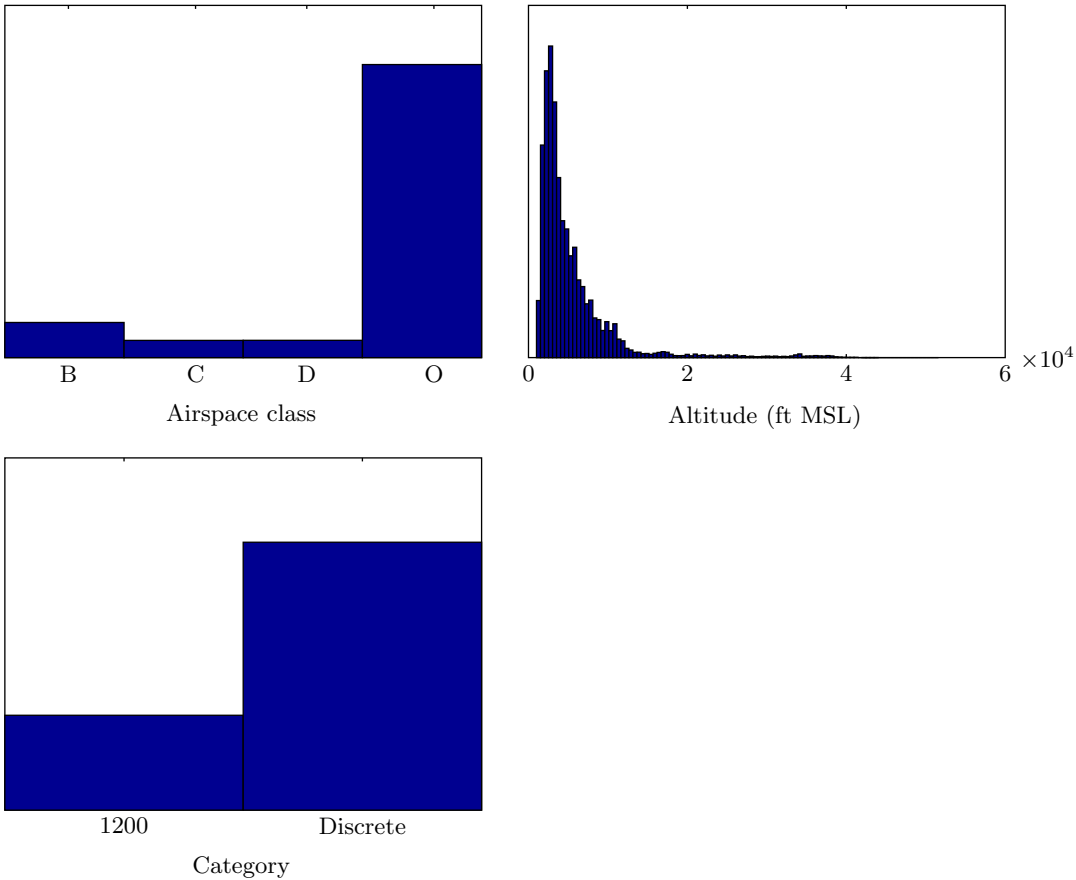


Figure I-1. Feature histograms of 411,867 observed encounters.

The histograms in Figure I-2 describe the features of encounters we consider to be functions of the aircraft dynamics, altitude layer, airspace class, and aircraft types. The horizontal miss distance has a peak around 0.5 NM, but a number of encounters have *hmd* between 100 and 200 ft. However, an analysis showed that most of these encounters had a vertical miss distance greater than 500 ft, which, while close, does not constitute an NMAC, and reinforces the necessity of jointly modeling the features rather than randomly selecting from independent distributions over these features.

Vertical miss distances were clustered around 500 ft, the minimum allowable vertical separation for visual separation. Cases where vertical separation is above 850 ft is rare: mainly a function of our encounter filter definition. If aircraft have a vertical separation at TCA above 850 ft, the two aircraft likely had a high vertical closing rate.

