A Synthetic Vision Preliminary Integrated Safety Analysis

Robert Hemm and Scott Houser
Logistics Management Institute, McLean, Virginia

The NASA STI Program Office . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) Program Office plays a key part in helping NASA maintain this important role.

The NASA STI Program Office is operated by Langley Research Center, the Lead Center for NASA's scientific and technical information. The NASA STI Program Office provides access to the NASA STI Database, the largest collection of aeronautical and space science STI in the world. The Program Office is also NASA's institutional mechanism for disseminating the results of its research and development activities. These results are published by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA's counterpart of peer-reviewed formal professional papers but has less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or cosponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services that complement the STI Program Office's diverse offerings include creating custom thesauri, building customized data bases, organizing and publishing research results . . . even providing videos.

For more information about the NASA STI Program Office, see the following:

- Access the NASA STI Program Home Page at *http://www.sti.nasa.gov*

- E-mail your question via the Internet to help@sti.nasa.gov

- Fax your question to the NASA Access Help Desk at 301–621–0134

- Telephone the NASA Access Help Desk at 301–621–0390

- Write to:
  NASA Access Help Desk
  NASA Center for AeroSpace Information
  7121 Standard Drive
  Hanover, MD 21076
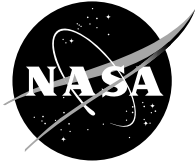
A Synthetic Vision Preliminary Integrated Safety Analysis

Robert Hemm and Scott Houser
Logistics Management Institute, McLean, Virginia

National Aeronautics and
Space Administration

Glenn Research Center

Available from

Contents

FIGURES

TABLES

Chapter 1
Introduction and Summary

This report documents efforts to analyze a sample of aviation safety programs, using the LMI-developed integrated safety analysis tool to determine the change in system risk resulting from Aviation Safety Program (AvSP) technology implementation. Specifically, we have worked to modify existing system safety tools to address the safety impact of synthetic vision (SV) technology. Safety metrics include reliability, availability, and resultant hazard. This analysis of SV technology is intended to be part of a larger effort to develop a model that is capable of "providing further support to the product design and development team as additional information becomes available." The reliability analysis portion of the effort is complete and is fully documented in this report. The simulation analysis is still underway; it will be documented in a subsequent report.

The specific goal of this effort is to apply the integrated safety analysis to SV technology. This report also contains a brief discussion of data necessary to expand the human performance capability of the model, as well as a discussion of human behavior and its implications for system risk assessment in this modeling environment.

PROBLEM BACKGROUND

NASA's Aviation Safety Program is pursuing technologies to identify and improve hazardous situations within the national airspace system. Reducing safety hazards is a key requirement for continued growth in air traffic density that is necessary to meet continually increasing demand. Integrated safety analysis of day-to-day operations and the risks within those operations will provide an understanding of the AvSP portfolio that is not now available. If technology buyers understand the potential for risk reduction provided by technology implementation, they can develop business cases to evaluate tradeoffs among cost, performance, and safety. Ultimately, integrated system analyses that address safety, performance, and economics can provide useful information for choosing and supporting development programs. Figure 1-1 depicts the conceptual structure for such analyses and indicates the role of integrated safety analyses.

Figure 1–1. Integrated System Analysis
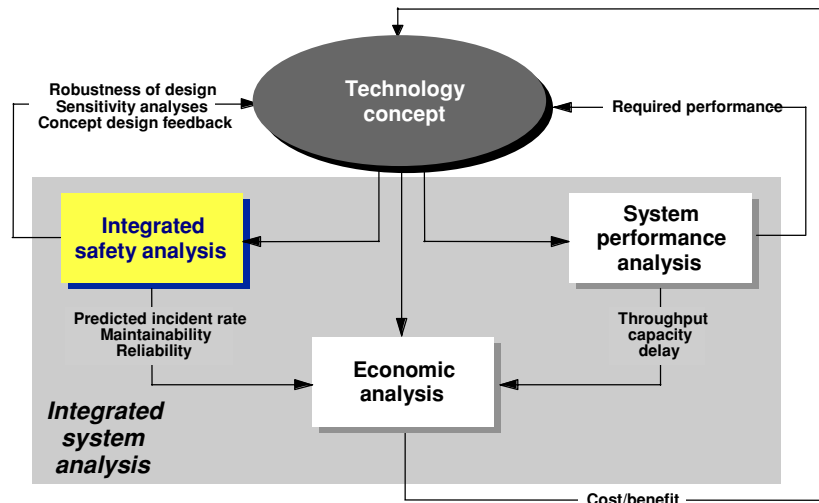
As Figure 1-1 shows, an integrated safety analysis is intended to provide useful data that can be applied to other analyses and to system design. An integrated safety analysis must be able to estimate the safety benefits from competing technologies for existing operations and investigate the safety implications of using new technologies to enable higher-capacity operations. Figure 1-2 depicts the methodology of an integrated safety analysis.

Figure 1–2. Integrated Safety Analysis
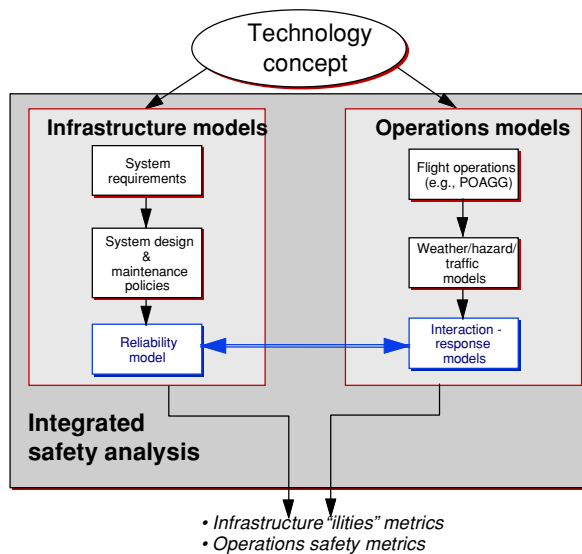
The reliability and interaction-response models are the two principal analytic components of the integrated safety analysis. These models are used to develop safety statistics necessary to determine whether the system meets acceptable safety levels. They also are necessary to conduct tradeoffs between safety levels and cost.

The basic algorithm for developing safety statistics is as follows:

$$P_{\text{Accident}} = P_{\text{Failure}} * (P_{\text{Accident}} \mid \text{Failure})$$

P_{Failure} is the probability of a hardware or software failure, as estimated by the reliability model. $P_{\text{Accident}} \mid \text{Failure}$ is the conditional probability of an accident given a failure, as estimated by the interaction-response model. We use Markov reliability models to estimate P_{Failure} and simulation to estimate $P_{\text{Accident}} \mid \text{Failure}$. [1]

A combination of reliability modeling plus simulation provides significant computational benefit over either method alone. Reliability models have difficulty addressing dynamic parameters such as pilot response and changing environment, whereas simulation of low-probability failures leads to an impractical number of computer runs. Using reliability models to estimate equipment failure rates and simulations to estimate the results when those failures occur adds dynamics to the reliability analysis and makes the simulation computationally tractable.

SYNTHETIC VISION—DESCRIPTION

Synthetic vision presents computer-generated views of the external environment to the pilot. The SV presentation is completely artificial. Typically it is based on a geographical and cultural database, supplemented by dynamic traffic information. For terrain, current experimental implementations of SV use global positioning satellite (GPS) position data to dynamically link the database to the aircraft's position and attitude. Supplemental sensors may be used to confirm the GPS position data and/or provide additional data (e.g., data about other aircraft, weather, or ground equipment). Airborne and ground traffic information may come from several sources, such as automatic dependent surveillance-broadcast (ADS-B) or cockpit display of traffic information (CDTI). SV systems can use head-up and head-down displays; the current concept focuses on a head-down display. Displays can include an artificial out-of-the-window view (in all directions) or a variety of symbolic and map presentations.

The primary purpose of SV is to improve safety by providing visual flight rule (VFR)-like situation awareness in instrument flight rule (IFR) conditions. The improved situation awareness provided by SV also can be used to improve operational performance. The performance goal of SV is to allow VFR-like operations in all IFR conditions down to Category IIIb. Such performance includes operations into runways that cannot use instrument landing systems because of terrain interference, independent approaches to closely spaced parallel runways, reduced in-trail separations between arriving aircraft, and low-visibility ground operations.

For analysis, SV performance can by defined by actual navigation performance (ANP) and traffic information. ANP is defined as the aircraft's (and aircrew's)

---

[1] Terms such as *weighted system safety statistic* and *overall hazard level* also have been used to refer to what we call P_{Accident}.

knowledge of its own position (plus speed and heading) with respect to the earth. Traffic information is defined as the aircraft's (and aircrew's) knowledge of the position, speed, and heading of other aircraft.

Alaska Airlines pioneered the use of navigational performance criteria for instrument flying, including development and approval of required navigational performance (RNP) IFR Category I approaches through mountain valleys to runways that have no instrument landing system (ILS) equipment. For those approaches, IFR landing is allowed when the actual navigational performance (ANP) meets or exceeds the RNP for the approach. Current RNPs for Category I approaches range from 0.15 to 0.30 nautical miles; lower RNPs allow lower decision heights, visibility, and ceilings. The Alaska Airlines RNP implementation establishes a series of waypoints that can be precisely flown on the basis of GPS and inertial reference data, coupled with the flight management system and autopilot. The Enhanced Ground Proximity Warning System (EGPWS), which provides a terrain display that is based on GPS data, provides supplemental information. Current ANP levels are achieved with nonaugmented GPS. Category IIIb operations, envisioned with SV, will require ANPs on the order of feet and consequently will require improved GPS accuracy provided by a local area augmentation system (LAAS).

SYSTEM DESCRIPTION FOR RELIABILITY MODELING

To estimate reliabilities we must specify SV hardware and software equipment, the number of redundant units, failure rates, recovery rates, and error detection coverage rates. The equipment includes items specifically identified with SV as well as items that are necessary for SV operations. Figures 1-3 and 1-4 are fault tree depictions of the equipment items and how they relate to support terrain avoidance and traffic avoidance respectively. Component definitions and reliability results are discussed in more detail in Chapter 2.

Figure 1–3. Fault-Tree Display of Terrain Avoidance Equipment
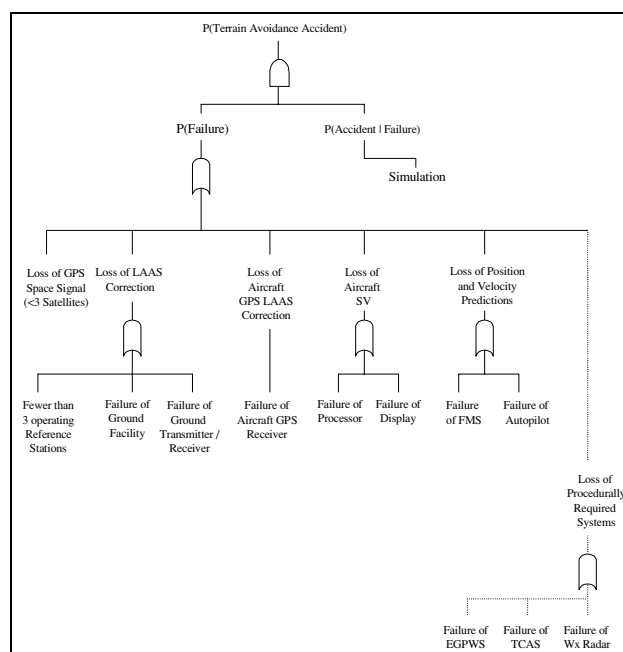
Figure 1–4. Fault-Tree Display of Traffic and Terrain Avoidance Equipment
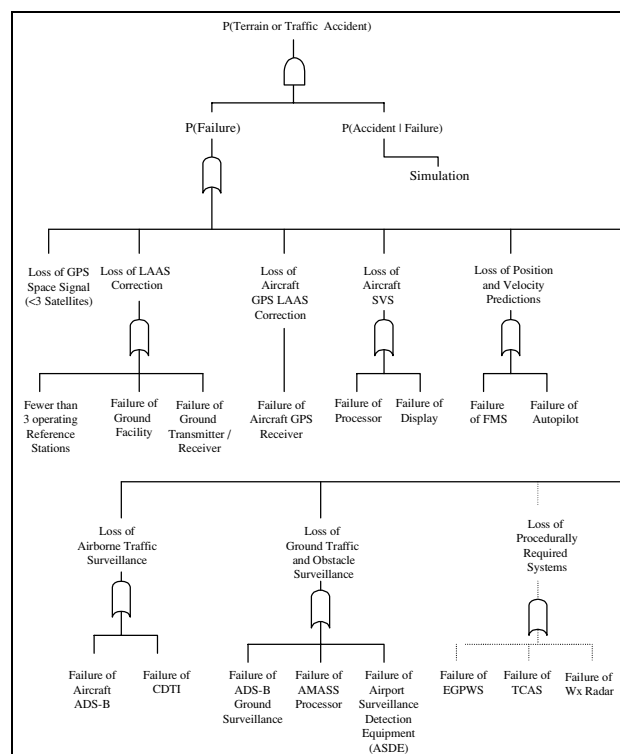
For simulation modeling, we need to define SVS performance parameter values for fully operational and degraded conditions. For all scenarios, we are concerned with the "own-aircraft" knowledge of its position, flight track, speed, and data

latency. For scenarios that involve other traffic, we must add the own-aircraft knowledge of traffic aircraft position, heading, and velocity, with data latency. Terrain avoidance scenarios in this study are assumed to be at airports without radar surveillance, so the controller has no tactical traffic data. Traffic avoidance scenarios are assumed to be at airports that have air and surface radar surveillance, plus ADS-B traffic information; therefore, for such scenarios we specify the controller's airborne and ground traffic data. Parameters considered in the simulation are as follows:

- Terrain avoidance scenario:

  - Own-aircraft knowledge of own-aircraft position, speed, track, data latency

- Traffic avoidance scenario:

  - Own-aircraft knowledge of own-aircraft position, speed, track, data latency

  - Own-aircraft knowledge of traffic position, speed, track, data latency

  - Controller knowledge of own-aircraft position, speed, track, data latency

  - Controller knowledge of traffic position, speed, track, data latency

- Additional simulation parameters:

  - Pilot and controller detection times

  - Pilot and controller communication times

  - Pilot and controller response times

  - Traffic density

  - Blunder geometry.

Most parameters are defined by a fixed bias and an uncertainty; others, such as ceiling, are defined by a single number, and some—such as blunder geometry— by extended definitions. Details of simulation parameters are discussed in Chapter 3.

Chapter 4 summarizes a subcontracted review of human factors modeling. Reliability data are discussed and documented in Appendix A. A brief discussion of Markov analysis and the input files for the reliability analysis are included in Appendix B. Finally, Appendix C contains miscellaneous references and data that are applicable to the simulations.

Chapter 2
Reliability Analysis

In this chapter, we document the assumptions, input parameters, and results of the Markov reliability analysis.

EQUIPMENT AND COMPONENTS

As discussed in Chapter 1, several systems in addition to the "synthetic vision" equipment are necessary for SV operation and must be included in the reliability analysis. Table 2-1 lists all equipment components and associated parameters. The components that make up equipment items are based on several sources, including a 1999 NASA report,[1] *Jane's Avionics*, equipment brochures, technical reports, and engineering judgment. Although the architectures generally are representative and are appropriate for the purposes of this study, they are not defined to the level necessary for a detailed system-specific safety analysis. The model and methods are capable of such an analysis, however, if detailed equipment architectures are provided.

Table 2-1. Reliability Analysis Components and Parameters

Element	Redundant components	Failure rate	Recovery rate	Coverage 2	Coverage 1	Mission time
GPS	up to 13					
Satellites (soft failure)		1.65 / year	12.2 / hour			10,000 days
Satellites (hard failure)		0.16 / year	1.25 / month			
LAAS range sensor (LRS)	6	per hour				10 hours
GPS antennas	1	1.0 e-5	0.2	N/A	N/A	
GPS receivers	1	1.0 e-4	0.5	N/A	N/A	
GPS processor	2	5.0 e-4	0.2	N/A	N/A	
Communications processor	2	5.0 e-4	0.2	N/A	N/A	
Communications modulator transmitter	1	1.0 e-4	0.5	N/A	N/A	
Communications antenna	1	1.0 e-5	0.2	N/A	N/A	
LAAS ground facility (LGF) processor	1	per day				1 day
Main processor (hardware)	3	1.0 e-3	4	N/A	N/A	
Main processor operating system (software)	3	1.0 e-2	144	N/A	N/A	
Correction software (5.5e-5 P(f) per 150 sec spec)	N/A	0.03168	576	N/A	2.304 e-5	
Integrity software (5.5e-5 P(f) per 150 sec spec)	N/A	0.03168	576	N/A	2.304 e-5	
LGF receiver/transmitter	1	per hour				10 hours
Modulator transmitter	2	1.0 e-4	0.25	N/A	N/A	
Receiver demodulator	2	1.0 e-4	0.25	N/A	N/A	
Antenna	2	1.0 e-5	0.1	N/A	N/A	
Airborne GPS receiver	1	per hour				10 hours
Antenna	2	1.0 e-5	N/A	1.0	1.0	

---

[1] P. Kostiuk et al., *A System for Integrated Reliability and Safety Analysis,* NASA CR-1999-209548, August 1999.

Table 2-1. Reliability Analysis Components and Parameters (Continued)

Receiver	3	5.0 e-5	N/A	0.99	.095	
Processor	2	5.0 e-5	N/A	0.99	.095	
SV processor and display	1	per hour				10 hours
SV processor	2	5.0e-5	N/A	0.99	0.95	
SV display	2	1.0e-4	N/A	.0999	0.99	
CDTI	1	per hour				10 hours
Ground transmitter	2	1.0 e-4	0.25	N/A	N/A	
Airborne receiver	2	5.0 e-5	N/A	N/A	N/A	
ADS-B	1	per hour				10 hours
Inertial Navigation System (INS)	2	2.0e-4	N/A	0.999	0.99	
ADS-B processor	2	5.0e-5	N/A	0.99	0.95	
ADS-B display	2	1.0e-4	N/A	0.999	0.99	
Modulator transmitter	1	5.0e-5	N/A	N/A	.099	
Receiver demodulator	1	5.0e-5	N/A	N/A	0.99	
Antenna	1	1.0e-5	N/A	N/A	1.0	
Flight Management System (FMS)	1	per hour				10 hours
Dual INS (failure set to 0)	2	0.0	N/A	0.999	0.99	
Dual FMS processor	2	1.0e-5	N/A	0.99	0.95	
Dual navigation radios	2	5.0e-5	N/A	0.99	1.0	
Autopilot		per hour				10 hours
Triple roll computer	3	5.0e-5	N/A	0.99	0.95	
Triple pitch computer	3	5.0e-5	N/A	0.99	0.95	
Autostabilizer	1	1.0e-4	N/A	n/a	0.99	
Dual Landing Roll-out Control Unit (LRCU)	2	1.0e-3	N/A	0.99	0.95	
Triple sensor suite	3	1.0e-3	N/A	0.99	0.95	
Airport surface detection equipment (ASDE radar)		per hour	per hour			
Antenna	1	1.0e-3	0.25	N/A	N/A	1,000 hours
Transmitter	2	1.33e-3	0.5	N/A	N/A	
Receiver	2	1.33e-3	0.5	N/A	N/A	
Processor	2	5.0e-5	2.0	N/A	N/A	
ADS-B ground surveillance stations	5	per hour	per hour			1,000 hours
Antenna	1	1.0e-5	0.2	N/A	N/A	
Processor	1	5.0e-4	0.2	N/A	N/A	
Receiver	1	1.0e-4	0.5	N/A	N/A	
Transmitter	1	1.0e-4	0.5	N/A	N/A	
Airport Movement Area Safety System (AMASS) processor		per hour	per hour			10 hours
Main processor (hardware)	3	5.0e-5	0.25	N/A	N/A	
Main processor operating system (software)	3	1.0 e-3	2.0	N/A	N/A	
Common avionics		per hour				10 hours
Weather radar	1	1.0e-4	n/a	N/A	N/A	
TCAS (transponder-based) system	1	1.0e-3	n/a	N/A	N/A	
EGPWS (GPS-based) system	1	1.0e-4	n/a	N/A	N/A	

FAILURE AND RECOVERY RATES

The failure and recovery rates used in the analysis are from three sources. The 1999 NASA report is the basis for several of the components modeled in the analysis.[2] The rates were based partly on government specifications and partly on engineering judgment. To substantiate the engineering judgment, we investigated two other sources. The first was the Air Force "DO41" peacetime maintenance database for the two years (eight quarters) ending in March 2000. The second was *Jane's Avionics*. Data from these sources are discussed in Appendix A. Although the new data added insight, there is still considerable engineering judgment included in the rate selection.

