SAFETY ANALYSIS OF ARES

by

William W. Weinstein

Philip S. Babcock IV

Frank Leong

October 1987

The Charles Stark Draper Laboratory, Inc.

Cambridge, Massachusetts 02139

TABLE OF CONTENTS

INTRODUCTION

In February of 1986 the Burlington Northern Railroad (BN) contracted with The Charles Stark Draper Laboratory (CSDL) to perform a study to determine how the safety of BN operations under the proposed ARES train control system would compare with the safety of operations under the current set of train control systems. This represented a novel undertaking for two reasons: such a safety prediction had, to the best of our knowledge, never before been applied to railroad operations; and the analytical tools (Markov modeling) had not previously been used to analyze a system with such a large number of components, some of which were human operators.

Predicting safety for a system like ARES requires the development of a mathematical model of the system. This model inputs statistical information about the reliability (failure rates and error rates) of the various system components and computes a predicted accident rate for the system. This model can also be used to determine the sensitivity of the predicted system accident rate to changes in the reliability of system components.

This report presents the results of this study along with a detailed description of the ARES models and the modeling methodology. The analysis indicates that, under ARES, BN should experience almost two orders of magnitude fewer control system related accidents than it does under the control systems currently in use. Also, this improvement appears to hold under a variety of assumptions used to generate various input parameters to the model.

The report is divided into an executive summary and a detailed description of the analysis. CSDL Report R-1899 (Reference 1) provides a detailed tutorial on Markov modeling.

- The *Executive Summary* discusses the goals and the general methodology of the study and presents a summary of results together with the qualitative rationale for why, from a safety viewpoint, ARES behaves as it does.
- *Section 1* addresses the selection of an appropriate modeling technique.
- *Section 2* briefly discusses the qualitative reliability characteristics of the current train control systems.
- *Section 3* describes the ARES system and its operation in detail, including the characteristics of failure modes that result in accidents.
- *Section 4* discusses the mathematical techniques used to generate the ARES model. Section 4.1 provides a brief introduction to Markov modeling. Section 4.2 provides a mathematical justification for the model decomposition techniques that were used — it can be skipped without affecting the reader's

qualitative understanding of the model.

- *Section 5* presents each of the 14 submodels that make up the complete ARES model and discusses the component reliabilities and repair rates that are used as inputs. The results are presented and sensitivity of these results to modeling assumptions and uncertainties in the inputs is discussed.
- *Section 6* provides a brief summary.

PART 1

EXECUTIVE SUMMARY

GOALS OF THE STUDY

The main purpose of this study is to ascertain the relative safety of the Advanced Railroad Electronics System (ARES) compared to conventional train control systems. There are several reasons to do this. The foremost is to provide evidence to the Federal Railroad Administration that the introduction of ARES will not negatively impact railroad safety. Since the FRA is entrusted with the responsibility of insuring safe railroad operation, it could not approve any new approach to train control that was not at least as safe as what is currently in operation, regardless of any other benefits to be gained by that new approach. Conversely, if the new approach is likely to provide significant safety improvements, then the FRA has reason to support its introduction for the safety benefits alone. The second reason is to demonstrate to railroad operations personnel (engineers, conductors, dispatchers, etc.) that ARES will not increase their own risk on-the-job. This concern is the primary argument against the acceptance of innovative technologies. The third reason is to enable the assessment of the total economic benefit of implementing ARES. Savings result not only from the improved efficiency of operation but also from the fact that fewer accidents mean lower losses due to equipment damage and related lawsuits.

A side benefit of this analysis is the demonstration of an effective methodology and tools to do safety prediction for advanced control systems like ARES. ARES and similar advanced systems will require a regulatory approach that is broader than the function-based component specifications now in use—an approach that will require the analytical verification of system safety.

The current approach to control system safety regulation addresses the signalling system hardware and the human operator separately. The signals provide local information that the engineer uses to effect safe movement of a train. The vitality of hardware is provided locally, by small groups of logically simple components which are interconnected in a straightforward, fail-safe manner. The large body of accumulated knowledge about the failure characteristics of these types of components means that the desired levels of component reliability and fail-safe functionality can be effected by the specification of physical and functional design characteristics for the signalling hardware. On the other side of the coin, the operator is required to adhere to the signal information in order to close the train control loop safely: There are various procedures that the operator must follow, and a built-in assumption of safe system operation is that the operator follows these procedures properly.

Because the fundamental premises of the ARES approach to train control differ from those of the current approach, a more suitable structure for control system regulations ought to be considered. In ARES:

1. Vitality is not provided locally, but rather by the control system as a whole. The reliability of any particular component has meaning (with regard to system safety) only insofar as it relates to the architecture of the system and the reliabilities of all other components in the system.

2. The operator and the system hardware function in parallel — they have overlapping responsibilities — so functional specifications for the system and procedures for the human operator are not independent as they are in the current control approach.

3. The components themselves (radios, computers, and various sensors and actuators) are more complex than the types of components now in use. Regulation of components by design specification would impede the incorporation of new technologies intended to improve the reliability and functionality of these components.

4. Specification of a system architecture and the functional specification for its components would impede the incorporation of any architectural improvements or alternatives which might be more suited to a particular railroad and would put the FRA in the position of acting as a system designer.

An extension of the regulatory environment to encompass advanced control systems would employ an appropriate measure of objective-based regulations, with function– and design–based regulations applied where they make sense. A primary safety objective would be a maximum allowable accident rate over a given territory as a function of train density and other parameters of the territory. Objective-based safety regulations are employed by the FAA, NRC, DoD and NASA to deal with systems under their purview. The safety of these large and complex systems must be demonstrated analytically. Thus there are numerous precedents for demonstrating system reliability/safety in this manner.

The reliability requirements for control systems such as those used in military aircraft, nuclear power plants, etc., are expressed as the probability of critical system failure during a specified time frame. (This can also be expressed as failures per unit time.) Measuring this failure rate requires that a statistically significant number of *system* failures be observed. Since these control systems are very reliable, it would take an unreasonably long time to gather statistics by observing one or two systems. An alternative is to build many systems to reduce the observation time, but this is impractical if only a limited number of systems are going to be produced. In order to obtain confidence that a system will demonstrate the desired reliability prior to the time it will be put into operation, analytical models are employed. These models describe the system in terms of the relationships between sub-components — items for which empirical failure rate measurements can be made in a practical manner.

The FAA sets failure rate requirements on certain life critical aircraft systems. These systems employ redundancy to achieve a very high level of reliability. The FAA accepts the use of analytical methods to certify that these systems will meet the specified reliability levels. The NRC goes a step further and specifies a particular analysis methodology for the demonstration of nuclear plant safety.[1]

As part of a system certification procedure, the FRA will need to specify a safety analysis methodology that is applicable, in a consistent way, to various control system architectures that employ different components. This study shows that such a methodology exists and can be applied to an advanced railroad control system in a meaningful and effective manner.

METHODOLOGY OF THE ANALYSIS

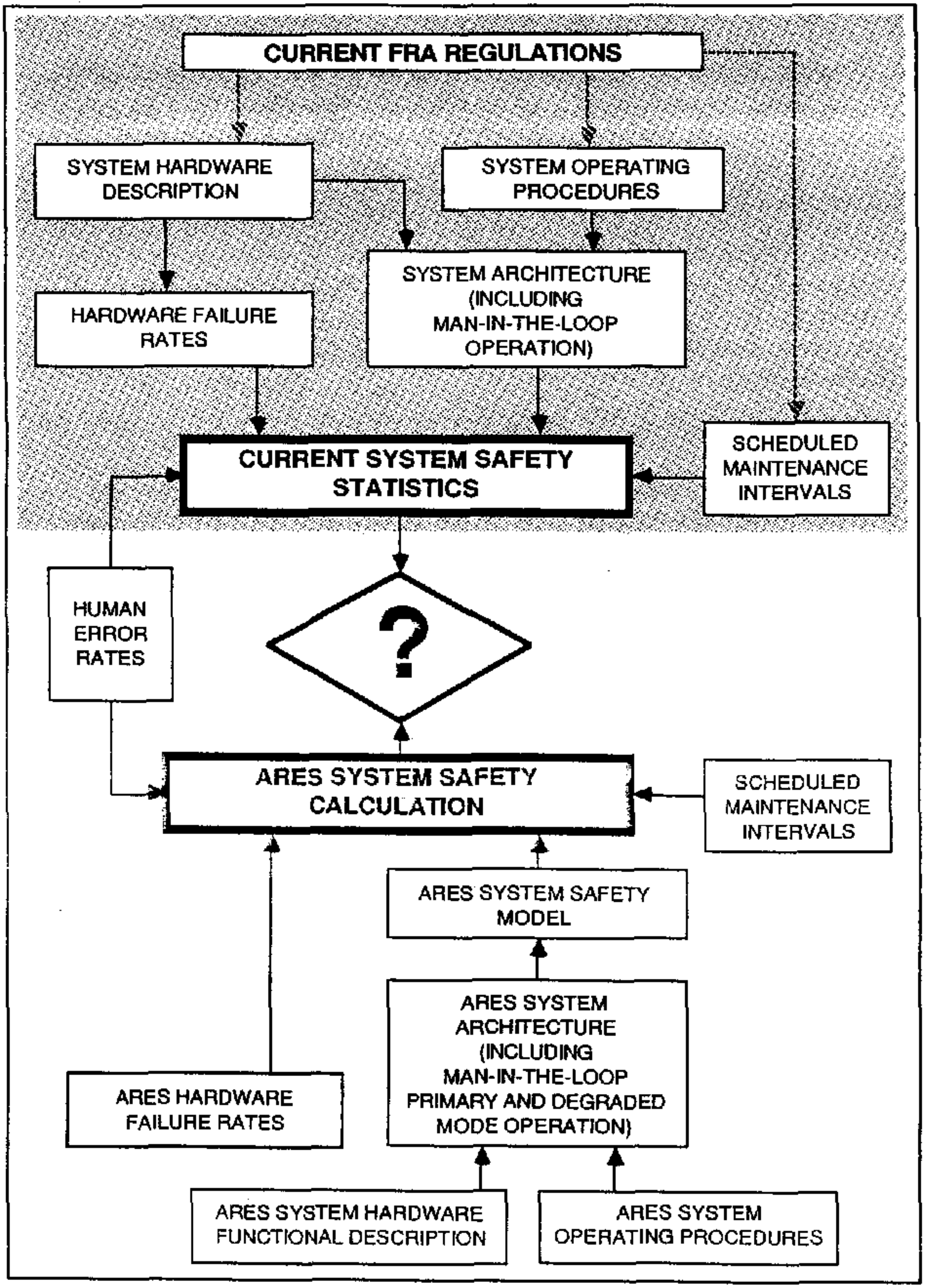
The approach to assessing ARES safety is to compare the actual accident rate attributable to failures in the current train control systems to the predicted accident rate under full ARES operation. The current accident rate data was extracted from BN accident statistics. The accident rate for ARES was predicted by modeling the effects of hardware failures and human errors within ARES.[2]

Figure S.1 shows the elements of the ARES safety analysis. The shaded box represents the current system accident numbers. (The flow inside the shaded box indicates how the current regulations influence the system safety.) The lower half of the figure depicts the steps in the ARES modeling process. It begins with a functional description of ARES and the ARES operating procedures. Together these define the ARES system architecture. A failure modes and effects analysis is generated from this system architecture, and this in turn serves as the basis of the ARES safety model. Hardware failure rates, human error rates, and maintenance information (equipment repair times and scheduled maintenance intervals) are input to the model to produce a predicted accident rate for ARES. This value can then be compared with the present accident rate.

The current control systems accident data was extracted from the BN scoreboard of reportable incidents for 1984 and 1985, and from the US DOT FRA Office of Safety Accident/Incident Bulletin Numbers 153 (for 1984) and 154 (for 1985). BN personnel indicated which subclasses of reportable incidents were attributable to failures within the control system (Table S.1).

[1] Probabilistic Risk Assessment Procedures Guide (NUREG/CR-2300) and Fault Tree Handbook (NUREG-0492).

[2] This approach addresses the bottom line safety consideration — the measured accident rate of the current system and the predicted accident rate of the ARES system. It compares apples to apples. It is not meaningful to attempt to compare only the hardware of the two systems, because the human *is* part of the control system. Since the architectures of the two systems are fundamentally different, the effects of human error cannot be factored out.


CURRENT FRA REGULATIONS
SYSTEM HARDWARE DESCRIPTION
SYSTEM OPERATING PROCEDURES
HARDWARE FAILURE RATES
SYSTEM ARCHITECTURE (INCLUDING MAN-IN-THE-LOOP OPERATION)
CURRENT SYSTEM SAFETY STATISTICS
SCHEDULED MAINTENANCE INTERVALS
HUMAN ERROR RATES
?
ARES SYSTEM SAFETY CALCULATION
SCHEDULED MAINTENANCE INTERVALS
ARES SYSTEM SAFETY MODEL
ARES SYSTEM ARCHITECTURE (INCLUDING MAN-IN-THE-LOOP PRIMARY AND DEGRADED MODE OPERATION)
ARES HARDWARE FAILURE RATES
ARES SYSTEM HARDWARE FUNCTIONAL DESCRIPTION
ARES SYSTEM OPERATING PROCEDURES

ELEMENTS OF THE ARES SAFETY ANALYSIS
Figure S.1

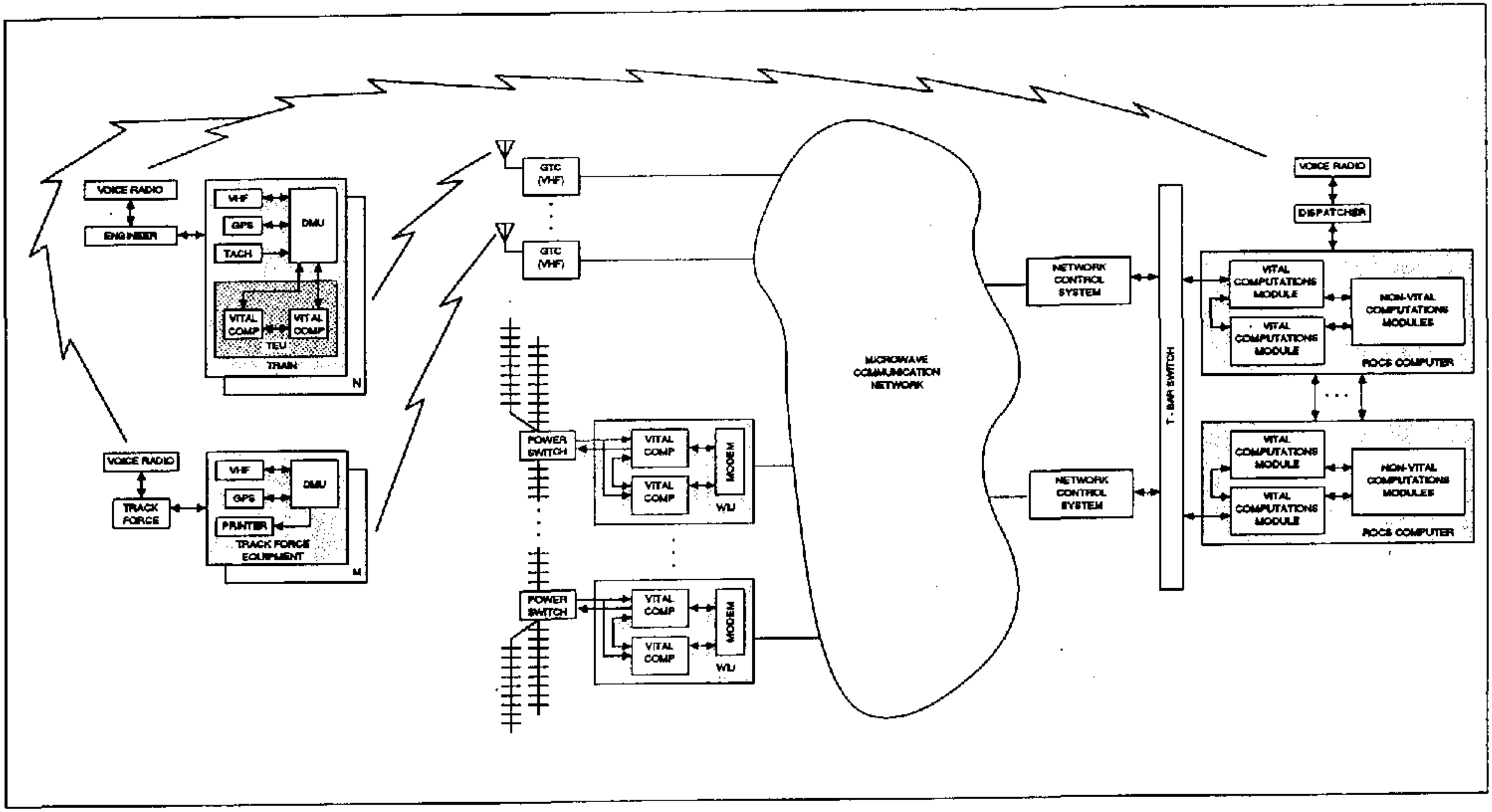
CURRENT CONTROL SYSTEMS ACCIDENT DATA

[BN / National]

	1984	1985
Human Factor Causes		
Buffing–slack		
Buffing or slack action excessive	15/175	13/167
Employee Physical Condition		
Impaired efficiency or judgement – drugs or alcohol	1/2	0/1
Employee falling asleep	1/2	0/2
Human Miscellaneous		
Lateral drawbar force on curve excessive	12/60	7/43
Other rules		
Failure to stop train in clear	4/48	6/45
Instruction to train/yard crew improper	2/9	3/9
Motor car – on track equipment rules, failure to comply	8/27	9/17
Special operating instructions, failure to comply	2/9	0/12
Train order or timetable authority, failure to comply	0/5	3/6
Train orders – error in preparation, transmission or delivery	1/1	0/1
Signalling		
Fixed signal improperly displayed	0/1	1/1
Block signal, failure to comply	1/5	0/5
Speed		
Train inside yard limits, excessive speed	2/15	3/16
Use of Brakes		
Automatic brake, improper use	1/20	0/19
Dynamic brake, improper use	1/5	1/4
Independent brake, improper use	2/10	1/16
Signals and Communication		
Other communication equipment failure	1/2	0/0
TOTAL	54/395	47/364

Notes: Includes all reportables (>$4,500 in 1984, >$4,900 in 1985)
BN data from BN Reportable Incident Scoreboard
National data from US DOT FRA Office of Safety Accident/Incident Bulletin Numbers 153 (for 1984) and 154 (for 1985)

Table S.1



ARES SYSTEM BASELINE

Figure S.2

The development of the ARES safety model began by defining an ARES system baseline for a single BN operating region. The region is the basic train control entity in that all train interactions within the geographical boundaries of the region are specified and monitored within a single regional control center. Inter-regional traffic flow is managed by handoffs from one control center to another, in a manner similar to that used in the air traffic control system. Since regions are basically independent of one another, the total accident rate for the entire BN system is just the sum of the individual accident rates for the various regions.

The ARES baseline shown in Figure S.2 contains just those elements of ARES that deal with train control and thus those that can impact system safety. The components of ARES fall into four categories:

1. Regional control center — There is a single control center per region, comprising a redundant regional operations computer system (ROCS) and a redundant network control system (NCS).

2. Communications network — This consists basically of the microwave backbone currently in place.

3. Wayside equipment — This includes wayside interface units (WIU) that interface with switches and wayside detectors, VHF digital radio sets, and the ground terminal controllers (GTC) that interface the VHF radios to the network.

4. Mobile equipment — This includes the navigation equipment (tachometer and global positioning system (GPS) radio set), VHF digital radio set, token enforcement unit (TEU), and the data management unit (DMU) which ties together the other on-train equipment. A similar set of navigation and communication equipment is used by track forces and inspectors to allow the system to keep abreast of their status.

A region contains large quantities of fixed and mobile ARES equipment. The baseline for a region assumes 100 microwave towers (each with a GTC and a VHF radio), 600 WIUs and an average track occupancy of 100 operating trains. At any given time it can be expected that some of this equipment is in a failed state. An individual component failure is not, in general, safety critical, since cross-checks are done to detect any incorrect information that it may inject into the system. An equipment failure may cause the loss of operational capability, but the system is designed to be aware of that loss, i.e., it is designed to be fail-safe. It generally requires specific sequences of hardware failures and human errors to occur for an accident to result.

Certain elements in the baseline (part of the ROCS, part of the WIU, and the TEU) which create or act upon train control commands are noted as vital components. The commands generated by these components, unlike sensor information, cannot be cross-checked against other measurements. The system must be carefully designed so as to detect rapidly, with a sufficiently high probability, any failure of one of these components. In fact, this is done by the use of locally redundant hardware monitors. The probability of rapid failure detection is described by a parameter called coverage. These vital components are modeled as dual redundant so as to provide a means of analyzing these coverage values.

Vitality of data transmission is achieved by the use of error detecting codes applied to the data by a vital component. This adds temporal redundancy to transferred data. Failures in the data communications path — the microwave system, network controller, VHF radios, GTCs, WIU modems or on-train computers — may cause a loss of communications, but have a very small probability of changing the transmitted data without also invalidating the error detection encoding. Navigation components (the tachometer and the GPS receiver) are not individually vital because there is redundant information in the system against which their outputs can be checked.

Errors in certain critical software modules could lead to an accident. These effects are not included in the model for the following reasons:

1. ARES software would have to sustain a critical failure[3] about once a week in order to raise the accident rate of ARES to that of the current systems. Experience indicates that the rate of software failures for configuration controlled software decreases with time[4], so after checkout the critical software error rate would be well below this value, and would not increase with time.

2. ARES software will be subject to rigorous configuration management procedures.

3. ARES software will be certified using procedures for the verification and validation of critical control software which are successfully employed by the FAA, DoD and NASA. The portion of the ROCS and onboard train control software that is "vital" is of limited size, thus further enhancing the effectiveness of these verification procedures.

4. Judicious modularization of the software will allow changes of track and switch configurations without the need to reverify the entire software package.

The modeling process consists of, 1) determining from the baseline system architecture and operating procedures the modes of system failure that can lead to an

[3] Not just a system crash, because this is easily detectable, but a failure that insidiously injects incorrect and unresolvable information into the system. For example, overlapping clearance tokens.

[4] IEEE Transactions on Software Engineering Volumes SE-11, Number 12, December 1985 and SE-12, Number 1, January 1986 discuss the current state of knowledge about software reliability.

accident, 2) analyzing the sequences of component failures that lead to these system failure modes, and 3) applying component failure rates, repair rates and parameters derived from the baseline system architecture and operating procedures to compute the rate of accident occurrence.

A detailed discussion of the model and a listing of specific input values (equipment failure and repair rates, etc.) is presented in Part 2 of this report. Sources for the baseline input values are shown in Table S.2 below.

SOURCES FOR MODEL INPUTS

Equipment failure rates	Actual values from Rockwell and BN — along with estimated values based on CSDL experience with similar systems
Communication error rates	Estimates from Rockwell
Human error rates	Derived from existing system accident data
Repair rates	Specified by BN
Failure coverages	A function of the system architecture and operating procedures
Exposure time to human errors	A function of the system architecture and operating procedures

Table S.2

SUMMARY OF THE RESULTS

For the current control systems, the average number of control system related accidents on all BN lines is about 50 per year. The predicted rate if the full ARES were employed on these lines is 0.5 accidents per year. Therefore, ARES is about two orders of magnitude safer with respect to control system related accidents. The reason for this is that ARES employs highly reliable computerized information cross-checks and clearance enforcement mechanisms that do not exist in the current system.

Almost all accidents caused by failures of the current control systems are attributable to the human element. The human is an integral part of the current train control systems. He is a critical, non-redundant element responsible for observing signals and operating the throttles and brakes, and thus he closes the train control loop. The system is exposed at all locations and at all times to single human error events that can directly cause accidents.

The human is also an element of the ARES control system, but the ARES architecture is such that the human does not represent a single point failure mode. There are no single human error events that can directly precipitate an accident because ARES clearance enforcement hardware operates in parallel with the train operator. The human can contribute to an accident only as part of a multi-event sequence of independent hardware failure and human error events.

In the same manner that ARES reduces the exposure to human errors, it provides redundant checks on its own hardware elements. Hardware component failures are detectable by ARES itself. The knowledge of which hardware components are inoperative allows the system to prevent these failed components from precipitating accidents, i. e., the system hardware is designed to be fail safe. Specific multi-event sequences of hardware failures are required to cause an accident due to hardware failure alone. These sequences turn out to be very improbable.

Sensitivity of the Results to Input Data and Modeling Assumptions

In order to gain confidence in the predictions of the model, it is necessary to assess the sensitivity of these predictions to certain aspects of the modeling process. These are the fidelity of the model, and the accuracy of the inputs. Fidelity relates to just how much detail of the actual system behavior is included in the model. The inputs are, of course, the failure rates of system hardware components, human error rates and certain system parameters such as failure coverage (the probability of detecting and annunciating the occurrence of a failure during system operation).

Fidelity needs to be such that model truncation, i.e., ignoring highly improbable events, and other approximations can be shown to be either conservative (produce pessimistic results) or to contribute only a negligible amount to the accident rate prediction. This model has been constructed so that the accident rate contribution of truncated events is several orders of magnitude below the contribution of modeled events, so as to produce errors in the fifth decimal place or less. The procedures for model truncation and actual error estimates are described in section 4.2.3.2 of the report.

The model is designed to be conservative, i.e., errors in the results due to approximations in modeling ARES behavior will always make ARES safety look worse than it actually is. By way of example, consider that the model does not attempt to address those cases where the human *could* in some cases detect and compensate for certain "undetected" ARES hardware errors (which are part of accident causing event sequences not involving human error).

Sensitivity to any particular input parameter is determined by running the model for a range of values of that parameter while holding all other inputs at a baseline value. The results indicate where, over the range of that parameter, the model predictions change significantly. If the change in the prediction of the model in negligible as the input parameter is varied around its own baseline value, then the model is insensitive to the particular value of that input parameter. One can also determine how large an error can be tolerated in the estimate of the input parameter before the predictions of the model are significantly affected.

It turns out that the model is not very sensitive to changes in the failure rates for the hardware components of ARES, and in fact these failure are fairly well known because similar components have been in operation for long periods of time and meaningful statistics are available.

The rate of human error was calculated from railroad accident statistics and then scaled by the average number of operators (dispatchers and engineers) in the system to determine a value for *errors that result in reportable accidents per man per hour*. While this value, itself, is sensitive to the assumed number of operators in the system, the prediction of the ARES safety model is almost insensitive to this assumption, because the assumed number of operators appears elsewhere in the model in a compensating manner. The details of the human error rate derivation appear in section 5.1.5 of the report.

Finally, it is necessary to consider how a different set of accident statistics for the current system would affect the ratio of the predicted ARES accident rate to the current accident rate. Recall that in the current BN accident statistics (Table S.1) all but one of the control system related accidents were caused by human error, and that it was from these errors that the human error rate for the ARES model was derived. Since the ARES model is linear in the human error rate parameter over a fairly wide range, a change in human component of the current system accident rate would result in a proportional change in the predicted ARES accident rate. Thus the ratio of accident rates would remain about the same.

If a larger proportion of the current system accident statistics were attributable to hardware problems, then these would "count against" the current system but would not influence ARES, since only the human component off the current statistics serves as input to the ARES safety calculation. Thus the ratio of predicted ARES accidents to current accidents would go down, making ARES look even better.

PART 2

DETAILED ANALYSIS

1.0 INTRODUCTION

ARES comprises those components which are part of the system responsible for train control: the management of the movement of trains, motive power, and track forces within an operating region, excluding yard activities. ARES is responsible for the control of anything moving or occupying the tracks within the region. This includes:

1. "Navigation" equipment to determine the position of trains and track forces;

2. Wayside equipment to sense and in some cases remotely control the position of switches, to monitor track status and integrity, and to detect abnormal train conditions such as overheated bearings;

3. A central computer facility to plan the movement of trains, to monitor train positions, and to issue the proper movement authorities;

4. A communications system to link the trains and wayside equipment to the central facility;

5. On-train computation elements with interfaces to the navigation equipment, throttle, brakes and data communication radios.