Approach angle smoothly drops in both directions from a peak at 180 deg. Approach angle is largely a function of airspeed, and as airspeed increases, the likelihood of a head-on encounter increases monotonically. The approach angle distribution shows a slight peak as it wraps around, indicating a slight tendency towards encounters between aircraft heading in the same direction. This effect would be far more drastic if parallel approaches and formation flight operations were not removed by our filtering process. It may be the case that these low approach angle encounters occur due to airspace structure and ATC operations. Finally, bearing appears to be relatively uniform, with perhaps slight peaks around 80 deg and 220 deg. However, as stated previously, in order to avoid over-specification of our encounters we discretized bearing into simply whether the intruder was “in front” or “behind” the own aircraft (see Section 5).

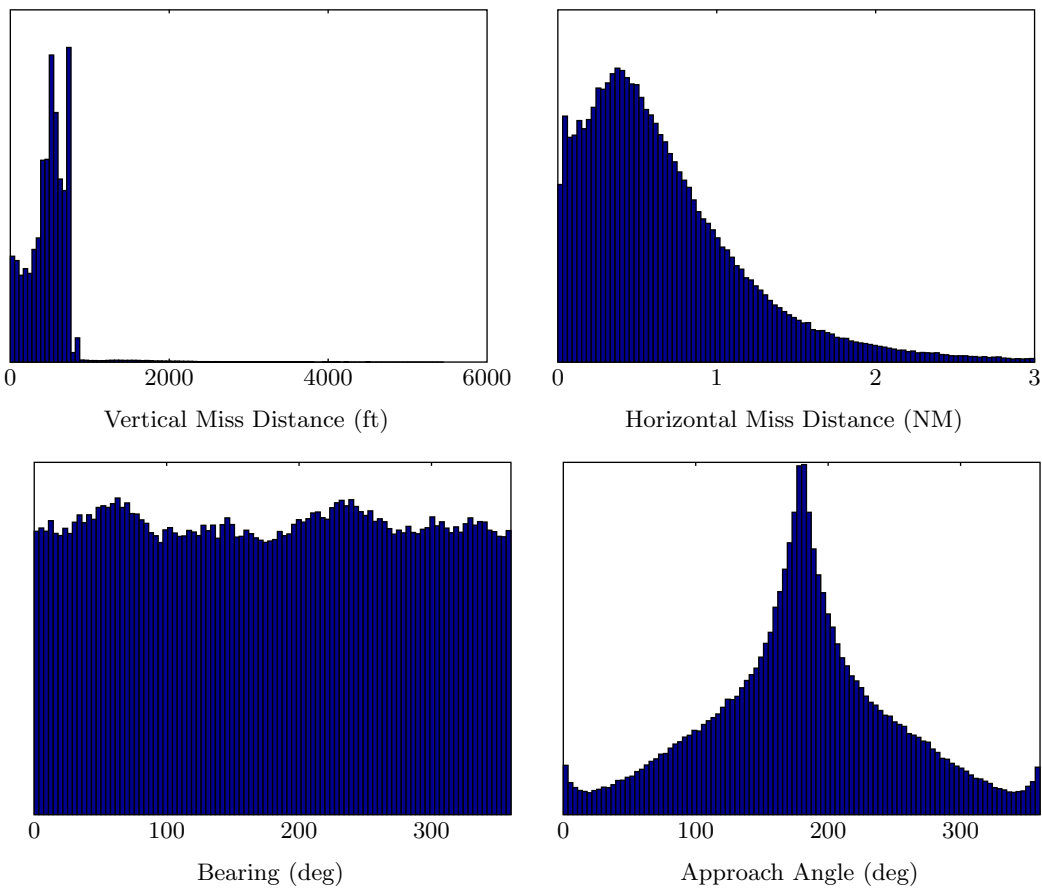


Figure I-2. Feature histograms of 427,367 observed encounters.

