Dynamic Risk Estimation: Development of the Safety State Model and Experimental Application to High-Speed Rail Operation

by

Edward John Lanzilotta

S.B.E.E., Massachusetts Institute of Technology (1982)
S.M.M.E., Massachusetts Institute of Technology (1992)

Submitted to the Department of Mechanical Engineering
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Mechanical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

January 1996

© Massachusetts Institute of Technology 1996. All rights reserved.

Author .....
Department of Mechanical Engineering
December 20, 1995

Certified by .....
Thomas B. Sheridan
Ford Professor of Engineering and Applied Psychology
Thesis Supervisor

Accepted by .....
Ain A. Sonin
Chairman, Departmental Committee on Graduate Studies

MASSACHUSETTS INSTITUTE
OF TECHNOLOGY

ARCHIVES

MAR 1 9 1996

LIBRARIES

Dynamic Risk Estimation: Development of the Safety State Model and Experimental Application to High-Speed Rail Operation

by

Edward John Lanzilotta

Submitted to the Department of Mechanical Engineering
on December 20, 1995, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Mechanical Engineering

Abstract

Current trends in transportation include an infusion of technologies aimed at improving service quality and efficiency. Broad examples include high-speed rail and Intelligent Transportation Systems (ITS). Many of the technologies utilize advanced sensing and control techniques to improve the overall system performance. However, these technologies are changing the nature of the human-machine interface in these transportation systems. Of major concern are the implications on safety—technological advances will be accepted only if the level of safety is not compromised. One component of safety is the risk probability of an undesirable event. It is intuitively clear that risk is dynamic in nature, and the ability to estimate dynamic risk probabilities would be advantageous in both systems analysis and operator performance evaluation.

A general model, called the safety state model, has been developed for estimating the dynamic risk probability. Based on discrete finite Markov processes, the safety state model defines a set of system states from the exhaustive combination of a set of binary conditions which might be contributory to an undesirable event. Observations of an existing system are used to determine the state transition probabilities. The mean time to failure, as a function of system state, is calculated from the state transition matrix. The resultant function is transformed into a risk probability function, as a function of system state.

The safety state model has been demonstrated through application to high-speed rail operation. A human behavior experiment was conducted to explore the effects of control automation on operator performance in high-speed rail. It was determined that various levels of control automation contributed to refocusing of operator visual attention, with implications to failure response. In conjunction, data obtained during the experiment was used to demonstrate application of the safety state model.

Thesis Supervisor: Thomas B. Sheridan
Title: Ford Professor of Engineering and Applied Psychology

Acknowledgments

I believe that each person's life is a strand of a huge web by which we are all interconnected. Any project of this magnitude has a host of contributors, many of whom aren't fully aware of their impact. I would like to thank all those people that helped to turn this dream into reality.

I particular, I would like to acknowledge the guidance of my thesis committee. I feel very fortunate that I could bring together such a distinguished and knowledgeable group of faculty to review and guide my work. My advisor, Professor Thomas Sheridan, provided the support and encouragement to pursue this research, and he provided an opportunity to work on a research project which was extremely well-suited to both my thesis topic and long-term career goals. Professor Joseph Sussman, of the Department of Civil and Environmental Engineering, lent his extensive knowledge of transportation systems. His expertise helped me address some of the strengths and weaknesses of my research. Professor John Hansmann, of the Department of Aeronautical and Astronautical Engineering, was particularly enthused with the research and provided valuable feedback with respect to human interface issues and potential applications. Professor Steven Dubowsky provided useful insight throughout the research, and he provided expertise in the areas of dynamics and controls. Finally, although he was not officially a member of the committee, Professor John Deyst, of the Department of Aeronautical and Astronautical Engineering, freely provided his time and expertise to review and comment on the details of Markov process theory.

One of the more rewarding aspects of life in graduate school is the opportunity for building professional and personal relationships with peers. My association with the fellow members of the Human-Machine Systems Laboratory at MIT has been enlightening as well as fun. In particular, I would like to thank Nick Patrick—his knowledge, combined with an uncanny knack for crystal clear explanation, helped turn a sea of confusion about statistics into a wealth of understanding. His continuous wit also helped forge a strong friendship. I would also like to acknowledge Shumei Yin Askey, my partner on the high-speed rail project. Our interaction gave new

meaning to my understanding of teamwork. I would also like to acknowledge the other members of the group, for their support and knowledge: Dave Schloerb, Shih-Ken Chen, Jianjuen Hu, Mark Ottensmeyer, Thomas Chao, Suyeong Kim, Mike Kilaras, and Jie Ren. The next generation of students working on the high-speed rail project, Jay Einhorn, Bernardo Aumond, Steve Villareal, and Helias Marinakos, have been a great help by checking my work and keeping me honest.

Thanks also go to other friends and colleagues from MIT. My study partners from qualifying exams, Tim Quinn and Kurt Roth, have been generous with well-placed critical commentary as well as with kindly support, both of which only come from true friends. I would also like to thank my colleagues from the Intelligent Machines Laboratory, especially Chris Jones, Kevin Brown, Mark West, Cliff Federspiel, Brenan McCarragher, Sean Li, and Xiandong Hong—our associations helped me find the best course of academic path. My association with the support staff, especially Kari Kulaszewicz and Veronica Culbert, have been fruitful and rewarding.

The bulk of my doctoral research was completed while in residence at the Volpe National Transportation Systems Center. The opportunity to work in this facility led to many productive relationships. I would like to thank Mr. Bob Dorer for sharing his extensive knowledge about rail systems. Drs. Peter Mengert and Robert Disario were indispensable with regard to statistical methods, experimental design, and review of the Markov process methods. Dr. Judith Bürki-Cohen provided overall guidance to the project, as well as effective editing for the documentation which accompanied the project. Finally, I would like to thank Dr. Donald Sussman for his support and guidance of the project. His influence was critical to the successful integration of the simulation system and the experiment.

Apart from all other influences, family can provide the bedrock of support that cannot be obtained from any other source. From my sister Dolores, I have learned to stand tall in the face of adversity. I thank my sister Mary Anne for providing the germ of the idea that became the topic of this research. My brother James has taught me much about dealing with people, and has helped keep the flame of technical passion alive. My brother Paul is the only one that truly understands the concept of the

firehose, and I thank him for sane perspective and companionship at critical points of juncture. Finally, I acknowledge my mother, Pat, for teaching me how to convey ideas to people in a way that they can really understand.

Without a doubt, this work would not have been possible without the generous love and support of my beautiful wife, Marita. She has provided me so much in the past few years that I might need the rest of my life to fully make it up to her.

****

This work is dedicated to my father. He taught me how to learn, how to love, and how to live. I am forever indebted to him.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

As our society grows more complex, with a greater reliance on technology, we cast a wary eye toward issues of safety. There is collective hope that technology growth will lead to longer and safer lives for all. Yet experience with some forms of technology might instill fear that systems have become too complex, that we might not appreciate all of the risks and costs, especially with regard to safety.

This is especially true when the system in question utilizes one or more human operators. Although people carry important skill characteristics which cannot be duplicated by machines, they are also subject to boredom, fatigue, complacency, and inconsistency. These are precisely the characteristics which can lead to the compromise of safety in a complex system.

Nowhere is this question raised more than within the context of transportation. Transportation systems of all modes—air, rail, highway, and marine—sustain our lives on a daily basis. Transportation domains, traditionally dominated by human control, are increasingly seen as ripe for application of advanced technology. A few examples of related technology include flight automation systems, the Automated Highway System (AHS), and magnetically levitated (maglev) trains.

When considering the application of advanced technologies to transportation systems, there is justifiable concern about safety implications. There are several areas of concern. We must be able to identify scenarios which pose a threat to safety. We must also be concerned with the division of control between human and automated

elements, so as to assure the highest level of safety. Finally, we must be able to evaluate the safety-related performance of human operators in a functioning system.

Typically, human operators face a complex task, involving decisions which trade off high-level performance objectives against the costs of achieving those objectives. In transportation, the objectives include travel time and passenger ride comfort, while the costs include energy usage, vehicle wear, and risk.

Of these costs, one of the more important and interesting is risk. Most people have an intuitive notion of safety, based on individual perception of risk and causality that has been developed through experience. Because of the intensely personal nature of experience and risk perception, a situation evaluated as "safe" by one person could be considered "highly risky" by another. To date, the methods used for evaluating safety performance in transportation are of a long-term averaging nature—count the number of accidents or injuries over a period of time, and divide by some index (miles, time, etc.) to obtain a relative risk level. As a measure of dynamic risk, these methods are wholly unsatisfactory, as they do not provide a means for discriminating between levels of risk in the absence of accidents. In addition, they do not make use of events that are of higher frequency but lesser consequences than significant accidents.

The overall focus of this research is the development and application of a method for estimating relative safety. A probabilistic model of system behavior is developed, which provides a mechanism for expressing dynamic risk probability as a function of system state. The model, termed the *safety state model*, is then applied to safety analysis of a high-speed rail system.

Chapter 2

Safety and Risk Assessment

Whether or not we realize it, safety is an integral component of our lives. Throughout our daily activities, we all make decisions which are impacted by our understanding of safety. In a sense, each decision is a form of risk assessment—we weigh the risks and costs against the benefits, and make a decision. In most cases, the understanding of *safety*, meaning the risk and cost of hazardous outcomes, is based on intuition.

The analysis of safety is an active field of research. Techniques generated by this research are applied in major decision processes, often in business. As a result of safety research, many of the concepts and terms relating to safety and risk have been formalized. This chapter serves to provide background from the safety research literature as it applies to the development and application of the safety state model.

2.1 Safety From a Systems Perspective

Because of the intuitive understanding of risk and safety, the words used to describe these concepts can sometimes take multiple or ambiguous meaning. Therefore, it is important in safety analysis to clearly define the terms that are used.

The dictionary definition of safety refers to freedom from risk of human injury or death [49]. However, this represents an ideal and realistically unachievable goal—it is impossible to eliminate all risk. While complete elimination of risk is not achievable, reduction of risk is possible, although at a cost. Therefore, it is more appropriate to

discuss the *pursuit of safety*, in which the level of risk is traded off against the costs of reducing risk.

Lowrance [29] defines safety as the "judgment of acceptability of risk." This definition provides a working framework in which safety is separated into two components: subjective and objective. The subjective component, which is the judgment of acceptability, evaluates whether a given level of risk is acceptable to the society which is affected. Based on that judgment, policies are set which determine the trade-off between a level of risk and the resources expended to reduce that level of risk.

The objective component is risk assessment. There are a variety of definitions of risk in the safety literature. Rowe [38] defines risk as "the potential for unwanted negative consequences of an event or activity," alluding to the notion of chance. Lowrance [29] explicitly includes the probabilistic component, defining risk as the "measure of probability and severity of adverse effects." Rescher [36] echoes that idea: "Risk is the chancing of negative outcome. To measure risk we must accordingly measure both of its defining components, the chance and the negativity." Gratt [16] specifies the relationship between probability and severity in risk assessment by stating that the "estimation of risk is usually based on the expected result of the conditional probability of the event times the consequences of the event given that it has occurred." In discussing risk outcome, Wharton [50] offers that "a risk is any unintended or unexpected outcome of a decision or course of action," thus including both positive and negative outcomes. We shall consider a positive unexpected outcome to be a windfall bonus which is separate from risk.

Thus, risk includes the notion of both probability of occurrence and ultimate outcome of some undesirable event. Typically, an unexpected event which may lead to human injury or death is known as an accident. In most cases, an accident represents a demarcation point in time—the events occurring before the accident contribute to the probability of accident occurrence, while the events occurring after the accident are in the domain of outcome analysis.

Whenever we, as individuals, are part of a system which has potential for personal injury, safety is of concern to us The system in question might be a transportation

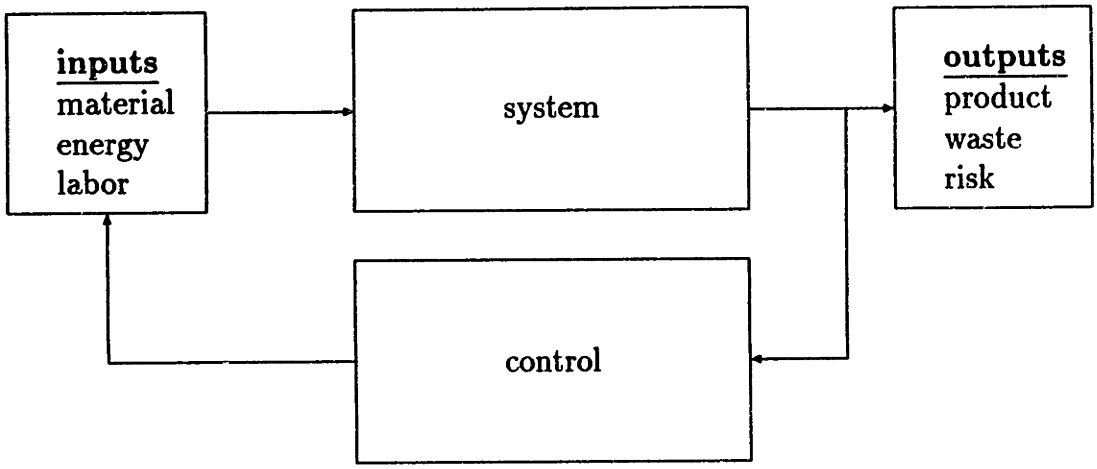

Figure 2-1: Closed Loop System Diagram

system used for commuting to the workplace, or might be a production line which is the workplace. Whatever the system in question, the goal of the pursuit of safety is to reduce the potential risk of personal injury.

Yet safety is but one component of the total picture. If we take a systems engineering perspective, a system can be considered to be an operator that transforms resources into a product or service. A manufacturing system is one which combines raw materials, machinery, and power to form a product. A transportation system uses vehicle and power resources to provide the service of moving people and goods from one place to another.

A generalized system can be represented as a black box operator (figure 2-1). Contained within the box is the equipment used by the system. Inputs to the system are the consumable resources, such as material, energy, and labor. Measurable units of production or service are output from the system. These outputs are used to evaluate the overall performance of the system.

In this context, risk is one of the outputs of a generalized system. Risk is not a physical output—there are no "risk widgets" tumbling off an assembly line. Rather, risk is considered an information output, which is but one component of overall sys-

tem performance evaluation. Whenever humans interact with the system, either as consumers of the system or labor in the system, there is potential risk of injury. Control of risk is the pursuit of safety, the goal of which is to minimize risk.

Risk performance can be controlled by modifying the system inputs. By changing the equipment used in a system or the resources that are put into a system, it is possible to cause change in the risk level of that system. Ultimately, these changes require resources—reductions in risk require expenditure of some combination of time, material, and money.

There is a tacit presumption that complete elimination of risk is an unattainable goal. Whenever there is human interaction with a system, there is some finite potential, no matter how slight, for injury from the system. Taking a fatalistic viewpoint, a truly probabilistic event (i.e., an event well-modeled by probability theory), will eventually occur given enough time. The only way to avoid the occurrence of such an event is to "get out of the game" before the event occurs. In fact, this is what happens to most of us with regard to rare catastrophic events.

Thus, we form the basis of risk exposure—given a constant risk probability, the expected number of failures over a defined period of time rises with the length of the period. The colloquial notion of risk exposure is that the "laws of probability catch up with you." While this presents a convenient rationalization, in fact the relationship is reversed—the occurrence of the accident provides statistical data to calibrate or validate the probabilistic model.

Through expenditure of resources, risks can be reduced. This is the task of risk management—strategically choosing the amount and application of resources used to reduce risk. Risk can be reduced either by reducing the probability of occurrence of the undesirable event, or by reducing the severity of the outcome when the event does occur. This is the basis for the distinction between *active safety* and *passive safety* devices, respectively. Active safety is a term typically applied to those devices or systems which assist in preventing accidents, while passive safety is applied to those devices or systems which reduce the severity of an accident when it does occur. In practice, the pursuit of safety involves a combination of the two.

The challenge of risk management is adequate determination of the relationship between safety expenditures and the resultant reduction of risk. This relationship is difficult to determine, in part because it is difficult to quantify the potential for risk. To further complicate matters, there is always some point of diminishing returns on investment—as risk is reduced, the relative expenditure required to further reduce risk is increased.

In one sense, perhaps it is easier to make decisions regarding safety investments when the relationship between expenditure and risk reduction is not well understood. There are difficult moral implications for decisions which involve trading off financial expenditure against risk of human injury. The Challenger tragedy and the Ford Pinto case are good examples of poor judgment in risk-expenditure tradeoff decisions, and the results were disastrous. Despite the challenge, moral implications underscore the importance of the pursuit of safety. Improvements in the quality of safety decisions are impossible without improving the objective understanding of risk.

A critical step is improvement of risk probability estimates. Risk probability is difficult to estimate, largely because most accident events are extremely rare. The information of highest interest is the chain of events that occur immediately prior to the accident. However, because of the rarity of the events, it is difficult to monitor for them effectively. In addition, accidents are often due to compounding of intermediate failures, the causality of which is often quite difficult to determine.

A guiding motivation in this work is the notion that *near collisions* are far more common than actual accidents. Near collisions are defined here as system states which have a higher probability of leading to failure than other operational states. If we have the capability of identifying near collisions and the conditions that lead to them, responses can be formulated to reduce the occurrence of near collisions, and ultimately reduce the number of accidents. The response may take the form of changes in design or operating procedure.

2.2 The Relationship Between System Reliability and Safety

Risk assessment is, in effect, a subset of reliability engineering. Reliability engineering is focused on estimating the probability and effects of system failures. Quantitative reliability techniques are used to predict the likelihood of failure or estimate the unavailability of a machine system. When a system failure can result in injury or death to a human, it becomes a safety issue. Assessment of system reliability with respect to a failure of this type becomes risk assessment.

In general, the occurrence of human injury is preceded by an event known as an accident. As defined by Senders and Moray [42], an accident is an "unwonted and unwanted exchange of energy." This definition captures two distinct characteristics of an accident: the event is unusual or unexpected, and it is undesirable.

Based on this definition of an accident, there are two key components of risk. Risk probability is the relative likelihood of an accident event, while risk outcome is the cost of that accident in terms of human injury. It is interesting to note the assymetry in the relationship between accidents and injury outcome—while it is clear that injury is the result of an accident, accidents do not necessarily result in injury.

In transportation systems, the accident event is alternatively known as a crash. Safety measures aimed at reducing the risk probability fall in the category of crash avoidance, while measures aimed at reducing the risk outcome are known as crash-worthiness.

In both highway and rail transportation systems, a significant proportion of the failures that lead to accidents are the result of human error [45]. The error can occur from a variety of sources. An operator error might lead to an undesirable system state. An error in design might lead to a component failure. A maintenance error might lead to a vehicle system that fails. In each of these examples, the precipitating error may subsequently cause an accident, either alone or in combination with other failures or conditions.

Human-machine interaction, by nature, raises the potential for safety issues. First,

safety has been defined as relative to the risk of human injury—if there is no interaction between human and machine, then by definition there is no safety problem. Second, human behavior is generally not deterministic. The inclusion of a human control element in a system represents a significant opportunity for error, which can lead to a higher level of risk.

Many researchers have generalized the relative capabilities of human and machine control elements [40][4][10][11][33]. The most succint summary is provided by Jordan [21]:

> Men are flexible but cannot be depended upon to perform in a consistent manner, whereas machines can be depended upon to perform consistently but have no flexibility whatsoever.

In many cases, the use of automatic control would eliminate the potential for human error leading to safety problems. Yet, for a variety of non-technical reasons, human controllers are often employed in systems where automation would be more sensible. Part of the problem lies with the difficulty of qualifying more sophisticated automatic control elements with regard to overall system safety. However, safety qualification of human-operated systems is at least as difficult, especially when the human operator must interface with increasingly complex systems.

Because of high public visibility, transportation systems have evolved into extremely reliable and safe systems. As a result, accidents have become relatively rare. While this, in itself, is a positive occurrence, it makes more difficult the task of further improving the reliability and safety of the system. The only true measure of the safety of a system is the occurrence of failures, and by improving the safety, we reduce the opportunity to measure the safety performance.

In addition, because many of the more obvious safety flaws have been discovered and rectified, the remaining safety problems involve the occurrence of complex combinations of precipitating events. In effect, we have moved from the realm of first order effects into higher order effects. This, in combination with the rarity of accidents, makes the task of identifying causal factors much more difficult.

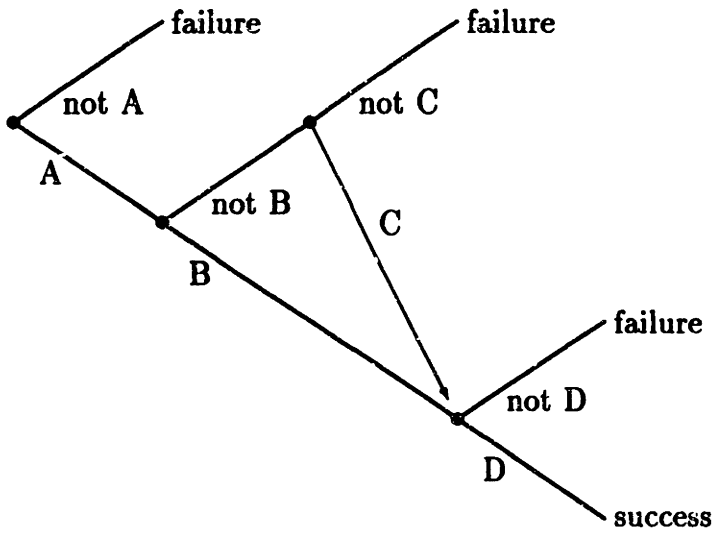

Figure 2-2: Example of a Simple Event Tree

Many techniques have been developed for quantitative analysis of system reliability and safety. In particular, the methods of fault tree analysis and event tree analysis provide useful background [48][28][32][15]. Both methods utilize a tree structure to organize the events which can lead to failures in complex systems. The event tree method is considered *forward-looking* in that the analysis commences with a precipitating event and explores the possible consequences in light of subsequent decisions and actions (figure 2-2). By comparison, the fault tree method is *backward-looking*—the analysis starts with a failure event and works backward to evaluate the possible combination of events that might have led to that failure (figure 2-3). Both techniques can be used either quantitatively or qualitatively. Fault tree techniques have been applied to highway systems analysis [22][25]. While fault tree and event tree methods are powerful techniques with wide application, an acknowledged weakness in both is difficulty in accounting for the time relation of events.

Markov process models have been employed in the related area of system reliability. Babcock [3] demonstrates the use of Markov process models to simplify the quantitative reliability analysis in fault tolerant systems. In this work, he contrasts the Markov process modeling technique to both fault tree methods and mean-time-to-

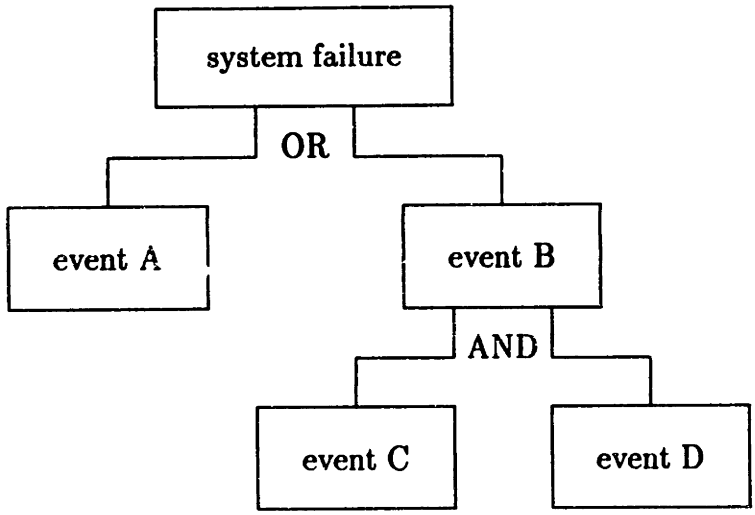

Figure 2-3: Example of a Simple Fault Tree

failure methods. In addition, several methods for reducing the size of the Markov network are demonstrated, to ease the loads of data storage and computation. Lewis [28] discusses the use of Markov process modeling to evaluate system reliability. He takes the significant step of using Markov states to represent the system states. In contrast, Babcock uses Markov states to represent events. The safety state model incorporates the method described by Lewis.

The safety state model is an extension and combination of event tree and fault tree models, using a discrete finite Markov process. By considering all combinations of the possible precipitating events, the safety state model covers all of the execution sequence paths included in the most general fault tree or event tree constructs. In addition, the time-based nature of dynamic risk probability is captured through use of the Markov process model. Thus, the safety state model represents a step forward in risk probability estimation techniques. The detailed development of the safety state model is contained in section 3.

2.3 Risk as a Dynamic Quantity

When comparing the relative safety of transportation modes, the traditional approach is to compare statistics relating the number of fatalities or injuries to a performance metric. For example, it would be reasonable to express risk as the fatalities per passenger-mile, or the injuries per vehicle-mile. While this approach is useful for comparing the safety performance of two or more competing modes of transportation, it does not provide for deeper analysis within one specific mode or system.

Consider, for a moment, highway safety. It is generally believed that human operator error is at least partly accountable for 85 to 90 percent of all highway accidents [45]. If we consider the human operator to be part of a closed-loop control system (figure 2-4), we see that there can be three potential types of operator error: failure to properly sense the system state, failure to make a correct decision, and failure to properly execute a decision. Yet, it would be very difficult to use accident statistics to identify the type of operator error.

Risk probability, especially in transportation systems, is not a static quantity. Instead, risk probability varies as a function of the state of the system, which includes the state of the vehicle, the state of the environment in which is travels (also known as the *wayside*), and the state of the operator. The system state in transportation systems is quite dynamic with respect to time. Operators are responsible for a constant stream of control decisions, the action of which determine the state of the vehicle in relation to the state of the environment. Thus, through these control decisions, the operator has a profound impact on the risk probability of the system. Many accident scenarios are the result of compounding several hazard conditions, each of which may be relatively innocuous when occurring in isolation. Some hazards may be due to operator errors, while others may be due to machinery failures in vehicle or wayside equipment. The collected set of potential hazard conditions leading to a particular accident scenario can be considered a system state. Because this state varies with time, and the risk probability is a function of this state, risk probability can also be considered to be a function of time.

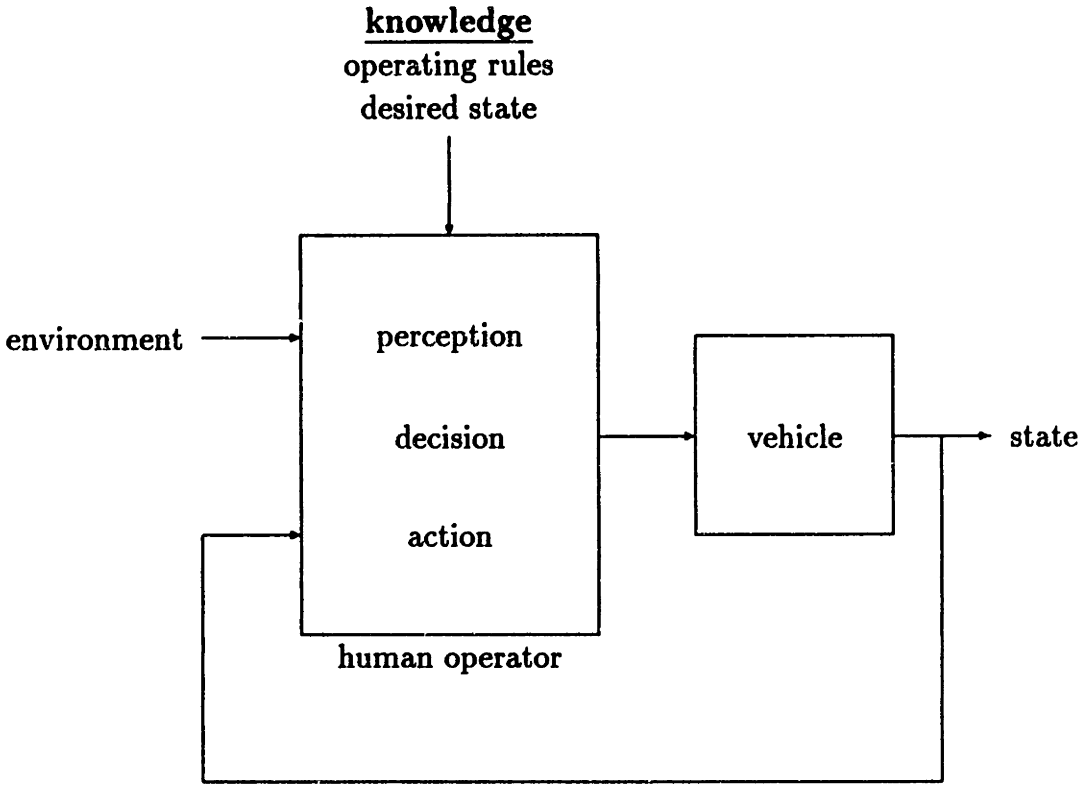

Figure 2-4: Human Operator in Closed-Loop System Control

Time is an integral component of dynamic risk probability. Using probability theory, we can model the risk probability as the relative likelihood of the occurrence of an accident. However, the risk probability of an accident only makes sense if we compare its occurrence to the alternative event, which is the non-occurrence of an accident. Since nothing "happens" during the non-occurrence, the event can only be considered with respect to some fixed metric. The safety state model considers the probability of an accident with respect to a fixed time frame, known as a *time slice*. Thus, the risk probability represents the relative likelihood of an accident in a single time slice. On the average, it also represents the percentage of time slices that result in an accident. An alternate expression is in terms of the mean time (number of time slices) between occurrence of accidents. This form is commonly known as the mean time to failures (MTTF), [1] and is used extensively in the field of reliability engineering. The relationship between risk probability and MTTF plays an integral role in the development of the safety state model.

2.4 Evaluating System Risks

One of the issues in system safety is evaluation of system risks. In effect, the goal is to identify the conditions and scenarios that might result in a high-risk situation.

In general, system risks tend to be the result of human error of some form. Human error can take many forms and can occur at a variety of points in the design and operation of a system. There are three particular points that are of concern: design, implementation, and operation.

In the design phase, human error might cause an error in the specification of a parameter, a component, or a subsystem. The resultant design may not meet the design specifications closely enough. If the system is complex, the fault may go undetected for an extended period of time. Note that the specification of test

[1]The *mean time to failure* (MTTF) is defined as the elapsed time between the onset of system operation and the occurrence of failure. The related *mean time between failure* (MTBF) nomenclature is used to distinguish the case where the failures are repaired and the system is returned to operation. Since we consider transportation accidents to be irreversible, we use the MTTF notation.

procedures is, in itself, a design task, and may contain an error which allows another design error to remain undetected.

In the implementation phase, a human error manifests itself as a error in translation from the design specification to a physical object. Such an error might cause a part to be insufficiently constructed, possibly resulting in a failure that was considered and accounted for during the design phase.

The operation phase is the point of human error which is most familiar, if only because many people have direct experience. In this phase, the human error can be characterized as an inability to perform the operation task as directed or trained. Using figure 2-4 as a guide, the error may take the form of an incorrect perception, an incorrect decision, or an incorrect action.

Ostensibly, the goal of evaluating operator performance is to identify the human errors in the oeprational phase. However, the deeper goal of this work is to enable the identification of all forms of human error that lead to system risks.

2.5 Controlling Operator Behavior

Let us now extend the closed-loop control paradigm to a different level. Consider a control loop, where the "plant" is the operator (figure 2-5). The output of the "plant" is the safety-related behavior, and the controller is intended to regulate the safety-related behavior through education.

In many forms of operator training, there are two possible impediments to this system. First of all, there is no effective objective method for measuring the safety-relevant behavior of the operator. Without this measure, it is virtually impossible to regulate the behavior. Secondly, operator training, which is the input to the plant, occurs only for a relatively short period of time, which typically occurs at the early period of operator experience. In effect, these factors lead to a control loop which is not closed.

Of course, the control loop described here sometimes gets closed through experience. After experiencing one or more accidents (or, perhaps, near collisions), an

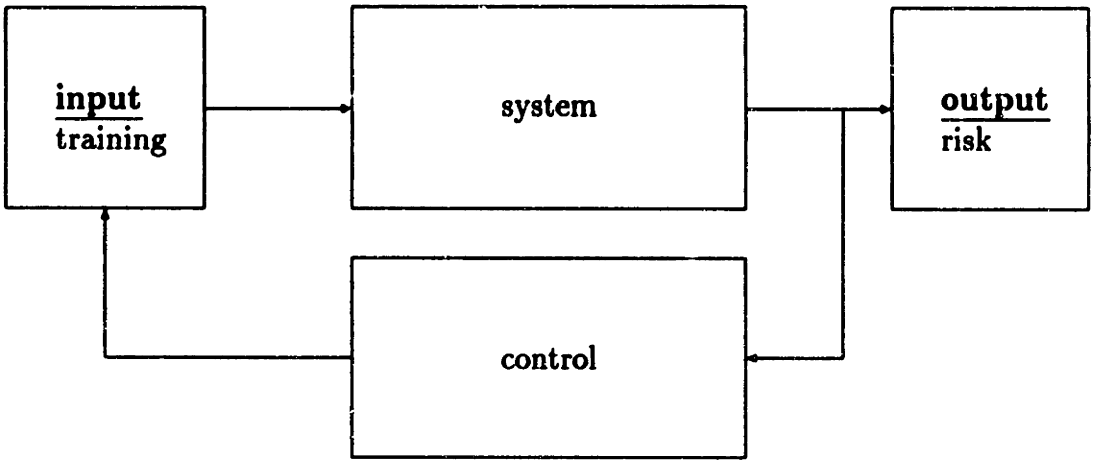

Figure 2-5: Closed-Loop Control of Operator Behavior

operator may learn to avoid certain system states that might lead to an accident, provided the accidents experienced were not fatal. However, the control authority in this case is quite limited.

One potential application of the safety state model is to provide an effective measure of the safety-related operator performance in a dynamic sense. This, combined with continuing education, will allow effective closed-loop control of operator safety performance in the absence of accidents.

Chapter 3

Theoretical Development

In this chapter, we develop the theory that allows expression of dynamic risk probability as a function of system state. This theory is centered on a model of system behavior, which can be used to predict future system behavior based on observation of previous behavior.

The systems considered include both machine and human operators. Any human that interacts with the system and can change system behavior is considered an operator, whether or not that person is trained for the operational task. Although deterministic models are often useful for machines, humans are much more variable in their perception, decision, and actuation processes. As a result, a stochastic model is more appropriate for modeling a human-machine system. The approach utilizes finite-state Markov process model theory.

3.1 Brief Review of Markov Process Theory

As background to Markov process theory and analysis, it is useful and appropriate to clearly define what is meant by system state. In the field of system dynamics, the state of the system can be represented as a vector of numbers, each of which is an expression of a continuous variable corresponding to a physical entity. Such variables might be measured by an instrument, or estimated by a mathematical model.

As an example, consider the dynamic state of a rail vehicle. If we know the path

of the tracks and assume that the tracks do not move, it might be sufficient to express the state of the vehicle in terms of instantaneous speed and position along the tracks. Depending on the desired analysis, we may include other state variables, such as the available voltage from the catenary, or the temperature of the wheel bearings.

If a state vector has n elements, it can be considered a vector in n-space. If each unique collection of state variable values is considered to be a distinct state, then there are an infinite number of states possible in this n-space.

In a finite Markov process, the number of allowable states is limited to a finite positive integer. For the moment, let us ignore the precise definition of those states, except that they are mutually exclusive (i.e., the system cannot be in more than one Markov state at any given time). Definition of the states is appropriately done at the point of application, when the problem to be solved is more clearly articulated. Each individual state has an identification number associated with it; the number serves as a label for the state. In most cases, the numbers constitute a sequential set of non-negative integers—an n-state Markov process would have state numbers from 1 through n (or perhaps 0 through $(n-1)$. The notation $S(i)$ indicates that the system is currently in the state labeled i.

A useful metaphor, used heavily by Howard [20], is a set of lily pads on a pond. Each lily pad represents a Markov state. A frog, representing the system, jumps from lily pad to lily pad. The jumps are instantaneous, so the frog is on only one of the lily pads at any point time. Whenever the frog moves from one lily pad to another, a *state transition* has occurred.

A simple example to illustrate this concept is a coin tossing process. Suppose we are interested in the combinations of the three most recent coin tosses. At this point, we will not assume that the coin is fair. There are only 8 possible combinations: HHH, HHT, HTH, HTT, THH, THT, TTH, and TTT (where the last letter represents the most recent toss in the sequence). We are only concerned with the three most recent tosses—anything prior to that is irrelevant. Let us assign each of these combinations to a Markov state, as shown in figure 3-1. Each circle represents one of the states, and the directed lines represent possible transition paths. We can see that it is possible

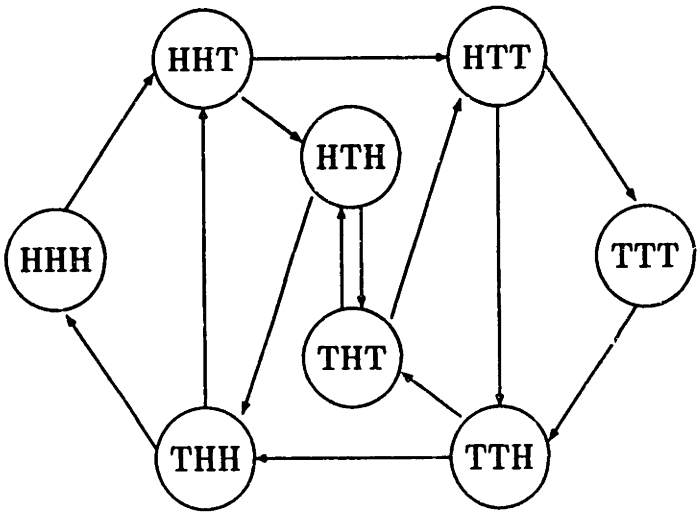

Figure 3-1: Example of Simple Markov Process

to go from state HHH to HHT with the toss of *tails*, but it is impossible to go from HHH to HTH, because we cannot change the state of the previous toss. Each of the states is assigned a state number. In this example, the selection of state numbers has no meaning beyond identifying the individual states.

In the discrete Markov process, the opportunity for a state transition (i.e., the chance for the frog to jump from one lily pad to another) occurs at specified points in time. In the coin toss example, these state transition opportunities occur each time the coin is tossed. In dynamic systems, an alternative approach is to define transition opportunities which occur at fixed intervals in time. Note that these are couched as *opportunities*—at a transition point, the system may change state, or it may remain in the same state. In the coin toss example, this corresponds to the path that leads from HHH back to itself.

If we consider that each transition path has a probability associated with it (figure 3-1), we can organize these probabilities in the form of a matrix. The matrix is known as the *state transition matrix* and denoted as P. Each element, $p_{i,j}$, is the probability of transition from state $S(i)$ to state $S(j)$. For a Markov process with n states, this state transition matrix will be of dimension $(n \times n)$. By convention, each row corresponds to the state $S(i)$ before the transition, and the columns correspond

to the state $S(j)$ after transition. Because the probabilities of all paths leading from any one path must sum to 1, it follows that each row of the state transition matrix must also sum to 1.

$$P = \begin{bmatrix} p_{1,1} & \cdots & p_{1,n} \\ \vdots & p_{i,j} & \vdots \\ p_{n,1} & \cdots & p_{n,n} \end{bmatrix} \tag{3.1}$$

If a Markov state has one or more transition paths leading to it but no transition paths leading away, it is known as a trapping state. Once the system enters a trapping state, it remains there permanently. For trapping state $S(k)$, the corresponding row in the state transition matrix has a value of 1 for $p_{k,k}$, and 0 for all other entries.

The state transition matrix serves to summarize the probabilistic nature of the Markov process at each transition point. In addition, it can be used to forecast the behavior of the system for an arbitrary number of transitions in the future. Consider the row vector $\pi(0)$, of dimension $(1 \times n)$, which contains a value of 1 in the position of the current state and 0 everywhere else. Since each row of the state transition matrix contains the probabilities of transition from the current state to any of the possible next states, we can multiply $\pi(0)$ by P to find $\pi(1)$, which summarizes the probability of being in any of the other states after one transition (equation 3.2).

$$\pi(1) = \pi(0)P \tag{3.2}$$

Let us now look ahead one additional step. We can calculate $\pi(2)$ in the same manner as $\pi(1) = \pi(0)P$, based on our knowledge of $\pi(1)$.

$$\pi(2) = \pi(1)P \tag{3.3}$$

Substituting $\pi(0)P$ for $\pi(1)$ (from equation 3.2, we can evaluate the probability of being in any of the other states at the second transition, based only on knowledge of

the current state and the state transition matrix (equation 3.3).

$$\pi(2) = \pi(1)P = \pi(0)PP = \pi(0)P^2 \tag{3.4}$$

Continuing the iteration, we can see that this relationship can be summarized in terms of the initial state and the power of the state transition matrix (equation 3.5).

$$\pi(\tau) = \pi(0)P^\tau \tag{3.5}$$

Equation 3.5 is known as the Chapman-Kolmogorov Equation [37]. Traditionally, the power of the state transition matrix is notated as $\Phi(\tau)$, following the relationship:

$$\Phi(\tau) = P^\tau \tag{3.6}$$

with

$$\Phi = P^\infty \tag{3.7}$$

being the limit of P^τ as τ goes to infinity. In effect, Φ is a summary of the limiting behavior of the system, as time proceeds forward from the current state to infinity.

In the case of Markov processes having one trapping state, Φ will always have a single column of ones in the column corresponding to the trapping state, and the remainder of the matrix will be zero. This makes sense intuitively—given an infinite number of transitions, and the system will ultimately reach the trapping state, which can never be left, regardless of any of the transient states occupied.

3.2 The Safety State Model—An Application of Finite Markov Processes

Our ultimate goal is to find a model which allows estimation of dynamic risk probabilities. As mentioned earlier, the motivation is straightforward—existing methods of safety analysis only provide mechanisms for determining the overall average safety

of a system, without regard to individual operators or system design features. To evaluate the safety-related behavior of a system with respect to individual operators or system design features, we need a dynamic estimation of risk probability.

Ideally, such a model would provide an instantaneous risk probability value, and would be an analytical function which uses all of the pertinent continuous state variables as input. Such a model would be deterministic by nature. But humans are not deterministic, as best we can model them, especially in response to real or perceived emergency situations. The difficulty in obtaining a deterministic model of human behavior hinders any efforts toward finding an analytic risk probability function. Another problem is the calibration of such a model—how do you obtain the values of the pertinent parameters? Still another problem is resolution of the model—how sensitive is the model to variations in the parameters?

An intermediate approach is to divide the state space into a set of *domains*. By calculating the probability of reaching the failure event from any of these domains, we achieve a more coarse estimate of dynamic probability than we might achieve with an analytical model form. These domains are the states in the discrete Markov process.

The development of the model requires several steps. First, the structure of the model is developed. In this step, the method for assigning Markov states is defined. Next, the method for determining the state transition probabilities is developed. Then, the algorithm for calculating the mean time to failure is derived. Finally, the technique for estimating the risk probability is described.

3.2.1 Structure of the Safety State Model

The safety state model is an application of a discrete Markov process model with a single trapping state. A set of domains in the continuous system state space is mapped into a set of Markov states, which forms the model. To define these domains, let us consider a set of n binary conditions (i.e., statements that are either true or false). All of these conditions are presumed independent, in the sense that the state of each

condition does not necessarily imply the state of any other.[1] For example, consider the following two conditions: 1) the vehicle is traveling faster than the speed limit, and 2) the operator has applied the vehicle brakes. Neither one of these conditions necessarily implies the other.

Each condition can be represented by a single bit of information (i.e., a 1 or a 0), corresponding to the state of the condition. If we concatenate the resultant set of n bits into a single number, that number can have a value from 0 to $(2^n - 1)$, and that set of 2^n numbers represents all of the possible combinations of those n conditions.

In the earlier review of Markov process analysis (section 3.1), we sidestepped the issue of defining the Markov states. The definition of these states is part of the application of the Markov process. For the earlier example problem (figure 3-1), the appropriate set of Markov states consisted of the eight possible coin toss histories in the event space.

The definition of the Markov states in the safety state model form the structure of the model. By using the set of n conditions to define a number which is then interpreted as the Markov state, we have created a link between the definition of the state and the state number used to represent that state. In the general case of a Markov process, this is not necessarily true—the state number is merely a label. In the safety state model application, the state number carries with it the definition of the state. Therefore, the set of Markov states represents the exhaustive list of all possible combinations of the specified conditions. As a result, the safety state model allows comprehensive investigation of many interacting conditions. Using other methods, such investigation would be tedious and time-consuming, if at all possible.

Defining the potential causal conditions defines the structure of the model. It is important to note that the causal factors are described as conditions, and not as events. A condition is defined as a state of being, while an event can be defined as demarcation point in time, which might signal the beginning or end of a condition.

[1]The definition of independence of random variables states that $p(y|x) = p(y)$ [6]. In this sense, the conditions described are not independent—in fact, we use the safety state model to determine the stochastic inter-dependence between the specified conditions and the failure event.

This is an important distinction—an event is a point in time, while a condition occurs over some period of time. An event marks the beginning or end of the occurrence of a condition.

The safety state model is based on the notion that there are a variety of condition combinations that might lead to a particular failure. Each condition is binary—it is either true or false. The binary condition may be based on a continuous state variable. For example, if vehicle speed is considered to be a factor, a binary condition based on speed could be whether the speed is above or below a specific threshold.

The set of binary conditions are then concatenated into a binary number, which is used as the Markov state. This implies that there is an individual Markov state for each possible combination of the specified conditions. There is also the implication that the number of states grows as a power of 2. That is, if there are n conditions specified, there are 2^n possible combinations of those conditions, leading to 2^n Markov states. In this context, these Markov states are known as fl states, as they represent the non-failure operation of the system. In Markov process nomenclature, these states are known as *transient states.*

In certain cases, we can combine a number of conditions into a fewer number of bits. This can be done only if the subset of conditions represent a mutually exclusive set—that is, no more than one of the set can be true at any one time. If we have m mutually exclusive conditions, we can combine them into b bits in the safety state number, provided that $m \leq 2^b$.

For example, consider a rail vehicle that has three types of automation available to the operator: cruise control, programmed stop, and autopilot. (The details of these systems will be described later. For the moment, we will just assume that they exist.) Only one of these modes can be active at any time; choosing one disables any other that might be active. If we also consider the case when none are active (known as the manual mode), then we have four possible states, which are mutually exclusive. These can be expressed using two bits of the safety state number, as two bits allows up to four distinct states.

One additional Markov state must be specified—the *failure state* is defined as

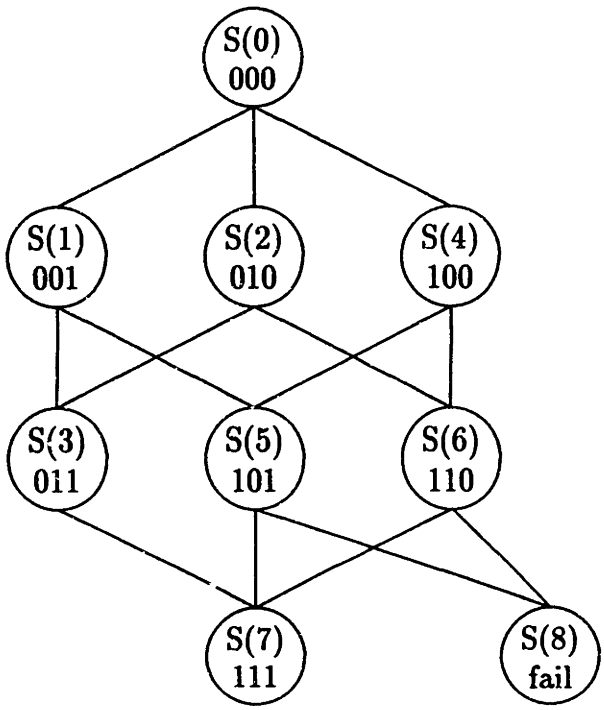

Figure 3-2: Example of Three Condition Safety State Model

a trapping state (meaning that, once entered, the state is never vacated), and it is predicated by the defined failure event. That is, occurrence of the failure event results in the Markov transition to the failure state. The definition of the failure state as a trapping state corresponds to the irreversible nature of a failure event.

