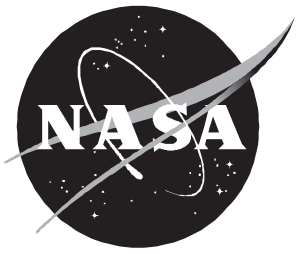
NASA / CR-1998-207660



Reliability Modeling Methodology for Independent Approaches on Parallel Runways Safety Analysis

P. Babcock, A. Schor, and G. Rosch
Charles Stark Draper Laboratory, Cambridge, Massachusetts

April 1998

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 counter-part or peer-reviewed formal professional papers, but having 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 co-sponsored 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 help round out the STI Program Office's diverse offerings include creating custom thesauri, building customized databases, 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***

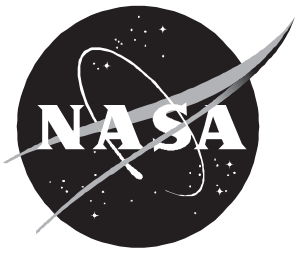
- Email your question via the Internet to help@sti.nasa.gov

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

- Phone 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 21076-1320

NASA / CR-1998-207660



Reliability Modeling Methodology for Independent Approaches on Parallel Runways Safety Analysis

P. Babcock, A. Schor, and G. Rosch
Charles Stark Draper Laboratory, Cambridge, Massachusetts

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23681-2199

Prepared for Langley Research Center
under Contract NAS2-14361

April 1998

Available from the following:

NASA Center for AeroSpace Information (CASI)
7121 Standard Drive
Hanover, MD 21076-1320
(301) 621-0390

National Technical Information Service (NTIS)
5285 Port Royal Road
Springfield, VA 22161-2171
(703) 487-4650

Contents

FIGURES

TABLES

Chapter 1
Introduction

This document is an adjunct to the final report *An Integrated Safety Analysis Methodology for Emerging Air Transport Technologies*. That report presents the results of our analysis of the problem of simultaneous but independent, approaches of two aircraft on parallel runways (independent approaches on parallel runways, or IAPR). This introductory chapter presents a brief overview and perspective of approaches and methodologies for performing safety analyses for complex systems. Ensuing chapters provide the technical details that underlie the approach that we have taken in performing the safety analysis for the IAPR concept.

SAFETY ANALYSIS: PERSPECTIVES

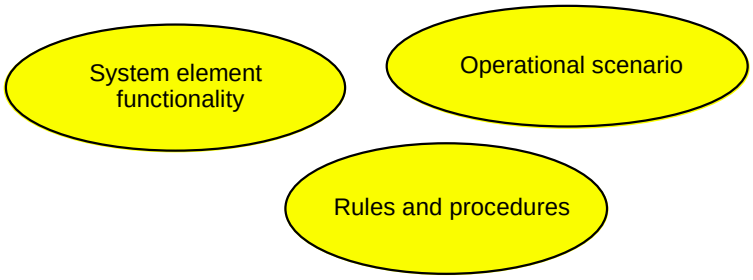
A thorough analysis of system safety must address the problem from a variety of perspectives—each impacting safety in a different way. One perspective relates to the operational environments or operational scenarios within which the system is expected to function. Those environments or scenarios that, by their nature, provide opportunities for unsafe operating conditions have an impact on system safety and approaches to modeling and understanding those impacts must be developed. Another perspective relates to the reliability and availability of the functions performed by the hardware, software, and human components of the system. Failures or degradation in the performance of elements of safety-critical system components have an impact on safety, and models of the reliability of those elements must be developed in order to determine the impact on system safety. Finally, the rules and procedures under which a system operates can have a significant impact on the system safety, and approaches must be developed to analyze the impact of those rules and procedures on safety for all modes of operation of the system.

The three perspectives are illustrated in Figure 1-1 and can be simply summarized as follows:

- *System element functionality*. This entails an analysis of how well and reliably system elements work and the attendant impact on safety.
- *Rules and procedures*. This involves analyses of how the system rules and procedures have been designed to respond in both safe and unsafe situations.

- *Operational scenario*. This involves analysis of the environment in which the system is expected to operate and its attendant impact on system safety.

Figure 1-1. Perspectives of a System-Level Safety Analysis



APPROACHES TO EVALUATING SYSTEM SAFETY

A variety of approaches have been developed to address the three perspectives of system safety analysis described in the preceding section. Some of those approaches are outlined below.

- Statistical analysis of existing systems (descriptive approaches).

  These approaches are based on statistical analyses of data that are collected over long periods of time. The work of Professor Arnold Barnett of Massachusetts Institute Technology (MIT) is an example of this kind of effort. This class of approaches to safety analysis is "after the fact" and is useful in identifying shortcomings in existing systems but has limited utility in predicting the safety consequences of proposed system concepts. Since our interest is in evaluation and analysis of the safety of *new* system concepts, we will not dwell on these approaches.

- Analysis of candidate designs that model human, technical (hardware and software), and procedural aspects of the system (predictive approaches).

  - "ility" analytical modeling.

    Markov, semi-Markov, combinatorial, and fault tree models are used to determine system reliability, availability, maintainability, etc. These approaches have matured over time, and the Markov reliability modeling approach is the one that we have chosen and that is elaborated upon in succeeding chapters. Further discussion is included in Chapter 2.

- Simulation.

  A variety of statistical event simulation approaches including discrete event simulations, importance sampled Monte Carlo simulations, and hybrid simulations with both human operators and hardware in the loop have been used to predict the safety of proposed system concepts. The advantage of simulations is that they are typically easier to design and implement than the analytical models described in (a) above. The disadvantage is that, in order to obtain statistically significant results for very low probability events, many simulations must be performed. The approach that we have taken in the IAPR study is to use a Markov model to determine the probabilities of being in potentially unsafe system states and to employ a deterministic simulation of the system operating in those states. The system safety metrics generated by the simulations are then weighted by the Markov state probabilities to obtain the total expected values of those metrics. Further discussion of simulation approaches is included in Chapter 2.

- Human performance modeling.

  The development of models to predict the effect of workload and task design on human error rates and human response times was considered beyond the scope of the effort for this task. Indeed, good models of the human are critical to the complete analysis of a system. Due to constraints on time and budget, we chose to use simple models of human performance and to embed those in our system Markov models. Thus, at the level of failure and performance degradation, the function of a person is characterized no differently than that of other system components.

- Formal methods.

  These are mathematical, logic-based approaches for specifying and implementing safety-critical hardware and software systems and for verifying correctness and completeness of their design and implementation. Typical of these approaches are those taken by professor Nancy Lynch at MIT and professor Nancy Leveson at the University of Oregon. These approaches are best applied when the system is defined at a higher level of detail than the IAPR concept that we are investigating.

- Information security.

  For safety-critical information exchanges, for example Automatic Dependent Surveillance-Broadcast (ADS-B) for local air traffic status between aircraft when air separation responsibility is transferred to pilots, the security and integrity of the exchanged information are clearly

critical. One way to view information security is in terms of protecting the computers and communications assets of the system. There are several protection mechanisms: protection against unauthorized alterations of the data and protection against denial of exchange of data. To date, little has been done in the development of models of information security and its impact on safety-critical functions. This is an area for research and is not addressed further here.

OBJECTIVE: A UNIFIED FRAMEWORK FOR INTEGRATED SAFETY ANALYSIS

How best can the many approaches to evaluating system safety be combined into a unified framework for safety analysis? Drawing the three perspectives shown in Figure 1-1 into a unifying framework that directly addresses the interactions and coupling among those perspectives, our first step toward such a unification is described below and has been applied in evaluating the IAPR concept. As time and experience in applying this unified approach evolve, we anticipate further refinements will be made.

The integrated safety analysis that we employ is distinguished by its ability to merge system design or functionality information with a parameterization of a system's situation. This is illustrated in Figure 1-2. The "system" may include both air and ground subsystems within this analysis framework.

In Figure 1-1, we see that system safety is being addressed from a variety of perspectives, each of which impacts safety. These include

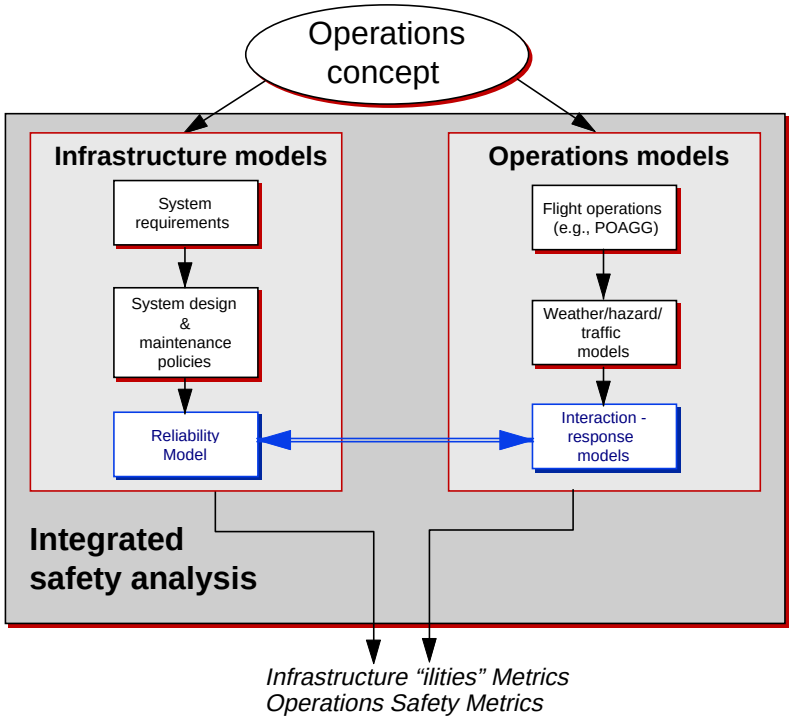
- system functionality, the analysis of how reliably the system components perform;
- rules and procedures, the analysis of how the system is designed to respond in both safe and unsafe situations; and
- operational scenario, the analysis of the environment in which the system is expected to operate.

Integrating models that quantify each one of these three elements creates an analysis capability that is now systemwide and responsive to ongoing changes in the definition and requirements of the operational concept.