Braking systems, for example, are not part of ARES, but those components which make and carry out the decision to apply the brakes are part of ARES. In particular, the human operators , i.e., engineers and dispatchers, are part of the control system.

1.1 ARES Approach to Safety

ARES is designed so that no individual component failure or human error can result in an accident condition. Broadly speaking, this is accomplished by the use of redundant components that are used to provide independent copies of information which can be checked for consistency. In some cases the inconsistent information can be arbitrated by a third source, allowing the system to continue normal operation. In cases where this is not possible, the system operating mode is changed to one that does not use information from the inconsistent sources. Sometimes this may require that local operations be halted until the problem can be resolved. In all cases, however, ARES has knowledge of the validity of its own information, and will cease the movement of trains if enough valid information is not available. Thus for individual component failures, ARES is designed to be fail-operational in many cases, but at a minimum to be fail-safe.

It is assumed that when a failed element has been identified that the problem is corrected expeditiously. However, it is possible to identify certain sequences of component failures such that if the early failures have not been corrected before the latter failures occur, an accident condition can result. The probability of occurrence of such sequences of failures provide a measure of the accident rate that can be expected under ARES operation. The safety of ARES is, therefore, a function of the reliability of the ARES components, the repair times for those components, and the manner in which given sequences of failures interact. This information can be employed in a mathematical model of ARES system failure behavior in order to predict the accident rate that can be expected from the ARES control system.

1.2 The Need for Safety Modeling

Henceforth, the terms reliability and safety will be used more or less interchangeably, since the aspects of system reliability being investigated here are exactly those that impact on the safe operation of the system.

The design and development of complex, highly reliable systems presents an interesting dilemma: how does one establish that the design does in fact have the potential to achieve the required reliability. The traditional approach of prototype testing is not feasible. Since the system is built such that failures are infrequent, a prohibitively long testing period is required. Constructing many copies of the complex system and testing them for a shorter time is not possible due to cost limitations. Further, since the system reliability corresponds directly to safety, then measures of the system safety should be obtained before operational testing of the system can be permitted.

ARES is a large and complex system. It addresses the problem of achieving high reliability at the system level by employing the appropriate configuration of redundant components which themselves need only be moderately reliable. The reliability (hence the safety) of such a system can be assessed by the application of mathematical reliability modeling techniques. ARES components (radios, computers, sensors, etc.) are such that their failure rates can be determined by empirical measurement.[1] These rates serve as inputs to an analytical model, which is based on architectural structure and operational rules, to obtain the overall accident rate for ARES. Specifically, the analytical model employs information about the system architecture (how the system's components are interconnected), system operating mode descriptions (what equipment and abilities are needed for the system to be operational in its various modes), and the redundancy management approach (how component failures are detected and identified, and how the system is reconfigured to accommodate these failures).

[1] The error rate of the human operators, which are components of the train control system, is also obtained by empirical observation, i.e., statistical data about human caused accidents.

1.3 The Selection of an Analytical Reliability Modeling Technique

The analytic approaches to quantifying system reliability fall into three classes: Monte Carlo simulations, combinatorial methods, and Markov models. The strengths and weaknesses of these three approaches are briefly described here. A more detailed discussion of these modeling techniques can be found in the Reference 1.

A *Monte Carlo simulation* can be used to determine reliability by simulating the failure of components at times distributed according to their failure rates. These simulations are repeated until statistically significant reliability measures are accumulated. An advantage of the simulation approach is that very little knowledge is required beyond a description of the system to be analyzed. However, a key difficulty of this method is that for systems that require analytic determination of their reliability, i.e., highly reliable systems, a very large number of simulations are needed to obtain statistically meaningful results. For example, if a system is designed to have a failure probability of 10^{-6} per year, then there would only be one failure out of one-million simulations. Many more simulations would be required to obtain a statistically meaningful reliability value. Further, consider the case of a comparison study. If one million simulations are performed and both systems have only one failure, then it is *not* correct to assume that they both have the same reliability. The proper conclusion is that they both have reliabilities that cannot be determined or distinguished from each other by a one million sample set.

Historically, *combinatorial reliability models* have been widely used. Fault-tree analysis , for example, has become a standard analytical method for reliability prediction in a wide variety of applications. This analytical technique statistically combines component failure probabilities, based on the system architecture and redundancy management approach, to determine the 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. Secondly, it is difficult to include events that have order dependencies, such as repairs and explicit modeling of reconfiguration strategies. Even in simple systems, there are often sequence dependencies which are quite subtle. 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 validate.

Recently, *Markov modeling* techniques have been used for reliability prediction. These techniques have also been used successfully to aid in the design of fault-tolerant systems. 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 transitions rates which reflect component failure rates and redundancy management performance. Elements in the model's state vector represent the probability of being in each state at a specified time. Since the Markov model traces the evolution of state probabilities based on the probability of component failure, it is not explicitly simulating the system, and therefore, does not have the associated deficiencies that are found in the Monte Carlo technique. The Markov model is equivalent to a system of differential equations. Hence, order dependencies such as repairs and redundancy management are included naturally. Further, 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 time period in question; it is only required to model events that can occur during a relatively short time step. A drawback to the Markov method is that the state space grows exponentially with the number of components. Techniques have been developed to deal with this problem.

It should be emphasized that the reliability of a system does not depend on the analytical method used to evaluate it. Thus, a combinatorial reliability model and a Markov reliability model represent equivalent and interchangeable methods for evaluating a system's reliability.

The Markov modeling technique has been selected to examine the safety (reliability) of ARES. ARES is a complex system comprising thousands of components. Repairs occur and human operators play an important role in the system's reliability. Monte Carlo simulations are ruled out due to their inefficiency. The combinatorial approach is not easily used because of the system's large size and the existence of sequence dependencies such as repairs. The Markov modeling techniques are particularly suited to this problem if the difficulties of state proliferation can be dealt with.

1.4 Overview of ARES Modeling Techniques

The key difficulty in using Markov models is the problem of state proliferation. In fact, appropriate means must be found to reduce the state space or the Markov model will be intractable even for moderately sized systems. Further, an appropriate model must be found for the humans which are integral elements of the system.

As an example of the problem of state proliferation, consider a system composed of 20 components. To model all (sequence independent) failure combinations requires 10^6 states. However, 10^{12} states are needed to model a system with 40 components. Hence, the number of states grows exponentially with the number of components. A brute force generation of an exact Markov model for an ARES control region, with its approximately 1000 components, is clearly an intractable problem.

Two techniques are used to alleviate this state space explosion. First, it is noted that in many cases failures of different component types do not interact in such a way as to cause an accident. Thus, there are certain sets of equipment whose interactions in the safety analysis are easily understood. The failure of a switch does not impact the failure of any locomotive equipment, for example. This permits the analysis of switch failures and repairs and their impact on system safety independently of the investigation of the impact of locomotive equipment failures. The result is a set of essentially independent Markov models for each set of components, the outputs of which are merged using a combinatorial technique. In Section 4 we will demonstrate the independence of these submodels, which appears as an orthogonality property, and permits the use of this hybrid approach.

The second technique applied to reduce the state space deals with the problems associated with the submodels. For example, the submodel for the switches may have hundreds of switches to keep track of. If all switches are assumed to fail in such a way as to have the same impact on safety, then there is no need to distinguish among the individuals. This property is known as symmetry. The resulting model has one state for "no switches failed", one state for "one switch failed", one state for "two switches failed", etc. This still results in a model with hundreds of states, a situation that is not desirable. Clearly, some states are more likely than others. The state where all switches are failed is very unlikely, as is the state where all but one have failed. Markov modeling permits the use of systematic techniques for focusing the modeling efforts on the failure modes that have significant impacts on the solution. These techniques are called model truncation.

Thus, a large, complex system such as ARES can be modeled using Markov techniques. The state space is controlled through a hybrid approach that permits the solution of a set of submodels that are combinatorially merged and the use of symmetry and model truncation to control the size of the submodels.

There are two unique aspects to the analytical evaluation of ARES. The first is the application of the above techniques to a system with repairs. The second is the inclusion of humans as critical elements of the system's operation by modeling human error events as virtual transitions. The modeling of the human error process and the determination of human error rates are discussed in detail in Sections 4.2.2 and 5.1.5.

1.5 The Modeling Process

The process of generating a reliability or safety prediction for a system can be divided into three steps. First, the system needs to be investigated. The goal is to discover how the system operates and what are its critical aspects. 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 accident 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. The first two steps are discussed in detail in Section 3 and the modeling step (step three) is described in Section 5.

The actual process of generating a model requires three types of information: architecture, 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 operational requirements provide a definition of what equipment or abilities are needed to achieve an operational state. There may be several operational modes, such as full ARES or radio blocking rules. Further, these various modes may exist at various locations within the system simultaneously. The reconfiguration procedures are the actions taken when a failure occurs so that system operation remains in the most desirable mode.

The notion of system reconfiguration is implicit in a redundant system. Having a redundant element for backup purposes is of no use if the system cannot detect the failure of the primary element, locate which element has failed, and take proper action to provide the system with access to the backup. All of this must take place in a time period such that system operation is not critically affected. If this process of fault detection, identification, and reconfiguration (FDIR) occurs in an acceptable time period, the component fault has been "covered". Thus, the performance of the system FDIR—the fraction of each component's faults that can be covered—must be included in the system model. Sometimes these coverage values are known; often they must be calculated using a Markov model to explore the performance of the FDIR process. Coverage models of the dualized vital elements of ARES, such as the ROCS and WIU, are shown in Section 4.2.1.

A variety of information can be obtained from the ARES safety model. The main result is a prediction of the overall accident rate that can be expected when ARES is put into operation over a given operating region. This value can be compared to the present measured accident rate for the region to determine the relative safety of ARES. The model provides information about which components of the system contribute the most to the accident rate. This allows efforts directed towards the improvement of hardware reliability to be focused where they will do the most good. The model also indicates when certain operational decisions impact safety (such as whether to continue operation in a backup mode, or to wait until the primary mode hardware has been fixed). Finally, sensitivity analyses indicate how different modeling assumptions and uncertainties in inputs to the model affect the results.

2.0 THE CURRENT TRAIN CONTROL SYSTEM

A representative accident rate due to failures of the current train control system was extracted from the BN Reportable Rail Equipment Incident Scoreboards for the years 1984 and 1985. This data is shown in Table 2.1 along with the equivalent numbers for all U.S. railroads. Note that the BN numbers are roughly in proportion (by accident type) to those for all railroads as a group. These numbers represent real accidents above a certain monetary threshold that actually occurred, and therefore serve as the best source of information about the accident rate of the current train control system. About 50 reportable accidents per year over the entire BN system (approximately ten percent of all reportable BN accidents) were attributable to the train control system.

Note that, the phrase "current train control system" actually refers to the group of train control methods currently in use on the BN system. These include Timetable and Train Order (TTTO) and Track Warrant Control (TWC) in use on unsignalled territory, Automatic Block Signaling (ABS), and Centralized Train Control (CTC). The available accident data is organized by accident type. Accidents within each type may have occurred under any of these various control methods, so a characteristic accident rate for each type of current control method cannot be easily determined from this data. However, the totals are all that is necessary to compare present BN control system safety with the same trackage under ARES control.

The human is an integral part of the current train control system. He is a critical, non-redundant element responsible for observing signals and operating the throttle and brakes, and thus he closes the train control loop. The system is exposed at all locations and at all times to single human error events that can directly cause accidents.

One fact is plainly obvious from the statistics: almost all current control system related accidents are the result of human error. Less than one percent of these control system accidents were attributable to hardware failure. Working backward from this characteristic, and the fact that in each of the current control systems the human represents a single-point failure, one can derive a deceptively simple "safety model" of the current control system that is nonetheless accurate to better than one percent. This model serves as a basis for determining *the rate of human error that results in accidents* which is used as input to the ARES safety model (Section 5.1.5).

CURRENT CONTROL SYSTEMS ACCIDENT DATA

[BN / National]

	1984	1985
Human Factor Causes		
Buffing--slack		
Buffing or slack action excessive	15/175	13/167
Employee Physical Condition		
Impaired efficiency or judgement – drugs or alcohol	1/2	0/1
Employee falling asleep	1/2	0/2
Human Miscellaneous		
Lateral drawbar force on curve excessive	12/60	7/43
Other rules		
Failure to stop train in clear	4/48	6/45
Instruction to train/yard crew improper	2/9	3/9
Motor car – on track equipment rules, failure to comply	8/27	9/17
Special operating instructions, failure to comply	2/9	0/12
Train order or timetable authority, failure to comply	0/5	3/6
Train orders – error in preparation, transmission or delivery	1/1	0/1
Signalling		
Fixed signal improperly displayed	0/1	1/1
Block signal, failure to comply	1/5	0/5
Speed		
Train inside yard limits, excessive speed	2/15	3/16
Use of Brakes		
Automatic brake, improper use	1/20	0/19
Dynamic brake, improper use	1/5	1/4
Independent brake, improper use	2/10	1/16
Signals and Communication		
Other communication equipment failure	1/2	0/0
TOTAL	54/395	47/364

Notes: Includes all reportables (>$4,500 in 1984, >$4,900 in 1985)
BN data from BN Reportable Incident Scoreboard
National data from US DOT FRA Office of Safety Accident/Incident Bulletin Numbers 153 (for 1984) and 154 (for 1985)

Table 2.1

3.0 ARES AND ITS OPERATION

The fundamental unit of ARES is the control *region*. The regional control center gathers information about the state of the track and all entities occupying track within the region and uses the information to direct the movement of these entities. From the viewpoints of train control and safety, regions are essentially independent of one another.[1] Therefore, once a model for an ARES region has been established, it can be easily customized to each of the control regions of the railroad and the resulting accident rates can be added.

3.1 ARES Baseline

A diagram of the ARES baseline for a single control region is shown in Figure 3.1. It depicts just those elements of ARES that are involved in train control, and whose failure could result in an accident. For example, the locomotive analysis and reporting system (LARS) is not included because it does not participate in any critical train control processes, nor does its failure have any diabolical effect on those components that *are* required for train control.

3.1.1 Components of ARES

ARES consists of four categories of equipment: mobile, wayside, communication network and control center. For purposes of modeling the effects of equipment failures, the components within these transceive are broken down into functional units. The criterion for scoping a functional unit is that any failure within the unit makes all the hardware of that unit, and thus all of the functions, unavailable to the system, i.e., all of the hardware in the unit is considered to be in series from the viewpoint of reliability.

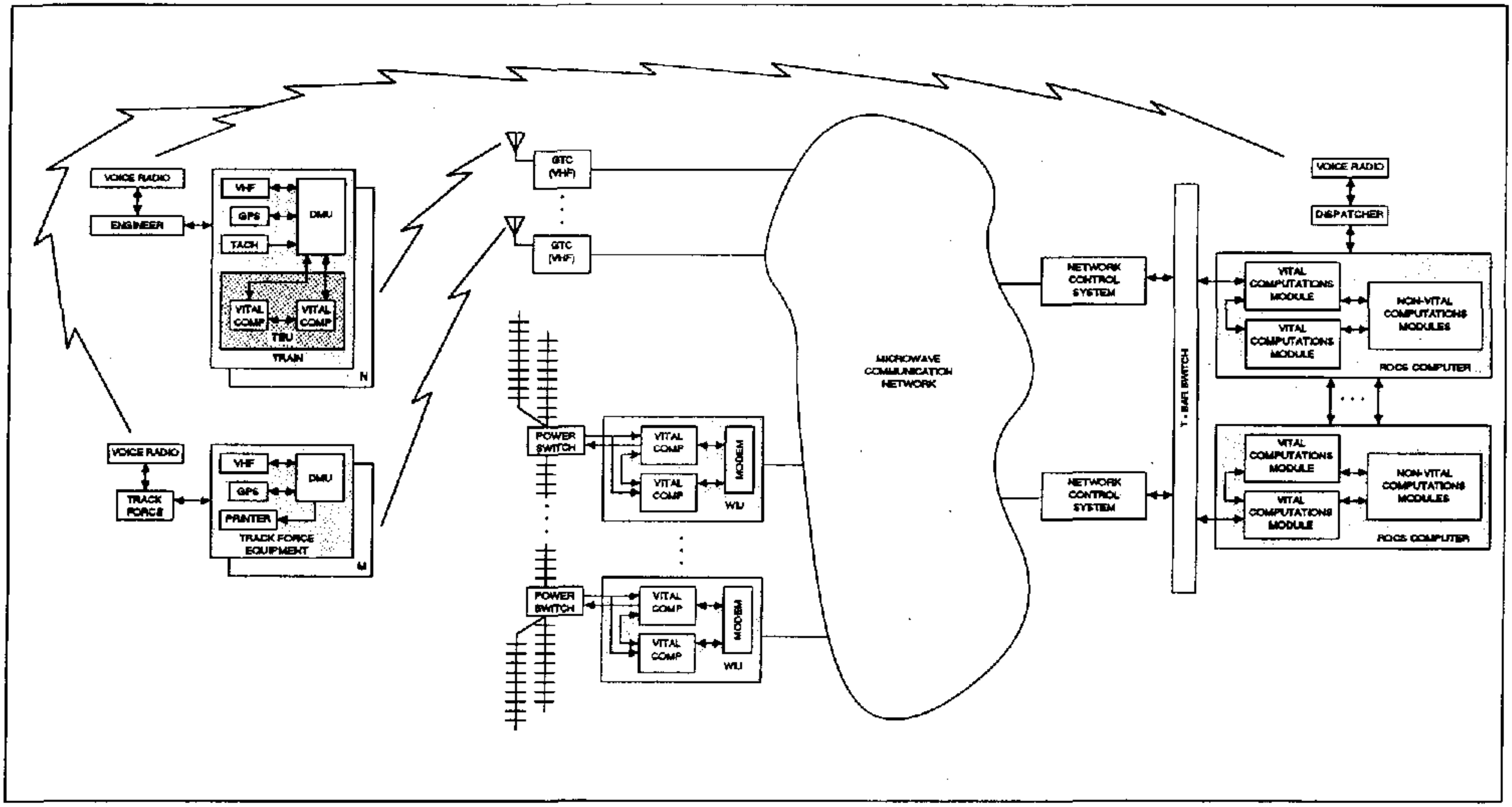
3.1.1.1 Mobile Equipment

Mobile ARES equipment is that carried by trains, track forces and inspectors — anything that moves on or occupies the tracks. A full set of mobile equipment is carried on-board the lead locomotive.[2] It supports navigation (train location), supplies data communications to the regional control center, and provides the interface to the locomotive's throttle and brake systems. This equipment consist of:

1. <u>A global positioning system (GPS) navigation radio set</u>. This device receives information from Navstar satellites, and using triangulation methods, calculates the position and velocity of the locomotive (in particular, the

[1] The handoff of a train between adjacent regions does represent regional interaction, but this has no significant effect on the safety modeling.

[2] This modeling effort assumes the most conservative scenario where only the lead locomotive has ARES navigation and communication equipment on-board. It is likely that, in four to six locomotive consists, more than one locomotive may be fully equipped. This provides operational backup and further enhances system safety.



ARES SYSTEM BASELINE
Figure 3.1

antenna atop the locomotive). These position fixes are available continuously. The position information is in coordinates of latitude and longitude, so it must be translated to location along the track using a map of the tracks in the region. While the accuracy of the GPS position fixes is better than 100 feet, it is not so good as to be able to distinguish which of a pair of parallel tracks the train is on. This must be determined at the control center by monitoring switch settings when the train crosses the switch.

2. A tachometer. This provides a second source of information about the train's location. Integrating the velocity information from the tachometer gives an estimate of the distance travelled over some time interval. The tach information serves as one check on the health of the GPS set, and provides a backup method for navigation for those periods when GPS signals are not available, such as in a tunnel.

3. A VHF digital data communication transceiver. This provides the radio link to the trackside VHF units that are part of the communications network.

4. A VHF voice radio. This radio set, similar to the ones now in use, is employed by the train engineer in the event of equipment failures that prevent full ARES operation. This backup mode of operation uses radio blocking rules similar to the track warrant control method now in use.

5. A token enforcement unit (TEU). This equipment allows ARES access to the throttle and brakes and thus provides ARES with the ability to slow or stop the train so as to preclude the development of a potential accident situation. Intervention may be remote, by the dispatcher or the central control computer, in the event of engineer incapacitation; or it may be local, by the TEU itself, when the movement authority given to the train by the regional control center is exceeded or expires.

6. A data management unit (DMU). This ties together the other ARES components on the locomotive and provides computation capability where needed.

Certain control system components onboard the train are not included in the model because 1) they are not vital in the sense discussed above, and 2) they are redundant, so that the probability of loss of function for this equipment has a negligible effect on the system accident rate. These components are the display devices used to present information to the engineer, and the power supplies for the ARES equipment on the train.

3.1.1.2 Wayside Equipment

This consists of trackside sensors and actuators and the wayside interface units (WIU) which connect them to the regional control center via the communications network. The sensors monitor such things as switch positions, track integrity, hot bearing detectors, over-switch (OS) circuits, etc. — those items commonly in use now. Powered switches would also be remotely operated from the regional control center. A WIU may connect to the communications network via either ground lines or a VHF radio.

3.1.1.3 Communications Network

This provides the data paths between the mobile equipment, the wayside equipment and the regional control center. Its consists of several parts:

1. A backbone of microwave towers with redundant transmitters and receivers spaced at an average interval of about 25 miles along the mainline tracks.

2. A ground terminal controller (GTC) and digital VHF data transceiver located on each tower. The GTCs store and forward information between the trains and WIUs, and the regional control center.

3. A network control system (NCS) with redundant main processors which interfaces the microwave network to the regional control center. The NCS also has the ability to access selected microwave towers over dial-up phone lines. This provides backup communication paths in the event of multiple microwave equipment failures or temporary local blackouts due to atmospheric conditions.

3.1.1.3 Regional Control Center

This consists primarily of a Rail Operations Control System (ROCS) with redundant computers connected via a simple switching mechanism to the redundant network control system. The ROCS is responsible for monitoring the status and location of trains and track forces, monitoring switch positions, track integrity, etc., and issuing movement authority (also called warrants, clearances or tokens) to trains and track forces.

3.1.1.4 Baseline Equipment Complement

The baseline region consists of: a ROCS; an NCS; 100 Microwave Towers, each with a VHF/GTC; 600 WIUs; and an average train density of 100 trains per region (100 each of the GPS, tachometer, DMU, onboard VHF, and TEU).

3.1.2 ARES System Operation

ARES controls the movement of trains in a manner similar to that of track warrant control. The ROCS , using its knowledge of the location and speed of all trains in the region, issues relatively short-term clearances designed to optimize the flow of traffic while maintaining safe headways between trains. The clearances, or tokens, contain a distance limit (from the present location), a speed limit and a time limit. The clearances are displayed in the cab and the engineer is free to control the train within these limits. If, for any reason, a clearance limit is exceeded or expires before an updated clearance has been received, ARES will act, through the TEU, to slow or stop the train. The system gives the engineer the latitude to operate the train, but will limit his actions if there is the possibility that they will lead to an accident.

Successful ARES operation depends upon

1. Accurate knowledge of the state of all system elements: train locations and speeds, switch positions and track conditions, and the health of the hardware elements providing this information.

2. A reliable communications system: one that experiences a minimum of *undetectable* communications errors. Loss of communication is an acceptable condition, but transmission and acceptance of incorrect movement authority is not.

3. Fail safe mechanisms for clearance generation and enforcement.

In general, the failure of an ARES component or garbled communication is not, in itself, critical. Timely detection and identification of the failure *is* critical. In particular, the system must know whether a clearance has been computed, transmitted or interpreted incorrectly, and it must know whether its knowledge of train locations and switch positions is correct. The design of ARES components and the way in which they interact addresses the problem of the system's knowledge about its own failures.

3.1.2.1 Clearance Integrity

Clearance authorities (tokens) are calculated by the ROCS and sent to the trains. The train is guaranteed clear track, proper headway and proper switch settings as long as it remains within the bounds of the clearance. This clearance remains valid even if data communications fails immediately after the token is received by the train. It is imperative, therefore, that the train actions not be based upon an invalid token.

An invalid token is caused by a failure within the system. The token may be created improperly in the first place, because of a ROCS failure; a proper token may be damaged by an error introduced in the communication system; or a TEU failure may cause the token to be interpreted improperly. The creation and interpretation elements, the ROCS and the

TEU, are considered individually vital, that is, if either fails in an unsafe manner a train may act upon an invalid clearance (or in the case of a failed TEU, may fail to enforce a proper clearance). In both of these elements, redundant hardware is employed so as to detect any errors as they occur.

That portion of the ROCS that is concerned with calculating clearances is internally dual-redundant. Any discrepancy due to a hardware failure will cause that ROCS to stop operation before a bad token can escape. Operations then switch to the other ROCS if it is healthy. If neither ROCS is working, automatic central operations cease. Trains can run to the ends of their tokens. At this point the system may be run manually over the voice radio, using the appropriate radio blocking rules, until a ROCS is made operational.

The token enforcement unit (TEU) is responsible for correct token interpretation and enforcement (via an interface to the throttle and brakes). It employs internal redundancy in a fail-safe mechanization so as to stop the train if an internal discrepancy is detected.

Bit errors can be expected to occur in any communication system. These are addressed by employing two levels of error detecting cyclic redundancy code (CRC). The inner level of CRC is applied by the vital portion of the ROCS where the clearance is created, and decoded within the TEU. The outer level of CRC is applied by the network control system and decoded within the data management unit on the train. If transmission of a token is not properly acknowledged by the TEU, then the transmission is repeated.

3.1.2.2 Knowledge of Switch Position and Track Condition

The wayside interface unit (WIU) gathers switch position information from sensors and sends this information to the ROCS. The WIU and the switch sensor are individually vital in that if incorrect switch status is used to prepare clearances for trains an accident will probably result. The part of the WIU that interfaces with the switch sensor and encodes the inner level CRC check is dual. Any hardware failure that occurs can be detected immediately, and incorrect information will not propagate.

The switch position sensor itself employs contacts at either switch point position. A switch position is "known" only if it is fully displaced one way or the other, i.e., one contact must be open and the other closed. Neither or both contacts closed represents an unknown switch position. Track integrity sensing mechanisms, of which track circuits are one example, are similarly fail-safe.

The modem in the WIU is not itself vital. First, its failure generally interrupts communication — a condition that is readily detectable at the ROCS. Second, any garbling of bits by the modem is protected by the vital CRC check placed on the data by the vital part of the WIU.

3.1.2.3 Knowledge of Train Location and Speed

ARES has several sources of information about train location. No single source is individually vital because these sources serve as checks on each other. GPS and tachometer data are sent from the train to the ROCS periodically, at a frequency that is a function of the speed of the train. Absolute train location information comes from the GPS receiver. This data is used to calibrate the tachometer, while at the same time the integrated tachometer data is used to catch any egregious errors in the GPS data. In addition, the ROCS applies reasonableness tests to all data, since there are limits to how far a train can move in any given time interval. Finally, OS circuits provide periodic fixes on train location.

The ROCS needs train location in order to direct train movements. The TEU needs to know location in order to effect token enforcement. The location data used by the TEU must come directly from the onboard sensors, since "refined" train location data from the ROCS would not be available if communications were lost — precisely the time when the autonomous TEU enforcement function is most critical. The token enforcement mechanism works in the following way:

1. Clearance limits for distance and time are stated in incremental terms, i.e., the train is authorized to go a given distance from its present location, or until a given time has elapsed. The speed limit for the clearance is stated absolutely. All three limits are independently enforced.