APPENDIX J BAYESIAN NETWORKS

This appendix briefly reviews Bayesian networks. Further discussion of Bayesian networks may be found elsewhere [32–34].

J.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 $P(x_i \mid \boldsymbol{\pi}_i)$, where $\boldsymbol{\pi}_i$ denotes an instantiation of the parents of X_i in the graph. The probability of an instantiation of the variables is specified directly by the conditional probability distributions in the Bayesian network:

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

J.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 [35].

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.

J.3 PARAMETER LEARNING

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

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

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

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

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

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

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

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

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

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

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

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 [34], 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 [27] use $\alpha_{ijk} = 1$. Heckerman, Geiger, and Chickering [36] 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

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

J.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,

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

The previous section explains how to compute the likelihood $P(D | \boldsymbol{\theta}, G)$ and the prior $p(\boldsymbol{\theta} | G)$. Cooper and Herskovits [27] show how to evaluate the integral above, resulting in

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

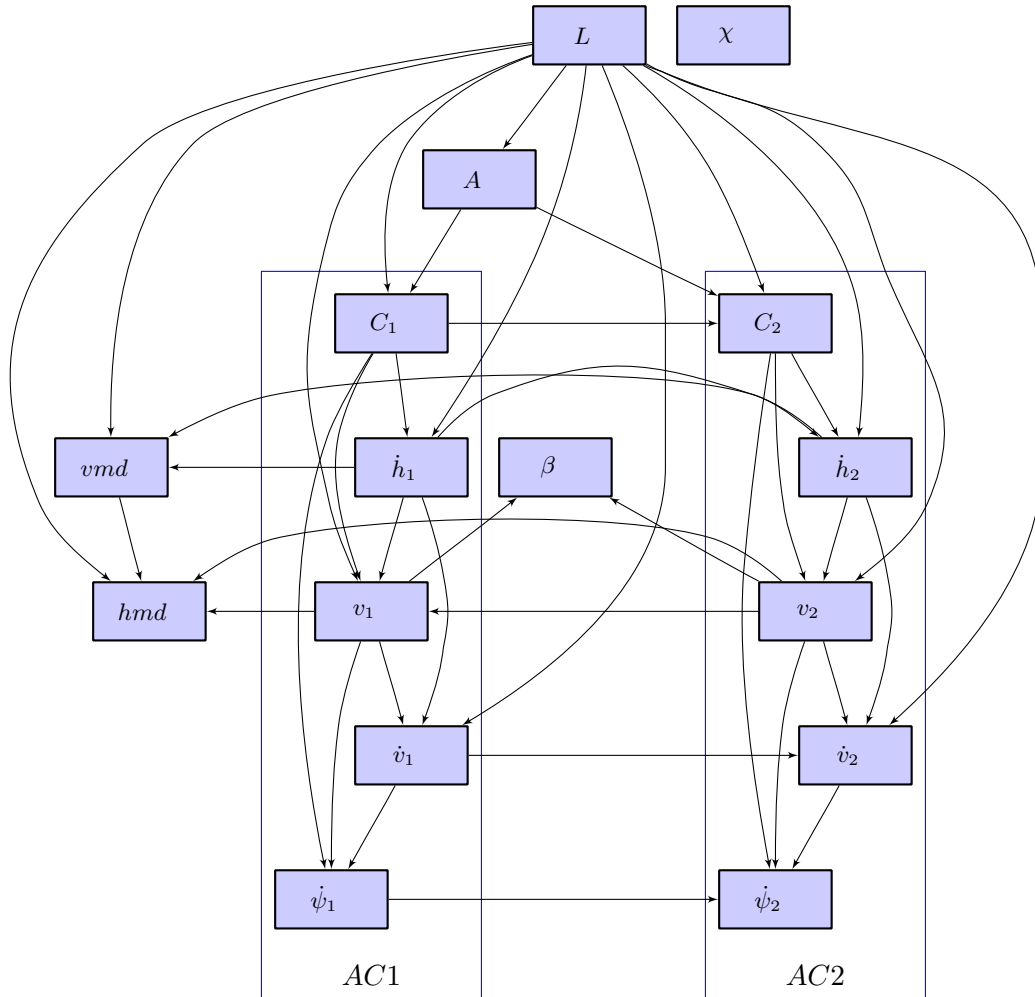
where $N_{ij} = \sum_{k=1}^r N_{ijk}$. Heckerman, Geiger, and Chickering [36] suggest priors over graphs, but it is not uncommon in the literature to assume a uniform prior. For numerical convenience, most Bayesian network learning packages calculate and report $\log P(G | D) + K$, where K is a constant independent of G . This quantity is often called the *Bayesian score* and may be used for structure comparison and search.