The steps leading from requirements derived for an operational concept to the development of a reliability model of the system architecture that has been proposed to meet those requirements are shown on the left side of Figure 1-2. This represents a traditional reliability/safety modeling process. On the right are the models required to capture the environment in which the system is to operate as well as

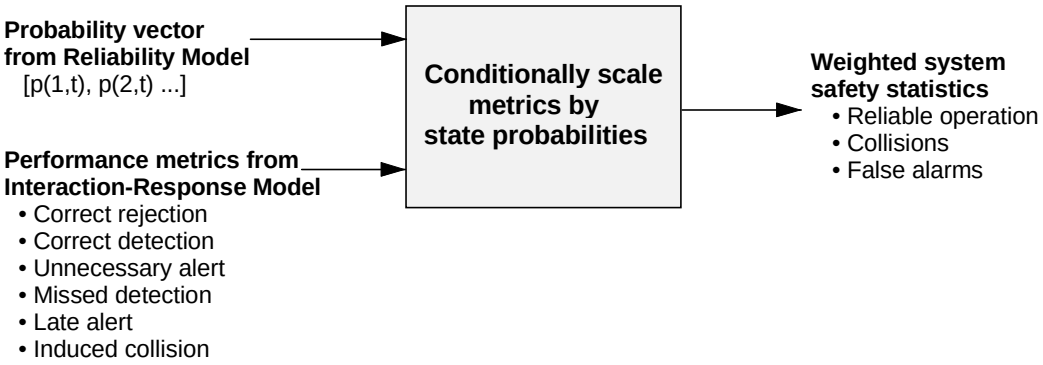
the interaction of those environmental models with response models that represent the execution of the rules and procedures that have been developed for the candidate concept. This represents a modeling process for the dynamic analysis of the system's situation.

Figure 1-2. Integrated Safety and Reliability Modeling and Evaluation



Our approach to system safety analysis results from the integration of the Reliability model and the Interaction-Response model. The Interaction-Response model provides information regarding the frequency of encounters and the predicted outcome of those encounters as a function of the system's alerting system and ability to resolve encounters. The Reliability model provides, as a function of time, probabilities associated with the critical systems' availability and failure states. Scaling the operations safety metrics from the Interaction-Response model by the system state probabilities from the Reliability model creates the system-level safety statistics. This process is illustrated in Figure 1-3.

Figure 1-3. Combining Model Outputs



Products of this analysis include

- predicted accident statistics,
- predicted false alarm statistics, and
- predicted system availability and reliability.

Moreover, as the operational concept evolves, the impact of changes in system architecture, rules and procedures, and operational scenarios can be easily re-evaluated with this methodology.

Chapter 2
Overview of Analytical Reliability Modeling Techniques

The analytic approaches to quantifying system reliability fall into three classes: Monte Carlo simulations, combinatorial reliability models (see below), and Markov modeling. The strengths and weaknesses of these three approaches are briefly described in this section.

MONTE CARLO SIMULATIONS

Simulation [1, 2] can be used to determine reliability by generating failure and repair events at times distributed according to the component failure and repair rates. These simulations are repeated until statistically significant reliability measures are accumulated. A major strength of the simulation approach is its ability to analyze very complicated repair and reconfiguration scenarios, with relatively little knowledge required beyond a description of the system to be analyzed. However, a key difficulty of this method is that for highly reliable systems the failure rate is so low that, in order to accumulate a statistically meaningful number of events, a very large number of simulations must be run. While there are means (collectively known as variance reduction techniques) of increasing the efficiency of the basic method, the underlying difficulty remains a drawback.

COMBINATORIAL RELIABILITY MODELS

Combinatorial reliability models [3] have been widely used. Fault-tree analysis [4], for example, has become a standard analytical method for reliability prediction in a wide variety of applications. This analytical technique combines component failure probabilities, based on the system architecture and redundancy management approach, to determine system reliability. Since there is no explicit simulation of system operation, the combinatorial technique avoids the deficiencies of the Monte Carlo simulation. There are, however, three limitations to this approach. First, the fault tree is constructed to predict the probability of the system being in a *particular* operating condition (for example, a working condition or a failed condition). If it is desired to investigate the probability of being in other conditions, such as a variety of different operating modes, then new fault trees have to be constructed. Second, it is difficult to include events that have order dependencies, such as repairs and explicit modeling of reconfiguration strategies. Even in relatively simple systems, there often are subtle sequence dependencies. Finally, the nature of the combinatorial analysis requires that all combinations of

events for the entire time period must be included. For complex systems, this results in a complicated fault tree that is difficult to construct and validate.

MARKOV MODELING

Markov modeling techniques have been increasingly used for reliability prediction [3, 5, 6]. These techniques have also been used successfully to aid in the design of fault-tolerant systems [7, 8]. A Markov reliability model calculates the probability of the system being in various states as a function of time. A state in the model represents the system status with respect to component failures and the behavior of the system's redundancy management strategy. Transitions from one state to another occur at given transition rates, which reflect component failure and repair rates and redundancy management performance. Each element in the model's state vector represents the time-dependent probability of the system being in a specific state. Since the Markov model traces the evolution of state probabilities based on the transition rates mentioned above, it is not explicitly simulating the system and, therefore, does not have the deficiencies associated with the Monte Carlo technique. The Markov model is cast into a system of differential equations. Sequence dependencies, such as repairs and redundancy management decisions, are included naturally. Furthermore, the differential nature of the model means that it is not necessary to generate *explicitly* all possible combinations of events that can occur over the entire time period in question; rather, it is only necessary to model events that can occur during an infinitesimal time step. Of course, there are also some drawbacks to this method. First, the state space grows exponentially with the number of components. However, techniques have been developed to render this problem tractable in many situations of interest [9]. Second, treatment of complex mission scenarios and repair strategies, although possible, are generally cumbersome.

It should be emphasized that the reliability of a system does not depend on the analytical method used to evaluate it, so long as any approximations and simplifications are consistently applied or interpreted.

Chapter 3
The Modeling Process

ROLE OF THE RELIABILITY MODEL

The objective of the reliability model is to predict the state of the aircraft capabilities at the start of and during an independent approach. In general, when an aircraft lines up for an independent approach, it will have been in-flight for several hours. Assuming that the aircraft had no failures prior to takeoff, in the time from takeoff until the start of the approach, failures of components within the systems of the aircraft may have occurred that have reduced its capabilities. The reduced capabilities, possibly undetected by the pilot, can affect the performance of the aircraft during the approach and result in the aircraft drifting or blundering into the path of an aircraft approaching the adjacent runway. Alternately, the component failures during en route flight may prevent an independent approach from taking place. Procedural rules may prohibit the pilot from attempting an independent approach if there is a known loss of a specific aircraft capability or, in the worst case, failures could cause the loss of the aircraft. The reliability model calculates the probabilities of the reduced capabilities impacting the safety of the aircraft when an independent approach is attempted.

The actual process of generating a reliability model requires information on architecture, component characteristics, operational requirements, and reconfiguration procedures. The system architecture provides information such as what components exist and how they are connected, both physically and logically. The model also needs to be fed various component characteristics, such as failure and repair rates. The operational requirements provide a definition of what equipment or abilities are needed to achieve an operational state. The reconfiguration procedures are the actions taken when a failure occurs so that system operation remains in the most desirable mode.

The process of generating a reliability model for a system can be divided into three steps. First, the system needs to be carefully examined. The goal is to discover how the system operates and its critical elements. This step results in a system description. Second, the impact of failures is explored. This step is often called a failure modes and effects analysis (FMEA). During this step, the failure modes of the system are delineated. Third, the Markov model is constructed. Information on system operation from step one is used to guide modeling decisions such as the proper representation for the human elements. The model is a systematic representation of the FMEA from step two. Each of these steps is discussed in a subsection below.

DEFINE AIRCRAFT FUNCTIONAL ELEMENTS

The first step in developing the reliability model needed for the IAPR system safety model is to define the aircraft functions that directly and uniquely impact the inputs of the Interaction-Response model. The functions or capabilities of the aircraft used in the IAPR system safety model are defined in Table 3-1. These functions were developed by reviewing the current status of the development of the Airborne Information for Lateral Spacing system (AILS). However, the function definitions and the system description of the IAPR system presented in the next section are not strictly based on the AILS system. The function definitions and the system description represent the capabilities and components, respectively, which are likely to comprise an IAPR system, since a specification of an AILS system does not yet exist.

Table 3-1. IAPR System Functional Elements

Element	Description
IAPR RNP	The ability to perform conformance monitoring of an aircraft's performance and adherence to its approach path (Required Navigation Performance, or RNP)
ADS-B/surveillance data link	The ability of an aircraft to broadcast, receive, and process ADS-B information for situational awareness, conflict avoidance, and airspace management
Collision-alerting avionics	The ability of an aircraft's system to predict a probable collision with another aircraft during approach and landing and to provide timely and reliable alerts so that the pilot can avoid the collision (this includes alerting logic, processing, and display monitors)
Guidance and control	The aggregate of all other aircraft capabilities (e.g., propulsion, flight control, engine control) and support subsystems exclusive of the previous three functions
Pilot	The capability of the pilot(s) to safely operate the aircraft

The function definitions are limited to the capabilities of a single aircraft. The IAPR system is an aircraft-based collision avoidance system, but there may be dependencies on systems external to the aircraft that can affect safety. The dependencies with the aircraft that may be approaching the adjacent runway will be accounted for because the same function definitions are applied to the adjacent aircraft. The dependencies on systems exclusive of the two aircraft are not included in the reliability model. These would include any monitoring and interaction from the ground controller or interaction with other aircraft in the airport area.

The functions defined in Table 3-1 are the capabilities of the aircraft required for an independent approach. The first three functions represent capabilities that need to be added to present commercial aircraft to support IAPR. The fourth function, guidance and control, represents all the capabilities and systems of the aircraft, exclusive of those required for the first three functions, which can affect safety of an independent approach. The fifth function isolates the capability the pilot (and crew) provide in the safe operation of the aircraft.

DESCRIBE THE SYSTEM

The following system description defines the reliability characteristics of the IAPR system. That is, the system description presented defines the individual components that can fail, how they are interconnected, the redundancy of the components and subsystem functions, and the redundancy management logic.

To demonstrate the safety analysis methodology, a low-fidelity description of a plausible IAPR system has been created. A design for the IAPR system presently does not exist. So, a system is created that provides the functionality expected for an IAPR system and includes some degree of fault tolerance. The system description constructed is complex enough to demonstrate the application of the safety analysis methodology, but simple enough so that minimal resources would be needed to develop the reliability model. The low-fidelity model does not limit the approach.

Each system component in the system description is assigned to only one function to maintain the independence of the functions. The advantage of maintaining the independence of the functions is that it enables the probability of any system state to be computed in a simple and direct manner. For example, the probability of the system being *fully operational*, at some time *t*, is simply the product of the probabilities of each of the functions being in their fully operational states at time *t*.

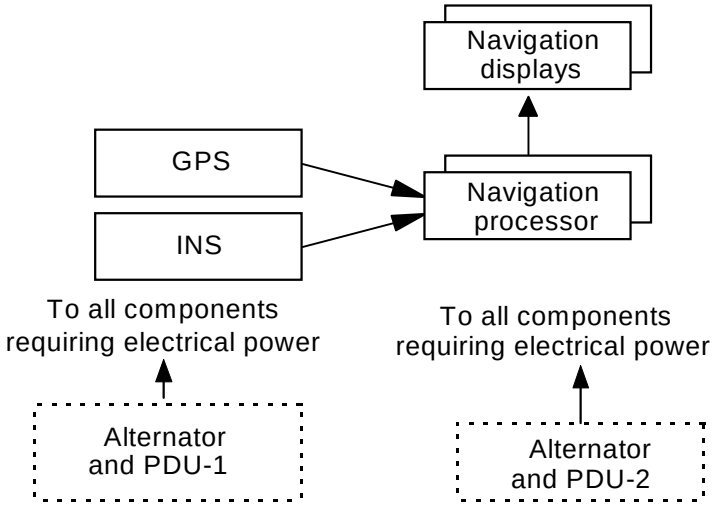
Figures 3-1 through 3-4 present the block diagrams for the system description. These are discussed in the following subsections. However, to fully understand the block diagrams, several conventions need to be defined.