2. The TEU contains a redundant clock.

3. Tachometer information is fed directly into the TEU, bypassing any possible contamination from the DMU. This provides speed information directly, and the speed information is integrated (with respect to time) by the TEU to provide incremental distance traveled.

Valid tachometer information is necessary for proper token enforcement by the TEU. Under normal operation, the GPS data can be used to detect a tachometer failure. If either the GPS *or* the tachometer on-board a train fails, then according to the proposed operating procedures that train proceeds under radio blocking rules, at which time TEU token enforcement is no longer in effect. Thus high failure coverage for the tachometer is available when it is needed.

Under radio blocking rules, the unfailed source of train location data (either the tachometer or the GPS) is used to provide the ROCS with train location. This information is backed up by the engineer's observation of mileposts and by OS circuit fixes when available.

3.1.3 Non-Vital Components

The elements of the communications network (including mobile and fixed VHF radios), the DMU, GPS and Tachometer are not individually vital. A failure of one of these elements is covered by ARES as a whole and will not directly lead to an accident.

3.1.4 Modeling Individually Vital Elements

Dual redundancy is conceptually the simplest way to obtain a very high probability of error detection, although, depending upon the component in question, a high probability of error detection can also be obtained by combinations of specialized monitoring equipment. In either case, individually vital elements have been modeled as being dual-redundant so that the coverage parameter used in the safety analysis can be calculated in a tractable manner. Some, and perhaps all of these components will actually be implemented as dual-redundant.

3.1.5 Track Force Equipment

Track forces and inspectors carry a subset of the ARES equipment that allows their location to be monitored by the ROCS and commands to be sent to them from the ROCS. There is no clearance enforcement mechanism needed. Separation between track forces and trains can be effected by controlling the trains.

3.1.6 Software

Errors in the software of the ROCS or any of the on-train components have the potential for introducing erroneous information into the system. Predictions of a specific rate of occurrence of software errors in ARES, at this stage of the system development, is at best guesswork. However, it is possible to make qualitative statements about the process of software development for ARES. It is also possible to invoke past experience in the development of similar control system software to determine that any residual errors in ARES software after validation procedures are completed will not significantly influence the system accident rate.

ARES software will be certified using procedures for the verification and validation of critical control software which are successfully employed by the DoD, NRC, FAA and NASA. As part of the ARES development and maintenance process, the ARES software will be subject to rigorous configuration management procedures. Historical analysis of large software projects shows that the rate of software errors for configuration controlled software decreases with time.[3]

[3] Various papers in the IEEE Transactions on Software Engineering Volumes SE-11, Number 12, December 1985 and SE-12, Number 1, January 1986.

Sensitivity analysis indicates that the ARES software would have to sustain a critical error (not simply a system crash, but an error such as the issuance of an undetected overlapped clearance authority) about once a week in order to raise the accident rate of ARES to that of the current system. Experience indicates that properly developed and validated control system software is substantially better than this.Therefore, it is not necessary to include the effects of software errors in the safety analysis.

3.2 Conditions that Cause Accidents

Table 3.1 provides a functional breakdown of the error conditions that can precipitate accidents. These conditions arise as the result of sequences of failures of ARES equipment and human errors. If the condition is not corrected, it will eventually result in an accident. The system is modeled by relating these conditions to the particular sequences of failures that cause them, and then computing the probability of occurrence of those failure sequences.

- ROCS error in the knowledge of the state of the track
    - Switch position
    - Integrity of upcoming track (unreported breaks, earthslides,washouts, etc.)
- ROCS error in knowledge of the state of a train or track force
    - Location of a train or track force along the track
    - Location on a set of adjacent tracks
    - Train speed
- An error in a train's knowledge of its own state
    - Location
    - Speed
    - Consist
    - Brake pressure
- A violation of rules (i.e., the ARES operating procedures)
    - Improper position of a switch that is within the headway safety zone of an approaching train
    - Repositioning of a switch under a moving train
    - Improper clearance sent to a train
    - Improper action on the part of a train (speeding or exceeding clearance limit)
    - Mishandling of a train

Conditions That Cause Accidents

Table 3.1

4.0 MARKOV MODELING TECHNIQUES FOR ARES

In this section we discuss the modeling of ARES. First an overview of the process of creating a Markov model for a simple system is presented. This demonstrates the state space explosion problem that arises in Markov models. Means to mitigate the state proliferation are examined next. These include decomposing the system into submodels, truncating submodels, and independently modeling component coverage. The orthogonality (independence) of the submodels is also shown.

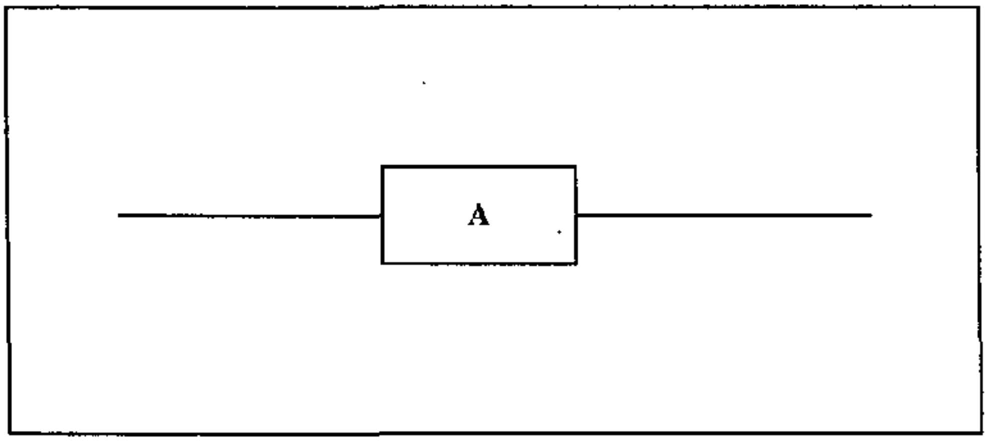
The model of ARES appears to be different from that of the current control system. At first glance the ARES model is obviously much more complex. This is not a reflection on the relative complexity of the two control systems. Rather, it represents the more complex interaction of humans and control system hardware in the development of accident situations. As was shown in Section 2, accidents in the current control system are caused, almost exclusively, by the humans involved. Hence a model of this control system can be quite accurate if it only models the human elements. The development of potential accidents in ARES is different in that the primary causes of accidents involve both the humans *and* the hardware. Hence the model of ARES must represent this hardware/human interaction. As will be shown in Section 4.1, it is the large number of hardware elements that must be modeled that gives rise to the ARES model complexity.

4.1 Markov Modeling Overview

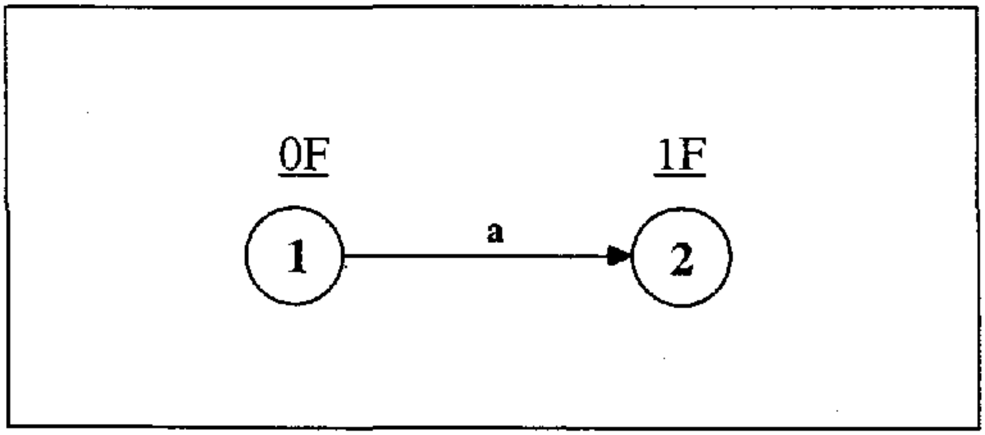
Markov modeling techniques provide a systematic means of investigating system reliability and safety for large, complex systems. They permit the inclusion of sequence dependent events such as repairs in a natural fashion. One of the most powerful aspects of Markov models is their ability to permit simplifying approximations to be made and to provide means to obtain bounds on these approximations. In this section two simple models are shown to introduce the concepts of Markov modeling. This sets the stage for an investigation of the difficulties, particularly that of an intractably large state space, in the modeling of ARES. Means to reduce the state space are described in Section 4.2. A more detailed introduction to Markov modeling and the control of the state space can be found in Reference 1.

4.1.1 Single-Component System

Figure 4.1a shows a single-component system. The first step in modeling the reliability of this system is to determine what is required for the system to be in an operational state. This single-component system has a trivial operational requirement: it is operational if the single component, A, has not failed (and the system is failed if component A has failed). While this step is simple for this system, it is often one of the most complicated steps in modeling a complex system.



Single-Component System Block Diagram
Figure 4.1a



Single-Component System Markov Model
Figure 4.1b

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. Figure 4.1b 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 all system components. In this case there is only one component, thus a transition noted with an a is created leading to state 2. State 2 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.

The system reliability is just the probability, as a function of time, of being in state 1. The Markov model of Figure 4.1b represents a set of differential equations whose solution is the probability of being in each state as a function of time. The notation a 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 a failures per hour. Each state has a probability associated with it. For example, at time = 0 the probability of being in state 1 (no failures) is 1 (or 100%) and the probability of being in state 2, or any other state, is 0. To obtain system reliability as a function of time we need to observe the probability "flowing" out of state 1 into state 2. 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 0 probability has no probability flowing out of it, a state with no exiting transitions (rate = 0) has no flow out, and a state with probability equal to 1 and an exiting transition rate of a has an instantaneous flow out equal to a. It will be assumed that all failure rates are constant in time.

Using the rule for determining probability flows the following equations are made from inspection of the Markov model in Figure 4.1a:

$$dP_1(t) / dt = - a P_1(t) \qquad (1)$$

$$dP_2(t) / dt = a P_1(t) \qquad (2)$$

These equations, which represent the changes in each state variable (P_1 and P_2), are called state equations. Equation (1) shows that the rate of change in probability for state 1 is the exiting transition rate a times the probability of being in state 1. The minus sign indicates that the transition is exiting and, therefore, reduces the probability of being in state 1. Equation (2) is interpreted similarly. Note that the flow is *into* state 2; the positive term indicates an entering transition which increases the probability in state 2. Also, the flow into state 2 is the rate a times the probability of *state 1*; the flow on this transition is due to state 1, the origin of the transition. Equations (1) and (2), along with the initial condition at time = 0 that the state probabilities are $P_1(0) = 1$ and $P_2(0) = 0$, provide a complete description of the system's reliability. Markov models have the property that flows leaving one state enter another, as is shown in Equations (1) and (2). Hence, the total system probability does not change as the system evolves. This property is called conservation of flow or conservation of probability.

There are many ways of solving Equations (1) and (2) in closed form, such as standard integration or Laplace transforms. Using a convenient technique, and noting that the failure rate a is constant, yields the following solution:

$$P_1(t) = e^{-at} \tag{3}$$

$$P_2(t) = 1 - e^{-at} \tag{4}$$

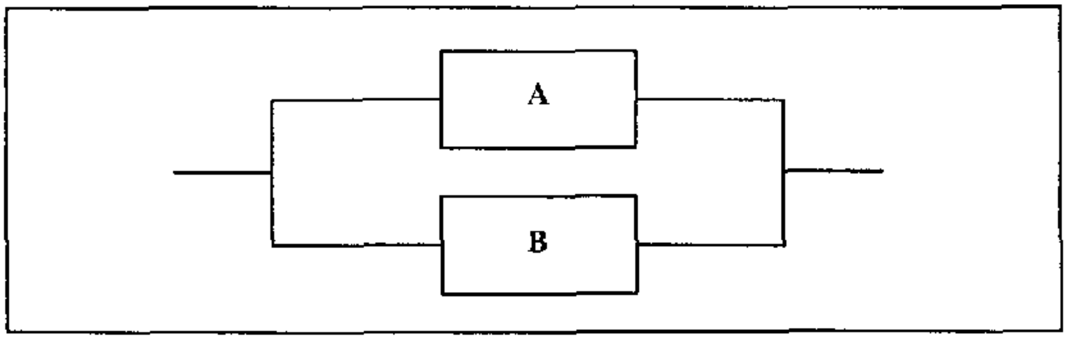
State 1 starts with a probability of 1 and decays exponentially toward 0, while state 2 has a probability initially at 0 which grows toward 1. Notice that the sum of the two states is 1 at all times. This shows the conservation of probability; the system started out with a total probability of 1 and this total remains constant.

4.1.2 Two-Component System with Repairs

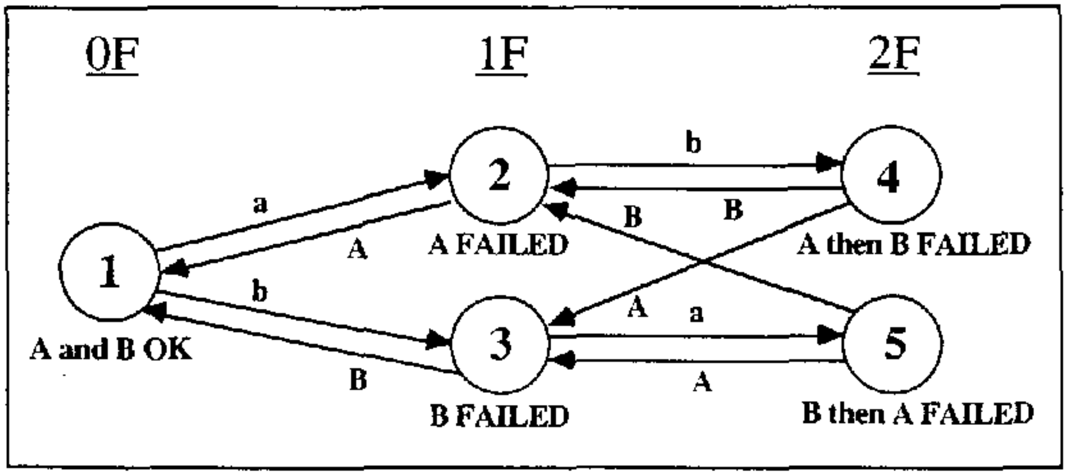
Figure 4.2a shows a two-component system where the components are connected in parallel. The requirement for system operation is that one of the two components is working. These components can be repaired when they are failed.

The Markov model of this system is shown in Figure 4.2b. State 1 represents the no-failure configuration. Possible events when in this state are that component A can fail or component B can fail. These two possibilities are captured in the transitions leaving state 1 with the rates a and b, respectively. State 2 represents component A failed and B working. Possible events leading out of state 2 are that component B can fail (exiting transition b) or that component A can be repaired (exiting transition A). The failure of component B leads to state 4, while the repair of component A leads back to state 1, returning the system to the no-failure state. Similarly, the exiting transitions for state 3, the B failed and A working state, are a failure of component A (transition a going to state 5) and a repair of component B (transition B going to state 1). Notice that repairs, which are sequence-dependent events (since they can only be performed *after* a component has failed), are easily included in the model. States 4 and 5 represent system failure since there is not at least one component working. States 1, 2, and 3 represent the system in an operational configuration. If one was concerned with degraded operational modes, such as operating without a backup, then this model could also provide that information by giving the probability of states 2 and 3 independent of state 1.

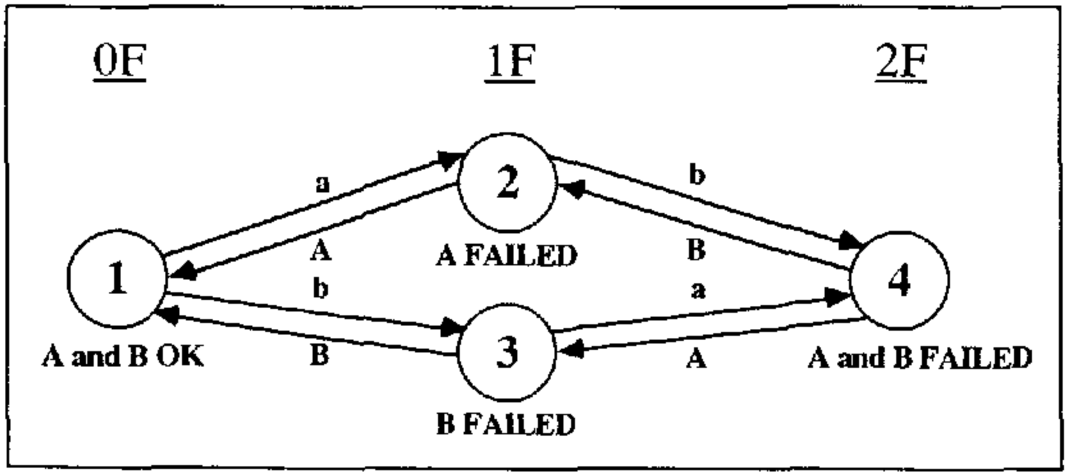
States 4 and 5 both represent system configurations where components A and B are failed. However, in state 4 component A failed first and in state 5 component B failed first. In both of these states, the possible events are the repair of A (transition A leading to state 3) and the repair of B (transition B leading to state 2). Since the possible actions taken, and



Two-Component System Block Diagram
Figure 4.2a



Two-Component System Markov Model
Figure 4.2b



Aggregated Two-Component System Markov Model
Figure 4.2c

their consequences, are the same in states 4 and 5, these states may be lumped together if the order-of-failure distinction is not needed in the analysis. The resulting model is shown in Figure 4.2c. This type of aggregation of states introduces no approximations. It is useful in systems where there are many identical components each with an identical impact on the system operation.

The state equations for the model in Figure 4.2c are obtained by inspection of the model diagram and applying the rule for determining flows. The state equations are:

$$dP_1(t) / dt = - (a+b) P_1(t) + A P_2(t) + B P_3(t) \quad (5)$$

$$dP_2(t) / dt = a P_1(t) - (b+A) P_2(t) + B P_4(t) \quad (6)$$

$$dP_3(t) / dt = b P_1(t) - (a+B) P_3(t) + A P_4(t) \quad (7)$$

$$dP_4(t) / dt = b P_2(t) + a P_3(t) - (A+B) P_4(t) \quad (8)$$

Note that all flows leaving a state (negative terms) appear as a flow entering a state (positive terms), thus probability is conserved. Equations (5) through (8), along with the initial condition that state 1 has a probability of 1 and all other states have probabilities of 0 at time = 0, provide a complete description of the system.

All systems reach a point where the state probabilities are no longer changing. In the example of the single-component system this situation occurred when all of the probability was in state 2 and none was in state 1. This is common for systems without repair: after a long period of time most states have probabilities of 0 and a few states, the trapping states, have probabilities that are between 0 and 1. Systems with repairs, however, have the property that when they get to this steady state all states may have probabilities that are between 0 and 1. This comes about because a balance is obtained on the flows leaving and entering the states. For example, when the flow leaving state 1 in Figure 4.2c equals the flow entering state 1, its probability no longer changes. This occurs when the probabilities of states 1, 2, and 3 obtain values such that the flows are in balance. Equation (5) shows that this balance is obtained when $dP_1(t) / dt = 0$. Similarly, when the differentials of all states are equal to 0, the system has come to its steady state. Additionally, this steady state must still conserve probability such that the sum of the state probabilities is 1. Thus, to find the steady state, set all state differentials equal to 0, generate the equation that sums all state probabilities to 1, and solve the resulting system of equations simultaneously.

The steady state is an important condition in the ARES analysis. ARES comes to its steady state in approximately 50 hours, yet it is a system that operates continuously over a much longer period of time. Although various components fail and are repaired as the system evolves, the probabilities of the various system states have come to steady state.

Therefore, the analysis of ARES is, in fact, an analysis of the system operating at steady state.

The closed-form solution of Equations (5) through (8) for this two-component system, as is true of most systems with repairs, is extremely complex and not particularly enlightening. It is more common to solve such system models numerically. First, the system equations (5) through (8) are written in matrix form:

$$\frac{dP(t)}{dt} = \begin{bmatrix} -(a+b) & A & B & 0 \\ a & -(b+A) & 0 & B \\ b & 0 & -(a+B) & A \\ 0 & b & a & -(A+B) \end{bmatrix} P(t)$$

where the state vector is:

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

Notice that the columns of the matrix add to zero. This represents the flow conservation in the system: all flows leaving a state must enter another state. The matrix equation may be written more concisely as:

$$dP(t) / dt = A\, P(t) \qquad (9)$$

Equation (9) 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, the one shown here is straightforward and sufficient to deal with the ARES model. The continuous differential is approximated with a discrete time step Δt:

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

Multiplying each side by Δt and moving the state vector P(t) to the right-hand side gives:

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

where matrix I is the identity matrix. The term in brackets may be relabeled as matrix M:

$$P(t+\Delta t) = M\, P(t) \qquad (10)$$

M is the discrete-time transition matrix. The use of the above approximation (Equation (10)) is called Euler integration.

Equation (10) represents an iterative solution for the Markov model. Given the system's initial condition, P(0), it is possible to use Equation (10) to propagate the state probability in time:

$$\begin{aligned} P(\Delta t) &= M\,P(0) \\ P(2\Delta t) &= M\,P(\Delta t) \\ P(3\Delta t) &= M\,P(2\Delta t) \\ P(4\Delta t) &= M\,P(3\Delta t) \\ P(5\Delta t) &= M\,P(4\Delta t) \\ &\bullet \\ &\bullet \\ P(n\Delta t) &= M\,P((n-1)\Delta t) \end{aligned}$$

The above procedure gives the state probabilities as a function of time from time = 0 to time = nΔt. This integration is easily coded on a computer.

A few notes need to be made concerning this solution procedure. First, Δt must be selected such that no elements in M are greater than 1. This maintains the stability of the integration and insures that the resulting probabilities remain between 0 and 1. In practice, reasonable results are obtained if the maximum off-diagonal element is no greater than 0.01. Second, in performing these calculations on a computer, double precision must be used unless a scheme to control roundoff error is included. Third, the steady state for the system can be found by evolving the state vector until the state probabilities do not change. Finally, a faster version of this integration is shown in Reference 1 (Section 7).

4.1.3 The State Space Explosion

It has been mentioned in previous sections that the major drawback of using Markov models to predict system reliability and safety is the problem of the growth of the state space. In this section we will demonstrate the rapid growth of the state space as a function of the number of components in the system. It is assumed that the systems used in this section do not have states that are distinguished by failure sequences. Hence, states which are only distinguished by the failure order are aggregated into one state. Each state is unique in that a specific list of components is failed; the order of these failures is not unique.

In Section 4.1.1 a single-component system was modeled. The model had 2 states. The two-component system modeled in Section 4.1.2 had 4 states when sequence dependencies were removed (Figure 4.2c). A 3-component system has 8 states, a 4-component system has 16 states, and a 5 component system has 32 states. Now consider a

20 component model. 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 of these 20 states has an exiting transition representing any of the other 19 components failing. Aggregating states with identical failed components gives 190 states at the second-failure level which describe the 190 combinations of dual failures. This pattern continues with 1140 states at the third-failure level, 4845 states at the fourth-failure level, etc., out to the 20th-failure level where there is one state representing all components failed. The total number of states is 10^6.

Repeating this exercise for a system with 40 components gives 1 state at the zero-failure level, 40 states at the first-failure level, 780 states at the second-failure level, 9880 states at the third-failure level, 91390 states at the fourth-failure level, etc., out to the 40th-failure level. The total number of states is 10^{12}. Storing this state vector requires one million megabytes of memory. Storage of the 10^{12} state equations would require much more memory. In general, a system with n components requires 2^n states to specify the model, if sequence dependencies are not of interest.

From these examples the exponential growth of the state space is apparent. However, these examples pale in the face of ARES. ARES contains *hundreds* of components. Clearly, a means of avoiding the state space explosion is needed for the ARES model to be tractable. In the following section a variety of means are employed to reduce the state space through various decompositions of the system and model truncation.

4.2 Decomposition in the ARES Model: The Subproblems

Given that the ARES model has potentially 10^{300} states, many techniques are used to reduce this intractable state space. The primary technique employed is that of dividing the system into subsystems whose interactions are easily understood. These subsystems are modeled independently (submodels) and their results are merged combinatorially. This hierarchical approach is applicable if the subsystems have a property we call orthogonality.

The division of the system into subsystems still leaves an intractably large state space for each submodel. Hence, further techniques are used to reduce the state space of these submodels. These include truncation of the submodels so they model only the significant contributors to the system accident rate, independent models of the dualized vital elements' coverage, and the use of virtual transitions to model the effects of human errors and transitions to the accident state.

The model for the coverage of the dualized vital elements is discussed in Section 4.2.1. The derivation of the virtual transitions used to model the human errors is shown in Section 4.2.2. Section 4.2.3 covers the generation, truncation, and orthogonality of the submodels. The complete system model construction, making use of all of the techniques for state space reduction and a combinatorial merging of the subsystem results, is shown in Section 4.2.4.

4.2.1 Coverage Model For Dualized Vital Elements

ARES system integrity is maintained by the use of dualized vital elements. It is necessary for the system to know when its vital elements are not operating correctly. If this is known in a timely fashion then the system can drop down to a fail-safe mode of operation. For example, it is required that the system know the position of a switch. This information is relayed to the Rail Operations Control System (ROCS) through a wayside interface unit (WIU). If the WIU fails and the system is not aware of the failure, it will think it knows the switch position when, in fact, it does not. This is a very dangerous situation. However, if the ROCS *always* knows when the WIU has failed, a fail-safe procedure can be implemented. This may involve directing engineers to manually inspect the switch position before proceeding across it. While this reduces the system performance, it permits safe operation even though a component has failed.

Given the need for detecting virtually all WIU failures, what procedures are used to obtain this detection ability? A common approach is to use two identical elements performing the same tasks at the same time. These dualized elements continually compare their outputs and shutdown if they do not agree. Thus the vital parts of the WIU are duplicated so that there is a means of detecting the failure of the vital elements. This detection procedure does not provide an indication of which of the two parts has failed, but this information is not required since the system has a backup, fail-safe mode. This approach of dualizing vital elements of other system components is used in ARES to insure that single hardware failures cannot lead directly to an accident.

In the above discussion it is assumed that the dualized elements will find all failures within an acceptable time. A Markov model can be used to investigate this claim of perfect coverage of element failures and provide a quantification of the perfection of the coverage. Thus, the failure of an element has two possible outcomes: the failure is covered and the system continues to operate, or the failure is not covered and an accident may result. Deciding which outcome occurs involves modeling the failure and detection processes for the element. The modeling requires many intermediate states to keep track of the competing actions of failures and detection. If this detailed modeling was included in the complete system model each state (i.e., each failure) would require many states to decide its outcome. The Markov model, which is already too large to analyze, would grow further if this detailed coverage modeling was included.

The alternative to detailed modeling of the coverage process *within* the complete model is to model the coverage in a separate model whose output is the fraction of failures that are covered. Now every failure in the complete model has two exit transitions; one reflecting the covered fraction of the failures and the other the uncovered fraction.

In summary, the dualized vital elements are needed to provide coverage of single hardware failures. The effectiveness of this coverage and its impact on system safety must be quantified. Explicitly including the details of the coverage process in the complete model would contribute to the state proliferation. Instead, independent coverage models are made and their results incorporated in the complete model as a fraction of the failures that are covered.

Figure 4.3 shows the Markov model for determining the coverage of dualized vital elements. The component is made up of some simplex elements which have a total failure rate of λ_2 and some dualized elements that have a failure rate of λ_1 for each half of the dual. The detection time is represented by the rate λ_3. In state 1 there are no failed elements. The failure of the simplex portion leads to state 2 and the failure of either half of the dual elements ($2\lambda_1$) leads to state 4.