The structure that results from the definition of the safety state model provides a framework for observing an existing system. Once we have defined the failure event and the causal conditions of interest, one may observe an operational system and record the state occupancies and transitions. Such an observation results in a safety state trajectory, which is simply the statement of occupied state as a function of time.

A simple example of a safety state model is shown in figure 3-2. Consider that there are three conditions in this example, leading to eight states plus the failure state. This diagram shows the possible transition paths between the defined states, assuming that only one of the conditions can change in a state transition period.

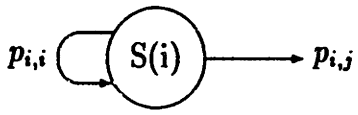

Figure 3-3: Single State Transition Paths

3.2.2 Finding the State Transition Probabilities

The next task is to determine the method used to obtain the probabilities for state transition matrix. In order to do this, we need to first consider the micro-actions of the model. Assume that the model is currently in state $S(i)$, and from $S(i)$ can transition only to state $S(j)$ (figure 3-3). At the next transition point, the system might remain in state $S(i)$ (with a probability of $p_{i,i}$), or it might transition to some other state $S(j)$ (with a probability of $p_{i,j}$). Therefore, if the system enters state $S(i)$, remains in that state for k transition periods, and then transitions to state $S(j)$, we can estimate that the probability $p_{i,i}$ is equal to $(k-1)/k$, and the probability $p_{i,j}$ is $1/k$. In other words, of the k transition opportunities during which the system was in state $S(i)$, $k-1$ resulted in holding the state $S(i)$, while one involved transition to state $S(j)$. Therefore, the relative likelihood (probability) of holding in state $S(i)$ is $(k-1)/k$, and the relative likelihood of going to state $S(j)$ is $1/k$.

To generalize this technique, the state transition probabilities are generated by collecting statistics about the state occupancies. By counting the occurrences of state holding times and transitions, the relative likelihood of each transition is computed. This is accomplished through the use of a construct known as the statistics matrix. This matrix is analogous to the state transition matrix, and is used for accumulation of the state occupancy statistics. The matrix is denoted by S, and the individual elements are $s_{i,j}$. Each $s_{i,j}$ takes an integer value, counting the number of times that the transition from state $S(i)$ to $S(j)$ has been experienced.

Data from the observed safety state trajectories are accumulated into the statistics matrix. For each state occupancy, we need to know the current state ($S(i)$), the next

state ($S(j)$), and the time spent in the state (t_i, expressed as a count of the transition intervals). We then increment $s_{i,i}$ by $t_i - 1$ and increment $s_{i,j}$ by 1 (corresponding to the number of times each transition was experienced). The process is iterated for all the available safety state trajectories.

When all the safety state trajectory data have been incorporated, the state transition matrix is computed from the statistics matrix. This is done by computing the sum of each row in the statistics matrix,

$$s_i = \sum_j s_{i,j} \tag{3.8}$$

and dividing that sum into each element of the row.

$$p_{i,j} = \frac{s_{i,j}}{s_i} \tag{3.9}$$

As an example, let us consider a three-condition safety state model (as illustrated in figure 3-2). There are nine states, numbered 0 through 8. Assume that we have observed that the system has entered state $S(1)$ on 12 different occasions. The total holding time in state $S(1)$ was 88 transition periods, and of those 12 occupancies, the system had transitions to state $S(0)$ on 3 occasions, to state $S(3)$ on 6 occasions, state $S(5)$ on 2 occasions, and state $S(8)$ once. The total time in $S(1)$ was 100 transition periods. The resultant state transition probabilities for state $S(1)$, corresponding to row 1 in the state transition matrix, would be estimated as shown in table 3.1.

It is a distinct possibility that, during our system observations, certain states are not occupied. This does not imply that these states will never be occupied in the future, only that they were not occupied during our observation. To account for the possibility that states might conceivably be occupied, there is an initial default state of the state transition matrix. This initial state assumes the most general form until we have gain experience from observing the behavior of the actual system. Thus, if the state were to be occupied, it is assumed to be occupied for a default amount of time, and the distribution of the transitions out of this state is equal among the

- $p_{1,0} = 3/100 = 0.03$
- $p_{1,1} = 88/100 = 0.88$
- $p_{1,2} = 0$
- $p_{1,3} = 6/100 = 0.06$
- $p_{1,4} = 0$
- $p_{1,5} = 2/100 = 0.02$
- $p_{1,6} = 0$
- $p_{1,7} = 0$
- $p_{1,8} = 1/100 = 0.01$

Table 3.1: Example, State Transition Probability Calculation

possible states.

To accomplish this, the statistics matrix is initialized with a set of default values. The initial holding time values ($s_{i,i}$) are set to a value which is a function of the network size. Transitions are assumed to be possible between states that are one bit away from each other—this corresponds to the notion that state transitions are based on a change in a single condition. These assumed possible transitions are assigned a value of one, and all others are assigned a value of zero. Until the risk event actually occurs, we make no assumptions about the ability of the system to transition to this state. As a result, the column of $s_{i,n}$ values are set to zero.

3.2.3 Finding the Mean Time to Failure From Each State

Let us assume we have a discrete Markov process with a single trapping state. Let us also assume that a state transition occurs at fixed regular intervals in time. Thus, a count of state transitions can be equated with a period of time, as the product of the number of transitions and the length of the time interval. The system may remain in a given state for longer than one interval—this is accounted for with the holding probability, which is the probability for each state that the system will "return" to

that state at the transition point.

One of the fundamental questions to be answered is as follows: Given a particular starting state, what is the mean time for the process to reach the trapping state? This question is of particular importance if we consider the trapping state to indicate an undesirable failure event. In this case, we are looking for the mean time to failure (MTTF).

Let us consider a large discrete Markov process with n transient states and one trapping state. The states are numbered from 0 to n, with $S(n)$ as the trapping state. The P matrix is of dimension $((n+1) \times (n+1))$, and has the form

$$P = \begin{bmatrix} p_{0,0} & p_{0,1} & \cdots & p_{0,n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{n-1,0} & p_{n-1,1} & \cdots & p_{n-1,n} \\ p_{n,0} & p_{n,1} & \cdots & p_{n,n} \end{bmatrix} = \begin{bmatrix} P_a & P_f \\ 0 & 1 \end{bmatrix} \tag{3.10}$$

where P_a represents an $n \times n$ sub-matrix of the transition probabilities between active (a.k.a. transient) states, P_f represents an $n \times 1$ column vector of the transition probabilities into the failure (trapping) state directly from the active states, a $(1 \times n)$ row vector of zeros (in the last row), and a single element of 1 for $p_{n,n}$.

$$P_a = \begin{bmatrix} p_{0,0} & \cdots & p_{0,n-1} \\ \vdots & \ddots & \vdots \\ p_{n-1,0} & \cdots & p_{n-1,n-1} \end{bmatrix} \tag{3.11}$$

$$P_f = \begin{bmatrix} p_{0,n} \\ p_{1,n} \\ \vdots \\ p_{n-1,n} \end{bmatrix} \tag{3.12}$$

The probability of occupying any of the possible states, as a function of the initial

state, after τ transitions can be expressed by:

$$\Phi(\tau) = \begin{bmatrix} \Phi_a(\tau) & \Phi_f(\tau) \\ 0 & 1 \end{bmatrix} = P^\tau = \begin{bmatrix} P_a^\tau & (\sum_{i=0}^{\tau-1} P_a^i)P_f \\ 0 & 1 \end{bmatrix} \tag{3.13}$$

Note that the partitioning which occurs is similar to that found in the P matrix (equation 3.10), with the last column representing the probability of getting to the trapping state after τ transitions. The matrix Φ is the limit of $\Phi(\tau)$ as τ goes to ∞ (equation 3.14).

$$\Phi = \lim_{\tau \to \infty} \Phi(\tau) \tag{3.14}$$

In the case of the P matrix for the safety state model, Φ will always have a column of ones in the rightmost column, and zeros elsewhere.

$$\Phi = \begin{bmatrix} 0 & \cdots & 0 & 1 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & 1 \end{bmatrix} \tag{3.15}$$

However, in order to obtain the mean time to failure (MTTF), we are more interested in the probability distribution of getting to the failure state on the τ^{th} transition, which can be expressed as:

$$\Psi(\tau) = \Phi_f(\tau) - \Phi_f(\tau - 1) = \left[\sum_{i=0}^{\tau-1} P_a^i - \sum_{i=0}^{\tau-2} P_a^i\right] P_f = P_a^{\tau-1} P_f \tag{3.16}$$

This is the first-order interarrival probability distribution. Because $P_a^\tau = P_a P_a^{\tau-1}$, we know that $P_a^{\tau-1} = P_a^{-1} P_a^\tau$. Therefore, we can say that:

$$\Psi(\tau) = P_a^{\tau-1} P_f = P_a^{-1} P_a^\tau P_f \tag{3.17}$$

The iterative method for calculating MTTF is obtained by multiplying the probability of the failure event occurring on the τ^{th} transition by the number of transitions.

This can be stated as

$$M = \sum_{\tau=0}^{\infty} \tau \Psi(\tau) \tag{3.18}$$

where M is a $((n-1) \times 1)$ row vector containing the MTTF for each of the $(n-1)$ operational states.

The disadvantage of this approach is that it is difficult, if at all possible, to predict the number of iterations required to achieve an acceptable answer. In addition, the solution is expressed as an infinite sum—it is also difficult to determine the bounds that will qualify an acceptable answer, even if we could predict the number of iterations required.

An alternative approach involves the use transform analysis. The MTTF can be calculated using z-transform analysis. By evaluating the differential of the z-transform at zero, the MTTF can be calculated directly.

$$M = \frac{d}{dz} \Psi(z) \Big|_{z=0} \tag{3.19}$$

Using the definition of $\Psi(\tau)$ from equation 3.17, we can evaluate the z-transform as:

$$\Psi(z) = P_a^{-1} \left[I - P_a z \right]^{-1} P_f \tag{3.20}$$

Substituting equation 3.20 into equation 3.19, the MTTF is:

$$M = \frac{d}{dz} \Psi(z)|_{z=0} = P_a^{-1} \left[I - P_a \right]^{-1} P_a \left[I - P_a \right]^{-1} P_f \tag{3.21}$$

This expression can be further simplified. We know that

$$\left[I - P_a \right]^{-1} = I + P_a + P_a^2 + P_a^3 + \ldots \tag{3.22}$$

which means

$$\left[I - P_a \right]^{-1} P_a = P_a \left[I - P_a \right]^{-1} \tag{3.23}$$

allowing us to simplify equation 3.19 to

$$M = P_a^{-1} P_a [I - P_a]^{-1} [I - P_a]^{-1} P_f = [I - P_a]^{-1} [I - P_a]^{-1} P_f \tag{3.24}$$

Furthermore, we know that

$$P_f = \begin{bmatrix} 1 - p_{0,0} - p_{0,1} - \ldots - p_{0,n-1} \\ 1 - p_{1,0} - p_{1,1} - \ldots - p_{1,n-1} \\ \vdots \\ 1 - p_{n-1,0} - p_{n-1,1} - \ldots - p_{n-1,n-1} \end{bmatrix} = \underline{1} - P_a \underline{1} = [I - P_a] \underline{1} \tag{3.25}$$

where $\underline{1}$ is a $((n-1) \times 1)$ column vector of ones. Using this, the expression can be simplified further to

$$M = [I - P_a]^{-1} \underline{1} \tag{3.26}$$

An alternative derivation utilizes the method for computing expectation by conditioning [37]. A recursive computation is derived for the MTTF from each state $S(i)$, as shown in equation 3.27. [2]

$$M_i = p_{i,0} M_0 + p_{i,1} M_1 + \ldots + p_{i,n-1} M_{n-1} + 1 \tag{3.27}$$

In vector form, this is expressed as

$$M = P_a M + \underline{1} \tag{3.28}$$

leading to an identical solution.

$$M = [I - P_a]^{-1} \underline{1} \tag{3.29}$$

This solution has been verified experimentally by the author, using a 7 condition (129 state) Markov model. Using the iterative approach, the calculations required

[2] In the general case, this is expressed as $E[X] = \sum_y E[X|Y = y] p\{Y = y\}$.

approximately 300,000 iterations to get to a reasonable but inexact solution. These calculations required approximately 20 hours on a Silicon Graphics Indigo-2 machine. The closed-form solution required less than 10 seconds to get an exact solution.

3.2.4 Converting MTTF to Risk Probability

In the previous section, we have developed a method for calculating the mean time to failure for each non-failure state in the system. However, this is not enough—we are really interested in estimating the risk probability at each operational state in the system. The probability of transition directly from each operational state to the failure state is expressed as P_f. However, for most of the states, this probability is zero, meaning that it is not possible to go directly from a specific state to the failure state.

On the other hand, we know that the model will always eventually reach the failure state, given enough opportunity—this is inherent in a Markov process with a single trapping state (as shown in equation 3.15). What we really want to know is the relative likelihood of reaching the failure states, based on the current system state.

To accomplish this, the MTTF vector is transformed to an equivalent risk probability. Let us assume that we have a Bernoulli process (i.e., the "coin toss"), which has probability p_f of failure and probability $(1 - P_f)$ of success, occurring at points in time corresponding to each transition point. Because the failure event is rare, we expect p_f to be small. The MTTF for such a system is equal to $1/p_f$.

In the safety state model, we have calculated the MTTF for each non-failure state of the Markov process. By assuming the form of a Bernoulli process, we can thus estimate the equivalent risk probability R_i as $1/M_i$. The MTTF vector M is transformed into a risk probability vector R by the element-wise transformation.

$$R_i = 1/M_i \tag{3.30}$$

This estimate underscores a very important concept: Although it may be impossible to reach the failure (trapping) state directly from a specified state, by nature

the trapping state will always be reached eventually, regardless of initial or transient state. Restated, although the probability of going directly from an operational state to the failure state may be zero, the probability of eventually reaching that state is one. This corresponds to the notion that there is always a finite risk in an operational transportation system. Computation of the equivalent risk allows us to express the differences in risk between the various defined states in the system.

3.2.5 Characteristics of the State Transition Matrix

As shown earlier, the state transition matrix is used to compute the MTTF, and the MTTF is used to compute the equivalent risk probability. Thus, there is a direct link between the state transition matrix and the resultant risk probability estimation vector. What is the nature of the "form" or "shape" of the state transition on that risk probability estimation vector? And what do the risk probability numbers really mean?

We expect that, given a suitable time scale for the transition point intervals, the number of periods spent in each state (holding time) is significantly greater than the number of times the state is visited. Or, stated differently, the average hold time for each state is substantially greater than the transition intervals.

Thus, if we view the state transition matrix as a mesh diagram, we expect the topography to appear as a row of "mountains" down the main diagonal, with shorter "hills" scattered sparsely about the remainder of the area. In a mesh diagram, the horizontal axes represent the rows and columns of the two-dimensional state transition matrix, and the height of the surface represents the values of the individual state transition probabilities. An example of a state transition matrix is shown in figure 5-1.

3.3 Implementation—Four Phases of Analysis

In the process of applying the safety state model to a practical risk analysis problem, there are four distinct phases of analysis. These are, in rough chronological order,

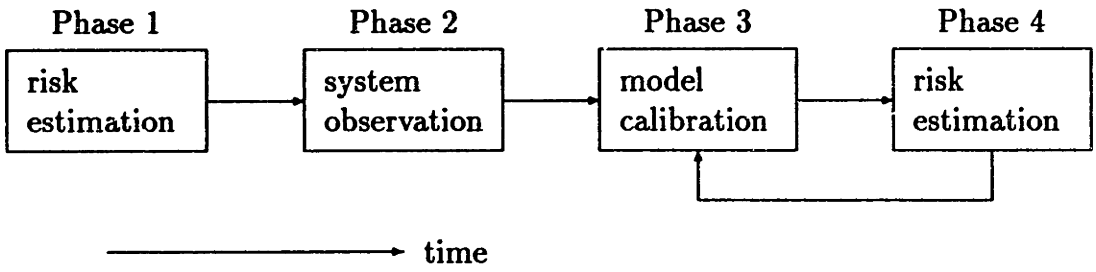

Figure 3-4: Logical Flowchart for Applying Safety State Model

the risk identification phase, system observation phase, model calibration phase, and risk estimation phase. Each phase represents a distinctly different type of activity, and can be represented with a set of inputs and outputs. A flowchart of these phases is shown in figure 3-4.

The risk identification phase is the starting point for the analysis. It is in this phase that the analyst clearly identifies the risk event of interest and the conditions that might lead to it. These define the size of the required safety state model. The input of this phase is a problem, while the output is a failure event and a set of conditions that are believed to lead to that failure.

The system observation phase is the period of time during which an operational system is observed and its behavior is recorded. The input to this phase is a failure event and set of conditions that might contribute to its occurrence (i.e., the output of the risk identification phase), and the output is a set of safety state trajectories representing measured system behavior.

At the commencement of this phase, the analyst must transform the failure event and set of contributing conditions into a set of measurements to be made on the system. Each condition of interest must be defined in terms of events that determine the beginning and end of that condition. The required measurements may be of two possible forms: event recording, and continuous variable recording. The former is appropriate for events that take place, such as operators pressing buttons or compo-

nents failing. The latter form is appropriate for measuring variables which truly are continuous in nature, such as vehicle speed or position, or perhaps the position of a control lever.

Once the specific system measures have been determined, the system is instrumented and observed. In practice, this phase represents the most labor intensive, and consumes the largest proportion of the project time. Because of the amount of effort required in this phase, it is appropriate, in the risk identification phase, to overspecify the conditions to be used. By identifying a large number of conditions, we assure ourselves that enough data will be collected during the system observation phase.

The model calibration phase consists of a series of calculations, which transform the measured system behavior from the system observation phase into the dynamic risk probability function used by the risk estimation phase. This is the phase which utilizes the Markov process model. The observed safety state trajectories are incorporated into the statistics matrix, which is used to determine the state transition matrix. The state transition matrix, in turn, is used to calculate the MTTF array, which is then used to determine the risk probability function.

The risk estimation phase is the point at which everything comes together—the risk probability function (output from the model calibration phase) is applied to the safety state trajectories (output from the system observation phase), transforming them into risk probability trajectories as a function of time. These represent the grand output of the entire process.

In practice, safety state model analysis might not necessarily proceed in such a linear fashion. As information is learned about the system, it may be necessary to respecify the conditions that are used in the model. Thus, there may be a certain amount of iteration during the model calibration and risk estimation phases, in which the conditions are respecified to suit the analysis requirements. For this reason, it is advantageous to broadly specify the potential conditions in the risk identification phase so as to assure that the necessary data is collected in the system observation phase.

3.4 Practical Limitations

One of the disadvantages of using a Markov process model is that the number of states tends to grow very quickly with increasing number of variables, and the number of elements in the state transition matrix grows as the square of the number of states. In the case of the safety state model, this growth occurs exponentially. In a system with n conditions, the number of states is approximately 2^n, and the number of elements in the state transition matrix is approximately 2^{2n}. In any computer-based system implementation, there are two fundamental constraints: processor speed and memory size. Both of these constraints will have implications on the maximum practical size of the safety state model.

During the model calibration phase, the state transition matrix is converted to an equivalent risk probability function (as a function of safety state) via matrix inversion. This matrix inversion is the most costly computation required in the method. The number of multiplications required to perform a general matrix inversion is roughly $O(N^3)$, and $N = 2^n + 1$. Therefore, a 15 condition system would require roughly 3×10^{13} multiplication operations, which would require about 300,000 seconds (about 3.5 days) using a dedicated computer with 100 Mflop performance. Each additional condition raises the required processing time by a factor of 8.

In terms of data storage, the largest load is placed by the model calibration phase for storage of the state transition matrix (which contains approximately 2^{2n} elements). Assuming that high-precision floating point numbers are used (each using 8 bytes), this means that a 15 condition model will require about 8 Gbyte of disk memory, if the state transition matrix is to be stored on disk. Each additional condition raises the memory requirement by a factor of 4.

A common technique for comutation reduction in Markov processes is to reduce the number of states. In the case of the safety state model, application of this technique would require identification of states which are not possible—i.e., combinations of conditions which cannot exist. While this is a valid approach in theory, practical realization would be an arduous task with limited promise of return.

A related approach is to consider the state occupancy statistics after the system observation phase. If less than 50 percent of the states have been occupied over a reasonable observation period, it suggests that one or more of the conditions have been inappropriately specified. A re-evaluation of the risk identification phase is required, which may lead to a reduction in the number of conditions specified.

In summary, under the constraints imposed by current conventional computer technologies, safety state model analysis is limited to roughly 15 contributing conditions. Larger models could be accommodated using more exotic technologies, such as super computers or parallel-processing machines. Investigation of such techniques constitutes an interesting topic for future research.

Chapter 4

High-Speed Rail Simulation System

An important component of this research has been the development of a high-speed rail simulation system. This system was developed for the purpose of performing laboratory experiments. The system architecture can be categorized as a *distributed interactive simulation* system. It is interactive in the sense that it is designed to be used by human operators in real-time—a *virtual reality* system for high-speed rail operation. The simulation system is considered a distributed system because it operates on multiple computers, which are interconnected by a local area network. This chapter describes the design and implementation of the simulation system, from a systems perspective.

4.1 Goals and Objectives

The overall objective of the simulation system is to provide a virtual environment for high-speed rail operations, with the intent of using the system for laboratory-based human factors studies. As a result, the overall design requirements of the system were to provide a sufficiently realistic environment for human-in-the-loop operation, such that the task objectives and constraints of the real operational tasks can be met.

In rail operation, there are two primary classes of operating personnel: vehicle

operators (traditionally known as *locomotive engineers*) and traffic controllers (sometimes referred to as *dispatchers*). Vehicle operators perform their duties in the vehicle, generally in the cab and while the vehicle is in motion. Their primary task is speed and position control of the vehicles, which includes detection and reaction to emergency situations. The traffic controllers perform their duties from the wayside, often in a centrally-located control room. The primary task of traffic control operators includes control of the switching points and overall system coordination.

The high-speed rail simulation system has provision for supporting complex rail system operations. The system supports multiple system elements, each implemented on one or more computer system, operating in conjunction with one another and communicating over a local area network. An example system configuration is shown in figure 4-19.

4.2 System Architecture Issues

During the development of the simulation system, considerable effort was spent addressing issues relating to the system architecture. Broadly speaking, the term "system architecture" refers to the organization, interconnection, and functionality of the individual elements which comprise the system. In the case of a human-interactive simulation system, the issues include design of storage representations for the various databases, specifying the mechanisms for intercommunication between the computers which form the system, and identifying appropriate user interfaces.

4.2.1 Road Database Representation

A primary issue in designing a transportation simulation system is selection of a database representation of the environment. Such a database must contain all the pertinent information needed to describe the environment, yet should also be stored in the smallest possible data space.

One characteristic of rail systems is that they cover large distances over paths of very small width. Because these vehicles operate at ground level and along narrow

paths, the view from a rail vehicle is quite limited relative to the distance traveled. This is in stark contrast to the view from an air vehicle, which is often unobstructed over a wide area. In addition, an air vehicle is not constrained to narrow corridors, like a rail vehicle. As a result, air vehicle simulators must be able to reproduce a wide area of visual field to a reasonable resolution. For the purposes of a rail vehicle simulation, however, the topology can be reduced to a network of interconnected paths.

An explicit design objective was to incorporate dynamic elements in the environment. Pertinent to rail operation, these include signal states, switch states, weather conditions, and hazards. In order to achieve this, the design of the database includes variables for these states. In addition, a mechanism is provided for communicating state changes among the simulation elements.

Two distinct database formats were designed to address these issues. One was designed to capture the topology of the road itself (the road database), while the second was designed for representation of the objects that appear along the road (the object database). Although the two are inter-related through reference to the same inertial coordinate system, they are implemented as separate entities because only a subset of the active simulator elements require the information contained in the object database. All of the active simulation elements require the road database.

The road database contains all of the information required to describe the network of guideway paths which comprise the road system. In the abstract form, the network is composed of nodes and links. Each link is a path between two nodes, and each node is a point where links interconnect.

To represent these elements, four data structures are defined: the road segment, the road unit, the connection unit, and the network header. A road database is a binary file containing all of these data structures, organized with the network header first, followed by arrays of the other three data structures (figure 4-1). The data structures are organized as multiply-linked lists, using indices into the respective arrays as the linking mechanism.

The most basic element is a road segment. A road segment is defined as a piece

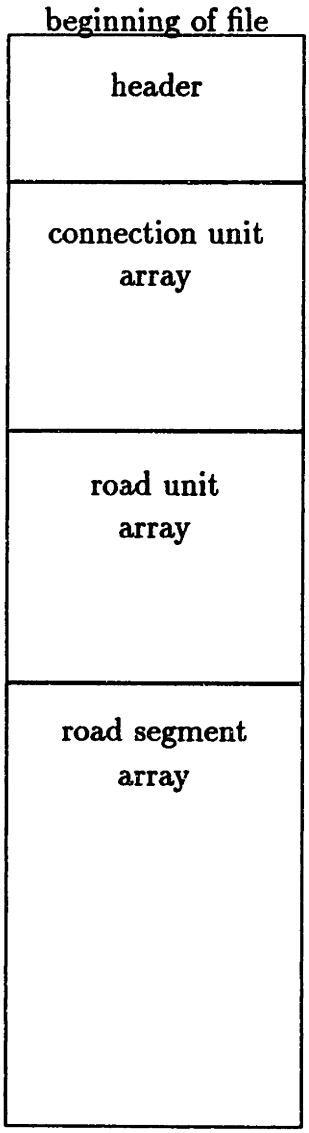

Figure 4-1: Road Database File Organization

```
/* definition of the road segment structure (and fields) */
struct road_element  {           /* one for each individual road piece */
  int road;                      /* index of parent road unit */
  float pitch;                   /* aka grade (radians) */
  float curvature;               /* inverse turning radius (1/meters) */
  float banking;                 /* tilt of roadway (radians) */
  float x_position;              /* distance east of origin (m) */
  float y_position;              /* distance north of origin (m) */
  float elevation;               /* above sea level (meters) */
  float orientation;             /* inertial yaw angle (radians) */
  unsigned int hazard_class;     /* potential hazard classes (bit-encoded) */
  unsigned int hazard_state;     /* hazard state (bit-encoded) */
  float hazard_position;         /* relative to start of segment */
  unsigned int weather_state;    /* weather state (value-encoded) */
  /* lower 8 bits are reserved for temperature information (deg C) */
#define WEATHER_CODE_MASK  0xffffff00
#define WEATHER_TEMP_MASK  0x000000ff
  unsigned int signal_class_up;  /* potential signal states (upstream) */
  unsigned int signal_state_up;  /* signal state (upstream) */
  unsigned int signal_class_dwn; /* potential signal states (downstream) */
  unsigned int signal_state_dwn; /* signal state (downstream) */
#define NO_SIGNAL          0x00000000
#define RAIL_CLEAR         0x00000001
#define RAIL_APP_START     0x00000002
#define RAIL_APP_MED       0x00000004
#define RAIL_APPROACH      0x00000008
#define RAIL_CAUTION       0x00000010
#define RAIL_RESTRICT      0x00000020
#define RAIL_STOP          0x00000040
  unsigned int speed_limit;      /* speed limit (value-encoded, kph) */
  float segment_length;          /* length of this segment */
};
```

Figure 4-2: Road Segment Data Structure

of road which has a fixed start point (represented in X-Y-Z coordinates, as well as heading, grade, and banking), a fixed length, a constant curvature in heading, and a constant curvature in pitch. The data structure for a road segment contains the preceding data, as well as dynamic state information (for signals, hazards, and weather) and a link to the parent road. A C implementation of the road element data structure is shown in figure 4-2.

A road unit is defined as a collection of road segments which collectively form an uninterrupted path. The data structure for the road contains links to the segments which define the start and end of the road, as well as the connection points which are used to connect this road to other roads. These links are specified using indices in the arrays containing the road element and connection unit data structures. The data structure also contains a name for the road. A C implementation of the road unit data structure is shown in figure 4-3.

```
/* definition of the road unit structure */
struct road_unit  {     /* for each roadway */
  char road_id[16];     /* ID string */
  int start_connect;    /* index of start connection */
  int end_connect;      /* index of end connection */
  int start_segment;    /* index of begin segment */
  int end_segment;      /* index of end segment */
};
```

Figure 4-3: Road Unit Data Structure

```
/* definition of the connect unit structure (and fields) */
struct connect_unit  {              /* for each connection */
  char connect_id[16];              /* identifier string */
  unsigned int connect_type;        /* type code */
#define RAIL_ENTRY        0x0001    /*  termination point */
#define RAIL_SWITCH       0x0002    /*  switch */
#define RAIL_STATION      0x0004    /*  station */
#define RAIL_INTERSECT    0x0008    /*  track crossing */
#define HWY_ENTRY         0x0010    /*  end of the road */
#define HWY_RAMP          0x0020    /*  on-ramp/off-ramp */
#define HWY_INTERSECT     0x0080    /*  intersection */
  int connect_state;                /* connection state */
  /* definition of connection state is TBD */
#define CU_MAX   6                  /*  maximum is 6-way intersection (?) */
  int connect_point[CU_MAX];        /* road unit index array */
  /* (organization depends on type of connect) */
};
```

Figure 4-4: Connection Unit Data Structure

A connection point is defined as a point at which two or more roads interconnect. In a rail system, these are used to represent switches, stations, and entry points. The data structure for a connection point, known as a connection unit, contains the links to the roads which connect at this point. It also contains a state variable, used by connection points which have a dynamic state (for example, a switch). Finally, the data structure includes a name of the connection point. As with the road unit, the connection unit references the road unit links via indices into the road unit structure array. A C implementation of the connection unit data structure is shown (figure 4-4).

The highest level data structure is the network header. This data structure contains all of the information which organizes the individual arrays into a system, including the locations and sizes of the individual arrays, as well as the name and type revision of the system. A C implementation of the network header data structure is shown in figure 4-5.

```
/* definition of the header structure */
struct road_network  { /* header, one per system */
  char network_id[16]; /* identification string */
  unsigned int id_code;          /* unique code for database identification */
  unsigned int pathdata_rev;     /* revision of database format used */
  unsigned long num_connects; /* number of connection points */
  unsigned long num_roads; /* number of road units */
  unsigned long num_segments;    /* total number of segments in database */
  struct connect_unit *connect_list;  /* ptr to start of connect unit array */
  struct road_unit *road_list;        /* ptr to start of road unit array */
  struct road_element *segment_list;  /* ptr to start of segment array */
  unsigned int checksum;         /* for stored database integrity check */
};
```

Figure 4-5: Network Header Data Structure

A graphical representation of the linked list interconnections is shown in figure 4-6. This figure illustrates the interconnections between different elements in the database. Each connection unit has pointers to the roads that link there. In addition, the road units include pointers to the connection units to which they are attached. For example, the figure shows that connection i links road j to road k, and both roads j and k have pointers to connection i. Each connection unit may link from one to six roads. Each road must point to two connection units. In a similar fashion, each road has pointers to two road segments. There may be any number of road segments which comprise the road—however, they are designed to be contiguous in the road segment array. As a result, the road unit needs only to point to the first and last segments included in the road. Each road segment has a pointer to its parent road unit.

The data are stored in a binary disk file. Creation and modification of these binary files is accomplished with an off-line tool called `Pathnet` (described in section 4.4.1).

The object database is used for representation of the visual environment, and is independent of the road environment database. That is, the two database entities are contained in separate disk files and there are no data references or links between them. However, they are related in that they share the same "physical" space. The visual environment database contains all of the information necessary to specify the objects in the visual field. In effect, the visual environment database is a comprehensive list of objects that exist in the visual field, such as trees, bridges, buildings, and such.

Like the road environment database, the object database is implemented as a

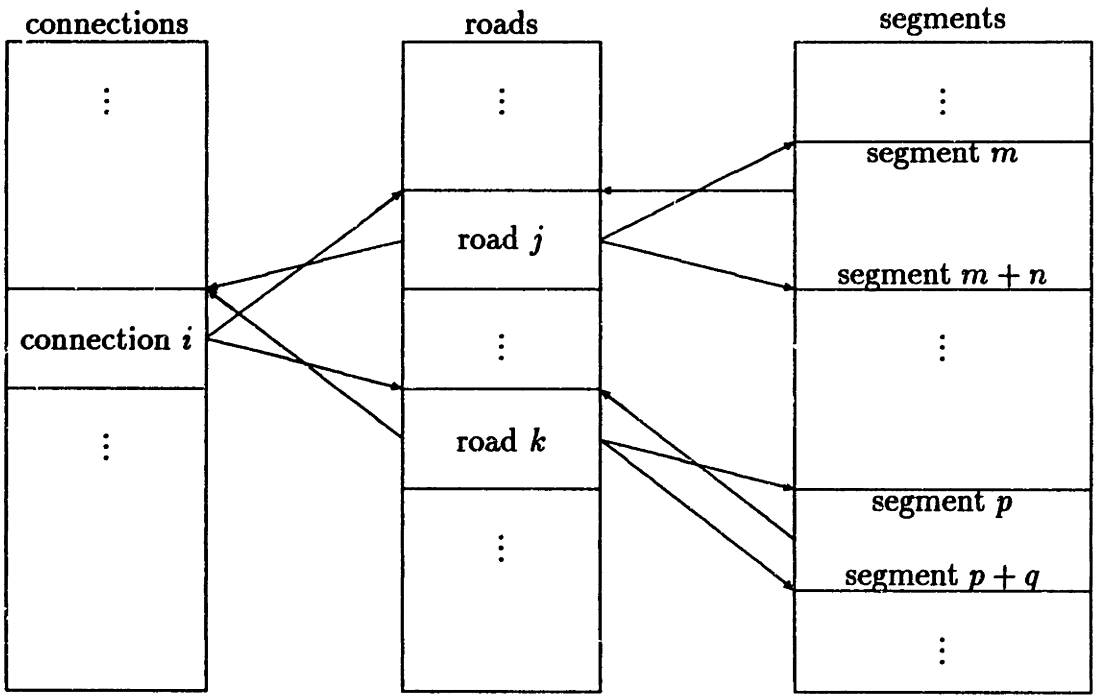

Figure 4-6: Linked List Interconnections in Road Database File

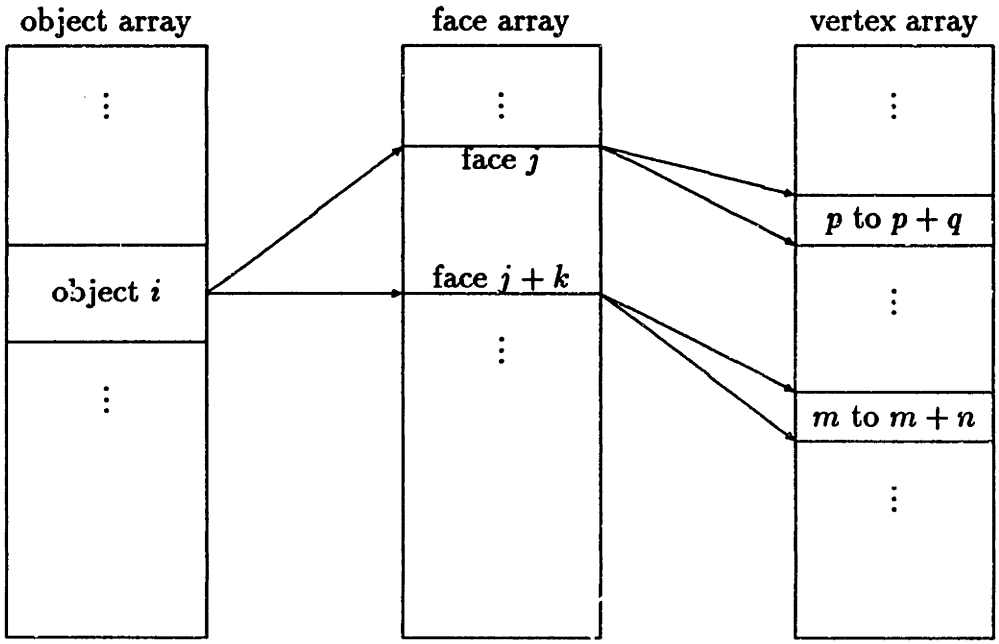

Figure 4-7: Object Database Hierarchy

set of arrays of data structures, which contain links between different levels of data structures. The object database is different from the road environment database in that it is more hierarchical (figure 4-7).

At the highest level is the header structure, containing references to the individual arrays. The next level is the list of objects. An object has the properties of object type, size, position in inertial space, and orientation in inertial space. The data structure also knows the number of faces that comprise the object and has reference to the list of these faces.

A face is defined as a single polygon that is part of the object. A face has the property of color, and the data structure includes the number of vertices that form the face and reference to that list of vertices. A vertex is a point in inertial space, and all of the vertices that form the objects in the database are contained in a single list. The data structure for each vertex contains specification of the three-dimensional point that locates the vertex.

The objects defined by the visual environment database are static. That is, they

```
/* define the database header for the entire terrain */
struct terrain_hdr  {
  char network_id[16];  /* should match that of path database file */
  unsigned int id_code; /* should match that of path database file */
  unsigned int otwdata_rev;  /* revision of OTW database format */
  float rcolor;       /* color coding for base terrain */
  float gcolor;
  float bcolor;
  int x_offset;       /* smallest X value */
  int y_offset;       /* smallest Y value */
  int x_cnt;          /* number of blocks in the X direction */
  int y_cnt;          /* number of blocks in the Y direction */
  int subblk_cnt;     /* number of sub-blocks in a block */
  int subblk_sz;      /* edge dimension of sub-block (m) */
  float **terr_zlist; /* ptr to terrain elevation row list (2D matrix) */
  float *terr_zdata;  /* ptr to terrain elevation data */
  int olist_cnt;      /* number of objects in the database */
  struct obj_header *obj_list;
  int flist_cnt;      /* number of object faces in the database */
  struct obj_face *face_list;
  int vlist_cnt;      /* number of face vertices in the database */
  struct vpoint *vrtx_list;
  int spare1, spare2, spare3;
};
```

Figure 4-8: Object Database, Header Data Structure

```
/* define the representation of an object */
struct obj_header  {
  int obj_id;     /* coded type of object (determines base dimensions) */
  float xlate_x;  /* location of object center */
  float xlate_y;
  float xlate_z;
  float rotate;   /* horizontal planar rotation of object */
  float hscale;   /* height scale factor */
  float wscale;   /* width scale factor */
  int num_faces;  /* count of face polygons */
  int face1_indx;
  struct obj_face *face1_optr;
  int active;     /* use for lighting cues */
};
```

Figure 4-9: Object Database, Object Data Structure

```
/* define the representation of an object face polygon */
struct obj_face  {
  float rcolor;       /* color coding */
  float gcolor;
  float bcolor;
  int vertex_cnt;     /* number of vertices in the face */
  int vertex1_indx;
  struct vpoint *vertex1_vptr;
  float oriented;     /* outward direction of face (wrt E) (horizontal) */
};
```

Figure 4-10: Object Database, Face Data Structure

```
/* define the representation of a polygon vertex point */
struct vpoint  {
  float y_pt;   /* north of origin */
  float z_pt;   /* elevation above origin */
  float x_pt;   /* east of origin */
};
```

Figure 4-11: Object Database, Vertex Data Structure

have no dynamic state, and their parameters cannot change in time. A graphical representation of the linked list interconnections is shown in figure 4-7. The data is stored in a binary disk file. Creation and modification of these binary files is accomplished using `Pathnet` (section 4.4.1).

4.2.2 Network Interconnection

The individual simulation elements are programs that run on high-performance graphics workstations. The workstations are interconnected via a local area network. The interconnection protocol used for these connections is generically TCP/IP. Specifically, the UDP (User Datagram Protocol) protocol is used, which is layered above IP (Internet Protocol).

The selection of this protocol is a significant element of the system architecture. The alternative protocol, TCP (Transmission Control Protocol), is connection-oriented and has guaranteed reliability, while UDP is a connectionless datagram protocol. In other words, when using TCP, each data packet sent from one machine to another is guaranteed to be delivered, regardless of the time required. If a network failure prevents a packet from reaching its destination, it will be retransmitted until successfully received. Retransmission could potentially require several seconds. The UDP protocol, on the other hand, is a datagram-based protocol, without guaranteed reliability. This means that once a packet has been sent out on the network, there are no mechanisms in place to check for receipt or to retransmit the packet if it does not reach its destination.

In the case of a distributed interactive simulation system, the information being broadcast over the network is real-time state data, which is more sensitive to transmis-

sion delay than reliability. The vehicle state data, which includes current vehicle position and speed, is time-sensitive and is transmitted fairly frequently (approximately once every half-second). Under these conditions, a connectionless datagram protocol is more apppropriate, since the reliability mechanisms of a connection-oriented protocol would result in unacceptable delay and queue overflow in the event of communication failure.

4.2.3 OTW View

The vehicle simulation includes an out-the-window (OTW) viewport for the vehicle operator. Through this view, the operator can look out into the physical environment and see objects that exist in that environment. These objects are rendered as true three-dimensional objects, and maintain true perspective as the viewer moves through the environment.

True three-dimensional graphics require an immense amount of computational power and are best implemented on specialized graphics workstations. The platform used for this implementation is an Indigo-2 Extreme workstation, manufactured by Silicon Graphics Inc. (SGI). This machine features computational performance of approximately 100 Megaflops and graphics performance of approximately 150,000 polygons per second.

Despite the specialized hardware, there remain limitations with synthesizing the OTW view. It was determined by the project team that, to maintain a jitter-free view, the frame update rate should be 20 frames per second or greater. In order to maintain this rate, and to enhance the perception of motion, several steps were taken in the design of the OTW view.

A night view was selected to reduce the number of required background objects. The view of objects up the track diminishes in light intensity, giving the perception that the objects are lighted by the vehicle headlights. To support that perception, the intensity of the block signal lights does not diminish with distance, since those signals are sources of light emission instead of light reflectors. The rails and roadbed are shown as fading into the distance. The roadbed includes simple rock-like objects

to provide sense of motion, especially at lower vehicle speeds. In addition, the rails "shake" to enhance the sense of vehicle motion. At higher speeds, the amplitude of the "shaking" increases. The static objects in the environment appear as wire-frame images. Kilometer posts appear as reflective black-on-white signs on the right side of the road, and block signals appear overhead, with lighted signals and reflective identification numerals above. Grade crossings appear as grey roadways, and the highway vehicles are solid blue cars. Stations appear as wireframe shapes, with walls colored in purple and blue. When the front edge of the vehicle is within the boundaries of the station, a head-up display appears on the windscreen, to be used by the operator as an aid for vehicle positioning in the station.

To limit the drawing requirements, the objects along the wayside are drawn as wire-frame objects. That is, only the outlines of the objects are drawn, and it is possible to see through the objects. In addition, objects are constructed from a small collection of fundamental shapes. This collection includes a cube, a pyramid, and an octagonal column. More complex objects are constructed from these basic shapes, such as a building (figure 4-12) or a pedestrian bridge (figure 4-13).

Certain objects in the view are filled as solid shapes. These include the rails, the roadbed, the grade crossing roadway, and cars going across the grade crossing. Also painted as solids are the block signs and the kilometer posts (figure 4-14), as they provide information via numerals on the signs.

In summary, the design of the OTW view represents a compromise between the desire to provide a realistic visual environment and the available computational resources. The resultant implementation provides a limited fidelity view with a medium fidelity frame rate. Despite the limitations, the view provides sufficient quality and task cues to allow for immersion of the human subject into the task of vehicle operation.

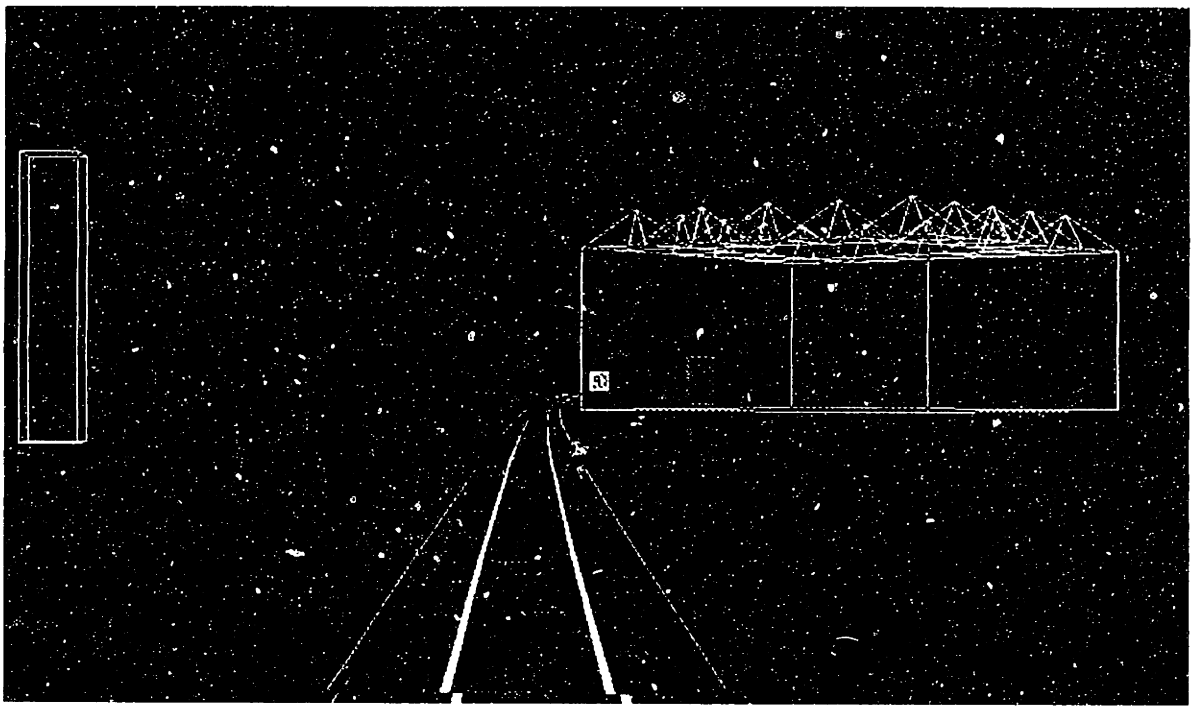

Figure 4-12: Example View of a Building

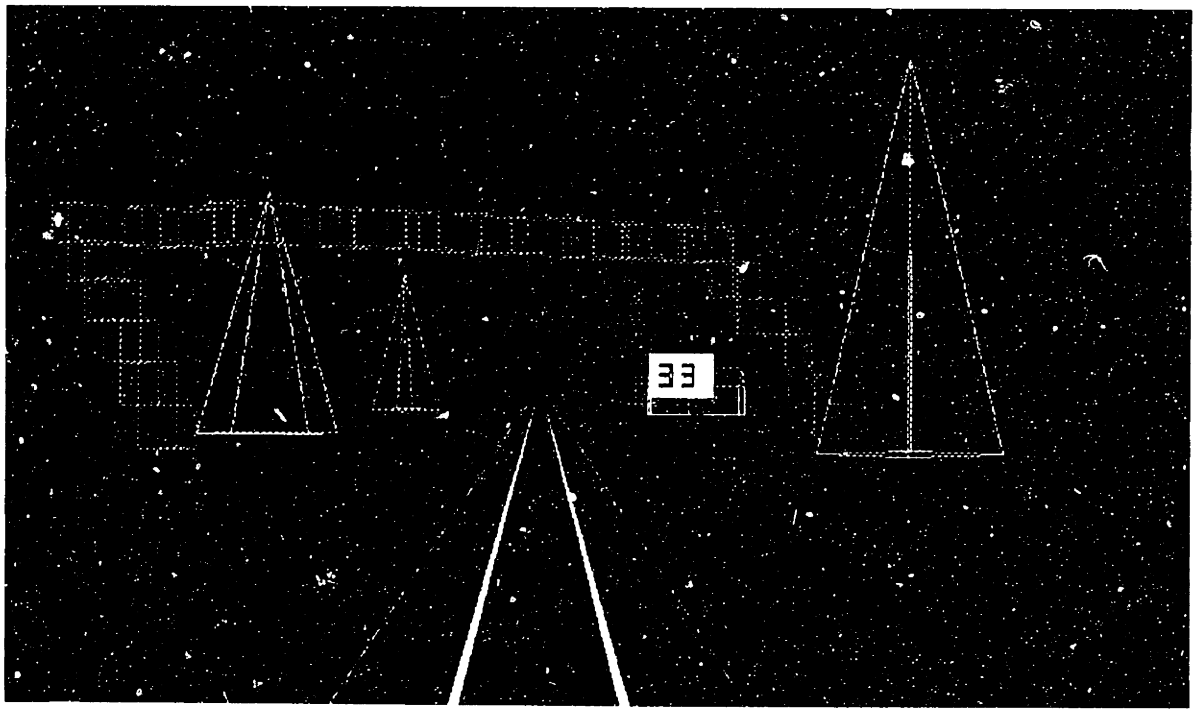

Figure 4-13: Pedestrian Bridge

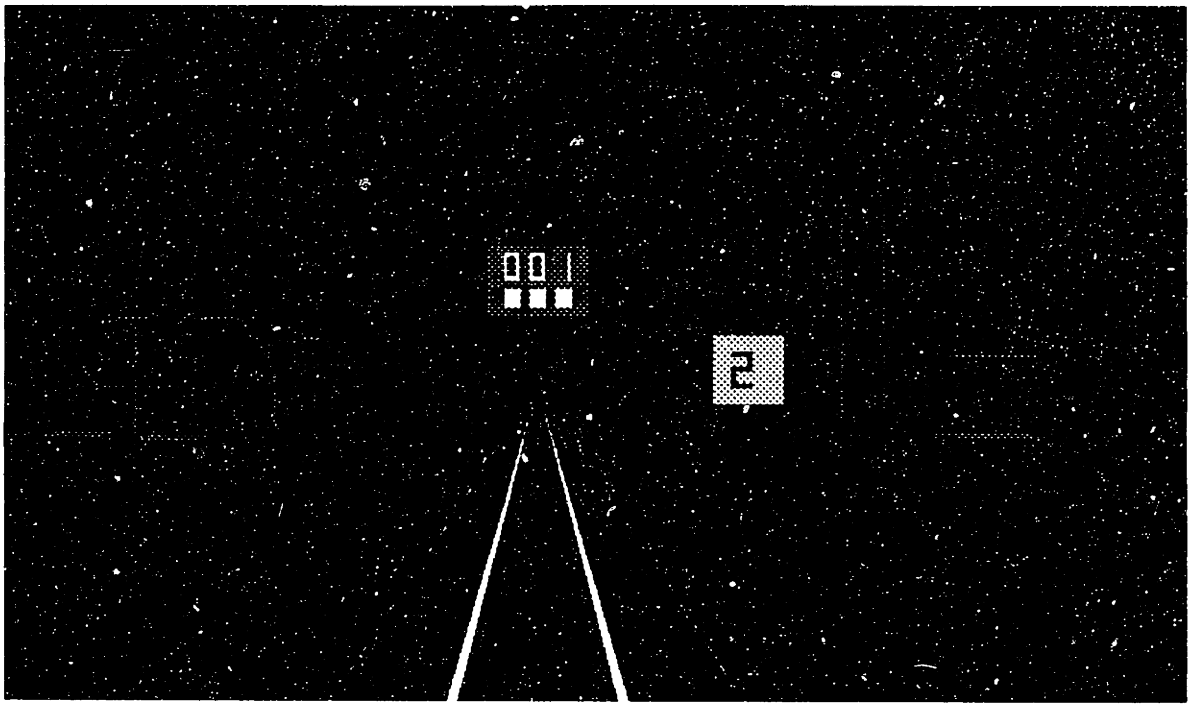

Figure 4-14: Block Signals and Kilometer Posts

4.3 Active Simulation Elements

In the high-speed rail simulation system, *active simulation* elements are software modules that execute when the simulator is in operation. The current implementation of the system has two distinct types of active simulation elements: *vehicle simulations* and *CTC simulations*. A vehicle simulation provides a human operator with a user interface for a vehicle which operates in the virtual environment. The user interface for a vehicle generally includes an instrument panel and an out-the-window view. In order to maintain high-quality graphics resolution, vehicle simulations in the high-speed rail simulator require two workstations for operation.