- Components shown with broken lines are assigned to another function. They are included in the following block diagrams of some of the functions to indicate the interconnection between the components of different functions, and they are not considered one of the components necessary for the function.

- Overlapping blocks indicate dual-redundant components. Dual-redundant components are both on-line if functional, but only one is necessary for the function to be fully operational.

- The connections between components shown should be understood to indicate that the connected components are fully cross-strapped. For example, in Figure 3-1 the connection between the “navigation processors” and the “navigation displays” (indicated by the arrow) means each of the two navigation processors is connected to each of the navigation displays.

Figure 3-1. IAPR RNP



Note: PDU-power distribution unit.

Figure 3-2. ADS-B/Surveillance Data Link

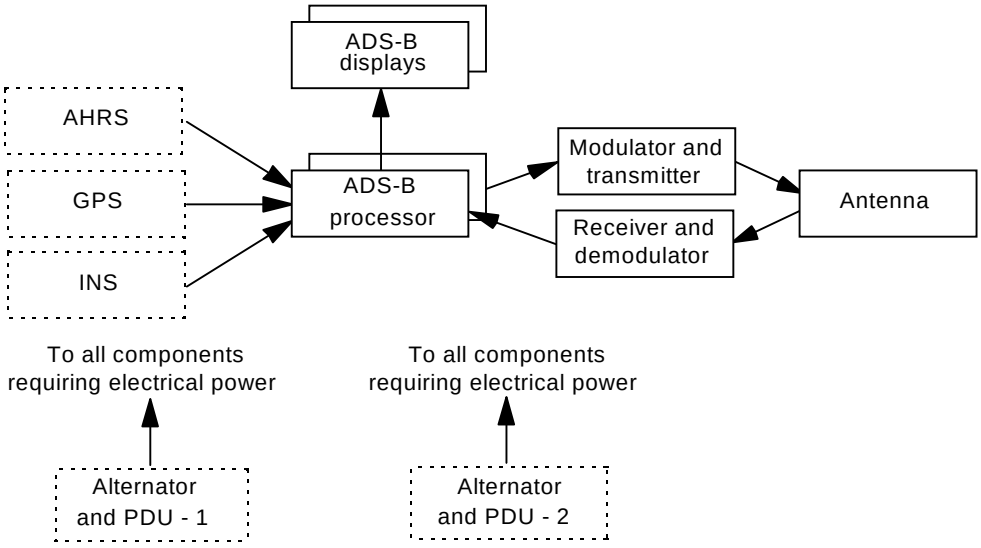
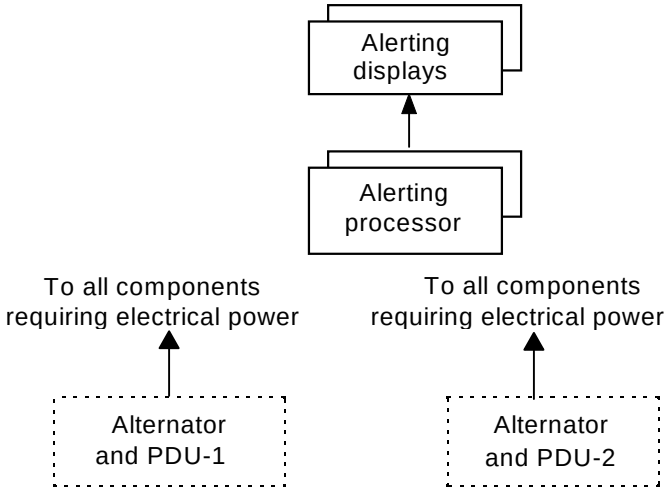


Figure 3-3. Collision-Alerting Avionics



Independent Approaches on Parallel Runways Required Navigational Performance

Figure 3-1 presents the block diagram of the IAPR RNP system. The six components shown framed with solid lines provide the IAPR RNP function defined in Table 3-1. The Global Positioning System (GPS) receiver and Inertial Navigation System (INS) provide the sensed position of the aircraft. The GPS receiver provides discrete position updates at fixed intervals in time. The INS data are integrated with the position updates from the GPS receiver to provide a more frequent position update than can be obtained with the GPS receiver alone. The data fusion and the navigation computation are done in the navigation processor. The navigation displays provide the flight crews with the navigation information and alerts when navigation containment is violated.

Table 3-2 presents the operational states of the IAPR RNP function that are pertinent to the IAPR safety model. The IAPR RNP system is fully operational if both the GPS receiver and the INS 1 navigation processor and 1 navigation displays are functional. The system is “degraded” if either the GPS or INS has failed, the failures are detected and compensated for, and an indication has been given to the pilot by the system. The “failed safe” state is the state of the system when component failures have caused the loss of the IAPR RNP navigation function and an indication is provided to the pilot to evidence that this capability no longer is available. Alternately, the “failed uncovered” state represents the loss of the function, but an indication is not provided to the pilot to indicate the loss of that function.

Table 3-2. IAPR RNP Navigation Operational States

State	Definition	Impact
Fully operational	TSE (total system error) is less than containment limit and no alert of loss of RNP capability	Navigation capability available for normal approach; ideal distributions
Degraded	Loss of either GPS or INS resulting in a degraded navigation capability	Navigation capability available for normal approach; nonideal distributions
Failed safe	Alert of loss of RNP capability	No longer able to perform independent approaches; approach aborted
Failed uncovered	TSE is greater than containment limit and no alert of loss of RNP capability	Invalid self-knowledge and broadcast of navigation data

ADS-B/Surveillance Data Link

Figure 3-2 shows the block diagram of the ADS-B/surveillance data link system. The ADS-B/surveillance data link system transmits the IAPR state variable data for the aircraft (which the aircraft performing an independent approach on the adjacent runway can monitor) and receives the IAPR state variable data from the adjacent aircraft. The IAPR state variable data broadcast from the aircraft enables the collision alerting avionics of other aircraft to predict a collision. Conversely, the IAPR state variable data the aircraft receives from other aircraft enables it to predict a collision with those aircraft. The Attitude Heading Reference System (AHRS), GPS receiver, and INS provide the sensor data that make up the IAPR state variable data. However, these three sensors provide redundant information and sufficient data are available if two of the three are functional. (Note that the GPS receiver and the INS are not included in the ADS-B/surveillance data link function, having already been included in the IAPR RNP navigation function.)

For the ADS-B/surveillance data link function to be fully operational, 1 ADS-B processor, 1 ADS-B displays, the modulator and transmitter, the receiver and demodulator, and the antenna must be functional. The degraded, failed safe, and failed uncovered states are defined in Table 3-3.

Table 3-3. ADS-B/Surveillance Data Link Operational States

State	Definition	Impact
Fully operational	Valid broadcast and reception of broadcasts from other aircraft	Transmit and receive functions are fully available
Degraded	Unable to receive broadcasts from other aircraft and may or may not receive alert of capability loss; broadcast capability functioning	Knowledge of other aircraft is invalid but approach is allowed
Failed safe	Invalid broadcast and alert of capability loss and, possibly also, loss of reception capability of broadcasts from other aircraft	No longer able to perform independent approaches; approach aborted
Failed uncovered	Invalid broadcast and no alert of capability loss	Other aircraft do not receive valid surveillance data

Collision-Alerting Avionics

The collision-alerting avionics block diagram and operational states are shown in Figure 3-3 and Table 3-4, respectively. The collision alerting avionics is fully operational if 1 alerting processor and 1 alerting displays are functional. The alerting processor receives the position of its own aircraft from the IAPR RNP function and the IAPR state variable data from the aircraft approaching on the adjacent runway from the ADS-B/surveillance data link system.

Table 3-4. Collision-Alerting Avionics Operational States

State	Definition	Impact
Fully operational	Collision-alerting capability functioning properly	Alerting-capability available for normal approach
Failed safe	Collision-alerting not available and alert of capability loss	No longer able to perform independent approaches; approach aborted
Failed uncovered	Collision-alerting not available and no alert of capability loss	Unable to detect blunders of other aircraft but approach is not aborted

Guidance and Control and Pilot

Figure 3-4 shows the block diagram of the Guidance and Control and Pilot systems. The pilot and crew are included here as the block denoted “Pilot.” The Guidance and Control system simply represents all of the systems of the aircraft exclusive of the IAPR RNP, ADS-B/surveillance data link, and Collision-Alerting Avionics systems that all impact safety. The pilot provides inputs to engine and

flight control to ultimately direct the thrust and flight path of the aircraft. Propulsion is provided via the engines. Engine control is provided by the engine processor using input from the pilot and engine sensors. Flight control is through the control processor, which moves the control surfaces based on inputs from the pilots and aircraft state and environment sensors. The alternator and PDUs generate and distribute electrical power to all components requiring it.

Figure 3-4. Guidance and Control and Pilot Systems

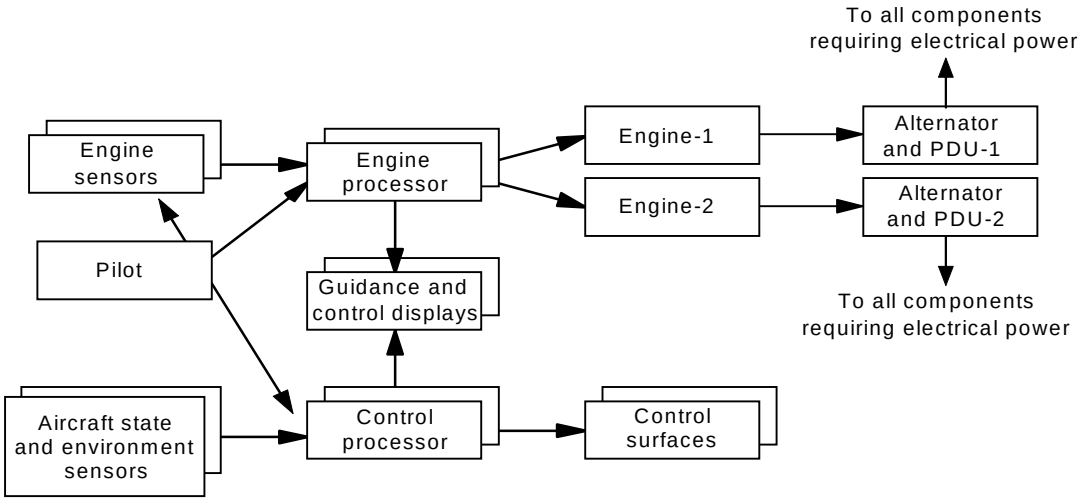


Table 3-5 presents the operational states of the guidance and control function. The guidance and control system is fully operational if 1 engine sensors, 1 aircraft state and environmental sensors, 1 engine processor, 1 control processor, 1 guidance and control displays, both engines, 1 control surfaces, and 1 alternator and PDU are functional. The failed safe state would result from the covered failure of 1 engine. Any uncovered failures or covered failures that result in the system not satisfying the definition of fully operational would place the guidance and control function in the failed uncovered state.