This page intentionally left blank.

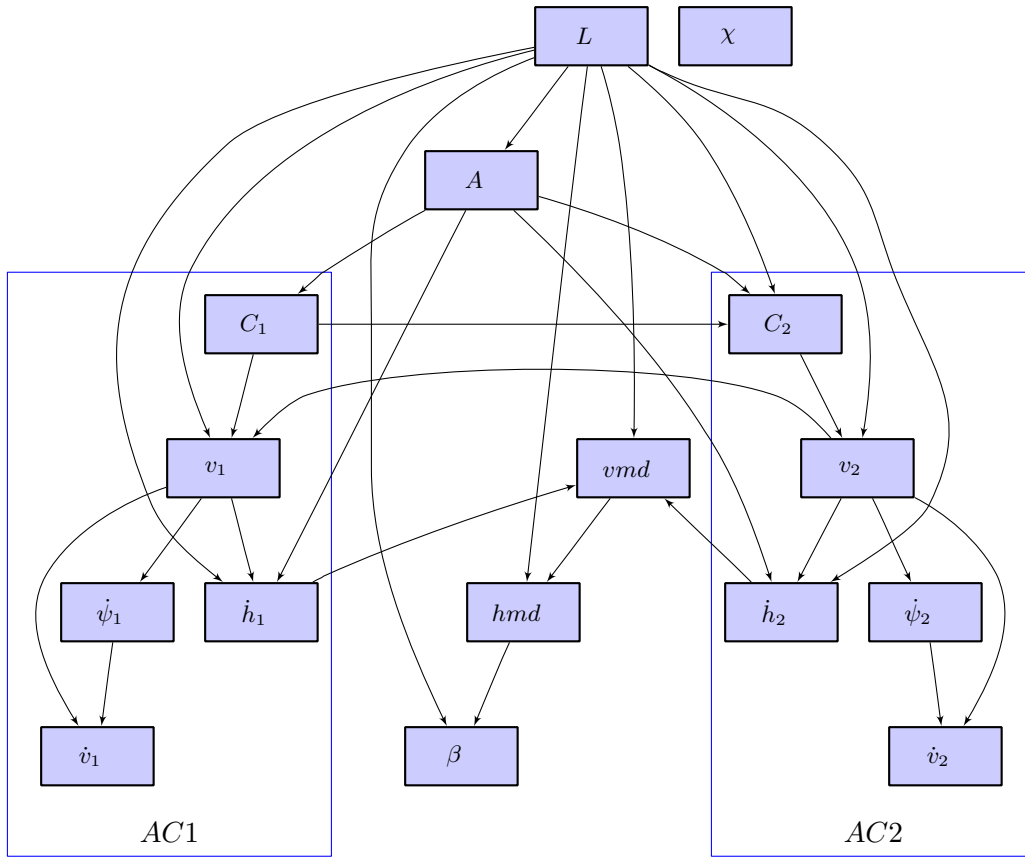
APPENDIX K NETWORK CANDIDATES

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 with the amount of data available. Candidate networks were chosen through a combination of prior expert knowledge, previous encounter model structures, and automated search techniques with a general goal of exploring the parameter space from very connected to very sparse network structures.