In state 4 there are two competing processes: the detection rate λ_3 vs. the failure of one of the remaining elements. The detection transition leads to the detected state (state 3) where the system goes into the fail-safe mode. The transition to state 6 accounts for a failure of the remaining half of the dualized elements occurring before the first failure is detected. This near-coincident failure state is conservatively considered uncovered since the comparison of two failed elements may have a questionable outcome. The remaining transition from state 4 is the failure of the simplex elements leading to state 5.

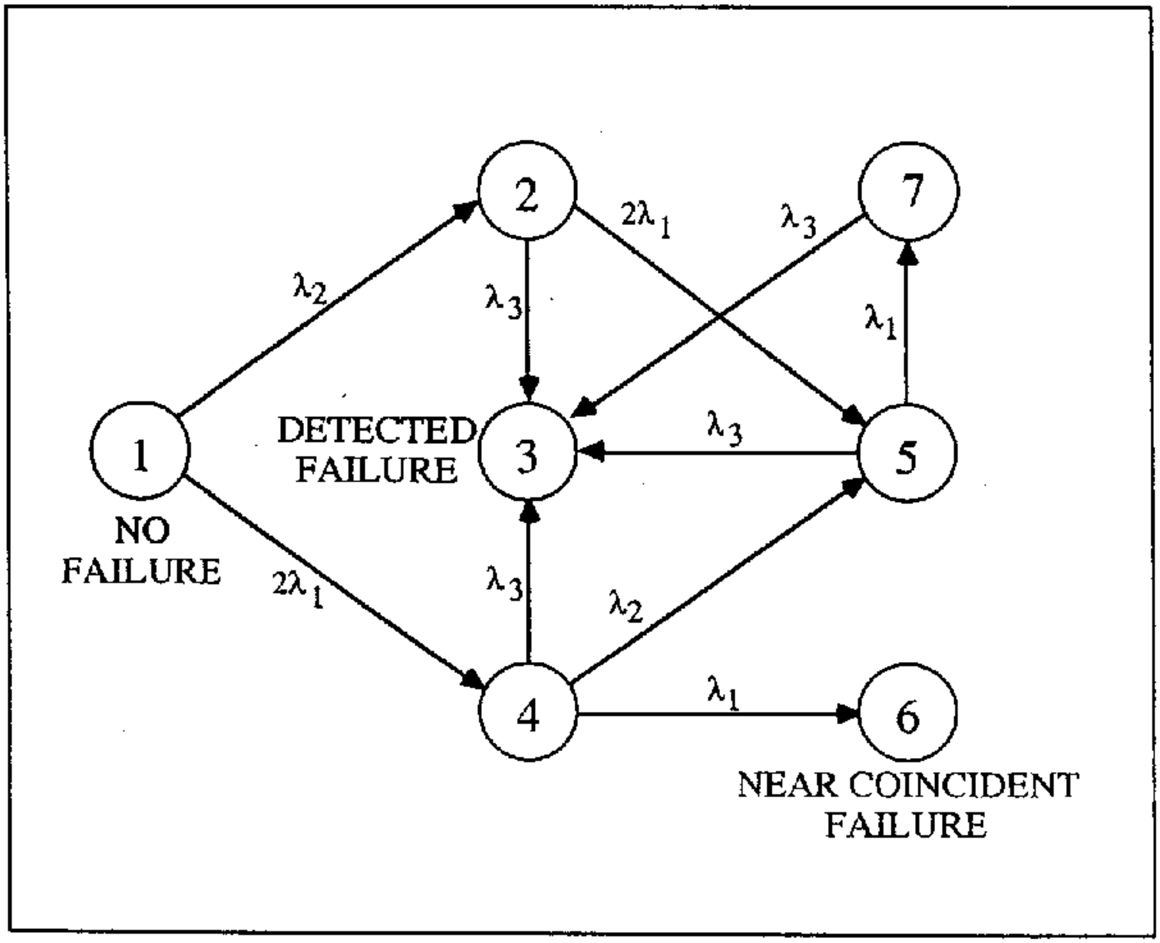
State 2 also shows a competition between the detection rate λ_3 vs. the failure of one of the remaining elements. In state 2 the failed component is the simplex element and it is assumed that this component's failure can always be detected within an acceptable time. Since a comparison is not needed to detect failures of the simplex element there is no transition to the near-coincident failure state. Instead, the only subsequent failures are one half of the dualized element. This leads to state 5.

In state 5 both the simplex and one-half of the dualized elements are failed. It is assumed that the component will not be labeled as operational if the simplex part is failed. Therefore, independent of the outcome of the dualized comparison, the component would be labeled as failed since its simplex element has failed. Thus, this state does not include an exit transition to the near-coincident failure state. The exits from state 5 are a detection transition to state 3 and a failure of the remaining half of the dualized elements which leads to state 7. In state 7 all elements of the component are failed and the detection, presumed to be accomplished by not replying to the ROCS, is the only subsequent event.

The model in Figure 4.3 permits an evaluation of the dualized vital component's coverage. The coverage value is the fraction of the faults that are detected such that the system can go to a fail-safe mode. The total probability that a fault has occurred is the sum of the probabilities of states 2 through 7. Therefore, the coverage value c is:

$$c = [P_2 + P_3 + P_4 + P_5 + P_7] / [P_2 + P_3 + P_4 + P_5 + P_6 + P_7]$$

The complement of the coverage, the fraction of faults that are near coincident and,



Coverage Model for a Dualized Vital Element
Figure 4.3

therefore, not detectable, is:

$$(1 - c) = P_6 / [P_2 + P_3 + P_4 + P_5 + P_6 + P_7]$$

The model in Figure 4.3 has two trapping states: 3 and 6. All of the other states are transient. Eventually all probability must end up in one of the two trapping states. However, it is of interest that the fractions c and (1 - c) are constant from times on the order of 1 hour to times approaching infinity. This means that one can talk about the coverage value as a constant. Table 4.1 shows the evolution of (1 - c) for the WIU. It is seen that fewer than one in a million of the faults result in an near-coincident, or undetectable, state.

This coverage model (Figure 4.3) provides a conservative approximation of the coverage values. The transition from state 4 to state 6 is the result of a failure of the remaining half of the dualized element before the failure of the first half has been detected. In the model it is assumed that any such failure in the second half will result in an ambiguous comparison between the dualized elements. In practice this is not true. For example, a dualized element made up of many chips will be able to successfully detect a miscompare for many of the possible combinations of the two chip failures. Thus, the model, which assumes that all second failures result in an uncovered state, is a conservative approximation of the real situation. This conservativeness is not a problem since the modelled coverage is sufficiently high that uncovered failures are not major contributors to the accident rate.

4.2.2 Incorporating Human Error Transitions into the Model

One of the key developments in modeling ARES is the inclusion of both humans and hardware components in the same Markov model. ARES is a system where both the hardware and humans interact in a way that prohibits the independent modeling of each. The development of an appropriate representation for the human part of this interplay is the subject of this section.

When a hardware component fails it is inoperative until it has been repaired. A human, however, does not generally exhibit this failure mode. It is assumed that the humans randomly introduce errors into the system at a specified average rate. The human does not need to be repaired before producing another error. It is this "instantaneous repair" that distinguishes the modeling of the human from that of the hardware.

Consider a two-component system comprised of a hardware component and a human. The hardware's role is to provide a check on the human's decisions. Thus, when both components of the system are operational the human makes decisions and the hardware confirms their validity or indicates an error in operating procedures has been made. If an error has been made the decision is not propagated further in the system. In this way the human errors are detected *before* an improper operational condition can exist. If, however, the hardware component has failed, there is no check on the human's decisions

TIME (HOURS)	(1 - c) FRACTION OF NEAR-COINCIDENT FAILURES
1.95	6.0140E-7
3.90	6.1480E-7
7.81	6.2150E-7
15.62	6.2485E-7
31.25	6.2652E-7
62.50	6.2736E-7
125.00	6.2778E-7
250.00	6.2799E-7
500.00	6.2809E-7
1000.00	6.2815E-7

Value of (1-C) for the WIU as a function of time
Table 4.1

and the system is vulnerable to the human errors. In the derivation of the human error rate (Section 5.1.5) only accident causing errors are considered. Therefore, a human error that is not caught by the hardware will propagate into the system and create an accident condition.

A Markov model of this scenario is shown in Figure 4.4a. Both the hardware and human are operational in state 1. In this state there is no impact from human errors since the hardware prevents their entering the system. Hence, the only exit from this state is that of a hardware failure. It is assumed that the hardware fails at the rate λ fails per hour and the hardware is repaired at the rate μ repairs per hour. State 2 represents the condition where the hardware has failed. Besides the hardware repair action, the other possible exit transition from state 2 is due to the now unchecked human error. These errors occur at the rate **H** and lead to state 3. Any time state 3 is entered an accident situation exists. The accident rate in this system is the number of times per unit time that state 3 is entered.

The unique aspect of the human is that it is "instantaneously repaired". This is reflected in the repair rate μ_H. Instantaneous repair implies a zero repair time or a repair rate of infinity. The implications of this repair rate are that any probability that enters state 3 is instantly transferred back into state 2. Examining the differential equation for state 3 confirms this:

$$dP_3(t) / dt = H\, P_2(t) - \mu_H\, P_3(t) \qquad (11)$$

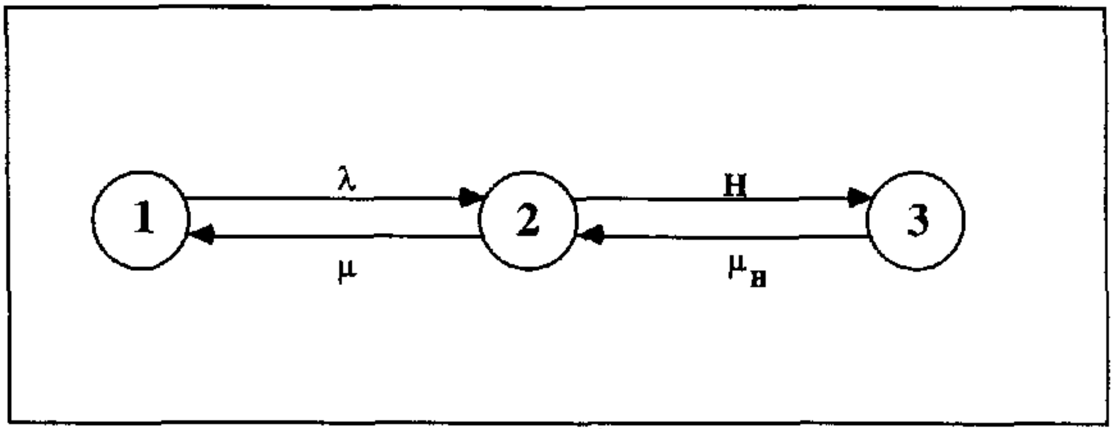
where $\mu_H = \infty$. The slightest nonzero contribution to P_3 causes a large negative term in the equation that instantly reduces the probability of state 3 back to 0.

This peculiar behavior of state 3 provides the means to obtain the number of times the state is entered per unit time. Any probability that goes from state 2 into state 3 is instantly returned. Therefore, state 3 can be eliminated from the model with no impact on the probabilities of states 1 and 2. Further, the rate at which state 3 is entered is just:

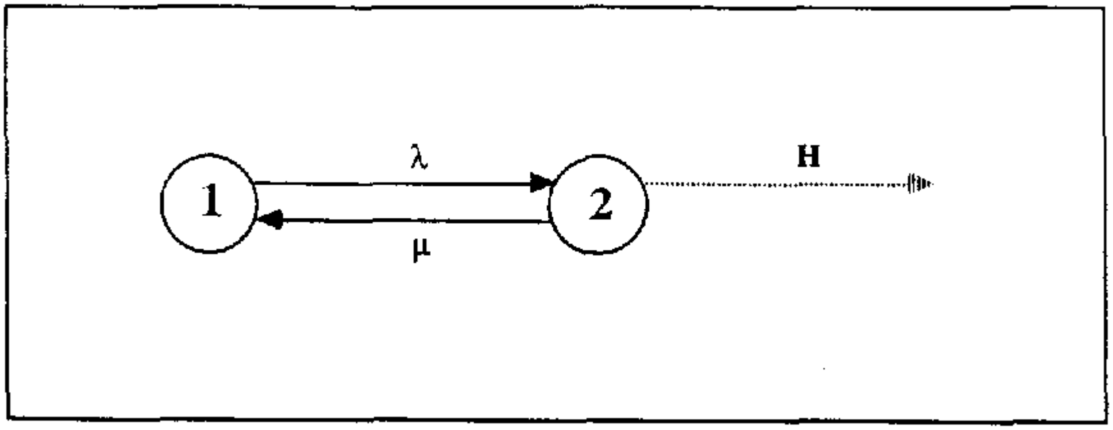
$$\text{rate}(t) = H\, P_2(t) \qquad (12)$$

Thus a Markov model that includes states 1 and 2 is solved for P_2 at a specified t. This result is multiplied by the transition rate **H** to obtain the rate at which, the now virtual, state 3 is entered. This model is shown in Figure 4.4b.

Note that the Markov model in Figure 4.4b does not actually contain the exit transition **H** from state 2. This is a "virtual transition" used to calculate the rate at which errors are generated from state 2. Equation (12) does represent a rate since it is the product of the rate at which humans generate errors times the probability that the system is in a state where the human error can propagate into the system. Contrast this situation with that of the current control system where system operation is *always* vulnerable to the human error



Human / Hardware Model
Figure 4.4a



Accident Rate Model with Virtual Transition H
Figure 4.4b

rate. It is this reduction in the exposure time to human errors as indicated by the probability of state 2 that provides for the reduction in the accident rate for ARES operations.

4.2.3 Modeling Components of One Type —The Chain Submodel

The primary technique used to mitigate the state proliferation problem in the ARES Markov model is the decomposition into submodels for each component type. For example, there is one submodel to account for the behavior of all of the WIUs (wayside interface units) and another submodel for all of the VHF GTCs (ground terminal controllers). In this section the decomposition process is discussed and the prototype submodel, called the chain submodel, is constructed. The independence of the chains, or orthogonality, is demonstrated. Finally, the merging process that combines submodels back into a complete ARES model is shown.

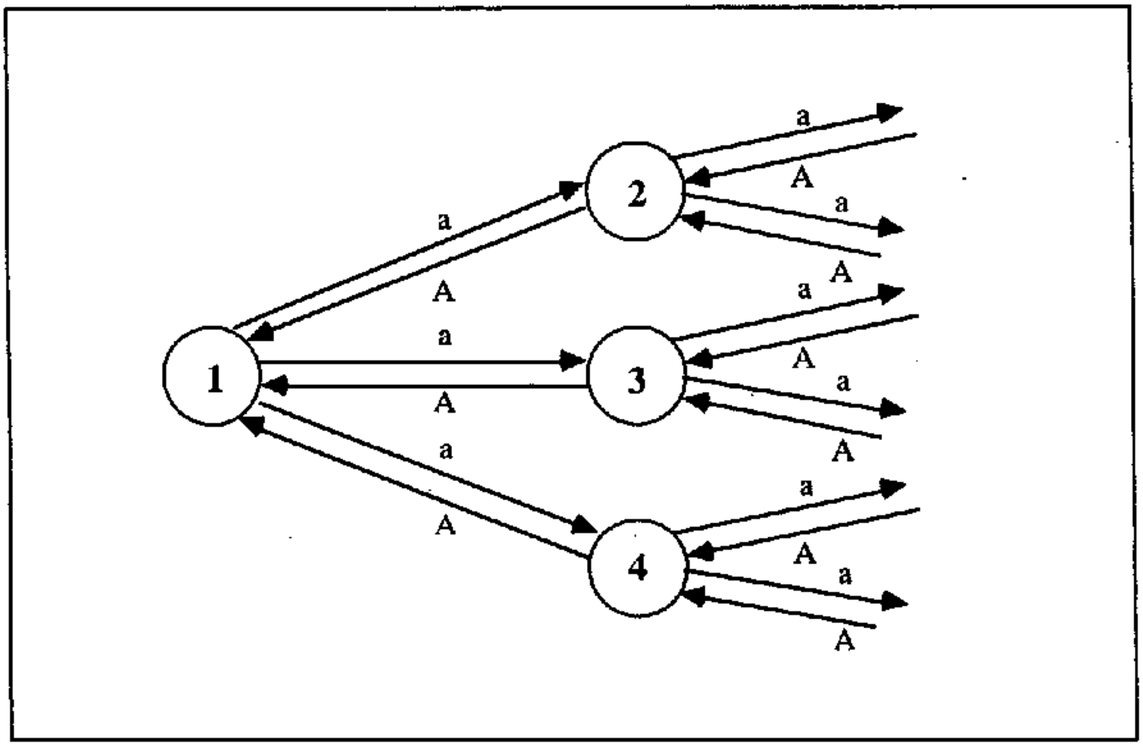
4.2.3.1 Decomposing into Submodels

The decomposition of the ARES model into submodels takes advantage of the relative independence of most of the component types. For example, the failure of a VHF GTC (ground terminal controller) results in a segment of track where trains are not in constant contact with the ROCS. Special operating rules must be applied to get trains through this segment until a repair is made. Further, trains on this segment have lost the remote intervention capability exposing them to operator errors. This condition is true for this segment independent of failures of on-board train equipment such as the TBI (throttle-brake interface). Notice, however, that the loss of two VHF GTCs means that there are now *two* segments where train communication is lost . This exposes the system to two places where operator errors cannot be corrected before an accident situation occurs.

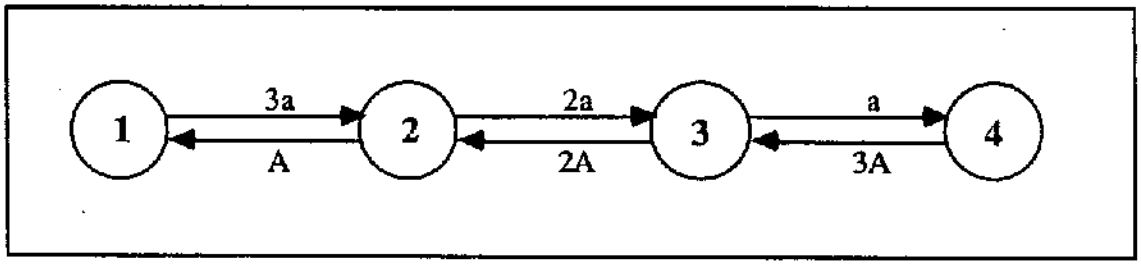
Using the above scenario, the ARES model is decomposed into submodels that contain all components of one type. There is one submodel for all of the WIUs, one for the VHF GTCs, etc. The deciding factor in selecting submodels is if there is the independence of the impact of failures on the system operations. Thus accident contributions will be accounted for in each submodel and the results merged to provide the accident rate for the complete ARES.

4.2.3.2 The Chain Submodel

The most common submodel used in ARES has a pictorial form that has led to its name of a "chain submodel". Consider the first failure level of a submodel of 3 components of type A (Figure 4.5a). At the first failure level there are three states representing the single failure (at rate **a**) of each of the three components. The exit transitions from each of these states shows the failure of the remaining two components or the repair of the single failed component (at repair rate **A**).



Three-Component Model - First Failure Level
Figure 4.5a



Three-Component Chain Model
Figure 4.5b

The system implication in each of the single-failure states is that one location or train is missing a certain capability. Thus, if there is no concern for which *specific* location or train has lost this capability then these three states can be aggregated into one. This aggregation does not introduce any approximations (see Section 4.1.2 and Reference 1, Section 6.2).

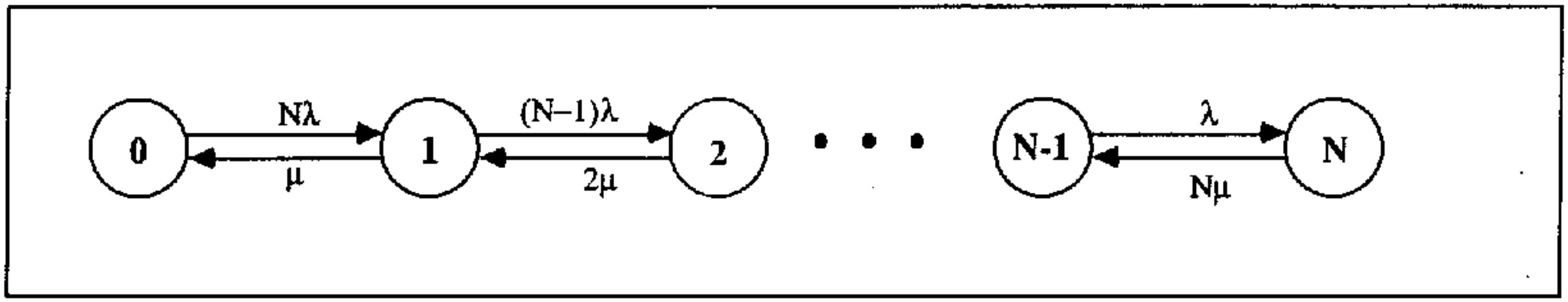
The resulting model is shown in Figure 4.5b where the states at each failure level have been aggregated. State 1 contains the configuration where no components have failed. The failure of any one of the 3 components, a transition rate of 3a, causes a transition into the state where one component has failed (state 2). In state 2 the failed component can be repaired (at rate A) or one of the remaining 2 components can fail, leading to state 3 at the rate 2a. State 3 is the aggregate of all configurations where two components have failed. From this state, one of the two failed components may be repaired (rate 2A) or the remaining component may fail. State 4 represents the failure of all three components. Any of the three components may be repaired (rate 3A) from state 4 causing a transition to state 3.

Figure 4.5b has the form of a chain. Notice that the failure of any of three components (the transition from state 1 to 2) is simply the sum of failing component 1 *or* component 2 *or* component 3. It is the uncertainty of precisely which component will fail that indicates the OR operation is needed, resulting in a sum of failure rates. The repair transition from state 4 to state 3 is 3A. Similarly, this implies an uncertainty in exactly which component is being repaired. This could only be true if there were three repairmen all doing the repairs at the same time.

The chain model has this simple form since all of its components have the same failure rate and the same repair rate. The implications on system operation are only dependent on the number failed, not which specific components are failed. Further, the chain model of Figure 4.5b contains an assumption of at least three repairmen. Note that the aggregation from individual states at each failure level to one state at each failure level accomplishes a substantial reduction in the state space.

Consider now the chain model for a component type that has N components in ARES. This results in the model shown in Figure 4.6. The failure rate for each component is λ and the repair rate is μ. The states are numbered *from 0 to N* so the state number corresponds to the number of failed components in that state. A typical value for N in ARES is 100. The chain model would therefore have 101 states, which is not an intractable number of states.

The Markov model in Figure 4.6 can be solved numerically for the steady state probabilities (Section 4.1.2). In this special case, a closed-form solution for the steady state probability of state i exists:



N-Component Chain Model
Figure 4.6

$$P(i) = \frac{1}{[1 + (\lambda/\mu)]^N} \binom{N}{i} (\lambda/\mu)^i \qquad i = 0,1,2, \ldots , N \qquad (13)$$

Notice that the probability for state i only depends on i, N, and the ratio (λ/μ); the specific magnitudes of λ and μ do not enter the equation. As will be shown, this is a direct consequence of the steady state condition: flows between states must be equal to maintain the steady state.

Figures 4.7 and 4.8 show the probability of i failures for various ratios of (λ/μ). Figure 4.7 is the case where there are 500 components (N = 500) and Figure 4.8 is for a chain of 1000 components. Examining these plots shows the probability of more components failed dropping off rather steeply, particularly for the smaller (λ/μ) ratios. Is this always the case?

If the value of N is greater than μ/λ, then a state where one or more components are failed will be the most probable. This can be derived from Figure 4.6. In steady state the flows into a state must equal those leaving the state. The transition from state 0 to state 1 has a rate of $N\lambda$ in one direction and a rate of μ in the other. Setting the flows between states 0 and 1 equal for the steady state gives:

$$P(0)\, N\, \lambda = P(1)\, \mu \qquad (14)$$

Therefore, if $N\lambda > \mu$ (or $N > \mu/\lambda$), P(1) must be greater than P(0). A similar calculation can be done to find the condition for P(2) > P(1), etc. Note in Figure 4.8 that for the ratio $(\lambda/\mu) = 10^{-3}$ the probabilities for states 0 and 1 (0 and 1 component failed, respectively) are identical. This must be so as an application of Equation (14) shows for $(\lambda/\mu) = 10^{-3}$ and N = 1000. .

One more property to observe is the effect of changing the number of components. For example, if the number of components is doubled from N to 2N, will the probability of 1 of N failed be the same as the probability of 2 of 2N failed? The probability of having one component of N failed is:

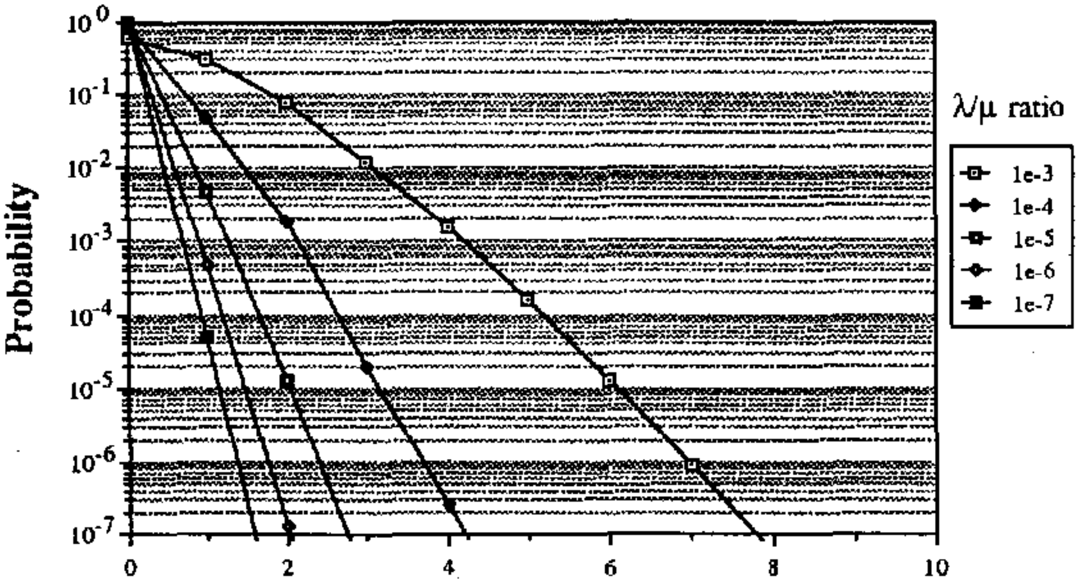
$$P_N(1) = [1 + (\lambda/\mu)]^{-N}\, N\, (\lambda/\mu)$$

and the probability of having 2 of 2N components failed is:

$$P_{2N}(2) = [1 + (\lambda/\mu)]^{-2N}\, 2N\, (2N-1)\, (\lambda/\mu)^2 / 2$$

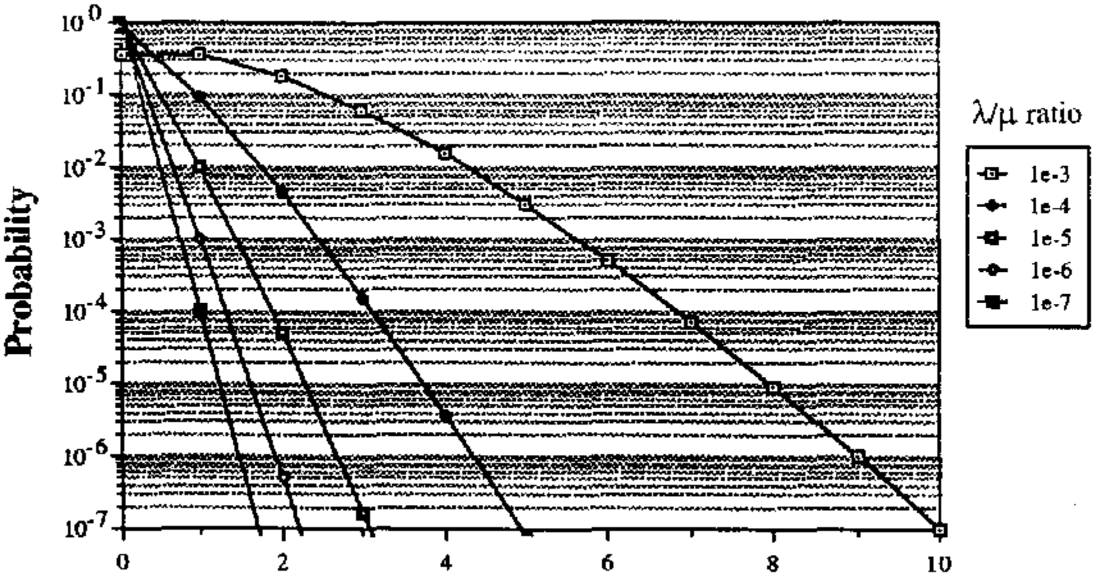
It can be shown that $P_N(1) > P_{2N}(2)$, which is equivalent to proving:

$$[1 + (\lambda/\mu)]^N > (2N-1)\, (\lambda/\mu)$$



Number of Components Failed out of 500

Figure 4.7



Number of Components Failed out of 1000

Figure 4.8

For N >> 1 this can be written as:

$$[1 + (\lambda/\mu)]^N > 2N\,(\lambda/\mu) \tag{14}$$

Consider the case where $1 > N(\lambda/\mu)$. The following series of equations can be made:

$$\begin{gathered} 1 > N(\lambda/\mu) \\ \mu/\lambda > N \\ \mu/\lambda + N > 2N \\ 1 + N\lambda/\mu > 2N\,(\lambda/\mu) \end{gathered}$$

Using:

$$[1 + (\lambda/\mu)]^N > 1 + N\lambda/\mu$$

we arrive at Equation (14):

$$[1 + (\lambda/\mu)]^N > 2N\,(\lambda/\mu)$$

(A more complex proof exists for the case where $1 < N(\lambda/\mu)$.) This demonstrates that the state probabilities do not scale linearly. In fact, the probability of failing i of N components is *greater* than that of failing 2i of 2N components.