---

[2] Kostiuk et al., *A System for Integrated Reliability and Safety Analysis.*

INDIVIDUAL EQUIPMENT DISCUSSIONS

GPS Satellites

The GPS system consists of 24 operational satellites, plus on-orbit spares. Three-dimensional navigation using GPS requires data from at least 4 satellites. The number of satellites that are within visual range at any one time varies from 4 to 13.[3] Determination of the probability of successful GPS data acquisition is performed in three steps. First, the probability that 4-of-*N* satellites will be operational is estimated for the 10 cases where *N* ranges from 4 to 13. Second, those probabilities are multiplied by the corresponding probabilities that 4 to 13 satellites will be in range. For example, the probability that 4-of-4 satellites will be working is multiplied by the probability that 4 satellites will be in range. Similarly, the probability that 4-of-5 satellites will be working is multiplied by the probability that 5 satellites will be in range, and so on through the probability of 4-of-13 working times 13 in range, for a total of 10 cases. Finally, the probabilities from the 10 cases are combined to determine the expected probability that at least 4 operational satellites will be acquired.

The probabilities that 4-of-*N* satellites will be working are estimated by using 10 different runs of a Markov model. The individual satellites are modeled as single units that can suffer hardware and software failures. Software failures are reparable, but hardware failures require satellite replacement. Failure and repair/replacement rates are listed in Table 2-1.

The probabilities that 4 to 13 satellites will be in range are taken from a 1997 NASA report.[4] The in-range probabilities, shown in Table 2-2, are functions of latitude and elevation angle. For this analysis we used the 10 degree elevation angle values.

Table 2-2. Probabilities of Having Satellites In Range

Number of satellites	4	5	6	7	8	9	10	11	12	13
Elevation 5° — Latitude 35°	0.0	0.008	0.08	0.22	0.32	.023	0.105	0.03	0.005	0.003
Elevation 10° — Latitude 35°	0.4	0.084	0.303	0.303	0.084	.04	0.02	0.002	0.0	0.0

LAAS Range Sensor

The Local Area Augmentation System (LAAS) uses ground reference sensors located in the vicinity of the airport to measure local errors in GPS position data. The sensors' basic functions are to determine their positions on the basis of GPS information and compare them to their known positions (based on surveys). The errors in the two positions are used to calculate corrections to GPS signals. At

---

[3] Ibid.

[4] B. Carpenter and J. Kuchar, *A Probability-Based Alerting Logic for Aircraft on Parallel Approach,* NASA CR 201685, April 1997.

least four reference stations must be operating to ensure the accuracy needed for IFR Category III operations.

The LAAS Range Sensor (LRS) is modeled as a GPS receiver system coupled to a communications transmitter system that transmits the sensor GPS position to the central LAAS ground facility. Figure 2-1 is a block diagram of the sensor. Multiple boxes in the block diagram indicate redundant equipment. For this study, we assume that all redundant equipment is cross-strapped and will automatically and instantaneously replace failed components.

Figure 2–1. Block Diagram of LAAS Range Sensor
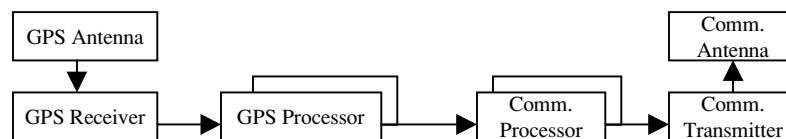
We assume that six range sensors are deployed and require that four be operating. We run a Markov estimate for a single sensor and determine the probability that four of six sensors will be operating on the basis of a binomial distribution.

LAAS Ground Facility

The LAAS Ground Facility (LGF) receives GPS position data from the range sensors, calculates local errors in the GPS signals, and transmits correction data to aircraft and ground vehicles. LGF components are shown in Figure 2-2. For ease of analysis, the reliabilities of the processor and communications equipment are calculated separately.

Figure 2–2. LAAS Ground Facility
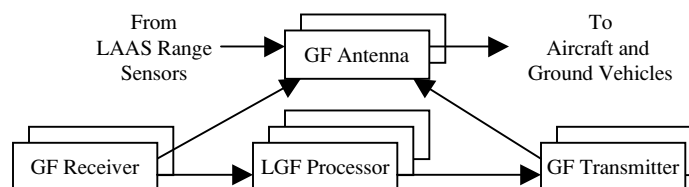
The processor model and associated failure and recovery rates are based on the analogous Wide Area Augmentation System (WAAS) processor.[5] The processor is triply redundant, and hardware and software failures are modeled. Software failures are identified separately as loss of the basic correction signal and loss of the integrity signal needed to certify data accuracy. Although these failures can be used to identify degraded operating states, in the current analysis they are combined.

---

[5] Kostiuk et al., *A System for Integrated Reliability and Safety Analysis.*

The LGF receiver/transmitter includes fully redundant antennas, receivers, and transmitters. Because it is located on the ground, its components are fully reparable. We assume that the LGF processor will be able to detect failures in ground facility signals and that there will be no undetected failures.

Airborne GPS Receiver

The components of the airborne GPS receiver are shown in Figure 2-3. The airborne components are not reparable during flight. The receiver and processor also may experience undetected failures. Coverage rates for undetected failures are based on values in the 1999 NASA report.[6]

Figure 2–3. Airborne GPS Receiver

From GPS and LAAS Ground Transmitter → GPS Antenna → GPS Receiver → GPS Processor → To Autopilot FMS and SVS
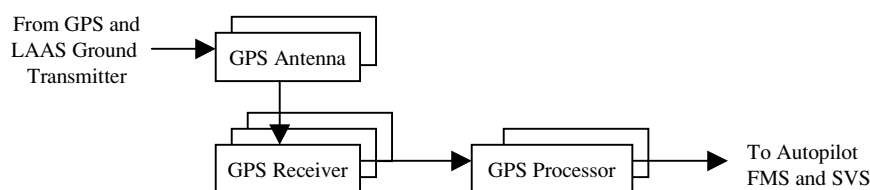
SV Processor and Display

The SV hardware includes redundant processors and displays, as shown in Figure 2-4. The equipment is not reparable during flight. The processor and displays may experience undetected failures. For safety analysis purposes, we assume that these undetected failures could result in production and display of geographically biased images.

Figure 2–4. Synthetic Vision Processor and Display

From Database, LAAS, GPS, etc. → SV Processor → SV Display → To Aircrew
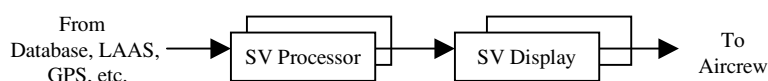
CDTI Datalink

The CDTI datalink includes redundant ground transmitters and redundant airborne receivers, as shown in Figure 2-5. The ground transmitters are assumed to be reparable. Datalink failures are assumed to be fully detectable.

Figure 2–5. CDTI Datalink

From Controller, ADS-B, Radar → Ground Transmitter → Airborne Receiver → To SVS Processor and Display

---
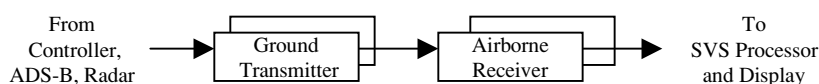
[6] Ibid.

CDTI and ADS-B (see below) provide traffic information to the aircrew. We assume that all information on ground traffic comes through CDTI; thus, if CDTI fails, the aircraft loses track of ground traffic.

Automatic Dependent Surveillance-Broadcast

The ADS-B components are based on the model in the 1999 NASA report. ADS-B components are shown in Figure 2-6.

Figure 2–6. ADS-B
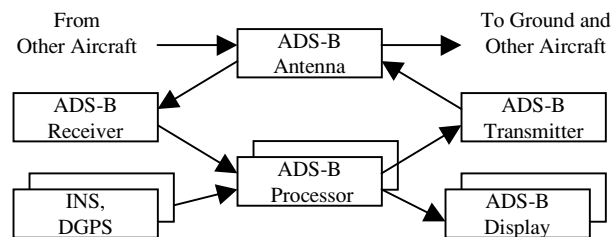
None of the ADS-B equipment is reparable during flight, and all of it can experience undetected failures. We retain the ADS-B display in the model, however. ADS-B data may be displayed on the SV display.

Flight Management System

The FMS components shown in Figure 2-7 were based on the description of the Smith's Industries FMS in the *1995 Jane's Avionics*.

Figure 2–7. FMS
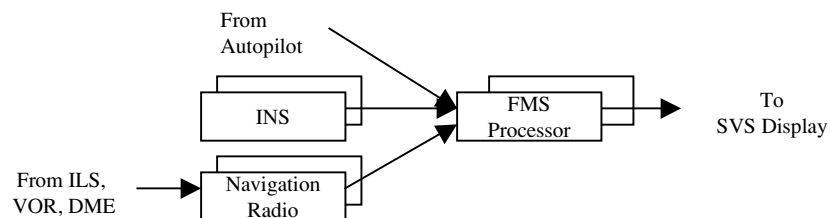
The failure rate for the FMS inertial navigation system (INS) is set to zero in the analysis because the INS is assumed to be the same equipment used by the ADS-B and including failure here would be double-counting.

Autopilot

Autopilot components, shown in Figure 2-8, are based on the description of the Boeing 747 SPZ-1 autopilot in the *1995 Jane's Avionics*.

Figure 2–8. Autopilot
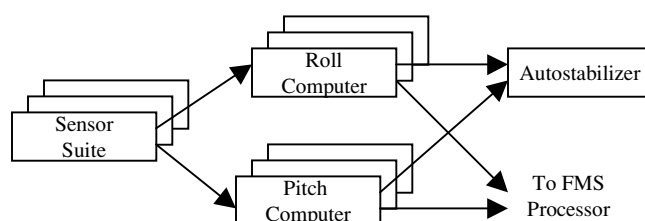
Airport Surface Detection Equipment

ASDE components, shown in Figure 2-9, are based on the primary radar model in the 1999 NASA report, with redundant processors added.[7]

Figure 2–9. ASDE
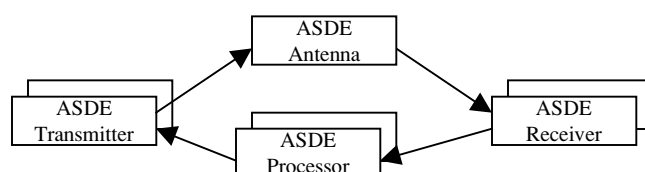
The failure and repair rates for ASDE were taken from the model in the 1999 NASA report, which was based on airport surveillance radar. Two versions of ASDE are being fielded: the Ku-band ASDE-3 and the X-band ASDE-X. The reliabilities and repair times for these two versions are likely to be significantly different.

ADS-B Ground Surveillance Stations

The ADS-B ground surveillance stations components shown in Figure 2-10 are based on technical reports[8] and industry information. The ADS-B ground sensors collect ADS-B transmissions from ground traffic and transmit them to a central processing facility. In this analysis, we assume that the central processing facility is the Airport Movement Area Safety System (AMASS).

Figure 2–10. ADS-B Ground Surveillance Station

---
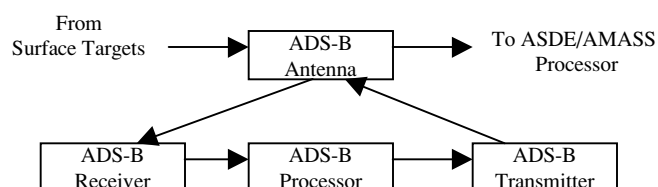
[7] Ibid.

[8] For example, C. Evers, R. Cassell, and D. Lee, "Analysis of ADS-B, ASDE-3, and Multilateration Surveillance Performance—NASA Atlanta Demonstration," presentation at 17th Digital Avionics Systems Conference, AIAA.

We assume that at least two of five stations must be operational to provide full airport coverage.

Airport Movement Area Safety System Processor

The AMASS processor, shown in Figure 2-11, is modeled as a triply redundant processor with cross-strapped, triply-redundant operating system software.

Figure 2–11. AMASS Processor

From ASDE and ADS-B → AMASS Processor → To Controller

The AMASS processor integrates data from ASDE, ADS-B, and any other surface surveillance sensors; calculates trajectories; and generates warnings of predicted collisions.

Common Avionics

The common avionics category is a holding location for airborne equipment that may be needed but is not now considered essential. These equipment items are modeled as single components with nominal failure rates. The three items now included in the common avionics category are the aircraft weather radar, the traffic alert and collision warning system (TCAS), and the enhanced ground proximity warning system (EGPWS).

RELIABILITY ANALYSIS

We used Markov analysis tools developed by NASA to estimate reliability. We used the NASA Scaled Taylor Exponential Matrix (STEM) Markov analysis model to perform reliability calculations. We used the NASA Abstract Semi-Markov Specification Interface to the SURE Tool (ASSIST) to generate inputs for the model. The ASSIST code automates the generation of input files for three Markov analysis models: the semi-Markov unreliability range estimator (SURE), the Padé approximation with scaling (PAWS), and STEM.

Markov analysis and the specifics of the NASA models are well documented,[9] and the mathematics underlying the SURE model are described elsewhere.[10]

---

[9] R. Butler and S. Johnson, *Techniques for Modeling the Reliability of Fault-Tolerant Systems with the Markov State-Space Approach,* NASA Reference Publication 1348, September 1995; S. Johnson and D. Boerschlein, *ASSIST User Manual,* NASA Langley Research Center, September 1993.

[10] R. Butler and A. White, *SURE Reliability Analysis, Program and Mathematics,* NASA Technical Paper 2764, 1988; R. Butler, "The SURE Approach to Reliability Analysis," *IEEE Transactions on Reliability* 41, no. 2 (June 1992).

Appendix A contains a sample of model inputs and results, as well as the input files for all of the systems we analyzed.

RELIABILITY SCENARIOS AND RESULTS

We modeled two reliability analysis scenarios. The first is a terrain avoidance scenario. This scenario consists of an SV-supported IFR approach to a single-runway airport surrounded by high terrain (e.g., Juneau, AK). The terrain prevents the use of radar surveillance or instrument landing equipment. We assume that ground surveillance systems are not deployed at these airports. Traffic is not modeled in this scenario. The equipment content and failure results for this scenario are listed in Table 2-3.

Table 2-3. Terrain Scenario Reliability Results

| | Individual Probabilities | | | | |
	Operational	Degraded 1 degraded LGF	Degraded 2 operational but undetected	Failed Safe	Failed Unsafe non-operational undetected
>=4 GPS Satellites: elev10Lat35	0.99885273			1.15E-03	
>=4 LAAS Reference Stations	0.99999999761			2.39E-09	
LAAS Ground Facility	0.99984402	1.1E-04		3.12E-11	4.61E-05
LAAS TX/RX/Antennas	0.999999997			2.74E-09	
GPS Receivers	0.99998974			2.47E-07	1.00E-05
SVS Processor & Displays	0.999986762		1.20E-05	1.23E-06	2.25E-08
CDTI TX/RX	0.99999952			4.78E-07	
FMS Processor & Radios	0.99999774			2.57E-07	2.00E-06
Autopilot	0.99870083		2.00E-04	1.08E-03	1.50E-05
Total Cumulative Probability Results	0.99737328	1.1E-04	2.1E-04	2.2E-03	7.3E-05

Table 2-3 displays the failure results of all components included in the scenario, which produces the probability that all systems will be operating properly. These component results can be combined in other ways to determine the probabilities of specific hazardous conditions. For example, the most hazardous condition for the terrain scenario is an undetected bias in the terrain display. Such a bias could be caused by an undetected SV processor or display error ($P = 2.248E-8$) that occurs only when no other detected failures have occurred (1- 2.2E-3 or .9978),[11] based on the assumption that known system failures would prompt the aircrew to execute an escape maneuver. The resulting probability would be 2.243E-8. Bias also could be caused by undetected erroneous GPS corrections, which we have not explicitly modeled.

The second scenario addresses traffic avoidance. In this case, the hazard is collision with other traffic—airborne and surface. The components of the system that are important in traffic avoidance include those from the terrain avoidance case as well as airborne and ground surveillance systems. Table 2-4 displays the equipment and the failures of all components in the traffic avoidance scenario.

---

[11] The result is 0.9978 when the probabilities are summed without the SV processor and display.

Table 2-4. Traffic Avoidance Scenario Results

	Individual Probabilities				
	Operational	Degraded 1 degraded LGF	Degraded 2 operational but undetected	Failed Safe	Failed Unsafe
>=4 GPS Satellites: elev10Lat35	0.99885273			1.15E-03	
>=4 LAAS Reference Stations	0.9999999976			2.39E-09	
LAAS Ground Facility	0.99984402	1.10E-04		3.12E-11	4.61E-05
LAAS TX/RX/Antennas	0.999999997			2.74E-09	
>=2 ADS-B Ground Sensors	0.999803026			1.97E-04	
ASDE Primary Radar	0.996001949			4.00E-03	
AMASS Processor	0.999999999996			4.00E-12	
GPS Receivers	0.99998974			2.47E-07	1.00E-05
SVS Processor & Displays	0.999986762		1.20E-05	1.23E-06	2.25E-08
CDTI TX/RX	0.99999952			4.78E-07	
ADS-B	0.99887941		1.60E-05	1.09E-03	1.01E-05
FMS Processor & Radios	0.99999774			2.57E-07	2.00E-06
Autopilot	0.99870083		2.00E-04	1.08E-03	1.50E-05
Total Cumulative Probability Results	0.992077	1.1E-04	2.3E-04	7.5E-03	8.3E-05

As with the terrain scenario, determination of the probability of specific hazardous conditions requires additional thought. For example, a major hazard in the traffic case is own-aircraft loss of the ADS-B signal from the nearest traffic. This hazard could result from an undetected failure of the own-aircraft ADS-B receiver or the traffic aircraft's ADS-B transmitter, again combined with the probability of no known escape-generating failures of other systems. A degraded state could exist if the own-aircraft was still receiving airport surveillance radar data on traffic from CDTI.