A CTC simulation creates the user interface used by a human CTC operator. The user interface for a CTC operator generally includes a map or layout of the road system. Through this interface, a CTC operator monitors the rail system and controls the state of switches and signals as required for system operation.

The system architecture has been designed to support multiple vehicles coexisting

in a common virtual environment, as well as multiple CTC interfaces. To date, the system has been operated with one each of the vehicle and CTC simulation elements.

The following subsections give a brief overview of the active simulation elements. Since these programs total approximately 50,000 lines of C source code, a detailed description of the internal operation of each is beyond the scope of this document. Instead, the intent is to highlight the major features of each. Figures are provided where appropriate, to convey a sense of perspective about the displays developed for the simulator. In all cases, the actual display views are color, and most feature light-colored objects on a dark background. However, to simplify the publication process, the displays are reproduced here in monochrome and in reverse video (dark objects on a light background). Unless otherwise specified, the software described was designed and developed by the author.

4.3.1 CTC Simulation

The CTC simulation element provides an interface from which a human CTC operator controls a rail system. The primary display is a two-dimensional plan display of the rail system. The display is geometrically accurate, providing the operator with an scaled view of the road curves and interconnections. An example of the CTC display is shown in figure 4-15.

The rail system supports a block signalling paradigm, as described in appendix A. Individual block segments are identified with white marks at the block boundaries. Each block segment is color-coded according to signal state, indicating the most restrictive signal level in that block. At the highest level of resolution (zoom), each block segment is shown with two lines, representing the signal indications in both directions.

The stations are depicted as magenta rectangles and are identified by name alongside the icon. The switches are identified with magenta circles. At the highest level of resolution, a text display of detailed block information, including signal status, is available (figure 4-16).

The computer mouse is the primary system input device for the CTC operator.

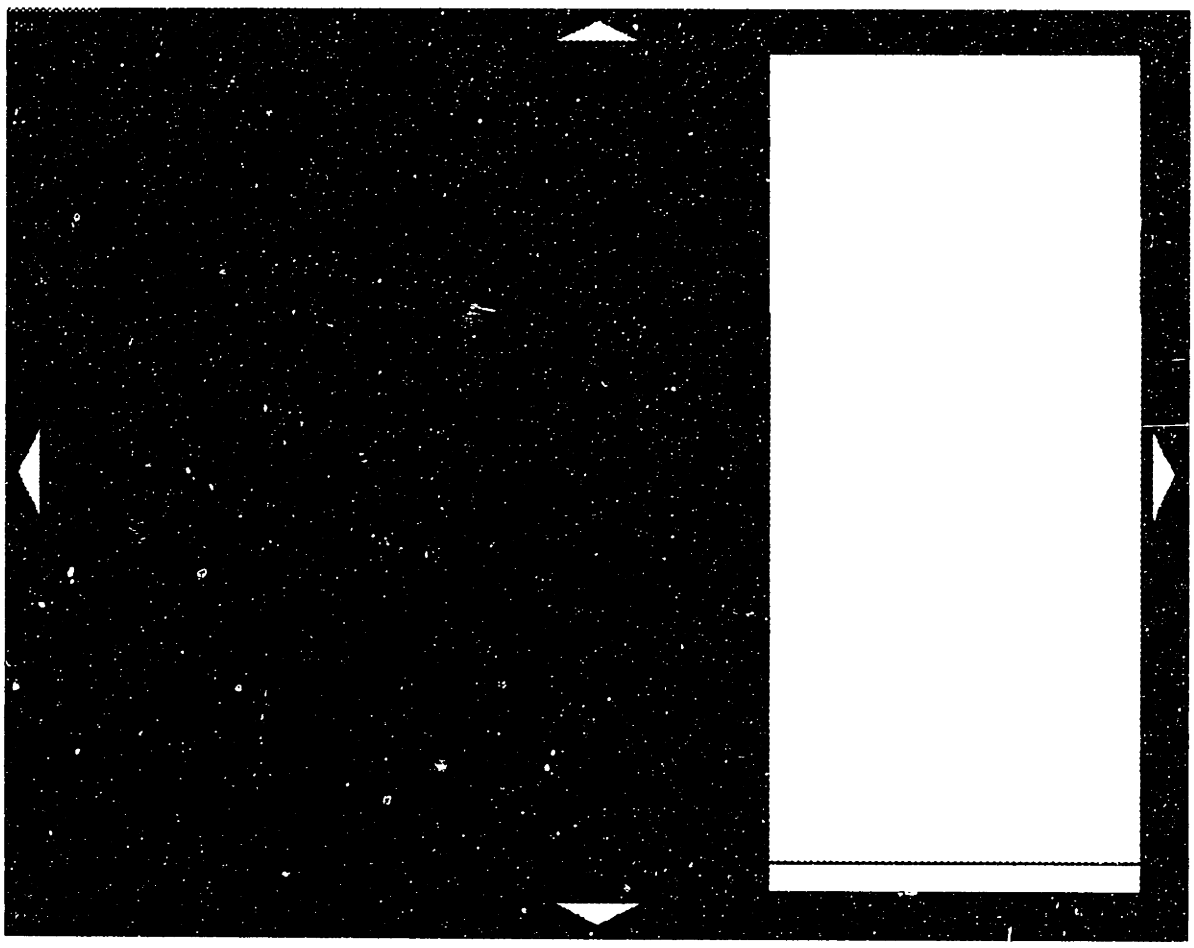

Figure 4-15: CTC Display

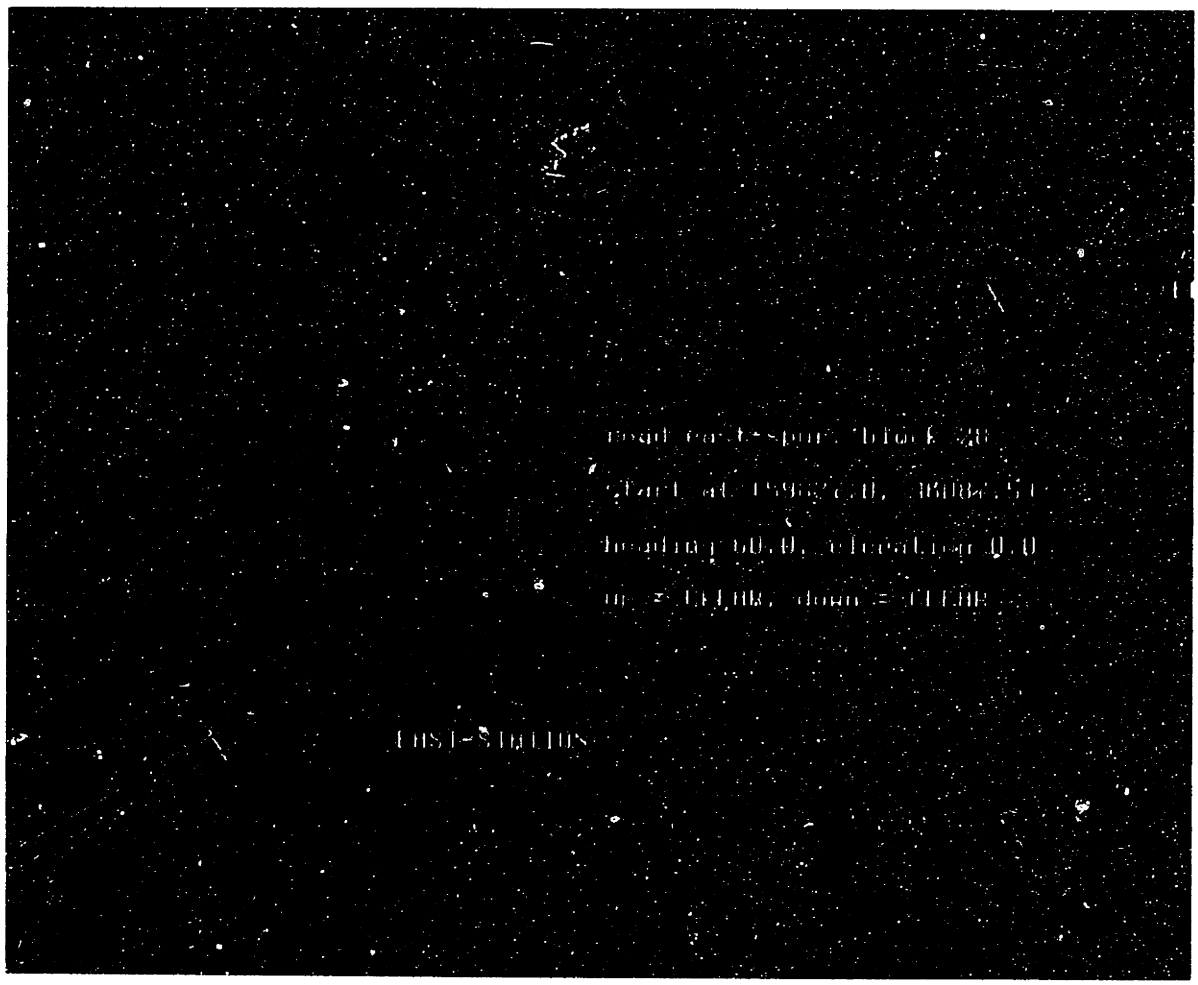

Figure 4-16: Rail Station Icons

Using this device, the operator is able to zoom up and down (i.e., decrease or increase the field of view), as well as pan (i.e., move left or right) and tilt (i.e., move up or down). The mouse is also used to change the state of a switch. Through the use of selected function keys on the keyboard, the operator can alter the state of a signal, either setting or clearing a signal in either direction.

In general, block signals are controlled automatically by the CTC simulation. The CTC simulation element implements the function of an automatic switch system, sensing the position of vehicles in the system and setting the signals accordingly. All signal and switch state changes are initiated through or by the CTC simulation, and it is the responsibility of the CTC simulation to apprise all active vehicle simulations of the changes in signal and switch states as they occur. Thus, the CTC simulation acts as the agent for the dynamic environment.

To date, two versions of the CTC simulation have been implemented. These differ only in the implementation of the signal control system—one version is set up to provide a seven-aspect signal system (in support of the display-aiding experiment [1][2]), while the other provides a five-aspect signaling system (for the control automation experiment described in section 5.2).

4.3.2 Vehicle Simulation—Control Automation Version

To suit the needs of the control automation experiment (described in detail in section 5.2), a second train simulation has been implemented. The advanced displays, as described in section 4.3.3, were not used in this experiment; as a result, the instrument panel was redesigned to better utilize the available space for the basic instrument set. In addition, the instruments themselves were redesigned to more closely approximate instruments currently in use. The primary input devices are the throttle lever and the computer keyboard. The output device is the computer screen.

The overall functional operation of this train simulation is similar to that of the train described in section 4.3.3. The vehicle dynamics approximate those of an actual vehicle (described in detail in appendix E), and the same safety systems (alerter and ATP) are included.

The instrument panel display, shown in figure A-3, provides the basic instruments for vehicle operation. At the center is a large round speedometer, with a red pointer to indicate the current speed. A smaller yellow pointer is located behind the red pointer, and is used in conjunction with the automation systems. The in-cab signaling system is located above the speedometer, and contains indications for both the current block (the signal most recently passed) and the next block (the upcoming signal). There are four small round gauges for system state monitoring, including two for brake pressure, and one each for bearing temperature and trolley voltage. The vertical LED bars are ammeters to indicate the electrical current applied to each of the four traction motors. There are status and warning indicator lights for emergency brake status, ATP warning, alerter warning, and door status, as well as indicator lights for the control automation modes and a digital clock.

Three control automation modes have been implemented: cruise control, programmed stop, and autopilot. The cruise control system operates much like an automotive cruise control. The operator selects a desired cruise speed, and the system regulates vehicle speed to the selected *set speed*, applying throttle as required. The operator also has provision for making fine adjustments of the set speed, via keyboard button inputs. The control loop for the cruise control is based on a proportional-integral (PI) control loop [34] for speed regulation, using thrust only.

The programmed stop system is designed to halt the vehicle at designated stopping points. If proper conditions are met, the vehicle is stopped at the end of the current block. Since stations are always located at a block junction, the programmed stop system may be used for station stopping. The programmed stop system utilizes a look-up table to modulate the brakes such that the vehicle stops at the appropriate point. The lookup table is based on the stopping curves for the vehicle. Several safeguards have been implemented to guard against high-risk situations. The first is an overspeed protector—the programmed stop system cannot be engaged if the vehicle speed is above 80 km/hr. Another safety system guards against late application of programmed stop—if the speed at the position of application is too high for the vehicle to be stopped using the service brakes, the system detects a fault. The emergency

brake is applied in response to either fault scenario.

The autopilot system utilizes a pre-programmed speed trajectory as a function of vehicle position, and uses both the traction motors and the brake system to regulate that speed. The autopilot utilizes a PI controller while underway, and invokes the programmed stop system when approaching a station. In effect, the pre-programmed speed trajectory takes the place of the operator command to the cruise control system. When approaching station stops, the autopilot invokes the programmed stop system automatically. Using the autopilot, normal speed control of the vehicle becomes a "set and forget" operation—once engaged, no further input is required until after the vehicle has arrived in the station.

The train simulation described in this section can be configured to display either the instrument panel or the OTW view. When configured as the instrument panel, the OTW server (section 4.3.4) may be optionally invoked on a separate machine to provide an OTW view. Similarly, when the train simulation is displaying the OTW view, the dashboard server (section 4.3.5) may be optionally invoked on a separate machine to provide an instrument panel.

During operation of the vehicle simulator, vehicle state data are written to a disk file. Each data record includes a time stamp (in milliseconds), an event code (text), the approximate elapsed distance (in km), the exact current position (in terms of the current block and the position in that block), and the vehicle speed (in km/hr). For certain events, there is an optional field, containing relevant information. The disk file format is ASCII text. The data record type is equivalent to the event code contained in the record.

A vehicle speed data record, indicating the vehicle position and speed, is written every 600 milliseconds. There is an additional field in this record type, containing a number between 1.0 and -1.0, indicating the position of the combined control lever. When the control lever is being moved by the operator, the rate of these records is increased to capture all of the operator input.

Data records are also written when the user depresses any of the control keys. These records, in conjunction with the vehicle state records, are used to identify the

operator actions. Other data records are written when the vehicle changes operating state (e.g., when an automation mode is successfully engaged). Note that there is a distinction made between an operator command that may result in vehicle state change, and the actual change of vehicle state that results from that command. By separating these, it is possible to identify erroneous operator input, as well as vehicle state changes that occurred independently of operator command.

4.3.3 Vehicle Simulation—Display-Aided Version

For the purposes of the display aiding experiment, a train simulation was implemented which featured advanced display technology. This simulation was designed and implemented by Shumei Yin Askey [1][2]. At the core of the simulation is a real-time process loop which includes vehicle dynamics and user interface I/O processing. The primary user input devices are a throttle lever, the computer keyboard, and the computer mouse. The throttle lever is mounted to the table near the computer display. The sole output device is the computer display.

The vehicle operation simulates the dynamics and interface of an actual rail vehicle. Accurate vehicle dynamics, as well as realistic safety systems, are included in the simulation. Safety systems include an *alerter system* and an *automatic train protection* (ATP) system. The alerter system, also known as a *dead man* system, is used to ensure that the operator remains active in system operation. The ATP system automatically enforces speed limit compliance.

A significant feature of this simulation is the provision for advanced display technologies, known as *display aids*. These modes are enabled via command line options when execution of the program is initiated. In the basic mode, a fundamental set of instruments is provided for the operator, including speedometer, odometer, vehicle system instruments (such as brake pressure, trolley voltage, and door status), in-cab signal indicator, system status and warning lights (such as emergency brake status, alerter warning, and ATP warning), and throttle position indicator.

The first level of display aiding includes a preview display. The purpose of the preview display is to provide the operator with information about the state of the

roadway beyond that visible through the out-the-window view. The additional information includes position of block boundaries and kilometer boundaries, civil and signal speed limits, multi-block signal preview, and position of stations, among others. The raneg of the preview distance may be adjusted by the operator, allowing trade-off between range and resolution.

The next level of display aiding includes a predictor display. The predictor display adds three curves that project from the current speed-position indicator. The topmost line provides a prediction of the speed trajectory, assuming that the throttle is maintained at the current level. Modulating the throttle will cause this prediction to change. The middle line indicates the stopping speed trajectory when full service braking is used. The bottom line indicates a similar trajectory for the emergency brake. Both of the braking trajectories are a function of speed. The predictor display allows an operator to improve the strategic planning of throttle and brake application to suit the conditions shown in the preview display.

The highest level of aiding includes the advisor display. The advisor display shows a pre-computed speed trajectory which has been optimized for various higher-order performance factors, such as fuel consumption and passenger comfort. This speed trajectory is overlaid on the preview display. The operator attempts to manipulate the throttle and brakes so that the the predictor display correlates with the advised speed trajectory.

The train simulation has the option of providing an OTW viewport in the upper third of the display area. In addition, the simulation can be configured to drive an external OTW server (section 4.3.4).

The train simulation provides a recording system for storing state vehicle data which occurs during system operation. The data are stored in the form of ASCII data records, each of which contains a time stamp, an event code, the position and speed of the vehicle, and additional information as appropriate.

Illustrations of the display-aided instrument panels, along with a description of experiments which evaluate the use of these driver aids, may be found in [1]. A brief summary of the vehicle dynamics models are included in appendix E; a more complete

description is included in [1].

4.3.4 Vehicle Simulation—OTW Server

The OTW server is an independent module used for the secondary display of the OTW viewport. The function of the OTW server is to display the OTW view only, as a slave to a primary train simulation.

Communication between the primary train simulation and the OTW server is accomplished via the local area network (LAN). The primary simulation element sends state information to the OTW server, which then updates its internal estimation of the vehicle state. To maintain smooth graphics output in the absence of new state information, the OTW server performs position estimation using a dead reckoning algorithm, based on the most recent state information received from the train simulation.

4.3.5 Vehicle Simulation—Dashboard Server

Similar to the OTW server, the dashboard server provides a dashboard display on a secondary machine. The dashboard format used is identical to the format used in the control automation train simulation (figure A-3). Like the OTW server, the dashboard server is in communication with the primary process via the local area network. In this case, the data transmitted from the primary process to the server is only the data required for the instrument panel display.

4.4 Support Software

In addition to the active simulation elements outlined in section 4.3, several programs have been implemented to support the overall simulation system. One of the programs is used for creating and modifying the road and object databases used by the train simulation system. The other programs are data analysis programs, used for post-processing the data obtained from the simulation system.

4.4.1 Pathnet

The off-line program `Pathnet` is an important support element of the simulation system. As discussed in section 4.2, the virtual environment is contained in two types of files, known as the road environment database and the object database. The database files are used by the active simulation elements during operation. `Pathnet` is a tool used for the creation and modification of these databases.

`Pathnet` is an interactive tool. After invoking `Pathnet` at the Unix shell prompt, an environment database file is opened or created. If a new database is created, the user begins with specification of an initial connection point. From that first connection point, the road system is built by specifying road paths and the interconnections between them. Existing roads may be modified or deleted. The database may be saved at any point. Modes are provided to allow the user to view the road system in a geometrically-correct graphics window and to inspect the data manually.

`Pathnet` is also used for creation and modification of object databases. Objects are created through specification of object type, size, position, and orientation in the environment. `Pathnet` automatically generates the face and vertex data structures, based on the information provided.

Both the road environment and object databases are stored in binary format. The use of a binary storage format results in disk files which are more compact. Use of a binary data format also reduces the likelihood that the files will be tampered with.

4.4.2 Data Analysis Tools

Additional programs have been implemented to provide support for the various experiments that are performed. This section provides a brief introduction to the programs used in support of the control automation experiment.

The program `transform` was developed to transform the raw data, generated by the train simulation, into the failure response times required by the control automation experiment. This program reads through an entire raw data file, identifies the failures that have occurred during the session, and identifies the operator response to

those failures. The program generates a new output file, containing the summary of the failure responses.

The program `bonus_process` is used to calculate the bonus point performance of the subjects in the control automation experiment. As is the case with `transform`, `bonus_process` reads through an entire raw data file, and selects those data points required for calculation of the bonus score. The pertinent data include response to failures, as well as station stopping performance and schedule maintenance. The output of the program is a text-based summary which is directed to the terminal output (computer screen). This output may be piped to a disk storage file.

The program `ss_process` is used to convert a raw data file into a safety state trajectory. Raw data files are used as input, and the output is a text-based summary of the safety state values as a function of time. Like `bonus_process`, the output of `ss_process` is directed to the terminal output, and may be piped to a storage file.

The program `mtbf_calc` is used to convert a collection of safety state trajectories into a risk probability function. The input to this program is a set of safety state trajectories generated by `ss_process`, and the output is a set of risk trajectories that result from transforming the safety state trajectories with the risk function. These risk trajectories are stored as disk files.

The program `risk_stats` is used to perform statistical analysis of the risk trajectory output from `mtbf_calc`. For each of the risk trajectory files, a corresponding statistics file is generated which summarizes the state occupancy statistics for that trajectory, as well as the average risk probability over the trajectory.

4.5 Software Engineering Issues

In the development of any substantial body of software like the high-speed rail simulation system, issues of software engineering must be addressed to ensure the ongoing viability of the project. These issues are especially critical if the project is to be developed and sustained by several software engineers. The following sections outline some of the software engineering issues addressed during implementation of the

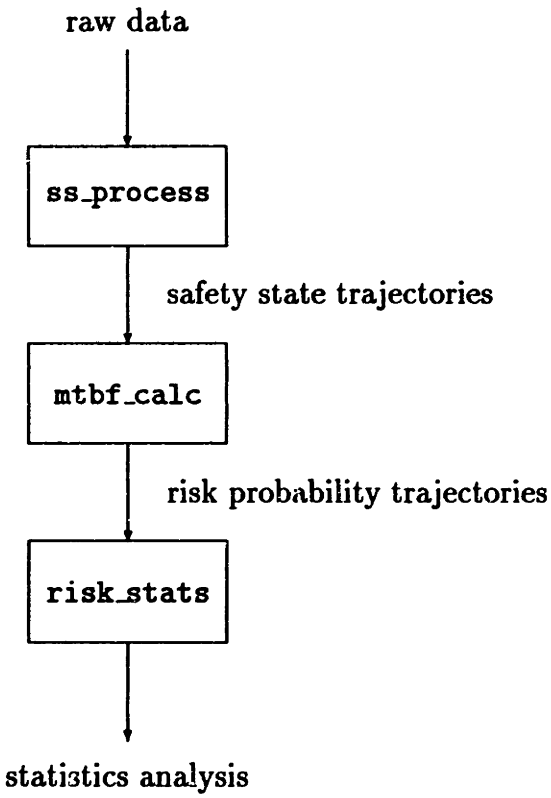

Figure 4-17: Flow Chart of Data Post-Processing Procedure

high-speed rail simulation system.

4.5.1 Shared Libraries

In an effort to modularize the code to the greatest degree possible, the concept of shared libraries is used wherever possible. This concept forms a basis for sharing code among separate elements in the system.

There are two distinct scenarios where libraries are especially prudent. In the first, there are a number of different pieces of software that need one or more related functions. Development of a single library for the related functions allows several modules to share the code, lessening the development load and unifying the interface.

The second scenario occurs in the case where two separate processes need to communicate using a common protocol. By incorporating all of the functions related to that protocol in a single library, it is easier to ensure consistency throughout the function set.

The high-speed rail simulation system has four shared libraries: the database interface library (`libdb`), the network interface library (`libnet`), the OTW interface library (`libotw`), and the schedule library (`libsched`). The database interface library contains those functions used for loading and interpreting a road database (section 4.2). The network interface library contains functions used for interprocess communication over the local area network. The OTW interface library contains functions used for displaying an OTW viewport. The schedule library contains functions for loading and interpreting a schedule database.

4.5.2 Development File Hierarchy

It was recognized early in the software development phase that the project would require a substantial amount of software development. In order to partition the project into manageable chunks, a development file tree was created.

The root of the tree exists at a level separate from and parallel to the users personal directories. Located at `/usr/projects/rail-sim`, the root directory contains

distinct directories for vehicle simulation code (`vehicles`), CTC simulation code (`ctc`), shared libraries (`lib`), database information (`database`), experiment-specific scripts (`exp`), and data recording (`data`).

In the `vehicles` directory, there are four subdirectories: `veh-1`, `veh-2`, `veh-otw`, and `veh-dash`. Each of the directories contains the source code modules for an active simulation element program (see section 4.3). In addition, this directory includes subdirectories for common code (`common`), local libraries (`lib`), and local include files (`include`).

Similarly, the `ctc` directory contains two subdirectories (`ctc-1` and `ctc-2`) for source code, as well as the subdirectories `common`, `lib`, and `include`.

The `lib` directory contains four subdirectories for source code: `libdb` for the database interface library, `libnet` for the network interface library, `libotw` for the OTW interface library, and `libsched` for the schedule library (developed by Shumei Askey).

The `exp` directory contains two subdirectories, one for the display aiding experiment (`exp-disp`) and the other for the control automation experiment (`exp-auto`). These subdirectories also appear under the `data` directory, to allow segregation of the data obtained from the two experiments.

A graphical depiction of the file system hierarchy is shown in figure 4-18. The file system hierarchy is replicated on all machines.

4.5.3 Software Build Engineering

The primary tool used to build the software (i.e., transform the source code files into executable code) is `make`, which is a standard Unix development tool. The build specifications for each of the executable modules in the system is contained in a file named `Makefile`, located in the local source code directory. In addition, there is a master `Makefile`, located in the root directory (`/usr/projects/rail-sim`), which will rebuild the entire system from scratch.

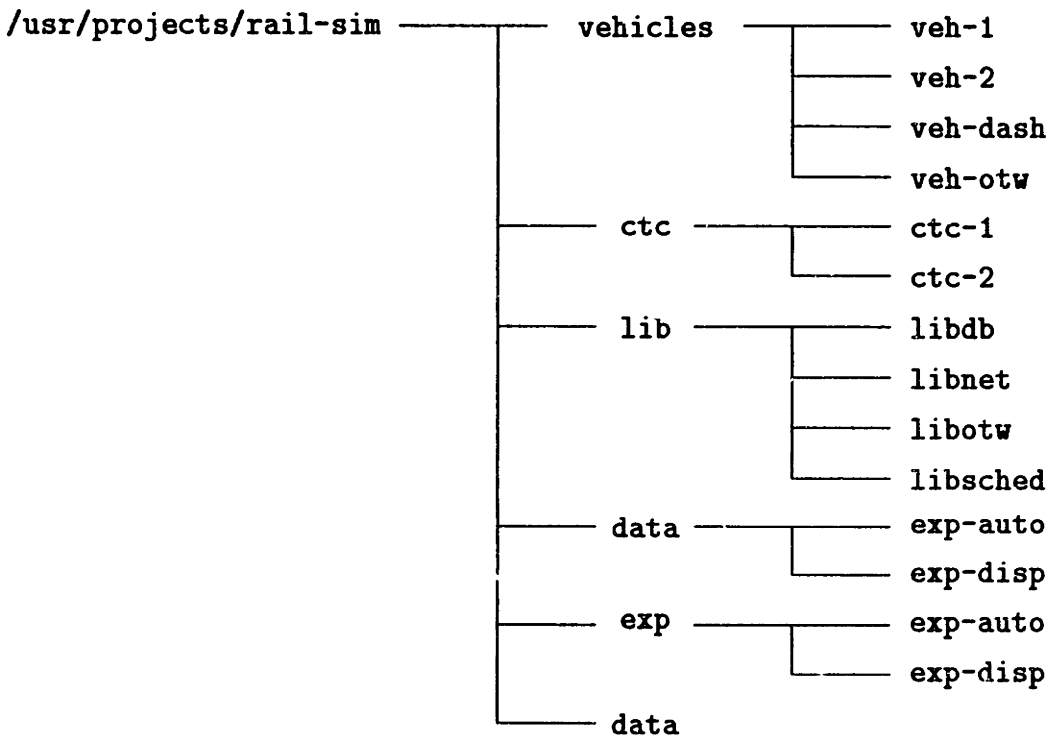

Figure 4-18: Development File System Hierarchy

4.5.4 Revision Control

The primary tool used to support software revision control is RCS, which is a publicly-available and widely used tool. It is supplied as part of the SGI development environment, along with the compiler, linker, and source code debugger. RCS stores multiple versions of a source code file in aseparate archive file. Each time the source code is changed and "checked-in" to the archive file, the differences between the new and old versions are recorded, and comments are inserted to provide a "paper trail." The comments include the date and time of revision, as well as the username of the person that was responsible for the changes.

At any point in time, any of the revisions that are stored in the archive file may be retrieved without risk to any of the other versions. Thus, it is possible (and easy, in fact) to revert back to an earlier version of the software, without losing any of the subsequent changes.

A majority of the source code modules also include "markers" for storing RCS header information. These "markers" allow RCS data to be included in the executable module. It is then possible to identify the source code that comprise an executable module, even if the executable module has been separated from the source code directories.

4.6 System Configurations for Experiments

To date, two experiments have been conducted using the high-speed rail simulation system. The first was an exploration into the effects of display aiding on operator performance. The second was focused on identifying the effects of control automation on operator performance. The configuration of the simulation system was tailored in each case to the objectives of the experiment.

For the display aiding experiment, the CTC simulation used was the version that supported the seven-aspect signal system. This element was run on a Personal Iris machine. The train simulation was run using two machines. The Indigo-2 was used as the primary train simulation machine, which executed the display-aided train sim-

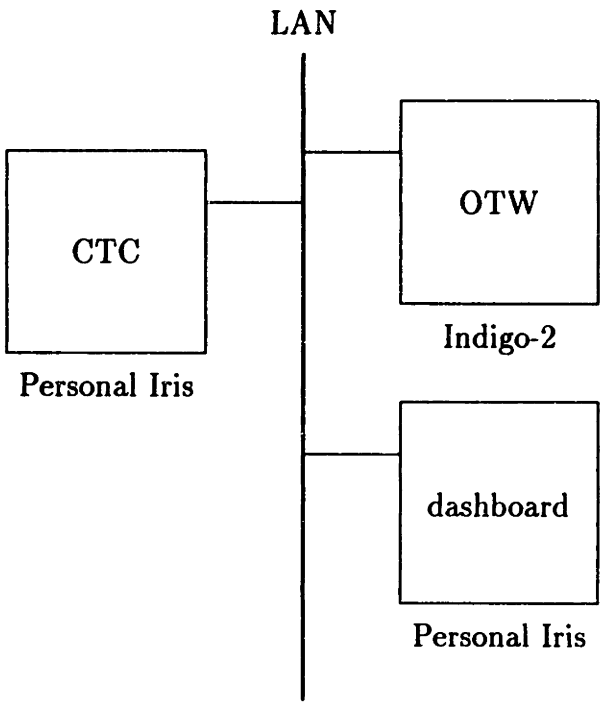

Figure 4-19: System Configuration for Control Automation Experiment

ulation. The second machine used was a Personal Iris, executing the OTW server.

In the case of the control automation experiment, the alternate CTC simulation (utilizing a five-aspect signaling system), was run on a Personal Iris. The control automation train simulation was run on the Indigo-2 machine, configured to display the OTW view, while the second Personal Iris machine was executing the dashboard server program. A system schematic of this configuration is shown in figure 4-19.

Chapter 5

Experiments

While the safety state model is of interest as a theoretical construct, its true value becomes much clearer when applied to an operational system. It is only when the risk event and the precipitating conditions are defined and a system is observed that the theory of the safety state model shows its potential.

High-speed rail provides an excellent testbed for understanding the mechanics of the safety state model. Conveniently, the experimental demonstration of the safety state model was coordinated with another area of research interest in high-speed rail. This research, aimed at understanding the relationship between vehicle control automation and operator situation awareness, is counterpart to concurrent research in the effects of display aiding on operator performance [1][2]. The high-speed rail simulation system, described in chapter 4 was utilized for both experiments.

The control automation experiment was conducted as a formal controlled behavioral study, addressing experimental design issues as necessary. The data used for demonstration of the safety state model was then extracted from the data recorded during the control automation experiment. In this sense, demonstration of the safety state model was "piggy-backed" onto the control automation experiment.

This chapter describes the experimental work that was performed. It is logically divided into two major sections. The first describes the safety state model demonstration, and the second describes the control automation experiment.

5.1 Demonstration of Safety State Model

The control automation experiment served as a data generator for demonstrating the safety state model. The relationship between these two activities highlights one of the strengths of the safety state model—it is useful in conjunction with other measurements or observations.

There is a fundamental difficulty in validating the safety state model. Typically, validation of a model requires the comparison of some measure or effect to another measure which has been previously qualified. However, the safety state model provides a new type of measurement. As a result, there is no other measure available for direct comparison. Hence, a problem arises—how are we to know if the model provides any useful information?

As a thought experiment, let us consider a world in which the speedometer has not yet been invented. The only method available for measuring speed is to measure the time used by a vehicle to cross a known distance, and to divide the distance by the time to obtain an average speed. One day, an inventor discovers a method for measuring speed as the vehicle is moving, and calls the instrument a *speedometer.* To validate that it works correctly, the inventor can only compare the average speed recorded by the speedometer over a finite distance, because there is no other method available for measuring instantaneous speed.

The safety state model presents a similar situation. There is no other method available for estimating dynamic risk probability as a function of system state, so the only level of validation that can be performed is comparison of the average risk level predicted by the model to the actual occurrence of failure events. We can also also demonstrate that the model provides additional useful information about the safety-relevant performance of individual operators.

5.1.1 Results of Safety State Model Application

As described in section 3.3, there are four distinct phases of operation with the safety state model (figure 3-4). In the risk identification phase, the risk event and contribut-

ing conditions are identified. The system observation phase involves measurement of an operational system. The model calibration phase is the point at which the observation data, in the form of safety state trajectories, are combined into the Markov model and used to estimate the risk probability as a function of system state. Finally, the risk estimation phase is the point at which the risk probability function is applied to the safety state trajectories, transforming safety state trajectories into risk probability trajectories.

Phase 1 Results—Risk Identification

The risk event chosen in the application of the safety state model to high-speed rail operation was a grade crossing collision between rail and highway vehicles. Due to immense momentum, rail vehicles have braking distances which are very long (typically several kilometers). This is especially significant considering that the braking distance is much longer than the sight distance of the operator. When a train approaches a grade crossing, drivers of highway vehicles often attempt to "beat the train" in an attempt to avoid waiting at the crossing while the train crosses. Grade crossing collisions are a serious problem in existing operational rail systems, and the situation promises only to worsen in high-speed rail operation.

A seven-bit safety state model was defined, using the conditions listed in table 5.1. Five independent conditions were chosen. Note that the *automation mode* and *distance to crossing* conditions use the technique for combining conditions into a fewer number of bits, as described in chapter 3. Each of these two conditions contain four mutually exclusive cases. These cases are each combined into two-bit conditions, so that four cases only require two bits of safety state number instead of four.

Using automation mode as an example, there are four cases: manual mode, cruise control, programmed stop, and autopilot. In the general case, four conditions would require four independent bits in the safety state number. However, we know that these four conditions are, by system design, mutually exclusive—one, and only one, of these modes can be active at any one time. Thus, we can represent these modes with a two bit sub-word---00 for manual, 01 for cruise control, 10 for programmed

- distance to crossing—there are four mutually-exclusive cases:
  1. near (crossing is closer than emergency braking distance)
  2. medium (crossing is farther than emergency braking distance but closer than full service braking distance)
  3. far (crossing is farther than full service braking distance)
  4. none (vehicle stopped)
- service brakes—true if the service brakes are applied
- emergency brakes—true if emergency brakes have been activated, for any reason and by any mechanism
- automation mode—four mutually-exclusive modes:
  1. manual
  2. cruise control
  3. programmed stop
  4. autopilot
- obstruction present—true if a highway vehicle is stuck in the crossing

Table 5.1: Conditions for Safety State Model

stop, and 11 for autopilot.

In total, the conditions listed combine to form a seven-bit safety state number. The value of the safety state number ranges from 0 to 127—there are 128 operational states. The risk event, collision in the grade crossing, is defined as state 128. State 128 is a trapping state—once the system experiences a collision, it cannot be "undone." The resulting state transition matrix will be of dimension (129 $\times$ 129).

The relationship between the state number and the conditions is summarized in table 5.2. The first column lists the safety state numbers, which are the state labels in the Markov model. The second column shows a hexadecimal representation of the safety state number, which gives a sense of the bit assignments in the number. (A hexadecimal number is a base-16 representation. Each digit is equivalent to a four-bit binary number. Therefore, hexadecimal representation allows rapid visual inspection of bit states.) The third column shows the set of conditions which combine to create the safety state number.

An interval of 1 second was specified as the time slice interval period. In general, it is not required that the interval period parameter be specified during the risk identification phase, as tuning of this value is useful during the model calibration phase. However, this interval cannot be smaller than the resolution of the time stamp in the data recording. Therefore, it is useful to give a preliminary specification of the interval period so as to design a proper bound in the time resolution of the system observation phase.

Given the scaling limitations of the model (chapter 3, an analyst might be tempted to limit the amount of data collected. Doing so, in fact, is the least optimal strategy. The highest cost component of the analysis, in terms of both time and equipment costs, is system observation. Therefore, the best approach is to specify as many conditions as possible in the first pass of phase 1, with the intent of gathering as much data as possible. This strategy affords the possibility of revising the set of conditions at a later point in time, after the system observation data has been gathered.

state	hex	description
0	0x00	nobstruct, manual, nestop, nbrake, none
1	0x01	nobstruct, manual, nestop, nbrake, far
2	0x02	nobstruct, manual, nestop, nbrake, close
3	0x03	nobstruct, manual, nestop, nbrake, medium
4	0x04	nobstruct, manual, nestop, brake, none
5	0x05	nobstruct, manual, nestop, brake, far
6	0x06	nobstruct, manual, nestop, brake, close
7	0x07	nobstruct, manual, nestop, brake, medium
8	0x08	nobstruct, manual, estop, nbrake, none
9	0x09	nobstruct, manual, estop, nbrake, far
10	0x0a	nobstruct, manual, estop, nbrake, close
11	0x0b	nobstruct, manual, estop, nbrake, medium
12	0x0c	nobstruct, manual, estop, brake, none
13	0x0d	nobstruct, manual, estop, brake, far
14	0x0e	nobstruct, manual, estop, brake, close
15	0x0f	nobstruct, manual, estop, brake, medium
16	0x10	nobstruct, cruise, nestop, nbrake, none
17	0x11	nobstruct, cruise, nestop, nbrake, far
18	0x12	nobstruct, cruise, nestop, nbrake, close
19	0x13	nobstruct, cruise, nestop, nbrake, medium
20	0x14	nobstruct, cruise, nestop, brake, none
21	0x15	nobstruct, cruise, nestop, brake, far
22	0x16	nobstruct, cruise, nestop, brake, close
23	0x17	nobstruct, cruise, nestop, brake, medium
24	0x18	nobstruct, cruise, estop, nbrake, none
25	0x19	nobstruct, cruise, estop, nbrake, far
26	0x1a	nobstruct, cruise, estop, nbrake, close
27	0x1b	nobstruct, cruise, estop, nbrake, medium
28	0x1c	nobstruct, cruise, estop, brake, none
29	0x1d	nobstruct, cruise, estop, brake, far
30	0x1e	nobstruct, cruise, estop, brake, close
31	0x1f	nobstruct, cruise, estop, brake, medium

Table 5.2: Summary of Safety States in Seven-Bit Model

state	hex	description
32	0x20	nobstruct, pstop, nestop, nbrake, none
33	0x21	nobstruct, pstop, nestop, nbrake, far
34	0x22	nobstruct, pstop, nestop, nbrake, close
35	0x23	nobstruct, pstop, nestop, nbrake, medium
36	0x24	nobstruct, pstop, nestop, brake, none
37	0x25	nobstruct, pstop, nestop, brake, far
38	0x26	nobstruct, pstop, nestop, brake, close
39	0x27	nobstruct, pstop, nestop, brake, medium
40	0x28	nobstruct, pstop, estop, nbrake, none
41	0x29	nobstruct, pstop, estop, nbrake, far
42	0x2a	nobstruct, pstop, estop, nbrake, close
43	0x2b	nobstruct, pstop, estop, nbrake, medium
44	0x2c	nobstruct, pstop, estop, brake, none
45	0x2d	nobstruct, pstop, estop, brake, far
46	0x2e	nobstruct, pstop, estop, brake, close
47	0x2f	nobstruct, pstop, estop, brake, medium
48	0x30	nobstruct, autop, nestop, nbrake, none
49	0x31	nobstruct, autop, nestop, nbrake, far
50	0x32	nobstruct, autop, nestop, nbrake, close
51	0x33	nobstruct, autop, nestop, nbrake, medium
52	0x34	nobstruct, autop, nestop, brake, none
53	0x35	nobstruct, autop, nestop, brake, far
54	0x36	nobstruct, autop, nestop, brake, close
55	0x37	nobstruct, autop, nestop, brake, medium
56	0x38	nobstruct, autop, estop, nbrake, none
57	0x39	nobstruct, autop, estop, nbrake, far
58	0x3a	nobstruct, autop, estop, nbrake, close
59	0x3b	nobstruct, autop, estop, nbrake, medium
60	0x3c	nobstruct, autop, estop, brake, none
61	0x3d	nobstruct, autop, estop, brake, far
62	0x3e	nobstruct, autop, estop, brake, close
63	0x3f	nobstruct, autop, estop, brake, medium

Table 5.2: Summary of Safety States in Seven-Bit Model (continued)

state	hex	description
64	0x40	obstruct, manual, nestop, nbrake, none
65	0x41	obstruct, manual, nestop, nbrake, far
66	0x42	obstruct, manual, nestop, nbrake, close
67	0x43	obstruct, manual, nestop, nbrake, medium
68	0x44	obstruct, manual, nestop, brake, none
69	0x45	obstruct, manual, nestop, brake, far
70	0x46	obstruct, manual, nestop, brake, close
71	0x47	obstruct, manual, nestop, brake, medium
72	0x48	obstruct, manual, estop, nbrake, none
73	0x49	obstruct, manual, estop, nbrake, far
74	0x4a	obstruct, manual, estop, nbrake, close
75	0x4b	obstruct, manual, estop, nbrake, medium
76	0x4c	obstruct, manual, estop, brake, none
77	0x4d	obstruct, manual, estop, brake, far
78	0x4e	obstruct, manual, estop, brake, close
79	0x4f	obstruct, manual, estop, brake, medium
80	0x50	obstruct, cruise, nestop, nbrake, none
81	0x51	obstruct, cruise, nestop, nbrake, far
82	0x52	obstruct, cruise, nestop, nbrake, close
83	0x53	obstruct, cruise, nestop, nbrake, medium
84	0x54	obstruct, cruise, nestop, brake, none
85	0x55	obstruct, cruise, nestop, brake, far
86	0x56	obstruct, cruise, nestop, brake, close
87	0x57	obstruct, cruise, nestop, brake, medium
88	0x58	obstruct, cruise, estop, nbrake, none
89	0x59	obstruct, cruise, estop, nbrake, far
90	0x5a	obstruct, cruise, estop, nbrake, close
91	0x5b	obstruct, cruise, estop, nbrake, medium
92	0x5c	obstruct, cruise, estop, brake, none
93	0x5d	obstruct, cruise, estop, brake, far
94	0x5e	obstruct, cruise, estop, brake, close
95	0x5f	obstruct, cruise, estop, brake, medium

Table 5.2: Summary of Safety States in Seven-Bit Model (continued)

state	hex	description
96	0x60	obstruct, pstop, nestop, nbrake, none
97	0x61	obstruct, pstop, nestop, nbrake, far
98	0x62	obstruct, pstop, nestop, nbrake, close
99	0x63	obstruct, pstop, nestop, nbrake, medium
100	0x64	obstruct, pstop, nestop, brake, none
101	0x65	obstruct, pstop, nestop, brake, far
102	0x66	obstruct, pstop, nestop, brake, close
103	0x67	obstruct, pstop, nestop, brake, medium
104	0x68	obstruct, pstop, estop, nbrake, none
105	0x69	obstruct, pstop, estop, nbrake, far
106	0x6a	obstruct, pstop, estop, nbrake, close
107	0x6b	obstruct, pstop, estop, nbrake, medium
108	0x6c	obstruct, pstop, estop, brake, none
109	0x6d	obstruct, pstop, estop, brake, far
110	0x6e	obstruct, pstop, estop, brake, close
111	0x6f	obstruct, pstop, estop, brake, medium
112	0x70	obstruct, autop, nestop, nbrake, none
113	0x71	obstruct, autop, nestop, nbrake, far
114	0x72	obstruct, autop, nestop, nbrake, close
115	0x73	obstruct, autop, nestop, nbrake, medium
116	0x74	obstruct, autop, nestop, brake, none
117	0x75	obstruct, autop, nestop, brake, far
118	0x76	obstruct, autop, nestop, brake, close
119	0x77	obstruct, autop, nestop, brake, medium
120	0x78	obstruct, autop, estop, nbrake, none
121	0x79	obstruct, autop, estop, nbrake, far
122	0x7a	obstruct, autop, estop, nbrake, close
123	0x7b	obstruct, autop, estop, nbrake, medium
124	0x7c	obstruct, autop, estop, brake, none
125	0x7d	obstruct, autop, estop, brake, far
126	0x7e	obstruct, autop, estop, brake, close
127	0x7f	obstruct, autop, estop, brake, medium
128	0x80	collision

Table 5.2: Summary of Safety States in Seven-Bit Model (continued)

Phase 2 Results—System Observation

In the system observation phase, an operational system is observed and data from that system are recorded. The system used for system observation was the high-speed rail simulation system, while in operation for the control automation experiment (described in section 5.2).