In the case of a 100 component chain, it is possible to model all 101 states. But is this necessary? It seems unlikely that all 100 components would be failed at a given time. Thus, the probability of state 100 may be sufficiently insignificant that it can be ignored. To find out if, in fact, any of the states can be ignored without impacting the numerical integrity of the solution, the state probabilities are examined.

Using the values $N = 100$, $\lambda = 10^{-4}\ \text{hr}^{-1}$, and $\mu = 10^{-1}\ \text{hr}^{-1}$, the first few state probabilities are:

$$\begin{aligned} P(0) &= 9.05\text{x}10^{-1} \\ P(1) &= 9.05\text{x}10^{-2} \\ P(2) &= 4.48\text{x}10^{-3} \\ P(3) &= 1.46\text{x}10^{-4} \\ P(4) &= 3.55\text{x}10^{-6} \\ &\bullet \\ &\bullet \\ &\bullet \\ P(10) &= 1.57\text{x}10^{-17} \\ P(11) &= 1.28\text{x}10^{-19} \end{aligned} \tag{15}$$

The state probabilities continue to decrease to the probability of being in state 100, which is 10^{-300}. In this case, the most probable state for the system is to have no components failed

(P(0)). The probability of one component failed (P(1)) is approximately 9%. The probability of *two or more* components failed is less than 0.5%. Clearly state 100 with its probability of 10^{-300} can be ignored. Further, all states after state 11 contribute less than 10^{-19} to the system probability. It seems reasonable that these can also be ignored.

The process of ignoring insignificant states is called model truncation. A requirement for truncating a chain is that the state probabilities must be dropping off for each subsequent state in the truncated part of the model. Since state probabilities drop off at an increasing rate, once they start decreasing, this point is easy to determine. A measure of the error introduced by the truncation can be obtained by examining a model with one more state. For example, if one truncates a model at state M, comparing the results to a model truncated at M+1 would provide a measure of the error introduced by the truncation. Figure 4.9 shows an N-component model truncated at the M^{th} failure.

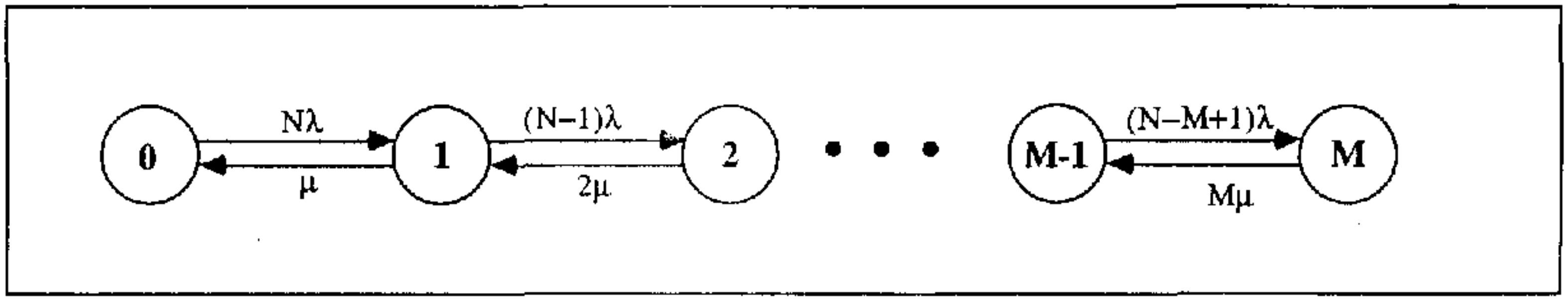
Returning to the example of a 100 component chain (Equations (15)), if one is interested in an accuracy of three significant digits for the probability that one or more components are failed, the model can be truncated at the third failure level since state 4 contributes in the 5th significant digit. In other words, the last modeled state is state 3. This truncated model has the state probabilities:

$$\begin{aligned} P(0) &= 9.05\text{x}10^{-1} \\ P(1) &= 9.05\text{x}10^{-2} \\ P(2) &= 4.48\text{x}10^{-3} \\ P(3) &= 1.46\text{x}10^{-4} \end{aligned}$$

The total probability of *one or more* components failed in this truncated model is $9.51\text{x}10^{-2}$ which differs from the full model (Equations (15)) in the third significant digit. Comparing the model truncated at the third failure level to a model truncated at state 4 also shows an error in the third significant digit.

A point that has been ignored in this analysis is that these chains assume there are N repairmen. It is unlikely that there are 100 repairmen assigned to a component type for which there are 100 components. The impracticality of this is indicated by the probability that all 100 would be failed at once, and thus require a force of 100 repairmen. The lack of 100 repairmen does not invalidate the chain model. It has been shown that the probability of most of the components being failed can be ignored and have no measurable impact on the model's results. Thus, the lack of sufficient repairmen for the cases where most components are failed similarly does not impact the results. All chain submodels used in ARES can be truncated at such low failure levels that providing sufficient repairmen is not a problem.

To summarize, the ARES model is divided into submodels to reduce the intractable size of the state space. The most common submodels deal with collections of components of one type. A complete model of a group of 100 components would take 10^{30} states.



N-Component Chain Model Truncated at State M
Figure 4.9

Since no distinction is required of which component is failed, i.e., only the number of failures is needed, the states at each failure level are aggregated into one state. This reduces the number of states to 101 without any approximations. The resulting model is called a chain model and has some interesting properties in the steady state. The majority of the states in the chain do not contribute significantly to the solution and can be ignored through model truncation. This truncation usually reduces the number of states to less than 10. Measures of the approximation introduced by the truncation are easily obtained.

4.2.3.3 Submodel Independence

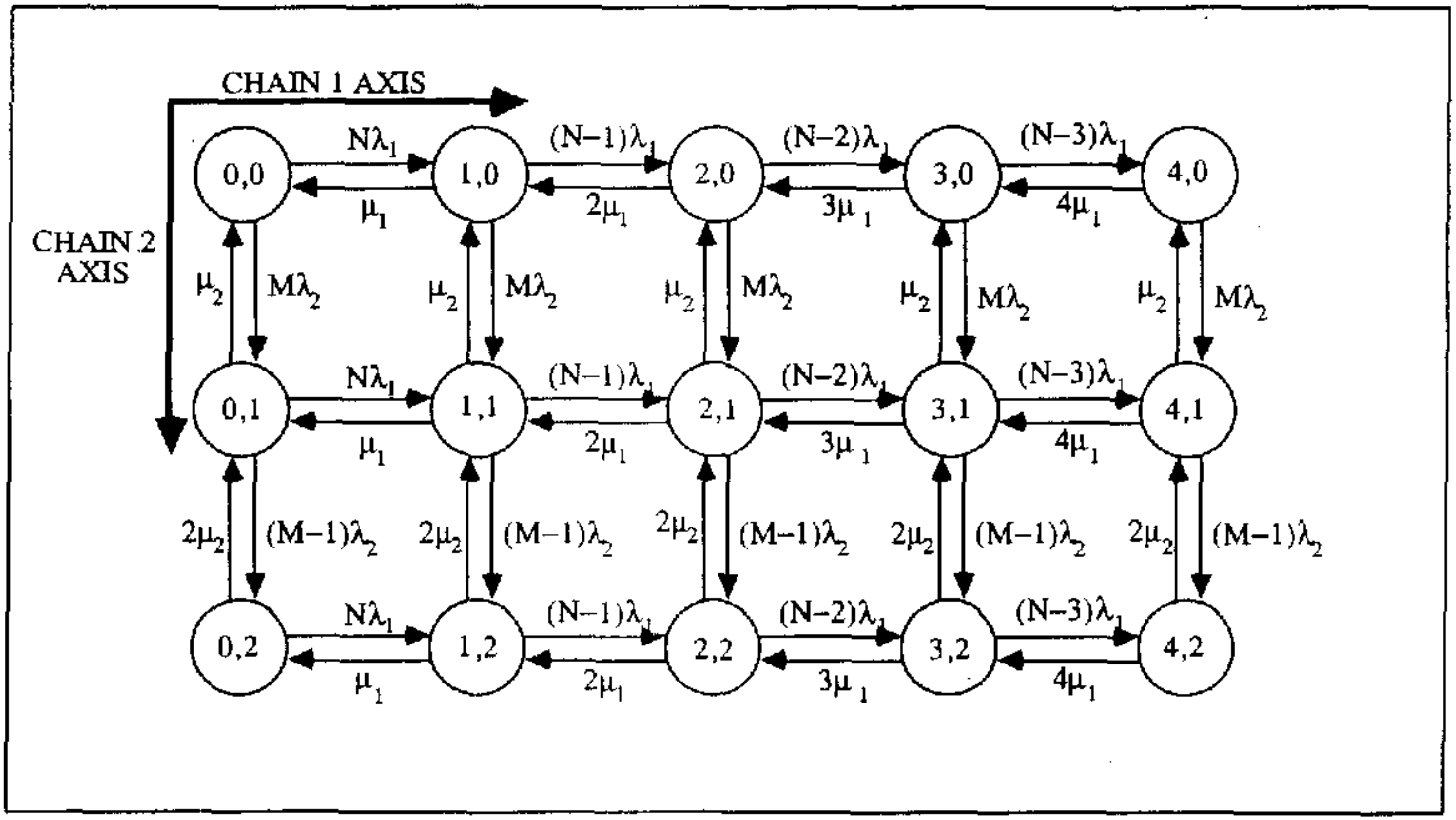
Decomposing the ARES model into submodels such as chains is only productive if the submodels can be solved individually and there is an efficient means of merging the results. These issues resolve around the nature of the independence of the submodels.

Consider two submodel chains. Chain 1 has N components with each component having a failure rate of λ_1 and a repair rate of μ_1. This chain is truncated at the fourth failure level. Chain 2 has M components with each component having a failure rate of λ_2 and a repair rate of μ_2. This chain is truncated at the second failure level.

Figure 4.10 shows a system that is composed of these two chains. Chain 1 has been truncated at four failures and chain 2 has been truncated at two failures. The state numbers are ordered pairs: "0,2" indicates a state where there are no failed components in chain 1 and two failed components in chain 2. Starting from the no failure state, "0,0", the possible transitions are that one of N components in chain 1 can fail or one of M components in chain 2 can fail. The diagram has a structure where failure and repair events concerning chain 1 are on the horizontal axis and those of chain 2 are on the vertical axis. Further, the same transitions appear repeatedly along any horizontal or vertical line. For example, the transition between states "2,0" and "3,0" is the same as those between "2,1" and "3,1", as it is between "2,2" and "3,2". These two chains are called orthogonal because of the overall symmetry and limitation of chain 1 events to one axis and chain 2 events to the other axis. The orthogonality of the chains provides a simple means of merging their results.

Chain 2 can be in states 0, 1, or 2. Notice that the chain 1 events of failing one of N components (transition $N\lambda_1$) or repairing a single failed component (transition μ_1) are potential events from all possible configurations of chain 2. Due to the orthogonality of the chains, this is true for all chain 1 events; they occur in *every* state in chain 2. Thus, it seems reasonable that chain 1 could be modeled independently of chain 2. Chain 1's probability of having no components failed would be distributed over states "0,0", "0,1", "0,2" based on the ratios of states 0, 1, 2 in chain 2. A similar procedure would be applied for all failure levels of chain 1. Thus, all states in the model of Figure 4.10 could be found.

To test this technique, the model in Figure 4.10 was solved for the steady state. The following values were used:



Full Two-Chain Model
Figure 4.10

$N = 1000$	$M = 100$
$\lambda_1 = 3.25x10^{-4}$	$\lambda_2 = 2x10^{-4}$
$\mu_1 = 0.1$	$\mu_2 = 0.1$

Figure 4.11 shows the model with state probabilities inside each state symbol (circle). These are the result of simulating the full two-chain model. Across the top of the model diagram are the state probabilities determined by solving chain 1 alone. Similarly, the state probabilities found by solving chain 2 alone are along the left-hand side of the diagram.

Taking the product of any of the probabilities of chain 1 (top margin of the diagram) with one of the probabilities of chain 2 (left margin of the diagram) gives the probability of the joint event (in the circle representing the state), which was derived from a full two-chain model. For example, the independent solution of chain 1 shows a probability of two components failed of 0.27. The independent solution of chain 2 shows a probability of no failures of 0.82. Therefore, the probability of the state in the two-chain model where there are two failures in chain 1 and none in chain 2 is the product (0.27) (0.82) = 0.22. As expected, this is the probability of state "2,0" from the two-chain solution. (Roundoff errors account for the minor discrepancies of the products and true solutions in Figure 4.11.) Therefore, it is possible to obtain the probability of any state in the two-chain model without ever solving the two-chain model. Instead, each chain is solved independently and the appropriate products are taken to determine the two-chain state probabilities.

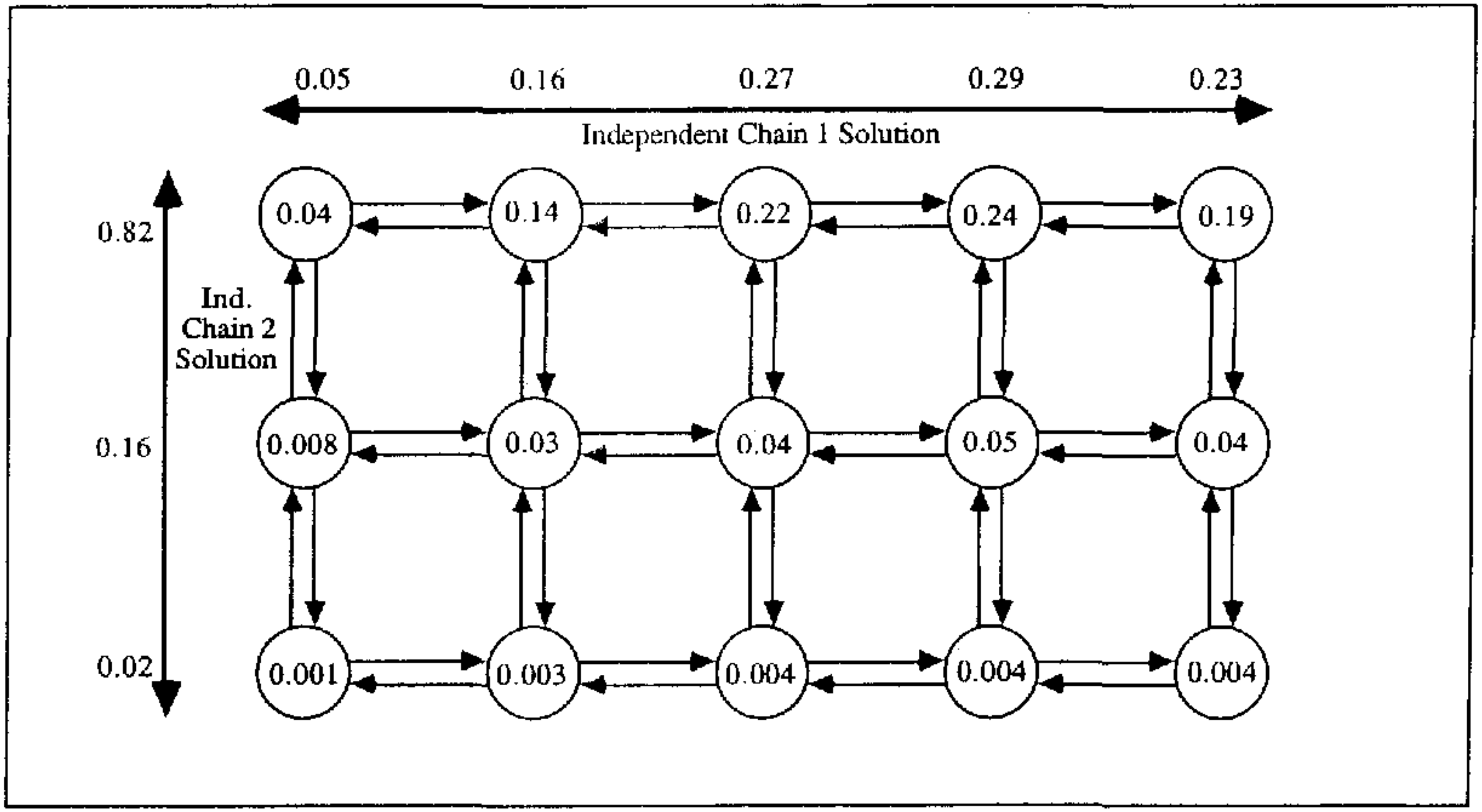
An intuitive understanding of this merging of independent solutions to obtain the two-chain model solution can be obtained by recalling that a system in steady state has flows into a state equal to those out of the state (see Section 4.2.3.2). Thus the ratio of states "0,0" and "1,0" must be the same as those of "0,1" and "1,1" since they have the same transitions connecting them. Further, the independent model of chain 1 has the same connections between its states 0 and 1. Thus, the ratio between the 0 and 1 failure states in chain 1 is the same for any number of failures in chain 2. This can only be true if the two-chain model probabilities are the product of the independent chain probabilities.

Often we are not interested in the probability of any specific state in the two-chain model but, rather, the probability of a group of states. For example, if there is a particular virtual transition to an accident potential associated with the failure of one component in chain 1 then we want the sum of the state probabilities for "1,0", "1,1", "1,2". However, this sum is just the probability of one failure in the independently solved chain 1 model. The two-chain model probabilities are the product of the independent chain solutions. Using subscripts to indicate the independent chain number, the total probability in the two-chain model of one failure in chain 1 is:

$$P_1(1)P_2(0) + P_1(1)P_2(1) + P_1(1)P_2(2)$$

or:

$$P_1(1)\ [P_2(0) + P_2(1) + P_2(2)]$$



Full Two-Chain Model Solution with Independent Chain Solutions
Figure 4.11

yet:

$$[P_2(0) + P_2(1) + P_2(2)] = 1$$

Therefore:

$$P_1(1)P_2(0) + P_1(1)P_2(1) + P_1(1)P_2(2) = P_1(1)$$

The total probability of having one component failed in chain 1 in the two-chain model is just the probability of failing that one component in the isolated chain 1 model.

The orthogonality of the chains says that each event in one chain occurs in *every* state in the other chain. Thus, the failure of one component in chain 1 is a potential event in every state in chain 2. The probability of being in chain 2 is equal to 1; there must be either 0, 1, or 2 components failed in chain 2. The two-chain model's probabilities for states where one component in chain 1 is failed are just the probability of one failure in chain 1 distributed over the probabilities of chain 2. Thus, the total must be simply the original probability of one failure in chain 1. The property that an event in one chain occurs in every state in the other chain plays a key role in the merging submodels to determine accident rates for ARES (Section 4.2.4).

The success of this decomposition method provides an impressive reduction in the size of the state space. In this simple example, the two-chain model had 15 states while the solution of the two independent chains required only 5 and 3 states, respectively. In ARES there are 8 chains, each truncated at approximately the tenth failure level. This 8-chain model would require more than 10^8 states while solving the chains independently uses 8 models with 11 states.

The above discussion is easily generalized for more than two chains; however the diagram of an 8-chain model is not particularly enlightening. For a case with 8 chains, the probability of being in one specific state in the full 8-chain model is the product of the appropriate states from the independent solution of the 8 chains. The total probability in the 8-chain model of a particular event in one chain is just the probability of that event in the independent solution of the one chain. Although most are, not all submodels in ARES are of the chain type. The other models are of a more "traditional" form such as those shown in Section 4.1. These models are orthogonal to the chains so all of the rules for merging chain results also apply to a full model comprised of traditional and chain submodels.

Dividing the system into independent submodels, in this case the submodels tend to be chains, is a powerful technique for simplifying the analysis of a system. Decomposing a system model into submodels is only productive if the submodels can be solved individually and there is an efficient means of merging the results. As has been shown, if the submodels capture the failure and repair of components that are independent from the other chains, then the chains are orthogonal and they may be solved independently. Probabilities of events in the full model are simply products and sums of the individual chain probabilities. The ARES model has the orthogonality property that permits its division into independent submodels.

4.2.4 The Complete Model

All of the constituents of the ARES model have been examined in Sections 4.2.1 to 4.2.3. In this section, the merging of these techniques will be shown. Once again, the focus is on chain submodels, although the results apply to any type of submodel.

Figure 4.12 shows an N-component chain submodel that has been truncated at the M^{th} failure level. The component failure rate is λ and the repair rate is μ. From each state there is a virtual transition V_i, where i represents the state number at the origin of the transition. All accident situations have a similarity to the generation of errors by a human: both involve instantaneous repair. When an accident occurs, the system (or the parts of the system not involved in the accident) continues to operate. This "instantaneous repair" requires a virtual transition to capture its behavior. Thus, virtual transitions will be used to represent events that cause potential accident conditions to exist. Recall that the virtual transitions are not used in solving the Markov model for the state probabilities P(i). Once the state probabilities have been obtained for the steady state (the condition that is appropriate for ARES), the accident rate A_j for chain j is determined as:

$$A_j = P(0)V_0 + P(1)V_1 + \bullet\bullet\bullet + P(M)V_M \qquad (16)$$

The units for V_i are hours^{-1} so the accident rate, $A_{j,}$ is in units of accidents per hour. This is easily converted into a yearly accident rate.

Consider a component that has vital internal elements that have been dualized such as the wayside interface unit (WIU). The component can create an accident potential by having an uncovered failure. The uncovered failure occurs with a rate $\lambda(1\text{-c})$, where c is the coverage derived in a coverage model (see Section 4.2.1). Thus in state 0, where no components have failed, the system is vulnerable to any of the N components having an uncovered failure. Therefore, the virtual transition from state 0 is:

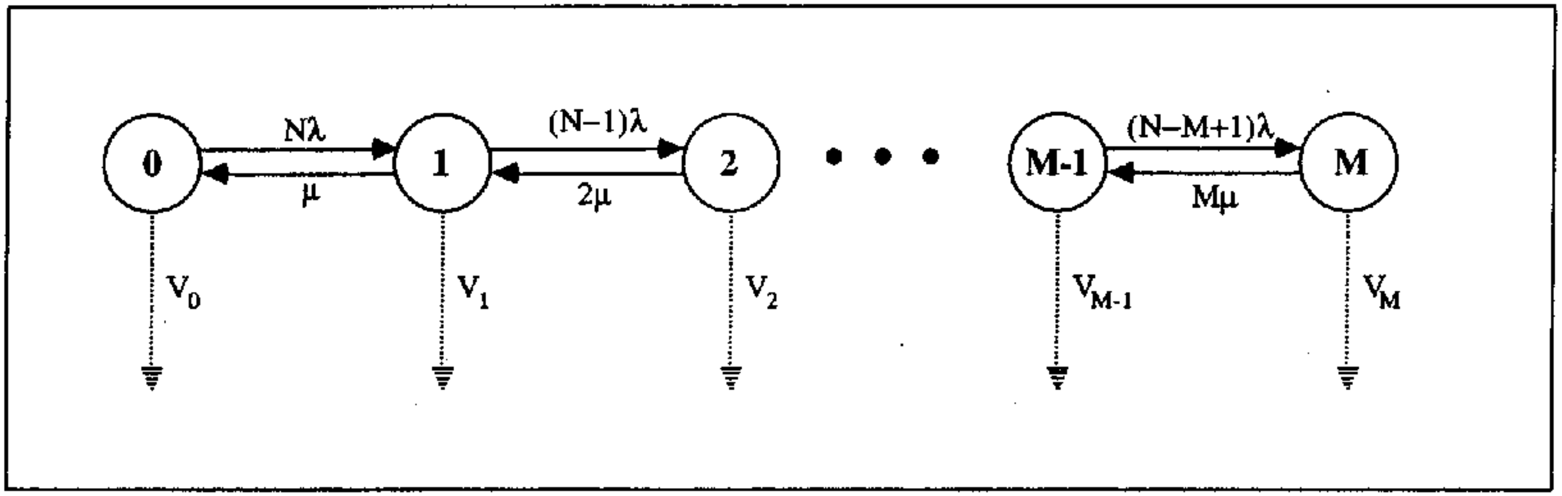
$$V_0 = N\lambda(1 - c)$$

State 1 has N-1 components operating. Thus, the system is vulnerable to N-1 uncovered failures from this state:

$$V_1 = (N - 1)\lambda(1 - c)$$

The virtual transitions exiting the other states are calculated in the same way. Using Equation (16) the total accident rate as a result of this chain is:

$$A_{WIU} = P(0)\,N\lambda(1\text{-}c) + P(1)\,(N\text{-}1)\lambda(1\text{-}c) + \bullet\bullet\bullet + P(M)\,(N\text{-}M)\lambda(1\text{-}c)$$



Truncated N-Component Chain Model with Virtual Transitions
Figure 4.12

Consider now a chain where the components do not have uncovered failures due to hardware operation, but the failure of a component exposes the system to human errors. For example, the VHF ground terminal controller (VHF GTC) is used to maintain communication between ROCS and trains on a segment of track. For simplicity, it is assumed that there is only one train on each GTC track segment. The failure of a GTC removes the remote intervention ability, eliminating the possibility of preventing accidents due to train operator error. Further, the ROCS and the dispatcher (for now it is assumed that there is only one dispatcher for the region) now do not have updates on the train status and position, exposing the system to dispatcher errors.