SUMMARY

The reliability analysis demonstrates the ability of the modeling approach to generate results that are adequate for integrated safety analysis of AvSP technologies. Individual component results provide insight into the benefits of redundancy and fault detection. The model is capable of accepting higher-fidelity system descriptions for safety analysis of specific hardware and software architectures. We note, however, that higher-fidelity descriptions are useful only if failure and repair data are available for the modeled items. Finally, we note that the reliability results can be readily manipulated to find the probabilities of specific failure combinations.

Chapter 3
Simulation Analysis

GENERAL MODEL APPROACH

The simulation model is used to determine the probability of a hazard or accident, given the existence of a system failure. Simulation also is the primary tool for investigating the impact of human response to equipment failures and the impact of human-initiated errors.

The simulation model is being developed under a parallel task. Unfortunately, it was not completed in time for this report. The analyses discussed in this chapter will be conducted when the model is complete; the results will be included in subsequent documentation.

SIMULATION STRUCTURE

Figure 3-1 outlines the programming structure of the simulation model. Here, as with any model, we must identify the best trade-off between model sophistication and realism versus complexity and processing time. The structure depicted is an object-oriented foundation architecture that is designed to allow "simple" initial models that can be replaced by more realistic algorithms as required.

Figure 3–1. Simulation Structure
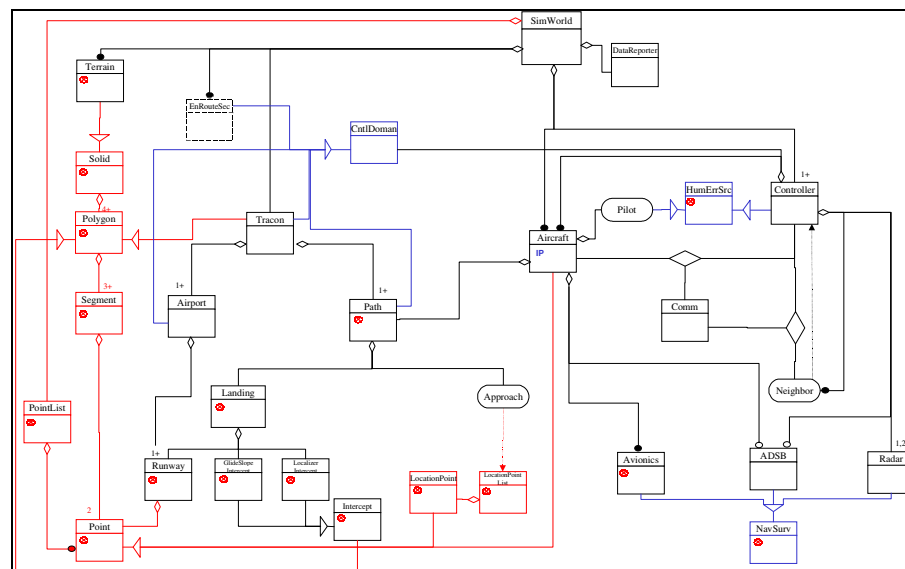
For the current task, aircraft movements are defined by position coordinates and velocity vectors. Basic technology operational performance and failure conditions are modeled by distributions of position and velocity uncertainties. Turn radii and rates are based on simple aerodynamic formulae. Similarly, evasive maneuvers are simple climbing turns. In future model versions, the simple position and velocity definitions can be replaced by more sophisticated aircraft models maneuvers as the analysis demands. The current simple assumptions are considered adequate for assessing feasibility and analyzing the scenarios we are considering in this study.

SIMULATION PARAMETERS

Identification of simulation parameters presents interesting challenges. The national airspace system is as safe as it is because the potential for equipment failures is built into the specification of operating minimums. Determining safety improvements from new technologies under current operating conditions requires us to inject rare blunders into the simulation and see how the new technology improves the chance of survival. Similarly, investigating the safety of higher-capacity operations with new technology requires us to determine the probability of accident when blunders are injected and operational margins are reduced. In summary, we must identify the operating conditions of the current and new technologies, the nature of the blunder, and the expected response to the blunder. Finally, the probability of encountering a blunder must be included in the safety calculation:

$$P_{\text{Accident}} = P_{\text{Failure}} * (P_{\text{Accident}} \mid \text{Failure}) * P_{\text{Blunder}}$$

Terrain Avoidance

The terrain avoidance scenario is concerned with controlled flight into terrain (CFIT). For terrain avoidance, we are concerned with simulating the approach of the aircraft through an obstacle-strewn path to a single runway airport. The scenario is modeled on the required navigation performance RNP approach to Juneau, AK, airport Runway 26. The approach path consists or a series of three-dimensionally defined waypoints that define a path between mountains to the runway. Neither radar surveillance nor radio landing aids are available. Figure 3-2 generally depicts the scenario.

Figure 3–2. Terrain Avoidance Scenario
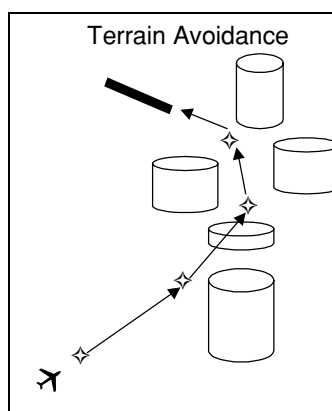
The most hazardous failure of either the current RNP equipment or the SV system in the terrain avoidance scenario is an undetected bias (also known as map error)in the FMS/INS/GPS (for RNP) and database or the display (for SV). In both cases, the aircraft will deviate from the flight path until the aircraft is below the minimum decision altitude (MDA), at which time we assume the pilot will detect the error and attempt a recovery maneuver. The hazard obviously depends on the size and direction of the bias and height of the MDA. It also will depend on the response time of the pilot and the probability that the pilot will choose the correct escape maneuver. The improvement in safety from SV with LAAS will be a function of any reductions in crashes and the reduced probability of combined undetected failure of the FMS/INS/GPS and the SV equipment. Improved capacity safety will be modeled by determining the lowest MDA, where the SV system provides the current level of safety given an undetected bias.

The current scenario is fairly basic. The model is inherently capable of more sophistication. For example, if the EGPWS is independent of the SV, the probability that the aircrew will detect and respond to the bias can be modeled. The same modeling can be carried out if the weather radar is used to corroborate the registration of the SV display with actual terrain. Such analyses are left for the future.

The model requires means and uncertainties for the own-aircraft state parameters as a function of technology, for normal and blunder conditions. Table 3-1 shows these parameters for the terrain avoidance case.

Table 3-1. Terrain Avoidance State Parameters

Airborne information state	Own-Aircraft Navigational Data (AND) 1 standard deviation uncertainties			
	position (nmi./m.)	speed (kts.)	Track (deg.)	latency (sec.)
Fully operational SV (LAAS, FMS)	0.0005/1.0	0.2	0.1	0.0
RNP (GPS, FMS, no LAAS or WAAS)	.011/20	0.5	1.0	0.0
Degraded RNP (INS only, no GPS or navaids)	1.0 m + 0.5 m/sec	5.0	0.5	0.0
Blunder (undetected INS/FMS drift or SV database bias)	0.1-0.5 nmi. bias	0.2	0.1	0.0

Finally, the simulation requires estimates of the probability that the aircrew will detect an error, the aircrew's expected response time, the nature of the aircrew's response, and the probability that the aircrew will choose the correct response. For the terrain avoidance case, we assume that the aircrew will detect the unsafe condition 2.0 seconds after reaching the MDA, respond in 1.0 second, and always respond with a climbing turn away from terrain.

FMS or SV drifts are basically caused by equipment failures; we need not include an additional blunder probability. The FMS and SV drifts can be modeled in more detail to improve the estimate of drift occurrence.[1]

Traffic Avoidance Scenario

The traffic avoidance scenario is concerned with midair and runway collisions. The scenario is depicted in Figure 3-3; it includes own-aircraft and traffic approaches to parallel runways, as well as ground traffic on the own-aircraft runway. Full radar surveillance and ILS landing aids are assumed.

Figure 3–3. Traffic Avoidance Scenario
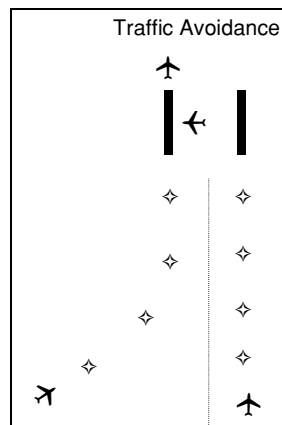
Two failures are considered in the traffic avoidance case. The first is an obstructed runway condition. The current-technology aircraft will be able to detect the threat visually when it reaches the MDA. The SV-equipped aircraft, supplemented by CDTI, will be able to detect the threat well out on the glideslope.

The second failure in the traffic avoidance scenario is a blunder toward the own-aircraft by the parallel traffic aircraft. This failure will be modeled with and without simultaneous loss of ADS-B data from the traffic aircraft. We further assume that the controller is able to detect the threat in the former case on the basis of surveillance radar information. In both cases, we assume that the own-aircraft can detect the intruder visually when they are both below the MDA. The probability of collision will be based on several factors, including the time required for the

---

[1] For further detail on sources of FMS drift, see "B737-300/400/500 FMC Position," available at <http://194.78.76.133/linepilot/FMCPosition.html>.

controllers and aircrews to detect and respond to the threat, the height of the MDA, and the runway separation. The probability of a blunder is small. The Federal Aviation Administration's Precision Runway Monitor Program Office estimates that the chance of a blunder is 1 in 25 million approaches.[2]

The traffic scenario requires state variables for the own-aircraft's knowledge of itself and the traffic, and the controller's knowledge of airborne and ground traffic. Tables 3-2 and 3-3 contain the state variables.

Table 3-2. Traffic Scenario State Parameters (Aircrew)

Airborne information state	Own-Aircraft Navigational Data (AND) 1 standard deviation uncertainties				Traffic Navigational Data (TND) 1 standard deviation uncertainties			
	Position (nmi./m.)	Speed (kts.)	Track (deg.)	Latency (sec.)	Position (nmi./m.)	Speed (kts.)	Track (deg.)	Latency (sec.)
Fully operational (LAAS, FMS)	0.0005/1.0	0.2	0.1	0.0	0.0005/1.0	0.2	1.0	2.0
Degraded (GPS, FMS; no LAAS or WAAS)	.011/20	0.5	1.0	0.0	0.011/20	0.5	1.0	2.0
Blunder (no ADS-B from traffic)	0.0005/1.0	0.2	0.1	0.0	n/a	n/a	n/a	n/a

Table 3-3. Traffic Scenario State Parameters (Controller)

Controller information state	Airborne Traffic Data (1 standard deviation uncertainties)				Surface Traffic Data (1 standard deviation uncertainties)			
	Position (nmi./m.)	Speed (kts.)	Track (deg.)	Latency (sec.)	Position (nmi./m.)	Speed (kts.)	Track (deg.)	Latency (sec.)
Fully operational (LAAS, ADS-B, ASDE)	0.0005/1.0	0.2	1.0	2.0	1.0 m	2.0	1.0	1.0
Degraded (GPS, FMS, no LAAS or WAAS)	0.011/20	0.5	1.0	2.0	0.011/20	0.5	1.0	2.0
Blunder (no ADS-B; ASR/ASDE only)	0.25	5.0	2.0	4.0	1.5 m	2.0	4.3	1.0

The traffic scenarios are more complex than the terrain scenario but are still relatively basic. Future additions will include intrusion alerting algorithms, detailed aircraft-specific aerodynamics, and more complex traffic patterns.

General Parameters

In addition to the state variables, we also need to quantitatively define the general parameters mentioned in Chapter 1:

- ◆ Pilot and controller detection times

- ◆ Pilot and controller communication times

- ◆ Pilot and controller response times

- ◆ Traffic density:

  - ➤ *Detection times:* Pilot and controller detection times are assumed to be two cycles of the appropriate surveillance sensor. For ADS-B and ASDE, the update rate is 1 Hz, and the detection time is 2 seconds.

---

[2] B. Carpenter and J. Kuchar, *A Probability-Based Alerting Logic for Aircraft on Parallel Approach,* NASA CR 201685, April 1997.

For the airport surveillance radar, the update rate is 0.25 Hz, and the detection time is 8 seconds.

➤ *Communication times:* Communication time applies to controller-to-pilot communications in cases in which the controller detects the threat. We use a communication time of 4 seconds.

➤ *Pilot and controller response time:* We assume a pilot response time of 2 seconds. This value is in accord with other analyses and with simulations. The controller is expected to respond immediately upon detection of a problem.

➤ *Traffic density:* Traffic is not modeled in the terrain scenario. For the current cases, only immediate-threat aircraft are included in the traffic scenario. The simulation structure is capable of full airport traffic modeling.

SUMMARY

We have chosen scenarios and parameters to demonstrate the feasibility of the estimating safety benefits of AvSP technologies. The success of the simulation effort is fully expected but, unfortunately, still unproven.

The foregoing parameters are based on interpolation from several sources. Appendix C contains a summary of references and parameter values.

Chapter 4
Human Factors Evaluation

This chapter summarizes the results of a subcontracted evaluation of the human factors modeling capacity of the LMI models. The analysis was conducted by Dr. Kevin Corker, San Jose State University.

There are three active functional elements in the LMI system. The first is *system functionality,* which can be described as a discrete state representation of the components. This element is addressed by Markov reliability analysis. The second element is the *operational scenario,* which provides the event driver for the simulation process. The third element is *rules and procedures;* in the current implementation, this element lacks an active agent—typically a human operator—to interpret, implement, and execute the rules and procedures. The role of an active agent (human element) can be filled by a submodel (or component model) that interacts with the simulation. Active-agent submodels can be identified for each active human element in the system to represent the response of the human component in the system. There can be different rules and procedures for each active agent. In addition to the diversity of rules and procedures, human performance also can be subject to processes that either degrade or improve a baseline performance.

The system architecture allows elaborated human operator performance models to contribute the output of the rules and procedures element of the system. The output of the human operator is not necessarily limited to failures or degradation. The human response to loads such as traffic density is to choose interpretations of rules to suit the required system state. The human operator may exhibit behavior that is more optimal if aiding systems are in place to improve the optimality of the system. The model requires a representation of the human operator that is consistent with the representation of the component systems and scenarios.

The standard representation of time lag is not sufficient to account for the parts of the perceptual and cognitive processes that will be affected by specific aiding technologies (e.g., enhanced SV). To accurately account for the impact of such technologies, there should be transition processes that represent the mechanisms of impact. In the SV case, there should be a modeled transition that represents the probability that the pilot will see potential obstructions. The effect of seeing something or not then changes the probability of one rule of behavior (e.g., if runway is clear, proceed with takeoff) as compared to another (if runway is occupied, reject takeoff). The change of rule probability would then affect the selection of behavior.

The probability of successful completion of an action also should be made responsive to conditions in the environment. For example, if the pilot encounters a

high task load because of traffic, weather, or communications, some reduction in the probability of successful performance could be implemented. If there are technologies to aid the performance of the output behavior, that technology—in combination with human performance transitions—should show an increase in the probability of successful performance. To summarize, the human performance submodel should support mechanisms by which the presumed performance environment can impact the output of that model.

Discrete event simulation human element submodels are consistent with the basic LMI discrete event simulation architecture. Specifically, current research with dynamically colored Petri net simulations could be applied to the LMI analysis. The dynamically colored representation adds functions to a standard Petri net that support the functionality required to represent improvements in safety (or capacity enhancement) that are being evaluated. This method has been explored and developed by the Dutch Aerospace Research Laboratory (NLR) and applied to several aeronautic scenarios. In collaboration with the NLR, San Jose State University (under NASA safety program funding) is making the Man-machine Integration Design and Analysis System (MIDAS) model a source of parameters to guide transitions in the Petri net simulation. The MIDAS human performance models respond to technological and procedural interventions. Discrete events generated by the Petri net model as a result of MIDAS input potentially can be fed into the standard risk assessment processing of the LMI simulation.

Figure 4-1 is a schematic of the MIDAS/Petri net interaction. The figure depicts the situation in which a pilot alternates between informed and uninformed states regarding traffic on the runway. Rules and procedures that are appropriate for each state are contained in the human performance matrix (shown twice for clarity). The state circles and transition rectangles are elements of the Petri net. The dot is a token that indicates that the pilot is in the "informed" state. Information about the external world and the pilot's internal human processes as modeled in MIDAS can stimulate or inhibit transitions between the informed and uninformed states, with corresponding changes in the pilot's expected performance. In this representation, the Petri net and MIDAS models interact to provide a process with sufficient complexity to represent human performance with a link to the discrete event simulation. In terms of safety analysis, the information processing element of the models allows error and degradation of performance in the workings of the MIDAS models. In addition, technologies that might improve awareness and/or increase speed or accuracy will be represented through the action of MIDAS models whose outputs modify the colored transitions in the Petri net representation. The output of the Petri net can then interact with the LMI safety/risk assessment through the updated human performance matrix.

Figure 4-1. Human Element Model Schematic
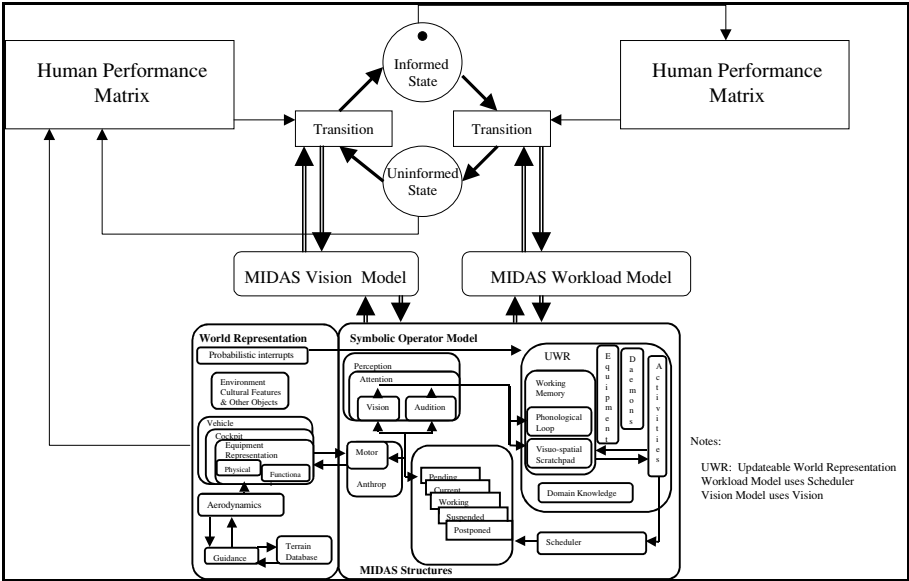
Appendix A
Reliability Data

As discussed in Chapter 2, the failure and recovery rates used in the analysis are from three sources. A 1999 NASA report is the basis for several of the components modeled in the analysis.[1] The rates in this report were based partly on government specifications and partly on engineering judgment. To substantiate the engineering judgment, we investigated two additional sources. The first was the Air Force "DO41" peacetime maintenance database for the 2 years (8 quarters) ending in March 2000. The second was *Jane's Avionics*. Although the new data added insight, considerable engineering judgment remains in the rate selection.