During the experiment, data records were recorded into a disk file by the train simulation program. The recorded data included periodic summaries of the vehicle state, with the vehicle position, speed, and operator input (throttle position) recorded. The data were recorded at least as frequently as 600 millisecond intervals, with a higher rate when the operator was moving the control lever. Also included in the data file are records of the operator input at the control buttons, as well as state changes within in the vehicle (such as vehicle system failures).

Because such a comprehensive collection of data was recorded, the resultant data files are very large. Typically, the raw data file for a three-hour session is on the order of 1.5 to 2 megabytes in size. The raw data file was post-processed by a program named `ss_process`, which reduces the raw data to a safety state trajectory as a function of time. The collection of safety state trajectories are used in phase 3 (model calibration).

Even before entering the model calibration phase, the safety state trajectories provide useful information. It is possible to quickly identify the occurrences of the risk event, and the sequence of states that led up to each risk event. An intuition is developed about the relative occurrence of risk events and the causality leading to those occurrences. Such an intuition, in itself, is useful in the arena of risk assessment and safety engineering.

For example, inspection of the 96 safety state trajectories recorded during the control automation experiment allowed identification of 10 grade crossing collisions. Table 5.3 lists the sequence of states that occurred in the 300 second (i.e., 5 minute) interval prior to each collision. The state numbers correspond with the list provided in table 5.2. We can see that states 70, 78, and 114 can be considered high-risk states,

subject number	test type	state sequence prior to collision
2	manual	5-1-5-1-3-2-1-3-2-1-65-69-71-70-128
2	full	49-1-5-1-49-51-50-49-113-65-69-71-69-71-70-128
3	partial	17-1-5-1-17-1-5-21-5-1-17-19-83-67-71-70-78-128
3	full	45-44-36-32-33-32-0-4-0-1-49-51-50-49-51-115-114-128
6	full	49-113-115-114-122-74-78-128
10	practice	1-5-7-71-67-71-70-78-128
11	full	49-1-5-1-49-113-115-67-71-70-128
13	practice	1-5-1-5-1-5-1-3-7-6-2-1-5-7-3-67-71-70-128
18	practice	1-5-1-5-1-5-1-65-67-71-70-128
18	full	49-113-115-114-66-70-78-128

Table 5.3: Summary of Collision Occurrences

as the state immediately preceding a collision event is always one of these three.

Phase 3 Results—Model Calibration

The purpose of the model calibration phase is to transform the safety state trajectories, generated during the system observation phase, into a risk transformation function. The risk transformation function summarizes the relationship between the safety state and the risk probability of failure. The algorithms for these calculations is presented in chapter 3.

A graphical representation of the state transition matrix which resulted from the recorded system observation data is shown in figure 5-1. In this view, the height of the surface represents the state transition probability, and the x- and y-axes correspond to the row and column indices of the state transition matrix. The point furthest from the viewpoint is the trapping state, with a probability of 1. In effect, we are viewing the "back side" of the state transition matrix, as the x-axis corresponds to the columns of the matrix and the y-axis corresponds to the rows of the matrix. Note that the higher transition probability values lie along the main diagonal of the matrix—this is an indication that the average holding time for each of the states is relatively long compared to the time interval used by the model.

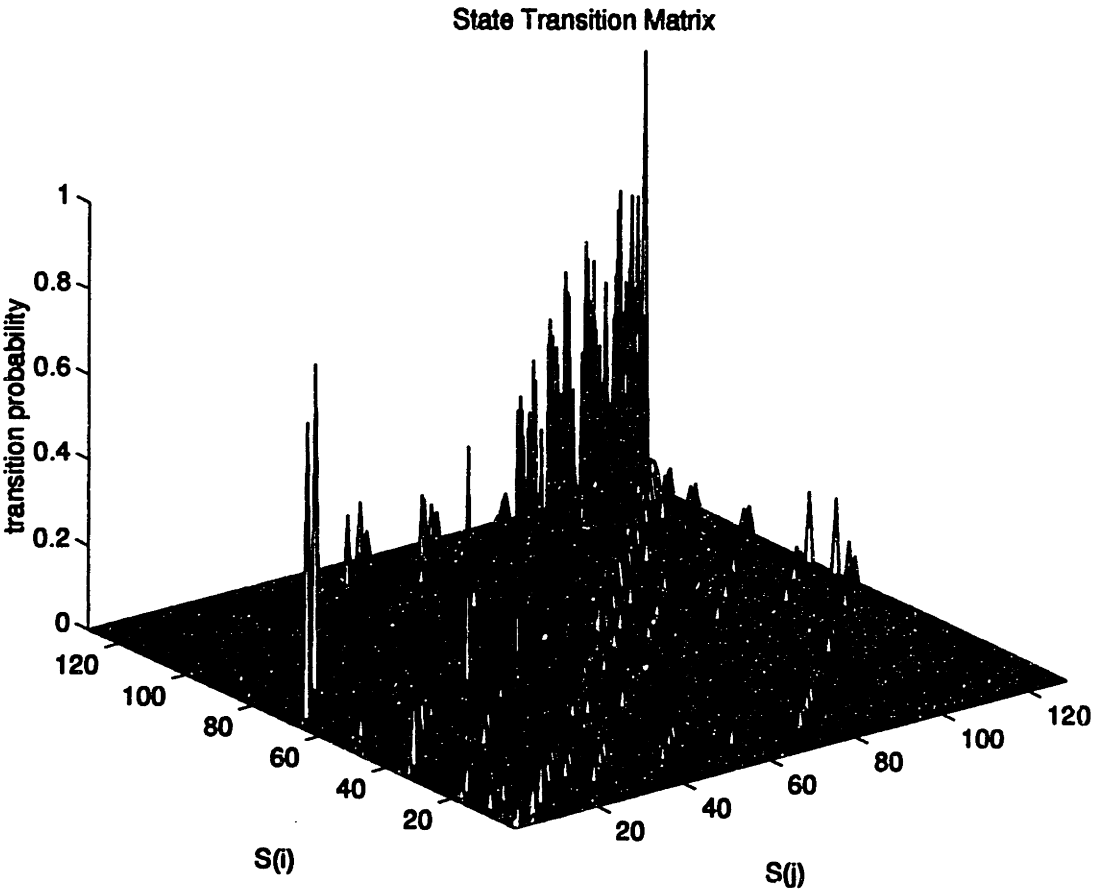

Figure 5-1: State Transition Matrix (Mesh View)

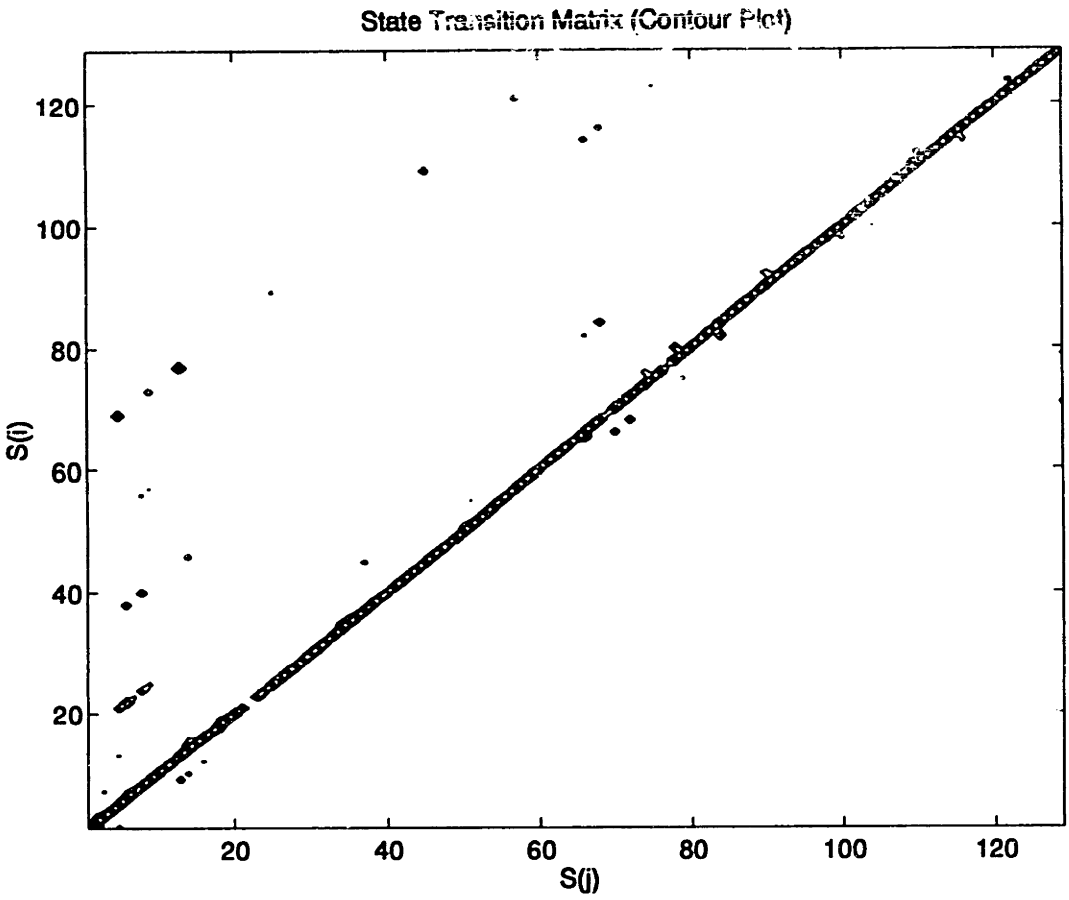

Figure 5-2: State Transition Matrix (Contour View)

An alternate view of the state transition matrix is shown in figure 5-2. This view is a plan view of the matrix, with contour lines showing ridges of equal probability. This view gives a sense of the sparseness of the state transition matrix, and reinforces the sense that the largest probability values lie along the main diagonal of the matrix.

Once the state transition matrix was calculated, it was used directly in computation of the mean time to failure estimates, as outlined in section 3.2.3. The resultant MTTF output is shown in figure 5-3. The MTTF was then converted, element by element, into an equivalent risk probability. Figure 5-4 shows the risk probability function obtained from the MTTF function shown in figure 5-3. Table 5.4 lists the values of the dynamic risk probability function.

The risk probability function was subsequently applied to the safety state trajectories, as described in section 3.3. The output of this phase (i.e., the dynamic risk probability trajectories) is also useful. Through knowledge of the risk probability function, we can identify the states that have a high relative risk probability. For

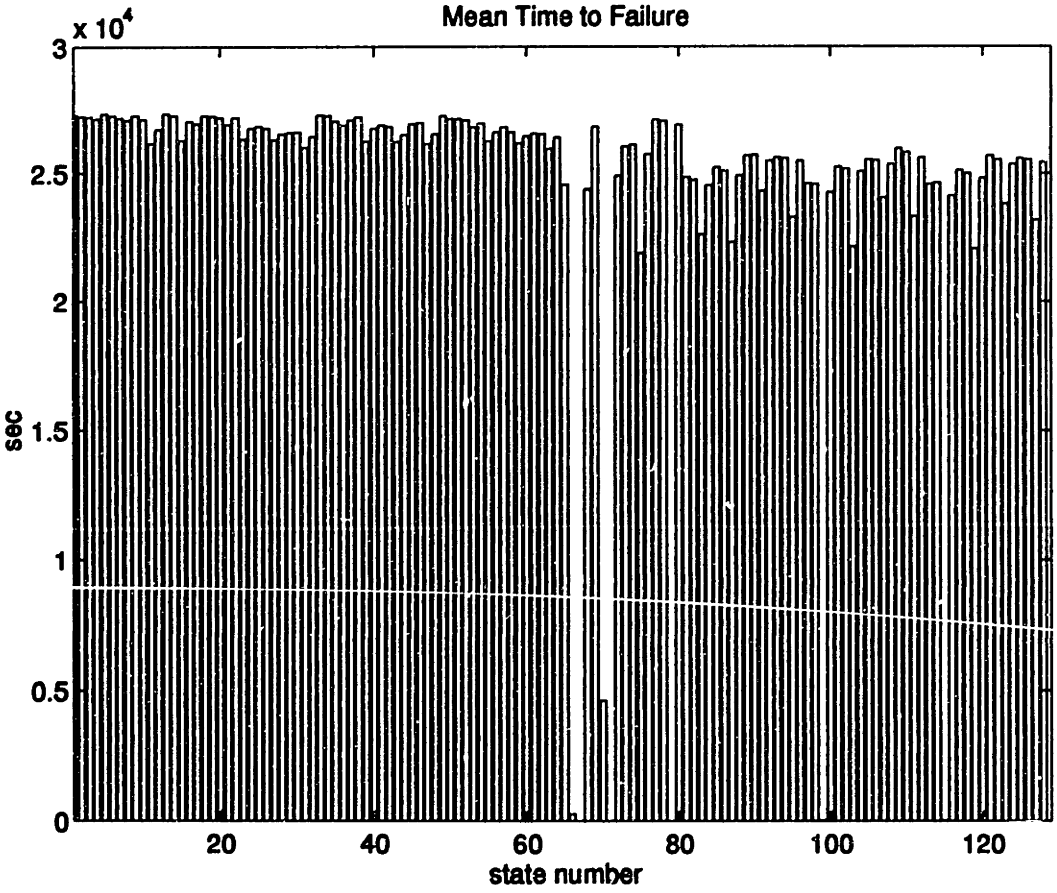

Figure 5-3: MTTF As a Function of Safety State

state	risk probability
0	$3.665911e-05$
1	$3.670811e-05$
2	$3.673025e-05$
3	$3.683592e-05$
4	$3.656192e-05$
5	$3.668415e-05$
6	$3.682628e-05$
7	$3.691074e-05$
8	$3.666303e-05$
9	$3.688520e-05$
10	$3.826712e-05$
11	$3.746290e-05$
12	$3.658067e-05$
13	$3.668879e-05$
14	$3.805374e-05$
15	$3.699488e-05$
16	$3.712783e-05$
17	$3.669579e-05$
18	$3.670769e-05$
19	$3.676927e-05$
20	$3.717917e-05$
21	$3.677300e-05$
22	$3.798726e-05$
23	$3.737859e-05$
24	$3.726584e-05$
25	$3.739357e-05$
26	$3.803250e-05$
27	$3.771626e-05$
28	$3.763017e-05$
29	$3.761168e-05$
30	$3.847551e-05$
31	$3.783688e-05$

Table 5.4: Risk Function Values

state	risk probability
32	$3.666176e-05$
33	$3.669719e-05$
34	$3.700730e-05$
35	$3.718579e-05$
36	$3.691381e-05$
37	$3.675101e-05$
38	$3.811438e-05$
39	$3.737176e-05$
40	$3.718822e-05$
41	$3.727720e-05$
42	$3.813179e-05$
43	$3.775556e-05$
44	$3.714476e-05$
45	$3.707438e-05$
46	$3.824729e-05$
47	$3.769418e-05$
48	$3.667969e-05$
49	$3.683916e-05$
50	$3.685009e-05$
51	$3.691703e-05$
52	$3.731020e-05$
53	$3.708427e-05$
54	$3.810838e-05$
55	$3.759305e-05$
56	$3.731806e-05$
57	$3.760678e-05$
58	$3.822687e-05$
59	$3.780334e-05$
60	$3.767245e-05$
61	$3.768800e-05$
62	$3.853966e-05$
63	$3.789223e-05$

Table 5.4: Risk Function Values (continued)

state	risk probability
64	$4.072072e-05$
65	$3.670344e-03$
66	$8.954155e-02$
67	$4.102894e-05$
68	$3.729218e-05$
69	$2.172839e-04$
70	$1.718398e-01$
71	$4.018481e-05$
72	$3.839946e-05$
73	$3.831102e-05$
74	$4.563649e-05$
75	$3.885834e-05$
76	$3.689360e-05$
77	$3.694418e-05$
78	$1.589488e-01$
79	$3.716552e-05$
80	$4.026902e-05$
81	$4.043406e-05$
82	$4.415102e-05$
83	$4.074997e-05$
84	$3.966327e-05$
85	$3.986338e-05$
86	$4.470954e-05$
87	$4.014169e-05$
88	$3.892607e-05$
89	$3.891547e-05$
90	$4.110291e-05$
91	$3.924490e-05$
92	$3.904216e-05$
93	$3.909455e-05$
94	$4.288274e-05$
95	$3.924964e-05$

Table 5.4: Risk Function Values (continued)

state	risk probability
96	$4.065515e-05$
97	$4.067023e-05$
98	$4.682744e-02$
99	$4.117748e-05$
100	$3.960947e-05$
101	$3.973881e-05$
102	$4.510500e-05$
103	$3.989938e-05$
104	$3.916737e-05$
105	$3.921421e-05$
106	$4.156458e-05$
107	$3.946551e-05$
108	$3.851417e-05$
109	$3.873631e-05$
110	$4.280711e-05$
111	$3.904190e-05$
112	$4.069965e-05$
113	$4.061480e-05$
114	$4.057289e-02$
115	$4.141718e-05$
116	$3.979895e-05$
117	$4.000007e-05$
118	$4.527933e-05$
119	$4.031103e-05$
120	$3.893162e-05$
121	$3.916019e-05$
122	$4.193459e-05$
123	$3.946246e-05$
124	$3.908867e-05$
125	$3.916637e-05$
126	$4.305445e-05$
127	$3.931972e-05$

Table 5.4: Risk Function Values (continued)

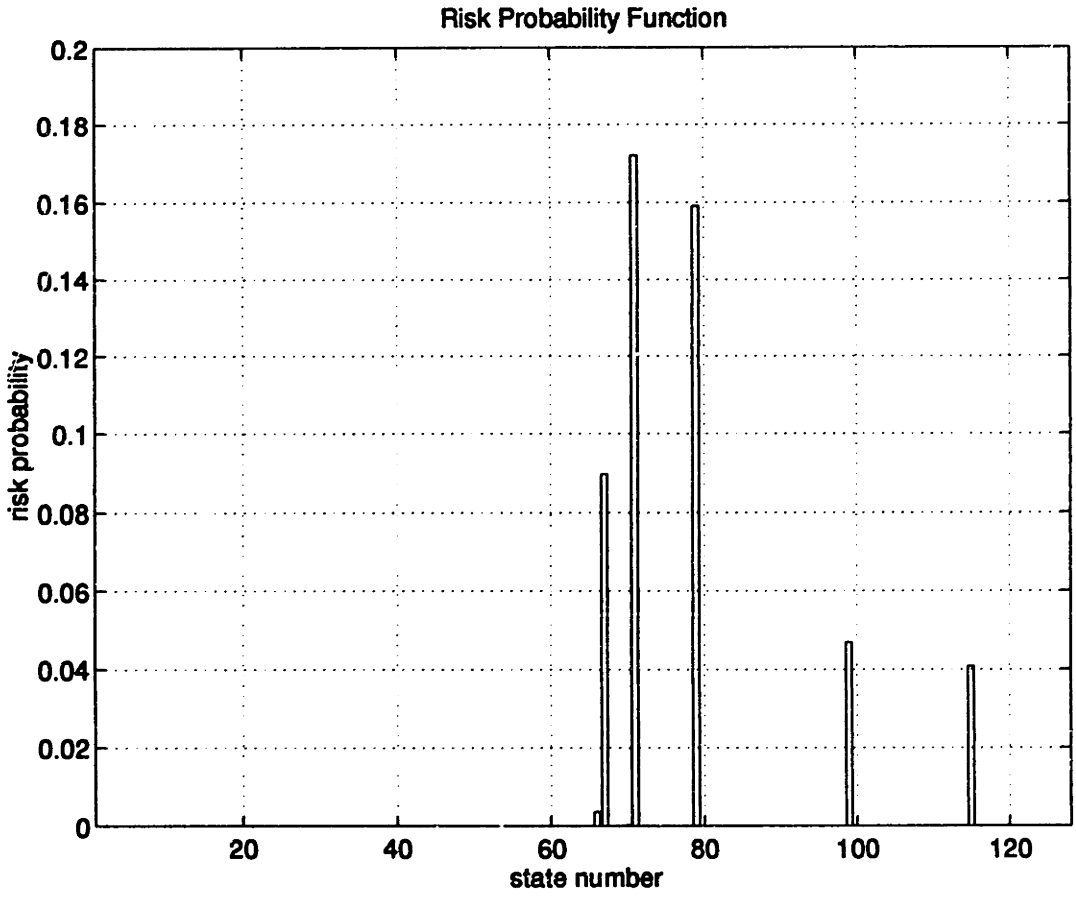

Figure 5-4: Risk Probability As a Function of Safety State

example, by inspecting figure 5-4 and table 5.4, we can clearly identify state 70 as the highest risk state. This analysis provides quantitative evidence of the intuition developed in the system observation phase.

Phase 4 Results—Risk Estimation

The risk estimation phase is the phase in which the risk probability function is applied to the safety state trajectories. The collection of safety state trajectories, which represent the output from the system observation phase, summarize the safety state values as a function of time. The risk probability function transforms each safety state into an equivalent risk probability. By applying the risk probability function to the safety state trajectories, a set of instantaneous risk probability trajectories are created.

A series of instantaneous risk trajectories is shown in figures 5-6 through 5-24. These figures show the instantaneous risk probability trajectories for all of the experimental sessions that were run as part of the control automation experiment (section 5.2). Each figure contains several trajectories, corresponding to the different test sessions that were conducted. The y-axis label indicates the automation level of the test. (The *random* test variant corresponds to the second training session, where the failure scenarios were generated by a random process.) In several cases, the instantaneous risk level rose much higher at isolated points—this phenomenon corresponds to the system occupying a high-risk state. Some of the high-risk states resulted in collisions, as outlined in table 5.3, while others could be considered "near collisions."

It is important to note the possible differences in displayed range in the instantaneous risk trajectories. The differences are due to the large dynamic range of the risk probability function (table 5.4, figure 5-4). As an example, we can look at the trajectories for subject 2 (figure 5-7). In the partial automation test, the instantaneous probability remained in the range between 0.000035 and 0.000040. In the random test, higher risks were experienced—two spikes near 0.004 dominate the range. Finally, in both the manual and full automation tests, spikes of 0.18 were experienced. We know that these spikes correlate with collisions that were experienced by subject 2.

Figures 5-25 through 5-43 show the cumulative risk probability trajectories for the same test sessions. These figures show the result of integrating the instantaneous risk probability trajectory with respect to time. Thus, the curves show continuously increasing functions, with significant jumps when a high-risk state is occupied. The cumulative risk functions show the effect of risk exposure.

In figures 5-44 through 5-62, we can see the average risk probability trajectories for the same test sessions. In these figures, the mean risk was calculated by dividing the cumulative risk by the amount of time that the system had been in operation. The spikes that occur in some of the average risk trajectories are evidence of high-risk state occupancy; the exponential decay following the spikes are evidence of the high-risk events being "absorbed" into the average value.

It is also important to note that the cumulative and average trajectories shown correspond to individual test sessions. If the average risk probability had been computed across all of the test sessions, the individual high-risk state occupancies would have a much smaller impact on the average, resulting in much smaller spikes in the trajectories.

5.1.2 Discussion

There are a number of interesting details that become apparent during application of the safety state model. First, there is the state transition matrix. As a two-dimensional matrix, it can be displayed as a surface, where the row and column indices constitute the x- and y-axes, respectively, and the state transition probabilities are represented as the height to form a surface (figure 5-1).

Observing this display, we can see that there is a row of "mountains" down the diagonal axis, with the last element being the largest. The last element is the probability of holding in the trapping state, which is by definition equal to one. The fact that the other diagonal elements are large indicates that the holding times are significant relative to the time scale, which is appropriate. If the holding times were on the same order as the time slice interval, then there is a possibility that state occupancies might not have been captured by the model during the system observation phase,

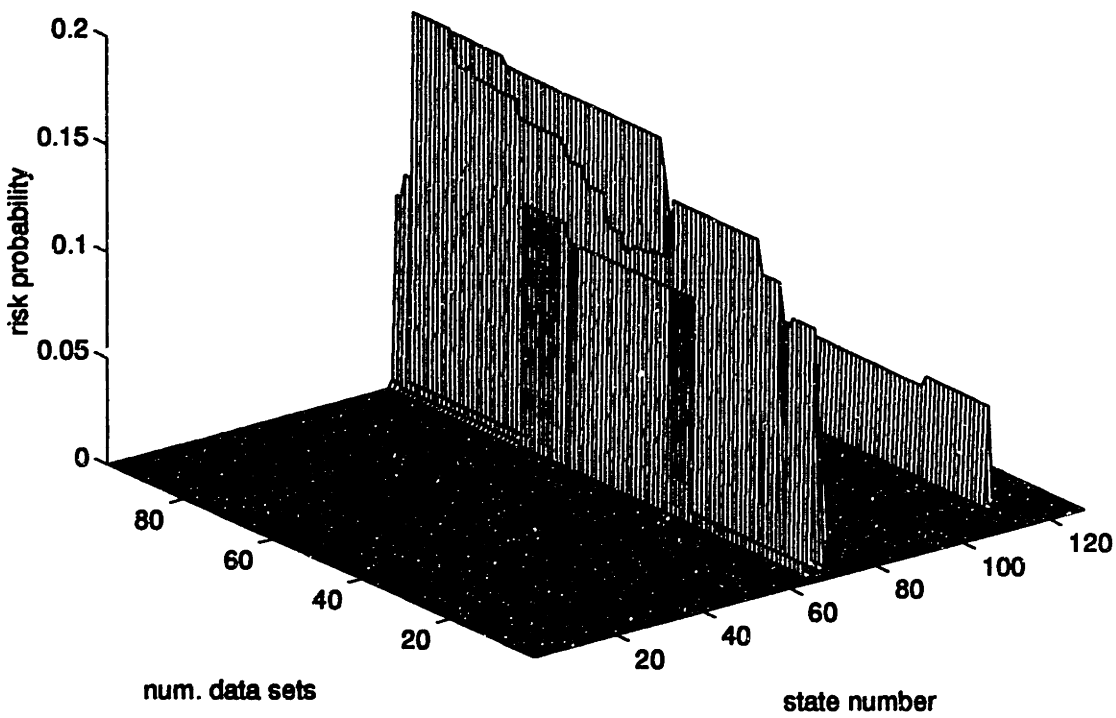

Figure 5-5: Evolution of Risk Probability Function

suggesting that the specified time slice interval was too large.

Let us also consider the evolution of the risk trajectory. Data from 96 test and training sessions was used in the model calibration phase. After each safety state trajectory was added to the statistics matrix, the state transition matrix, MTTF function, and risk function were calculated. In figure 5-5, the evolution of the risk probability function is shown. The safety state is shown along the x-axis The z-axis represents the risk probability. The y-axis represents the number of files that have been included to that point. If we were to slice a plane parallel to the x-z plane (perpendicular to the y-axis), the intersection with the surface would be the risk function after y files were incorporated.

The surface that results is quite interesting. When a smaller number of files are included, the risk probability function is flat, indicating that, as far as we know from the data presented, there is no significant difference in risk probability. The trend correlates with our intuitive sense, because until there is a collision, we do not assume

that one will happen (and hence P_f is an array of zeros). As the number of included files is increased, and the number of collisions rises, safety state trajectory paths with collisions are incorporated into the state transition matrix, and peaks start to appear in the risk function. By the tail end of the evolution, that is, after many of the safety state trajectories have been incorporated, the patterns have leveled out along the y-axis. This trend indicates that there exists a point where additional data does not substantially change the shape of the risk function (figure 5-5). At that point, the average behavior of the system has been learned.

Finally, to calibrate our results, we compare the collision data from the tests to the expected number of collisions that are predicted by the model. During the practice and test sessions, there were a total of 10 collisions (table 5.3). In addition, there were 20 training sessions, during which there were a total of 22 collisions. Thus, there were a total of 32 collisions that occurred in the data used for model calibration.

By integrating the risk trajectories for all the sessions, the expected number of failures are found, based on the safety state model risk estimates. A summary chart is shown in table 5.5, which shows the cumulative risk for each test session.

Based on the overall results found, we can state that the safety state model is a useful tool for estimating the dynamic risk probability.

subject	manual	partial	full	random
1	$4.035290e-01$	$5.014959e-01$	$3.892437e-01$	
2	$7.608489e-01$	$3.851640e-01$	$4.795381e-01$	$3.890328e-01$
3	$4.023435e-01$	$1.258353e+00$	$6.330565e-01$	$5.136620e-01$
4	$3.899003e-01$	$3.845045e-01$	$3.816475e-01$	$5.788982e-01$
5	$3.953636e-01$	$3.843103e-01$	$3.910797e-01$	$6.486107e-01$
6	$3.928005e-01$	$3.829303e-01$	$9.981355e-01$	$4.032766e-01$
7	$3.930121e-01$	$3.812272e-01$	$3.870368e-01$	$3.816339e-01$
8	$4.004559e-01$	$3.791819e-01$	$3.844246e-01$	$3.876574e-01$
9	$3.928296e-01$	$3.813987e-01$	$3.815847e-01$	$3.981727e-01$
10	$3.806737e-01$	$3.902816e-01$	$4.074672e-01$	$1.145297e+00$
11	$4.108821e-01$	$3.810783e-01$	$1.263948e+00$	$3.841769e-01$
12	$3.934166e-01$	$3.820926e-01$	$3.984218e-01$	$3.950625e-01$
13	$3.930352e-01$	$3.877314e-01$	$4.029230e-01$	$1.297403e+00$
14	$3.898836e-01$	$3.835280e-01$	$3.824788e-01$	$4.082567e-01$
15	$4.057855e-01$	$3.819577e-01$	$4.073429e-01$	$4.441813e-01$
16	$4.046387e-01$	$3.898611e-01$	$3.815841e-01$	$4.145703e-01$
17	$3.917257e-01$	$3.928006e-01$	$3.821744e-01$	$3.871100e-01$
18	$4.020111e-01$	$3.813885e-01$	$1.664306e+00$	$9.999186e-01$
19	$3.884216e-01$	$3.812561e-01$	$3.819115e-01$	$4.416250e-01$