Table 3-5. Guidance and Control Operational States

State	Definition	Impact
Fully operational	All other capabilities and support subsystems operational	Capability is fully available for normal approach
Failed safe	Loss of sufficient capability and knowledge of loss	No longer able to perform independent approaches; approach aborted
Failed-uncovered	Loss of sufficient capability and no knowledge of loss or inability to control aircraft	Worst-case blunder

Table 3-6 presents the operational states of the pilot function. The pilot function is meant to capture the effect of human error in the safety of an independent approach. While an actual model of the reliability of the human in control of the aircraft is beyond the level of work being presented here, the pilot function can still be broken down into operational states to demonstrate how the reliability of the human is integrated into the safety analysis methodology.

Table 3-6. Pilot Operational States

State	Definition	Impact
Fully operational	Pilot functioning nominally without any faults	Alerting capability available for normal approach
Recoverable fault	Pilot fault has occurred; is possible to recover from fault	No impact prior to approach; aircraft blunder after start of approach
Nonrecoverable fault	Pilot fault has occurred; is not possible to recover from fault	No impact prior to approach; aircraft blunder after start of approach

BUILD A MARKOV MODEL

A set of Markov reliability models are constructed from the system described earlier in the system description section. The Markov models are developed in accordance with the techniques presented in Chapter 4. A separate model is constructed for each function defined in Table 3-1.

RESULTS AND DISCUSSION

Markov reliability models are used to calculate the probabilities of being in the operational states of each of the functions. Table 3-7 presents the baseline failure rates and coverage probabilities for each of the components identified in the system description for the IAPR system. The failure rates and coverage probabilities constitute nearly all of the input parameters for the models. The only missing input parameter is recovery rate from an intermittent human failure for the Pilot model. The baseline value for this rate is set at 3.6×10^2 recoveries/hour.

Table 3-7. Baseline Failure Rates and Coverage Probabilities

Component	Failure rate (failures/hour)	Coverage probability
IAPR RNP		
GPS	3.0E-5	0.99
INS	1.0E-4	0.99
Navigation displays	2.0E-5	0.999, 0.99
Navigation processor	1.0E-5	0.99, 0.95

Table 3-7. Baseline Failure Rates and Coverage Probabilities (Cont.)

Component	Failure rate (failures/hour)	Coverage probability
ADS-B/Surveillance Data Link		
AHRS	1.0E-5	0.99
ADS-B displays	2.0E-5	0.999, 0.99
ADS-B processor	1.0E-5	0.99, 0.95
Modulator and transmitter	5.0E-5	0.99
Receiver and demodulator	5.0E-5	0.99
Antenna	1.0E-6	1.00
Collision-alerting logic		
Alerting displays	2.0E-5	0.999, 0.99
Alerting processor	1.0E-5	0.99, 0.95
Guidance and control		
Engine sensors	4.0E-5	0.99
Engine processor	1.0E-5	0.99
Engine	1.0E-5	0.999
Alternator and PDU	2.0E-5	0.99
Guidance and control displays	2.0E-5	0.999
State and environment sensors	4.0E-5	0.99
Control processor	1.0E-5	0.99
Control surfaces	5.0E-6	0.99
Pilot		
Intermittent human failure	1.0E-4	1.00
Permanent human failure	1.0E-6	1.00

Note: For coverage probabilities entered as two numbers, the first number is the coverage probability of first failure in redundant components, and the second number is for second failure in the redundant components.

The input parameters used are not from any specific source and are selected with the intent of highlighting the fidelity of the Markov reliability models. Typical values of failure rates and coverage probabilities are assigned for the components that are likely to comprise the system. The failure and recovery rates for the pilot model are not based on any empirical data.

Table 3-8 shows the calculated probabilities for the operational states of each function. The Markov models are evaluated using Version 7.9.8 of the SURE Reliability Analysis program developed by NASA Langley Research Center [12]. The Markov model state probabilities are calculated for 4 and 10 hours. These represent two time intervals from aircraft takeoff to the lineup point for an independent approach.

Table 3-8. Probabilities of Operational States

Operational state	Probability of being operational state			
	At 4 hours		At 10 hours	
	Lower bound	Upper bound	Lower bound	Upper bound
IAPR RNP				
Fully operational	9.9948E-1	9.9948E-1	9.9870E-1	9.9870E-1
Degraded	5.15E-4	5.15E-4	1.29E-3	1.29E-3
Failed safe	5.49E-8	5.49E-8	3.43E-7	3.43E-7
Failed uncovered	6.16E-6	6.16E-6	1.54E-5	1.54E-5
ADS-B/surveillance data link				
Fully operational	9.9959E-1	9.9960E-1	9.9899E-1	9.9899E-1
Degraded	2.00E-4	2.00E-4	5.00E-4	5.00E-4
Failed safe	2.02E-4	2.02E-4	5.05E-4	5.05E-4
Failed uncovered	2.96E-6	2.96E-6	7.40E-6	7.41E-6
Collision-alerting logic				
Fully operational	1.0000E+0	1.0000E+0	1.0000E+0	1.0000E+0
Failed safe	7.83E-9	7.83E-9	4.90E-8	4.90E-8
Failed uncovered	9.60E-7	9.60E-7	2.40E-6	2.40E-6
Guidance and control				
Fully operational	9.9991E-1	9.9991E-1	9.9977E-1	9.9978E-1
Failed safe	7.99E-5	8.00E-5	2.00E-4	2.00E-4
Failed uncovered	1.03E-5	1.03E-5	2.60E-5	2.61E-5
Pilot				
Fully operational	1.0000E+0	1.0000E+0	9.9999E-1	9.9999E-1
Recoverable failure	2.78E-7	3.59E-7	2.78E-7	7.83E-7
Nonrecoverable failure	4.00E-6	4.00E-6	1.00E-5	1.00E-5

Note that the results in Table 3-8 are presented as bounds on the probabilities of being in the states of each function. The bounds occur from two sources. The first source, which affects all of the models, is that the SURE program calculates and outputs the bounds of the probability of being in the states of the model (numerical approximation error). The second source, which affects just the ADS-B/surveillance data link and guidance and control Markov models, is the model truncation aggregation technique used to limit the size of these models. Model truncation introduces some uncertainty into the predictions [11].

The probabilities shown in Table 3-8 are used by the Impact Model discussed in Chapter 4. However, there are some system probabilities produced by the Markov reliability models that are also of interest. Some component failures occurring before the approach lineup can preclude an independent approach. Table 3-9 pres-

ents two metrics of interest. The first is the probability that insufficient capability is available to attempt an independent approach and the approach is aborted by the pilot. This is the probability that one or more of the functions, excluding the Pilot function, is in its failed safe operational state. The second metric is the probability of a loss of the aircraft before the approach lineup. This is the probability of being in the failed uncovered operational state of the guidance and control function.

Table 3-9. Probabilities of Other Operational States

Metric	Probability	
	Upper bound at 4 hours	Upper bound at 10 hours
Insufficient capability independent approach	2.82E-4	7.06E-4
Loss of aircraft before approach lineup	1.03E-5	2.61E-5

Chapter 4
Markov Modeling Method

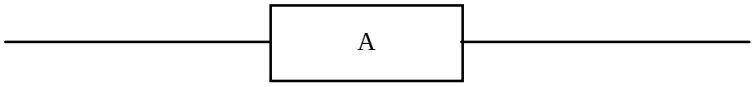
BACKGROUND

Markov modeling techniques provide a systematic means of investigating system reliability for large, complex systems. They permit the inclusion of sequence-dependent events, such as repairs, in a natural fashion. One of the most powerful aspects of Markov models is their ability to permit simplifying approximations to be made and to provide means to obtain bounds on these approximations. The basic concepts of Markov modeling are introduced via simple, but representative examples. These examples clearly point out the general flexibility as well as the main drawback of the technique, particularly the rapidly proliferating state space.

SINGLE-COMPONENT SYSTEM

Figure 4-1 shows a single-component system. The first step in modeling the reliability of that system is to determine what is required for the system to be in an operational state. That single-component system has a trivial operational requirement: it is operational if the single component, A, has not failed. (Conversely, the system is failed if component A has failed). While this step is simple for that system, it is often one of the most complicated steps in modeling a complex system, characterized by many operational states and subtle interactions among components.

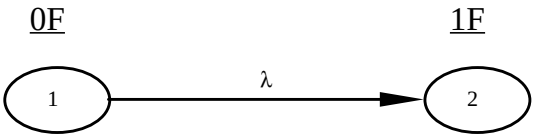
Figure 4-1. Single-Component System Block Diagram



Given the system operational requirements, the next step is to construct Markov model states. A *state* represents a unique configuration of failed and operational elements, sometimes distinguished by the sequence of the failures that led to it. Figure 4-2 shows the Markov model for the one-element system. In general, a model is generated by first creating state 1, the state where there are no failed components in the system. The various transitions out of state 1 represent failures of the system components, accounted for individually or in groups. In this case, there is only one component; thus, a transition denoted λ is created leading to state 2. This state represents this system when component A is failed. Noting the operational requirements for this system, state 2 is labeled as a system failure.

Since there is only one component in the system and its failure has been accounted for, the Markov model is complete.

Figure 4-2. Single-Component System Markov Model



This system's reliability is just the probability, as a function of time, of being in state 1. Actually, there is a probability associated with each state. For example, at time zero the probability of being in state 1 (no failures) is 1 (or 100 percent) and the probability of being in state 2, or any other state, is 0. Parameter λ on the transition in the model not only indicates that component A has failed along this transition, but that the component's failure rate is λ failures per hour. Throughout this discussion, it will be assumed that all failure rates are constant in time. To obtain the system reliability as well as other state probabilities of interest as a function of time, the probability "flowing" out of state 1 into state 2 needs to be tracked.

Probability flow is the product of the transition rate and the state probability for the state at the origin of the transition. Thus, a state with zero probability has no probability flowing out of it, a state with no exiting transitions has no flow out, and a state with probability equal to 1 and an exiting transition rate of λ has an instantaneous flow out equal to λ. The rate of change of each probability is then given by the net probability flow into the corresponding state. Therefore, a Markov model is thus mathematically described by a set of differential equations governing the evolution in time of the probabilities of being in each state.