The DO41 data are extracted from a database that LMI produced and maintains for the Air Force. The characteristics of the specific data used here include the following:

◆ *Data type:* Peacetime failure data

◆ *Data period:* March 1999 and previous 8 quarters (2 years)

◆ *Data records:* Total organizational and intermediate demand rates (TOIMDR)—unscheduled maintenance data for actual failures; failed to replicate cases are removed from the database

◆ *National Stock Number (NSN) categories searched:*

➤ 5281: Radio and TV Communications—Airborne

➤ 5826: Radio Navigation Equipment—Airborne

➤ 6605: Navigation Instruments

The data were transferred to a Microsoft ACCESS database and searched to extract data on receivers, transmitters, inertial navigation/measurement systems, and so forth. The resulting tables were converted to Excel spreadsheets for analysis and display.

The equipment descriptions included in the DO41 data do not provide enough information to determine the assembly level of the equipment listed. The On-Line Federal Logistics Information System (Haystack service) was searched by NSN to gather more descriptive data. The Haystack data were added to the spreadsheets. Even with these data, questions remain. For future work, explicit information can

---

[1] P. Kostiuk et al., *A System for Integrated Reliability and Safety Analysis,* NASA CR—1999-209548, August 1999.

be found through research into the Air Force work unit codes (WUCs) and interviews with Air Force logistics personnel.

Several issues of *Jane's Avionics* were searched to extract equipment descriptions any failure data. Supplemental data found in *Jane's Avionics* were added to the spreadsheets.

Tables A-1 through A-3 are spreadsheets for receivers, transmitters, and inertial reference units. Study of these tables indicates the degree to which engineering judgment must be applied to the choice of rates for analysis.

Table A-1. Receiver Data

Receivers						
					MTBF (hours)	
		Use for Modeling Receiver MTBF				
		Ground Receivers			10,000	
		Airborne MMR-type Receivers			20,000	
NSN	ITEM	Description	Note	Manufacturer	MTBF	
5821006829337	RECEIVER	Airborne radio comm set	7	RC	25000	
5826010863038	RECEIVER	Receiver for ARN-11 nav radio		RC	22222	
5821010774298	RECEIVER A	Receiver for ARC-186 VHF radio	1	RC	20000	
5826010857281	RCVR PR UN	Receiver, detector & clock for ARN-131		Northrup	17241	
5821014451430	REC TRANSM	ARC-230	2	RC	15873	
5821011372976	RECRT1168B	ARC-164(V) UHF	4	Raytheon	12658	
5826010131836	RECEIVER	AWACS Type 411L receiver		Allied	10417	
5826009941578	RECEIVER	ARN-83 Direction Finder		RC	6211	
5821014173216BY	REC TRANS	JTIDS II		RIC	5236	
5826014411054LD	RECEIVER,	ARN-155 ILS radio	3	Marconi	4505	
5821011369512	REC/TRANS	ARC-164(V) UHF (2000 MTBF in Jane's)	4	Raytheon	4032	
5821014209652	REC/TRAN	ARC-222 (SINCGARS) replaces ARC-186		Raytheon	3650	
5821013016357	REC.TRANSM	ARC-171(V) UHF	6	RC	3610	
5821013925718	RCVR TRANS	ARC-190 HF (lots of aircraft, 2000 Jane's)		RC	2571	
5821013063385	RECEIVER T	ARC-164 UHF Receiver-Transmitter, Radio	4	Raytheon	2188	
5826000606055	RCVRADF203	ARN-83 Nav Radio		RC	1992	
5821013740202	RECEI-TRAN	ARC-187 UHF (P-3, FltSatCom capable)		Raytheon	1887	
5821010621019	REC/TRANSM	ARC-186 VHF (9000 Hrs. MTBF in Jane's)	1	RC	1802	
5826002266030	RCVR.ARN58	ARN-58 receiver		Spartan	1362	
5826010211744FX	REC/ARN112	ARN-112? F-15		RC	1276	
5826007210168	RAI REC ST	ARN-67 Nav radio		RC	1258	
5826013681538FW	RECEIVER T	Pacer Special		RC	1201	
5826013842173FW	RECEIVER,T	Pacer Special		RC	1033	
5821011581319	REC TRANS	ARN-171(V) EC-135 UHF		RC	890	
5826014331555NS	RECVR,RADI	Navstar (GPS)		RC	865	
5826010121938	REC TRANS	ARN-118(V) TACAN	5	RC	853	
5826013671307FW	RECEIVER-T	ARC-215(V) Pacer Special		RC	787	
5821011416133	RECRT1168B	ARC-164 UHF	4	Raytheon	698	
5821011948161	RECTRANS	ARC-171X(V) UHF	6	RC	653	
5826010080540	REC COMPUT	ARN-120 Nav Set (Omega)		Northrup	637	
5821012287058	REC TRANS	ARC-164(V)	4	Raytheon	560	
5821013820706BY	REC-TRANS	URC-107(V)1 (200W) JTIDS 960-1215 MHz		RC	441	
5821013015620	RECEIVER T	ARC-164(V) 100W UHF	4	Raytheon	370	
Notes						
	1	ARC-186: Standard VHF AM/FM, 28,000 units produced, F-16, A-10, C-130, 9000 MTBF demonstrated, 1995 *Jane's*, pg 676				
	2	ARC-230, Standard HF radio, aka ARC-153, -157, -191(V), -207(V), URC-91, - 97(V), ARC-512, pg 77 2000 *Jane's*				
	3	ARN-155: PLSR Precision Landing System Receiver, ILS,MLS,GPS,DGPS VHF DL, 10,500 hr. MTBF per MIL-HDBK-217E 2000 *Jane's* pg 540				
	4	ARC-164(V): Std AF UHF radio, 2000 hr MTBF in 2000 *Jane's* pg 69, 1884 *Jane's* pg 699, 42,000 produced by 1984				
	5	ARC-118(V): TACAN, 38,000 units, entered service 1975, 2000 *Jane's* pg 559				
	6	ARC-171(V): UHF family of radios: AM through AM/FM/FSK , 2000/yr prod rate in 1984, 1984 *Jane's* pg 37				
	7	RC = Rockwell Collins				

Table A-2. Transmitter Data

Transmitters						
		Use for modeling Transmitters	**MTBF**			
		Airborne	**10,000**			
		Ground	**10,000**			
NSN	ITEM	Description	Manufacturer	MTBF	PRICE $	
5821010607326	TRANS REC	ARC-171-1C	RC	500,000	2,122	
5821010649794	REC+TRANS	ARC-171X(V) RT-1264(V)4	RC	333,333	33,100	
5821013296779	REC TRANSM	ARC-187(V) UHF Radio	Raytheon	71,429	2,820	
5826010480212	TRANSMIXER	Matched Set, TX and Mixer, E-3A AWACS	RC	55,556	5,878	
5821013292568	TRANS RADI	ARC-187(V) UHF Radio	Raytheon	43,478	8,510	
5821014451430	REC TRANSM	ARC-230	RC	15,873	57,911	
5826010869637	TRANSMITTR	ARN-118	RC	12,048	4,019	
5821010704457	TRANSMITTE	ARC-164(V)	Raytheon	8,696	8,333	
5821014173216BY	REC TRANS	JTIDS-II	RC	5,236	212,640	
5821011369512	REC/TRANS	ARC-164(V)	Raytheon	4,032	8,990	
5821013016357	REC.TRANSM	ARC-171(V) RT-1270(V)2	RC	3,610	34,225	
5821013925718	RCVR TRANS	ARC-190 RT-1341(V)	RC	2,571	36,031	
5826012996162	REC TRANS	RT-1364	BF Goodrich	1,818	18,819	
5821010621019	REC/TRANSM	ARC-186(V) RT-1300	RC	1,802	9,033	
5821014221645	REC-TRANS	ARC-187(V) RT-1571A	Raytheon	1,681	55,634	
5821013943306LG	REC-TRANS	RT-9600F no other data	Allied Signal	1,650	11,465	
5821011193956	TRANSCEIVR	AWACS VHF/AM	RC	1,565	22,150	
5821011581319	REC TRANS	ARC-171(V) RT-1440 EC-135	RC	890	34,016	
5826013821577BY	REC TRANSM	URC-107(V) F-15 JTIDS II (Sub-assy)	RC Bae	881	25,951	
5826010121938	REC TRANS	ARN-118(V) RT-1159/A TACAN	RC	853	18,549	
5821011948161	RECTRANS	ARC-171X(V) RT-1270(V)2	RC	653	29,286	
5821012287058	REC TRANS	ARC-164(V) RT-1504	Raytheon	560	16,312	
5821013820706BY	REC-TRANS	URC-107(V)1 JTIDS	RC, Bae	441	199,870	
5821005764866	TRANSMITTE	ARC-164(V) T-1307	Raytheon	90	3,454	
Notes						
	1	ARC-186: Standard VHF AM/FM, 28,000 units produced, F-16, A-10, C-130, 9000 MTBF demonstrated, 1995 Jane's, pg 676				
	2	ARC-230, Standard HF radio, aka ARC-153, -157, -191(V), -207(V), URC-91, - 97(V), ARC-512, pg 77 2000 Jane's				
	3	ARN-155: PLSR Precision Landing System Receiver, ILS,MLS,GPS,DGPS VHF DL, 10,500 hr. MTBF per MIL-HDBK-217E 2000 Jane's pg 540				
	4	ARC-164(V): Std AF UHF radio, 2000 hr MTBF in 2000 Jane's pg 69, 1884 Jane's pg 699, 42,000 produced by 1984				
	5	ARC-118(V): TACAN, 38,000 units, entered service 1975, 2000 Jane's pg 559				
	6	ARC-171(V): UHF family of radios: AM through AM/FM/FSK , 2000/yr prod rate in 1984, 1984 Jane's pg 37				

Table A-3. Inertial Reference Equipment

Inertial Navigation System / Inertial Navigation Unit						
				MTBF (hours)		
		Use for INS/IRU/IMU Modeling:		5000		
INS/IRU/IMU Reliability Data						
The C-17 is the latest technology IMU with the best data in Air Force database						
NSN	ITEM	Description	Manufacturer	MTBF	PRICE $	
6605013399252	C17 INU	C-17 INU	Honeywell	4255	124221	
6605013524574	F117 INU	F-117 INU (no data)	Honeywell	2,439	80,000	
6605014260552	INU,EMBGPS	A-10 INU - 2 GPS antennas	Honeywell	2,203	67,000	
6605013803681FW	IMU	Pacer Special IMU	Northup, Kearfott	1,033	1,565,892	
6605013578976	STD INU	F-16 self-contained INU	OCALA, Litton	742	104,950	
6605014557795	F16 RLG	F-16 A-D model INU	OCALA, Litton	694	103,150	
6605013574519	F-15 RLG	F-15 A-E model INU	Honeywell	460	112,151	
6605013471667	SPA INU	C-130E INU	Kearfott	370	386,141	
6605010182181	INU	Nav Computer including gyros	Litton, Delco	336	132,470	
6605012562380	INU LN39	A-10A, F-16 C/D, FB-111 INU	Litton	259	108,785	
6605010847343	INU	AWACS AN/ASN-119 INU	Delco, Northup, Boeing	225	178,889	
6605012529480	INERT NAV	B-1B INU AN/ASQ-149	Kearfott	206	198,988	
6605010787915BF	IMU	F-4E INU	Kearfott	133	107,303	
6605010876645WF	INU 74DA0	F-16 INU roll,pitch, course dev.		80	149,875	
Data from 1995 *Jane's Avionics*						
Note	System		Manufacturer	MTBF		
1	Carousel IV Inertial Nav System (AN/ASN-119)		Delco	3000+		
2	MAGR/RCVR 3M GPS Receiver Family		Rockwell Collins	5000+		
3	HG2001 Advanced Inertial Ref Unit		Honeywell	10,000		
4	Integrated Global Positioning/Inertial Ref Sys		Honeywell			
	IRS			5,000		
	GPS Sensor Unit			20,000		
5	Laser Inertial Ref Unit		Honeywell	7,000		
6	H-423 Ring Laser Inertial Nav Sys		Honeywell	2,000		
			Honeywell	4,000		
7	LTN-90 Ring Laser Gyro Inertial Ref Sys					
	Inertial Ref unit		Litton	2,500		
	Mode Select unit		Litton	50,000		
	Inertial sensor display unit		Litton	15,000		
	Notes					
1	on commercial aircraft					
2	Military Airborne GPS Receiver					
3	GG1320 laser gyro					
4	demonstrated in 767,757,737,MD-80,MD-11,A320,A330,A340,F100					
5	guaranteed fighter f-15E, F-117 upgrade, others					
6	guaranteed transport, C-17					
7	A-310, A-300-600, Navy E-6A TACAMO					

Appendix B
Reliability Analysis Background and ASSIST Listings

This appendix briefly describes the Markov reliability analysis and documents the model input files.

The reliability analysis was performed with the NASA Scaled Taylor Exponential Matrix (STEM) Markov analysis model. The inputs for the model were generated by using the NASA Abstract Semi-Markov Specification Interface to the SURE Tool (ASSIST). The ASSIST code automates the generation of input files for three Markov analysis models: the semi-Markov unreliability range estimator (SURE), the Padé approximation with scaling (PAWS), and STEM.

Markov analysis and the specifics of the NASA models are well documented,[1] and the mathematics underlying the SURE model is described elsewhere.[2] Therefore, in this appendix we present only an outline of the method to illustrate the process.

PROCESS OUTLINE AND EXAMPLE CASE

The analysis is performed in four steps. The first step is definition of components to be analyzed and determination of appropriate failure, repair, and fault detection coverage rates. The second step is the writing of an ASSIST input file. The third step is running ASSIST to generate input files for the Markov analysis programs. Finally, a Markov analysis model is run to generate reliability results. The following paragraphs detail the process for a simple example: cockpit display of traffic information (CDTI).

In our analysis, CDTI is modeled as two ground transmitters and two airborne receivers. Ground transmitters can be repaired. We define a three-element state in which the first element is the number of operating transmitters, the second is the number of operating receivers, and the third is a failure flag that is 0 for operating conditions and 1 whenever all transmitters and/or all receivers have failed. Figure B-1 displays the potential states of the system.

---

[1] See R. Butler and S. Johnson, *Techniques for Modeling the Reliability of Fault-Tolerant Systems with the Markov State-Space Approach,* NASA Reference Publication 1348, September 1995; S. Johnson and D. Boerschlein, *ASSIST User Manual,* NASA Langley Research Center, September 1993.

[2] R. Butler and A. White, *SURE Reliability Analysis, Program and Mathematics,* NASA Technical Paper 2764, 1988; R. Butler, "The SURE Approach to Reliability Analysis," *IEEE Transactions on Reliability* 41, no. 2 (June 1992).

Figure B–1. CDTI Markov Tree Diagram
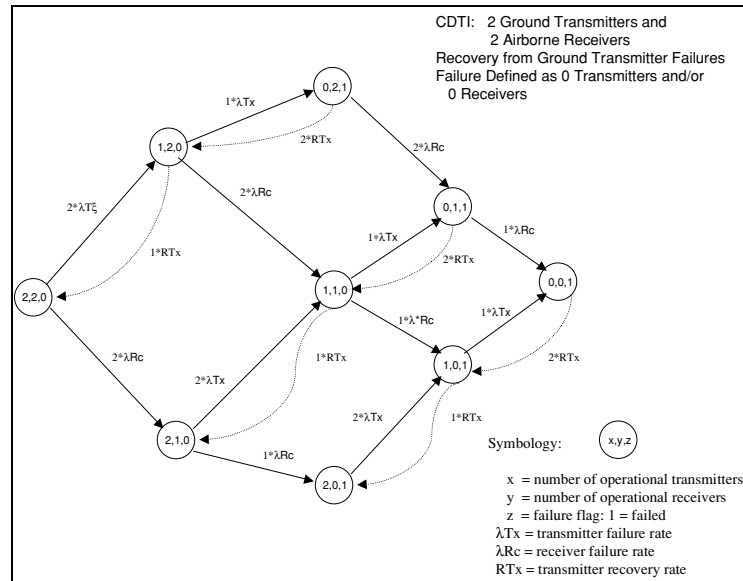
Note that there are only 9 states and 18 transitions (arrows) in the tree. More complex systems can have hundreds of states and thousands of transitions, which become practically intractable without models such as ASSIST.

The purpose of the ASSIST model is to automatically generate the states and transitions, based on a description of system components as well as failure and recovery processes. By replacing numbers with algebraic variables, ASSIST can generate huge state/transition files, using very succinct code statements. The ASSIST coding for the CDTI is listed in Figure B-2.

Figure B–2. CDTI ASSIST Listing

```
(* Cockpit Display of Traffic Information (CDTI) ASSIST model *)
(* This model uses ADS_B receiver and transmitter parameters, *)
(* but assumes no uncovered hardware failures *)
(* Repair capability is added to the ground receiver and transmitter *)
(* Use of a common antenna with other equipment is assumed for both ground and
air *)
(* There is only one CDTI per airport *)

LIST = 3;                       (* Needed for the .mod file *)

(* Numbers of each component *)
n_gtx =  2;  (* Modulator and Transmitter *)
n_arx =  2;  (* Receiver and Demodulator *)

(* other parameters *)
n_modes = 1;         (* Number of system failure modes which
               will be differentiated in model *)

(* Failure rates -- per hour *)
l_gtx    = 1e-4;       (* Ground Transmitter *)
l_arx    = 5.0e-5;       (* Airborne Receiver *)

(* Recovery rates - per hour *)
R_gtx     = 0.25;       (* Ground Modulator and Transmitter repair, 4 hrs *)

(* State space definition:            *)
(* m_gtx # Number of on-line Ground Transmitter channels *)
(* m_arx # Number of on-line Airborne Receiver channels *)
(* f_mode # Failure mode: 0 = operational, 1 = failed safe *)
```

```
SPACE = (m_gtx: 0..n_gtx, m_arx: 0..n_arx, f_mode: 0..n_modes);

(* Starting Info *)
START = (n_gtx, n_arx, 0);

(* Set up failure rates *)

(* Failure of Ground Transmitter channel *)
if (m_gtx > 1) tranto m_gtx = m_gtx - 1, by m_gtx* l_gtx;
if (m_gtx = 1) tranto m_gtx = m_gtx - 1, f_mode = 1 by m_gtx* l_gtx;

(* Failure of Airborne Receiver channel *)
if (m_arx > 1) tranto m_arx = m_arx - 1, by m_arx* l_arx;
if (m_arx = 1) tranto m_arx = m_arx - 1, f_mode = 1 by m_arx* l_arx;

(* Set up recovery rates *)
(* Recovery of Ground Transmitter *)
IF (m_gtx < n_gtx) AND (m_gtx > 0) TRANTO m_gtx = m_gtx + 1, BY (n_gtx-m_gtx) *
R_gtx;
IF (m_gtx = 0) THEN
  IF (m_arx <> 0) THEN
    TRANTO m_gtx = m_gtx + 1, f_mode = 0 BY R_gtx;
  ELSE
    TRANTO m_gtx = m_gtx + 1 BY R_gtx;
  ENDIF;
ENDIF;
```

In this listing, the numbers of transmitters and receivers—*n_gtx* and *n_arx*—are variables (now set to 2, although they could be changed at will). The key to the ASSIST code is the transfer (TRANTO) statement, which controls the transfers among states:

```
 if (m_gtx > 1) TRANTO m_gtx = m_gtx – 1 by m_gtx * l_gtx;
```

This statement says that for states in which more than one transmitter is operating, the assist code will generate a transfer to a state where one less transmitter is operating with a failure rate equal to the number of operating transmitters times the failure rate for transmitters (this rate assumes continuously operating spares). Thus, one statement can control all transitions that involve states with more than one operating transmitter. In the present case, we only have two transmitters; although the efficiency is not great, as the number and complexity of components increases, the power of the method becomes clear.