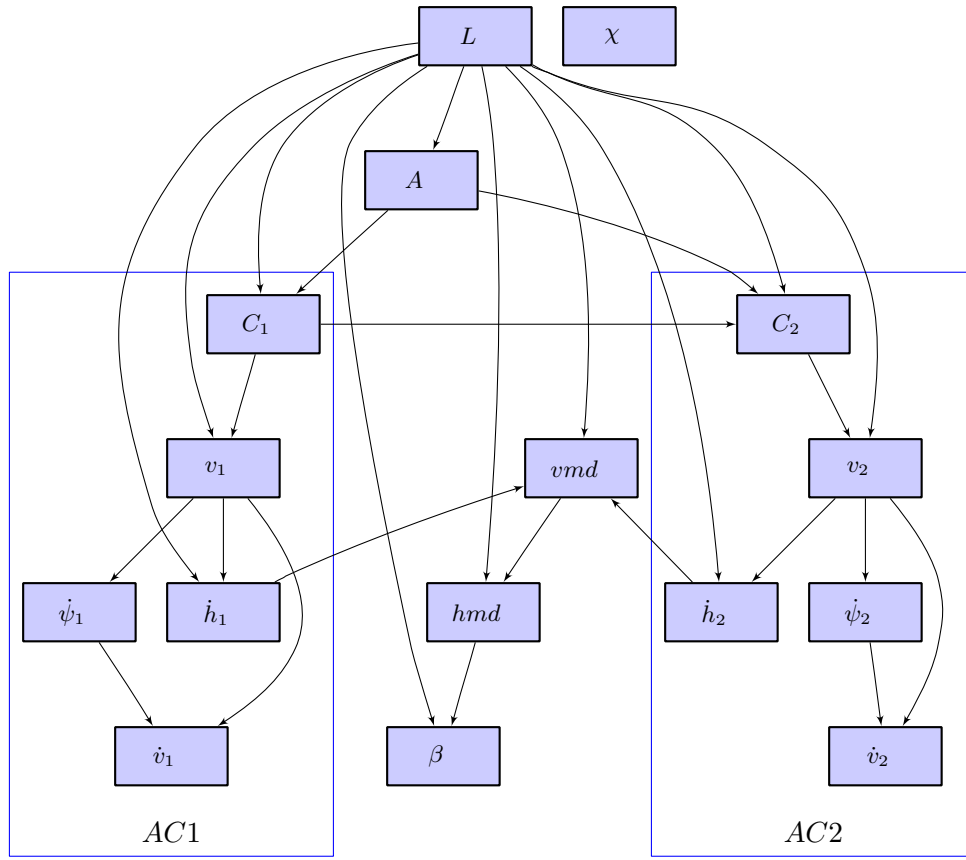
K.1 INITIAL NETWORK CANDIDATES



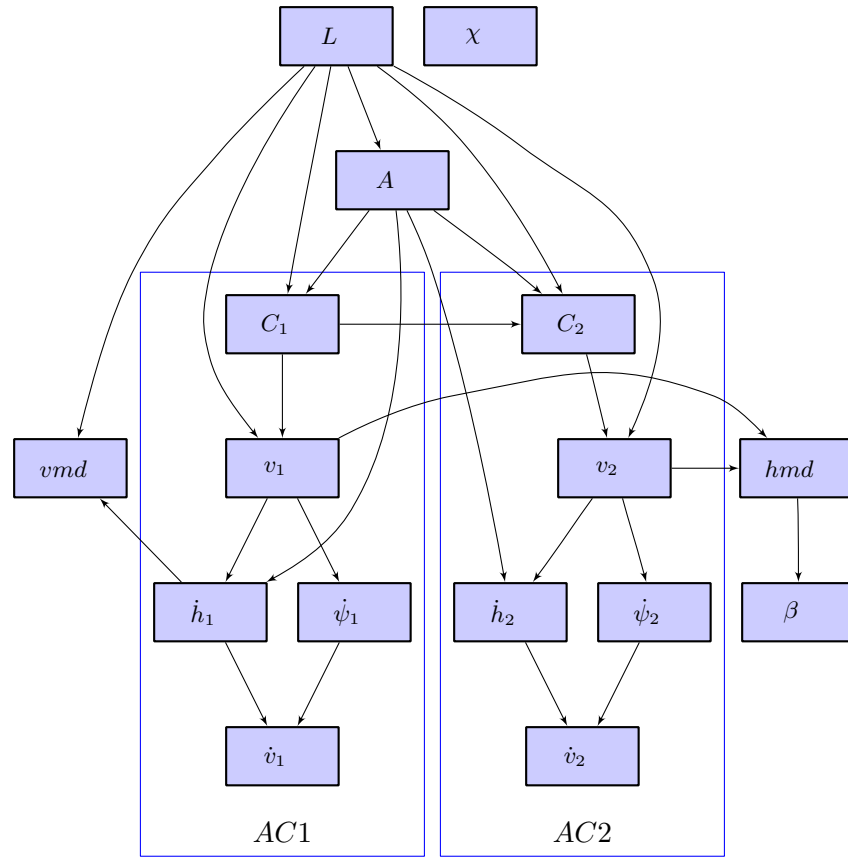