Using the definition of the probability flows, the following equations are obtained for the Markov model shown in Figure 4-2:

$$dP_1(t)/dt = -\lambda P_1(t) \qquad \text{[Eq. 4-1]}$$

$$dP_2(t)/dt = \lambda P_1(t) \qquad \text{[Eq. 4-2]}$$

These equations, representing the rate of changes in each state variable (P_1 and P_2), are called state equations. Equation 4-1 shows that the rate of change in probability for state 1 is the exiting transition rate λ times the probability of being in state 1. The minus sign indicates that the transition is out of the state and, therefore, reduces the probability of being in state 1. Equation 4-2 is interpreted similarly. Note that the flow is *into* state 2; the positive term indicates an entering transition that increases the probability in state 2. Also, the flow into state 2 is the

rate λ times the probability of state 1; the flow on this transition is due to state 1, the origin of the transition. Equations 4-1 and 4-2, along with the initial condition of the state probabilities, $P_1(0) = 1$ and $P_2(0) = 0$, provide a complete description of the system's reliability. Markov models have the property that a flow leaving one state enters another, as shown in Equations 4-1 and 4-2. Hence, the *total* system probability does not change as the system evolves. This fundamental property is called conservation of probability. The sum of all the system's state probabilities is always equal to 1.

There are many ways of solving Equations 4-1 and 4-2 in closed form, such as standard integration or Laplace transform. Using any convenient technique and recalling that the failure rate λ is constant, yields the following solution:

$$P_1(t) = e^{-\lambda t} \qquad \text{[Eq. 4-3]}$$

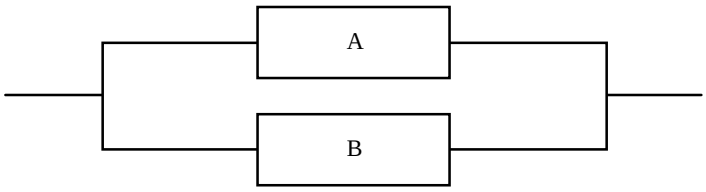
$$P_2(t) = 1 - e^{-\lambda t} \qquad \text{[Eq. 4-4]}$$

State 1 starts with a probability of 1 and decays exponentially toward 0, while state 2 has a probability initially at 0 that grows toward 1. Notice that the sum of the two state probabilities is 1 at all times, thus indicating the conservation of probability.

TWO-COMPONENT PARALLEL SYSTEM

Figure 4-3 shows a two-component system in which the components are connected in parallel. The requirement for system operation is that at least one of the two components is working.

Figure 4-3. Two-Component Parallel System Block Diagram



A Markov model of this system is shown in Figure 4-4. Notice that states 4 and 5 distinguish between the two possible orders of component failure leading to a system loss. The nature of the Markov model makes such order dependencies easy to include. The state equations for this system are

$$dP_1(t)/dt = -(\lambda_A + \lambda_B) P_1(t)$$

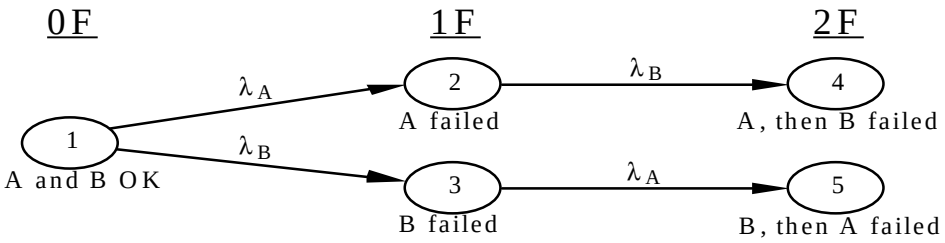
$$dP_2(t)/dt = \lambda_A P_1(t) - \lambda_B P_2(t)$$

$$dP_3(t)/dt = \lambda_B P_1(t) - \lambda_A P_3(t) \qquad \text{[Eq. 4-5]}$$

$$dP_4(t)/dt = \lambda_B P_2(t)$$

$$dP_5(t)/dt = \lambda_A P_3(t)$$

Figure 4-4. Two-Component Parallel System Markov Model



For constant failure rates and an initial condition of

$$P(0) = [1\ 0,\ 0\ 0\ 0]^t,$$

the solution is

$$P_1(t) = e^{-(\lambda_A + \lambda_B)t}$$

$$P_2(t) = e^{-\lambda_B t} - e^{-(\lambda_A + \lambda_B)t}$$

$$P_3(t) = e^{-\lambda_A t} - e^{-(\lambda_A + \lambda_B)t} \qquad \text{[Eq. 4-6]}$$

$$P_4(t) = [\lambda_A + \lambda_B(e^{-(\lambda_A + \lambda_B)t}) - (\lambda_A + \lambda_B)(e^{-\lambda_B t}]/(\lambda_A + \lambda_B)$$

$$P_5(t) = [\lambda_B + \lambda_A(e^{-(\lambda_A + \lambda_B)t}) - (\lambda_A + \lambda_B)(e^{-\lambda_A t}]/(\lambda_A + \lambda_B)$$

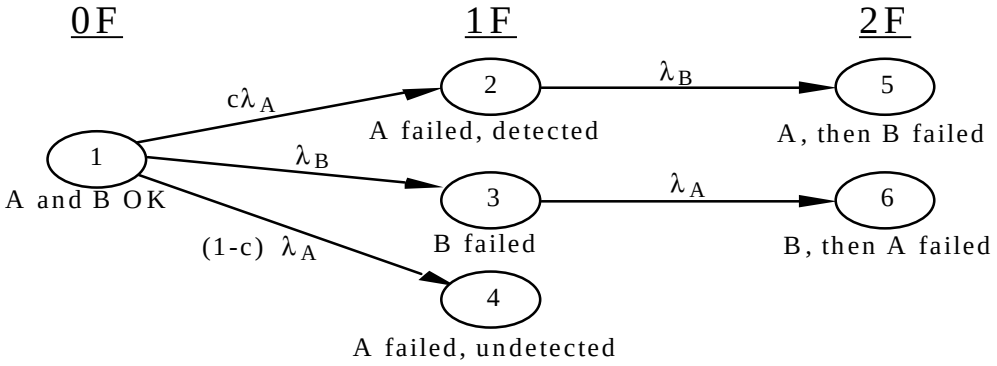
As expected, $P_1(t)$, $P_2(t)$, and $P_3(t)$ have final values of 0 and the sum of the final values of $P_4(t)$ and $P_5(t)$ is 1.

TWO-COMPONENT PARALLEL SYSTEM WITH IMPERFECT COVERAGE

The architecture of a two-component parallel system (Figure 4-3) will be used again. In this example, imperfect coverage is added to the description of the system's operation. Coverage is the ability to successfully recover from a component failure. Coverage for a specific component is represented as the fraction of that component's failures that can be detected and successfully recovered from.

If the system starts out using component A and a covered failure occurs, then the system reconfigures to rely on component B. Any subsequent failure of B results in a system failure. If the system is in component A, and there is a failure in component B, then a system loss will result when A fails, since there are no operating components left. If the system is using component A and it suffers an uncovered failure, by definition the system does not successfully recover, so a system loss results. This situation is shown in the Markov model in Figure 4-5.

Figure 4-5. Parallel System Markov Model with Imperfect Coverage



States 4, 5, and 6 represent system failure states, and states 2 and 3 represent system operation states where one component is failed. The coverage parameter is c. The transition into state 2 represents the covered failure of component A while the transition into state 4 represents the uncovered failure of component A. The state equations for this Markov model are

$$dP_1(t)/dt = -(\lambda_A + \lambda_B) P_1(t)$$

$$dP_2(t)/dt = c\lambda_A P_1(t) - l_B P_2(t)$$

$$dP_3(t)/dt = \lambda_B P_1(t) - \lambda_A P_3(t) \qquad \text{[Eq. 4-7]}$$

$$dP_4(t)/dt = (1 - c)\lambda_A P_1(t)$$

$$dP_5(t)/dt = \lambda_B P_2(t)$$

$$dP_6(t)/dt = \lambda_A P_3(t)$$

If λ_A, λ_B, and c are constant in time and the initial condition is

$$P(0) = [1\ 0\ 0,\ 0\ 0\ 0]^t,$$

then the solution is

$$P_1(t) = e\text{-}(\lambda_A + \lambda_B)t$$

$$P_2(t) = c[e\text{-}\lambda_B t - e\text{-}(\lambda_A + \lambda_B)t]$$

$$P_3(t) = \text{e-}\lambda_A t - e\text{-}(\lambda_A + \lambda_B)t \qquad \text{[Eq. 4-8]}$$

$$P_4(t) = (1 - c)\lambda_A [1 - e\text{-}(\lambda_A + \lambda_B)t]/(\lambda_A + \lambda_B)$$

$$P_5(t) = c[\lambda_A + \lambda_B(e\text{-}(\lambda_A + \lambda_B)t - (\lambda_A + \lambda_B)\ e\text{-}\lambda_B t]/(\lambda_A + \lambda_B)$$

$$P_6(t) = [\lambda_B + \lambda_A(e\text{-}(\lambda_A + \lambda_B)t - (\lambda_A + \lambda_B)\ e\text{-}\lambda_A t]/(\lambda_A + \lambda_B)$$

Notice that including the effects of imperfect coverage, some of which are sequential in nature, is straightforward in Markov models. The coverage values may come from engineering experience or may be the result of Markov models that examine the fault detection, identification, and reconfiguration process used to obtain coverage. While a system is still in the design stage, coverage values are often taken as constants. As the design progresses, it may be possible to refine the coverage values through modeling. The issue of coverage modeling is discussed further in reference [10].

TWO-COMPONENT PARALLEL SYSTEM WITH REPAIRS

Once again, the architecture of a two-component parallel system (Figure 4-3) will be used. In this example, repairs are added to the description of the system's operation. Hence, there are transitions from this state to states at higher failure levels representing failure events, and there are transitions back to lower failure levels representing the repair of components. It is assumed that there are two repairmen in this system. This enables the repair of two components at their respective repair rates at the second failure level. Other repair strategies could also be modeled. Notice that repairs are a highly order-dependent phenomenon: a failure must occur before a repair can be performed.