Running ASSIST generates a *".mod"* input file that can be used with any of the three Markov analysis models. The *cdti.mod* file listing is shown in Figure B-3.

Figure B–3. CDTI ASSIST Output File

```
LIST = 3;
N_GTX = 2;
N_ARX = 2;
N_MODES = 1;
L_GTX = 1E-4;
L_ARX = 5.0E-5;
R_GTX = 0.25;


  1(* 2,2,0 *),   2(* 1,2,0 *)    = 2*L_GTX;
  1(* 2,2,0 *),   3(* 2,1,0 *)    = 2*L_ARX;
  2(* 1,2,0 *),   4(* 0,2,1 *)    = 1*L_GTX;
  2(* 1,2,0 *),   5(* 1,1,0 *)    = 2*L_ARX;
  2(* 1,2,0 *),   1(* 2,2,0 *)    = (N_GTX-1)*R_GTX;
  3(* 2,1,0 *),   5(* 1,1,0 *)    = 2*L_GTX;
  3(* 2,1,0 *),   6(* 2,0,1 *)    = 1*L_ARX;
```

```
          4(* 0,2,1 *),    7(* 0,1,1 *)    = 2*L_ARX;
          4(* 0,2,1 *),    2(* 1,2,0 *)    = R_GTX;
          5(* 1,1,0 *),    7(* 0,1,1 *)    = 1*L_GTX;
          5(* 1,1,0 *),    8(* 1,0,1 *)    = 1*L_ARX;
          5(* 1,1,0 *),    3(* 2,1,0 *)    = (N_GTX-1)*R_GTX;
          6(* 2,0,1 *),    8(* 1,0,1 *)    = 2*L_GTX;
          7(* 0,1,1 *),    9(* 0,0,1 *)    = 1*L_ARX;
          7(* 0,1,1 *),    5(* 1,1,0 *)    = R_GTX;
          8(* 1,0,1 *),    9(* 0,0,1 *)    = 1*L_GTX;
          8(* 1,0,1 *),    6(* 2,0,1 *)    = (N_GTX-1)*R_GTX;
          9(* 0,0,1 *),    8(* 1,0,1 *)    = R_GTX;

     (* NUMBER OF STATES IN MODEL = 9 *)
     (* NUMBER OF TRANSITIONS IN MODEL = 18 *)
     (* 0001 WARNING *)
```

Running the STEM model with the input file is straightforward. The command is `stem cdti.mod.` Because the analysis is time-dependent, we need to define the time period. For this case, we use 10 hours (based on the assumption that all equipment is operating at the beginning of the flight). The STEM command line statement is simply `time = 10`. The statement `run cdti.out` causes STEM to execute and store the results in a file `cdti.out`. The `cdti.out` file is shown in Figure B-4. Note that the *L* in the numbers represents the base 10 exponent.

Figure B–4. CDTI 10-Hour Output File

```
      MODEL FILE = cdti.mod          STEM V7.9.8 Mon Nov 27 13:45:48 2000


        *** START STATE ASSUMED TO BE 1

                       ----- RUN #1


        D-STATE    PROBABILITY    ACCURACY
        -------  -------------------  --------

        -------  -------------------
        TOTAL   0.0000000000000L+00  12 DIGITS


        STATE     PROBABILITY
        -------  -------------------
          1   9.9826721154481L-01
          2   7.3306059566549L-04
          3   9.9851681994737L-04
          4   2.2769290483337L-07
          5   7.3324389136242L-07
          6   2.4969162269060L-07
          7   2.2774983754797L-10
          8   1.8335680822273L-10
          9   5.6951696124529L-14

      -------------------------------------------------------
```

The analysis automatically checks for absorbing states that have no existing transitions. In the CDTI case, there are none of these *D* or "death" states. We do have quasi-death conditions, however, because there is no repair of airborne receivers. The probability that we will find the system in states with failed receivers grows as time gets longer. Table B-1 and Figure B-5 display the results for 10 hours, 10,000 hours, and 100,000 hours.

Table B-1. CDTI State Probabilities Versus Operating Time

	State populations			State probabilities at time =		
State	Operational transmitters	Operational receivers	Failure code	10 hours	10,000 hours	100,000 hours
1	2	2	0	9.98E-01	3.68E-01	4.54E-05
2	1	2	0	7.33E-04	2.94E-04	3.63E-08
3	2	1	0	9.99E-04	4.77E-01	1.34E-02
4	0	2	1	2.28E-07	1.18E-07	1.45E-11
5	1	1	0	7.33E-07	3.82E-04	1.07E-05
6	2	0	1	2.50E-07	1.55E-01	9.86E-01
7	0	1	1	2.28E-10	1.53E-07	4.28E-09
8	1	0	1	1.83E-10	1.24E-04	7.89E-04
9	0	0	1	5.70E-14	4.95E-08	3.15E-07

Figure B–5. CDTI State Probabilities Versus Operating Time
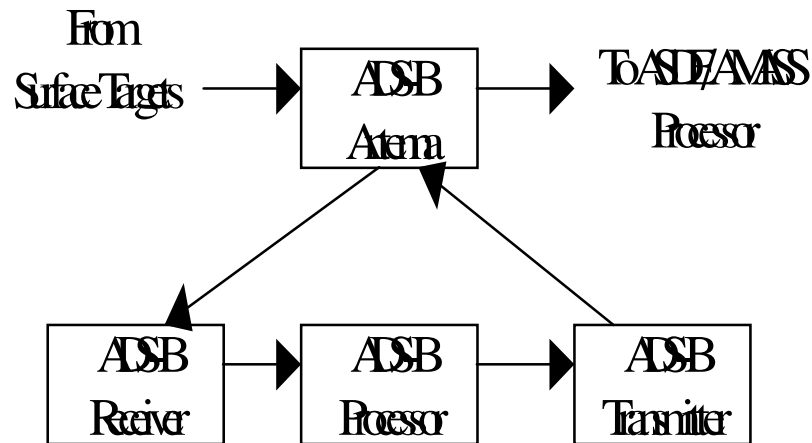
As operating time increases, the population moves toward states with no operating receivers and—because the repair time for transmitters is rapid—toward states with two operating transmitters.

ASSIST LISTINGS

The remainder of this appendix contains ASSIST code listings for the systems analyzed. These listings can be copied, converted to text files, renamed with *".ast"* extensions, and run.

GPS

```
(* Needed for the .mod file *)
LIST = 3;

(* Number of satellites – varies from 4 to 13 *)
INPUT REDUNDANT;  (* command line entry: 4 to 13 *)
(* REDUNDANT = 4; *) (* alternate hard coded entry: 4 to 13 *)
```

```
(* Number of failure states *)
STATES = 2;
(* Failure Rates *)
FAIL1 = 4.51E-3;
FAIL2 = 4.39E-4;
(* Recovery rates *)
RECOVER1 = 1.968;
RECOVER2 = 2.64E-2;
(* 1 means the repairs are done in parallel, 0 means in series *)
PARALLEL = 0;

(* Starting Info *)
SPACE = (WORKING: 0..REDUNDANT, ITEM : ARRAY[1..STATES] OF 0..REDUNDANT);
START = (REDUNDANT, STATES OF 0);

(* Set up the failure rates *)
IF (WORKING > 0) THEN
    FOR I = 1,STATES
        TRANTO WORKING = WORKING - 1, ITEM[I] = ITEM[I] + 1 BY WORKING * FAIL^I;
    ENDFOR;
ENDIF;

FOR I = 1,STATES
    IF (ITEM[I] > 0) THEN
        IF (PARALLEL = 1) THEN
            TRANTO WORKING = WORKING + 1, ITEM[I] = ITEM[I] - 1 BY
ITEM[I]*RECOVER^I;
        ELSE
            TRANTO WORKING = WORKING + 1, ITEM[I] = ITEM[I] - 1 BY RECOVER^I;
        ENDIF;
    ENDIF;
ENDFOR;
```

This version of GPS requires a command line entry for the number of satellites.
STEM automatically provides a prompt.

LAAS Ground Reference Station

```
(* ASSIST Input File to generate Sure/Stem/Paws input file *)
(* LAAS RANGE SENSOR Input File  *)
(*  ****** NO DEATH STATES ***** *)
(* THIS VERSION ASSUMES 100% FAILURE COVERAGE BY DIAGNOSTICS BASED *)
(*  ON THE CENTRAL PROCESSOR'S ABILITY TO DOUBLE CHECK RS DATA *)
(* THIS VERSION INCLUDES REPAIRS BECAUSE THE EQUIPMENT IS GROUND-BASED *)
(* This version is built using the code from previous WAAS and ADS_B models. *)
(* Included are: GPS antennas, GPS receivers, WAAS/LAAS processor, *)
(*         ADS_B processors, ADS_B modulator and transmitter, ADS_B antenna *)
(* The result is for a single Range Sensor *)
(* Last edit 11/16/00 *)

(* Number of Redundant Components of Each Type *)

n_ant = 1;  (* RS/GPS Antennas *)
n_rx = 1;  (* RS/GPS Receivers *)
n_proc = 2; (* RS/GPS Processors *)
n_procA = 2;  (* RS/ADS-B-type Processors *)
n_txA =  1;  (* RS/ADS_B-type Modulator and Transmitter *)
n_antA = 1;  (* RS/ADS_B-type Antenna *)

(* Failure Rates per hours *)
l_ant = 1.e-5;  (* RS/GPS Antennas *)
l_rx =  1.e-4;  (* RS/GPS Receivers *)
l_proc = 5.e-4;  (* RS/GPS Processor *)
l_procA = 5.0e-4;  (* RS/ADS_B-type Processors *)
l_txA =  1.0e-4;  (* RS/ADS_B-type Modulator and Transmitter *)
l_antA = 1.0e-5;  (* RS/ADS_B-type Antenna *)
```

```
            (* Repair Rates per hour *)

            r_ant = 0.2;  (* RS/GPS Antennas - 5 hours *)
            r_rx =  0.5;  (* RS/GPS Receivers - 2 hours *)
            r_proc = 0.2;  (* RS/GPS Processor - 5 hours *)
            r_procA = 0.2;  (* RS/ADS_B-type Processors - 5 hours *)
            r_txA =  0.5;  (* RS/ADS_B-type Modulator and Transmitter - 2 hours *)
            r_antA = 0.2;  (* RS/ADS_B-type Antenna - 5 hours *)


            (* Other Parameters *)
            LIST = 3;          (* Needed for the .mod file *)
            n_modes = 1;          (* Number of system failure modes which
                           will be differentiated in model *)


            space = (m_ant: 0..n_ant,    (* Number of on-line Antennas *)
                 m_rx: 0..n_rx,    (* Number of on-line GPS Receivers *)
                 m_proc: 0..n_proc,  (* Number of on-line LAAS Processors *)
                 m_procA: 0..n_procA, (* Number of on-line ASD-B Processors *)
                 m_txA: 0..n_txA,   (* Number of on-line Modulator and Transmitter channels *)
                 m_antA: 0..n_antA,  (* Number of on-line Antennae *)
                 f_mode: 0..n_modes); (* Flag indicating system failure mode
                           0 = operational state,
                           1 = failed safe *)

            start = (n_ant, n_rx, n_proc, n_procA, n_txA, n_antA, 0 );


            (* Including the deathif statements will aggregate each trapping state into
              one state *)
            (*DEATHIF f_mode = 1; *)(* failed safe *)

            (* Set up event transitions *)

            (* FAILURES *)
            (* Failure of RS/GPS Antenna *)
            if (m_ant >= 2) tranto m_ant = m_ant - 1 by m_ant*l_ant;
            if (m_ant = 1) tranto m_ant = m_ant - 1 , f_mode = 1 by m_ant*l_ant;

            (* Failure of RS/GPS Receiver *)
            if (m_rx >= 2) tranto m_rx = m_rx - 1 by m_rx*l_rx;
            if (m_rx = 1)  tranto m_rx = m_rx - 1, f_mode = 1 by m_rx*l_rx;

            (* Failure of RS/GPS Processor *)
            if (m_proc >= 2) tranto m_proc = m_proc - 1 by m_proc*l_proc;
            if (m_proc = 1) tranto m_proc = m_proc - 1, f_mode = 1 by m_proc*l_proc;

            (* Failure of RS/ADS-B Processor *)
            if (m_procA >= 2) tranto m_procA = m_procA - 1 by m_procA*l_procA;
            if (m_procA = 1) tranto m_procA = m_procA - 1, f_mode = 1 by m_procA*l_procA;

            (* Failure of RS/ADS_B Modulator and Transmitter channel *)
            if (m_txA >= 2) tranto m_txA = m_txA - 1 by m_txA*l_txA;
            if (m_txA = 1) tranto m_txA = m_txA - 1, f_mode = 1 by m_txA*l_txA;

            (* Failure of RS/ADS_B Antenna *)
            if (m_antA >=2) tranto m_antA = m_antA - 1 by m_antA*l_antA;
            if (m_antA = 1) tranto m_antA = m_antA - 1, f_mode = 1 by m_antA*l_antA;

            (* REPAIRS *)
            (* Repair of RS/GPS Antenna *)
            if (m_ant < n_ant) and (m_ant > 0) tranto m_ant = m_ant + 1 by (n_ant-
            m_ant)*r_ant;
            if (m_ant = 0) then
              if (m_rx<>0) and (m_proc<>0) and (m_procA<>0) and (m_txA<>0) and (m_antA<>0)
            then
                 tranto m_ant= m_ant + 1, f_mode = 0 by n_ant*r_ant;
              else tranto m_ant= m_ant + 1 by n_ant*r_ant;
              endif;
            endif;
```

```
                    (* Repair of RS/GPS Receiver *)
                    if (m_rx < n_rx) and (m_rx > 0) tranto m_rx = m_rx + 1 by (n_rx-m_rx)*r_rx;
                    if (m_rx = 0) then
                      if (m_ant<>0) and (m_proc<>0) and (m_procA<>0) and (m_txA<>0) and (m_antA<>0)
                    then
                          tranto m_rx= m_rx + 1, f_mode = 0 by n_rx*r_rx;
                      else tranto m_rx= m_rx + 1 by n_rx*r_rx;
                      endif;
                    endif;

                    (* Repair of RS/GPS Processor *)
                    if (m_proc < n_proc) and (m_proc > 0) tranto m_proc = m_proc + 1 by (n_proc-
                    m_proc)*r_proc;
                    if (m_proc = 0) then
                      if (m_rx<>0) and (m_ant<>0) and (m_procA<>0) and (m_txA<>0) and (m_antA<>0) then
                          tranto m_proc= m_proc + 1, f_mode = 0 by n_proc*r_proc;
                      else tranto m_proc= m_proc + 1 by n_proc*r_proc;
                      endif;
                    endif;

                    (* Repair of RS/ADS_B Processor *)
                    if (m_procA < n_procA) and (m_procA > 0) tranto m_procA = m_procA + 1 by (n_procA-
                    m_procA)*r_procA;
                    if (m_procA = 0) then
                      if (m_proc<>0) and (m_rx<>0) and (m_ant<>0) and (m_txA<>0) and (m_antA<>0) then
                          tranto m_procA= m_procA + 1, f_mode = 0 by n_procA*r_procA;
                      else tranto m_procA= m_procA + 1 by n_procA*r_procA;
                      endif;
                    endif;

                    (* Repair of RS/ADS_B Transmitter Modulator *)
                    if (m_txA < n_txA) and (m_txA > 0) tranto m_txA = m_txA + 1 by (n_txA-
                    m_txA)*r_txA;
                    if (m_txA = 0) then
                      if (m_procA<>0) and (m_proc<>0) and (m_rx<>0) and (m_ant<>0) and (m_antA<>0)
                    then
                          tranto m_txA= m_txA + 1, f_mode = 0 by n_txA*r_txA;
                      else tranto m_txA= m_txA + 1 by n_txA*r_txA;
                      endif;
                    endif;

                    (* Repair of RS/ADS_B Antenna *)
                    if (m_antA < n_antA) and (m_antA > 0) tranto m_antA = m_antA + 1 by (n_antA-
                    m_antA)*r_antA;
                    if (m_antA = 0) then
                      if (m_procA<>0) and (m_proc<>0) and (m_rx<>0) and (m_ant<>0) and (m_txA<>0) then
                          tranto m_antA= m_antA + 1, f_mode = 0 by n_antA*r_antA;
                      else tranto m_antA= m_antA + 1 by n_antA*r_antA;
                      endif;
                    endif;
```