In state 0 there are no GTCs failed so there is no exposure to human errors. Therefore $V_0 = 0$. In state 1 there is one GTC failed so the system is exposed to two humans, the dispatcher and train operator, that generate errors at the rate H. The virtual transition rate leaving state 1 is:

$$V_1 = 2H$$

In state 2, two GTCs have failed removing communication from two segments of track. The system is exposed to three humans: the dispatcher and two train operators. The virtual transition rate leaving state 2 is:

$$V_2 = 3H$$

The virtual transitions exiting the other states are calculated in the same way. Using Equation (16) the total accident rate as a result of this chain is:

$$A_{GTC} = P(1)\,2H + P(2)\,3H + \bullet\bullet\bullet + P(M)\,(M+1)H$$

Detailed derivation of the transitions to accident states for each ARES submodel can be found in Section 5.2.

Having derived the accident rate for a chain submodel, what is the total accident rate for a multi-chain model? Recalling the discussion of the two-chain model in Section 4.2.3.3, the total probability of having a given number of components failed in chain 1 in the two-chain model is just the probability of failing that number of components in the isolated chain 1 model.

Consider chain 1 to be the VHF GTC chain. The state with 1 component failed has a virtual transition of 2H so its contribution to the accident potential in an isolated chain is:

$$P_{GTC}(1)\,2H$$

where the subscript GTC identifies the chain. The orthogonality of the chains says that each event in one chain occurs in *every* state in the other chains. Thus, the failure of one component in the GTC chain is a potential event in every state in the other chains. Since the probability of being in a chain is equal to 1, $P_{GTC}(1)$ represents the total probability of that event in the multi-chain model. Similar reasoning is applied to the other chains in the multi-chain model. Thus, the total accident rate in the multi-chain model is the sum of the individual chain's accident rates:

$$A_{TOTAL} = A_1 + A_2 + \bullet\bullet\bullet + A_K \tag{17}$$

where there are K chains in the complete model. The individual chain accident rates are derived from isolated chain submodels. It is possible to confirm this formula for a two-chain model using the results in Figure 4.11 with the virtual transitions derived above for the WIU and GTC. This merging technique applies to submodels that have forms other than a chain. The success of this formula for merging submodel results rests on the orthogonality of the submodels.

To summarize, the submodels are created in such a way that the orthogonality principle holds. The models are solved independently using a numerical integration for the steady state probabilities of occupying each state. Virtual transitions reflecting events that lead to a potential accident condition are defined for each chain's states. Each chain contribution to the accident rate is calculated using Equation (16). The total ARES accident rate is obtained by merging the results using Equation (17).

5.0 THE ARES SAFETY MODEL

This section describes the model in detail. First, the nature of the inputs is discussed and their baseline values are enumerated. Next, the logical structure of the model is developed. Then, the baseline results are presented, followed by an analysis of the sensitivity of these results to variations in the modeling assumptions and uncertainties in the input values.

5.1 Inputs to the Model

The inputs to the model supply information about the failure rates of individual ARES hardware elements, the expected times that failed elements are out of service before they are repaired or replaced, atmospheric effects on microwave communications, data transmission errors due to noise, and the occurrence of critical human errors. The values used for the baseline ARES safety model are given in Tables 5.1a and 5.1b. The meaning of these parameters is described below.

5.1.1 Hardware Failure Rates

The characteristic reliability of a given (non-redundant) hardware component type can be expressed in terms of the mean of an exponential failure distribution function (mean time before failure or MTBF). If a particular component is repaired immediately or replaced each time it fails then the average interval between failures is the MTBF. Similarly, if one starts out with a new set of like components, then the average of the times to first failure of each component is also the MTBF. The reciprocal of MTBF is the failure rate for a component of that type, usually expressed as failures per hour.

5.1.2 Repair Rates

The mean time to repair (MTTR) of a particular component includes both the time to actually perform the repair or replacement once a repairman arrives at the site, and the actual time it takes the repairman to get to the site after the failure has been detected. The reciprocal of MTTR expresses the repair statistic as a rate.

The concept of MTTR (and repair rate) is extended in two ways. The first applies to those cases where full integrity of the system is restored by *removing* an inoperative element. Consider that when the VHF data radio on-board a train fails, the train continues to run under voice radio blocking rules. The full integrity of ARES is restored when the train gets off the main track onto a siding or into a yard. Actual repair of the failed equipment can take place any time thereafter. The actual repair time is not relevant because the model is predicting system safety, not operational availability. The second applies to the restoration of a service loss to the system. The average duration of communication dropouts between microwave towers due to atmospheric conditions fits the form of an MTTR.

Name	Description	MTBF or MTTR (in hours)	λ or μ (hrs^{-1})
LARC 400	Locomotive Analysis and Reporting Computer	15000	6.67 E-5
LSP 400	Locomotive Analysis and Reporting System Sensors (20)	3000	3.3 E-4
LIM 400	Locomotive Interface Monitor	15000	6.67 E-5
LIS 400	LIM sensors (6)	10000	1.0 E-4
DMU 400	Data Management Unit	11000	9.09 E-5
TPS 400	Power Supply (two per locomotive)	7000	1.43 E-4
DCR 400	Card Reader	50000	2.0 E-5
TDP 400	Printer	12000	8.33 E-5
RDT 400	VHF radio	5000	2.0 E-4
NAVCOR	GPS receiver	9300	1.07 E-4
GPA 400	GPS preamps and antenna	20000	5.0 E-5
EMC 400	Energy Management Computer	9600	1.04 E-4
TSI 400	Train Situation Indicator (primary display to engineer)	5600	1.79 E-4
CDU 400	Backup display unit	15200	6.58 E-4
TBC 400	Token Enforcement Computer (part of TEU)	20000	5.0 E-5
TBU 400	Throttle Brake Isolation Box (part of TEU)	10000	1.0 E-4
TBA 400	Throttle Brake Actuator (part of TEU)	4000	2.5 E-4
EOT 400	End of Train interface	60000	1.67 E-5
WIU 400	Wayside Interface Unit (does not include sensors or actuators)	8000	1.25 E-4
GTC 400	Ground Terminal Unit (does not include VHF radio)	8000	1.25 E-4

MTBF => Mean Time Between Failures λ = 1/MTBF => Failure Rate

MTTR => Mean Time To Repair μ = 1/MTTR => Repair Rate

Failure Rates and Repair Times for ARES Components
(Supplied by Rockwell)

Table 5.1a

Name	Description	MTBF or MTTR (in hours)	λ or μ (hrs^{-1})
Tach	Tachometer (including A/D conversion, etc.)	30000	3.33 E-5
ROCS	Regional Operations Control System	2000	5.0 E-4
NCS	Network Control System	2000	5.0 E-4
MicroT	Microwave transmitter	5000	2.0 E-4
MicroR	Microwave receiver	5000	2.0 E-4
MicroC	Combiner for microwave receiver signals	1000000	1.0 E-6
Comm loss	Loss of communication between microwave towers due to atmospheric conditions	2924	3.42 E-4
Xmit error	Undetected transmission error over the network (ROCS to train or ROCS to WIU)	1.0 E+9	1.0 E-9
Arrive	Rate of train arrival at a switch	0.25	4.0
Human	Human error rate	100000	1.0 E-5
μMaint	Scheduled maintenance interval (for TEU, etc.)	5000	2.0 E-4
μTrain	Repair rate for on-train equipment (time to clear train from track)	5	2.0 E-1
μWIU	Repair rate for WIU	4	2.5 E-1
μVHF	Repair rate for VHF GTC (travel time plus part replacement)	4	2.5 E-1
μNet	Repair rate for microwave xmtr or rcvr failure (travel time plus part replacement)	4	2.5 E-1
μAtmos	Average microwave link outage due to atmospheric conditions	0.33	3.0
μCent	Repair rate for ROCS and NCS	5	2.0 E-1
μRexmit	Retransmission interval for critical information	0.083	12.0

MTBF => Mean Time Between Failures   λ = 1/MTBF => Failure Rate

MTTR => Mean Time To Repair   μ = 1/MTTR => Repair Rate

Failure Rates and Repair Times for ARES Components

(Baseline)

Table 5.1b

5.1.3 Microwave Outages

Occasionally the microwave system will experience a transmission outage between two towers as a result of atmospheric conditions. Characterizing the occurrence of such outages can be done in the form of a failure rate; the duration of such outages takes the form of a repair rate.

5.1.4 Data Transmission Error Rate

The rate of occurrence of undetected data transmission errors was calculated by Rockwell. Their recent experience with the implementation of a similar communications scheme indicates that the actual error rate is less than predicted.

5.1.5 Human Error Rate

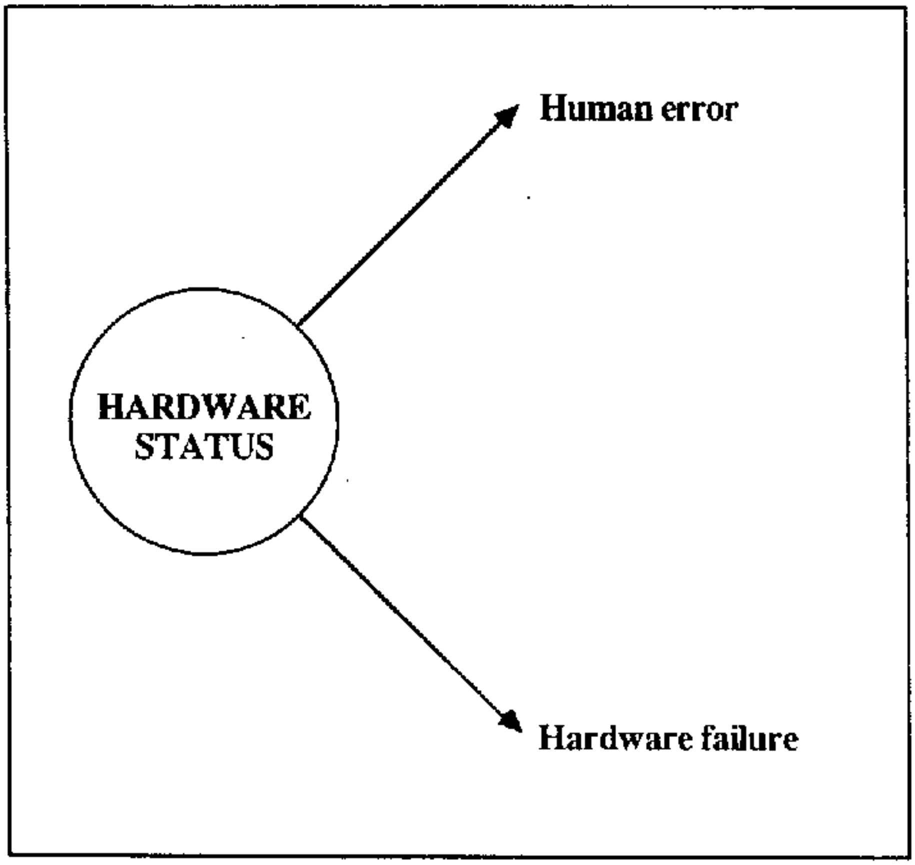
The one element that is common to both the current control systems and ARES is the human operators (engineers and dispatchers). Assuming that the average operator workload under ARES is no more than that under the current system, then the rate of operator error under ARES ought to be no more than that under the current system, and the profile of these errors ought to be similar under both systems. In particular, the rate of operator error that results in reportable accidents ought to be no worse under ARES than under the current system.

A simple, but accurate, model of the current control system is shown in Figure 5.1. The single state represented by the circle called *hardware status* is an aggregate state containing all of the failed and unfailed states of the current system hardware components. There are two virtual exit transitions to reportable accident occurrence (see sections 4.2.3.2 and 4.2.4 for details of virtual transitions and accident rate modeling). One represents accidents due to hardware failure and the other represents accidents due to human error. The model can be compressed into this simple form because:

1. The human operator represents a single-point failure mode in the system. Human error can directly cause an accident whether control system hardware is failed or not, therefore this single accident transition models the effect of human error exactly.

2. The reportable rail incident data indicates that almost all current control system related accidents are due to human error, so the *details* of the hardware failure accident transition are not significant with respect to the overall accident behavior of the system.

Because the human operator represents a single-point failure mode, the accident data can be directly translated to human error rate — specifically, the rate of human error that results in reportable accidents. The 1984 – 1985 data from the **BN Reportable Rail Equipment Incident Scoreboard** (Table 2.1) shows a total of 100 *control system related*



Current System Accident Model
Figure 5.1

accidents attributed to human error over the two year period. This is an average of 9.5 x 10^{-4} accidents per hour per region.[1] Dividing by an average of 100 train engineers and 6 to 10 dispatchers active in a region at any one time yields a human error rate of approximately 10^{-5} errors per person per hour.

Two questions must be addressed in order to gain confidence that this value for human error rate will result in a valid comparison of the ARES system accident rate to the current system accident rate.

The first is the nature and the magnitude of the operator workload under ARES. Will the operator be more or less likely to commit critical errors? In general, the operator support provided by ARES is likely to reduce the operator workload. There is no reason to expect it to increase. Also, the character of the workload will change because routine operations will be done by ARES. The dispatcher, in particular, will no longer have to keep track of information in his head. The operator will, therefore, be better able to devote his attention to important details and exceptional conditions. On balance, the critical human error rate may actually go down under ARES. Thus the human error rate from the current system is conservative bound on the expected human error rate under ARES.

The second question concerns the actual calculation of the human error rate. The calculation of individual human error rate is based on reportable accidents caused by human error and the total operator (engineer and dispatcher) hours over this time period. An average train density per region was used to determine the number of operator hours per year (with one critical operator, the engineer, per train). Clearly, changing the value of train density will change the estimate of the human error rate. As it turns out, however, most of the submodels that involve human error are insensitive, over a range of at least an order of magnitude, to the choice of average train density used to calculate the human error rate. This is because these submodels contain a compensating train density factor which appears as the number of trains per region. Those submodels without a factor involving train density do not make a significant contribution to the accident rate.

In particular, the baseline uses an average of 100 trains per region. This results in a human error rate of 10^{-5}. Assuming an average of only 10 trains per region (not an unreasonable lower bound on train density) gives a human error rate of 10^{-4}. Running the model with this lower train density and higher human error rate produces essentially the same result as the baseline prediction.

5.1.6 Coverage

In the context of this model, coverage is defined as the probability that when a failure or error occurs it will be detected before it can precipitate an accident. Coverage is a function of the architecture of the system and the fault detection, isolation and

[1] The BN system was divided into six control regions at the time of this analysis.

reconfiguration strategies employed. The model for analyzing the coverage of vital elements is discussed in section 4.2.1. The coverage values used in the ARES model are shown in Tables 5.2a and 5.2b.

5.2 ARES System Submodels

The model used to calculate ARES system safety for a region is composed of 14 submodels. Each of these submodels represents the contribution to the regional accident rate that results from certain sequences of events, i.e., hardware failures and/or human errors. The majority of these submodels are independent of each other and, due to orthogonality, exhibit the property that each submodel is regenerated from all states of every other submodel (section 4.2.3.3). The independence of some submodels is not perfect in that a particular type of failure event may appear in more than one submodel. In such cases the effect of the event may be counted more than once, therefore the assumption of independence always produces a conservative result. A regional accident rate is obtained by summing the accident rate results of the 14 submodels for the given region (see section 4.2.4). The total system accident rate is obtained by modeling each region and summing the results. This analysis assumed that the six BN regions were identical so a single region was modeled and the results were multiplied by six.

Certain hardware failures reduce the capability of ARES to obtain high quality navigation information or to implement token enforcement. When such failures occur, the affected train or trains must operate under radio blocking rules until such time as the condition is corrected. Control of the train is accomplished by voice communication between the engineer and the dispatcher (over an independent VHF radio) in a manner similar to track warrant control. If voice communication is also lost, then the train must stop until the problem can be resolved.[2] For a train under radio blocking rules, ARES will continue to calculate a zone of protection around the train, taking into consideration the uncertainties in the train's position, and will keep other trains out of this zone. However, under radio blocking rules the system is exposed to the effects of human error, specifically to the error contributions of two humans, the engineer on the affected train and the dispatcher controlling that train. It is a human error event while operating under radio blocking rules that precipitates an accident in most of the submodels. The submodels assume that a single dispatcher controls the region.

5.2.1 VHF/GTC Submodel

The wayside VHF data radios are the links between the communications network and the trains. These radios are located along the tracks at about 25 mile intervals. In most cases they will be located on the same towers used by the microwave network.

[2] In fact, the train will be stopped automatically by the TEU when the clearance token expires.

Item	Total Failure Rate	λ_1	λ_2	Detection Time ($1/\lambda_3$)	Coverage
WIU	1.25 E-4	0.35 E-4	0.55 E-4*	5 minutes	1 – (6.28 E-7)
TEU	4.0 E-4	1.0 E-4	2.0 E-4	5 minutes	1 – (4.16 E-6)
ROCS	5.0 E-4	1.0 E-4	3.0 E-4	0.1 minutes	1 – (6.66 E-8)

*The calculation adds the failure rate of a VHF radio (2.0 E-4 hr^{-1}) to the simplex part of a WIU.

Inputs to the Coverage Calculation for Dualized Vital Elements
Table 5.2a

Coverage for VHF onboard a train = 0.95

Coverage for MicrowaveTransmitter = 0.999

Other Coverage Parameters
Table 5.2b

The radio connects to the network via a ground terminal controller (GTC) — a device that acts to store and forward information from trains and wayside interface units (WIUs) to the ROCS, and vice versa. For purposes of reliability analysis, the VHF radio and the GTC are in series: if either fails, then the ROCS cannot communicate with trains on the given segment of track. Also, when a VHF fails, ROCS communication with any WIUs which are serviced by that GTC over a radio link will be lost, and when a GTC fails, ROCS communication with those WIUs serviced by the GTC over either radio *or* wire will be lost.

When either the VHF data radio or the GTC fails, any trains on the affected segment of track must switch to radio blocking rules using the voice radio until they get within range of another VHF/GTC. If the ROCS is unable to obtain switch status because of the VHF or GTC failure, then the train will have to stop and manually inspect the position of the "unmonitored" switches until it clears the affected segment of track.

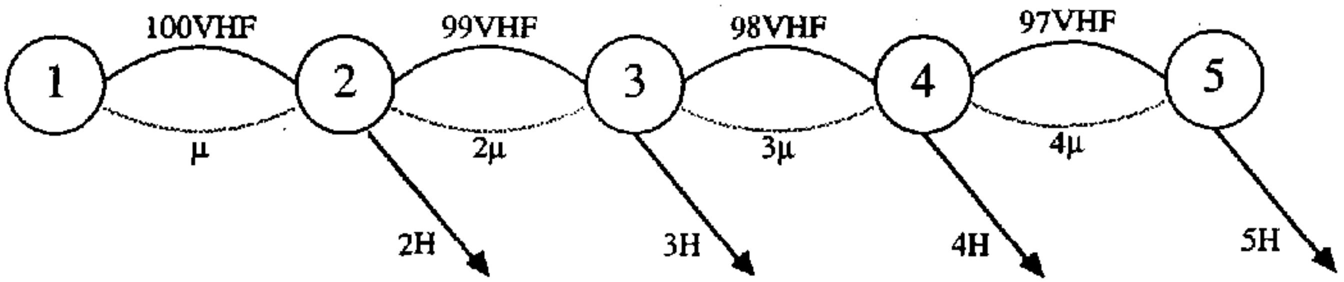
The baseline assumes 100 VHF/GTCs per region. At an average density of 100 trains per region, there will be, on average, one train within the track segment covered by a particular VHF/GTC. The effects of failures of multiple VHF/GTCs are independent of one another because the track segments affected by these failures are independent of one another. With one VHF/GTC down, an error on the part of either of two operators (the dispatcher and the engineer of the affected train) can lead to an accident, with two VHF/GTCs down then an error on the part of any of three operators (the dispatcher and the engineers of both affected trains) can lead to an accident, etc.

The VHF/GTC Markov chain submodel is shown in Figure 5.2.

5.2.2 WIU Submodel

The wayside interface unit (WIU) services the wayside sensors and actuators. It passes information from sensors back to the ROCS and delivers ROCS commands to the actuators which are generally switches. It is not critical if a WIU fails such that the ROCS cannot command an actuator. This can always be done manually. It *is* critical if the ROCS acts upon incorrect information about switch position, track integrity, etc. In other words, an accident will not result from a failed WIU as long as the ROCS knows that the WIU has failed, and that any information coming from the WIU should be ignored. Such a failure is called a covered failure. Trains will be instructed to stop and to inspect the switch position manually until such time as the WIU is repaired.

In general, the WIU detects its own failures, and will simply shut down if it has failed so badly that it cannot send its own status to the ROCS. Lack of response from a WIU is a clear indication of failure. In the event that the modem interface fails so as to corrupt a data transmission, the CRC check created by the vital part of the WIU will allow the ROCS to detect the corrupted information.



ACCIDENT RATE = P(2)*2H + P(3)*3H + P(4)*4H + P(5)*5H

VHF = VHFGTC FAILURE RATE (RDT 400 + GTC 400)
μ = VHFGTC REPAIR RATE (μVHF)
H = HUMAN ERROR RATE (Human)

VHFGTC Submodel
Figure 5.2

An uncovered WIU failure, by definition, results in incorrect knowledge sent to the ROCS of switch position or track integrity. The model assumes that this condition will always result in an accident. Coverage was calculated using the model developed in section 4.2.1 and is shown in Table 5.2a.

WIUs operate independently of each other. No sequence of covered WIU failures can itself result in an accident. A WIU can fail uncovered regardless of the state of other WIUs. The WIU is one of the two submodels in which a hardware failure can lead to an accident without the occurrence of a human error.

The WIU Markov chain submodel is shown in Figure 5.3.

5.2.3 Train Communication Submodel

Failure of the VHF data radio on board a train causes loss of communications between the train and the ROCS. This submodel behaves in a manner similar to the VHF/GTC submodel except that the train remains under radio blocking rules regardless of what track segment it occupies. The repair rate in this submodel is not derived from the actual repair time for the radio, rather it reflects the average time that the train remains on the tracks operating under radio blocking rules. It is during this time period that the system is exposed to the effects of human error.

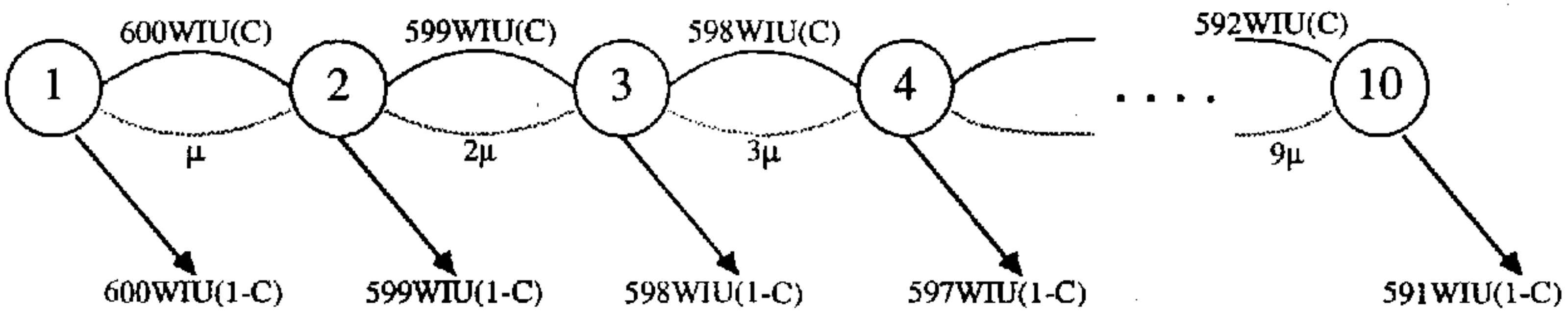
The train communication Markov chain submodel is shown in Figure 5.4.

5.2.4 GPS Submodel

The submodel for the global positioning system (GPS) receiver is similar in form to the model for train communication. A failure of the GPS causes the train to operate under radio blocking rules, and therefore to be exposed to human error.

Dropping back to radio blocking rules is the current operational specification. Alternatively the train could continue to run under ARES control by using the tachometer alone to derive train position by dead reckoning. In this case there would be no immediate exposure to human error, but a subsequent *uncovered* failure of the tachometer or the DMU could result in an accident because of incorrect train location information passed back to the ROCS. (Also at this stage, loss of the tachometer would imply loss of the TEU token enforcement function.) The system is not currently designed to provide extremely high coverage of the DMU or the tachometer in this situation, since without the GPS data there is no continuous[3] cross-check of the tachometer data (which could be corrupted by either a tachometer or a DMU failure).

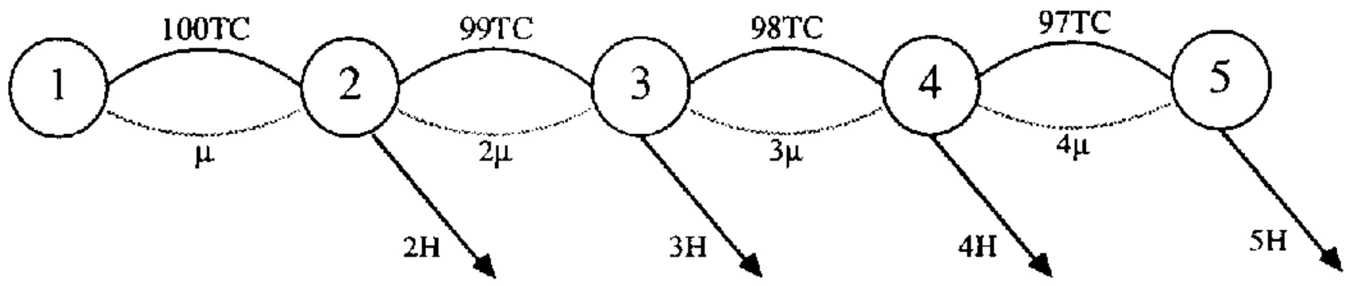
[3] While use of overswitch detector fixes does provide a periodic check of tachometer data integrity, the system is still exposed to the effects of a tachometer failure between these fixes.



ACCIDENT RATE = P(1)*600WIU(1-C) + P(2)*599WIU(1-C) + . . . + P(10)*591WIU(1-C)

WIU = WIU FAILURE RATE (WIU 400 + RDT 400)
μ = WIU REPAIR RATE (μWIU)
C = COVERAGE FOR THE WIU

Wayside Interface Unit Submodel
Figure 5.3



ACCIDENT RATE = P(2)*2H + P(3)*3H + P(4)*4H + P(5)*5H

TC = TRAIN COMMUNICATION FAILURE RATE (RDT 400)
μ = TRAIN COMMUNICATION REPAIR RATE (μTrain)
H = HUMAN ERROR RATE (Human)

Train Communication Submodel
Figure 5.4

The GPS submodel accident rate turned out to be *slightly* better if the train fell back on radio blocking rules rather than continuing under ARES operation using the tachometer for backup navigation. This was because, *in this situation*, the DMU coverage would be about 80% (as estimated by Rockwell) and the tachometer coverage was assumed to be about 50%. If DMU and tachometer coverages for this case were raised sufficiently (at some cost in hardware), the backup navigation mode would be preferable.

As in the train communication submodel, the "repair rate" for the GPS is derived from the average time that the train remains under radio blocking rules.

The GPS Markov chain submodel is shown in Figure 5.5.

5.2.5 Tachometer Submodel

This submodel is similar to the GPS submodel. In this case, however, failure of the tachometer forces reversion to radio blocking rules, since the tachometer data is no longer available for the TEU.

The tachometer Markov chain submodel is shown in Figure 5.6.