Score = -6760017 (best), edges = 41, parameters = 24751



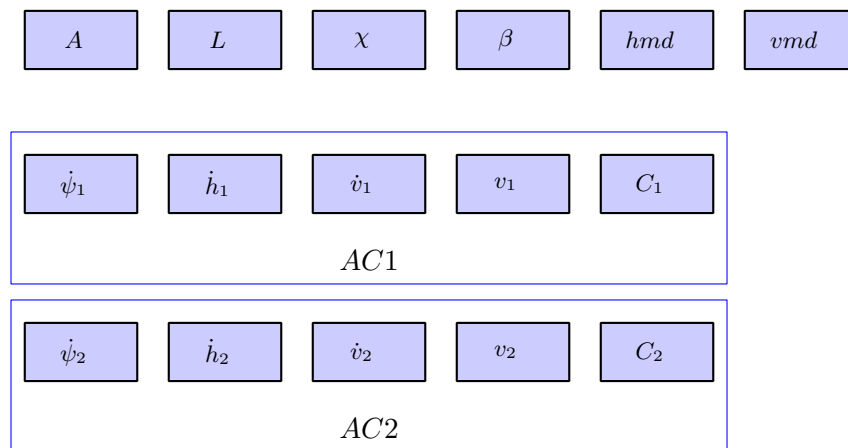
Score = -6776640, edges = 29, parameters = 6877