Table 5.5: Cumulative Risk Summary

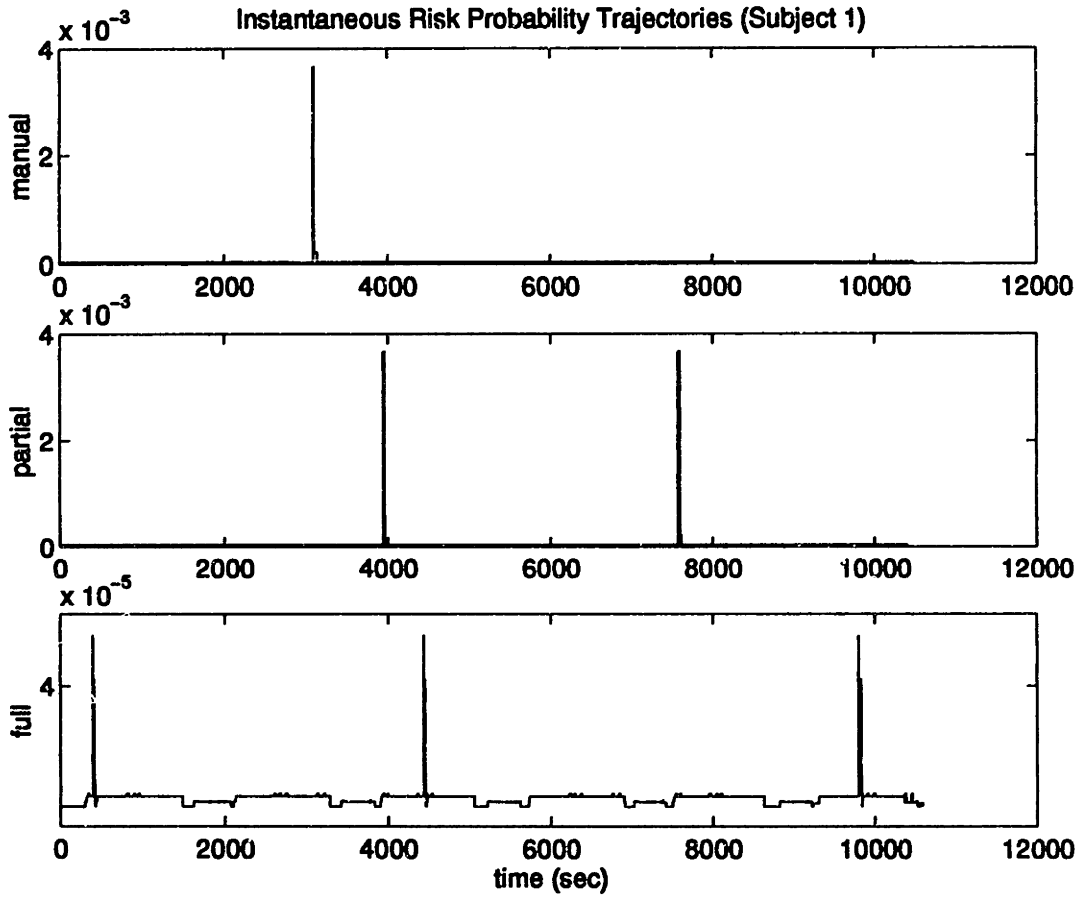

Figure 5-6: Instantaneous Risk Trajectories—Subject 1

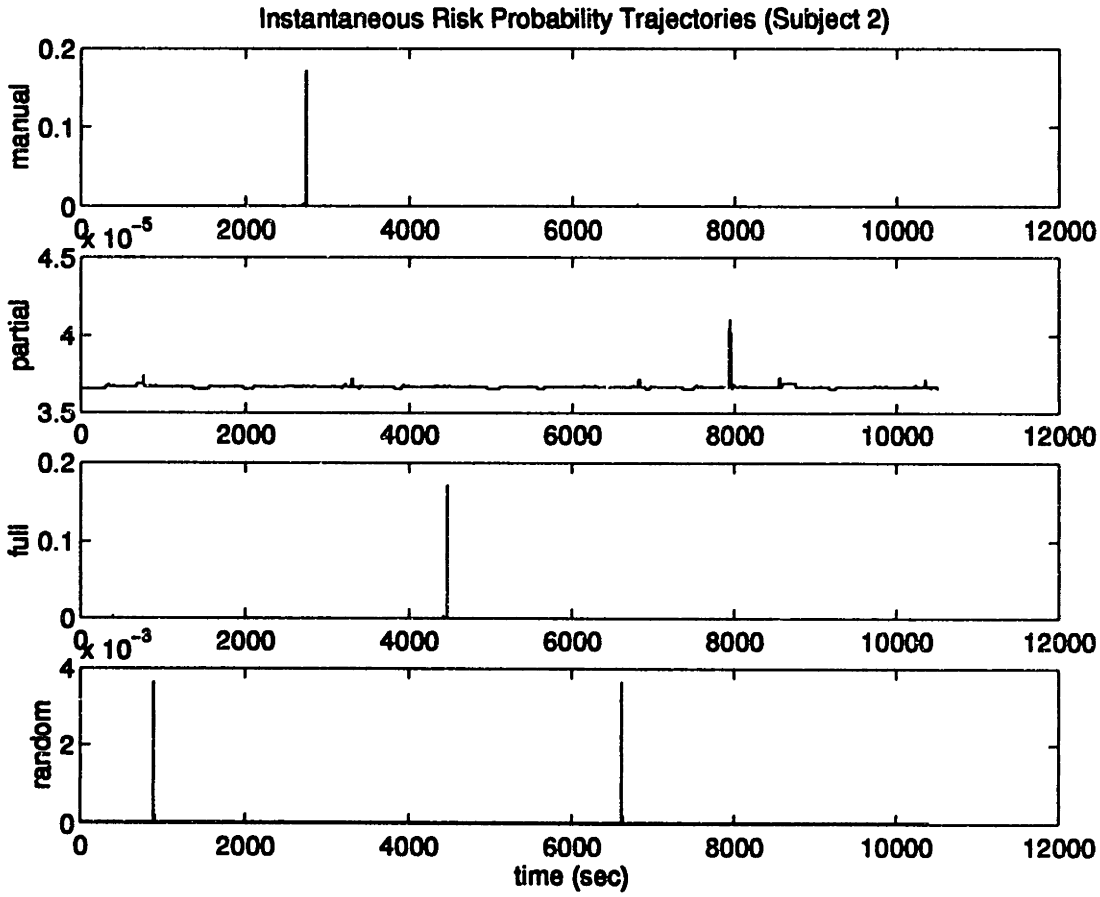

Figure 5-7: Instantaneous Risk Trajectories—Subject 2

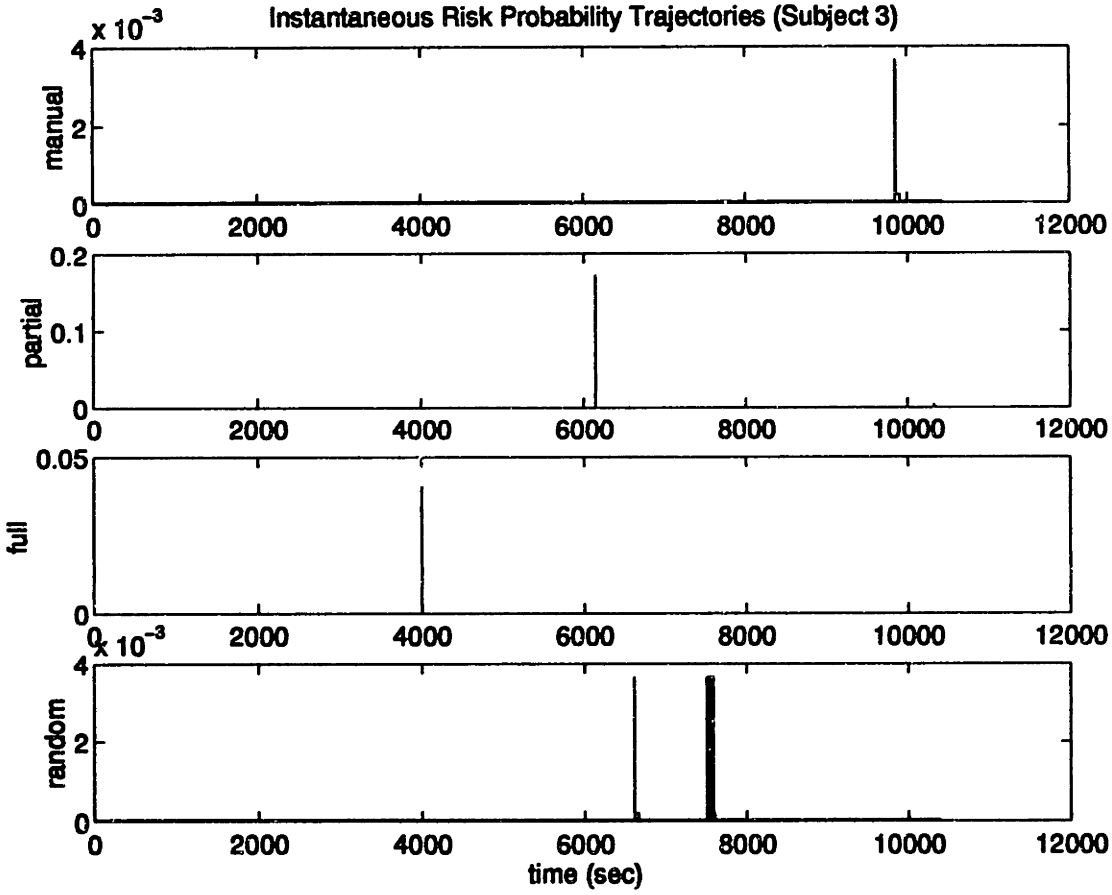

Figure 5-8: Instantaneous Risk Trajectories—Subject 3

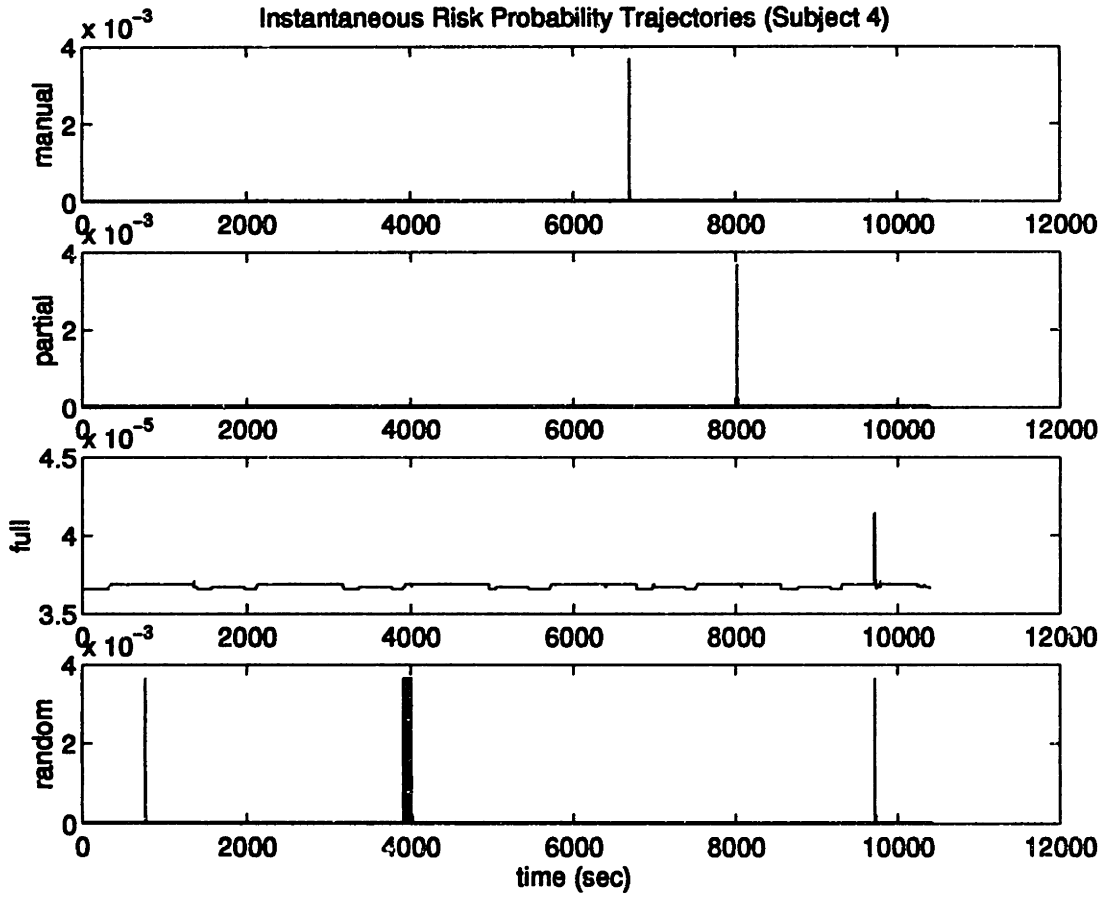

Figure 5-9: Instantaneous Risk Trajectories—Subject 4

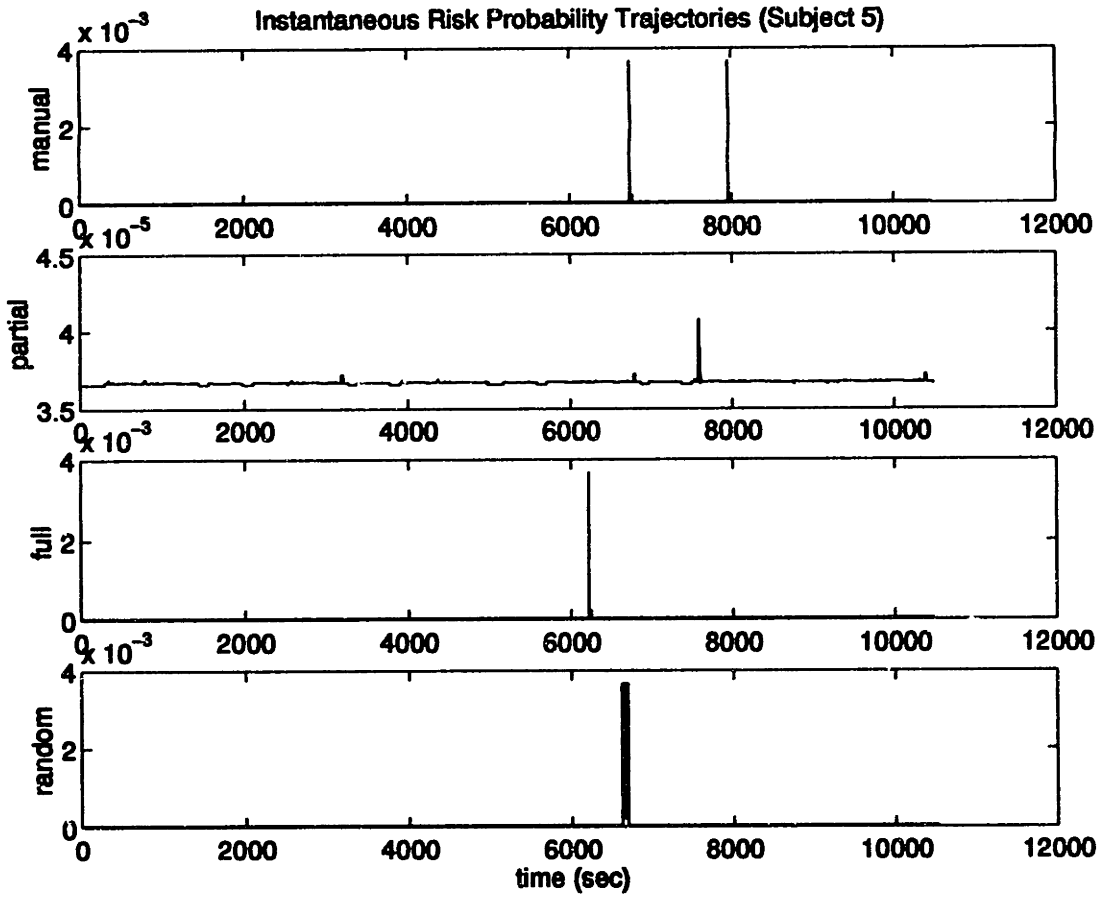

Figure 5-10: Instantaneous Risk Trajectories—Subject 5

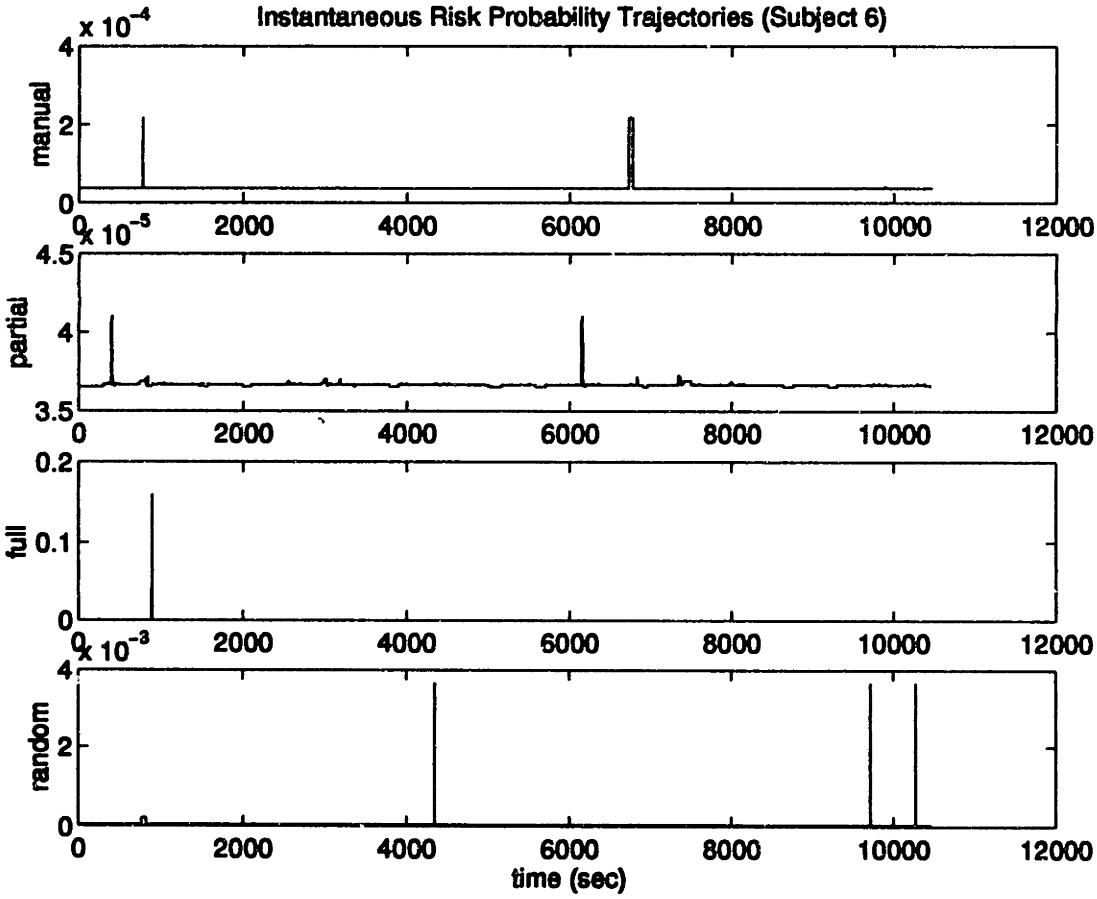

Figure 5-11: Instantaneous Risk Trajectories—Subject 6

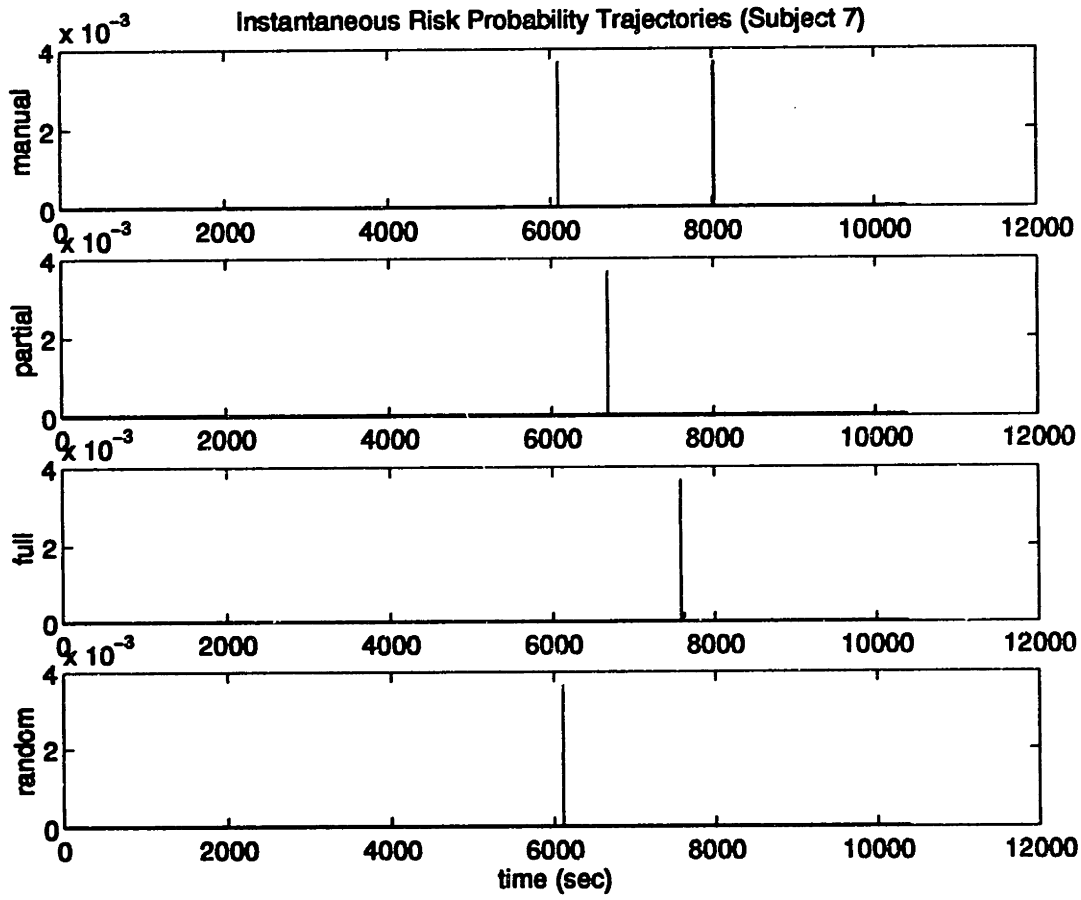

Figure 5-12: Instantaneous Risk Trajectories—Subject 7

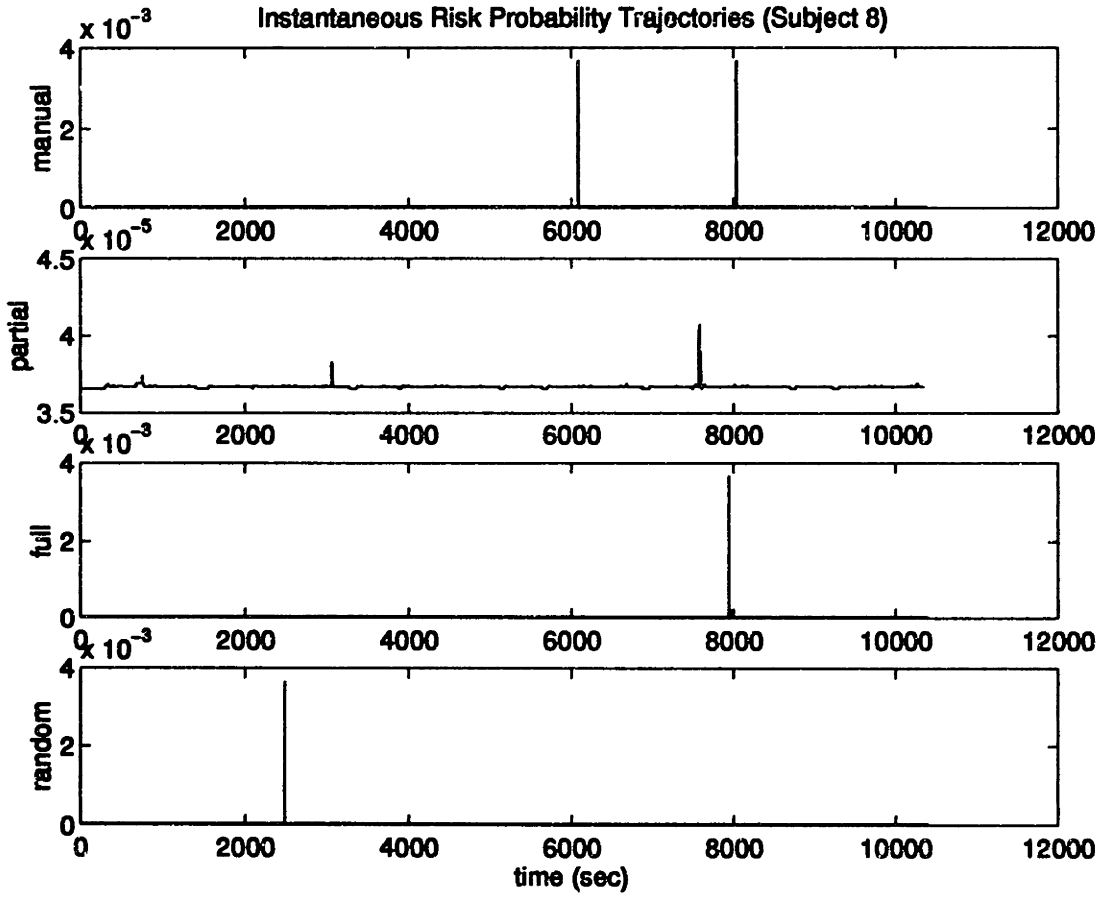

Figure 5-13: Instantaneous Risk Trajectories—Subject 8

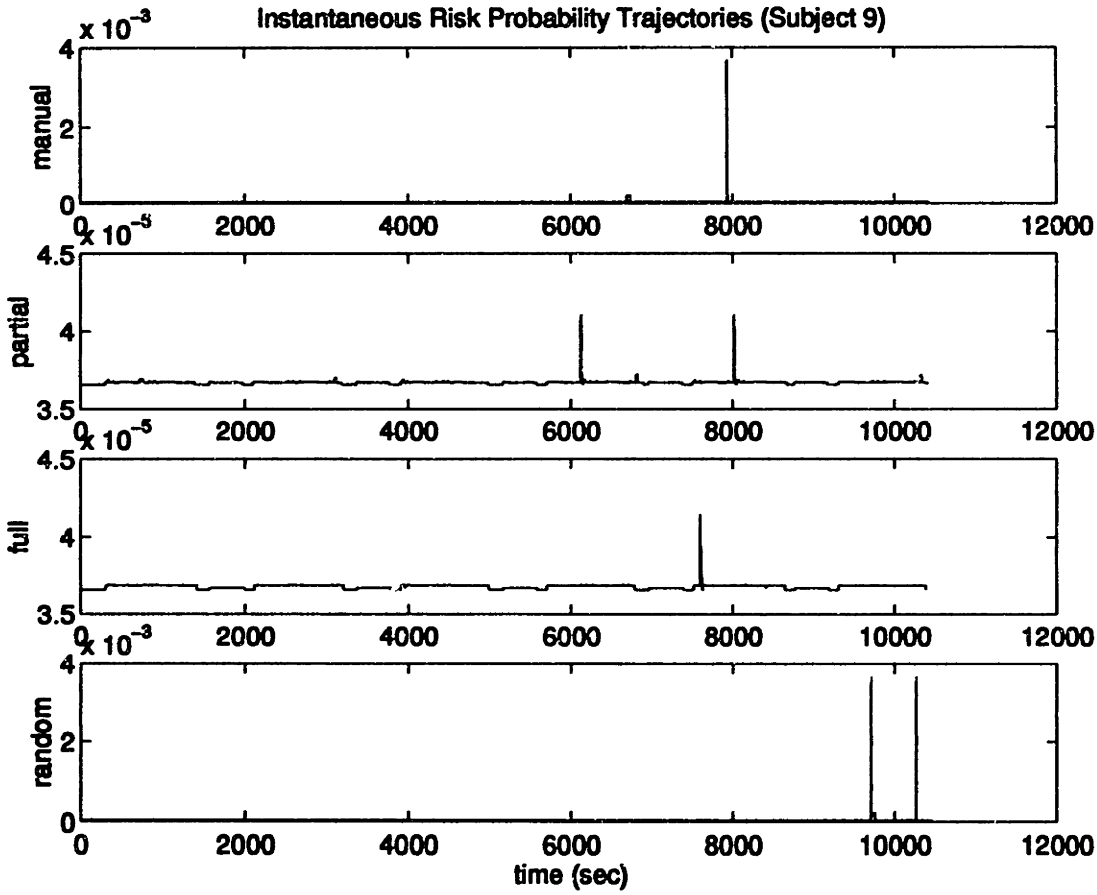

Figure 5-14: Instantaneous Risk Trajectories—Subject 9

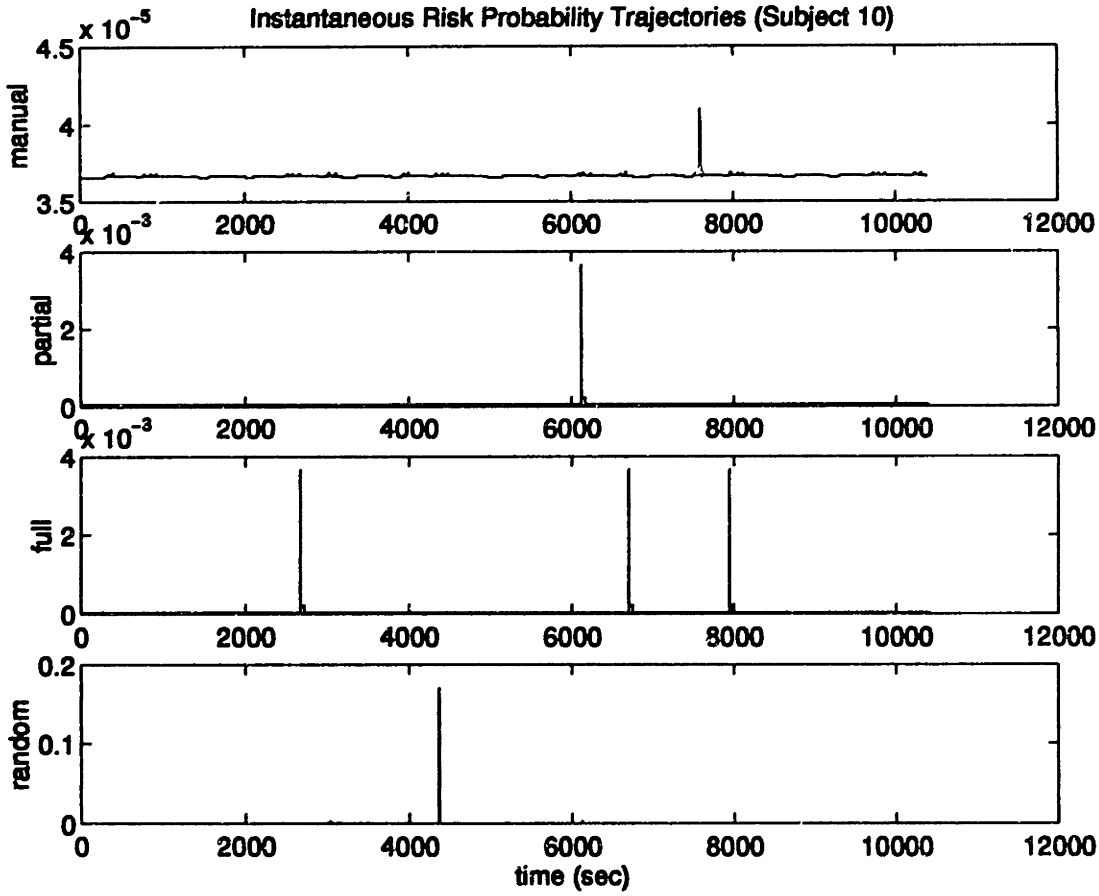

Figure 5-15: Instantaneous Risk Trajectories—Subject 10

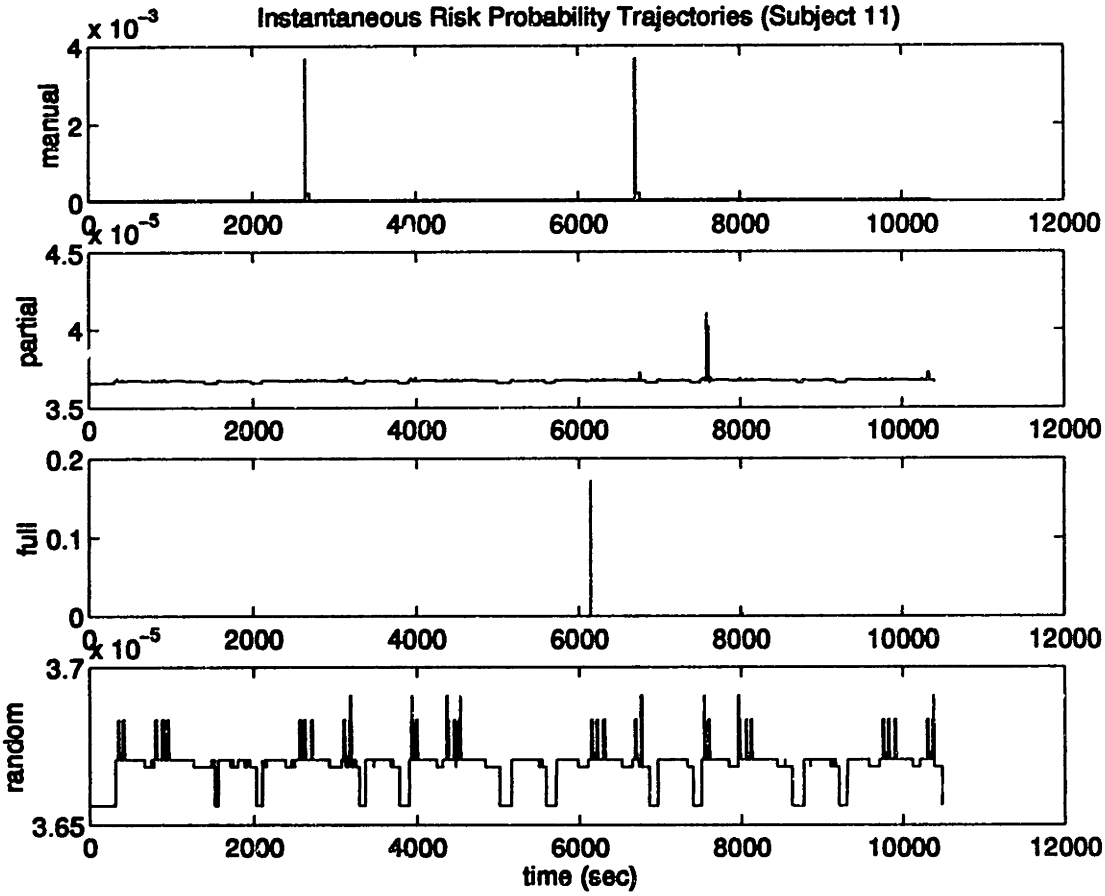

Figure 5-16: Instantaneous Risk Trajectories—Subject 11

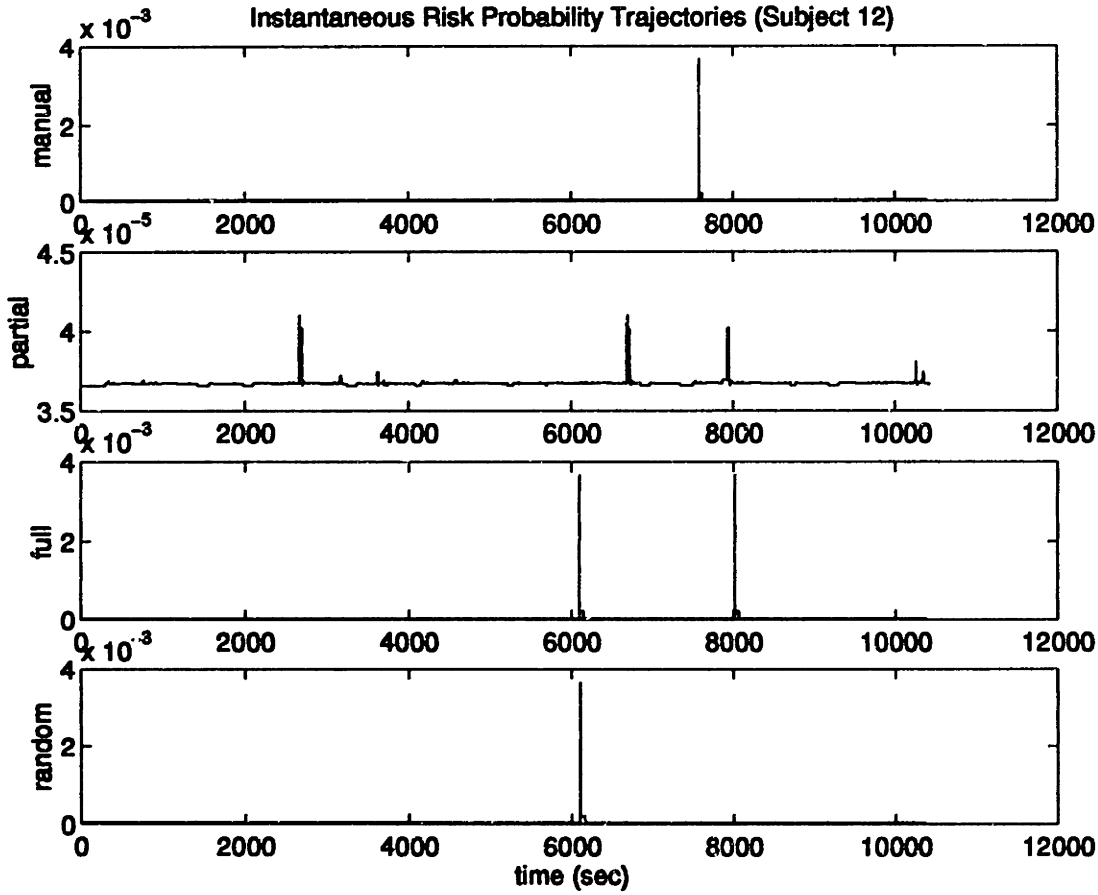

Figure 5-17: Instantaneous Risk Trajectories—Subject 12

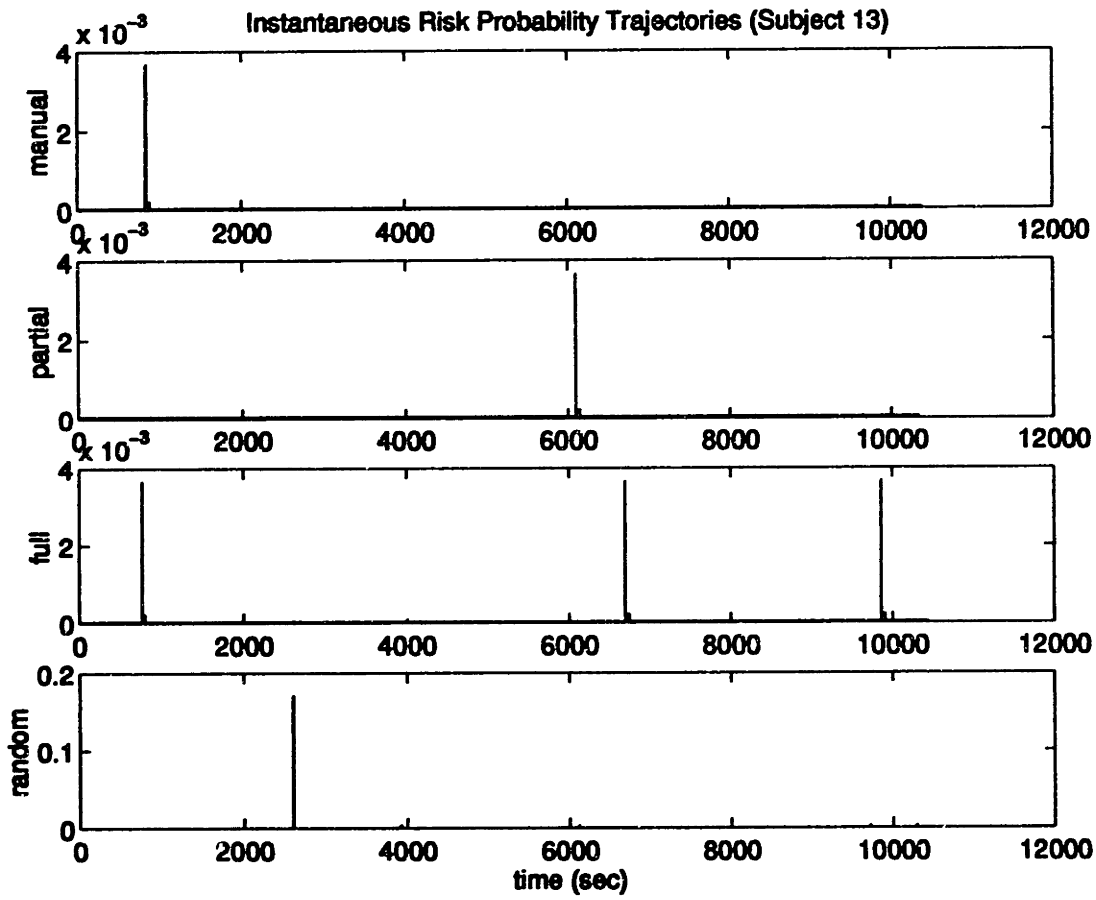

Figure 5-18: Instantaneous Risk Trajectories—Subject 13

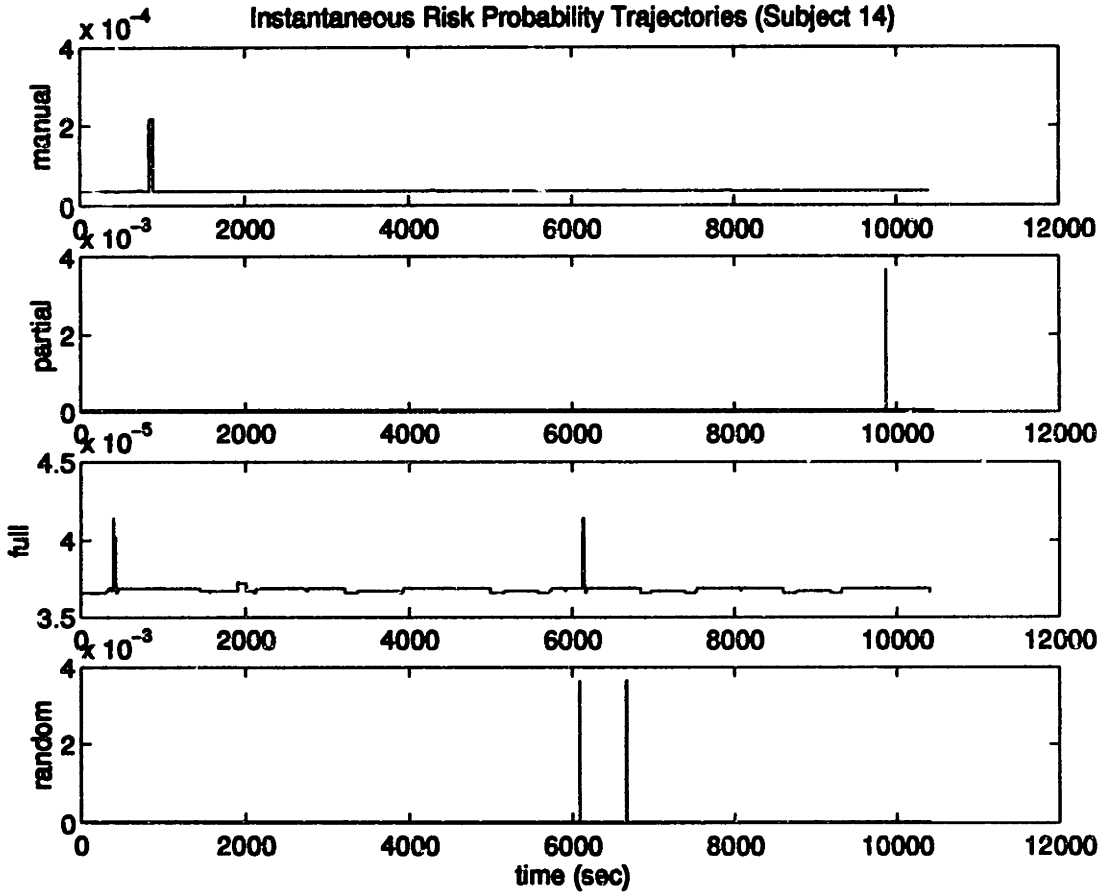

Figure 5-19: Instantaneous Risk Trajectories—Subject 14

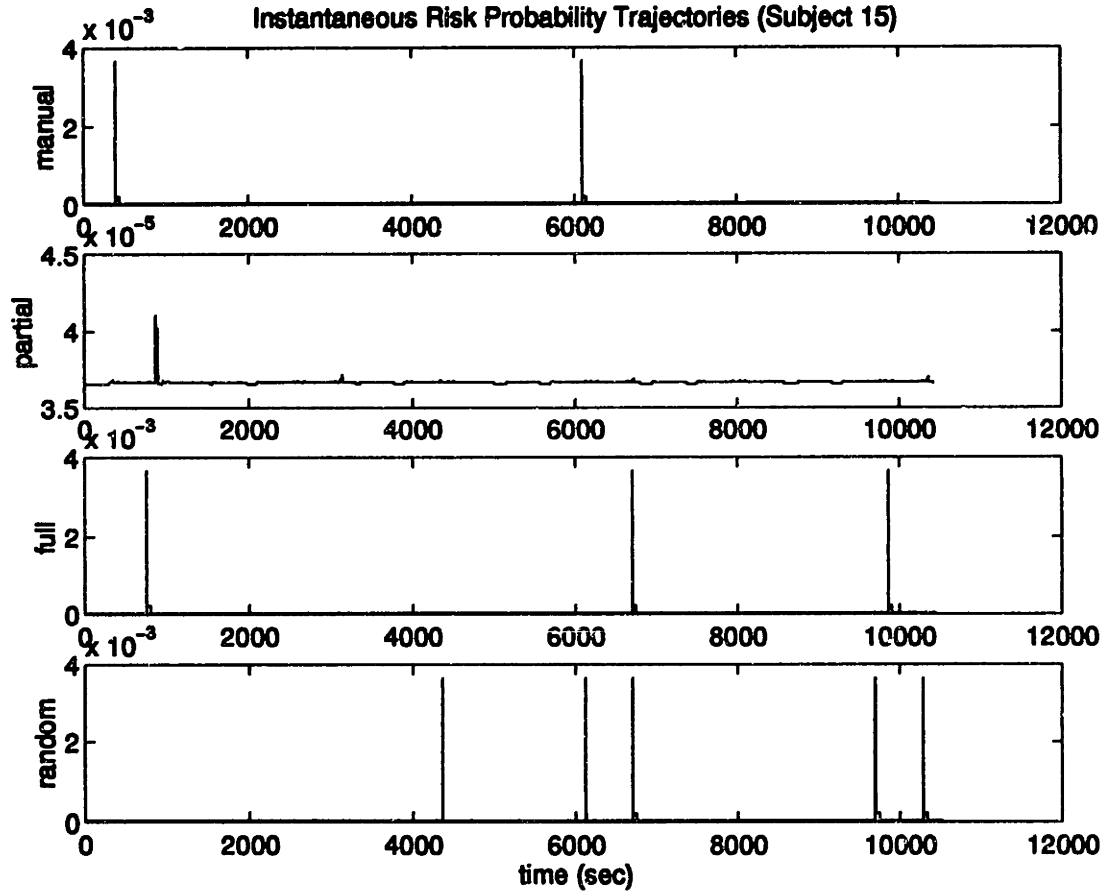

Figure 5-20: Instantaneous Risk Trajectories—Subject 15

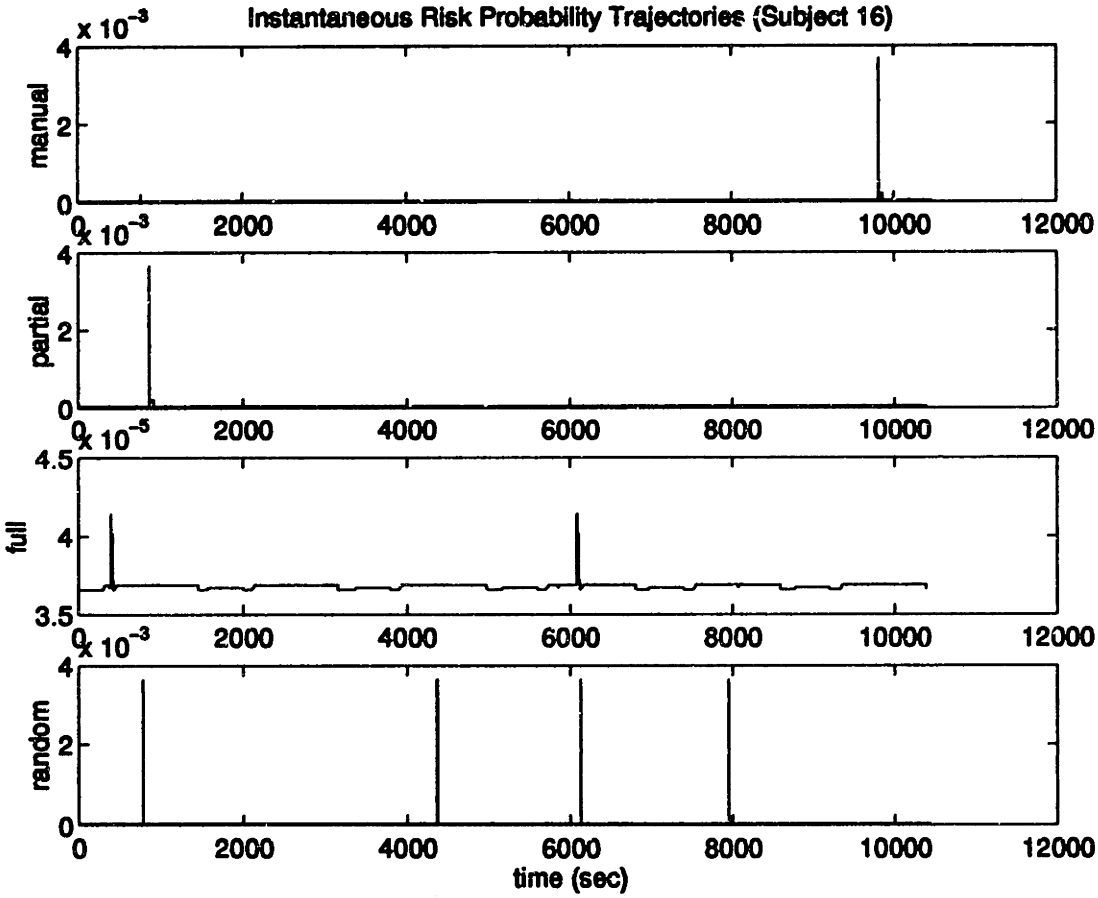

Figure 5-21: Instantaneous Risk Trajectories—Subject 16

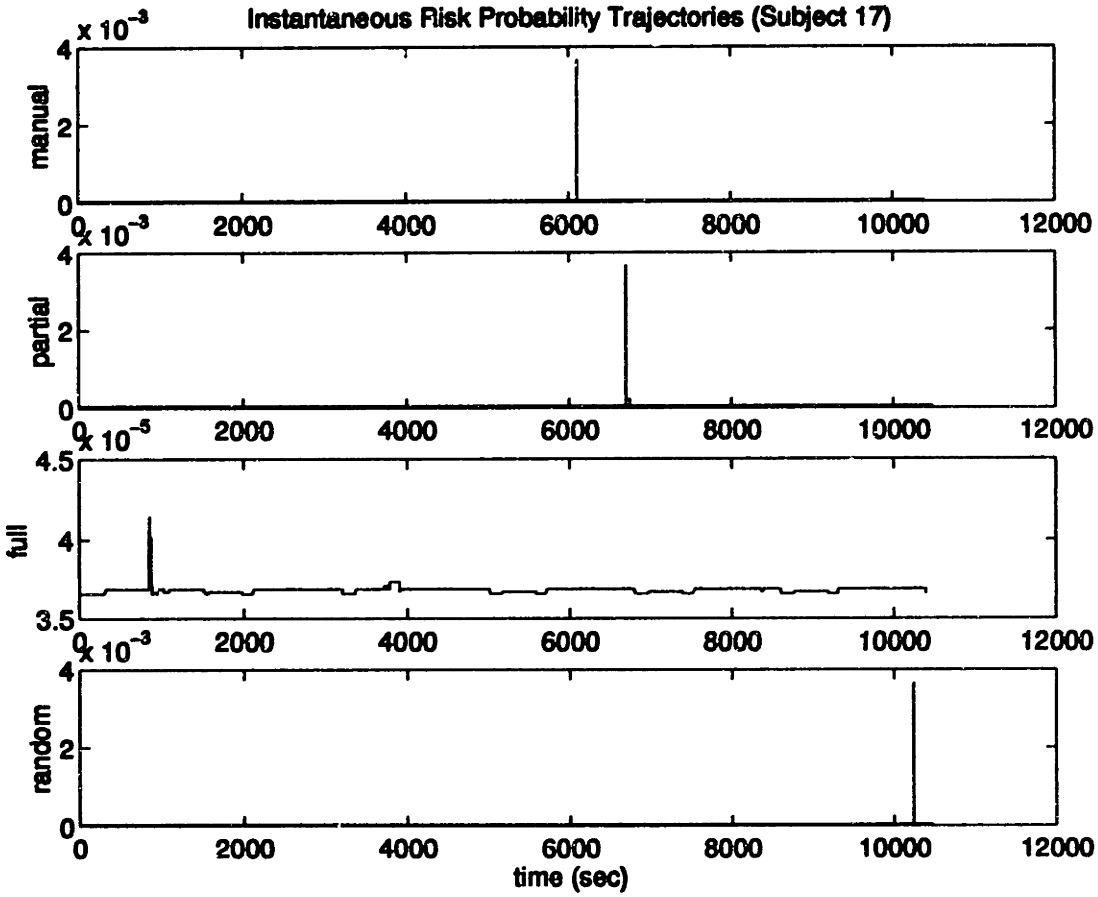

Figure 5-22: Instantaneous Risk Trajectories—Subject 17

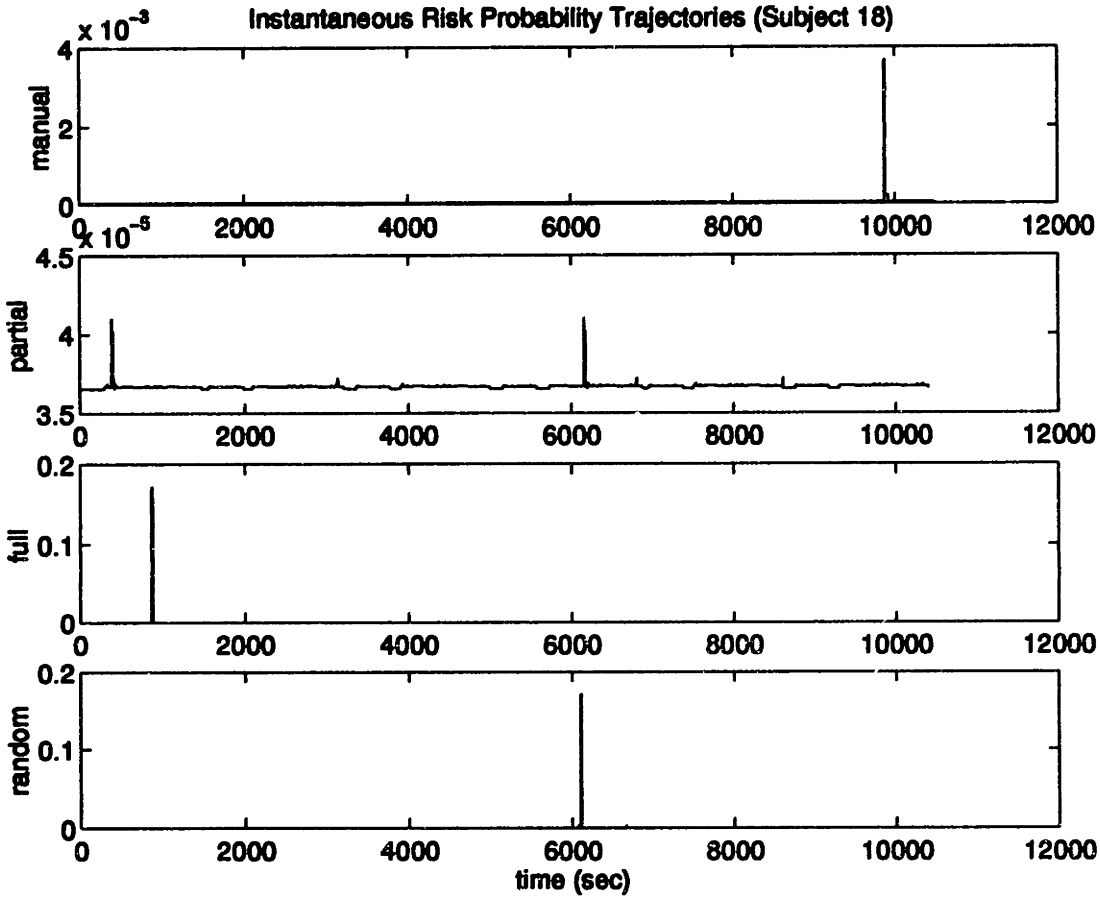

Figure 5-23: Instantaneous Risk Trajectories—Subject 18

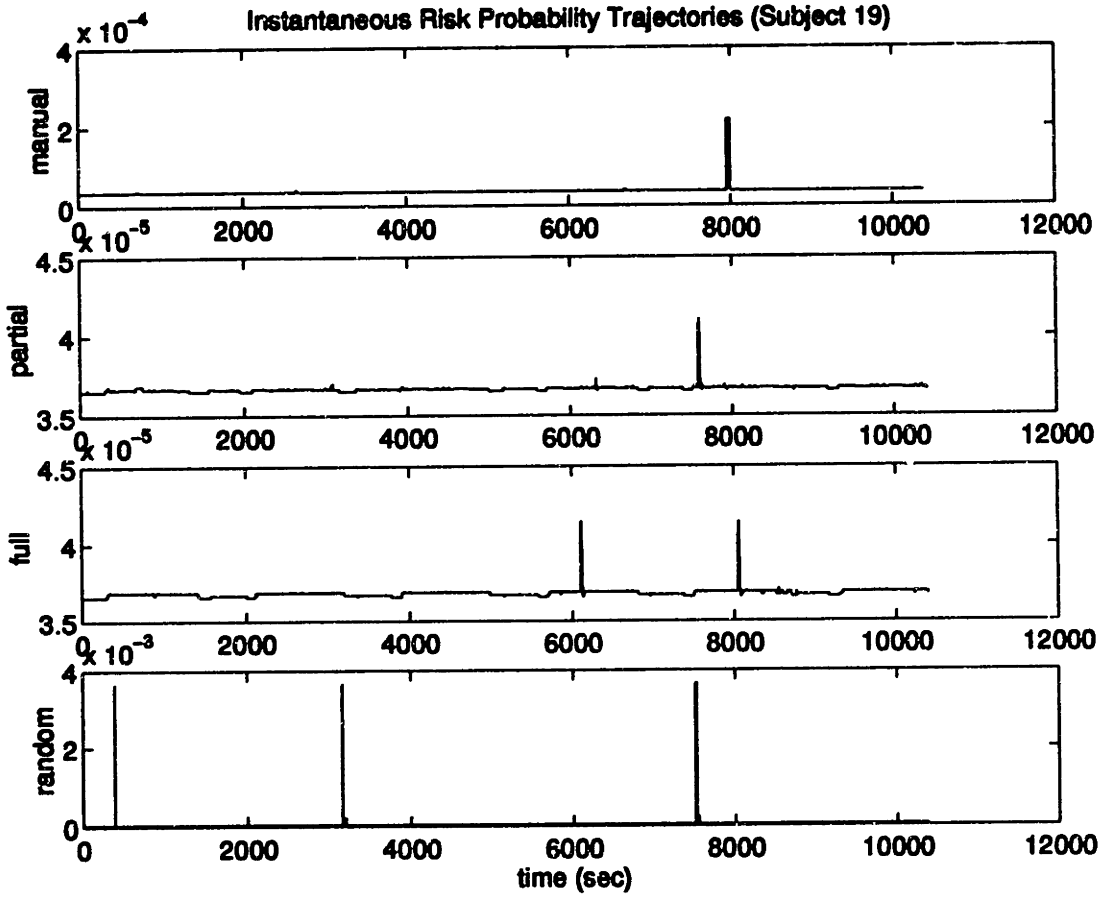

Figure 5-24: Instantaneous Risk Trajectories—Subject 19

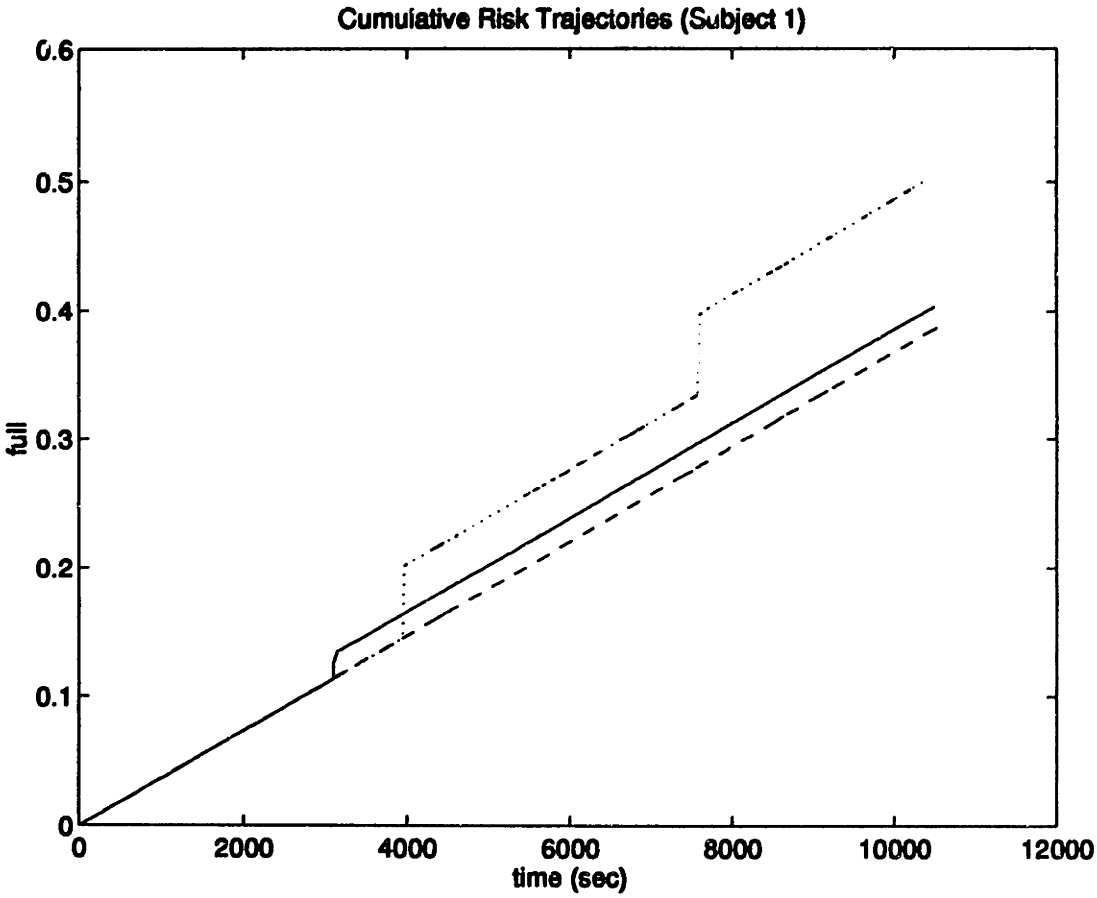

Figure 5-25: Cumulative Risk Trajectories—Subject 1

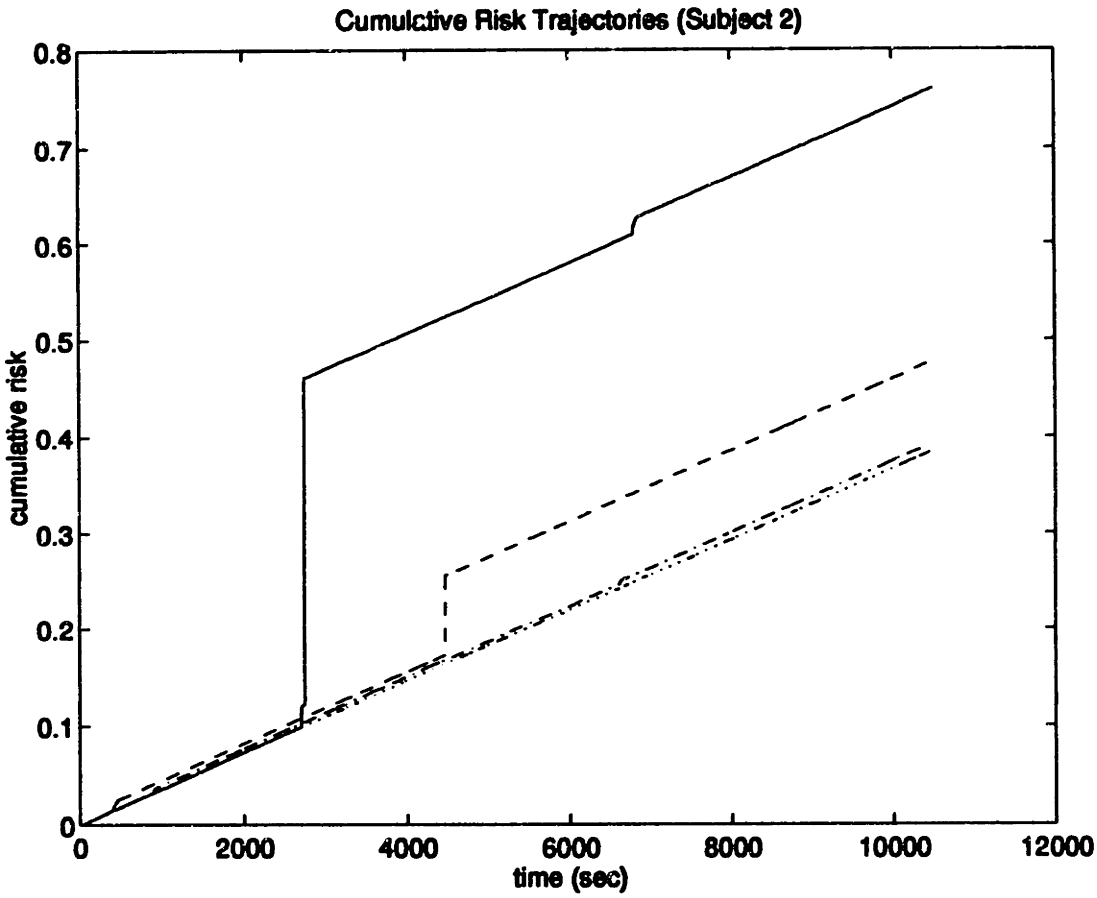

Figure 5-26: Cumulative Risk Trajectories—Subject 2

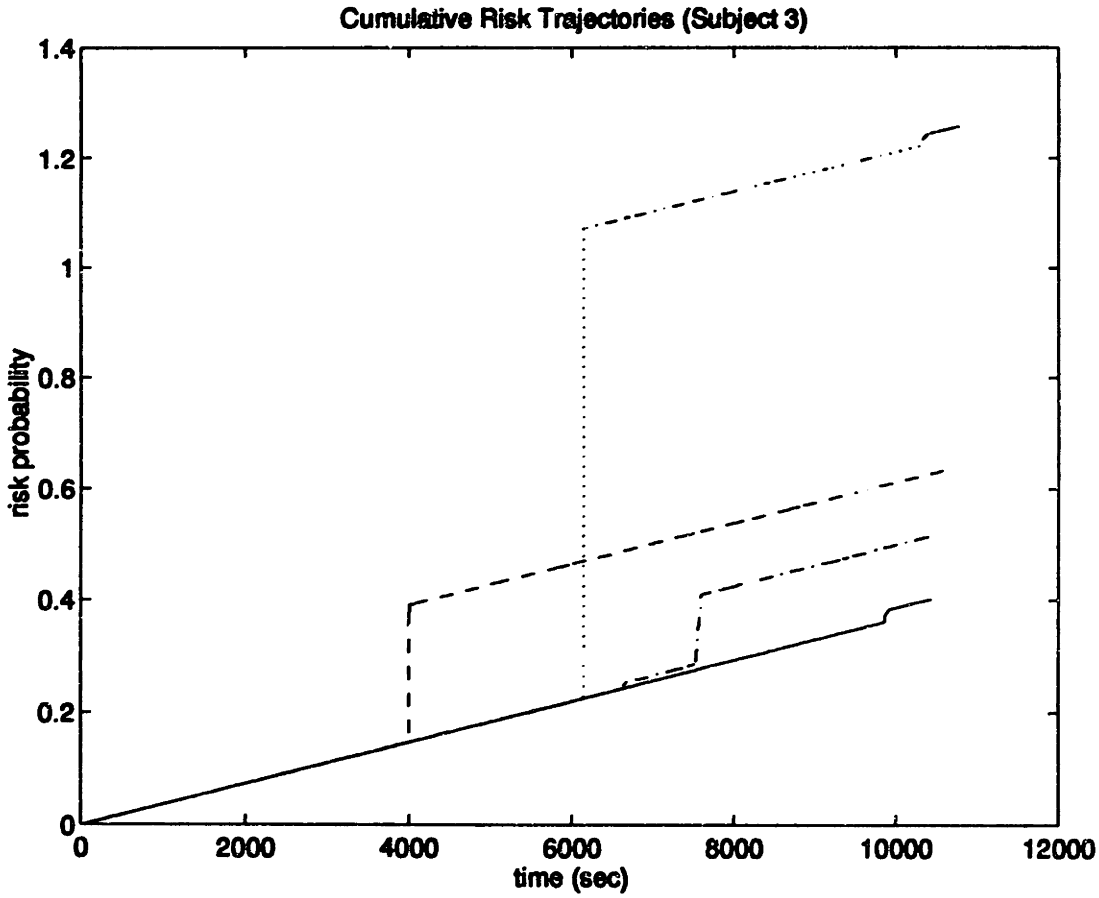

Figure 5-27: Cumulative Risk Trajectories—Subject 3

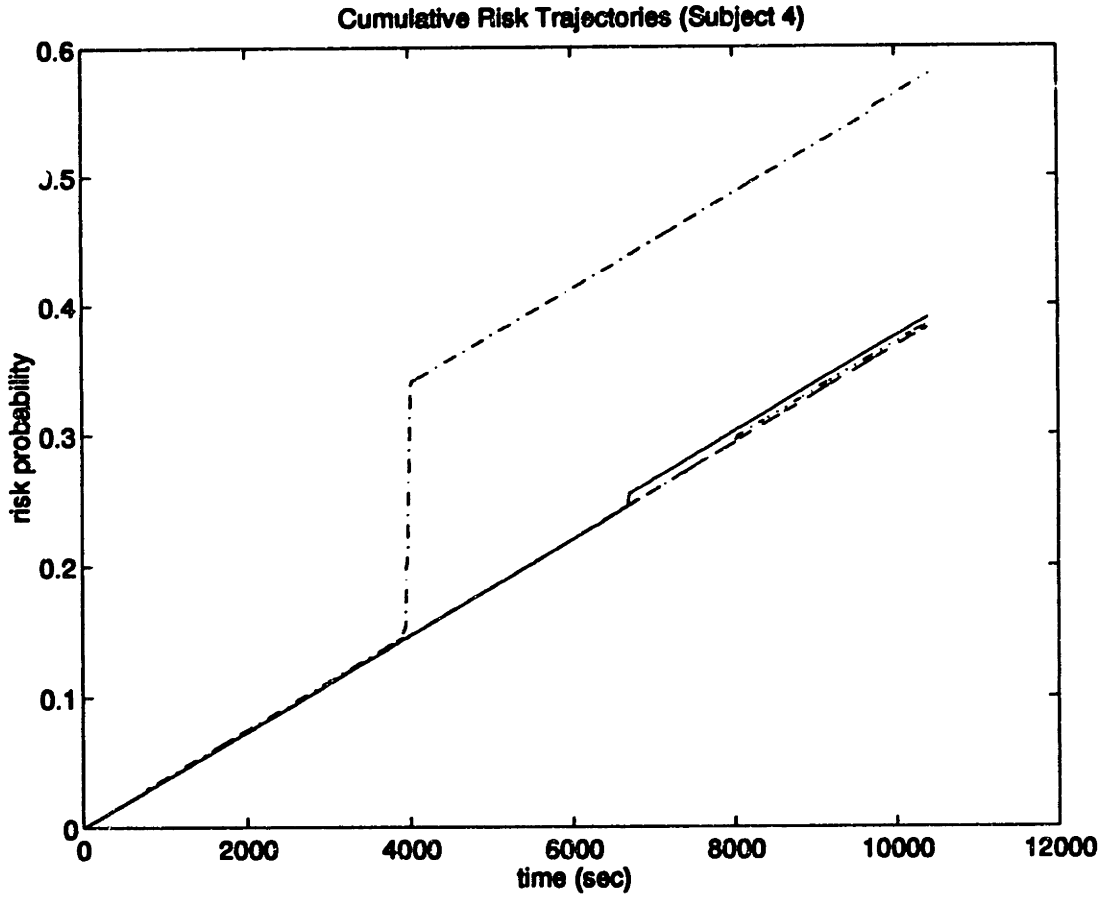

Figure 5-28: Cumulative Risk Trajectories—Subject 4

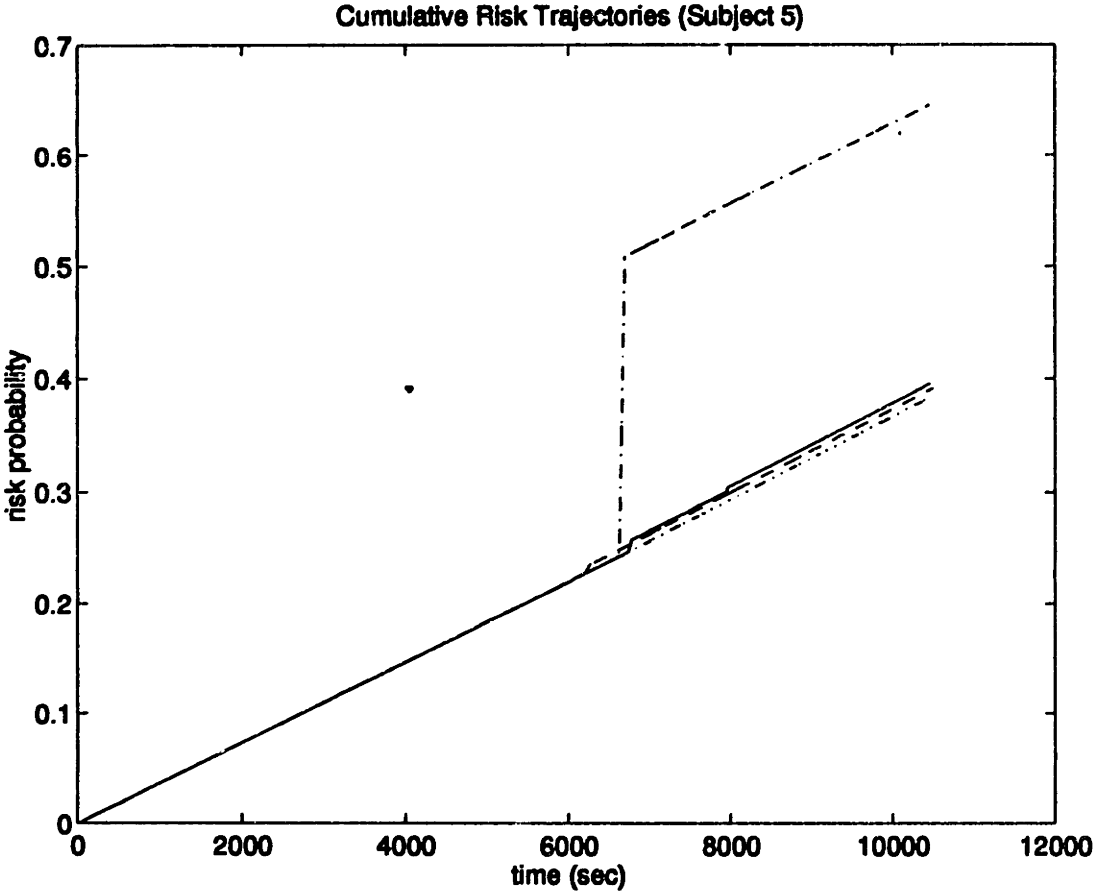

Figure 5-29: Cumulative Risk Trajectories—Subject 5

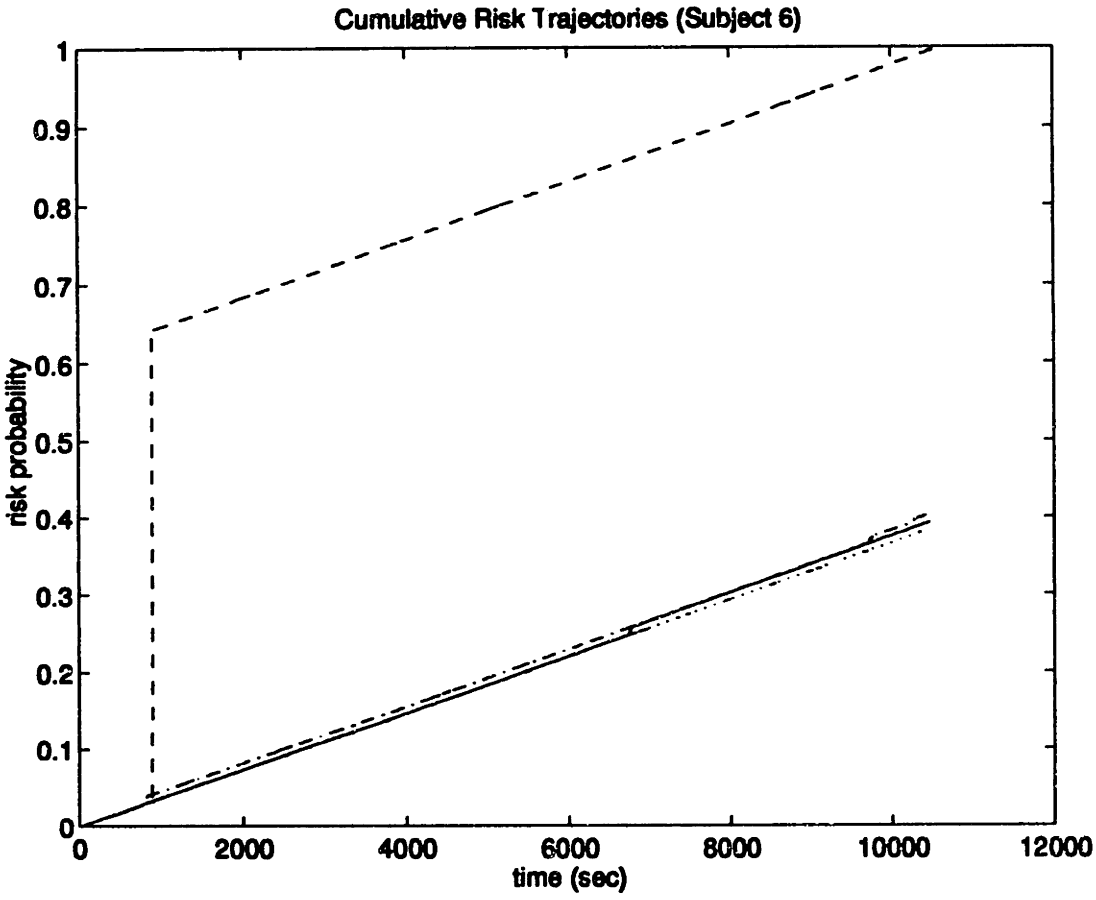

Figure 5-30: Cumulative Risk Trajectories—Subject 6

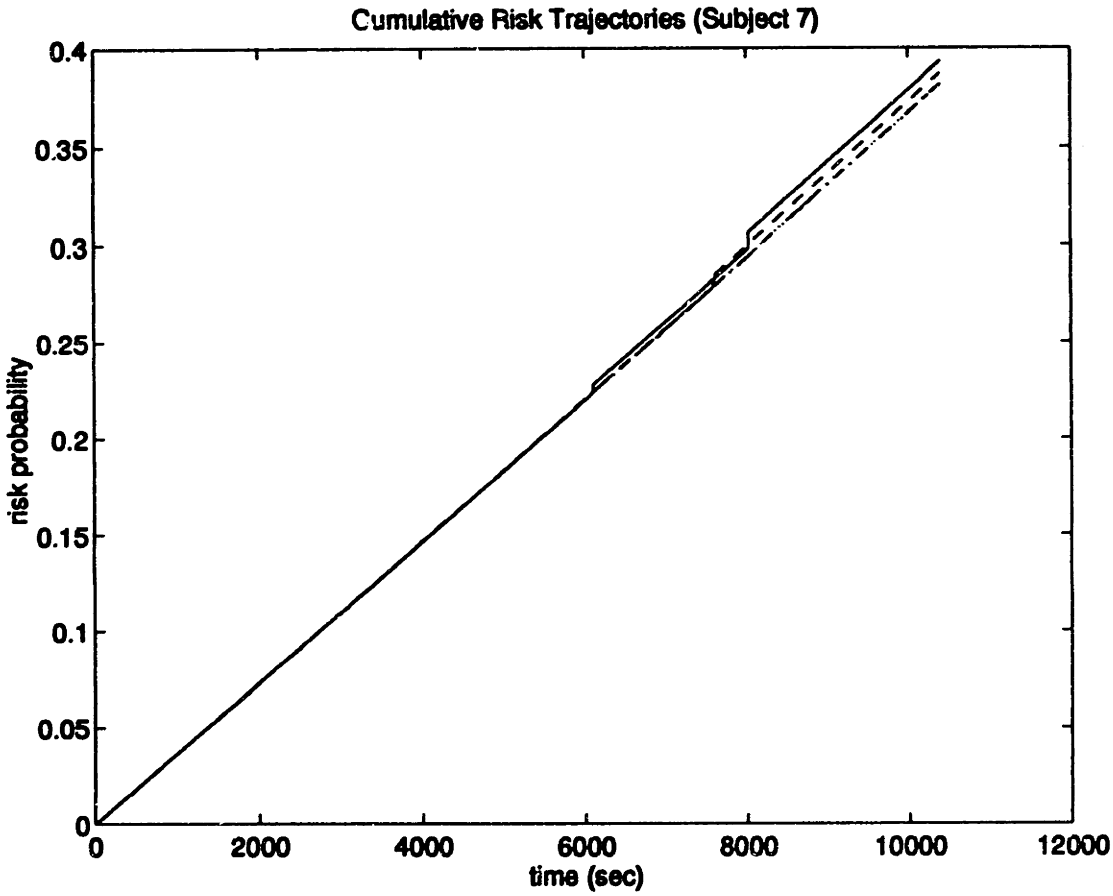

Figure 5-31: Cumulative Risk Trajectories—Subject 7

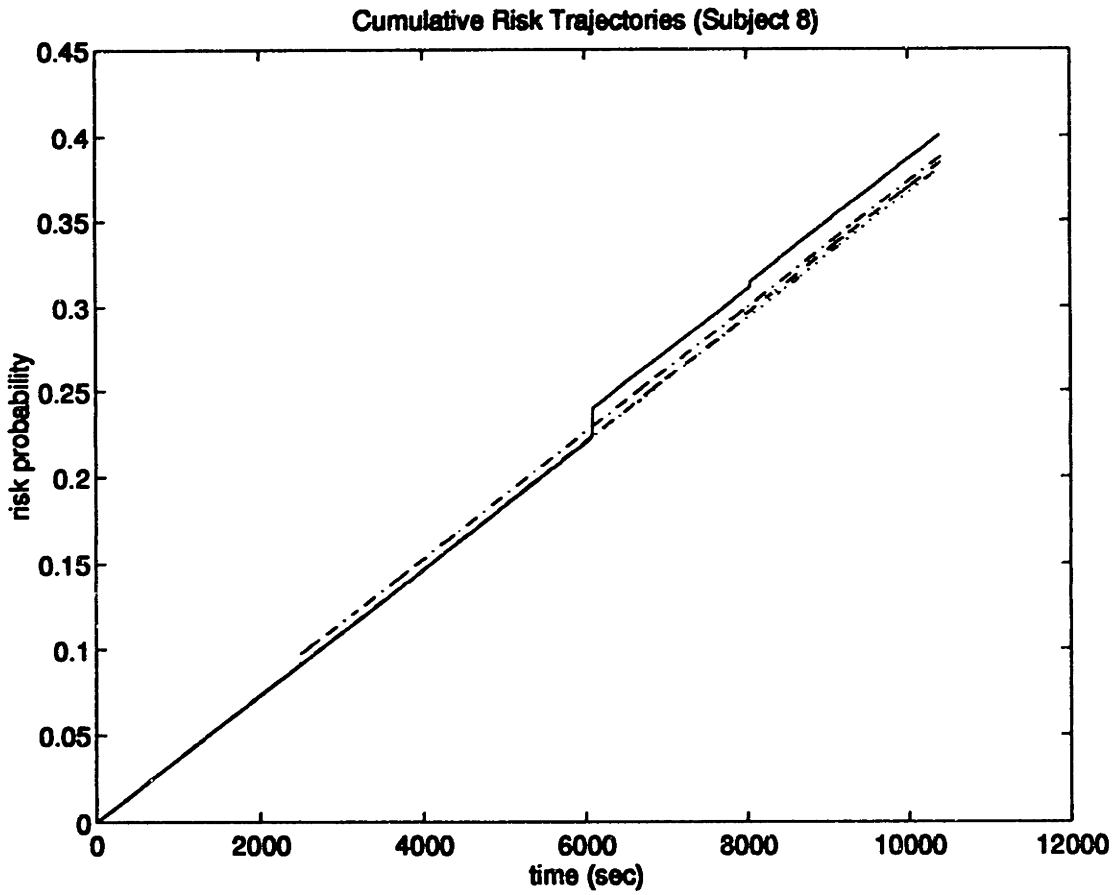

Figure 5-32: Cumulative Risk Trajectories—Subject 8

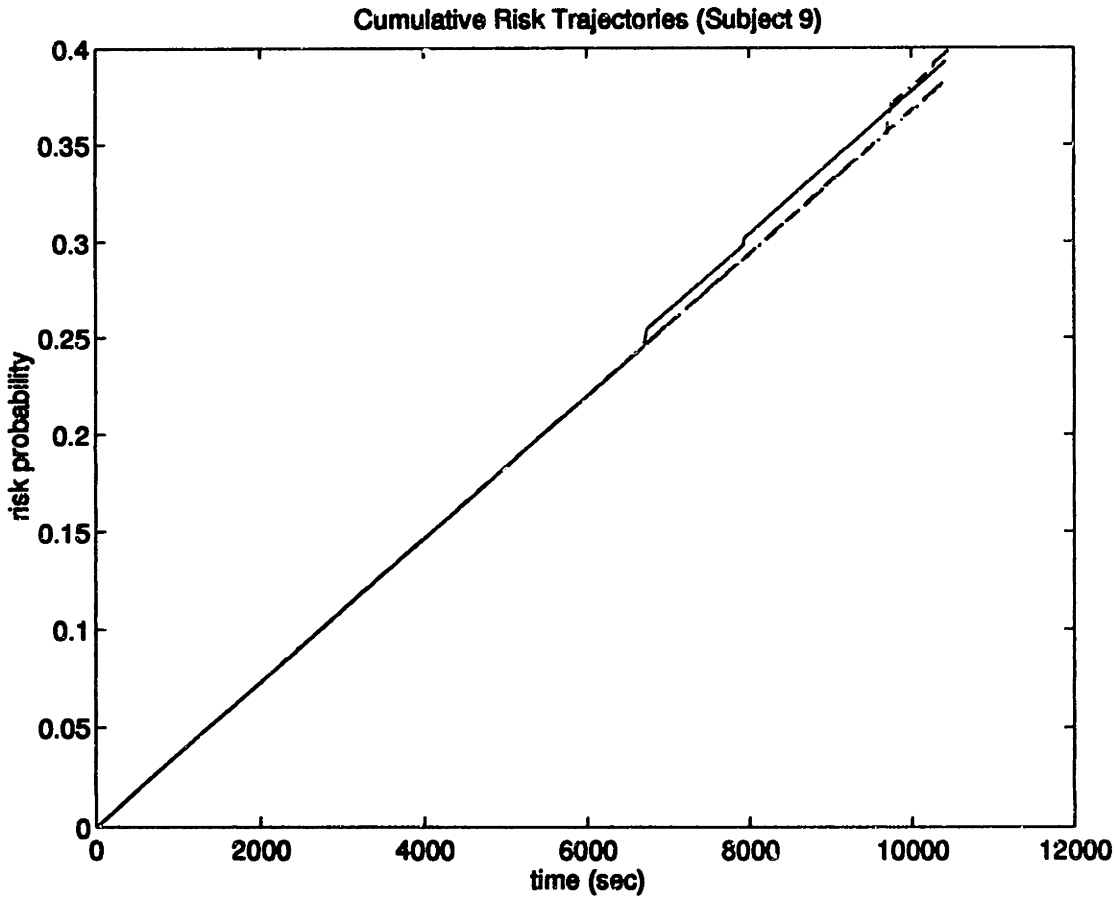

Figure 5-33: Cumulative Risk Trajectories—Subject 9

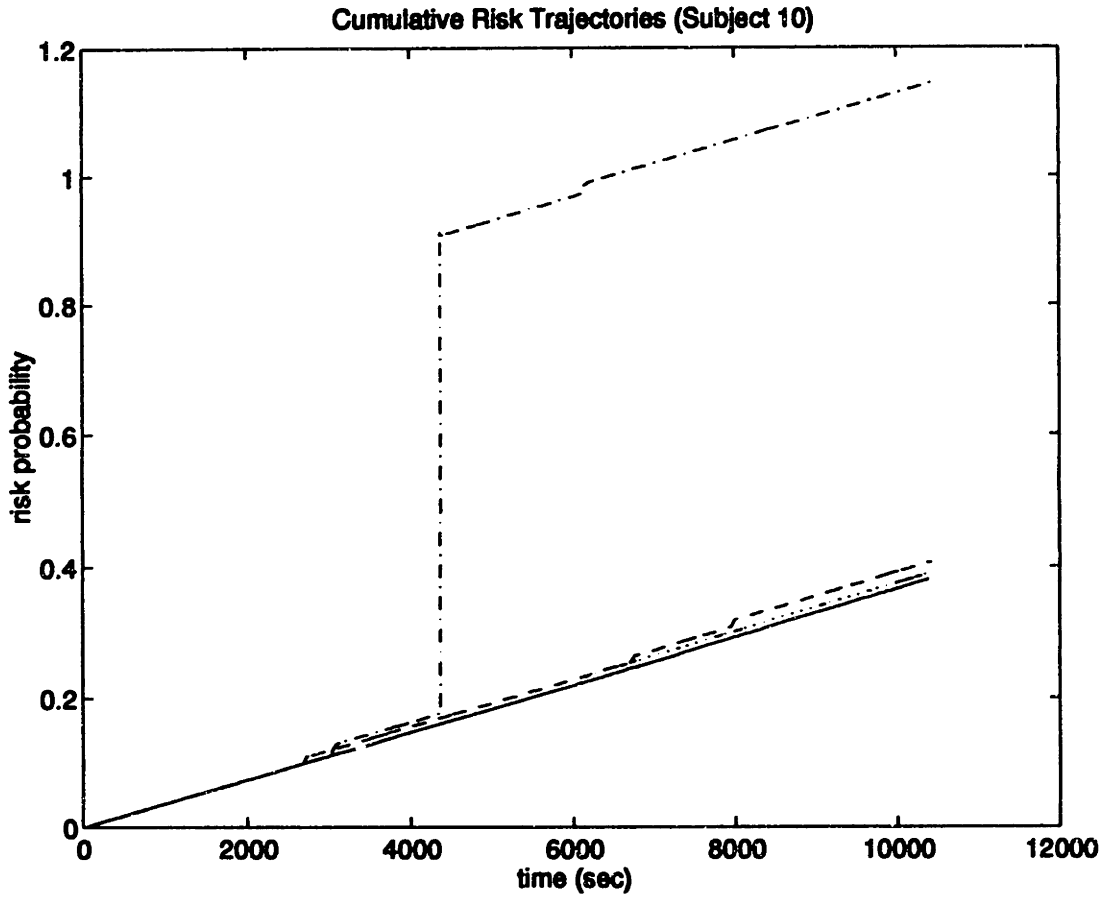

Figure 5-34: Cumulative Risk Trajectories—Subject 10

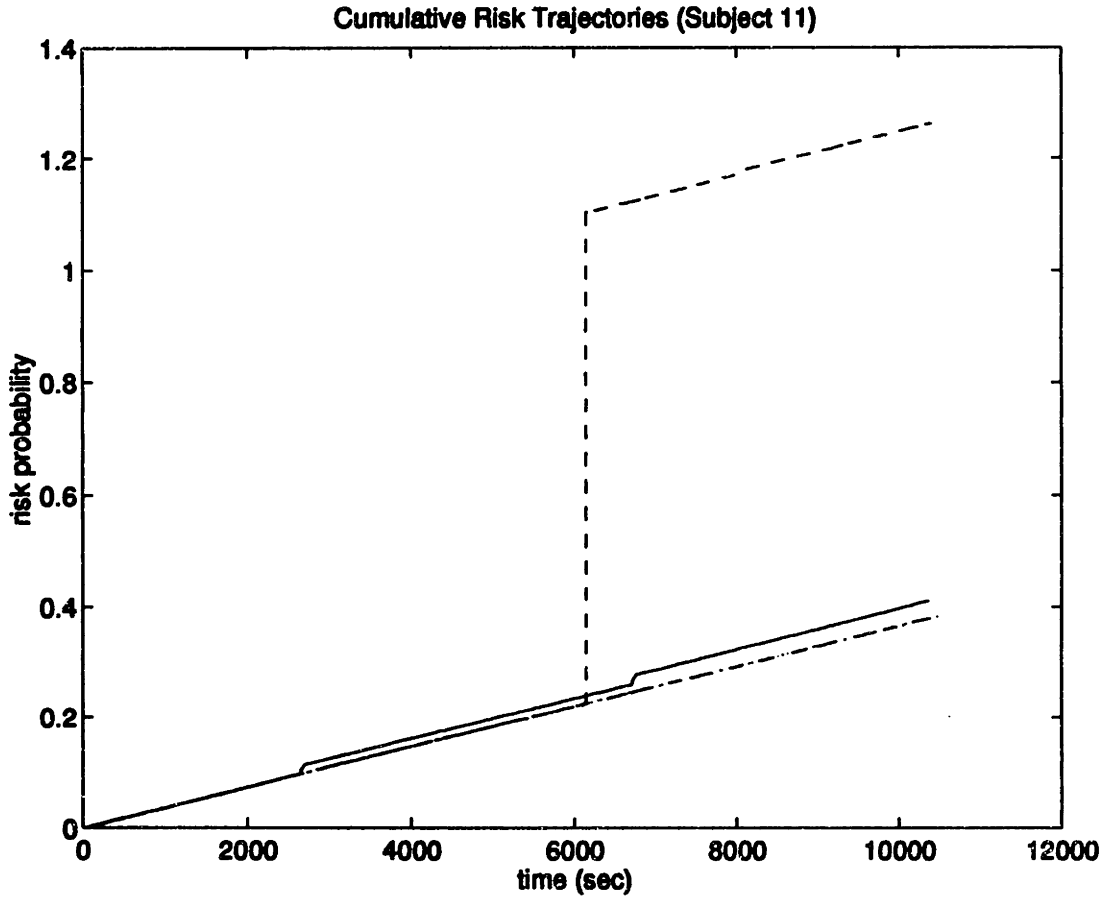

Figure 5-35: Cumulative Risk Trajectories—Subject 11

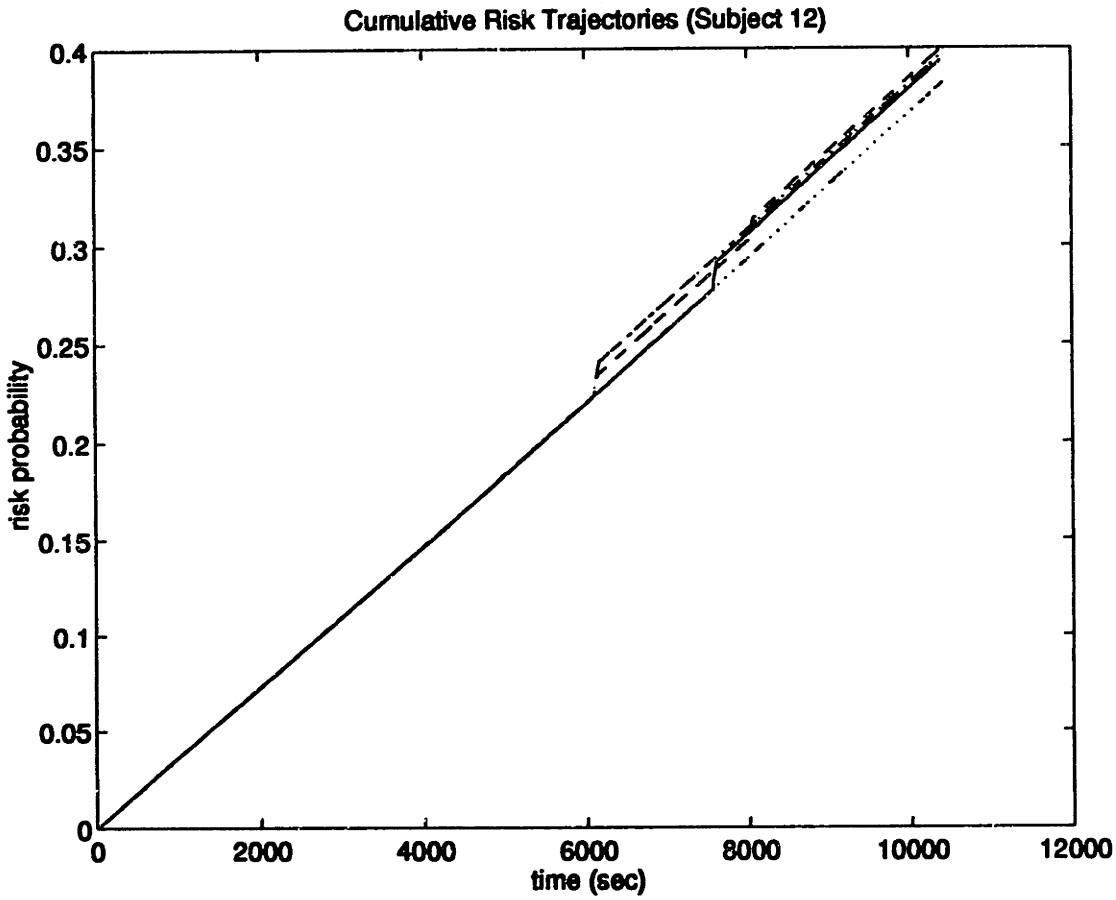

Figure 5-36: Cumulative Risk Trajectories—Subject 12

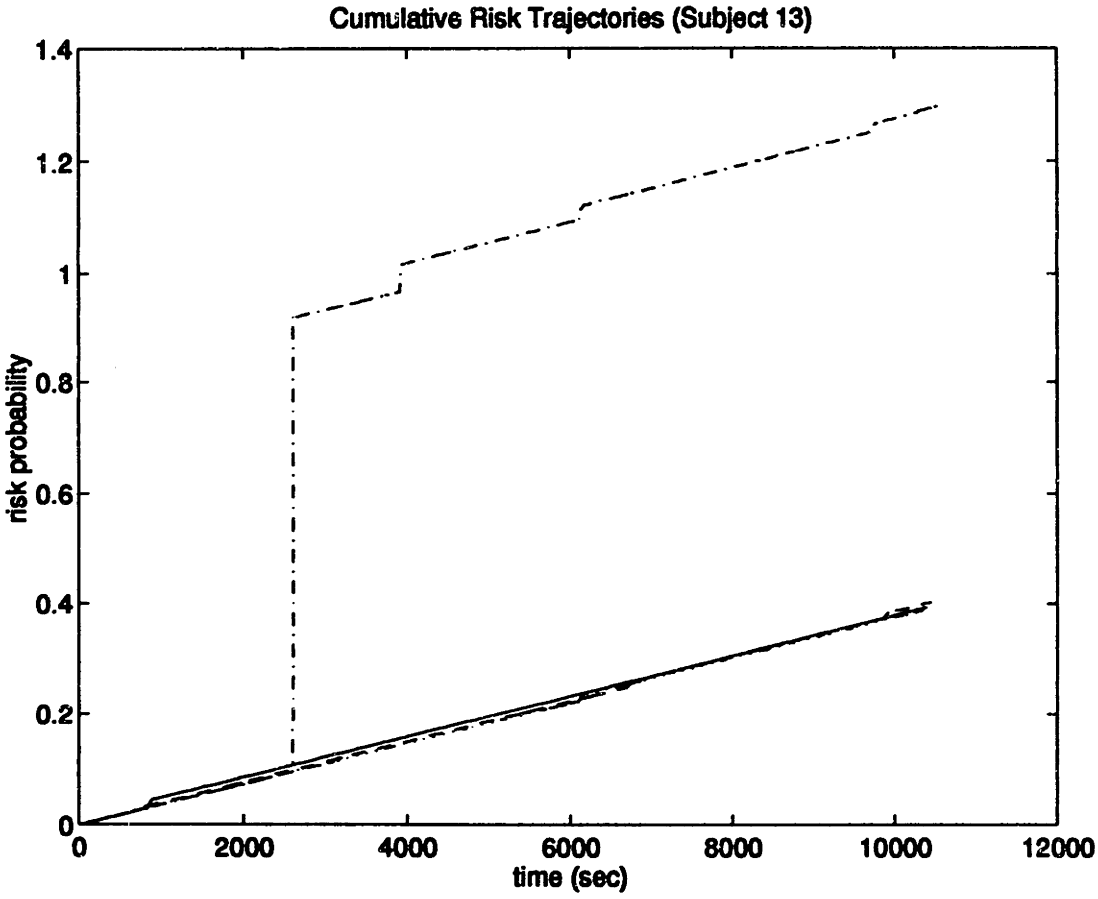

Figure 5-37: Cumulative Risk Trajectories—Subject 13

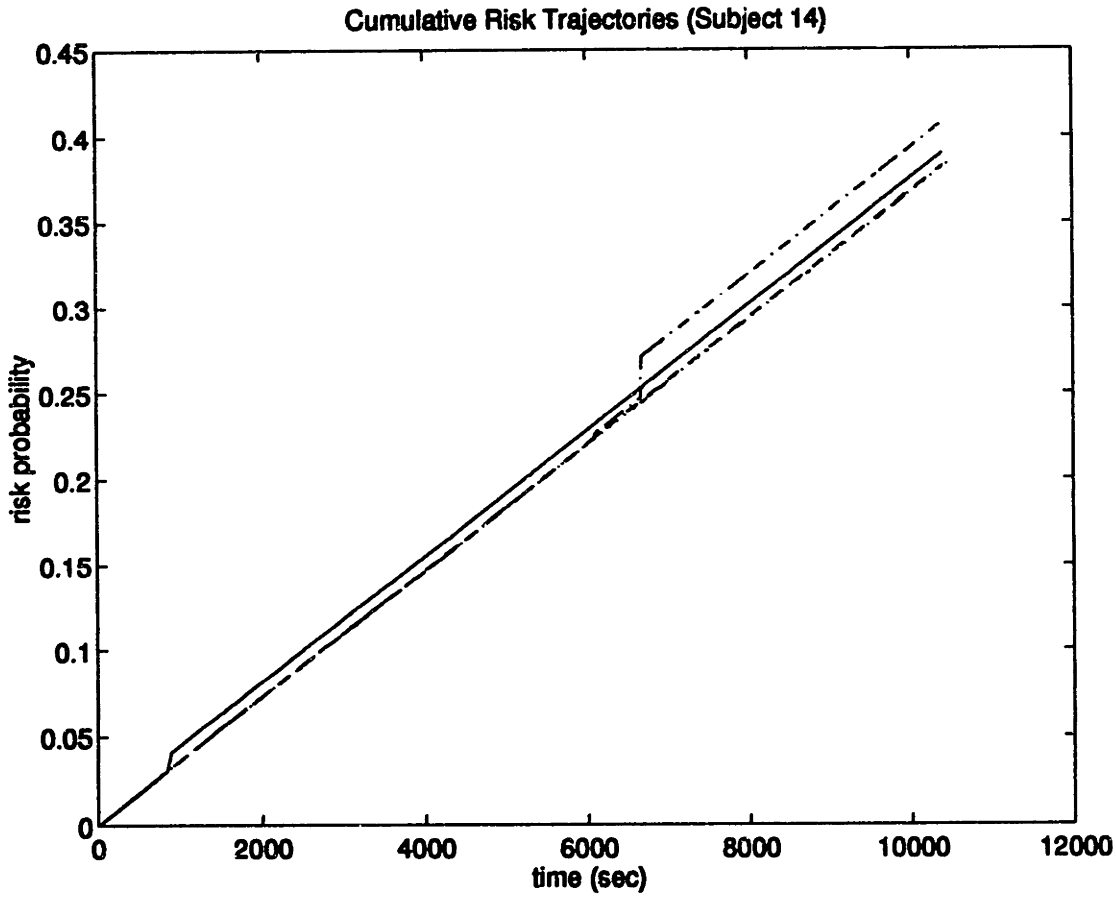

Figure 5-38: Cumulative Risk Trajectories—Subject 14

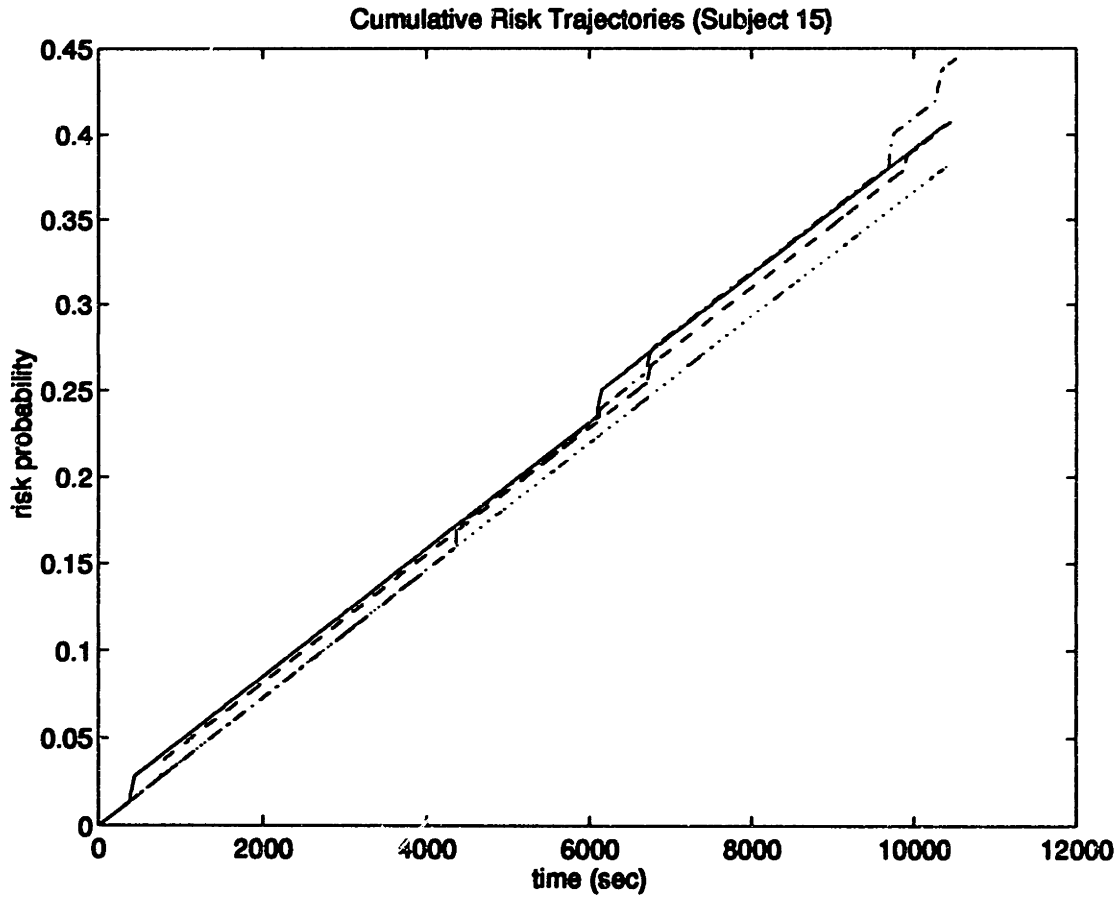

Figure 5-39: Cumulative Risk Trajectories--Subject 15

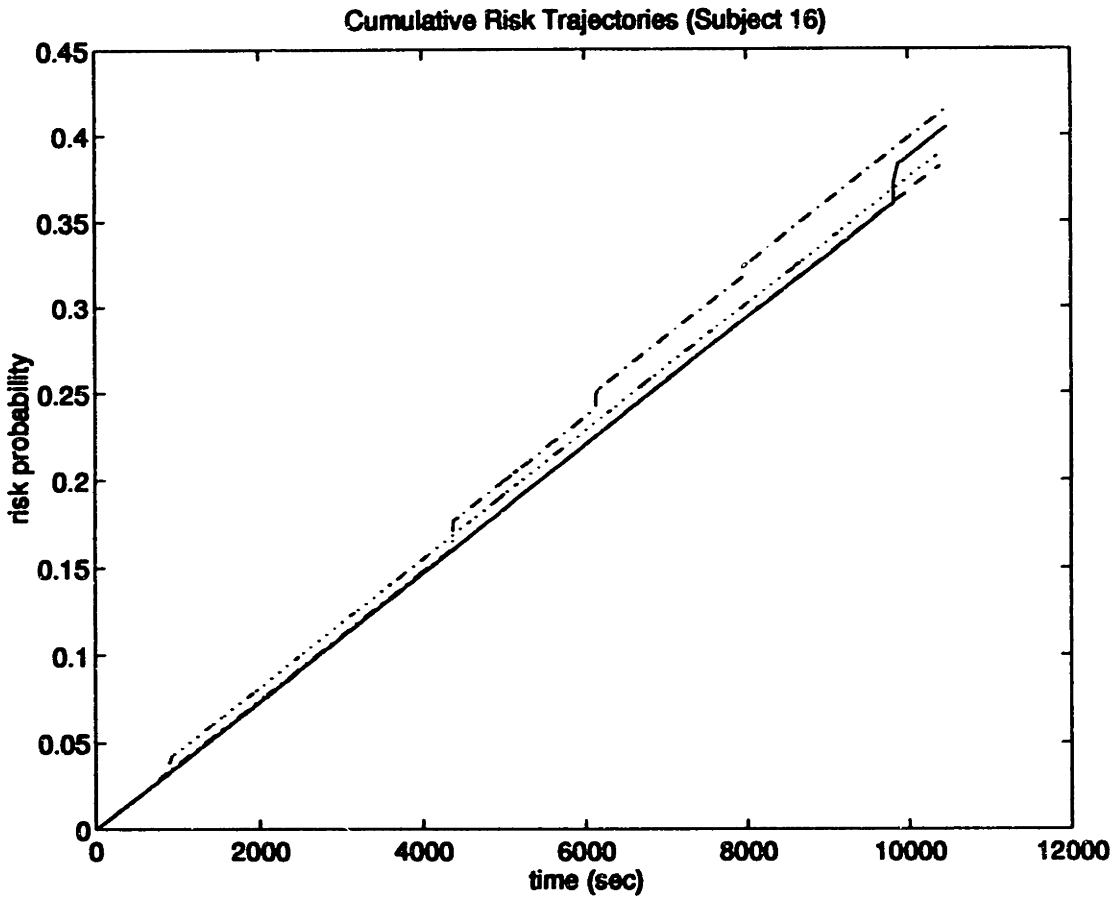

Figure 5-40: Cumulative Risk Trajectories—Subject 16

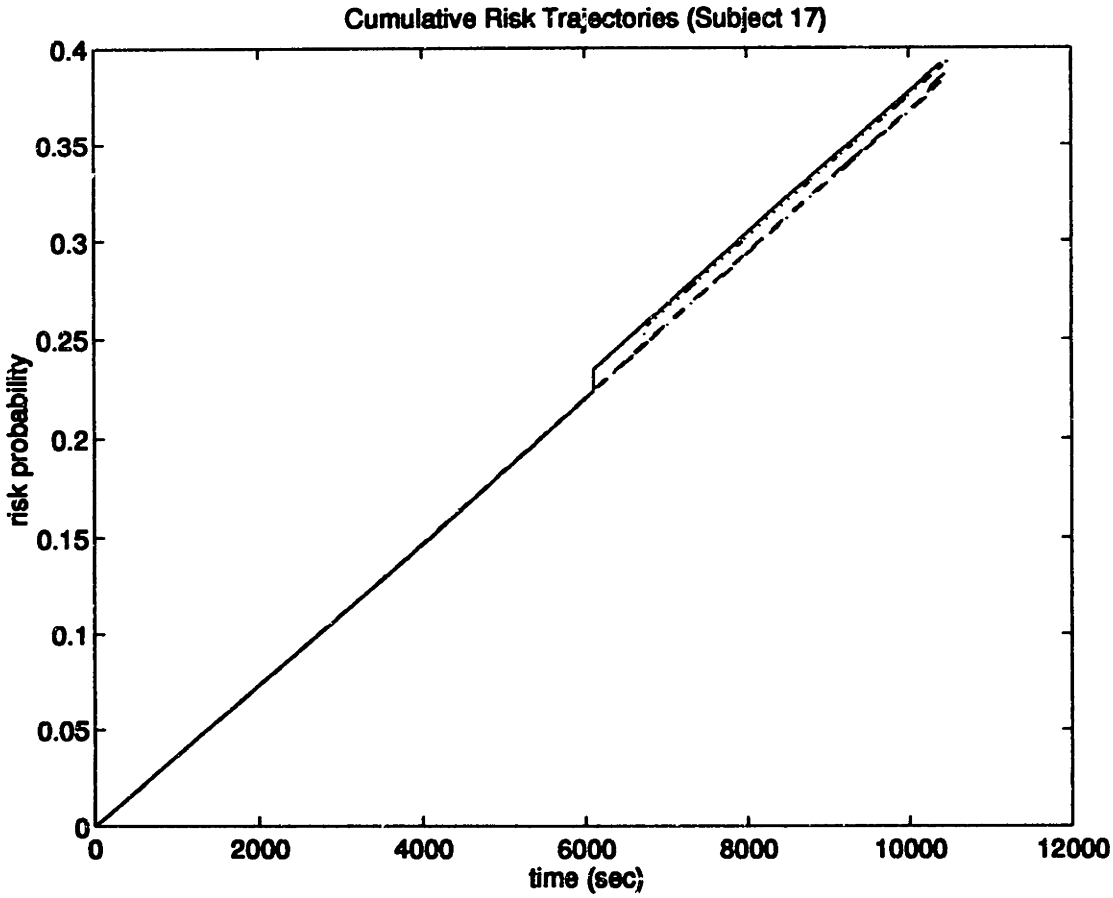

Figure 5-41: Cumulative Risk Trajectories—Subject 17

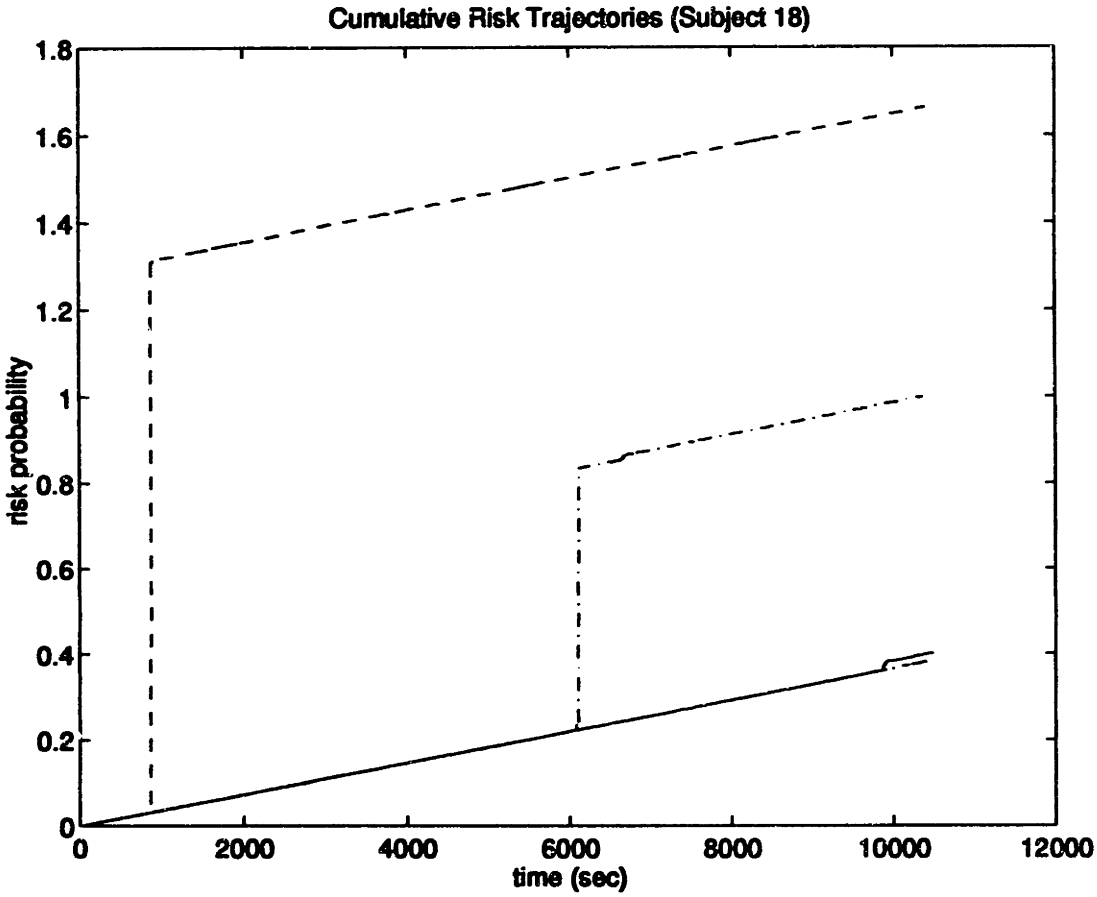

Figure 5-42: Cumulative Risk Trajectories—Subject 18

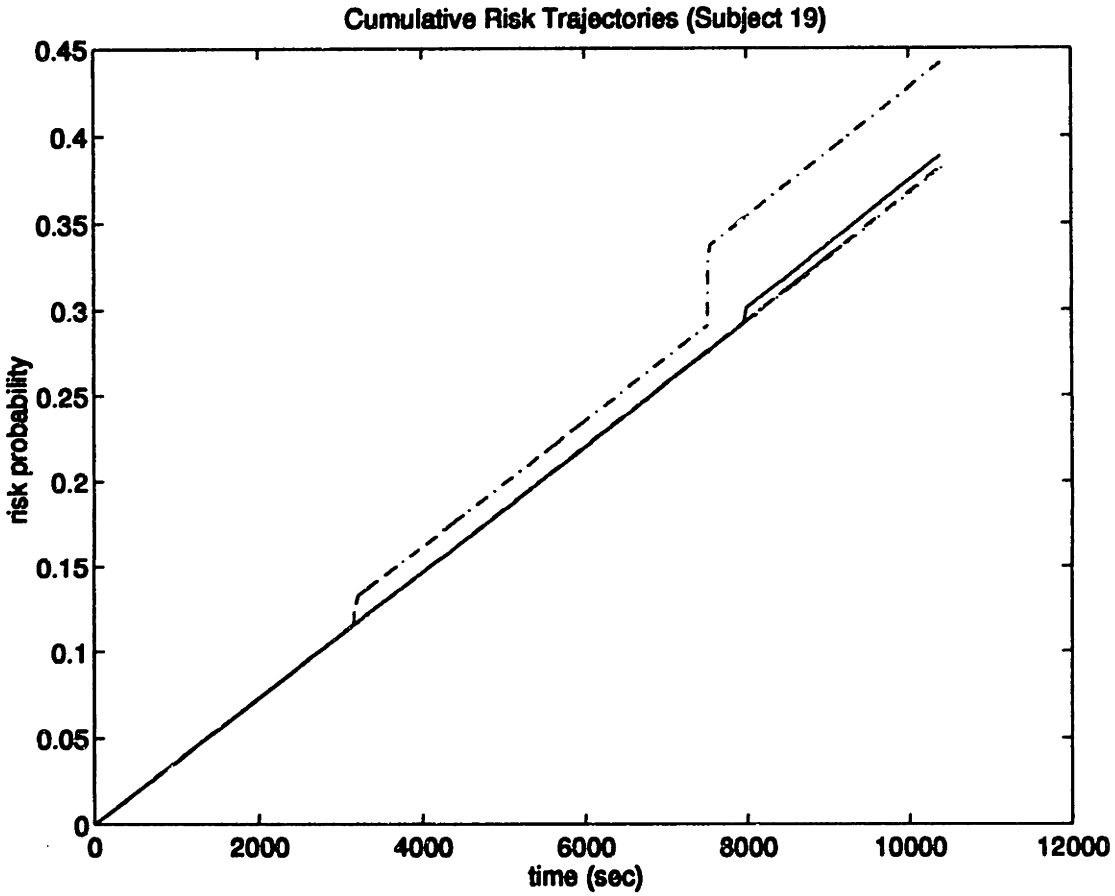

Figure 5-43: Cumulative Risk Trajectories—Subject 19

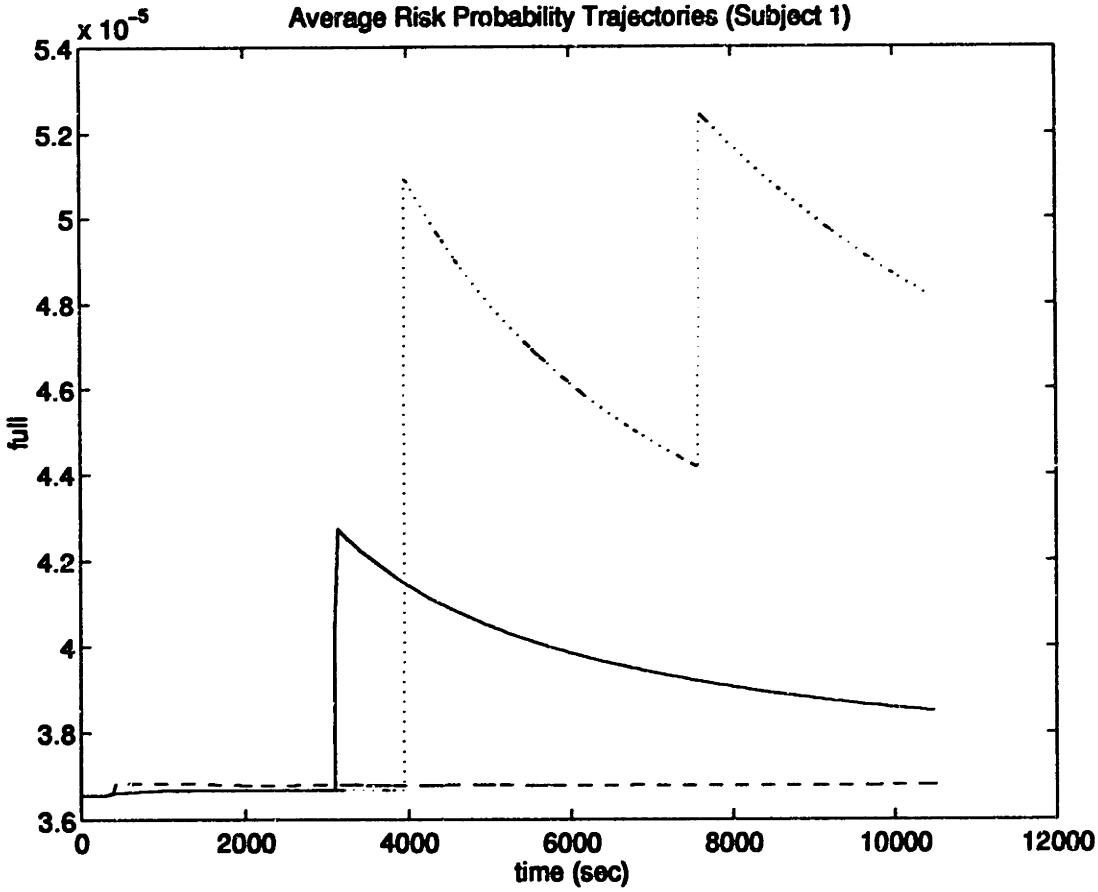

Figure 5-44: Average Risk Trajectories—Subject 1

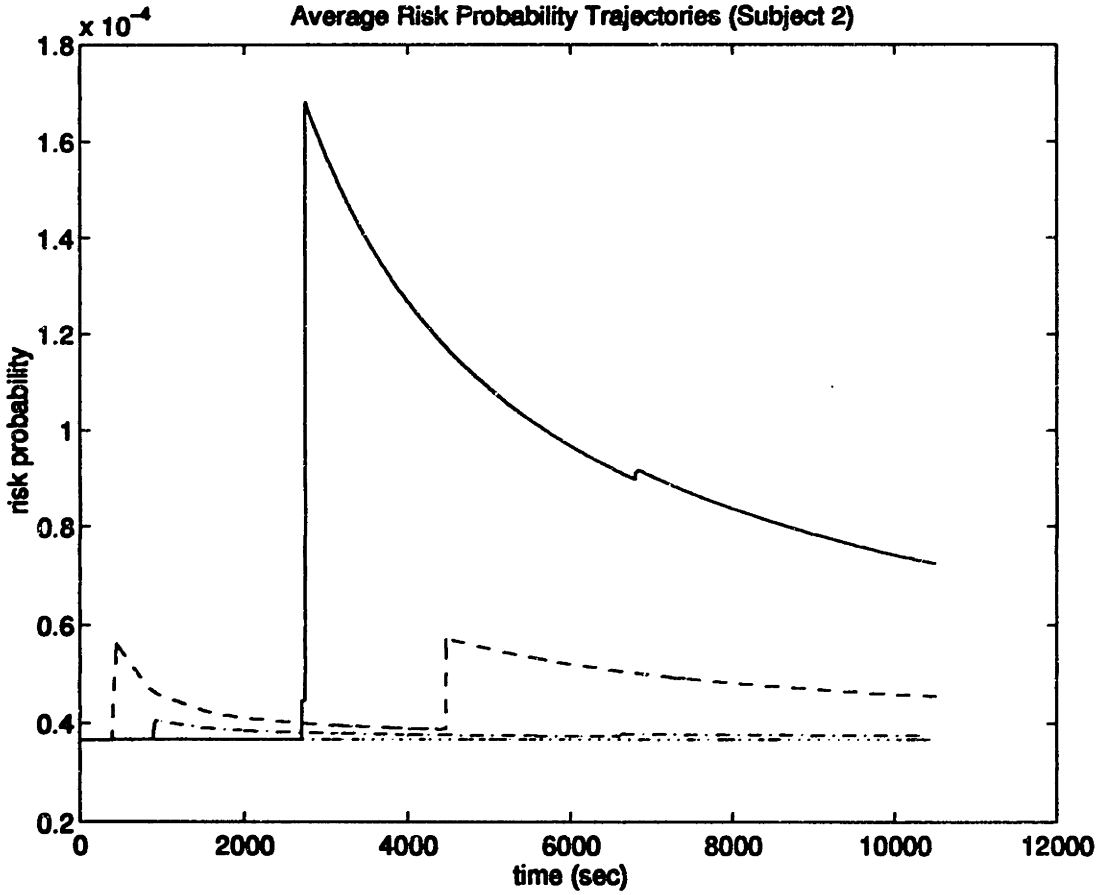

Figure 5-45: Average Risk Trajectories—Subject 2

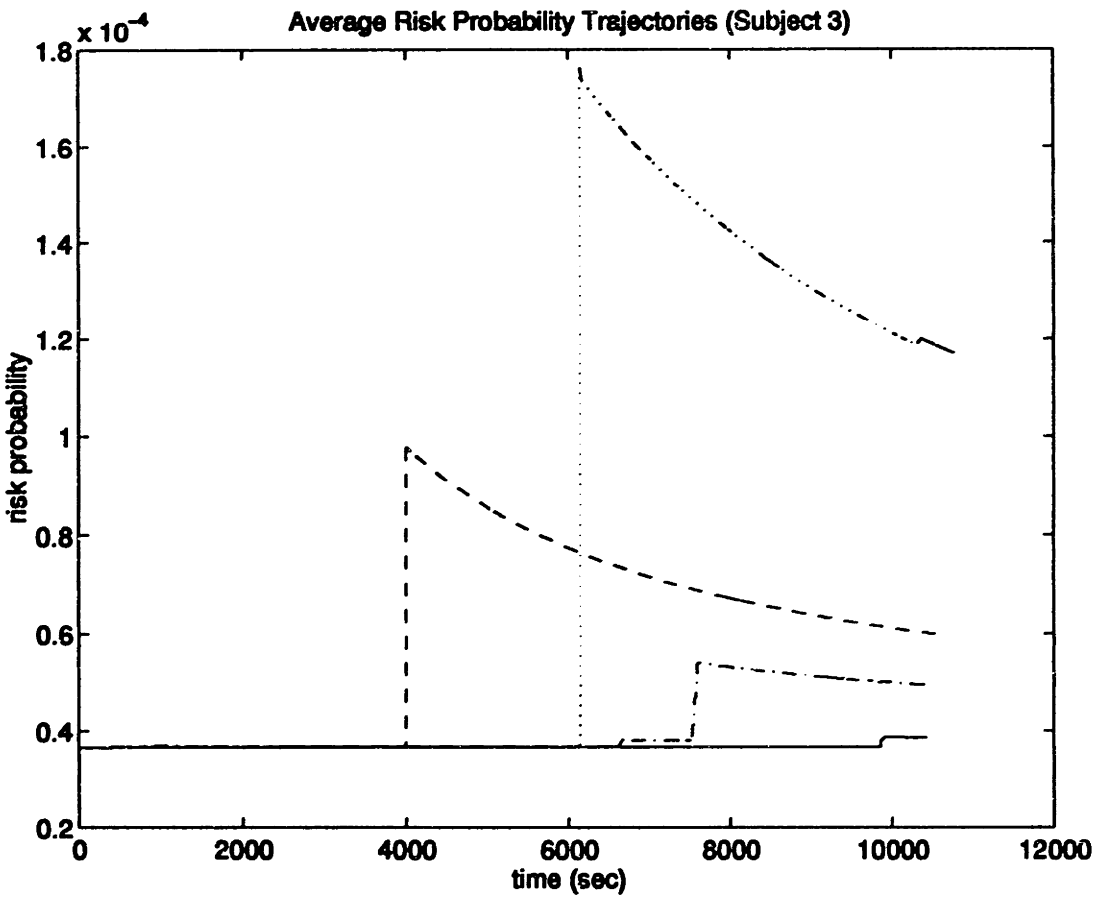

Figure 5-46: Average Risk Trajectories—Subject 3

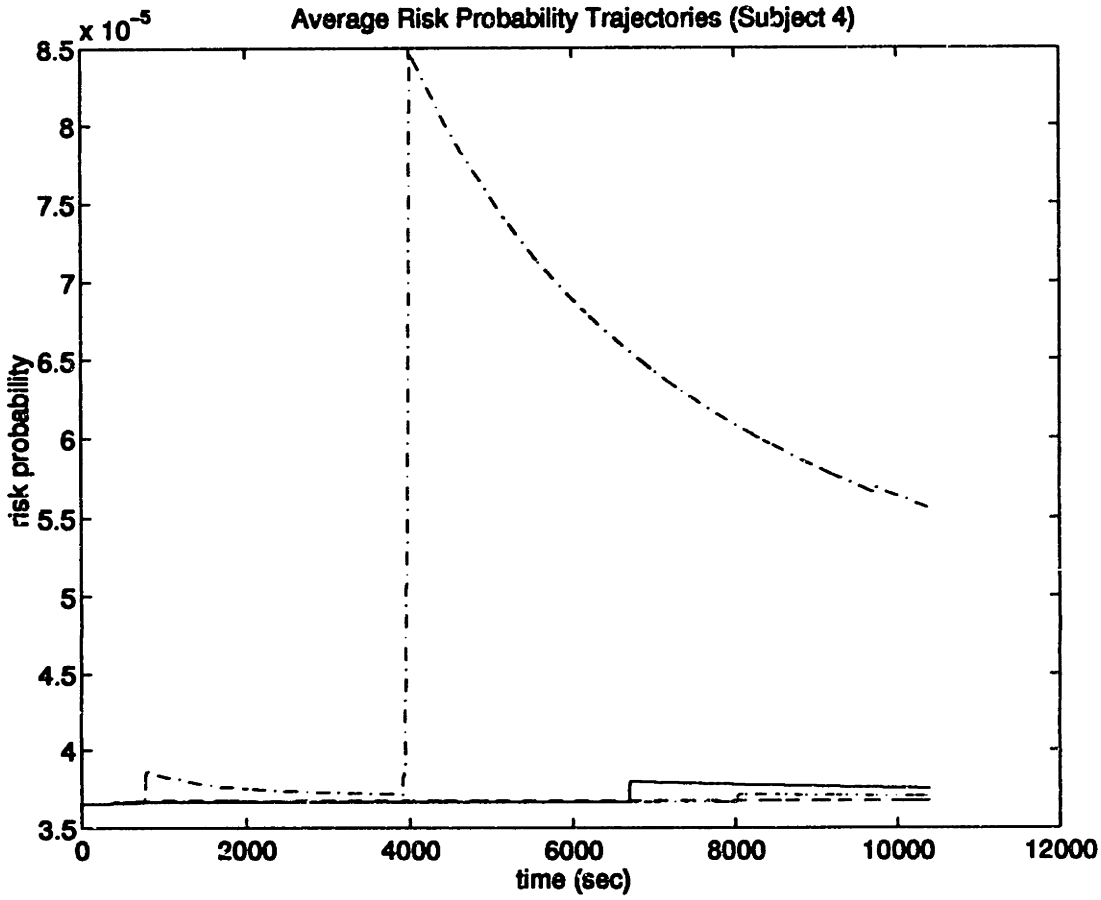

Figure 5-47: Average Risk Trajectories—Subject 4

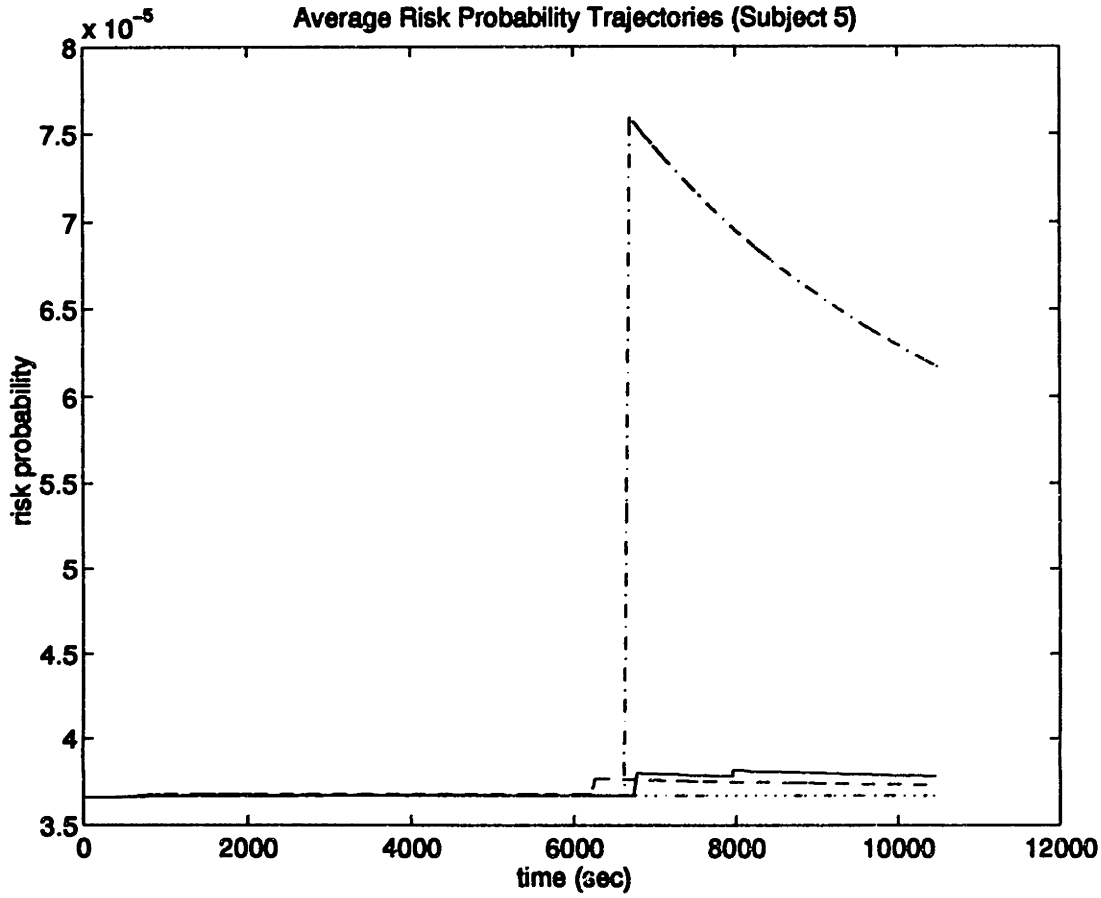

Figure 5-48: Average Risk Trajectories—Subject 5

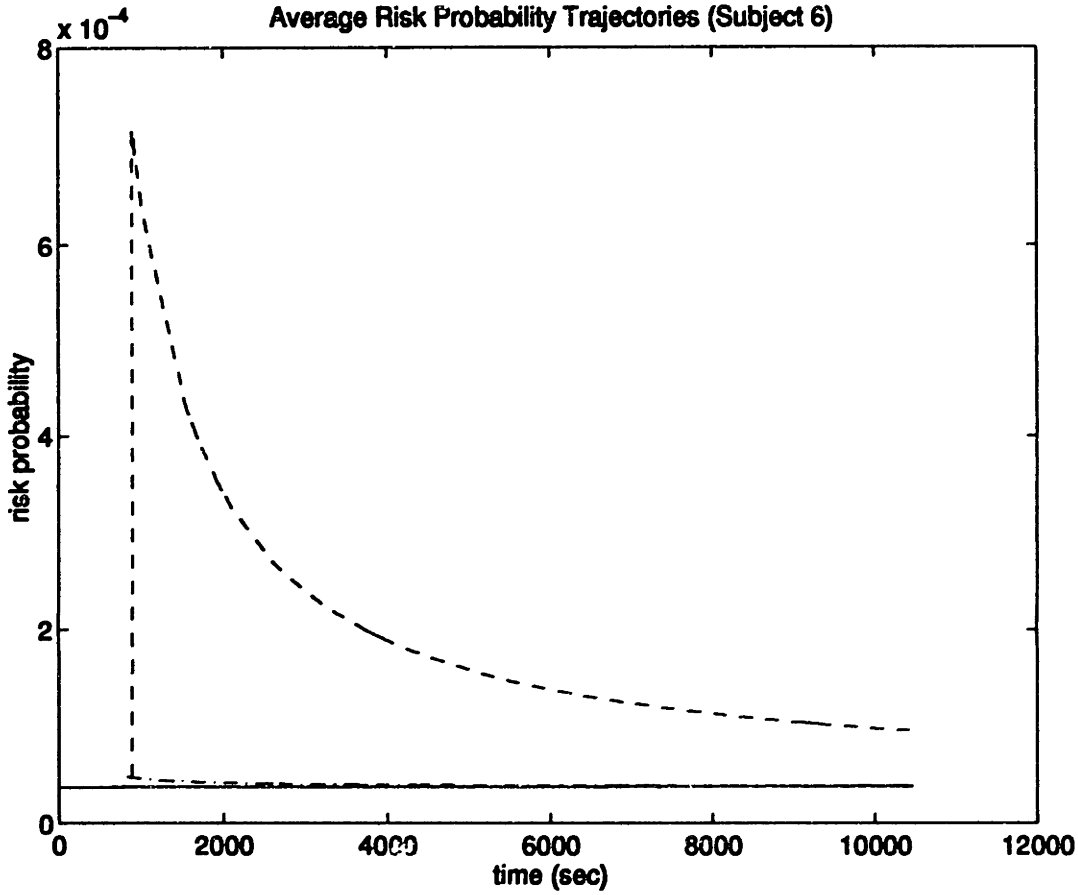

Figure 5-49: Average Risk Trajectories—Subject 6

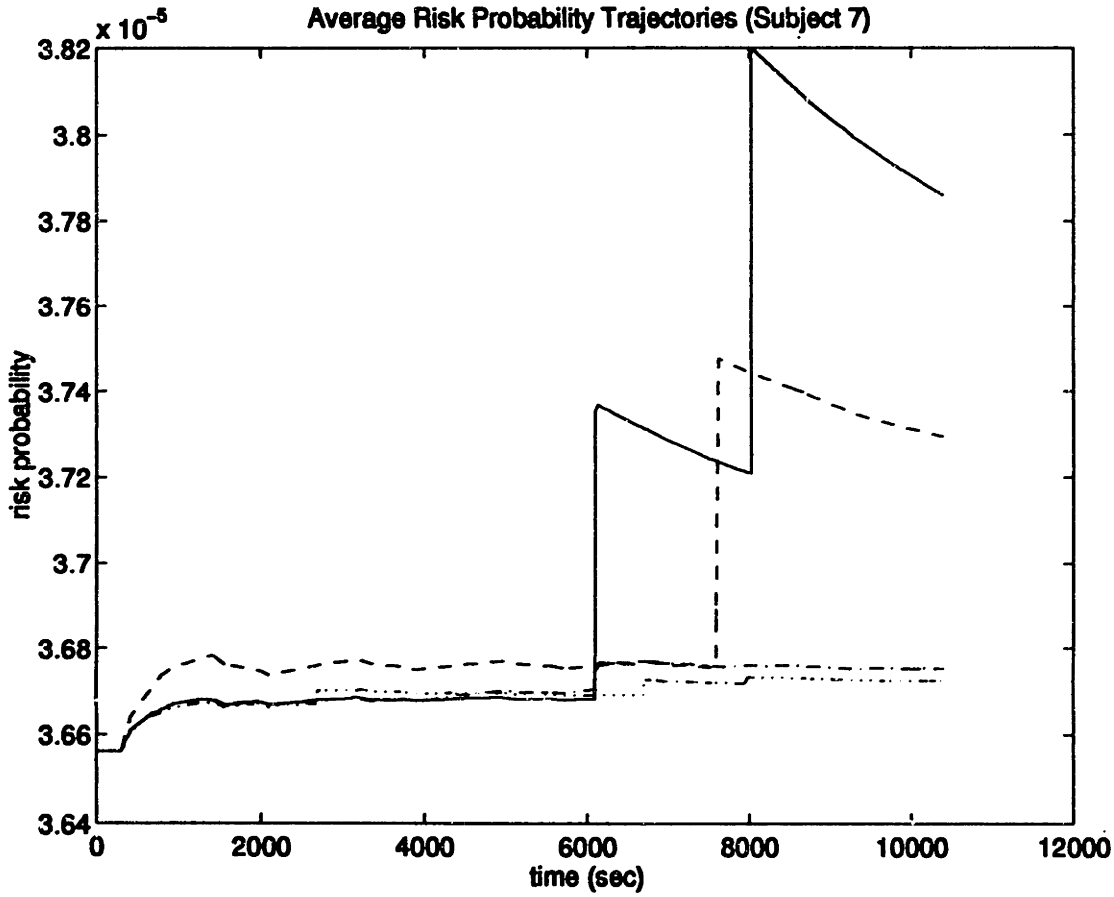

Figure 5-50: Average Risk Trajectories—Subject 7

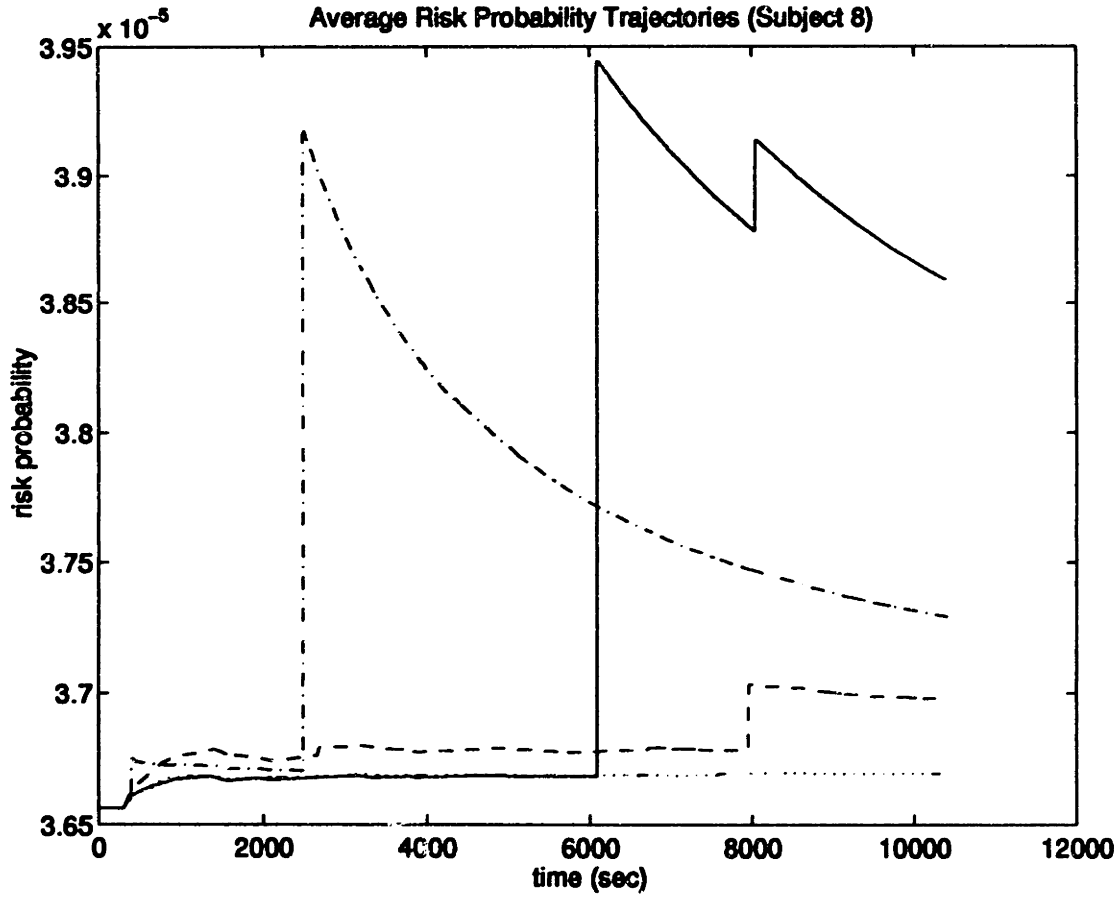

Figure 5-51: Average Risk Trajectories—Subject 8

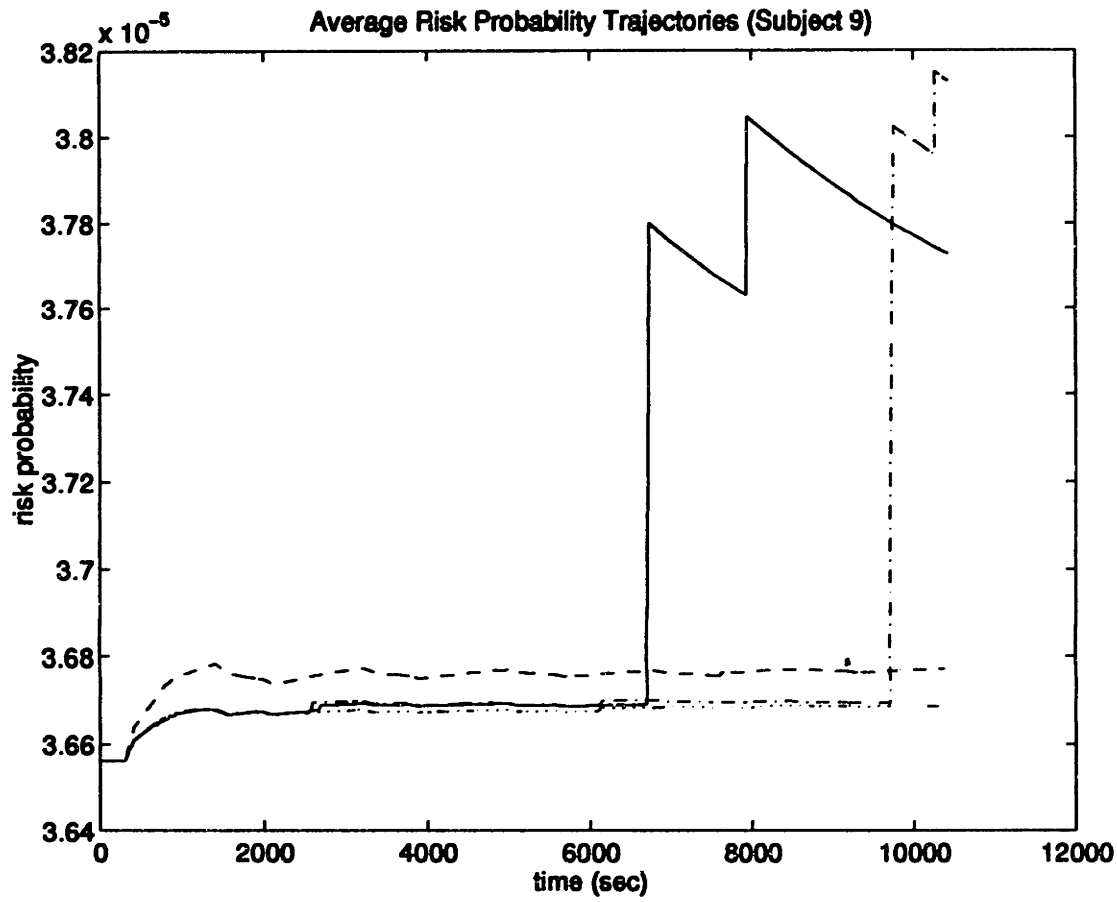

Figure 5-52: Average Risk Trajectories—Subject 9

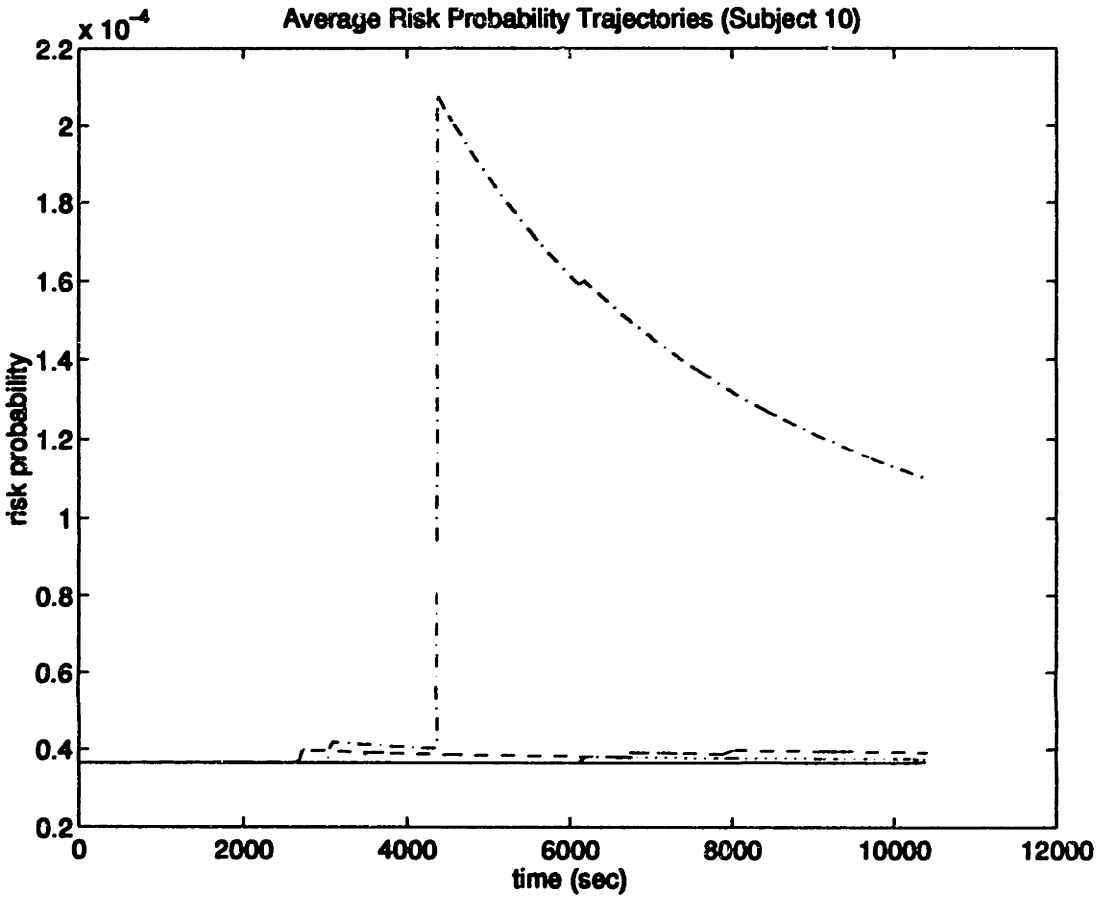

Figure 5-53: Average Risk Trajectories—Subject 10

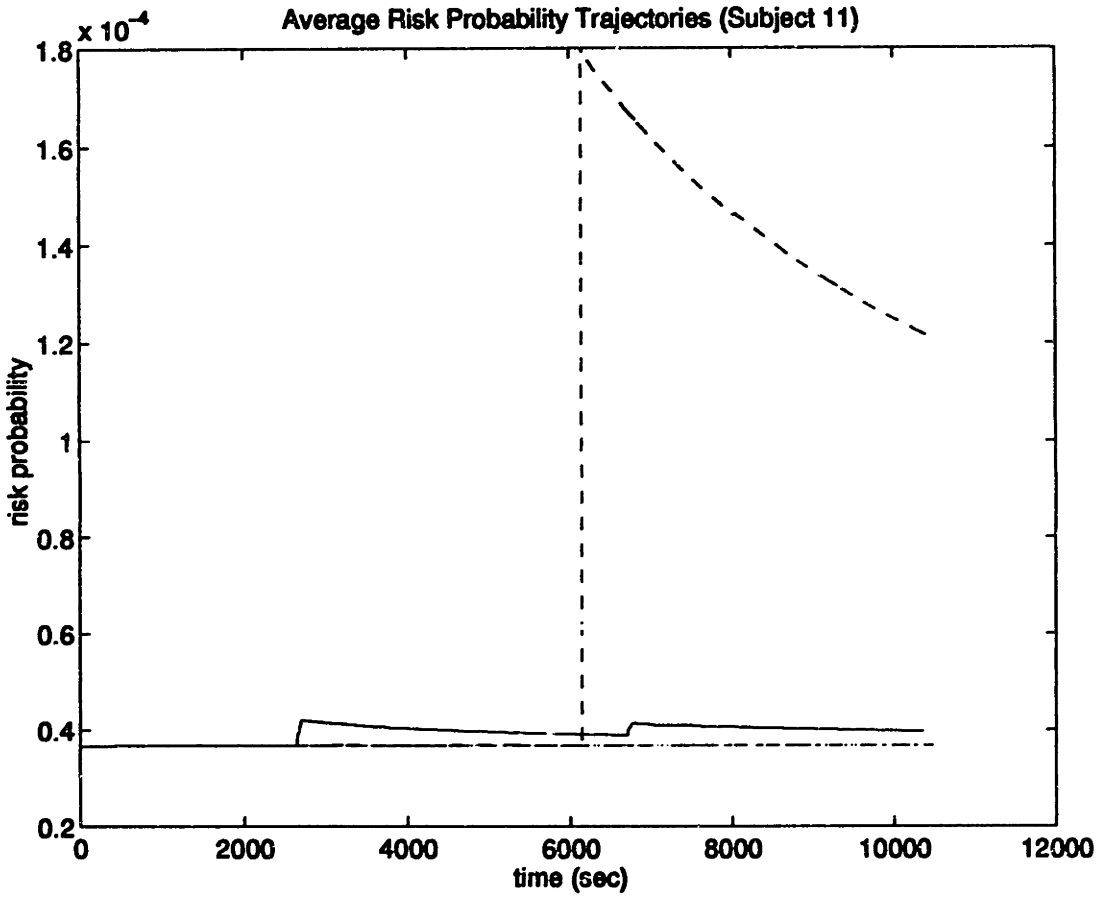

Figure 5-54: Average Risk Trajectories—Subject 11

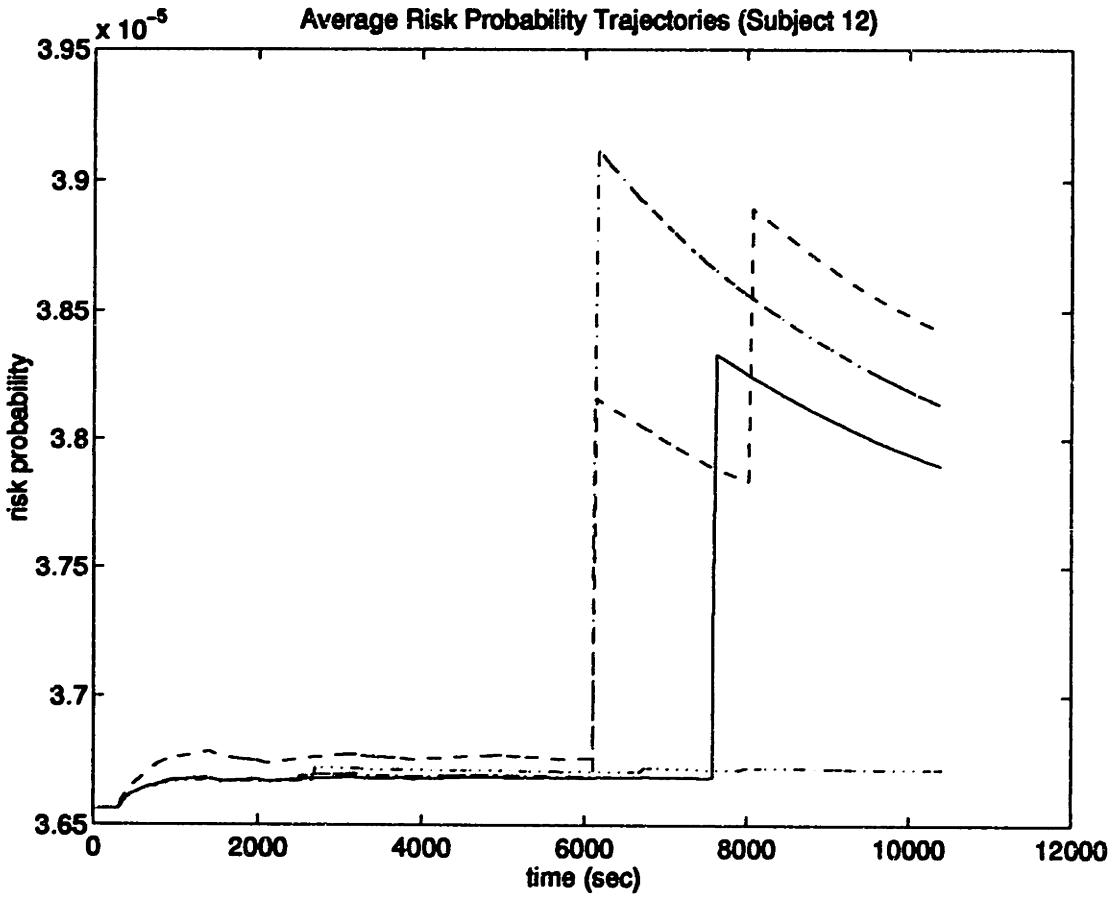

Figure 5-55: Average Risk Trajectories—Subject 12

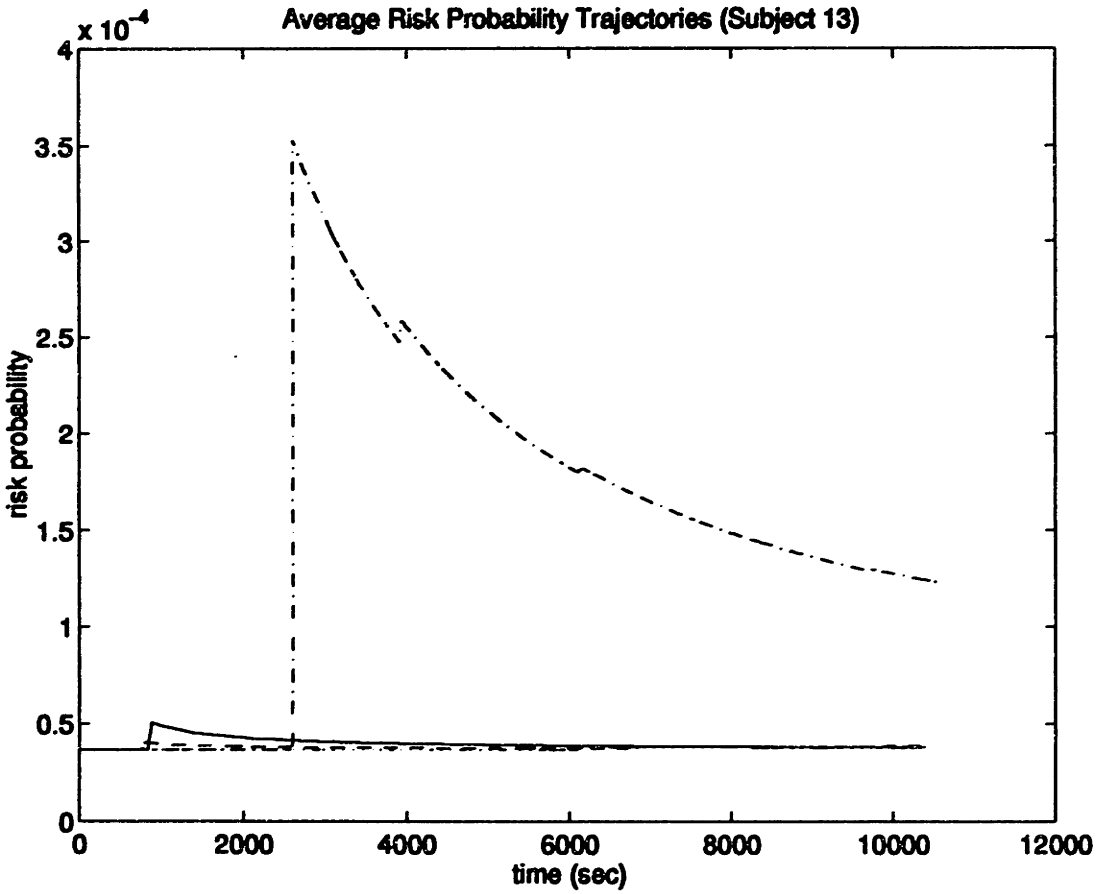

Figure 5-56: Average Risk Trajectories—Subject 13

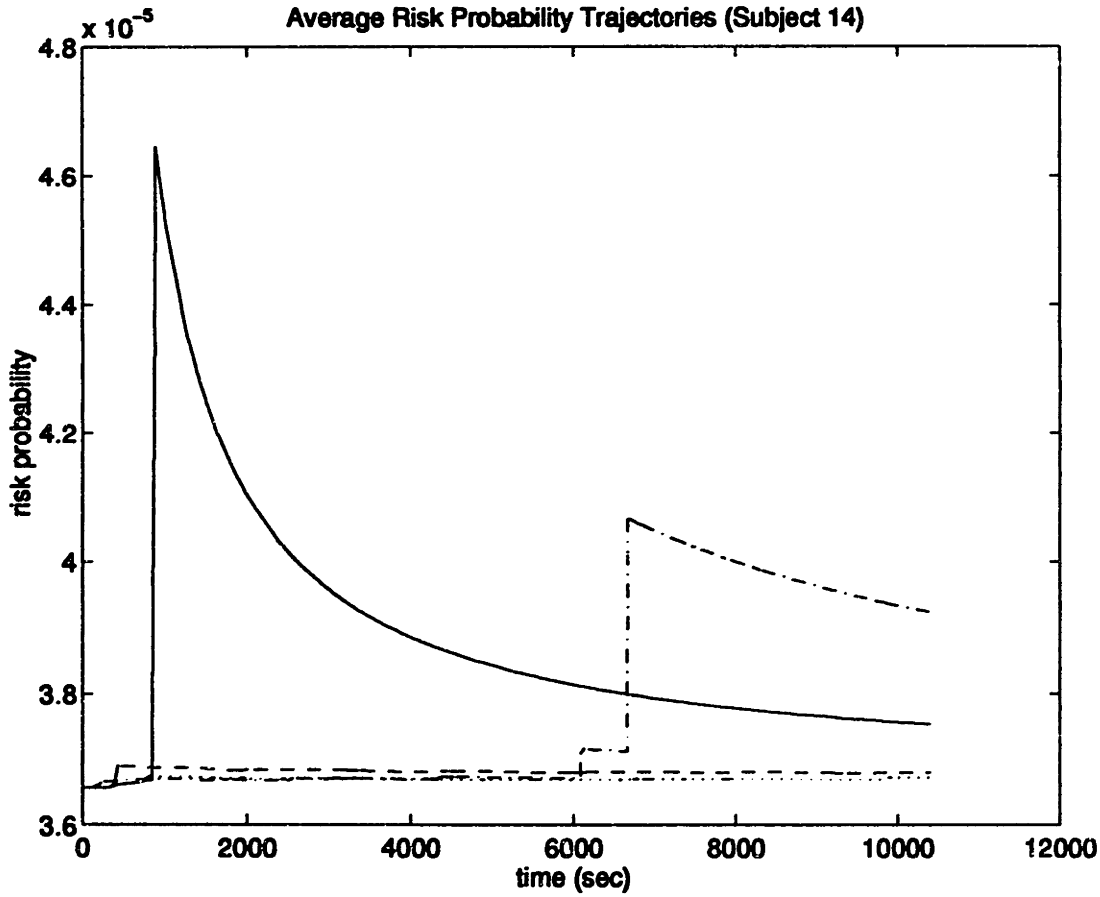

Figure 5-57: Average Risk Trajectories—Subject 14

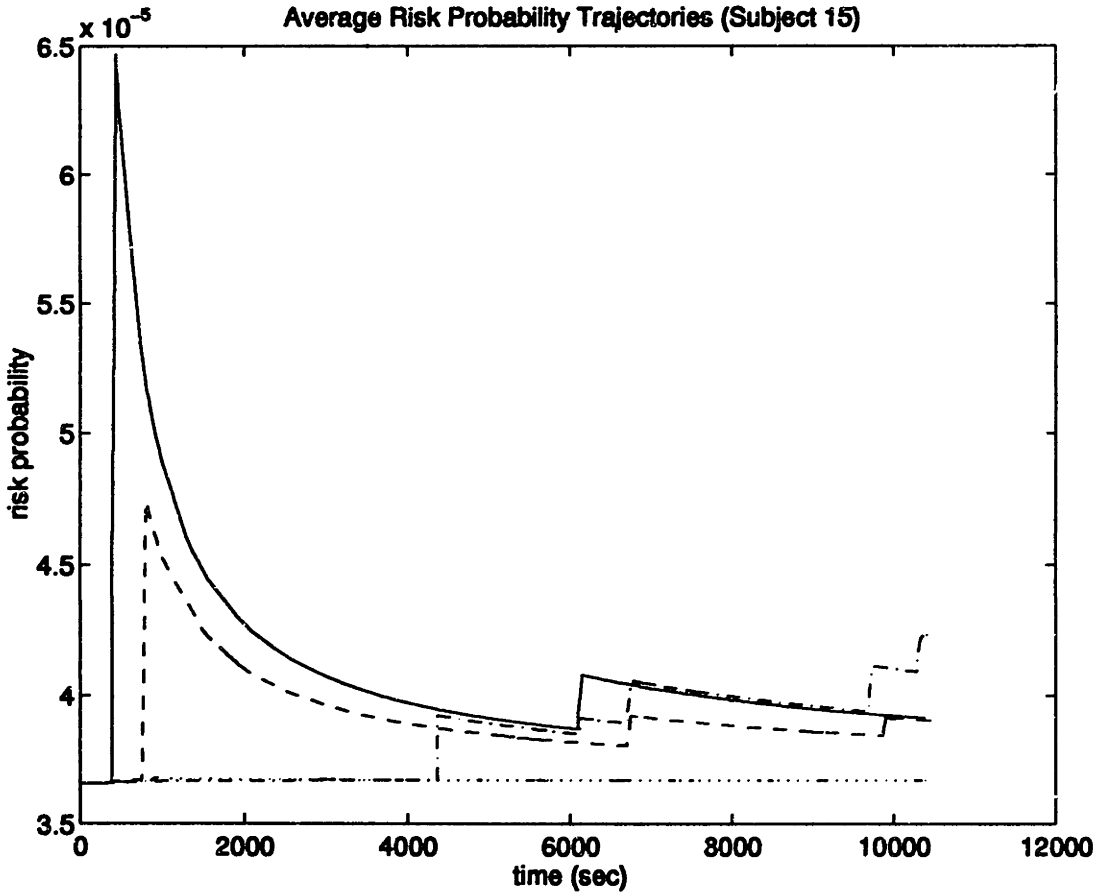

Figure 5-58: Average Risk Trajectories—Subject 15

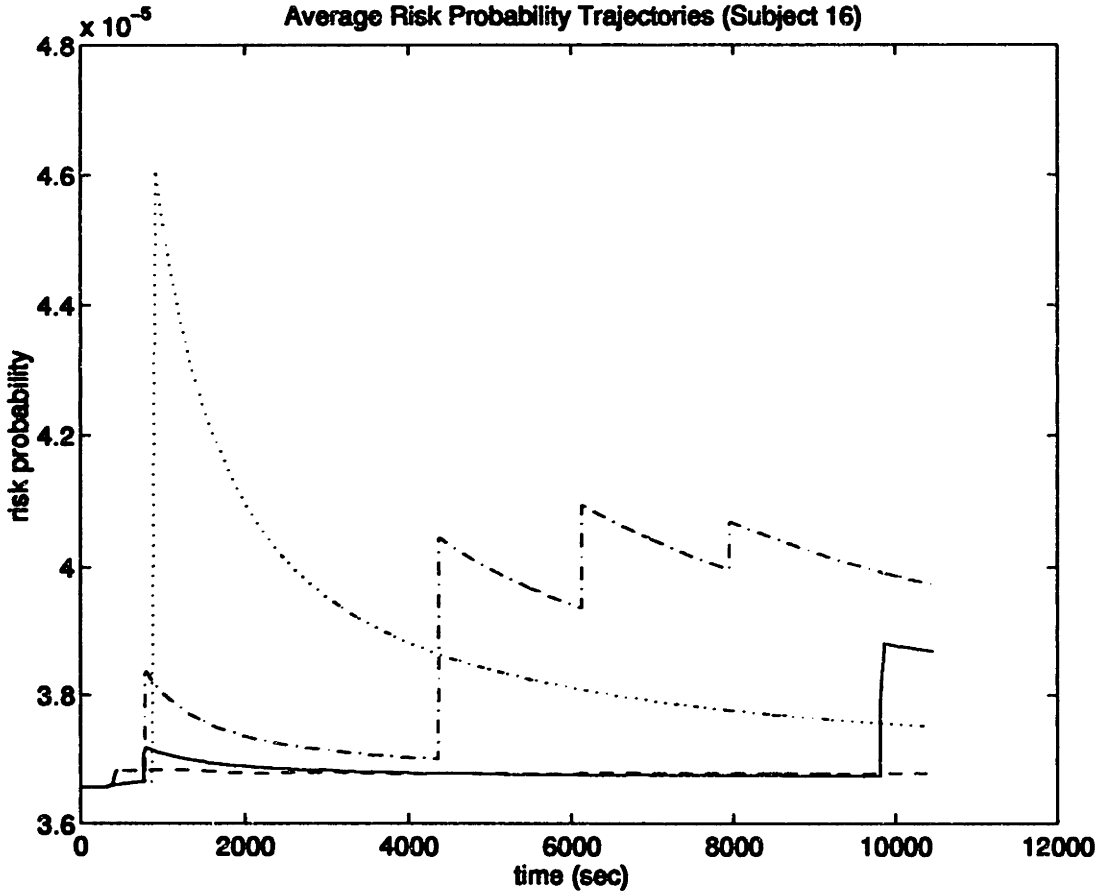

Figure 5-59: Average Risk Trajectories—Subject 16

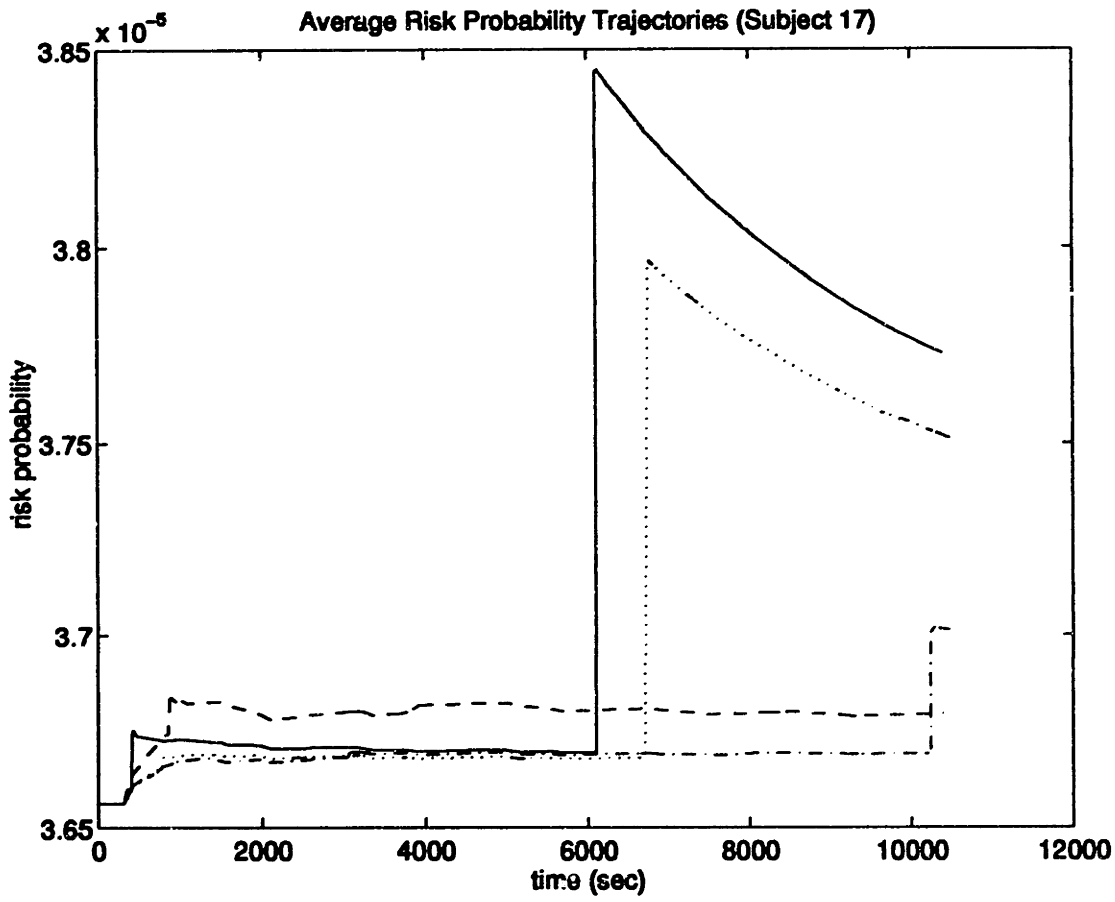

Figure 5-60: Average Risk Trajectories--Subject 17

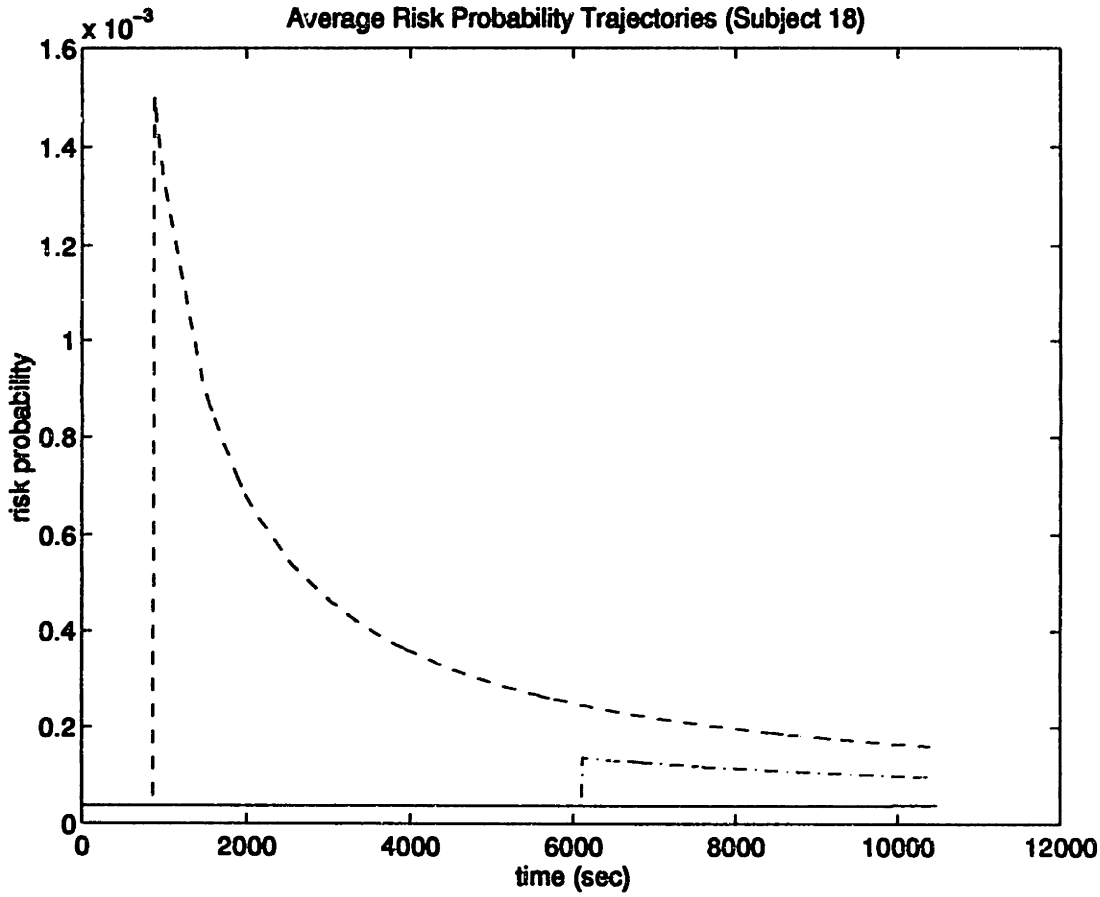

Figure 5-61: Average Risk Trajectories—Subject 18

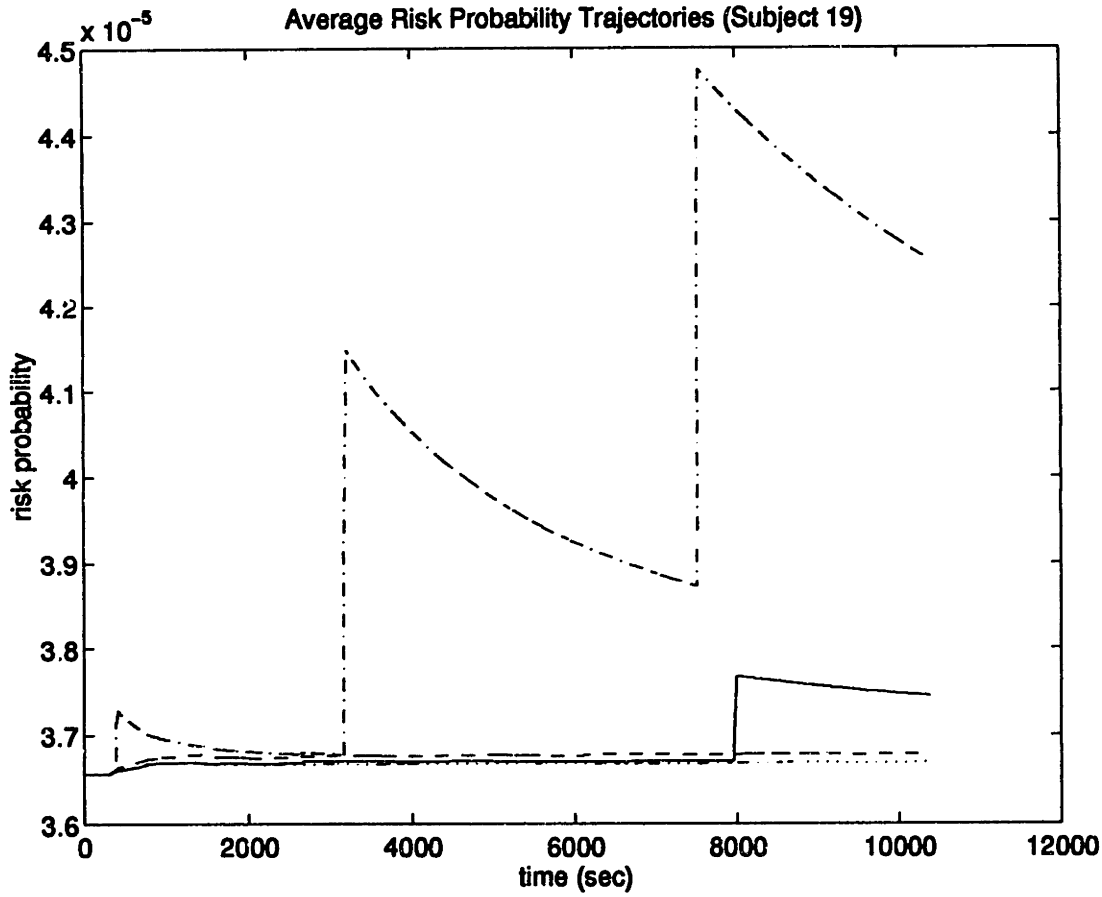

Figure 5-62: Average Risk Trajectories—Subject 19

5.2 Control Automation Experiment

As vehicle speeds increase, the information processing demands on the operator become more significant. Relevant information is presented to the sensory systems at a higher rate, while the available latency time for decisions and control actuation is reduced. In order to ensure that the operator is capable of adequately performing the task of controlling the vehicle, it is appropriate to investigate human factors issues relative to that task before such systems are put into operation.

In an earlier component of this research program, it was determined that are two categories of operator aids most likely to be implemented in actual vehicles: advanced display-based decision aids, and advanced control automation. In the first case, the operator is provided with additional information via advanced display technology. The intended goal of the advanced displays is to improve the planning and control skills of the operator. The second approach is oriented toward providing the operator with control automation systems in order to reduce the required decision and actuation task load. Laboratory experiments have been conducted to explore the effects of the aids on operator performance. In order to determine the individual effects of each class of operator aid, separate experiments were conducted.

An experiment to evaluate the effects of display aiding on operator performance was conducted by Askey [1]. The human performance concern, with regard to display aiding, is potential overload of the operator sensory channels. In particular, there is concern that too much information will ultimately lead to a degradation in overall performance, due to the operator's inability to process the information and extract the pertinent data from it. Experimental evidence was gathered, using the high-speed rail simulation system (chapter 4), with the finding that the display aids generally improved the operator performance in the areas of station stopping, schedule adherence, emergency response, and fuel consumption, at the cost of increased "head-down" time (i.e., the operators concentrate more on the instrument panel than on the OTW view).

The experiment described here is focused on exploring the human factors issues

relative to control automation. With control automation, the functionality of vehicle speed and position control is assumed by an automatic system. Three types of control automation expected in high-speed rail systems are cruise control, programmed stop, and autopilot. A cruise control system is designed to maintain a constant speed, which is set by the vehicle operator. The functionality is generally similar to that found in cruise control systems commonly found in automobiles. An autopilot is similar to cruise control, except that the desired speed is pre-programmed into the autopilot. Thus, the operator is freed from explicitly controlling the speed of the vehicle in normal operational scenarios. However, the operator must still monitor the wayside and vehicle systems, and be prepared to take control of the vehicle if the situation warrants. A programmed stop system is designed to stop the vehicle at a pre-specified position, typically at a station.

In the case of control automation, the human performance concern is that there might be a reduction in operator situation awareness as more of the control task is assumed by automation systems. This is characterized as an "out-of-the-loop" situation, where the operator becomes removed from the immediate task. The potential problem is that, if an emergency or fault situation arises which requires the human operator to take control of the vehicle from the automation system, the operator may consume precious response time trying to gain perspective on the system state, resulting in an extraordinarily long response time to the emergency. In addition, the operator may respond in a manner which is inappropriate to the failure, as a result of being "out-of-the-loop" (e.g., the operator issues the wrong response command). Such scenarios pose serious safety implications.