5.2.6 DMU Submodel

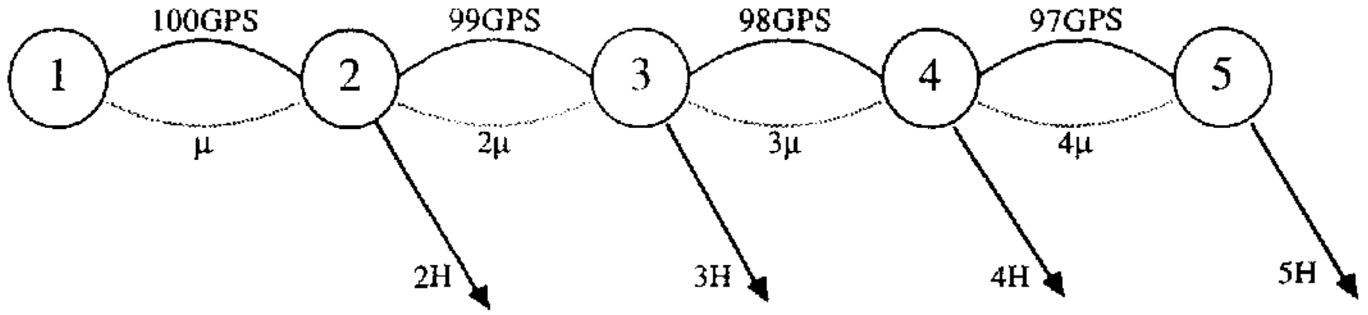
The data management unit (DMU) on board the train coordinates the flow of information among the other ARES components on the train and provides computational capability as well. Vital information flows through the DMU, but it is not created there, nor does the DMU directly act on vital information. Errors in the DMU may corrupt a clearance token from the ROCS, but the TEU will detect such errors using the CRC check on the data. GPS and tachometer data may be corrupted before being sent to the ROCS, but because this information is redundant, discrepancies will be detected by the ROCS. Under the operational rules specified in the baseline, a DMU failure has an effect similar to the failure of the on board VHF data radio, and so it is implicitly covered.

The DMU Markov chain submodel is shown in Figure 5.7.

5.2.7 TEU Submodels

The token enforcement unit (TEU) is responsible for token enforcement. A TEU failure by itself has no adverse consequences until a human error occurs. Like the other on board equipment, a TEU failure affects only the train which it is on.

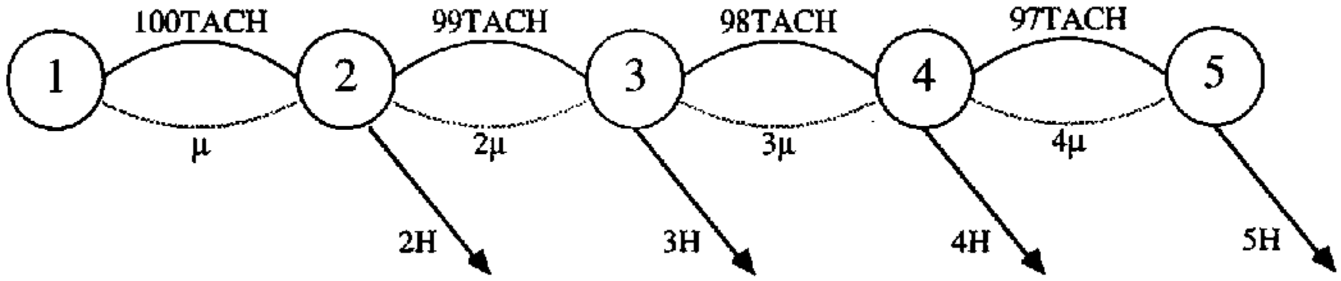
Unlike other on board equipment failures, if a TEU fails covered ARES control information is still available — only the token enforcement is lacking. The train can still operate under *de facto* ARES control until it pulls off of the active track to repair the TEU. This period of operation without a working TEU is reflected in the model as the repair rate. Human errors that can cause an accident are limited to the engineer. He is aware of the lack



ACCIDENT RATE = P(2)*2H + P(3)*3H + P(4)*4H + P(5)*5H

GPS = GLOBAL POSITIONING SYSTEM FAILURE RATE (NAVCOR + GPA 400)
μ = GLOBAL POSITIONING SYSTEM REPAIR RATE (μTrain)
H = HUMAN ERROR RATE (Human)

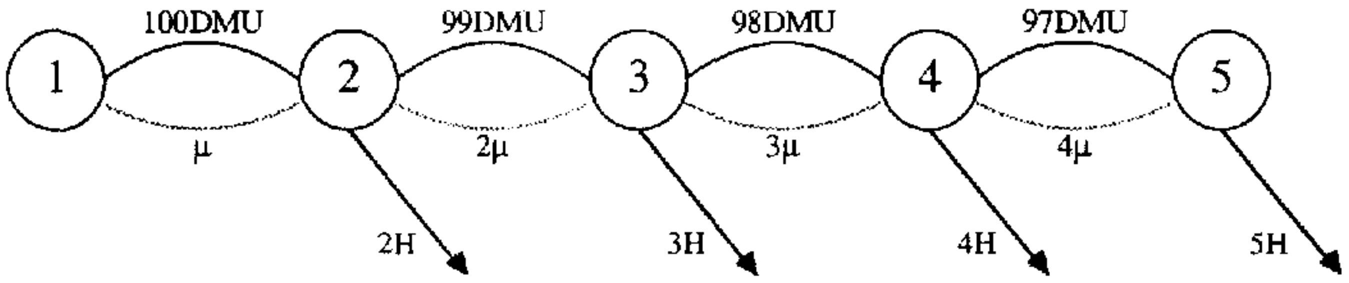
Global Positioning System Submodel
Figure 5.5



ACCIDENT RATE = P(2)*2H + P(3)*3H + P(4)*4H + P(5)*5H

TACH = TACHOMETER FAILURE RATE (Tach)
μ = TACHOMETER REPAIR RATE (μTrain)
H = HUMAN ERROR RATE (Human)

Tachometer Submodel
Figure 5.6



ACCIDENT RATE = P(2)*2H + P(3)*3H + P(4)*4H + P(5)*5H

DMU = DATA MANAGEMENT UNIT FAILURE RATE (DMU 400)
µ = DATA MANAGEMENT UNIT REPAIR RATE (µTrain)
H = HUMAN ERROR RATE (Human)

Data Management Unit Submodel
Figure 5.7

of token enforcement and must deliberately ignore the ARES information, or become incapacitated for some reason, in order to cause an accident. This submodel is shown in Figure 5.8.

If a TEU fails uncovered, operation also continues under *de facto* ARES control, but no one is aware of the lack of token enforcement. The model is similar to that for the covered failure, only the repair rate is a function of the scheduled maintenance interval. This submodel is shown in Figure 5.9.

5.2.8 ROCS and NCS Submodel

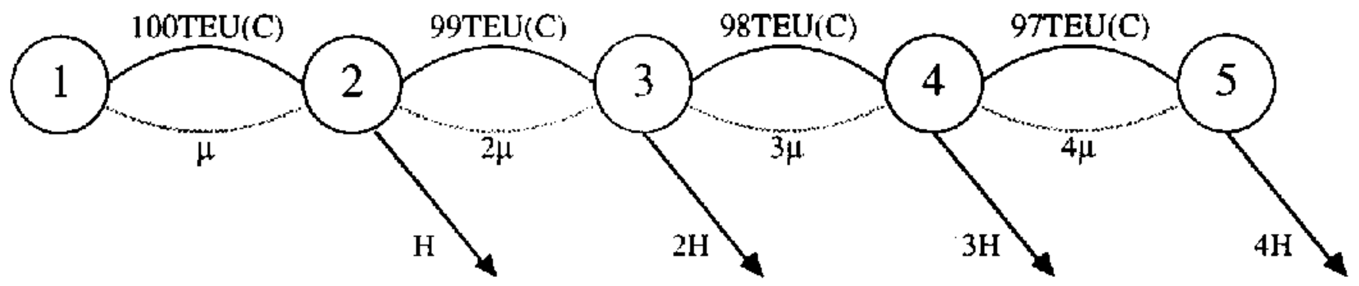
The Rail Operations Control System (ROCS) and the network control system (NCS) are the central elements of the control region. Each is dual-redundant and they are cross-strapped through a T-bar switch, so that the system can tolerate a failure of either ROCS *and* either NCS and still continue to support ARES operation.

The NCS is not a vital component in the sense that it does not generate or act upon vital information, and its failure will not corrupt information in a way that is undetectable because of the CRC checks applied to the data by vital elements. If neither NCS is working then the system reverts to radio blocking rules for all trains in the region. At this point the system is exposed to a possible human error by any of the operators (100 engineers and the dispatcher).

The ROCS is vital because it generates clearance tokens for trains, thus it is possible for an uncovered failure of the ROCS to lead directly to an accident. The model assumes that both ROCS are on-line with one actively in control and the other configured as a "hot spare" which is cognizant of the state of the region and ready to assume control at any time. Thus, when both ROCS are operating, either could fail uncovered. In particular, the model assumes that the hot spare could fail so as inject an improper token onto the network and that this behavior would go undetected. Subjectively, this seems less probable than the uncovered failure of the active ROCS, but including this effect precluded second guessing the fine details of the implementation. As it turned out, an uncovered failure of the backup ROCS made no significant contribution to the submodel accident rate.

The ROCS, like the WIU and the TEU, is modeled as having a vital part that is dual redundant and a non-vital part that is simplex. The vital part calculates clearance tokens and encodes the inner level of CRC check. When a discrepancy is detected between the dual elements within a ROCS, that ROCS is shut down and the backup ROCS takes over control if it is healthy, else the system reverts to radio blocking rules for all trains in the region. The coverage is derived from the model developed in section 4.2.1 and is shown in Table 5.2a.

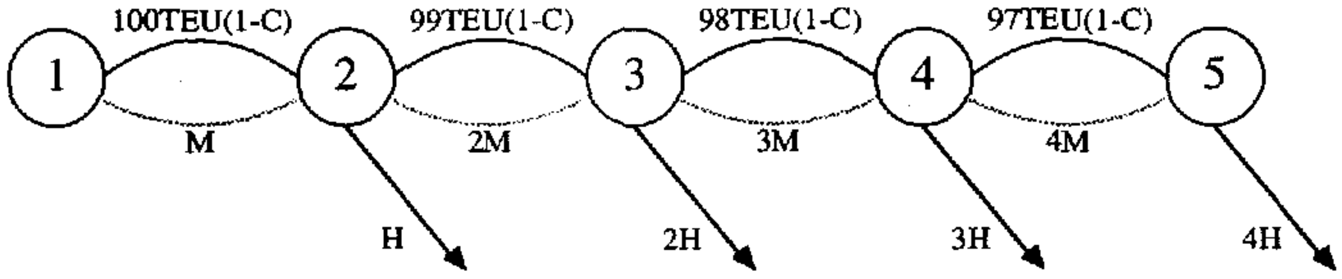
The submodel for the ROCS and NCS is shown in Figure 5.10.



ACCIDENT RATE = P(2)*H + P(3)*2H + P(4)*3H + P(5)*4H

TEU = TOKEN ENFORCEMENT UNIT FAILURE RATE (TBA 400 + TBC 400 + TBU 400)
μ = TOKEN ENFORCEMENT UNIT REPAIR RATE (μTrain)
H = HUMAN ERROR RATE (Human)
C = COVERAGE FOR TEU

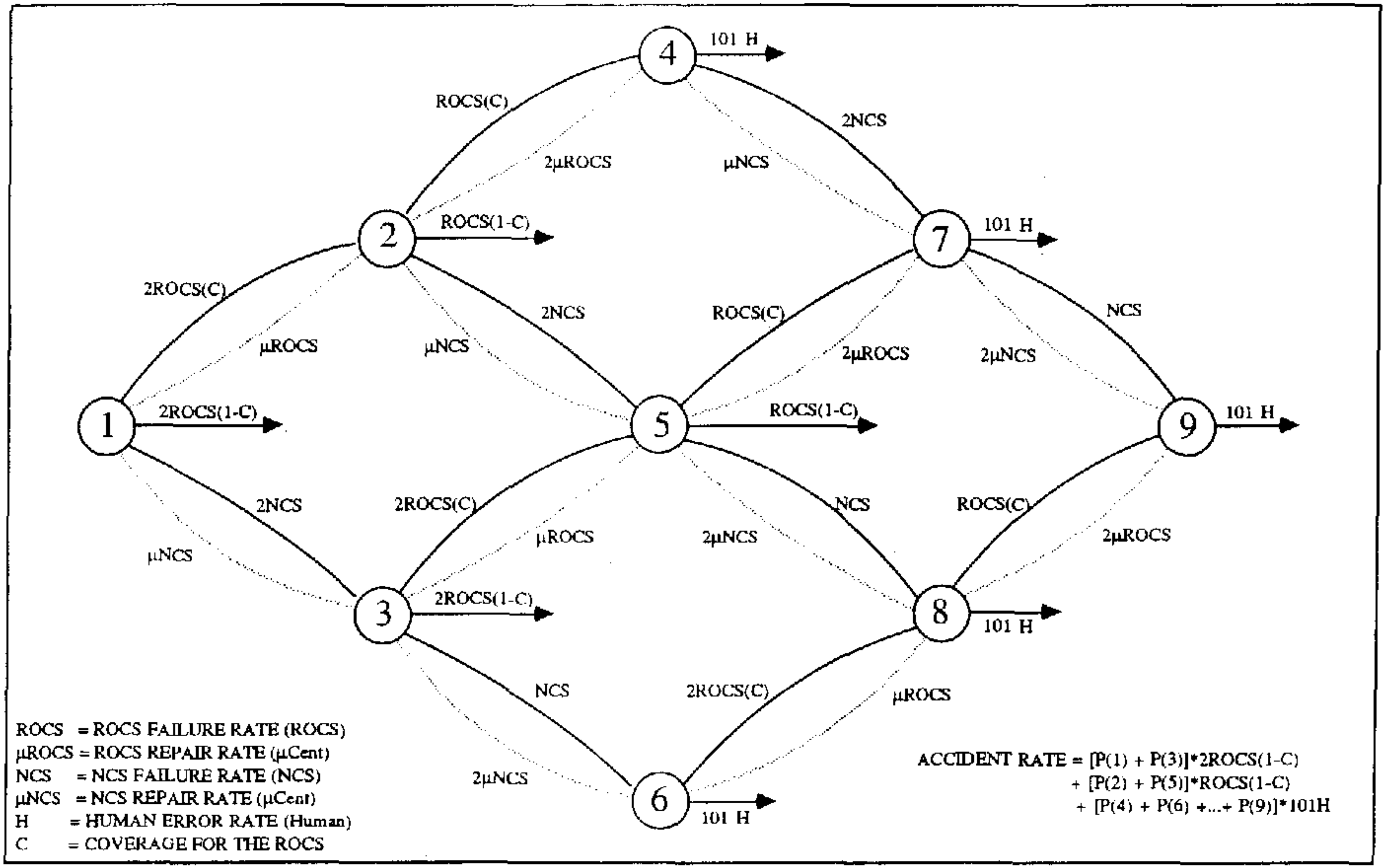
Covered Token Enforcement Unit Submodel
Figure 5.8



ACCIDENT RATE = P(2)*H + P(3)*2H + P(4)*3H + P(5)*4H

TEU = TOKEN ENFORCEMENT UNIT FAILURE RATE (TBA 400 + TBC 400 + TBU 400)
M = TOKEN ENFORCEMENT UNIT MAINTENANCE RATE (µMaint)
H = HUMAN ERROR RATE (Human)
C = COVERAGE FOR TEU

Uncovered Token Enforcement Unit Submodel
Figure 5.9



ROCS = ROCS FAILURE RATE (ROCS)
µROCS = ROCS REPAIR RATE (µCent)
NCS = NCS FAILURE RATE (NCS)
µNCS = NCS REPAIR RATE (µCent)
H = HUMAN ERROR RATE (Human)
C = COVERAGE FOR THE ROCS

ACCIDENT RATE = [P(1) + P(3)]*2ROCS(1-C)
+ [P(2) + P(5)]*ROCS(1-C)
+ [P(4) + P(6) +...+ P(9)]*101H

ROCS and NCS Submodel
Figure 5.10

5.2.9 Microwave Network Submodel

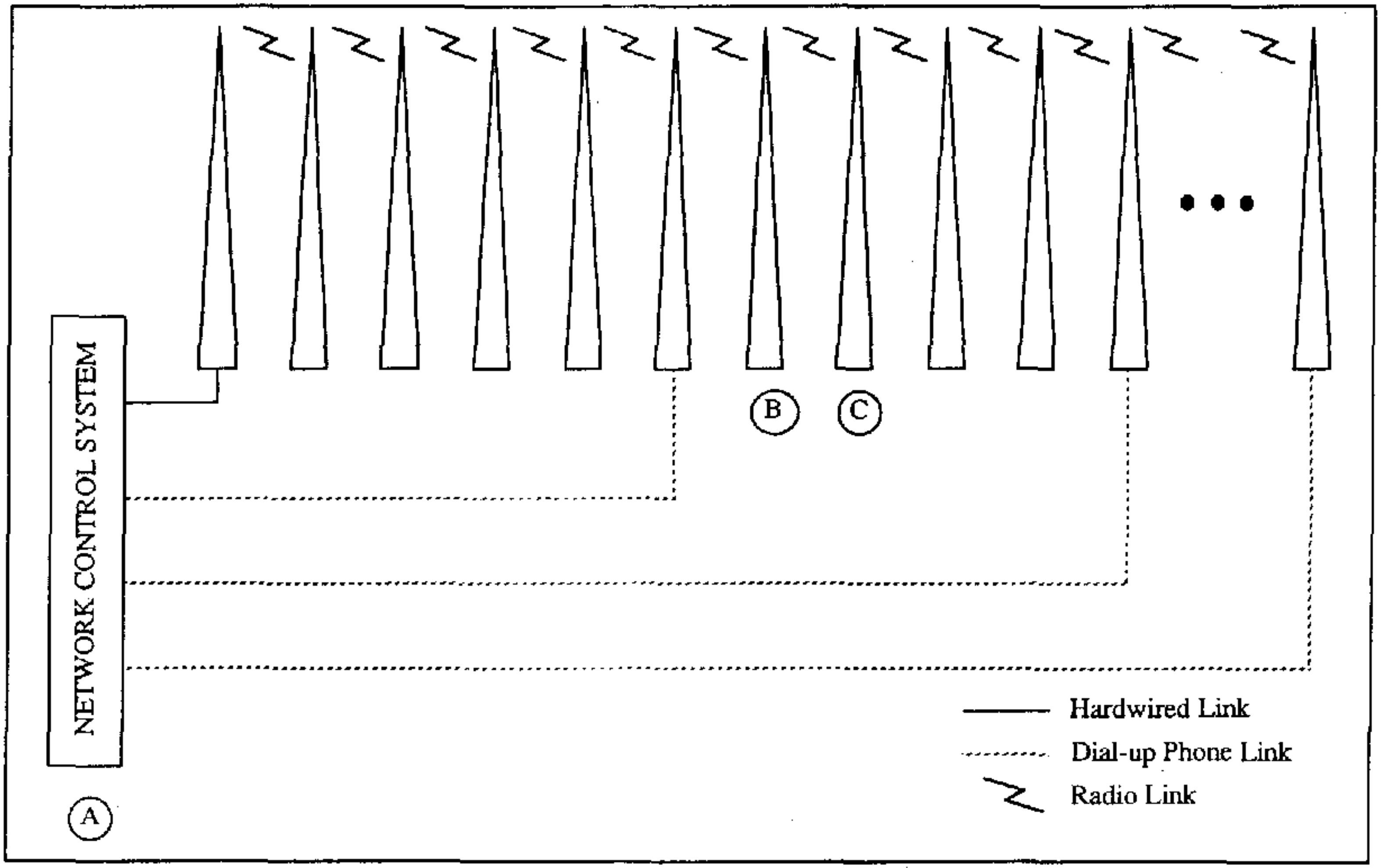
The regional microwave network is modeled as a linear array of 100 microwave relay towers. The NCS has a dedicated hardwire connection to one end of the array and has the ability to connect to every fifth tower in the array (including the last one) via a dial-up phone link. Each interior tower has redundant microwave transmitters and receivers looking in either direction, and electronic interface equipment that connects the two microwave "sides" to each other and to the GTC/VHF radio that resides on the tower. The end towers are similar, except that they look only in one direction. (Figure 5.11).

The ability to pass information between *adjacent* towers represents a radio communications link. A link comprises the equipment on opposite sides of adjacent towers, e.g., the "right" side of tower **B** and the "left" side of tower C. Under normal circumstances, a link is bi-directional. When the ROCS (point A) is talking to the GTC located on a particular tower (point **B**), information flows between **A** and **B** in both directions simultaneously. This method of operation, which is essentially full-duplex between the ROCS and any *one* point at a time, is called point multi-point. In order for data transactions to occur between points A and **B**, all intervening links must be operable in both directions.Thus, any failure that results in the inability to pass data along one direction of a link renders the entire link failed.

A set of link hardware failures and/or atmospheric outages that isolate a GTC from the ROCS causes reversion to voice radio blocking rules on the affected segments of track. This exposes train operations on these track segments to the direct affects of human error. The loss of a single link does not, by itself, result in the inability of the ROCS to reach a GTC because the network controller can "backfeed" to equipment on the far side of the break by using the dial-up phone links. In fact, since these dial-up links are placed at every fifth tower, the network could tolerate a single radio link failure between each pair of dial-up links and the ROCS could still reach all GTCs. Isolation of a GTC requires the loss of two radio links in the sub-network between a pair of dial-up links. Loss of a radio link can result from the the failure of microwave equipment on one side of the link or from atmospheric conditions between the adjacent towers.

The total accident rate attributable to communication loss in in the network is obtained by modeling a five link sub-network and multiplying the result by twenty to account for all of the sub-networks in a region. This symmetry simplifies the modeling, but in this case introduces a conservative bias, i.e., a sub-network has a slightly greater probability of experiencing two link failures when modeled alone than it would have if the entire network was modeled at once. Thus, this factorized network model yields a slightly higher accident rate than would actually be expected.

The sub-network model is done in two parts. The first models the probability of failure of one "side" of a microwave tower. The resulting value is used in the second model which looks at the combinations of "side" failures and atmospheric outages that can isolate



Network Schematic
Figure 5.11

an interior[4] GTC of the sub-network from the ROCS. For simplicity, the model assumes that the isolation of a GTC implies the isolation of all four interior GTCs. Given the baseline assumption of 100 trains per region, there will be, on average, one train talking to each GTC. This loss of communication causes the system to be exposed to five sources of human error (the dispatcher and the four engineers).

One "side" of a microwave tower consists of dual-redundant transmitters, dual-redundant receivers, a combiner for selecting which receiver information is to be used, and two antennas. The configuration is shown in Figure 5.12. A side fails if

1. Both receivers fail, or

2. The combiner fails, or

3. Both transmitters fail, or

4. A transmitter fails uncovered, i.e., in such a manner that the failure is not detected or such that the the second transmitter is rendered unusable.

The rate of mechanical antenna failure is much smaller than the rate of electronic equipment failure. The antennas are connected one-to-one with the receivers, so that the receiver failure rate easily absorbs that of the antenna. From the viewpoint of the transmitters which share the upper antenna, its failure is equivalent to an uncovered[5] transmitter failure, and its effects are absorbed in the transmitter coverage parameter.

The model shown in Figure 5.13 is used to calculate the unavailability of a side. A characteristic failure rate for a side is calculated using the model shown in Figure 5.14.

Finally, the sub-network accident model is shown in Figure 5.15. A GTC will be isolated if two links fail. This occurs when at least:

1. Two sides which are not part of the same link fail, or

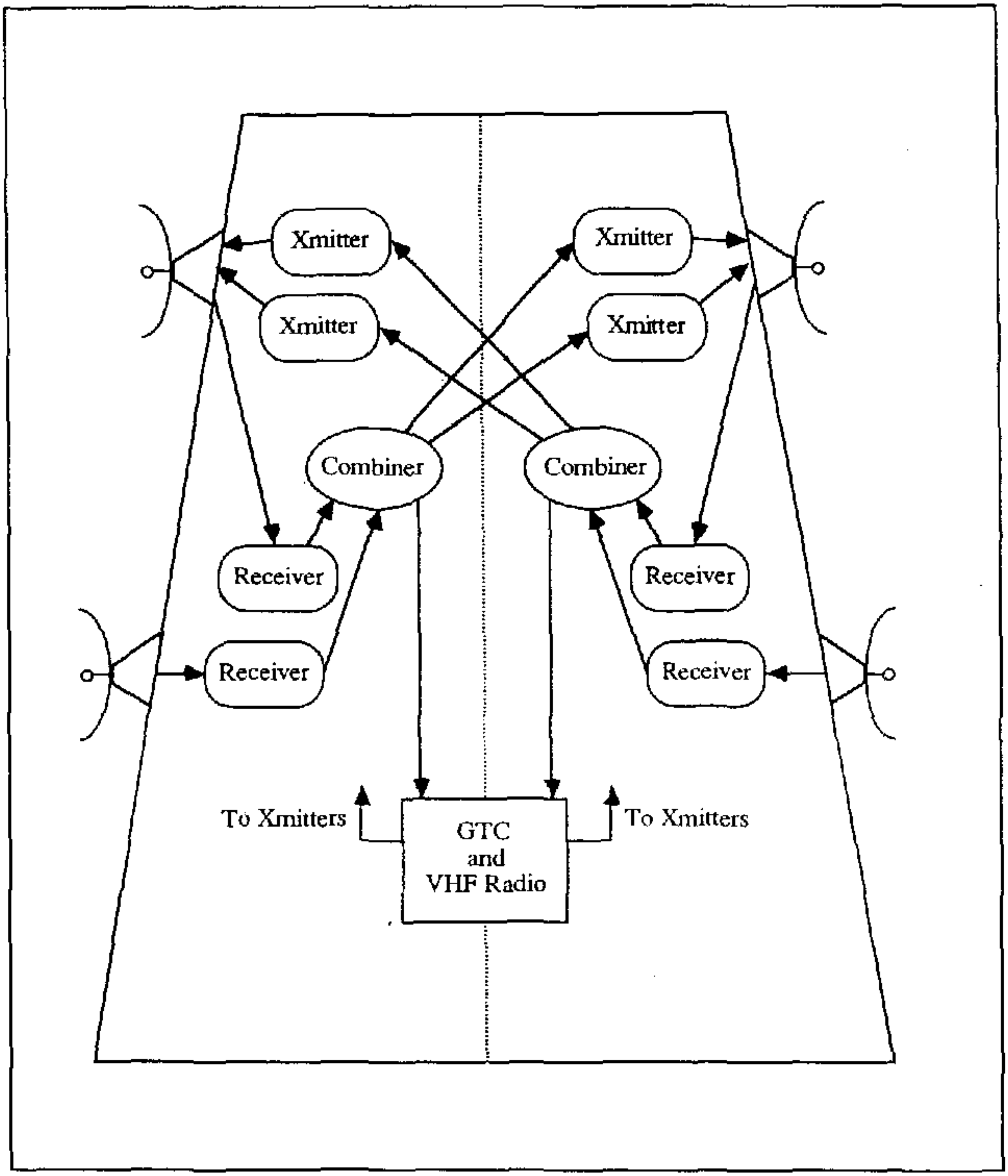
2. One side fails and a link not containing that side is down due to atmospherics, or

3. Two links are down due to atmospherics.

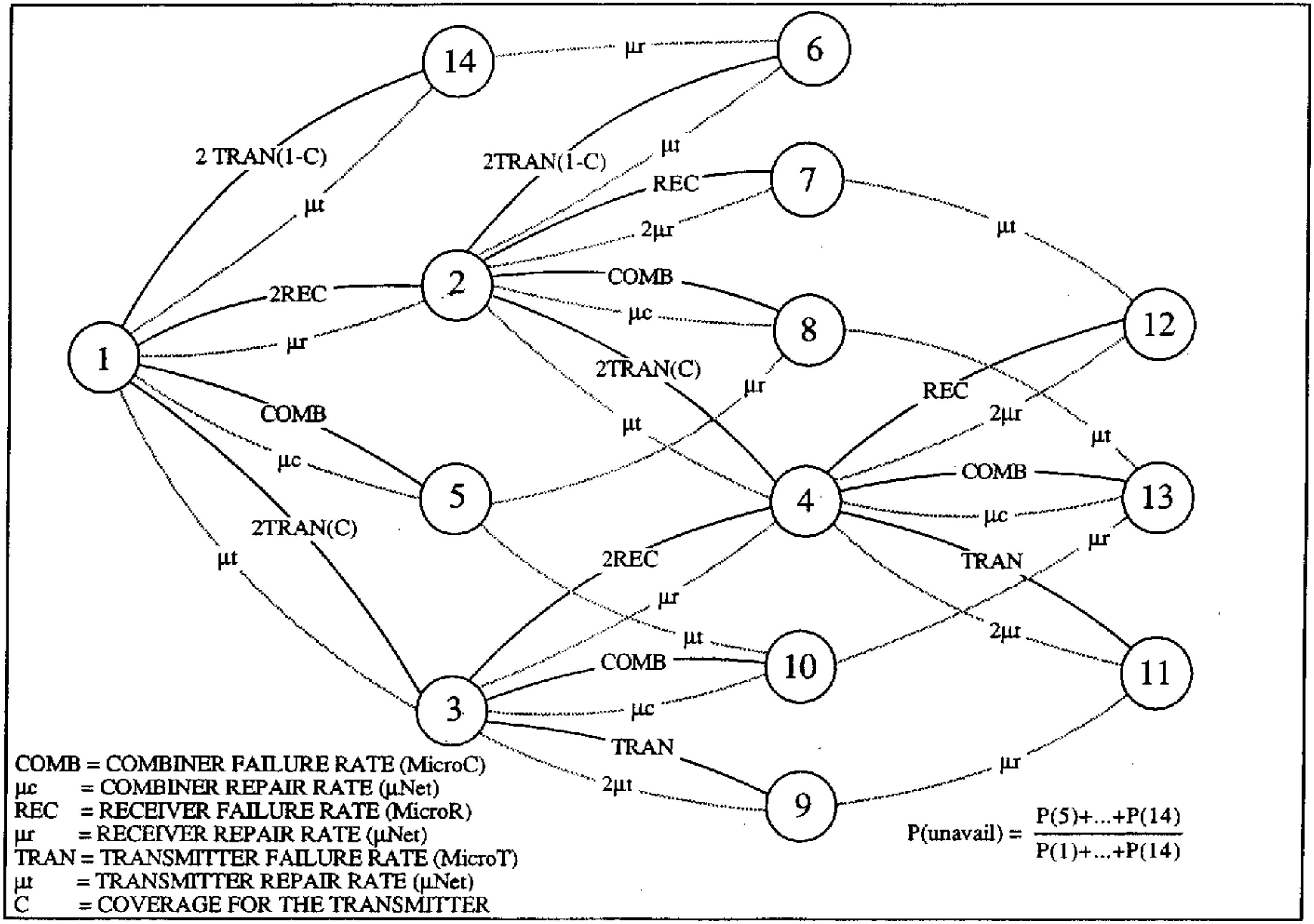
Multiplying the accident rate for the sub-network model by twenty yields the accident rate for the entire communications network.