Score = -6779843, edges = 27, parameters = 5203

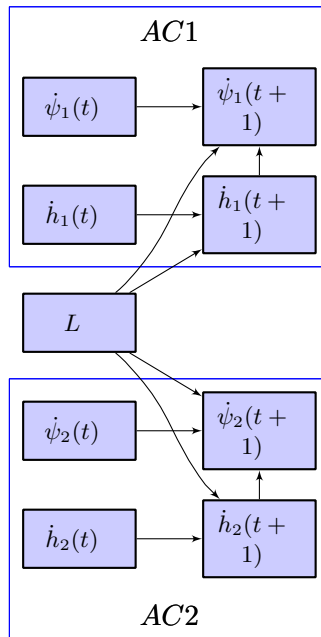


Score = -6817114, edges = 25, parameters = 1865

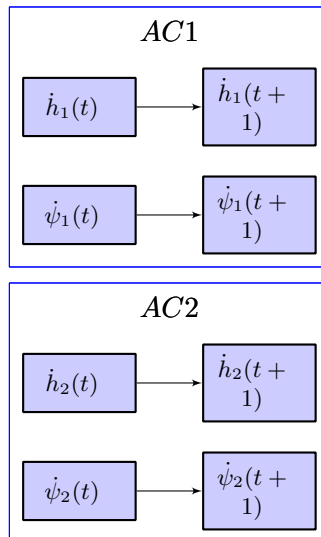


Score = -7259201, edges = 0, parameters = 83

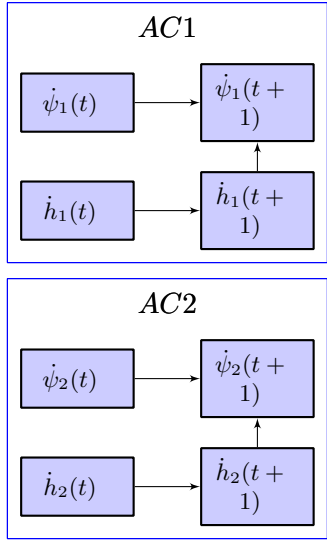
K.2 TRANSITION NETWORK CANDIDATES



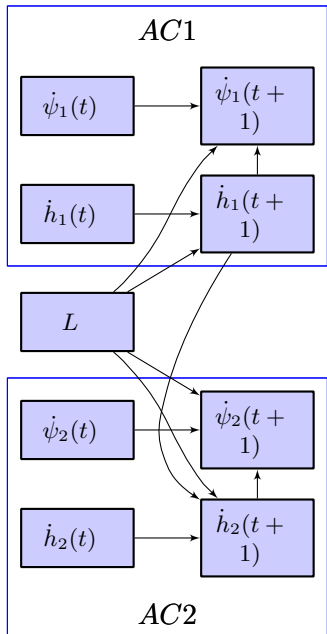
Score = -9822334 (best), edges = 10, parameters = 7200



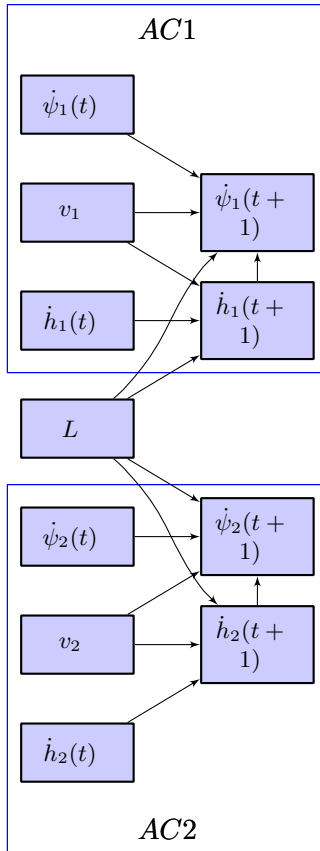
Score = -9826752, edges = 4, parameters = 288



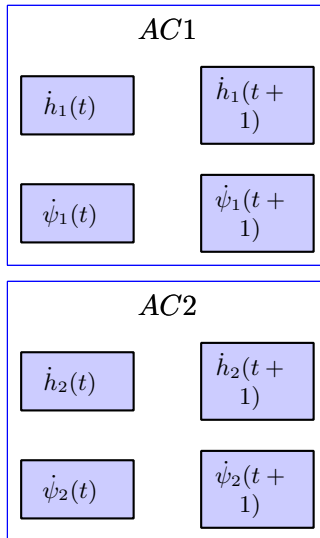
Score = -9828653, edges = 6, parameters = 1440



Score = -9834130, edges = 11, parameters = 10080



Score = -9856175, edges = 14, parameters = 43200



Score = -90986115, edges = 0, parameters = 32

This page intentionally left blank.