The "operator out-of-the-loop" problem is often linked with the term *situation awareness.* The definition and measurement of situation awareness remain topics of debate in the human factors community. Some believe that it can be objectively defined and measured, while others believe that it is a useful conceptual construct which is beyond measurement. Despite the differences, there is a common thread which includes a cognitive model. This cognitive model incorporates elements such as attention, workload, stress, long-term memory, and goals [9][14] [19]. Endsley [7][8]

offers the following definition: "Situation awareness is the perception of the elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future." This definition parallels the perception-decision-action paradigm of a classic closed-loop control system with a perception-comprehension-projection paradigm. This paradigm forms a higher-level control loop which allows the operator to set tactical and strategic goals which are subsequently used for guiding lower-level control loops. Applications include aircraft control, air traffic control, large-system operations, and tactical and strategic systems [9]. Application to high-speed rail is in line with current applications.

Smith and Hancock [46] define situation awareness as "adaptive, externally directed conciousness." This definition includes the argument that situation awareness is more than just operator performance, but rather "the capacity to direct consciousness to generate competent performance given a particular situation as it unfolds." Thus, the definition explicitly includes performance as one of the components of situation awareness, which goes beyond Endsley's perception-comprehension-projection paradigm—under this definition, you need action to imply situation awareness.

There are also those who denigrate attempts to define situation awareness. Sarter and Woods [41] assert that situation awareness "should be viewed as just a label for a variety of cognitive processes that are critical to dynamic, event-driven, and multi-task fields of practice." Such cognitive processes might include "control of attention, mental simulation, directed attention, and contingency planning." They continue to state that "it appears to be futile to try to determine the most important contents of situation awareness, because the significance and meaning of any data are dependent on the context in which they appear." These researchers are instead focused on specific measures of performance.

For the purpose of our research, we consider situation awareness to be a useful qualitative concept for discussing the effects of automation. The overall notion that situation awareness includes the adaptive coupling beween humans and machines is of central importance. However, many of the techniques currently available for

evaluating situation awareness incur some level of intrusiveness into the operational scenario. The technique of measuring reponse to emergencies, as implemented in this experiment, was chosen to avoid the problem of task intrusion.

The goal of the experiment is to investigate the relationship between control automation and operator situation awareness in high-speed rail operation. Through the use of a laboratory-based simulation system, operator response to a variety of failures were measured under several applied levels of control automation. The experimental subjects are also requested to complete an exit questionnaire upon completion of the experiment, in which they indicate their preferences and their subjective assessment of the relationship between automation and situation awareness. Based on the objective and subjective results, conclusions and recommendations are made regarding the use of automation in high- speed rail vehicles.

5.2.1 Facilities—High-speed Rail Simulation System

The experiment was conducted using the High-Speed Rail Simulation System, located in the Laboratory for Human Factors Research in Transportation, at the Volpe National Transportation Systems Center, USDOT-RSPA, Cambridge, Massachusetts. The simulation system is a medium fidelity, distributed interactive simulation system, designed for use in human factors experiments. (A detailed description of the simulation system is found in chapter 4 and appendix A.)

For the purposes of the experiment, the system was configured with two SGI Personal Iris workstations and one SGI Indigo-2 workstation. One Personal Iris was used as the CTC operator interface, operated by the experimenter. The subjects were seated at the train simulation. An Indigo-2 (with Extreme graphics) was used for the out-the-window (OTW) display, operating in conjunction with the vehicle dynamics computations. A Personal Iris was used to create the instrument panel display (figure A-3. The operator input was obtained from a control lever and keyboard inputs. The control lever was used for the combined functionality of thrust and brake actuation.

For the experiment, the vehicle was operated in a rail system which forms a two-

station shuttle system (figure A-1). The two stations are separated by approximately 50 km of single track. Beyond each station is a short connector road with a switch at the end for directing traffic to either part of a looping road. The vehicle operator was responsible for moving the train from one station to the other, then proceeding into the loop to reverse direction. The CTC operator was responsible for changing the switch state, which is necessary for the train to pass out of the loop to the main track in the opposite direction.

5.2.2 Method

Subject Task

The primary subject task was to control a simulated rail vehicle through a virtual rail system. The subject was to be immersed, to the highest degree possible, in the task of operating a train. The overall task can be broken down into two components—speed-position control of the vehicle, and system monitoring.

The rail system in use was a two station system, separated by 50 km of track. Outside each station is a looping section of track (with a switch), used to reverse the vehicle on the main track (see figure A-1). The one-way travel time between the two stations is approximately 18 minutes, and the travel time around the loop is approximately 7 minutes. In each test session, the subject operated the vehicle on three round-trip circuits of the system. Each test session required approximately three hours to complete.

The subject controlled the speed of the vehicle by applying either thrust or braking forces, through the combined control lever. The selected speed was subject to speed limit constraints, which may be due to either civil speed limits (static) or signal speed limits (dynamic). The selected speed was also driven by the prescribed schedule—the schedule implies that a minimum average speed be maintained over each trip leg. Through control of vehicle speed, the subject also controlled the position of the vehicle.

The subject was also responsible for monitoring the state of the vehicle and the

overall system. The task included monitoring for specific vehicle failures, as well as external safety-related conditions.

Independent Variable

The independent variable for this experiment is the level of control automation. Three treatments were used: no automation (manual control only), partial automation (combination of cruise control and programmed stop), and full automation (autopilot). Each subject was required to operate in three separate test sessions (*shifts*). Each shift required three hours for completion. The subjects were required to use a single automation mode throughout an entire shift.

Dependent Variables

Over the course of the experiment, the subjects were exposed to a series of unexpected failures. Two types of measures were taken at each failure point. One was the response time, which was measured from the onset of the failure to the correct operator response to that failure. The other measure was more qualitative, rating the "accuracy" of the response. The response accuracy was judged relative to the correct course of action, as specified during the training process.

Subject Counterbalancing

To counterbalance possible learning effects, the presentation order of the automation level was counterbalanced among each group of six subjects. That is, each subject within a group of six experienced the three automation modes in a different order. The resultant design is shown in table 5.6.

Failure Scenarios

There were three different types of failure modes possible during the test sessions: brake failure, motor failure, and grade crossing obstruction. Each represents a different type of detection-response paradigm, allowing for a broad range of inference from the data obtained.

subject number	shift number	automation type