[4] By definition, the end GTCs of a sub-network are located on towers that are connected to dial-up phone lines, so they cannot be isolated by two radio link failures, a dial-up link failure is also required. The effect of this three failure case is negligible.

[5] In this case, "uncovered" means that a transmitter fails in such a way that control is not switched to the other transmitter.



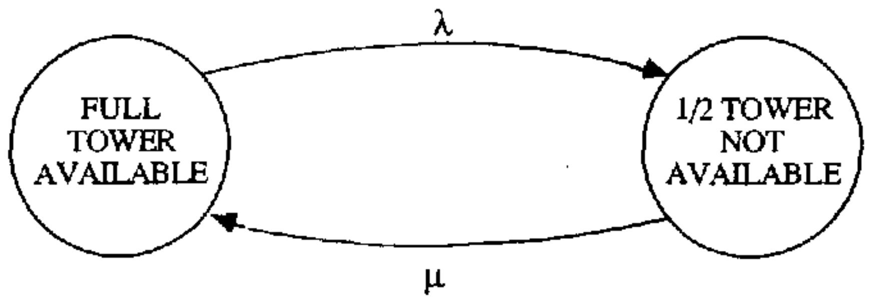
Microwave Tower Schematic
Figure 5.12



COMB = COMBINER FAILURE RATE (MicroC)
μc = COMBINER REPAIR RATE (μNet)
REC = RECEIVER FAILURE RATE (MicroR)
μr = RECEIVER REPAIR RATE (μNet)
TRAN = TRANSMITTER FAILURE RATE (MicroT)
μt = TRANSMITTER REPAIR RATE (μNet)
C = COVERAGE FOR THE TRANSMITTER

$$P(\text{unavail}) = \frac{P(5)+\ldots+P(14)}{P(1)+\ldots+P(14)}$$

Unavailability of One Side of a Microwave Tower
Figure 5.13



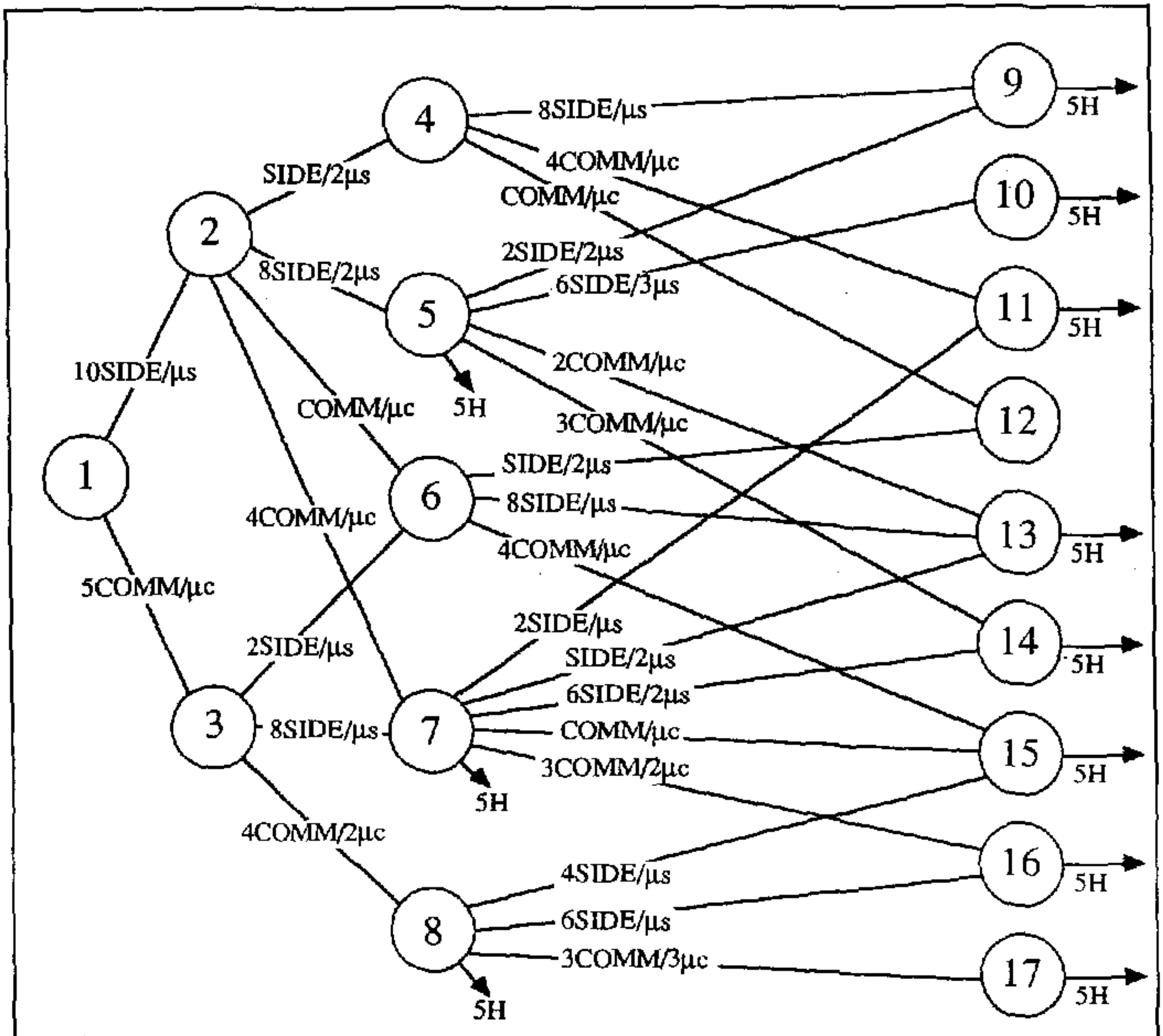
μ EQUALS THE REPAIR RATE FOR TOWER BASED EQUIPMENT

THE STEADY STATE SOLUTION FOR THIS MODEL IS

$$\text{Pr(1/2 TOWER UNAVAILABLE)} = P = \frac{\lambda}{\lambda + \mu}$$

$$\text{THE "FAILURE RATE" FOR ONE SIDE OF A TOWER} = \lambda = \frac{\mu P}{(1\text{-}P)}$$

Calculating the Failure Rate for1/2 of a Microwave Tower
Figure 5.14



ACCIDENT RATE = [P(5) + P(7) +...+ P(11) + P(13) +...+ P(17)]*5H

SIDE = FAILURE RATE FOR 1 SIDE OF A TOWER (from Figure 5.14)
μs = REPAIR RATE FOR 1 SIDE OF A TOWER (μNet)
COMM = RATE OF OCCURRENCE OF COMMUNICATION LOSS BETWEEN ADJACENT TOWERS DUE TO ATMOSPHERIC INTERFERENCE (Comm loss)
μc = RECOVERY RATE FROM ATMOSPHERIC INTERFERENCE (μAtmos)
H = HUMAN ERROR RATE (Human)

Communications Sub-Network Submodel
Figure 5.15

5.2.10 Data Transmission Error Submodels

Undetected errors in a data transmission from the ROCS to a train or to a WIU may result in an improper clearance token or switch command being accepted. This information is normally updated periodically, but there is a window of vulnerability during which action based on this information could result in an accident.

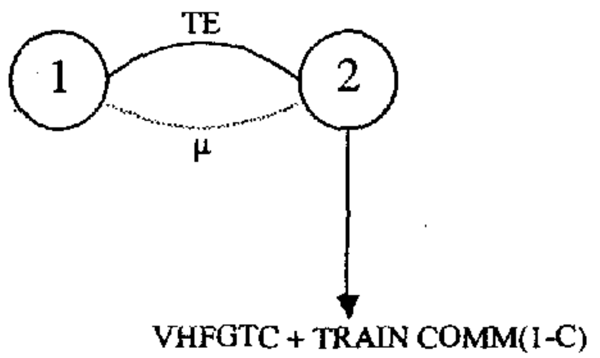
The first case involves an undetected transmission error resulting in an improper clearance token (a clearance that overlaps that of another train) being sent to a train. If communication subsequently fails, the train will run to the end of the token, causing an accident. Normally, a new token *would have been* sent before the previous token expired. This submodel is shown in Figure 5.16. This is a worst-case model, since a detection of data communication loss would cause a reversion to voice radio blocking rules and thus a verbal update of the clearance.

The second case involves an undetected transmission error resulting in an improper switch position command. If the switch position is not correctly updated before a train arrives, then an accident will occur. Although switch position commands are not normally repeated, the switch position is monitored periodically, so that the condition will be detected and the proper command retransmitted at that time. If communication fails just after the erroneous command is sent, and the trains clearance limit is beyond the switch, then the train may cross the improperly set switch and cause an accident. But as in the previous case, detection of data communications loss will cause a reversion to voice radio blocking rules, which in this case would require the crew to stop and manually inspect the switch position before proceeding. This submodel is shown in Figure 5.17.

5.2.11 Track Force Submodel

If the ROCS is unaware of the existence of a work force on the tracks then it could clear a train through the area which would result in an accident. The track forces carry ARES communication and navigation equipment, but the ROCS cannot know where the work crew is if they do not turn on the equipment. In fact, the ROCS should know when and where a crew is *scheduled* to work, and expects to make data contact with them. So, two human errors are required for an unaccounted track force to exist: the work force must fail to be scheduled, and the crew must fail to turn on the ARES equipment and log in when they start work.

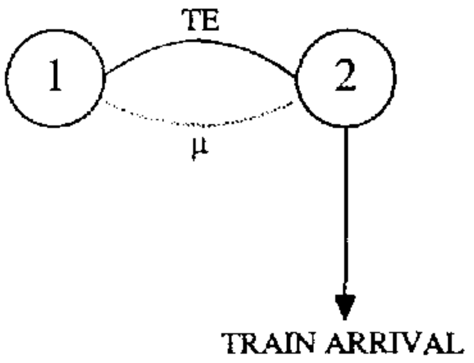
Assume an average track force makeup of 8 inspectors at 3 *logins* per day and 6 track crews at 2 logins per day. The *login* should occur when the crew or inspector first gets on the tracks. Assume a five minute window during which they either log in or forget to do so. This equals three person-hours of exposure in a 24 hour period, or an average error exposure of 0.125 humans ($0.125\text{x}10^{-5}$ errors per hour). The dispatcher should also be aware of the scheduling of the track forces, so the absence of a *login* should trigger some action on his part. Assume, conservatively, a constant exposure to dispatcher error, i.e.,



ACCIDENT RATE = P(2) * (VHFGTC + TRAIN COMM(1-C))

TE = RATE OF TRANSMISSION ERRORS TO TRAIN (Xmit error)
μ = RETRANSMISSION RATE (μRexmit)
VHFGTC = VHFGTC FAILURE RATE (RDT 400 + GTC 400)
TRAIN COMM = ON-BOARD VHF FAILURE RATE (RDT 400)
C = COVERAGE OF ON-BOARD VHF RADIO

Transmission Error Submodel: ROCS To Train
Figure 5.16



ACCIDENT RATE = P(2) * (TRAIN ARRIVAL)

TE = RATE OF TRANSMISSION ERRORS TO WIU (Xmit error)
μ = RET RANSMISSION RATE (μRexmit)
TRAIN ARRIVAL = RATE OF TRAINS ARRIVING AT THE SWITCH (Arrive)

Transmission Error Submodel: ROCS To WIU
Figure 5.17

one human (10^{-5} errors per hour). This ignores the possibility that ARES could be programmed to remind the dispatcher about the work force schedule.

Both the work crew *and* the dispatcher must commit an error if a potential accident situation is to exist, so the accident rate is the product of these two error rates, that is,

$$(0.125\text{x}10^{-5})(10^{-5}) = 1.25\text{x}10^{-11} \text{ accidents per hour.}$$

5.2.12 Consist Error Submodel

Assume, conservatively, that any error in the consist information will cause the engineer to mishandle the train. Consist information is entered into the management information system when the train makeup is scheduled. It is checked again manually when the train is configured in the yard. So two independent human errors are required to present bad consist information to the engineer.

Assume 100 train originations per day with 100 cars per train. With two minutes per car required to type car data into the MIS system the human error exposure for data entry is 13.9 humans ($13.9\text{x}10^{-5}$ errors per hour). With one half minute to verify each car in the consist the human error exposure in the yard is 3.48 humans ($3.48\text{x}10^{-5}$ errors per hour). The accident rate is the product of these two error rates, that is,

$$(13.9\text{x}10^{-5})(3.48\text{x}10^{-5}) = 4.84\text{x}10^{-9} \text{ accidents per hour.}$$

5.3 Detailed Results

Each of the submodels was run using the baseline input data from Table 5.1 and the coverage values from Tables 5.2a and 5.2b. The results are enumerated in Table 5.3 and shown graphically in Figure 5.18. For each submodel, this table gives the the accident rate contributed to a region by that submodel in terms of accidents per hour and accidents per year. Also, the magnitude of the error introduced by model truncation is shown for those submodels where it is appropriate. Summing the results of the submodels gives the accident rate for the region. Multiplying by the total number of regions (in this case, six) yields the predicted total system accident rate.

The baseline model predicts that the BN system under ARES control will experience about 0.5 reportable accidents per year as a result of control system failures. This compares very favorable with the current statistics of about 50 accidents per year. Sensitivity analyses (section 5.4) indicate that this accident ratio is relatively insensitive to modeling assumptions and to uncertainties in the values of baseline inputs.

5.4 Sensitivity Analysis

The purpose of sensitivity analysis is two-fold: to determine the components that are the major contributors to the solution and to determine the impact on the solution of

SUBMODEL	ACCIDENTS PER HOUR	ERROR BOUND	ACCIDENTS PER YEAR
ROCS AND NCS	1.25E-8	-	2.18E-4
TRACK FORCE	1.25E-11	-	1.09E-7
DATA ENTRY	1.93E-8	-	1.69E-4
VHFGTC	2.52E-6	1.47E-11	2.21E-2
TRAIN COMM.	1.95E-6	4.08E-12	1.70E-2
TACHOMETER	3.31E-7	5.68E-16	2.89E-3
WIU	1.22E-7	1.18E-15	1.07E-3
TBI (COVERED)	1.99E-6	9.86E-11	1.74E-2
TBI (UNCOVERED)	6.88E-9	5.82E-23	6.02E-5
GPS	1.53E-6	1.24E-12	1.34E-2
DMU	8.98E-7	8.37E-14	7.86E-3
XMIT ERR TO WIU	8.33E-11	-	7.29E-7
XMIT ERR TO TRAIN	2.79E-14	-	2.44E-10
NETWORK COMM	1.63E-10	2.68E-17	1.43E-6
TOTAL PER REGION	9.38E-6	1.19E-10	0.082

TOTAL ACCIDENTS PER YEAR FOR ALL SIX REGIONS ≈ 0.5

Submodel Results
Table 5.3

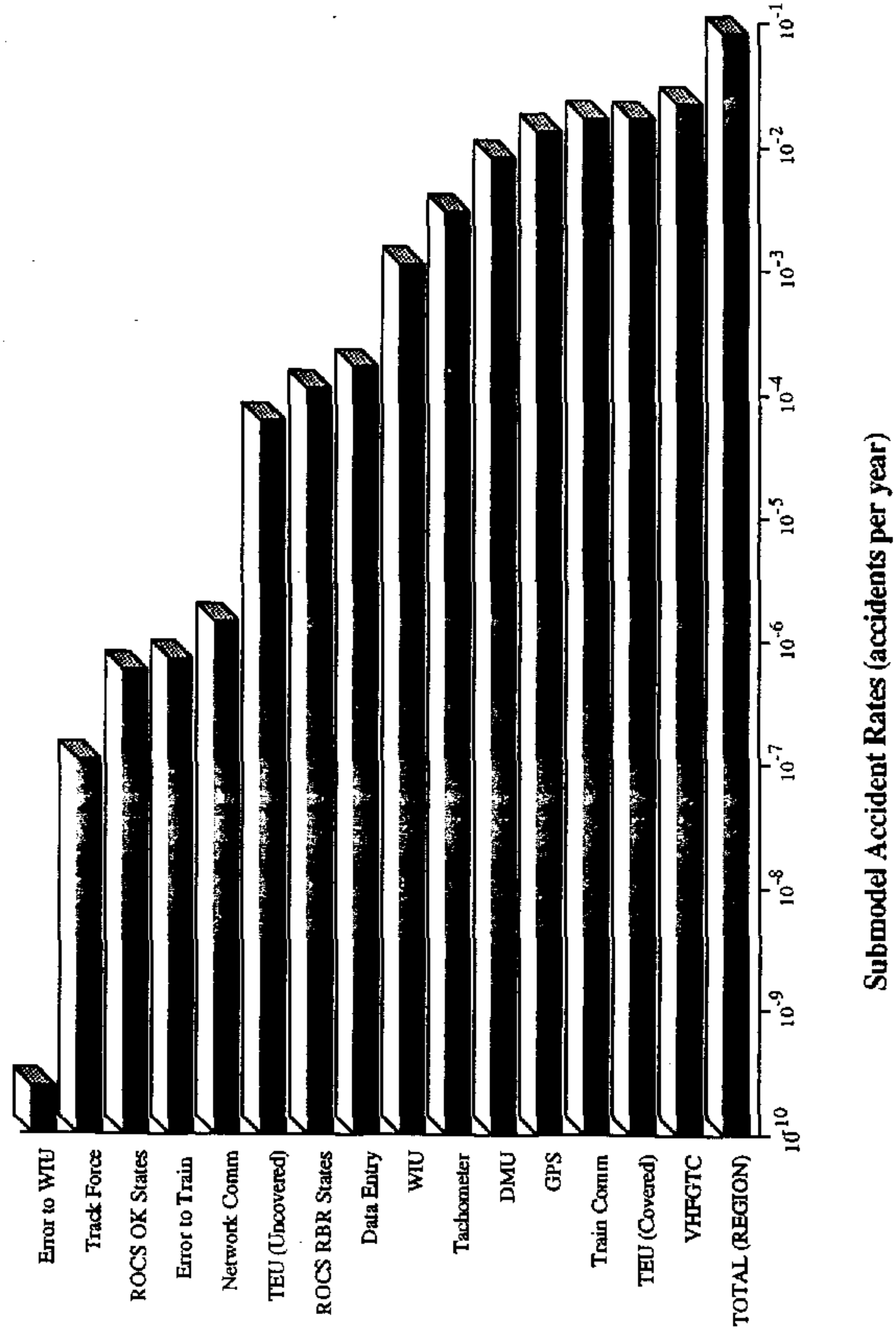

Error to WTU
Track Force
ROCS OK States
Error to Train
Network Comm
TEU (Uncovered)
ROCS RBR States
Data Entry
WIU
Tachometer
DMU
GPS
Train Comm
TEU (Covered)
VHF/GTC
TOTAL (REGION)
10^{-10}
10^{-9}
10^{-8}
10^{-7}
10^{-6}
10^{-5}
10^{-4}
10^{-3}
10^{-2}
10^{-1}
Submodel Accident Rates (accidents per year)

Figure 5.18

changes in the various parameters. The former has been handled through the model decomposition into component-type submodels. The latter is often more useful in determining the parameters that must be pinned down to obtain an accurate result. We will address both of these aspects in this section.

Consider the accident rate contributions of each of the submodels. The primary contributor is the VHF GTC chain. It contributes *one quarter* of the total accident rate. The majority of the remainder of the accident rate is due to the train communication, TEU (covered), and GPS chains. Together these four chains generate nearly *90%* of the accident rate. If one is concerned with reducing the predicted accident rate under ARES operation, these are the components that should be addressed. Even if the remaining components could be made to have no contribution to the accident rate, it would only be reduced by 10%. This analysis directs the efforts of system designers to the portions of the system that are the primary drivers of the accident rate.

Another type of sensitivity analysis is concerned with the impact on the solution of changes in the system parameters. For example the change in the probability of state i in a chain as the component failure rate λ changes is:

$$dP(i) / d\lambda \;=\; P(i)\,[(i / \lambda) \;-\; N/(\mu + \lambda)]$$

where equation (13) of section 4.2.3.2 defines P(i). The parameter μ is the component repair rate.

Since λ is usually a rather small number, it is more convenient to write the derivative in units of the baseline λ. Defining $dS = d\lambda / \lambda$, the derivative becomes:

$$dP(i) / dS \;=\; P(i)\,[i \;-\; N\lambda/(\mu + \lambda)]$$

Using the fact that in ARES λ is always much smaller than μ, the following approximate derivative is formed:

$$dP(i) / dS \;\approx\; P(i)\,[i \;-\; (N\lambda / \mu)]$$

Doubling the size of λ is equivalent to changing S by 1. Therefore, the *relative* change in P(i) when λ is doubled is:

$$dP(i) / P(i) \;\approx\; [i \;-\; (N\lambda / \mu)] \qquad (18)$$

The term $(N\lambda / \mu)$ determines the rate at which the state probabilities in the chain decrease (or initially increase) as i increases (see Section 4.2.3.2). In all cases in ARES the term $(N\lambda / \mu)$ is less than 1. This indicates that the most probable state in the chain is state 0, followed by state 1, etc. Therefore, for these chains the relative change in state probability is approximately i. Although this relative change increases for states at higher failure levels, the probabilities of these states are falling off at a faster-than-linear rate. The result is that the largest magnitude of change is in states 0 and 1. Neither of these states have a relative change greater than 1. In fact, state 0 decreases in probability while state 1 increases. Even when chains with virtual transitions that depend on λ are examined, the relative change in the total chain accident rate is less than the relative change in λ. In other words, doubling the failure rate λ only causes state probabilities to change by less than factors of two. Thus these chains are insensitive to λ.

The above derivation is repeated for changes in the state probability due to changes in the repair rate μ. Applying the approximation that λ is always much smaller than μ leads to:

$$dP(i) / dR \approx P(i) [(N\lambda / \mu) - i]$$

where $dR = d\mu / \mu$ scales the derivative to relative changes in the repair rate. Writing the equation in terms of the *relative* change to the state probabilities gives:

$$dP(i) / P(i) \approx [(N\lambda / \mu) - i] \qquad (19)$$

Comparing Equation (18) to (19) shows that the arguments that lead to a conclusion of insensitivity to changes in λ also apply to changes in μ.

Having shown the chains to be insensitive to changes in failure rate and repair rate, the sensitivity of the non-chain submodels is examined. The contributions of most of the non-chain submodels are in the second or greater significant digit of the total accident rate. No reasonable change in the failure and repair rates associated with these models, such as a change of an order of magnitude, will have an impact on the total accident rate.

Thus, the accident rate prediction of the ARES model is not particularly sensitive to changes in any one failure or repair rate. However, a constructive combination of failure and repair rate changes in a few of the chains that are the major contributors to the accident rate could cause noticeable changes. Still, the relative change in the accident rate would not be greater than the total relative change in the failure and repair rates.

Sensitivity to human error rate is discussed in the human error rate calculation (section 5.1.5).

6.0 SUMMARY

The BN system under ARES should experience about two orders of magnitude fewer control system related accidents than under the current set of control systems.

Virtually all accidents caused by failures of the current control systems are attributable to the human element. The human is an integral part of the current train control systems. He is a critical, non-redundant element responsible for closing the sensor (signals) to actuator (throttle and brakes) control loop. The system is exposed at all locations and at all times to single human error events that can directly cause accidents.

The human is also an element of the ARES control system, but the ARES architecture is such that the human does not represent a single-point failure mode. There are no single human error events that can directly precipitate an accident because ARES clearance enforcement hardware operates in parallel with the train operator. The human can contribute to an accident only as part of a multi-event sequence of independent hardware failures and human error events.

In the same manner that ARES reduces the exposure to human errors, it provides redundant checks on its own hardware elements. Hardware component failures are detectable by ARES itself. The knowledge of which hardware components are inoperative allows the system to prevent these failed components from precipitating accidents, i. e., the system hardware is designed to be fail safe. Specific multi-event sequences of hardware failures are required to cause an accident due to hardware failure alone. These sequences turn out to be very improbable.

The fidelity of the ARES model is such that model truncation produces only a negligible error in the accident rate prediction. In addition, the model is designed to be conservative, i.e., errors in the results due to lack of fidelity in modeling the ARES system behavior will always make the ARES system safety appear worse than it actually is. By way of example, consider that the model does not attempt to address those cases where the human *could* in some cases detect and compensate for certain "undetected" ARES hardware errors which are part of accident causing event sequences not involving human error.

The sensitivity analyses indicate that the output of the model is relatively insensitive to the expected range of variations in the model's inputs. Even assuming worst-case values for all the inputs, the prediction for the ARES system accident rate is lower than that demonstrated by the current control systems.

REFERENCES

1 Babcock, Philip S. An Introduction to Reliability Modeling of Fault-Tolerant Systems. CDSL Report R-1899. September 25, 1986.