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* (1998), vol. IV, annex 10.
- [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] A.C. Drumm, B.J. Chludzinski, and G.S. Harris, “Monitoring of TCAS Resolution Advisories,” Presentation to RTCA SC-147, Requirements Working Group 2006.
- [12] T. Dean and K. Kanazawa, “A model for reasoning about persistence and causation,” *Computational Intelligence* 5(3), 142–150 (1989).
- [13] K. Murphy, *Dynamic Bayesian Networks: Representation, Inference and Learning*, Ph.D. thesis, University of California, Berkeley (2002).
- [14] R.D. Grappel, “ASR-9 Processor Augmentation Card (9-PAC) phase II scan-scan correlator algorithms,” MIT Lincoln Laboratory, Project Report ATC-298 (2001).
- [15] J.L. Gertz, “Mode S surveillance netting,” MIT Lincoln Laboratory, Project Report ATC-120 (1983).

- [16] F.N. Fritsch and R.E. Carlson, “Monotone piecewise cubic interpolation,” *SIAM Journal of Numerical Analysis* 17(2), 238–246 (1980).
- [17] E.M. Shank, “A coordinate conversion algorithm for multisensor data processing,” MIT Lincoln Laboratory, Project Report ATC-139 (1986).
- [18] W.M. Bolstad, *Introduction to Bayesian Statistics*, Wiley, 2nd ed. (2007).
- [19] M. Henrion, “Propagating uncertainty in Bayesian networks by probabilistic logic sampling,” in J.F. Lemmer and L.N. Kanal (eds.), *Proceedings of the Second Annual Conference on Uncertainty in Artificial Intelligence* (1986), p. 1986.
- [20] H.L. Chin and G.F. Cooper, “Belief network inference using simulation,” in *Uncertainty in Artificial Intelligence* (1989), pp. 129–147.
- [21] W. Buntine, “Operations for learning with graphical model,” *Journal of Artificial Intelligence Research* 2, 159–225 (1994).
- [22] R.D. Shachter and M.A. Peot, “Approaches to general probabilistic inference on belief networks,” in *Uncertainty in Artificial Intelligence* (1990), pp. 221–231.
- [23] RTCA, “Safety analysis of proposed change to TCAS RA reversal logic,” RTCA, Technical Rep. RTCA/DO-298 (2005).
- [24] J.K. Kuchar, “Methodology for alerting-system performance evaluation,” *Journal of Guidance, Control, and Dynamics* 19(2), 438–444 (1996).
- [25] L.F. Winder and J.K. Kuchar, “Evaluation of collision avoidance maneuvers for parallel approach,” *Journal of Guidance, Control, and Dynamics* 22(6), 801–807 (1999).
- [26] R. Srinivasan, *Importance Sampling: Applications in Communications and Detection*, Springer (2002).
- [27] G.F. Cooper and E. Herskovits, “A Bayesian method for the induction of probabilistic networks from data,” *Machine Learning* 9(4), 309–347 (1992).
- [28] RTCA, “Minimum operational performance standards for Traffic Alert and Collision Avoidance System (TCAS) airborne equipment,” RTCA (1990).
- [29] T.M. Doyle and F.G. McGee, “Air traffic and operational data on selected U.S. airports with parallel runways,” NASA, Langley Research Center, Technical Rep. CR-1998-207675 (1998).
- [30] “Federal Aviation Regulations,” U.S. Department of Transportation, URL http://www.faa.gov/regulations_policies/.
- [31] 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).

- [32] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, San Francisco, CA: Morgan Kaufmann (1988).
- [33] F.V. Jensen, *Bayesian Networks and Decision Graphs*, Springer Verlag (2001).
- [34] R.E. Neapolitan, *Learning Bayesian Networks*, Upper Saddle River, NJ: Prentice Hall (2004).
- [35] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein, *Introduction to Algorithms*, MIT Press, 2nd ed. (2001).
- [36] 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).