1	1	manual
	2	partial
	3	full
2	1	manual
	2	full
	3	partial
3	1	partial
	2	manual
	3	full
4	1	partial
	2	full
	3	manual
5	1	full
	2	manual
	3	partial
6	1	full
	2	partial
	3	manual

Table 5.6: Subject Counterbalancing Design

The "simplest" type of failure, with respect to detection-response, is the brake failure. The scenario was posed to the subject as follows: One of the brake tanks loses pressure. The pressure loss is indicated on the brake tank pressure gauge. Once the failure is detected, the correct response is to switch to an alternate compressor via a manually controlled switch on the control panel. This failure has the simplest detection-response pattern—the detection is direct, via instrument panel gauges, and response requires one step.

The other vehicle-based failure is a motor failure. In this scenario, the circuit breaker for one motor has been triggered, requiring reset. The failure results in an absence of current flow through one of the motors, indicated on the motor ammeters located on the instrument panel. In this case, the correct response has two steps: Power to the other motors is removed by pulling back on the control lever. The appropriate circuit breaker reset switch is then depressed. If the circuit breaker reset button is depressed before power is removed, all of the remaining circuit breakers will trip, necessitating reset of all the motors.

The third failure is a grade crossing obstruction. There are five grade crossings in the system. At each grade crossing, cars pass over the train tracks. In most cases, when a train is approaching the crossing, a car approaching the crossing on the highway side will stop before entering the crossing. In the failure scenario, a car gets stuck in the grade crossing while passing through, resulting in an obstruction for the approaching train. The subject must identify that the car is stuck and apply the brakes such that the train does not collide with the car. Depending on the speed of the train and the point at which the obstruction is detected, the proper response may be either full service braking or emergency braking. Thus, required response has two steps: First, the obstruction must be identified. Once the failure has been detected, the operator must select the appropriate braking strategy.

Over the course of three test sessions, each subject experienced a total of six occurrences of each failure type. An important design goal was maintenance of the perception, from the perspective of the subjects, that the failures were generated randomly. To maintain experimental control, the design goal was subject to the

constraint that the failures actually occur in fixed positions on the track.

Six track positions were designated for each failure type (table 5.7). Track positions are specified in terms of the block number and the distance from the start of the block. The total number of failures in each test session was either four, six, or eight. Each counterbalanced group of six subjects (table 5.6) received the same presentation of failures, with respect to shift number. However, because of the counterbalancing design, the distribution of failure scenarios with respect to automation mode is also counterbalanced.

Table 5.8 shows the failure scenario specification used for the control automation experiment. Each subject group consisted of six subjects. The order of the automation mode presentation for each subject group followed the counterbalancing design as specified in table 5.6. Each shift required the completion of three round trips of the track system. Table 5.8 specifies the track locations at which the failures occurred, relative to the distance from West Station. The notation specifies each failure by type (designated by a letter code, using *o* for grade crossing obstruction, *b* for brake failure, and *m* for motor failure) and position (designated by a number code, as specified in table 5.7). The total number of failures experienced by each subject was 18, with the failures distributed evenly by type across the set of subjects.

Performance Monitoring and Incentives

In order to assure that each vehicle operator is capable of adequately controlling the train, the performance of each operator (test subject) was monitored throughout the test sessions. As an incentive, there was a bonus system to provide monetary rewards for good performance. Penalties were assessed if operator performance did not fall within certain minimum criteria. At the end of each experiment session, subject performance was evaluated with regard to bonuses and penalties.

Because of federal regulations [5], speed and signal compliance are considered key performance items. In the simulation system, violations are defined by ATP-induced or signal-induced penalty applications of the emergency brakes. During the experiment, the first violation in a shift resulted in a penalty of 100,000 bonus points.

grade crossing failure number	distance from last station (km)	direction
1	1.523	eastbound
2	21.451	eastbound
3	22.726	eastbound
4	22.607	westbound
5	26.549	westbound
6	48.181	westbound

brake failure number	distance from last station (km)	direction
1	2.1	eastbound
2	19.6	eastbound
3	42.3	eastbound
4	4.9	westbound
5	26.1	westbound
6	43.2	westbound

motor failure number	distance from last station (km)	direction
1	1.2	eastbound
2	24.7	eastbound
3	38.4	eastbound
4	7.4	westbound
5	31.1	westbound
6	46.5	westbound

Table 5.7: Track Locations of Failure Points

subject group	shift number	round trip number 1	2	3
A	1	m2,b4	b3,m4,o4	o3,b5,m6
	2	o5,b6	b2,o6	o2,m5
	3	b1	m1	o1,m3
B	1	b2,o3	b4	m3
	2	m1,o1,b6	m4,o4,b5	b1,m2
	3	o2,b3	m6,o6	o5,m5

Table 5.8: Counterbalancing Design for Failure Scenarios

If a second violation had occurred, the subject would have been disqualified from further participation in the experiment. This situation did not occur during the experiment. In addition, willful circumvention of the vehicle safety systems, such as the alerter system, was not tolerated. Subjects found to have bypassed any of the safety systems were disqualified from further participation. This situation occurred once during the course of the experiment.

Other operator performance items subject to bonus or penalty included station stopping accuracy, schedule maintenance, and response to emergency (or failure) situations. In general, good performance in these areas resulted in the award of bonus points. The bonus and penalty points schedules are shown in tables A.6 through A.10. After the total bonus points are computed for a shift, the bonus points are converted into a pay bonus, at the rate of one dollar for each ten thousand points.

5.2.3 Subject Selection and Training

The selection and training of a subject pool is of significant importance in human behavior experiments. In the following sections, the methods and criteria used for selection, training, and acceptance of test subjects for this experiment are described.

Subject Candidate Selection Criteria

In determining subject selection criteria, characteristics of the ultimate target population must be considered. In the case of locomotive engineers, a typical member of the overall population is likely to have an enthusiasm for transportation systems in general, and for rail systems in particular. In addition, through extensive training and apprenticeship programs, these people tend to have a solid working knowledge of vehicle dynamics and control, system operating rules and regulations, and operating territories. They also tend to display a great deal of pride in their work, and correctly view themselves as an integral component of the overall system operation.

Our chosen criterion for subject selection was current or recent status as a student at MIT. Characteristics of the student body at MIT include enthusiasm with respect

to transportation and technology, high level of training in the relevant physical principles, and capability of rapid assimilation of technical material. No arbitrary limits were placed on age, gender, or academic experience. Instead, level of interest in the project was used to filter candidates, through the use of written preparatory material.

Written Tutorial Material

To provide the students with relevant background in rail systems operation, a written tutorial was prepared for their review. This tutorial was written at a relatively low level (compared to the materials used in MIT engineering classes), to facilitate easy reading and motivate interest in the project. The material includes general rail concepts (such as block signaling), implementation-specific design features (such as the available control automation modes), and experiment details (e.g., training procedures, performance incentives). The document is comprised of individual topic blocks which are organized to present the material in a cohesive manner. The tutorial is included in Appendix A.

The use of written preparatory material served two purposes. One was the consistent and efficient presentation of fundamental concepts. All of the potential subjects were exposed to the same basic material in the same format. Great efforts were taken to ensure that the material was written in a clear and concise manner. The second purpose was to act as a filter for subject interest in the project.