LAAS Ground Facility

```
                    (* LAAS Local Ground Facility/Station ASSIST model *)
                    (* This model uses the WAAS processor model *)
                    (* There is only one LGF/LGS per airport *)

                    LIST = 3;                    (* Needed for the .mod file *)

                    (* Numbers of each component *)
                    N_COMPUTER = 3;              (* Number of master computers *)

                    (* Failure rates -- per day *)
                    F_COMP_HW  = 1E-3;     (* Master computer hardware failure, 1 in 1000 days *)
                    F_COMP_OS  = 1E-2;     (* Master computer operating system failure, 1 in 100 days
                    *)
```

```
        F_CORR_SW  = 0.03168;       (* Correction software failure, 5.5E-5 / 150 sec spec *)
        F_INTE_SW  = 0.03168;       (* Integrity software failure, 5.5E-5 / 150 sec spec *)

        (* Recovery rates -- per day *)
        R_COMP_HW  = 4;             (* Master computer hardware replaced, 6 hrs *)
        R_COMP_OS  = 144;           (* Master computer operating system reboot, 10 min spec
        *)
        R_CORR_SW  = 576;           (* Correction software recalculation, 150 sec *)
        R_INTE_SW  = 576;           (* Integrity software recalculation, 150 sec *)

        (* Coverage rates -- per day *)
        CR_CORR_SW = 2.304E-5;      (* Correction software coverage rate, 4E-8 / 150 sec *)
        CR_INTE_SW = 2.304E-5;      (* Integrity software coverage rate, 4E-8 / 150 sec spec
        *)

        (* Coverage probabilities *)
        C_CORR_SW  = 1 - CR_CORR_SW / F_CORR_SW;
        C_INTE_SW  = 1 - CR_INTE_SW / F_INTE_SW;

        (* Abbreviations for state definition vector *)
        N = N_COMPUTER;

        (* State space definition:              *)
        (* COM: # operational computers         *)
        (* FHW: # computers with hardware failures      *)
        (* FOS: # computers with operating system failures *)
        (* CSW: correction software: 0 nominal, 1 detected failure, 2 undetected failure
        *)
        (* ISW: integrity software: 0 nominal, 1 detected failure, 2 undetected failure *)

        SPACE = (COM: 0..N, FHW: 0..N, FOS: 0..N, CSW: 0..2, ISW: 0..2);

        (* Starting Info *)
        START = (N, 0, 0, 0, 0);

        (* Set up failure rates *)
        IF (COM > 0) THEN
                TRANTO COM = COM - 1, FHW = FHW + 1 BY COM * F_COMP_HW;
                TRANTO COM = COM - 1, FOS = FOS + 1 BY COM * F_COMP_OS;
        ENDIF;

        IF (CSW = 0) THEN
                TRANTO CSW = 1 BY   C_CORR_SW * F_CORR_SW;
                TRANTO CSW = 2 BY (1 - C_CORR_SW) * F_CORR_SW;
        ENDIF;
        IF (ISW = 0) THEN
                TRANTO ISW = 1, BY   C_INTE_SW * F_INTE_SW;
                TRANTO ISW = 2, BY (1 - C_INTE_SW) * F_INTE_SW;
        ENDIF;

        (* Set up recovery rates *)
        (* recovery from back-up operations *)
        IF (FHW > 0) TRANTO COM = COM + 1, FHW = FHW - 1 BY FHW * R_COMP_HW;
        IF (FOS > 0) TRANTO COM = COM + 1, FOS = FOS - 1 BY FOS * R_COMP_OS;

        (* recovery from non-operational to operational *)
        IF (CSW = 1) TRANTO CSW = 0 BY R_CORR_SW;
        IF (ISW = 1) TRANTO ISW = 0 BY R_INTE_SW;
```

LAAS Ground Facility Receiver/Transmitter

```
        (* LAAS Local Ground Facility Transmitter/Receiver/Antenna ASSIST model *)
        (* This model uses ADS_B receiver, modulator/transmitter, and antenna parameters,
        *)
        (* but assumes no uncovered hardware failures *)
        (* Repair capability is added to the receiver, transmitter, and antenna *)
        (* There is only one LGF/LGS per airport *)
```

```
        LIST = 3;                    (* Needed for the .mod file *)

        (* Numbers of each component *)
        n_tx =  2;   (* Modulator and Transmitter *)
        n_rx =  2;   (* Receiver and Demodulator *)
        n_ant = 2;   (* Antenna, n_ant *)

        (* Failure rates -- per hour *)
        l_tx      = 1e-4;        (* Modulator and Transmitter *)
        l_rx      = 1e-4;        (* Receiver and Demodulator *)
        l_ant     = 1e-5;        (* Antenna *)


        (* Recovery rates - per hour *)
        R_tx      = 0.25;        (* Modulator and Transmitter repair, 4 hrs *)
        R_rx      = 0.25;        (* Receiver and Demodulator repair, 4 hrs *)
        R_ant     = 0.1;         (* Antenna Repair 10 hrs*)

        (* State space definition:          *)
        (* m_tx # Number of on-line Modulator and Transmitter channels *)
        (* m_rx # Number of on-line Receiver and Demodulator channels *)
        (* m_ant # Number of on-line Antennae *)

        SPACE = (m_tx: 0..n_tx, m_rx: 0..n_rx, m_ant: 0.. n_ant);

        (* Starting Info *)
        START = (n_tx, n_rx, n_ant);

        (* Set up failure rates *)

        (* Failure of Modulator and Transmitter channel *)
        if (m_tx > 0) tranto m_tx = m_tx - 1, by m_tx* l_tx;

        (* Failure of Receiver and Demodulator channel *)
        if (m_rx > 0) tranto m_rx = m_rx - 1, by m_rx*l_rx;

        (* Failure of Antenna *)
        if (m_ant > 0) tranto m_ant = m_ant - 1, by m_ant*l_ant;


        (* Set up recovery rates *)

        (* recovery from back-up operations *)
        IF (m_tx < n_tx) TRANTO m_tx = m_tx + 1, BY (n_tx-m_tx) * R_tx;
        IF (m_rx < n_rx) TRANTO m_rx = m_rx + 1, BY (n_rx-m_rx) * R_rx;
        IF (m_ant < n_ant) TRANTO m_ant = m_ant + 1, BY (n_ant-m_ant) * R_ant;
```

Airborne GPS Receiver

```
        (* ASSIST Input File to Generate    *)
        (* GPS Airborne Receivers/Antennas *)
        (* Based on Kostiuk & Shapiro WAAS GPS Receiver SURE Input File  *)
        (* WAAS/LAAS processors are included but displays are included with SVS *)
        (* Note there is no recovery for airborne systems so DEATHIF may be useful *)

        (* Number of Redundant Components of Each Type *)
        n_ant = 2;  (* GPS Antennas *)
        n_rx = 3;   (* GPS Receivers *)
        n_proc = 2; (* LAAS/WAAS Processors *)

        (* Failure Rates per hour *)
        l_ant = 1.e-5;  (* GPS Antennas *)
        l_rx =  5.e-5;  (* GPS Receivers *)
        l_proc = 5.e-5; (* LAAS/WAAS Processor *)

        (* Coverage Probabilities *)
        c_ant_2 = 1.00;  (* GPS Antennas, two on-line *)
        c_ant_1 = 1.00;  (* GPS Antennas, one on-line *)
```

```
                c_rx_2  =  0.99;  (* GPS Receivers, two on-line *)
                c_rx_1  =  0.95;  (* GPS Receivers, one on-line *)
                c_proc_2 = 0.99;  (* LAAS/WAAS Processors, two on-line *)
                c_proc_1 = 0.95;  (* LAAS/WAAS Processors, one on-line *)

                (* Other Parameters *)

                LIST = 3;         (* Needed for the .mod file *)
                n_modes = 2;           (* Number of system failure modes which
                               will be differentiated in model *)

                space = (m_ant: 0..n_ant,   (* Number of on-line Antennas *)
                     m_rx: 0..n_rx,     (* Number of on-line GPS Receivers *)
                     m_proc: 0..n_proc,  (* Number of on-line WAAS Processors *)
                     f_mode: 0..n_modes); (* Flag indicating system failure mode
                                   0 = operational state,
                                   1 = failed safe,
                                   2 = failed uncovered *)


                start = (n_ant, n_rx, n_proc, 0);

                (* Including the deathif statements will aggregate each trapping state into
                  one of two states *)

                (* deathif f_mode = 1;*)
                (* deathif f_mode = 2;*)


                (* Set up event transitions *)

                (* Failure of Antenna *)

                if (m_ant >= 3) tranto m_ant = m_ant - 1 by m_ant*l_ant;
                if (m_ant = 2) then
                    tranto m_ant = m_ant - 1 by m_ant*c_ant_2*l_ant;
                    tranto m_ant = m_ant - 1, f_mode = 2 by m_ant*(1 - c_ant_2)*l_ant;
                endif;
                if (m_ant = 1) then
                    tranto m_ant = m_ant - 1, f_mode = 1 by m_ant*c_ant_1*l_ant;
                    tranto m_ant = m_ant - 1, f_mode = 2 by m_ant*(1 - c_ant_1)*l_ant;
                endif;

                (* Failure of GPS Receiver *)

                if (m_rx >= 3) tranto m_rx = m_rx - 1 by m_rx*l_rx;
                if (m_rx = 2) then
                    tranto m_rx = m_rx - 1 by m_rx*c_rx_2*l_rx;
                    tranto m_rx = m_rx - 1, f_mode = 2 by m_rx*(1 - c_rx_2)*l_rx;
                endif;
                if (m_rx = 1) then
                    tranto m_rx = m_rx - 1, f_mode = 1 by m_rx*c_rx_1*l_rx;
                    tranto m_rx = m_rx - 1, f_mode = 2 by m_rx*(1 - c_rx_1)*l_rx;
                endif;

                (* Failure of LAAS/WAAS Processor *)

                if (m_proc >= 3) tranto m_proc = m_proc - 1 by m_proc*l_proc;
                if (m_proc = 2) then
                    tranto m_proc = m_proc - 1 by m_proc*c_proc_2*l_proc;
                    tranto m_proc = m_proc - 1, f_mode = 2 by m_proc*(1 - c_proc_2)*l_proc;
                endif;
                if (m_proc = 1) then
                    tranto m_proc = m_proc - 1, f_mode = 1 by m_proc*c_proc_1*l_proc;
                    tranto m_proc = m_proc - 1, f_mode = 2 by m_proc*(1 - c_proc_1)*l_proc;
                endif;
```

SVS Processor and Display

```
(* ASSIST Input File to Generate *)
(* SVS Input File      *)
(* Includes only the processor and display *)
(* The GPS Receiver, and INS are covered elswhere *)
(* There is no recovery for airborne equipment *)

(* Number of Redundant Components of Each Type *)

n_proc = 2;  (* SVS Processors *)
n_dis = 2;  (* SVS Displays *)

(* Failure Rates *)

l_proc = 5.0e-5;  (* SVS Processors *)
l_dis = 1.0e-4;  (* SVS Displays *)

(* Coverage Probabilities *)

c_proc_2 = 0.99;  (* SVS Processors, two on-line *)
c_proc_1 = 0.95;  (* SVS Processors, one on-line *)
c_dis_2 = 0.999; (* SVS Displays, two on-line *)
c_dis_1 = 0.99;  (* SVS Displays, one on-line *)


(* Other Parameters *)

LIST = 3;         (* Needed for the .mod file *)
n_modes = 3;          (* Number of system failure modes which
                 will be differentiated in model *)

space = (m_proc: 0..n_proc,  (* Number of on-line SVS Processors *)
     m_dis: 0..n_dis,   (* Number of on-line SVS Displays *)
     f_mode: 0..n_modes); (* Flag indicating system failure mode
                    0 = operational state,
                    1 = failed safe non-operational,
                    2 = failed uncovered still operational
                    3 = failed uncovered non-operational *)

start = (n_proc, n_dis, 0);

(* Including the deathif statements will aggregate each trapping state into
  one of two states *)

(* mapping code bombs on deathif states *)
(* comment out deathif states until mapping code upgraded *)
(* deathif f_mode = 1; *)
(* deathif f_mode = 2; *)
(* deathif f_mode = 3; *)

(* Set up event transitions *)

(* Failure of SVS Processor *)

if (m_proc >= 3) tranto m_proc = m_proc - 1 by m_proc*l_proc;
if (m_proc = 2) then
    tranto m_proc = m_proc - 1 by m_proc*c_proc_2*l_proc;
    tranto m_proc = m_proc - 1, f_mode = 2 by m_proc*(1 - c_proc_2)*l_proc;
endif;
if (m_proc = 1) then
    tranto m_proc = m_proc - 1, f_mode = 1 by m_proc*c_proc_1*l_proc;
    tranto m_proc = m_proc - 1, f_mode = 3 by m_proc*(1 - c_proc_1)*l_proc;
endif;

(* Failure of SVS Display *)

if (m_dis >= 3) tranto m_dis = m_dis - 1 by m_dis*l_dis;
if (m_dis = 2) then
    tranto m_dis = m_dis - 1 by m_dis*c_dis_2*l_dis;
```

```
    tranto m_dis = m_dis - 1, f_mode = 2 by m_dis*(1 - c_dis_2)*l_dis;
endif;
if (m_dis = 1) then
    tranto m_dis = m_dis - 1, f_mode = 1 by m_dis*c_dis_1*l_dis;
    tranto m_dis = m_dis - 1, f_mode = 3 by m_dis*(1 - c_dis_1)*l_dis;
endif;
```

CDTI

```
(* Cockpit Display of Traffic Information (CDTI) ASSIST model *)
(* This model uses ADS_B receiver and transmitter parameters, *)
(* but assumes no uncovered hardware failures *)
(* Repair capability is added to the ground receiver and transmitter *)
(* Use of a common antenna with other equipment is assumed for both ground and air
*)
(* There is only one CDTI per airport *)

LIST = 3;                   (* Needed for the .mod file *)

(* Numbers of each component *)
n_gtx =  2;  (* Modulator and Transmitter *)
n_arx =  2;  (* Receiver and Demodulator *)

(* other parameters *)
n_modes = 1;         (* Number of system failure modes which
                will be differentiated in model *)

(* Failure rates -- per hour *)
l_gtx      = 1e-4;          (* Ground Transmitter *)
l_arx      = 5.0e-5;        (* Airborne Receiver *)

(* Recovery rates - per hour *)
R_gtx      = 0.25;          (* Ground Modulator and Transmitter repair, 4 hrs *)

(* State space definition:           *)
(* m_gtx # Number of on-line Ground Transmitter channels *)
(* m_arx # Number of on-line Airborne Receiver channels *)
(* f_mode # Failure mode: 0 = operational, 1 = failed safe *)

SPACE = (m_gtx: 0..n_gtx, m_arx: 0..n_arx, f_mode: 0..n_modes);

(* Starting Info *)
START = (n_gtx, n_arx, 0);

(* Set up failure rates *)

(* Failure of Ground Transmitter channel *)
if (m_gtx > 1) tranto m_gtx = m_gtx - 1, by m_gtx* l_gtx;
if (m_gtx = 1) tranto m_gtx = m_gtx - 1, f_mode = 1 by m_gtx* l_gtx;

(* Failure of Airborne Receiver channel *)
if (m_arx > 1) tranto m_arx = m_arx - 1, by m_arx* l_arx;
if (m_arx = 1) tranto m_arx = m_arx - 1, f_mode = 1 by m_arx* l_arx;

(* Set up recovery rates *)
(* Recovery of Ground Transmitter *)
IF (m_gtx < n_gtx) AND (m_gtx > 0) TRANTO m_gtx = m_gtx + 1, BY (n_gtx-m_gtx) *
R_gtx;
IF (m_gtx = 0) THEN
  IF (m_arx <> 0) THEN
    TRANTO m_gtx = m_gtx + 1, f_mode = 0 BY R_gtx;
  ELSE
    TRANTO m_gtx = m_gtx + 1 BY R_gtx;
  ENDIF;
ENDIF;
```

ADS-B

```
(* ASSIST Input File to Generate *)
(* ADS-B SURE Input File      *)


(* Number of Redundant Components of Each Type *)

n_ins = 2;  (* INS *)
n_proc = 2;  (* ADS-B Processors *)
n_dis = 2;  (* ADS-B Displays *)
n_tx =  1;  (* Modulator and Transmitter, n_tx <= 1*)
n_rx =  1;  (* Receiver and Demodulator, n_rx <= 1 *)
n_ant = 1;  (* Antenna, n_ant <= 1 *)


(* Failure Rates *)

l_ins = 2.0e-4;  (* INS *)
l_proc = 5.0e-5;  (* ADS-B Processors *)
l_dis = 1.0e-4;  (* ADS-B Displays *)
l_tx =  5.0e-5;  (* Modulator and Transmitter *)
l_rx =  5.0e-5;  (* Receiver and Demodulator *)
l_ant = 1.0e-5;  (* Antenna *)


(* Coverage Probabilities *)

c_ins_2 = 0.999; (* INS, two on-line *)
c_ins_1 = 0.99;  (* INS, one on-line *)
c_proc_2 = 0.99;  (* ADS-B Processors, two on-line *)
c_proc_1 = 0.95;  (* ADS-B Processors, one on-line *)
c_dis_2 = 0.999; (* ADS-B Displays, two on-line *)
c_dis_1 = 0.99;  (* ADS-B Displays, one on-line *)
c_tx =   0.99;  (* Modulator and Transmitter *)
c_rx =   0.99;  (* Receiver and Demodulator *)
c_ant =  1.00;  (* Antenna *)


(* Other Parameters *)

LIST = 3;          (* Needed for the .mod file *)
n_modes = 3;          (* Number of system failure modes which
                 will be differentiated in model *)


space = (m_ins: 0..n_ins,    (* Number of on-line INSs *)
     m_proc: 0..n_proc,  (* Number of on-line ASD-B Processors *)
     m_dis: 0..n_dis,    (* Number of on-line ASD-B Displays *)
     m_tx: 0..n_tx,     (* Number of on-line Modulator and Transmitter channels *)
     m_rx: 0..n_rx,     (* Number of on-line Receiver and Demodulator channels *)
     m_ant: 0..n_ant,    (* Number of on-line Antennae *)
     f_mode: 0..n_modes); (* Flag indicating system failure mode
                0 = operational state,
                1 = failed safe non-operational,
                2 = failed uncovered still operational
                3 = failed uncovered non-operational *)


start = (n_ins, n_proc, n_dis, n_tx, n_rx, n_ant, 0);

(* Including the deathif statements will aggregate each trapping state into
  one of two states *)

(* mapping code bombs on deathif states *)
(* comment out deathif states until mapping code upgraded *)
(* deathif f_mode = 1;*)
(* deathif f_mode = 2;*)
(* deathif f_mode = 3;*)
```

```
(* Set up event transitions *)

(* Failure of INS *)

if (m_ins >= 3) tranto m_ins = m_ins - 1 by m_ins*l_ins;
if (m_ins = 2) then
    tranto m_ins = m_ins - 1 by m_ins*c_ins_2*l_ins;
    tranto m_ins = m_ins - 1, f_mode = 2 by m_ins*(1 - c_ins_2)*l_ins;
endif;
if (m_ins = 1) then
    tranto m_ins = m_ins - 1, f_mode = 1 by m_ins*c_ins_1*l_ins;
    tranto m_ins = m_ins - 1, f_mode = 3 by m_ins*(1 - c_ins_1)*l_ins;
endif;

(* Failure of ADS-B Processor *)

if (m_proc >= 3) tranto m_proc = m_proc - 1 by m_proc*l_proc;
if (m_proc = 2) then
    tranto m_proc = m_proc - 1 by m_proc*c_proc_2*l_proc;
    tranto m_proc = m_proc - 1, f_mode = 2 by m_proc*(1 - c_proc_2)*l_proc;
endif;
if (m_proc = 1) then
    tranto m_proc = m_proc - 1, f_mode = 1 by m_proc*c_proc_1*l_proc;
    tranto m_proc = m_proc - 1, f_mode = 3 by m_proc*(1 - c_proc_1)*l_proc;
endif;

(* Failure of ADS-B Display *)

if (m_dis >= 3) tranto m_dis = m_dis - 1 by m_dis*l_dis;
if (m_dis = 2) then
    tranto m_dis = m_dis - 1 by m_dis*c_dis_2*l_dis;
    tranto m_dis = m_dis - 1, f_mode = 2 by m_dis*(1 - c_dis_2)*l_dis;
endif;
if (m_dis = 1) then
    tranto m_dis = m_dis - 1, f_mode = 1 by m_dis*c_dis_1*l_dis;
    tranto m_dis = m_dis - 1, f_mode = 3 by m_dis*(1 - c_dis_1)*l_dis;
endif;

(* Failure of Modulator and Transmitter channel *)

if (m_tx = 1) then
    tranto m_tx = m_tx - 1, f_mode = 1 by m_tx*c_tx*l_tx;
    tranto m_tx = m_tx - 1, f_mode = 3 by m_tx*(1 - c_tx)*l_tx;
endif;

(* Failure of Receiver and Demodulator channel *)

if (m_rx = 1) then
    tranto m_rx = m_rx - 1, f_mode = 1 by m_rx*c_rx*l_rx;
    tranto m_rx = m_rx - 1, f_mode = 3 by m_rx*(1 - c_rx)*l_rx;
endif;

(* Failure of Antenna *)

if (m_ant = 1) then
    tranto m_ant = m_ant - 1, f_mode = 1 by m_ant*c_ant*l_ant;
    tranto m_ant = m_ant - 1, f_mode = 3 by m_ant*(1 - c_ant)*l_ant;
endif;
```