This system is shown in the Markov model in Figure 4-6. The failure rates λ_A and λ_B and the repair rates are μ_A and μ_B. The state equations are formed as before; the change in state probability is equal to the flows into and out of that state:

$$dP_1(t)/dt = -(\lambda_A + \lambda_B) P_1(t) + \mu_A P_2(t) + \mu_B P_3(t)$$

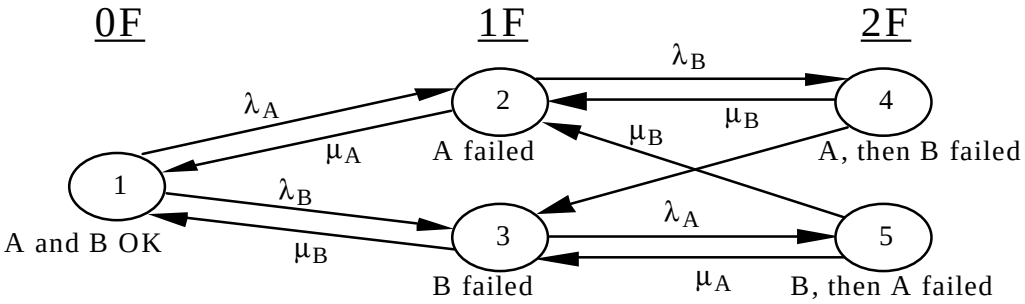
$$dP_2(t)/dt = \lambda_A P_1(t) - (\mu_A + \lambda_B) P_2(t) + \mu_B P_4(t) + \mu_B P_5(t)$$

$$dP_3(t)/dt = \lambda_B P_1(t) - (\lambda_A + \mu_B) P_3(t) + \mu_A P_4(t) + \mu_A P_5(t) \qquad \text{[Eq. 4-9]}$$

$$dP_4(t)/dt = \lambda_B P_2(t) - (\mu_A + \mu_B) P_4(t)$$

$$dP_5(t)/dt = \lambda_A P_3(t) - (\mu_A + \mu_B) P_5(t)$$

Figure 4-6. Parallel System Markov Model with Repairs



If the initial condition is

$$P(0) = [1\ 0,\ 0\ 0\ 0]^T$$

and if all transition rates are constant in time, then a closed form solution can be found. However, the form of this solution is quite complex. For systems with this level of complexity, numerical integration solutions are easier and less expensive to obtain than closed-form analytical solutions.

A key point to notice is that constant repair rates were used. As is true for constant failure rates, these imply that the repair times are exponentially distributed in time with a mean time equal to the inverse of the repair rate. During a system design, this approximation may be adequate. As the design progresses, more information may be available to refine the repair distribution. These other distributions will result in time-varying repair rates, which the numerical solution of the Markov model can easily handle.

STATE SPACE-REDUCTION TECHNIQUES FOR MARKOV MODELS

The Need for State Space Reduction

When real-world fault-tolerant systems are analyzed, the state space is so large that some form of state space reduction is needed to make the analysis tractable. Consider a system with 20 components. Furthermore, assume that the order of component failure does not impact the system performance. Hence, each state is unique in that a specific list of components is failed; the order of these failures is not unique.

At the zero failure level there is one state—no components have failed. At the first failure level, there are 20 states representing the single failure of each of the 20 components. Each state at the first failure level has 19 exit transitions representing the failure of the remaining 19 components. Therefore, at the second failure level, there are potentially 20×19 states. Removing the pairs that have the same 2 components failed, but in different orders, eliminates half of the states. Thus, at the second failure level, there are 190 states describing the 190 combinations of dual failures (20×19/2!). This pattern continues with 1,140 states at the third failure level (20×19×18/3!), 4,845 states at the fourth failure level (20×19×18×17/4!), out to the 20th failure level where there is one state representing all components failed. The total number of states is about 10^6.

Although 20 components is not a very large system to analyze, some form of state space reduction is needed. Notice that the size of the state space grows exponentially with the number of components in the system. For example, a three-component system has 8 states, a four-component system has 16 states, a five-component system has 32 states, etc. In general, a system with n components has 2^n states if failure order is irrelevant.

It is clear that this problem of state space size explosion is a serious limitation on Markov models. In the following subsections, methods for controlling the size of the state space are discussed. One method of reducing the state space, mentioned in an earlier section, the detailed processes of fault detection, identification, and reconfiguration are not explicitly modeled, but rather, a single parameter, the coverage value c, is used to capture the performance of the fault-handling process. This behavioral decomposition [11] reduces the number of states by capturing the consequences of failures, namely the system recovers or does not, and relegates the details of this operation to a separate fault-handling model. In the following subsections, two other techniques for controlling the state space size are discussed.

Exact State Aggregation

In order to control the state space size of a Markov model, exact state aggregation is introduced. This technique aggregates two states at a common failure level into a single state. If certain conditions are met on the two states' exit transitions, then there are no approximations introduced in the aggregation.

Figure 4-7 shows a piece of a Markov model. The consequences of aggregating states 3 and 4 into one state at the n^{th} failure level are examined. Although there may be more transitions (indicated by dashed lines) into and out of states 1, 2, 5, and 6 at failure levels (n-1) and (n+1), respectively, they will not impact the aggregation since they do not interact directly with the states to be aggregated. However, all transitions into and out of the two states to be aggregated are depicted in this figure. This example easily extends to situations where there are more entering and exiting transitions for the states to be aggregated. The state equations for the model in Figure 4-7 are shown with the extraneous transitions to and from states 1, 2, 5, and 6 omitted

$$dP_1(t)/dt = -(a + b)\, P_1(t)$$

$$dP_2(t)/dt = -(c + d)\, P_2(t)$$

$$dP_3(t)/dt = a\, P_1(t) + c\, P_2(t) - (w + x)\, P_3(t) \qquad \text{[Eq. 4-10]}$$

$$dP_4(t)/dt = b\, P_1(t) + d\, P_2(t) - (y + z)\, P_4(t)$$

$$dP_5(t)/dt = w\, P_3(t) + y\, P_4(t)$$

$$dP_6(t)/dt = x\, P_3(t) + z\, P_4(t)$$

Figure 4-7. System Before Aggregation

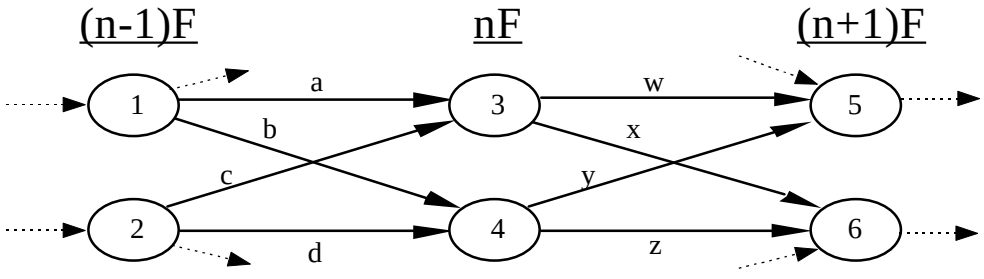


Figure 4-8 shows the same portion of the Markov model but the two states at the n^{th} failure level have been aggregated into one state (state a.3). States in the aggregated model are denoted with an "a" in their designations. The state equations

for the aggregated system, once again ignoring the extraneous transitions at the (n-1) and (n+1) failure levels are:

$$dP_{a.1}(t)/dt = - \alpha P_{a.1}(t)$$

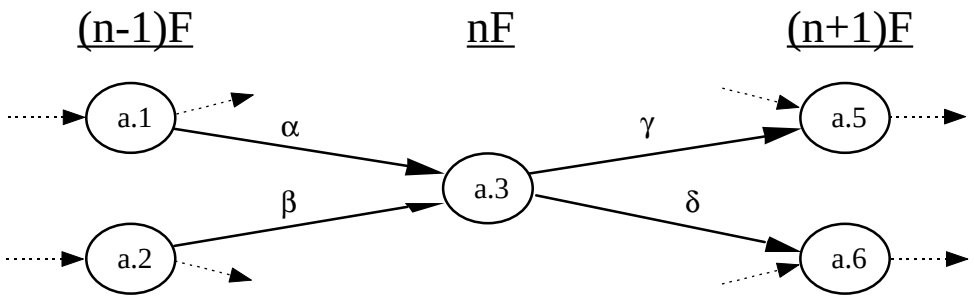
$$dP_{a.2}(t)/dt = - \beta P_{a.2}(t)$$

$$dP_{a.3}(t)/dt = \alpha P_{a.1}(t) + \beta P_{a.2}(t) - (\gamma + \delta) P_{a.3}(t) \qquad \text{[Eq. 4-11]}$$

$$dP_{a.5}(t)/dt = \gamma P_{a.3}(t)$$

$$dP_{a.6}(t)/dt = \delta P_{a.3}(t)$$

Figure 4-8. System After Aggregation



The goal is to determine what the transition rates in the aggregated model, α, β, γ, and δ are. To do this, the following state probabilities are set equal:

- $P_1(t) = P_{a.1}(t)$
- $P_2(t) = P_{a.2}(t)$
- $P_3(t) + P_4(t) = P_{a.3}(t)$
- $P_5(t) = P_{a.5}(t)$
- $P_6(t) = P_{a.6}(t)$.

These equalities lead directly to the equality of the differentials:

- $dP_1(t)/dt = dP_{a.1}(t)/dt$
- $dP_2(t)/dt = dP_{a.2}(t)/dt$
- $dP_3(t)/dt + dP_4(t)/dt = dP_{a.3}(t)/dt$

- $dP_5(t)/dt = dP_{a.5}(t)/dt$
- $dP_6(t)/dt = dP_{a.6}(t)/dt$.

In other words, all states have a one-to-one correspondence in the two models except states 3 and 4, which are aggregated into state a.3.

Applying the equalities for the differentials for states 1, 2, 5, and 6 (Equation 4-10) and their corresponding states in the aggregated model (Equation 4-11) gives the following:

- (state 1) $\alpha = a + b$
- (state 2) $\beta = c + d$
- (state 5) $\gamma = [w\,P_3(t) + y\,P_4(t)]/[P_3(t) + P_4(t)]$
- (state 6) $\delta = [x\,P_3(t) + z\,P_4(t)]/[P_3(t) + P_4(t)]$

These equations also satisfy the equalities for the aggregated state a.3. The equations for α and β make intuitive sense: the transition rate into the aggregated state a.3 is the sum of the rates into 3 and 4 since state a.3 is the combination of states 3 and 4. The transition rates γ and δ are slightly more complex.

The flows (i.e., differentials) in the unaggregated and aggregated models must be the same. Flows are the product of transition rates and the probability of the state at the origin of the transition. Hence, the transition rate leaving the aggregated state a.3, going to state a.5, must be the sum of the flows leaving states 3 and 4, going to state 5, divided by the probability of being in state a.3 (which equals the sum of states 3 and 4's probability). In other words, the exit transition from an aggregated state is a "mix" of the unaggregated states' exit transitions. Notice that in general, the aggregated system's exit rates γ and δ are time varying functions even though the unaggregated system's rates w, x, y, and z may be time invariant.

To perform an exact state aggregation consider a special case of the above system. If w = y and x = z, then

- $\gamma = w = y$
- $\delta = x = z$.

Performing aggregations in this case gives time-invariant transition rates in the aggregated model if the unaggregated model also has time invariant rates. This special case is considered exact since no approximations are introduced and the aggregated transition rates can be found by inspection: rates into the aggregated state are the sum of the rates into the unaggregated states, and the exit rates for the

aggregated state are the same as those to each destination as from the unaggregated states. Notice that a sufficient condition for exact aggregation is that the two states to be aggregated must be at the same failure level and have identical transition rates with one another to each of their destinations. This aggregation applies for any initial conditions on the unaggregated states.

Two examples are now presented to show the technique of exact state aggregation. First, return to the parallel system with imperfect coverage of Figure 4-5. States 5 and 6 are at a common failure level and have identical exit transitions (they both have no exit transitions). Hence, these two states can be aggregated. The resulting model is shown in Figure 4-9. The state equations are

$$dP_1(t)/dt = -(\lambda_A + \lambda_B) P_1(t)$$

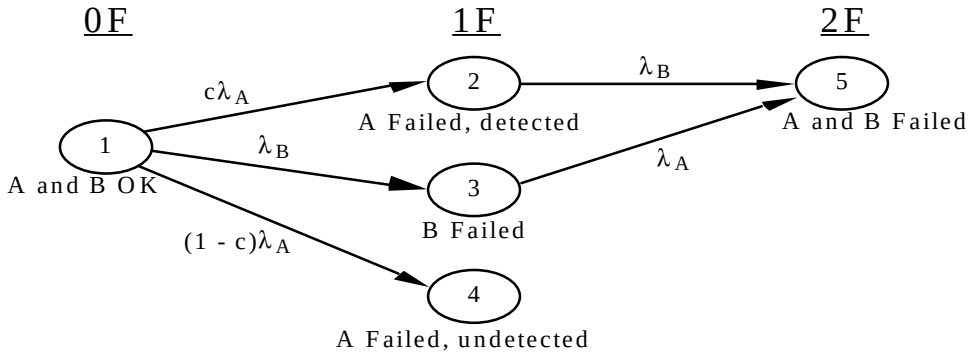
$$dP_2(t)/dt = c\lambda_A P_1(t) - \lambda_B P_2(t)$$

$$dP_3(t)/dt = \lambda_B P_1(t) - \lambda_A P_3(t) \qquad \text{[Eq. 4-12]}$$

$$dP_4(t)/dt = (1 - c)\lambda_A P_1(t)$$

$$dP_5(t)/dt = \lambda_B P_2(t) + \lambda_A P_3(t)$$

Figure 4-9. Aggregated Parallel System with Imperfect Coverage



If λ_A, λ_B, and c are constant in time and the initial condition is

$$P(0) = [1\ 0,\ 0\ 0\ 0]^t,$$

then the solution is:

$$P_1(t) = e^{-(\lambda A + \lambda B)t}$$

$$P_2(t) = c[e^{-\lambda Bt} - e^{-(\lambda A + \lambda B)t}]$$

$$P_3(t) = e\text{-}\lambda_A t - e\text{-}(\lambda_A + \lambda_B)t \qquad \text{[Eq. 4-13]}$$

$$P_4(t) = (1 - c)\lambda_A [1 - e\text{-}(\lambda_A + \lambda_B)t]/(\lambda_A + \lambda_B)$$

$$P_5(t) = [c\lambda_A + \lambda_B + (\lambda_A + c\lambda_B)\, e\text{-}(\lambda_A + \lambda_B)t - (\lambda_A + \lambda_B)\, e\text{-}\lambda_A t - c(\lambda_A + \lambda_B)\, e\text{-}\lambda_B t]$$
$$/(\lambda_A + \lambda_B)$$

Comparing this solution with that of the unaggregated case (Equation 4-8) shows the result to be identical, except that the aggregated state 5 is the sum of the unaggregated states 5 and 6.

As a second example, consider the parallel system with repairs (Figure 4-6). States 4 and 5 have identical exit transition rates to each of their destinations; both have rate μ_A going to state 3 and rate μ_B going to state 2. Hence, states 4 and 5 can be aggregated. The resulting model is shown in Figure 4-10 and the state equations are as follows:

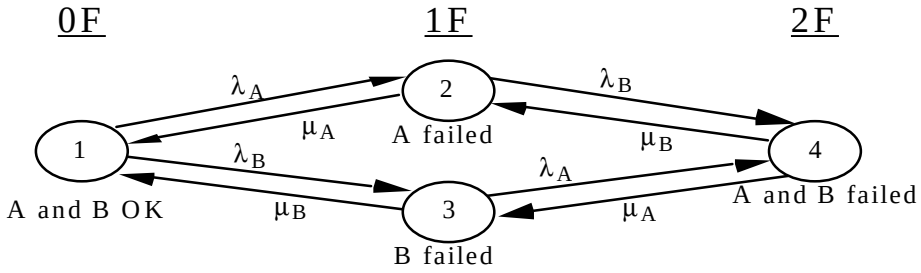
$$dP_1(t)/dt = -(\lambda_A + \lambda_B)\, P_1(t) + \mu_A\, P_2(t) + \mu_B\, P_3(t)$$

$$dP_2(t)/dt = \lambda_A\, P_1(t) - (\mu_A + \lambda_B)\, P_2(t) + \mu_B\, P_4(t) \qquad \text{[Eq. 4-14]}$$

$$dP_3(t)/dt = \lambda_B\, P_1(t) - (\lambda_A + \mu_B)\, P_3(t) + \mu_A\, P_4(t)$$

$$dP_4(t)/dt = \lambda_B\, P_2(t) + \lambda_A\, P_3(t) - (\mu_A + \mu_B)\, P_4(t)$$

Figure 4-10. Aggregated Parallel System Markov Model with Repairs



As was the case before the aggregation was performed, these equations are quite complex to solve in closed form. A numerical solution shows correspondence between the unaggregated and aggregated solutions.

Notice that the process of state aggregation reduces the level of detail of the model. What were two unique states are now mixed together as one. As long as the states to be aggregated share common characteristics that are of interest to the

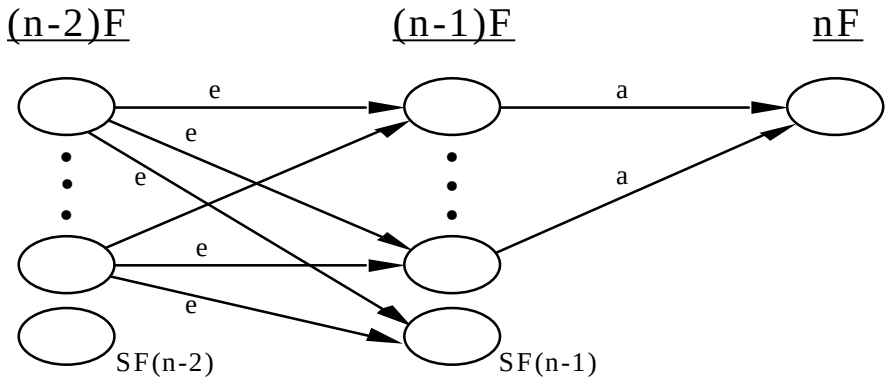
modeler, such as common operating modes, this loss of detail is usually not critical.

Model Truncation

Another method of controlling the size of the state space is model truncation. Frequently, in highly reliable systems, the probability of system failure caused by many components being failed is much less likely than system failures caused by a few specific component failures. Hence, only the most probable failure sequences impact the solution and only these sequences need to be modeled in detail. Although the accuracy of the solution is impacted by the unmodeled failure sequences, bounds on the solution's accuracy will be directly obtained.

Figure 4-11 shows a Markov model that has been truncated at the no failure level. Thus, any configuration with n or more failures is called a system failure and all of these states are lumped into the state SF(n). Clearly, this will lead to an overestimate of the system failure probability for the n^{th} and greater failure levels since some of these configurations are operational. Therefore, the state SF(n) will be used in establishing an upper bound on the system failure probability.

Figure 4-11. Truncated Markov Model



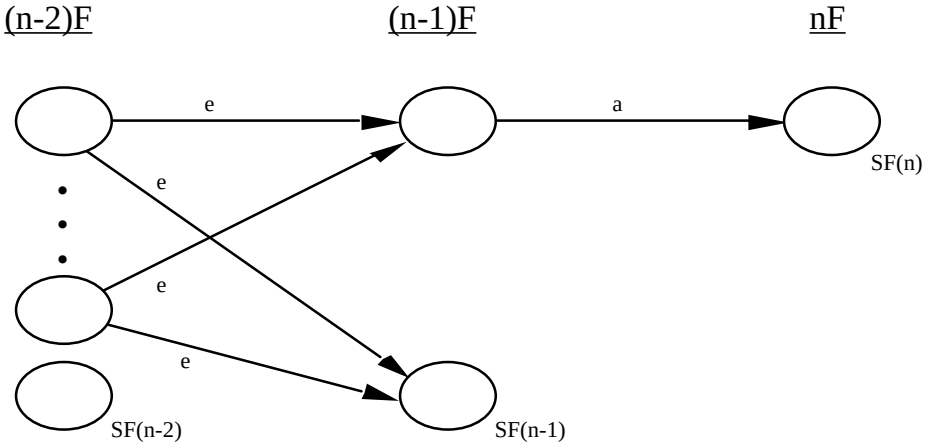
States SF(n-1) and SF(n-2) are the contributions to the system failure probability at the (n-1) and (n-2) failure levels, respectively. The "e" notation on the transitions leading to states at the (n-1) failure level indicate that these should be exact transitions. This results in an exact flow into states at the (n-1) failure level. Thus, the probability of the system failure state SF(n-1) is exact. The "a" transition entering the failure state at the no failure level may be an approximation that is greater than, or equal to, the actual transition rate. This results in a larger probability flowing into state SF(n) than would be seen in an exact model. This, in addition to the assumption that all n and greater failure configurations cause a system failure, ensures that state SF(n) contains a higher probability than exists in an exact model.

The sum of all failure state probabilities up to and including state SF(n-1) provides a lower bound on the system failure probability. This is valid since these states have probabilities equal to the exact case, and the failure contributions at failure levels n and greater have not been included. The upper bound is obtained by adding system failure state SF(n) to the lower bound probability. This provides an upper bound since flows into state SF(n) are equal to, or greater than, those in an exact model and it is an overestimate of the true system failure probability at the n and greater failure levels.

The tightness of these bounds is measured by their difference, SF(n). If this approximation is too large for the given problem, then the model must be constructed out to the next failure level. For most highly reliable, integrated control systems, three to five failure levels provides sufficient accuracy. Given the example of a 20-component system needing 10^6 states to model all 20 failure levels, truncating the model at the third failure level requires 212 states. Hence, there is a substantial savings in using model truncation.

Returning to the truncated model in Figure 4-10, a further reduction in the size of the state space can be made. The transition rates from the operational states at the (n-1) failure level to the SF(n) state are allowed to be approximations. Therefore, it is convenient to set all of the "a" rates equal. The rates "a" must be equal to, or greater than, the exact rates. For example, they may be set to the maximum of the exact rates or to the sum of all the component rates. The latter case may be interpreted as assuming that any component can fail at the (n-1) failure level, even components that have previously failed. This ensures that the rates "a" are identical and are greater than the exact exit transition rates. Since the operational states at the (n-1) failure level all have the same exit transition rates and they lead to the same destination, they can be exactly aggregated into states with common operational modes. This aggregation is shown in Figure 4-12.