FMS

```
(* ASSIST Input File to Generate *)
(* FMS Input File      *)
(* Roughly Based on Smiths's Industries FMS in Janes's Avionics *)
(* The GPS receiver is handled as a sparate system *)
(* The dual INS is also included in ADS-B so its failure rate here is 0 *)
```

```
(* Similarly, the display is included with the SVS system *)
(* No recovery of airborne equipment *)

(* Number of Components of Each Type *)

n_ins = 2;  (* INS *)
n_proc = 2;  (* FMS Processors *)
n_navrad = 2;  (* Navigation Radios *)


(* Failure Rates per hour *)

l_ins = 0.0;     (* INS 0 to prevent double counting with ADS-B *)
l_proc = 1.0e-5;  (* FMS Processors *)
l_navrad = 5.0e-5; (* Navigation Radio *)

(* Coverage Probabilities *)

c_ins_2 = 0.999; (* INS, two on-line *)
c_ins_1 = 0.99;  (* INS, one on-line *)
c_proc_2 = 0.99;  (* FMS Processors, two on-line *)
c_proc_1 = 0.95;  (* FMS Processors, one on-line *)
c_navrad_1 = 0.99; (* Navigation radio two on-line *)
c_navrad_2 = 1.0; (* Navigation radio one on-line - no hidden failure for lost
radio*)

(* Other Parameters *)

LIST = 3;          (* Needed for the .mod file *)
n_modes = 2;          (* Number of system failure modes which
                 will be differentiated in model *)

space = (m_ins: 0..n_ins,     (* Number of on-line INSs *)
     m_proc: 0..n_proc,    (* Number of on-line FMS Processors *)
      m_navrad: 0..n_navrad, (* Number of on-line Navigation radios *)
      f_mode: 0..n_modes);  (* Flag indicating system failure mode
                  0 = operational state,
                  1 = failed safe,
                  2 = failed uncovered *)


start = (n_ins, n_proc, n_navrad, 0);

(* Including the deathif statements will aggregate each trapping state into
  one of two states *)

(* mapping code bombs on deathif states *)
(* comment out deathif states until mapping code upgraded *)
(* deathif f_mode = 1; *)
(* deathif f_mode = 2; *)

(* Set up event transitions *)

(* Failure of INS *)

if (m_ins >= 3) tranto m_ins = m_ins - 1 by m_ins*l_ins;
if (m_ins = 2) then
    tranto m_ins = m_ins - 1 by m_ins*c_ins_2*l_ins;
    tranto m_ins = m_ins - 1, f_mode = 2 by m_ins*(1 - c_ins_2)*l_ins;
endif;
if (m_ins = 1) then
    tranto m_ins = m_ins - 1, f_mode = 1 by m_ins*c_ins_1*l_ins;
    tranto m_ins = m_ins - 1, f_mode = 2 by m_ins*(1 - c_ins_1)*l_ins;
endif;

(* Failure of FMS Processor *)

if (m_proc >= 3) tranto m_proc = m_proc - 1 by m_proc*l_proc;
if (m_proc = 2) then
    tranto m_proc = m_proc - 1 by m_proc*c_proc_2*l_proc;
    tranto m_proc = m_proc - 1, f_mode = 2 by m_proc*(1 - c_proc_2)*l_proc;
```

```
                endif;
                if (m_proc = 1) then
                    tranto m_proc = m_proc - 1, f_mode = 1 by m_proc*c_proc_1*l_proc;
                    tranto m_proc = m_proc - 1, f_mode = 2 by m_proc*(1 - c_proc_1)*l_proc;
                endif;

                (* Failure of Navigation Radio *)

                if (m_navrad >= 3) tranto m_navrad = m_navrad - 1 by m_navrad*l_navrad;
                if (m_navrad = 2) then
                    tranto m_navrad = m_navrad - 1 by m_navrad*c_navrad_2*l_navrad;
                    tranto m_navrad = m_navrad - 1, f_mode = 2 by m_navrad*(1 -
                c_navrad_2)*l_navrad;
                endif;
                if (m_navrad = 1) then
                    tranto m_navrad = m_navrad - 1, f_mode = 1 by m_navrad*c_navrad_1*l_navrad;
                    tranto m_navrad = m_navrad - 1, f_mode = 2 by m_navrad*(1 -
                c_navrad_1)*l_navrad;
                endif;
```

Autopilot

```
                (* ASSIST Input File to Generate *)
                (* Autopilot Input File      *)
                (* Roughly based on 747 SPZ-1 for components *)
                (* No recovery for airborne equipment *)

                (* Number of Redundant Components of Each Type *)

                n_rcomp = 3; (* Roll computer *)
                n_pcomp = 3; (* Pitch computer *)
                n_astab = 1; (* Autostabilizer *)
                n_lrcu = 2; (* Landing Roll-out Control Unit *)
                n_sens = 3; (* Sensor Suite *)

                (* Failure Rates per hour *)

                l_rcomp = 5e-5; (* Roll computer *)
                l_pcomp = 5e-5; (* Pitch computer *)
                l_astab = 1e-4; (* Autostabilizer *)
                l_lrcu = 1e-3; (* Landing Roll-out Control Unit *)
                l_sens = 1e-3; (* Sensor Suite *)

                (* Coverage Probabilities *)

                c_rcomp_2 = 0.99; (* Roll computer, two on-line *)
                c_rcomp_1 = 0.95;  (* Roll computer, one on-line *)
                c_pcomp_2 = 0.99;  (* Pitch Computer, two on-line *)
                c_pcomp_1 = 0.95;  (* Pitch Computer, one on-line *)
                c_astab =  0.99; (* AutoStabilizer *)
                c_lrcu_2 =  0.99;  (* Landing Roll-out Control Unit, two on-line *)
                c_lrcu_1 =  0.95;  (* Landing Roll-out Control Unit, one on-line *)
                c_sens_2 =  0.99;  (* Sensors, two on-line *)
                c_sens_1 =  0.95;  (* Sensors, one on-line *)

                (* Other Parameters *)

                LIST = 3;          (* Needed for the .mod file *)
                n_modes = 3;          (* Number of system failure modes which
                             will be differentiated in model *)


                space = (m_rcomp: 0..n_rcomp,  (* Number of on-line Roll Computers *)
                    m_pcomp: 0..n_pcomp,  (* Number of on-line Pitch Computers *)
                    m_astab: 0..n_astab,  (* Number of on-line AutoStabilizers *)
                    m_lrcu: 0..n_lrcu,    (* Number of on-line Landing and Roll-out Control Units
                *)
                    m_sens: 0..n_sens,    (* Number of on-line Sensor Suites *)
```

```
            f_mode: 0..n_modes);  (* Flag indicating system failure mode
                        0 = operational state,
                        1 = failed safe non-operational,
                        2 = failed uncovered still operational
                        3 = failed uncovered non-operational *)

        start = (n_rcomp, n_pcomp, n_astab, n_lrcu, n_sens, 0);

        (* Set up failure transitions *)

        (* Failure of Roll Computer *)
        if (m_rcomp >= 3) tranto m_rcomp = m_rcomp - 1 by m_rcomp*l_rcomp;
        if (m_rcomp = 2) then
            tranto m_rcomp = m_rcomp - 1 by m_rcomp*c_rcomp_2*l_rcomp;
            tranto m_rcomp = m_rcomp - 1, f_mode = 2 by m_rcomp*(1 - c_rcomp_2)*l_rcomp;
        endif;
        if (m_rcomp = 1) then
            tranto m_rcomp = m_rcomp - 1, f_mode = 1 by m_rcomp*c_rcomp_1*l_rcomp;
            tranto m_rcomp = m_rcomp - 1, f_mode = 3 by m_rcomp*(1 - c_rcomp_1)*l_rcomp;
        endif;

        (* Failure of Pitch Computer *)
        if (m_pcomp >= 3) tranto m_pcomp = m_pcomp - 1 by m_pcomp*l_pcomp;
        if (m_pcomp = 2) then
            tranto m_pcomp = m_pcomp - 1 by m_pcomp*c_pcomp_2*l_pcomp;
            tranto m_pcomp = m_pcomp - 1, f_mode = 2 by m_pcomp*(1 - c_pcomp_2)*l_pcomp;
        endif;
        if (m_pcomp = 1) then
            tranto m_pcomp = m_pcomp - 1, f_mode = 1 by m_pcomp*c_pcomp_1*l_pcomp;
            tranto m_pcomp = m_pcomp - 1, f_mode = 3 by m_pcomp*(1 - c_pcomp_1)*l_pcomp;
        endif;

        (* Failure of Autostabilizer *)
        if (m_astab = 1) then
            tranto m_astab = m_astab - 1, f_mode = 1 by m_astab*c_astab*l_astab;
            tranto m_astab = m_astab - 1, f_mode = 3 by m_astab*(1 - c_astab)*l_astab;
        endif;

        (* Failure of Landing and Roll-out Control Unit *)
        if (m_lrcu >= 3) tranto m_lrcu = m_lrcu - 1 by m_lrcu*l_lrcu;
        if (m_lrcu = 2) then
            tranto m_lrcu = m_lrcu - 1 by m_lrcu*c_lrcu_2*l_lrcu;
            tranto m_lrcu = m_lrcu - 1, f_mode = 2 by m_lrcu*(1 - c_lrcu_2)*l_lrcu;
        endif;
        if (m_lrcu = 1) then
            tranto m_lrcu = m_lrcu - 1, f_mode = 1 by m_lrcu*c_lrcu_1*l_lrcu;
            tranto m_lrcu = m_lrcu - 1, f_mode = 3 by m_lrcu*(1 - c_lrcu_1)*l_lrcu;
        endif;

        (* Failure of Sensor Suite *)
        if (m_sens >= 3) tranto m_sens = m_sens - 1 by m_sens*l_sens;
        if (m_sens = 2) then
            tranto m_sens = m_sens - 1 by m_sens*c_sens_2*l_sens;
            tranto m_sens = m_sens - 1, f_mode = 2 by m_sens*(1 - c_sens_2)*l_sens;
        endif;
        if (m_sens = 1) then
            tranto m_sens = m_sens - 1, f_mode = 1 by m_sens*c_sens_1*l_sens;
            tranto m_sens = m_sens - 1, f_mode = 3 by m_sens*(1 - c_sens_1)*l_sens;
        endif;
```

ASDE Radar

```
        (* ASSIST model for *)
        (* ASDE primary radar*)
        (* based on corrected LMI/Draper/MIT Primary Radar model and reliability numbers
        *)
        (* The ASDE-3 Ku-band and ASDE-X X-band reliabilities may well be different *)
        (* Last edit 11/16/00 *)
        LIST = 3;
```

```
            PRUNE = 0;
            STATES = 1;

            (* Failure Rates *)
            F_PRIM_RAD_ANT = 1/1000; (* Primary radar Antenna *)
            F_PRIM_RAD_TRN = 1/750; (* Primary radar Transmitter *)
            F_PRIM_RAD_RCV = 1/750;     (* Primary radar Receiver *)
            F_PRIM_RAD_PROC = 1/20000; (*Primary radar Processor *)

            (* Repair Rates *)
            R_PRIM_RAD_ANT = 1/4;   (* Primary radar Antenna *)
            R_PRIM_RAD_TRN = 1/2;   (* Primary radar Transmitter *)
            R_PRIM_RAD_RCV = 1/2;   (* Primary radar Receiver *)
            R_PRIM_RAD_PROC = 2;    (* Primary radar Processor *)

            (* SYS_MODE: failure mode: 0 = operational, 1 = failed *)
            (* This is reversed from the LMI/Draper version for consistancy with other compo-
            nent models *)
            (* The recovery error where the state is operational even when some components are
            still failed *)
            (* is corrected in this version - 11/16/00 *)

            SPACE = (SYS_MODE: 0..1, PRIM_RAD_ANT: 0..1, PRIM_RAD_TRN: 0..2, PRIM_RAD_RCV:
            0..2, PRIM_RAD_PROC: 0..2);

            START = (0, 1, 2, 2, 2);

            (* Loss of the single Primary radar antenna is considered loss of the primary ra-
            dar *)
            IF PRIM_RAD_ANT > 0 TRANTO SYS_MODE = 1, PRIM_RAD_ANT = PRIM_RAD_ANT - 1 BY
            F_PRIM_RAD_ANT;
            (* Recover from antenna loss *)
            IF PRIM_RAD_ANT < 1 THEN
               IF (PRIM_RAD_TRN<>0) and (PRIM_RAD_RCV<>0) and (PRIM_RAD_PROC<>0) THEN
                  TRANTO SYS_MODE = 0, PRIM_RAD_ANT = PRIM_RAD_ANT + 1 BY R_PRIM_RAD_ANT;
               ELSE TRANTO PRIM_RAD_ANT = PRIM_RAD_ANT + 1 BY R_PRIM_RAD_ANT;
               ENDIF;
            ENDIF;

            (* Loss of both of the Primary radar transmitters is considered loss of the pri-
            mary radar *)
            IF PRIM_RAD_TRN > 1 TRANTO PRIM_RAD_TRN = PRIM_RAD_TRN - 1 BY F_PRIM_RAD_TRN;
            IF PRIM_RAD_TRN = 1 TRANTO SYS_MODE = 1, PRIM_RAD_TRN = PRIM_RAD_TRN - 1 BY
            F_PRIM_RAD_TRN;
            (* Recover from transmitter loss *)
            IF (PRIM_RAD_TRN>0) and (PRIM_RAD_TRN<2) TRANTO PRIM_RAD_TRN = PRIM_RAD_TRN+1 BY
            R_PRIM_RAD_TRN;
            IF PRIM_RAD_TRN = 0 THEN
               IF (PRIM_RAD_ANT<>0) and (PRIM_RAD_RCV<>0) and (PRIM_RAD_PROC<>0) THEN
                  TRANTO SYS_MODE = 0, PRIM_RAD_TRN = PRIM_RAD_TRN + 1 BY R_PRIM_RAD_TRN;
               ELSE TRANTO PRIM_RAD_TRN = PRIM_RAD_TRN + 1 BY R_PRIM_RAD_TRN;
               ENDIF;
            ENDIF;

            (* Loss of both of the Primary radar receivers is considered loss of the primary
            radar *)
            IF PRIM_RAD_RCV > 1 TRANTO PRIM_RAD_RCV = PRIM_RAD_RCV - 1 BY F_PRIM_RAD_RCV;
            IF PRIM_RAD_RCV = 1 TRANTO SYS_MODE = 1, PRIM_RAD_RCV = PRIM_RAD_RCV - 1 BY
            F_PRIM_RAD_RCV;
            (* Recover from receiver loss *)
            IF (PRIM_RAD_RCV>0) AND (PRIM_RAD_RCV<2) TRANTO PRIM_RAD_RCV = PRIM_RAD_RCV+1 BY
            R_PRIM_RAD_RCV;
            IF PRIM_RAD_RCV = 0 THEN
               IF (PRIM_RAD_ANT<>0) and (PRIM_RAD_TRN<>0) and (PRIM_RAD_PROC<>0) THEN
                  TRANTO SYS_MODE = 0, PRIM_RAD_RCV = PRIM_RAD_RCV + 1 BY R_PRIM_RAD_RCV;
               ELSE TRANTO PRIM_RAD_RCV = PRIM_RAD_RCV + 1 BY R_PRIM_RAD_RCV;
               ENDIF;
            ENDIF;

            (* Loss of both of the Primary radar processors is considered loss of the primary
            radar *)
```

```
          IF PRIM_RAD_PROC > 1 TRANTO PRIM_RAD_PROC = PRIM_RAD_PROC - 1 BY F_PRIM_RAD_PROC;
          IF PRIM_RAD_PROC = 1 TRANTO SYS_MODE = 1, PRIM_RAD_PROC = PRIM_RAD_PROC - 1 BY
          F_PRIM_RAD_PROC;
          (* Recover from processor loss *)
          IF (PRIM_RAD_PROC>0) AND (PRIM_RAD_PROC<2) TRANTO PRIM_RAD_PROC=PRIM_RAD_PROC+1 BY
          R_PRIM_RAD_PROC;
          IF PRIM_RAD_PROC = 0 THEN
             IF (PRIM_RAD_ANT<>0) and (PRIM_RAD_RCV<>0) and (PRIM_RAD_TRN<>0) THEN
               TRANTO SYS_MODE = 0, PRIM_RAD_PROC = PRIM_RAD_PROC + 1 BY R_PRIM_RAD_PROC;
             ELSE TRANTO PRIM_RAD_PROC = PRIM_RAD_PROC + 1 BY R_PRIM_RAD_PROC;
             ENDIF;
          ENDIF;
```

ADS-B Ground Surveillance Stations

```
          (* ASSIST Input File to generate Sure/Stem/Paws input file *)
          (* ADS-B GROUND STATION Input File  *)
          (* THIS VERSION ASSUMES 100% FAILURE COVERAGE BY DIAGNOSTICS BASED *)
          (*  ON THE CDTI PROCESSOR'S ABILITY TO DOUBLE CHECK ADS-B DATA *)
          (* THIS VERSION INCLUDES REPAIRS BECAUSE THE EQUIPMENT IS GROUND-BASED *)
          (* This version is built using the code from previous ADS_B models. *)
          (* Included are: ADS_B processor, ADS-B receiver, ADS_B modulator and *)
          (*  transmitter, ADS_B antenna *)
          (* The result is for a SINGLE ADS-B ground station. Multiple sites will *)
          (*  be needed for full airport coverage, and the probability of N out of *)
          (*  M sites operating is calculated based on a binomial distribution based on*)
          (*  the probability for a single site. *)
          (* Last edit 11/7/00 *)

          (* Number of Redundant Components of Each Type *)

          n_ant = 1;  (* Ground station ADS-B Antenna *)
          n_proc = 1;  (* Ground station ADS-B Processor *)
          n_rx =  1;  (* Ground station ADS-B Receiver *)
          n_tx =  1;  (* Ground station ADS-B Modulator and Transmitter *)

          (* Failure Rates per hours *)
          l_ant = 1.0e-5;  (* Ground station ADS-B Antenna *)
          l_proc = 5.0e-4;  (* Ground station ADS-B Processor *)
          l_rx =  1.0e-4;  (* Ground station ADS-B Receiver *)
          l_tx =  1.0e-4;  (* Ground station ADS-B Modulator and Transmitter *)

          (* Repair Rates per hour *)
          r_ant = 0.2;  (* Ground Station Antenna - 5 hours *)
          r_proc = 0.2;  (* Ground station ADS-B Processor - 5 hours *)
          r_rx =  0.5;  (* Ground station ADS-B Receiver - 2 hours
          r_tx =  0.5;  (* Ground Sattion Modulator and Transmitter - 2 hours *)


          (* Other Parameters *)
          LIST = 3;          (* Needed for the .mod file *)

          space = (opstate: 0..1,    (* 0 is operational, 1 is failed *)
              m_ant: 0..n_ant,   (* Number of on-line Antennas *)
              m_proc: 0..n_proc,  (* Number of on-line Processors *)
              m_rx: 0..n_rx,    (* Number of on-line Receivers *)
              m_tx: 0..n_tx);   (* Number of on-line Modulator and Transmitters *)

          start = (0, n_ant, n_proc, n_rx, n_tx);

          (* Set up event transitions *)

          (* FAILURES *)
          (* Failure of ADS-B Antenna *)
          if (m_ant >= 2) tranto m_ant = m_ant - 1 by m_ant*l_ant;
          if (m_ant = 1) tranto m_ant = m_ant - 1, opstate = 1 by m_ant*l_ant;

          (* Failure of ADS-B Processor *)
```

```
            if (m_proc >= 2) tranto m_proc = m_proc - 1 by m_proc*l_proc;
            if (m_proc = 1) tranto m_proc = m_proc - 1, opstate = 1 by m_proc*l_proc;

            (* Failure of ADS-B Receiver *)
            if (m_rx >= 2) tranto m_rx = m_rx - 1 by m_rx*l_rx;
            if (m_rx = 1) tranto m_rx = m_rx - 1, opstate = 1 by m_rx*l_rx;

            (* Failure of ADS-B Modulator and Transmitter *)
            if (m_tx >= 2) tranto m_tx = m_tx - 1 by m_tx*l_tx;
            if (m_tx = 1) tranto m_tx = m_tx - 1, opstate = 1 by m_tx*l_tx;

            (* REPAIRS *)
            (* Repair of ADS-B Antenna *)
            if (m_ant < n_ant) and (m_ant > 0) tranto m_ant = m_ant + 1 by (n_ant-
            m_ant)*r_ant;
            if (m_ant = 0) then
              if (m_proc<>0) and (m_rx<>0) and (m_tx<>0) then
                tranto m_ant= m_ant + 1, opstate = 1 by n_ant*r_ant;
              else tranto m_ant= m_ant + 1 by n_ant*r_ant;
              endif;
            endif;

            (* Repair of ADS-B Processor *)
            if (m_proc < n_proc) and (m_proc > 0) tranto m_proc = m_proc + 1 by (n_proc-
            m_proc)*r_proc;
            if (m_proc = 0) then
              if (m_ant<>0) and (m_rx<>0) and (m_tx<>0) then
                tranto m_proc= m_proc + 1, opstate = 0 by n_proc*r_proc;
              else tranto m_proc= m_proc + 1 by n_proc*r_proc;
              endif;
            endif;

            (* Repair of ADS-B Receiver *)
            if (m_rx < n_rx) and (m_rx > 0) tranto m_rx = m_rx + 1 by (n_rx-m_rx)*r_rx;
            if (m_rx = 0) then
              if (m_ant<>0) and (m_proc<>0) and (m_tx<>0) then
                tranto m_rx= m_rx + 1, opstate = 0 by n_rx*r_rx;
              else tranto m_rx= m_rx + 1 by n_rx*r_rx;
              endif;
            endif;

            (* Repair of ADS-B Transmitter Modulator *)
            if (m_tx < n_tx) and (m_tx > 0) tranto m_tx = m_tx + 1 by (n_tx-m_tx)*r_tx;
            if (m_tx = 0) then
              if (m_ant<>0) and (m_proc<>0) and (m_rx<>0) then
                tranto m_tx= m_tx + 1, opstate = 0 by n_tx*r_tx;
              else tranto m_tx= m_tx+1 by n_tx*r_tx;
              endif;
            endif;
```

AMASS Processor

```
            (* ASDE radar processor ASSIST model *)
            (* This model is loosely based on the WAAS/LAAS processor model *)
            (* There is only one ASDE per airport *)

            LIST = 3;                   (* Needed for the .mod file *)

            (* Numbers of each component *)
            N_COMPUTER = 3;             (* Number of master computers *)

            (* Failure rates -- per hour *)
            F_COMP_HW  = 5.0E-4;        (* Hardware failure, 2000 hour MTBF *)
            F_COMP_OS  = 1E-3;      (* Software failure, 1000 hour MTBF *)

            (* Recovery rates -- per hour *)
            R_COMP_HW  = 1/4;           (* Computer hardware replaced, 4 hour MTTR *)
            R_COMP_OS  = 1;             (* Software repair, test and reboot, 1 hour *)
```

```
(* Abbreviations for state definition vector *)
N = N_COMPUTER;

(* State space definition:             *)
(* COM: # operational computers          *)
(* FHW: # computers with hardware failures     *)
(* FOS: # computers with operating system failures *)
(* Fail: 0 = operational, 1 = failed  *)

SPACE = (COM: 0..N, FHW: 0..N, FOS: 0..N, Fail: 0..1);

(* Starting Info *)
START = (N, 0, 0, 0);

(* Set up failure rates *)
IF (COM > 1) THEN
        TRANTO COM = COM - 1, FHW = FHW + 1 BY COM * F_COMP_HW;
        TRANTO COM = COM - 1, FOS = FOS + 1 BY COM * F_COMP_OS;
ENDIF;

IF (COM = 1) THEN
         TRANTO COM = COM - 1, Fail = 1, FHW = FHW + 1 BY COM * F_COMP_HW;
        TRANTO COM = COM - 1, Fail = 1, FOS = FOS + 1 BY COM * F_COMP_OS;
ENDIF;

(* Set up recovery rates *)
(* recovery from back-up operations *)
IF (FHW > 0) AND (COM > 0 ) TRANTO COM = COM + 1, FHW = FHW - 1 BY FHW *
R_COMP_HW;
IF (FOS > 0) AND (COM > 0 ) TRANTO COM = COM + 1, FOS = FOS - 1 BY FOS *
R_COMP_OS;
IF (FHW > 0) AND (COM = 0 ) THEN
  IF (FOS <> N) THEN
    TRANTO Fail = 0, COM = COM + 1, FHW = FHW - 1 BY FHW * R_COMP_HW;
  ELSE TRANTO COM = COM + 1, FHW = FHW - 1 BY FHW * R_COMP_HW;
  ENDIF;
ENDIF;

IF (FOS > 0) AND (COM = 0 ) THEN
  IF (FHW <> N) THEN
    TRANTO Fail = 0, COM = COM + 1, FOS = FOS - 1 BY FOS * R_COMP_OS;
  ELSE TRANTO COM = COM + 1, FOS = FOS - 1 BY FOS * R_COMP_OS;
  ENDIF;
ENDIF;
```

Common Avionics: Weather Radar, TCAS, EGPWS

This code is a placeholder for equipment that may be available as adjuncts or
back-ups to SVS. The models are simple and would need to be revised for use in
an analysis.

```
(* ASSIST Input File to Generate    *)
(* WX Radar, TCAS, EGPWS - e.g., Common Avionics*)
(* These systems are assumed useful back-ups to SVS *)
(* Their failure may result in degraded mode operation *)
(* Simple unit model using speculative failure rates  *)
(* Assume 100% failure coverage *)
(* Note there is no recovery for airborne systems so DEATHIF may be useful *)

(* Number of Redundant Components of Each Type *)
n_wx   = 1;  (* Wx Radar *)
n_tcas = 1;  (* TCAS *)
n_egpws = 1;  (* EGPWS *)
```

```
(* Failure Rates per hour *)
l_wx =  1.e-4;  (* Wx Radar *)
l_tcas = 1.e-3;  (* TCAS *)
l_egpws = 1.e-4;  (* EGPWS *)

(* Other Parameters *)
LIST = 3;          (* Needed for the .mod file *)


space = (m_wx: 0..n_wx,     (* Number Wx Radars *)
    m_tcas: 0..n_tcas,   (* Number TCAS *)
    m_egpws: 0..n_egpws); (* Number of EGPWS *)

start = (n_wx, n_tcas, n_egpws);


(* Set up event transitions *)

(* Failure of Wx Radar *)
if (m_wx >= 1) tranto m_wx = m_wx - 1 by m_wx*l_wx;

(* Failure of TCAS *)
if (m_tcas >= 1) tranto m_tcas = m_tcas - 1 by m_tcas*l_tcas;

(* Failure of EGPWS *)
if (m_egpws >= 1) tranto m_egpws = m_egpws - 1 by m_egpws*l_egpws;
```

Appendix C
Miscellaneous References

The appendix lists information that was collected during this study on parameters related to the simulation. Accurate values of aircraft performance and air and ground surveillance parameters are required for valid simulation of the terminal airspace and new technologies. Many of the required parameters are for systems that are still in development—such as LAAS—or for systems for which definitive performance is hard to measure, such as ASDE. The information in this appendix provides data for parameter selection and insight into the quality of the data.

AIRBORNE PERFORMANCE

Russ Bowie, *GPS and Beyond: The SATNAV Transition*, SatNav Program Office, NAV CANADA, <http://www.avw*eb.com/articles/satnav.html*>, no date.

- ◆ *WAAS testing results:* 3 m horizontal, 4.5 m. vertical

- ◆ *LAAS test vehicle:* 0.2 m "accuracy" in NAV CANADA Cessna 182.

"Marconi Canada 12-Channel ALLSTAR GPS Receiver OEM Module" Product Brochure, <www.mtl.marconi.ca/Docs/CMC/Components/allstar.html>.

- ◆ <1 m CEP DGPS RTCM-104

- ◆ < 3 m CEP WAAS DGPS

- ◆ < 16 m CEP GPS w/o SEP

- ◆ < 40 m CEP GPS w/SEP

- ◆ *Update rates:* 1, 2, 5, and 10 Hz.

Federal Aviation Administration (FAA) AC 20-130A, *Airworthiness Approval of Navigation or Flight Management System Integrating Multiple Navigational Sensors*, June 14, 1995.

	Terminal	Nonprecision
◆ 2-D accuracy, nmi (95 percent confidence)		Approach
◆ GPS position fixing error	0.124	0.056
◆ Equipment without GPS	1.7	0.3

◆ Single VOR/DME 0.5

R. A Swider, R. Braff, J. Warburton, and V. Wullschleger, *Local Area Augmentation System (LAAS) Update,* FAA Report, no date, paragraph 2.2

Required LAAS vertical position accuracy:

◆ CAT I CAT II CAT III

◆ 4.0 m 2.5 m 2.5 m

but integrity requirements will result in 0.6–1.0 m need for CAT III.

GROUND SURVEILLANCE

C. Evers, R. Cassell, and D. Lee, *Analysis of ADS-B, ASDE-3, and Multilateration Surveillance Performance— NASA Atlanta Demonstration*, Rannoch Corp. AIAA 17th Annual Digital Avionics Systems Conference, no date.

ASDE/AMASS: *Speed accuracy:* 1.9 kt mean, 1.6 kt standard deviation

◆ *Heading accuracy:* 4.7 deg. mean, 4.3 deg. standard deviation

◆ *Position accuracy:* 3.5 m mean, 2.0 m standard deviation

◆ (95 percent position): ± 6.4 m.

ADS-B: *Total horizontal position error:* mean = 0.7 m, 1 σ = 0.4 m,

◆ (95 percent position) = ± 1.42 m.

Multilateration: *Total horizontal position error:* mean = 3.1-8.5 m, 1 σ = 2.1–4.8 m,

◆ (95 percent position) = ± 7.4 – 15.3 m.

V. Capezzuto, D. Olster, M. Curry, and S. Pendergast, *Runway Incursion Reduction Program (RIRP) Surveillance System, NASA/FAA Atlanta Demonstration,* 17th DASC, <http://www.faa.gov/faa_office/rirp/HTML/DASC_paper.html>.

Table C-1. Estimated Sensor/Fusion Relative Errors

Data source	Standard deviation
ASDE-3/AMASS	3.73 ft
ATIDS/ADS-B—raw	7.50 ft
ATIDS/Multilateration—raw	15.92 ft
ATIDS Tracked	10.60 ft
Fusion Tracked	4.64 ft
Optimum Federated Fusion (Projected)	3.52 ft

ADS-B mean position lagged ASDE as much as 83 ft for accelerating targets because of latency. One Mode-S ADS-B transponder had a process time of 220 msec and a total response time of up to 1.22 sec.

ASDE-X Surface Surveillance System, Raytheon Command, Control, Communication and Information Systems Brochure

Table C-2. ASDE-X Specifications

Frequency	9.0–9.2 GHz
Scan Rate	60 Hz
Azimuth Resolution	50 ft at 1 nmi
Range Resolution	48 ft at 1 nmi

"Multistatic Dependent Surveillance," Sensis Corp. online brochure, <http://www.sensis.com/docs/49/p>, copyright 2000.

This brochure includes a description of a multilateration ground sensor.

CONTROLLER AND PILOT RESPONSE

B. Carpenter and J. Kuchar, *A Probability-Based Alerting Logic for Aircraft on Parallel Approach,* NASA CR 201685, April 1997.

This report describes simulation and analysis of independent approaches to closely spaced runways during use of a precision runway monitor (PRM) radar. The authors develop a sophisticated and elegant alerting logic. The report cites the following FAA information from sources that are referenced but unavailable.

Based on "several experimental measurements:"

◆ "One or two radar updates may occur before the controller decides that an alert is necessary."

◆ "The time delay between Not Transgression Zone penetration and controller response of over 5 seconds in over 20 percent of the time."

◆ "Recordings made of terminal area communications indicate that the controller will be able to transmit without delay [to clear the channel] with a probability of 0.93, but the maximum wait may be over 8 seconds."

◆ "A study by the Precision Runway Monitor Program Office estimates that only one blunder will occur during every 25 million approaches." The same study estimates that the PRM allows only 1 accident in every 250 worst-case blunders.

State variables in the study, based on Rockwell-Collins simulation data and use of DGPS, were assumed to have 0-centered Gaussian distributions standard deviations:

State measurement	Standard deviation
x (relative lateral position)	$\sigma_x = 35$ ft
y (relative longitudinal position)	$\sigma_y = 35$ ft
ψ (heading)	$\sigma_\psi = 2.5°$
ϕ (bank angle)	$\sigma_\phi = 5.0°$

Cassell, et al., *Reduced Aircraft Separation Risk Assessment Model (RASRAM) Description*, Rannoch Corp., February 24, 1997.

This report also describes modeling and analysis of independent approaches to closely spaced runways, using PRM. The report includes distributions of controller and pilot response times. The controller is multimodal, with peaks at the PRM scans. The pilot response is assumed to have a Rayleigh distribution with a sigma of 5 seconds. The convolution of the two is roughlyRayleigh, with a range of 1-20 sec and a broad peak at 7–8 sec.

Appendix D
Abbreviations

ADS-B	automatic dependent surveillance-broadcast
AMASS	airport movement area safety system
AND	aircraft navigational data
ANP	actual navigational performance
ASDE	airport surface detection equipment
ASR	airport surveillance radar
CDTI	cockpit display of traffic information
CFIT	controlled flight into terrain
EGPWS	GPS-based system
FAA	Federal Aviation Administration
FMS	flight management system
IFR	instrument flight rules
ILS	instrument landing system
INS	Inertial Navigation System
LAAS	local area augmentation system
NASA	National Aeronautics and Space Administration
PAWS	Padé approximation with scaling
PRM	precision runway monitor
RASRAM	reduced aircraft separation risk management assessment model
RIRP	runway incursion reduction program
RNP	required navigational performance

SV   synthetic vision

TCAS   transponder-based system

TND   traffic navigational data

TOIMDR   total organizational and intermediate demand rates

VFR   visual flight rules

VOR   VHF Omni Range

WAAS   wide area augmentation system

References

1. *A System for Integrated Reliability and Safety Analysis*, Kostiuk, et al., NASA CR—1999-209548, August 1999.

2. *Analysis of ADS-B, ASDE-3, and Multilateration Surveillance Performance-NASA Atlanta Demonstration*, Evers, Cassell, and Lee, Rannoch Corp. AIAA 17th Annual Digital Avionics Systems Conference, no date.

3. *Techniques for Modeling the Reliability of Fault-Tolerant Systems with the Markov State-Space Approach*, R. Butler, S. Johnson, NASA RP–1348, Sept. 1995.

4. *ASSIST User Manual*, S. Johnson, D. Boerschlein, NASA Langley Research Center, Sept. 14, 1993.

5. *SURE Reliability Analysis, Program and Mathematics,* R. Butler, A. White, NASA TP–2764, 1988.

6. *The SURE Approach to Reliability Analysis*, R. Butler, IEEE Transactions on Reliability, Vol. 41, No. 2, June 1992.

7. B737–300/400/500 FMC Position, http://194.78.76.133/linepilot/FMCPosition.html.

8. *A Probability-Based Alerting Logic for Aircraft on Parallel Approach,* B. Carpenter, J. Kuchar, NASA CR–201685, Apr. 1997.

REPORT DOCUMENTATION PAGE

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (*Leave blank*)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	December 2001	Final Contractor Report

4. TITLE AND SUBTITLE

A Synthetic Vision Preliminary Integrated Safety Analysis

6. AUTHOR(S)

Robert Hemm and Scott Houser

5. FUNDING NUMBERS

WU–728–80–20–00
NAS2–14361

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Logistics Management Institute
2000 Corporate Ridge
McLean, Virginia 22102–7805

8. PERFORMING ORGANIZATION REPORT NUMBER

E–13085

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

National Aeronautics and Space Administration
Washington, DC 20546–0001

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

NASA CR—2001-211285
LMI–NS009S1

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Unclassified - Unlimited
Subject Categories: 03, 66, and 65   Distribution: Nonstandard

Available electronically at http://gltrs.grc.nasa.gov/GLTRS

This publication is available from the NASA Center for AeroSpace Information, 301–621–0390.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This report documents efforts to analyze a sample of aviation safety programs, using the LMI-developed integrated safety analysis tool to determine the change in system risk resulting from Aviation Safety Program (AvSP) technology implementation. Specifically, we have worked to modify existing system safety tools to address the safety impact of synthetic vision (SV) technology. Safety metrics include reliability, availability, and resultant hazard. This analysis of SV technology is intended to be part of a larger effort to develop a model that is capable of "providing further support to the product design and development team as additional information becomes available." The reliability analysis portion of the effort is complete and is fully documented in this report. The simulation analysis is still underway; it will be documented in a subsequent report. The specific goal of this effort is to apply the integrated safety analysis to SV technology. This report also contains a brief discussion of data necessary to expand the human performance capability of the model, as well as a discussion of human behavior and its implications for system risk assessment in this modeling environment.

14. SUBJECT TERMS

Aircraft safety; Enhanced vision; Flight safety; Reliability; Risk

15. NUMBER OF PAGES

68

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	