Figure 4-12. Truncated Markov Model with Aggregation



The ability to aggregate states at the (n-1) failure level provides a further reduction in the size of the state space. For example, the 20-component system that requires 10^6 states without truncation, and 212 states when truncated at the third failure level, takes only 24 states when the second-failure-level states are aggregated into a single-system-failure state and a single operational state.

NUMERICAL SOLUTIONS OF MARKOV MODELS

As has been seen, moderate-size, fault-tolerant systems (for example, with 20 components) generate reliability models with approximately 20 to 100 states depending on how the model is truncated and aggregated, how many operating modes are present, and whether or not the components have imperfect coverage. Given models of this size or larger, it is impractical to obtain closed form solutions. The most efficient means of obtaining results is a numerical solution of the Markov model.

As has been seen in previous sections, the Markov model represents a system of differential equations. For example, the parallel system in Figure 4-3 has the state equations

- $dP_1(t)/dt = ((\lambda_A + \lambda_B)\, P_1(t)$
- $dP_2(t)/dt = \lambda_A\, P_1(t) - \lambda_B\, P_2(t)$
- $dP_3(t)/dt = \lambda_B\, P_1(t) - \lambda_A\, P_3(t)$
- $dP_4(t)/dt = \lambda_B\, P_2(t)$
- $dP_5(t)/dt = \lambda_A\, P_3(t)$

These can be written in matrix form:

$$\frac{d\mathbf{P}(t)}{dt} = \begin{bmatrix} -(\lambda_A + \lambda_B) & 0 & 0 & 0 & 0 \\ \lambda_A & -\lambda_B & 0 & 0 & 0 \\ \lambda_B & 0 & -\lambda_A & 0 & 0 \\ 0 & \lambda_B & 0 & 0 & 0 \\ 0 & 0 & \lambda_A & 0 & 0 \end{bmatrix} \mathbf{P}(t)$$

where the state vector is

$$P(t) = [P_1(t), P_2(t), P_3(t), P_4(t), P_5(t)]^t$$

Notice that the columns add to zero. This represents a conservation property of the system: all flows leaving a state must enter another state, and transitions do not store any flow. The matrix equation may be written more concisely as

$$dP(t)/dt = A\,P(t) \qquad \text{[Eq. 4-15]}$$

Equation 4-15 is the continuous-time representation of the Markov model. Matrix A is the continuous-time transition Matrix. While there are many ways of numerically integrating this equation, one that is particularly fast and accurate will be shown [12]. To begin, the continuous differential is approximated with a discrete time step:

$$[P(t+\Delta t) - P(t)]/\Delta t = A\,P(t)$$

$$P(t+\Delta t) = [I + A\,\Delta t]\,P(t)$$

$$P(t+\Delta t) = M\,P(t) \qquad \text{[Eq. 4-16]}$$

Matrix I is the identity matrix and M is the discrete-time transition matrix. The use of the above approximation is called Euler integration.

Equation 4-16 represents an iterative solution for the Markov model. Given the system's initial condition, $P(0)$, it is possible to use Equation 4-16 to propagate the state probability in time:

$$\begin{gathered} P(\Delta t) = M\,P(0) \\ P(2\Delta t) = M\,P(\Delta t) \\ P(3\Delta t) = M\,P(2\Delta t) \\ P(4\Delta t) = M\,P(3\Delta t) \\ P(5\Delta t) = M\,P(4\Delta t) \\ \vdots \\ P(n\Delta t) = \boldsymbol{M}\,\boldsymbol{P}((n-1)\Delta t) \end{gathered} \qquad \text{[Eq. 4-17]}$$

While this method is sufficiently stable for appropriate Δts, it can be rewritten in a form that is faster to solve. If the state transition matrix M is constant in time (i.e., failure rates are constant in time) then the following observation can be made:

- $P(\Delta t) = M\,P(0)$
- $P(2\Delta t) = M\,P(\Delta t) = M\,M\,P(0) = M^2\,P(0)$
- $P(3\Delta t) = M\,P(2\Delta t) = M\,M\,M\,P(0) = M^3\,P(0)$
- etc.

In general, the state probability at time t = m Δt is

$$P(m\Delta t) = \mathbf{M}^m \mathbf{P}(0) \qquad \text{[Eq. 4-18]}$$

It is efficient to represent m as a binary number. The digits of this number represent a recipe for the powers of 2 required to obtain M^m. For example, if m = 13, its binary representation is 1101. Therefore, M^{13} can be found by performing two steps. First, calculate M^2, M^4, and M^8 by repeated squaring of matrix M. Second, use the binary representation of 13 to determine which powers of M need to be used. In this case, it is the 8^{th}, 4^{th}, and 1^{st} powers that are needed. Performing these indicated multiplications gives the result desired:

$$\mathbf{M}^{13} = \mathbf{M}^8 \mathbf{M}^4 \mathbf{M}$$

To get a feel for the increase in speed obtained by this method of matrix doubling (Equation 4-18) over that of traditional stepping in time (Equation 4-17), consider a problem where the state vector dimension is n and the number of time steps required is m. For a matrix of dimension n by m, the process of squaring the matrix takes on the order of n^3 operations. The process of multiplying a matrix by a vector takes on the order of n^2 operations. Hence, traditional stepping in time requires ($m\ n^2$) operations to achieve a result. If m is an integer power of 2, then z matrix doubles are required to obtain M^m, where z = integer[ln(m)/ln(2)]. Usually, m is not an integer power of 2 so powers of M must be multiplied to obtain M^m. On average, this means that 1.5z matrix multipliers are needed. Therefore, the matrix doubling method requires ($1.5\ z\ n^3$) operations to get a solution.

Consider an example where the state vector dimension is n = 50. For a mission time of t = 1,000 hours and a time step of Δt = 1 minute and m = 60,000 time steps. Thus, z = 15. The number of operations for stepping in time (Equation 4-17) is $1.5x10^8$ and the number of operations (on average) for matrix doubling (Equation 4-18) is $2.8x10^6$. For this case, stepping in time requires more than 50 times the number of operations required for matrix doubling. Therefore, when implemented on a computer, matrix doubling would be expected to be 50 times faster than stepping in time. A further improvement in speed can be obtained by selecting a time step that gives an m that is an integer power of 2. For example, Δt = 0.915 minutes gives $m = 2^{16}$. Therefore, z = 16, and only 16 matrix multiplications are needed to obtain a result. The total operations for matrix doubling in this case would be $2x10^6$. This would reduce computation time further by a factor of 1.4.

One final note on integrating the state equations of a Markov model is the need for double-precision calculations. Notice that on the diagonals of the discrete-time transition matrix are terms like

$$1 - \lambda\Delta t$$

where λ is a failure rate. Typical values for λ and Δt are $\lambda = 10^{-5}$ hr^{-1} and $\Delta t = 10^{-2}$ hr. Thus, the diagonal entry is

$$1 - 10^{-7}$$

Using a machine with single precision limits the accuracy of a number to approximately seven significant digits. Hence, the diagonal entry would be stored as a 1. Squaring the matrix would maintain this 1 on the diagonal for all time. This means that the state associated with this diagonal would never have flows leaving it, even though flows would be appearing in states "downstream" of it. This violates the conservation of probability required for Markov models. Although the solution can be performed in single precision, some subtle renormalization procedures are required to maintain the conservation of probability. Performing the integration in double precision avoids this problem.

References

[1] "TIGER Computer Program," Naval Sea Systems Command, Washington, D. C.: 1985.

[2] R. Y. Rubinstein, *Simulation and the Monte Carlo Method*, J. Wiley, Inc., New York: 1981.

[3] M. Shooman, *Probabilistic Reliability: An Engineering Approach*, McGraw-Hill, New York: 1968.

[4] W. E. Vesely, et al., *Fault Tree Handbook*, NUREG-0492, Office of Nuclear Regulatory Research, U. S. Nuclear Regulatory Commission, Washington, D. C.: January 1981.

[5] R. A. Howard, *Dynamic Probabilistic Systems*, J. Wiley, Inc., New York: 1971.

[6] J. G. Kemeny and J. L. Snell, *Finite Markov Chains*, Springer-Verlag, New York: 1976.

[7] E. Gai, J. V. Harrison, and R. H. Luppold, "Reliability Analysis of a Dual Redundant Engine Controller," *IEEE Transactions on Reliability*, Vol. R-32, April 1983.

[8] R. S. Schabowsky, Jr., and W. W. Weinstein, "On the Evaluation and Validation of Fault-Tolerant Digital Control Systems," *Proceedings of the EPRI Seminar: Power Plant Digital Control and Fault-Tolerant Microcomputers*, Scottsdale, Arizona: April 9–12, 1985.

[9] P. S. Babcock, R. S. Schabowsky, Jr., and J. V. Harrison, *Reliability Modeling of the V2500 Electronic Engine Controller (EEC-150)*, CSDL-R-1847, The C. S. Draper Laboratory, Inc., Cambridge, MA: December 1985.

[10] R. M. Geist, K. S. Trivedi, J. B. Dugan, and M. K. Smotherman, "Design of the Hybrid Automated Reliability Predictor," *Proceedings of 5th IEEE/AIAA Digital Avionics Systems Conference*: November 1983.

[11] K. S. Trivedi, J. B. Dugan, R. M. Geist, and M. K Smotherman, "Modeling Imperfect Coverage in Fault-Tolerant Systems," *Proceedings IEEE FTCS-14*, 1984.

[12] C. Moler and C. van Loan, "Nineteen Dubious Ways to Compute the Exponential of a Matrix," *SIAM Review*, Vol. 20, No. 4, October 1978.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	April 1998	Contractor Report

4. TITLE AND SUBTITLE
Reliability Modeling Methodology for Independence Approaches on Parallel Runways Safety Analysis

5. FUNDING NUMBERS
C NAS2-14361
Task 96-05

WU 538-04-14-02

6. AUTHOR(S)
P. Babcock, A. Schor, and G. Rosch

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Logisitics Management Institute
2000 Corporation Ridge
McLean, Virginia 22102-7805

8. PERFORMING ORGANIZATION REPORT NUMBER
NS605S2

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Langley Research Center
Hampton, VA 23681-0001

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
NASA/CR-1998-207660

11. SUPPLEMENTARY NOTES
Langley Technical Monitor: Robert E. Yackovetsky
Final Report

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Unclassified - Unlimited

Subject Category 01 Distribution: Nonstandard
Availability: CASI (301) 621-0390

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This document is an adjunct to the final report An Integrated Safety Analysis Methodology for Emerging Air Transport Technologies. That report presents the results of our analysis of the problem of simultaneous but independent, approaches of two aircraft on parallel runways (independent approaches on parallel runways, or IAPR). This introductory chapter presents a brief overview and perspective of approaches and methodologies for performing safety analyses for complex systems. Ensuing chapter provide the technical details that underlie the approach that we have taken in performing the safety analysis for the IAPR concept.

14. SUBJECT TERMS
air traffic management reliability
safety
air transportation

15. NUMBER OF PAGES
48

16. PRICE CODE
A03

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