Training Sessions

For each subject, training was conducted over two sessions, each lasting three hours. The first training session started with a brief written quiz, used to gauge the understanding of the tutorial material. The quiz consisted of twenty five multiple choice questions, and required approximately ten minutes for completion. When each subject completed the quiz, the experimenter and subject reviewed the quiz together and identified potential problems areas that required attention during the hands-on training. The quiz is included in Appendix B.

The first training session then continued with the simulator. For approximately

one-half hour, the experimenter demonstrated the displays and controls of the simulator, as well as the operational modes and automation systems. Strategies for operating the train and utilizing the automation were discussed. Each subject was then instructed to take one solo round trip passage. The experimenter acted as the CTC operator, and the subject used this period for practice in operation of all the automation modes. In addition, the subject used this period as an opportunity to gain familiarity with the wayside environment and the schedule. No failures were activated on this run. The session continued with another round trip passage, where the explicit intent was to provide the subject with experience in the failure scenarios.

The second training session was used primarily for practice and evaluation of manual control skills. The subject was instructed to take a full shift, three round-trips, using manual control only. The subject was to abide by the published schedule, and was instructed that failures may occur. The first hour was considered a practice period, during which the subject was encouraged to experiment with strategies to improve station stopping and schedule performance. During the latter two hours of the shift, the performance of station stopping was measured. At the end of the shift, the station stopping performance was evaluated, and a recommendation for continuation with the test sessions was made.

Subject Acceptance Criteria

The final criterion used for accepting a subject for experimental sessions was performance during the road test portion of the second training session. At the beginning of the second training session, each subject was told that the evaluation criteria included station stopping performance. This criterion specified a maximum allowable stop accuracy of 10 meters overshoot or undershoot. In addition, the subject was informed that penalty application of the emergency brake, due to the ATP or alerter systems, was not acceptable.

A total of 20 subject candidates completed the training session. Of those, 7 were disqualified due to insufficient training. These subjects, however, did complete all of the steps in the training and test process, including completion of the exit question-

naire. The training procedures were revised after these subjects were processed.

Of the remaining 13 subjects, 1 was unable to pass the competency criteria at the end of the training period. This subject was not permitted to continue with the testing phase. The 12 remaining subjects were used for compilation of the test data.

5.2.4 Experimental Results

The objective of the experiment was to determine whether there are any differences in operator situation awareness performance under the three conditions of control automation, as measured by emergency scenario response. Statistical analysis was used to determine whether the observed differences in response are more likely due to effects of the treatment or simply the result of noise in the system. The general null hypothesis is stated as follows:

> H_0: There is no difference between the population means among the three treatments.

For the purposes of the statistical analysis, the null hypothesis is restated as:

$$H_0: \mu_m = \mu_p = \mu_f$$

The response time data is summarized in tables 5.9 through 5.11. Each table represents one of the three different failure scenarios, and each includes three columns of data corresponding to the control automation variants. The data are ordered in increasing value, to give a sense of the distribution. [1] At the bottom of each column is the average (mean), standard deviation, and variance for that column.

Since the sample mean is an unbiased estimator, we use the sample mean to estimate the population mean. A significance of 5% ($p = 0.05$) is selected for the test. To compare the means, an analysis of variance (ANOVA) is used [18][47][17].

[1] Because of the counterbalancing design, the number of failures relative to test type was varied between subjects. All of the subjects experienced each of the failures a total of six times, but the distribution of those failures relative to the automation level varied from subject to subject. As a result, it does not make sense to present the data with any correlation to subject identification. Instead, each data group is rank ordered by value, from least to greatest. Displaying the data in this manner provides a sense of the distribution characteristics.

One of the fundamental assumptions of the ANOVA technique is that the population variances are equal among the treatments considered. To verify the truth of this condition for the test data, Bartlett's test [17] is applied. In this case, the null hypothesis is:

> H_0: There is no difference between the population variances among the three treatments.

Restated in the statistical form:

$$H_0: \sigma_m^2 = \sigma_p^2 = \sigma_f^2$$

A significance of 1% ($p = 0.01$) is selected for the test.

Upon inspection of the data, it is immediately apparent that, in both the brake and motor failure data, the standard deviation is quite different between the treatments. Bartlett's test was performed, with the results shown in table 5.12. Based on this analysis, we can justify refuting the null hypothesis regarding the equality of variance in the cases of the brake and motor failures. In other words, the difference in the variance is significant at the 1% level for both brake and motor failure data.

Bartlett's test includes a test to determine the data series which have a variance which is significantly different from the aggregate. This portion of the test is only suitable with a large number of categories—because our test has only three categories, the test cannot be applied effectively here. However, based on inspection of the values of variance shown, it is reasonable to conclude that the variance of the partial automation mode is different from the other two. Comparison of the mean response time for both the brake and motor failures shows a larger value for the partial automation mode, which is consistent with behavior of the variance statistics. The difference in mean cannot be shown to be statistically significant, in part because the appropriate tests rely on the assumption that the population variances between the samples is equal. Figures 5-63 and 5-64 provide a graphical representation of the data distribution, using a box-plot display.

In the case of the obstruction failure, application of Bartlett's test showed that there was no significant difference in the variance of the response time data. An

	manual	partial	full
1	5.5	5.6	4.4
2	5.8	5.8	6.3
3	6.9	8.2	6.5
4	7.0	8.6	7.3
5	7.6	11.1	7.7
6	8.5	11.5	9.1
7	10.1	13.4	11.8
8	10.3	13.7	14.4
9	10.9	16.3	14.6
10	13.6	17.2	16.9
11	14.1	17.5	17.3
12	15.9	18.4	19.3
13	16.3	26.4	19.9
14	18.5	26.5	20.4
15	29.8	28.1	20.8
16	30.6	31.4	20.8
17	31.9	35.8	21.4
18	33.2	35.9	24.0
19	35.2	40.6	31.3
20	39.5	49.2	32.5
21	39.6	63.7	34.7
22	43.4	78.5	43.7
23	52.1	103.3	45.1
24	61.9	119.2	78.1
count	24	24	24
mean	22.8	32.7	22.0
stdev	16.26	30.37	16.39
variance	264.409	922.347	268.668

Table 5.9: Brake Failure Response Time Data

	manual	partial	full
1	2.7	2.6	2.9
2	2.9	3.0	2.9
3	3.1	3.2	3.1
4	3.2	3.6	3.1
5	3.2	4.1	3.2
6	3.2	4.7	3.8
7	3.8	5.2	4.6
8	5.2	5.3	5.3
9	6.5	6.1	6.4
10	7.0	6.7	6.4
11	7.0	6.8	8.6
12	7.6	7.3	9.0
13	7.9	7.7	9.6
14	8.3	9.1	12.1
15	8.8	10.3	12.7
16	10.6	14.3	13.3
17	12.4	15.1	13.9
18	14.6	15.8	14.0
19	14.6	26.3	17.6
20	14.6	30.8	21.2
21	15.0	32.8	23.7
22	15.2	46.0	31.0
23	17.8	92.6	36.8
24	78.7	122.6	38.6
count	24	24	24
mean	11.4	20.1	12.7
stdev	15.11	29.49	10.62
variance	228.284	869.672	112.752

Table 5.10: Motor Failure Response Time Data

	manual	partial	full
1	0.9	1.0	1.3
2	1.3	1.3	1.3
3	1.3	1.7	1.5
4	1.3	2.0	1.5
5	1.4	2.0	1.6
6	1.5	2.1	1.9
7	1.5	2.1	2.0
8	1.6	2.1	2.0
9	1.8	2.1	2.5
10	1.8	2.3	2.7
11	2.0	2.5	2.8
12	2.0	3.0	3.3
13	2.4	3.2	3.6
14	2.7	3.4	3.6
15	3.0	3.7	4.8
16	3.6	3.8	5.6
17	3.7	4.3	5.9
18	3.7	4.6	6.7
19	4.5	4.8	7.2
20	6.5	5.6	7.3
21	11.0	5.7	8.1
22	12.1	5.8	13.5
23	13.1	14.5	13.6
24	14.8	14.8	17.4
count	24	24	24
mean	4.1	4.1	5.1
stdev	4.17	3.54	4.36
variance	17.383	12.509	19.040

Table 5.11: Obstruction Response Time Data

	brake	motor	obstruction
χ^2	12.5	24.3	1.1

Table 5.12: Results of Bartlett's Test for Equality of Variance

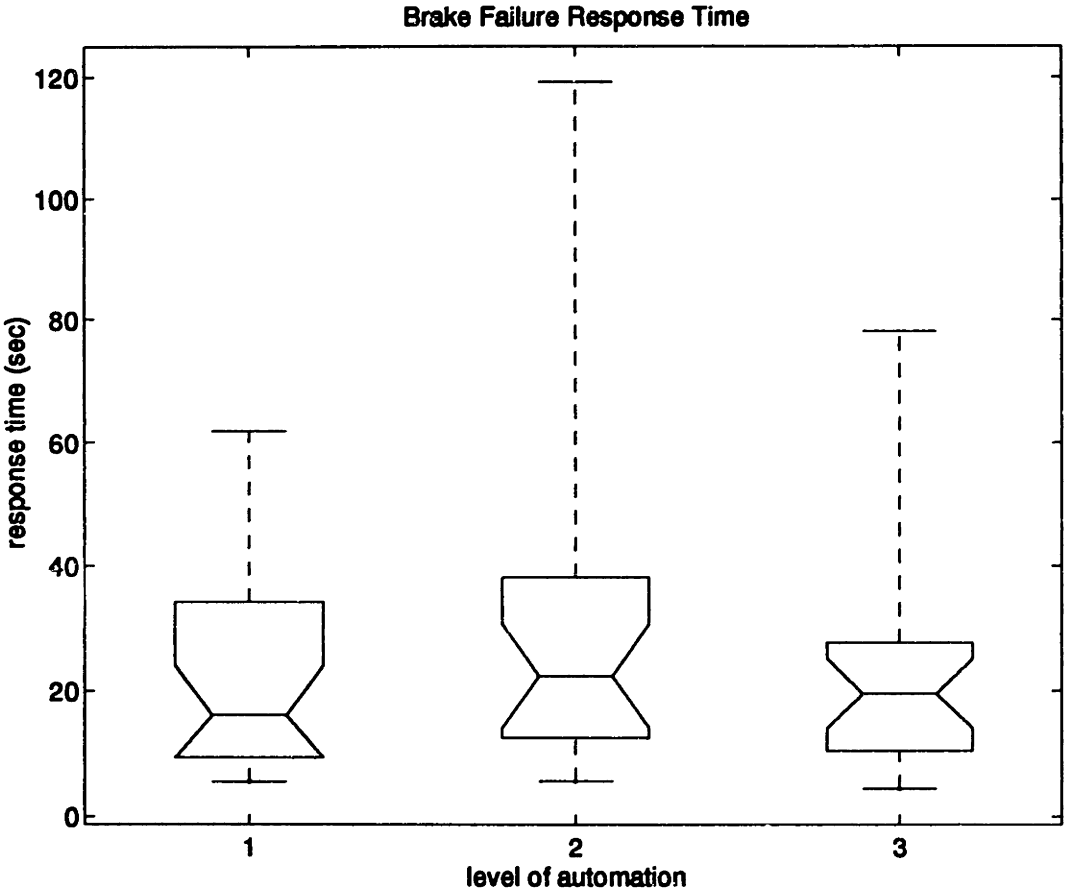

Figure 5-63: Box-Plot Display of Brake Failure Response Time

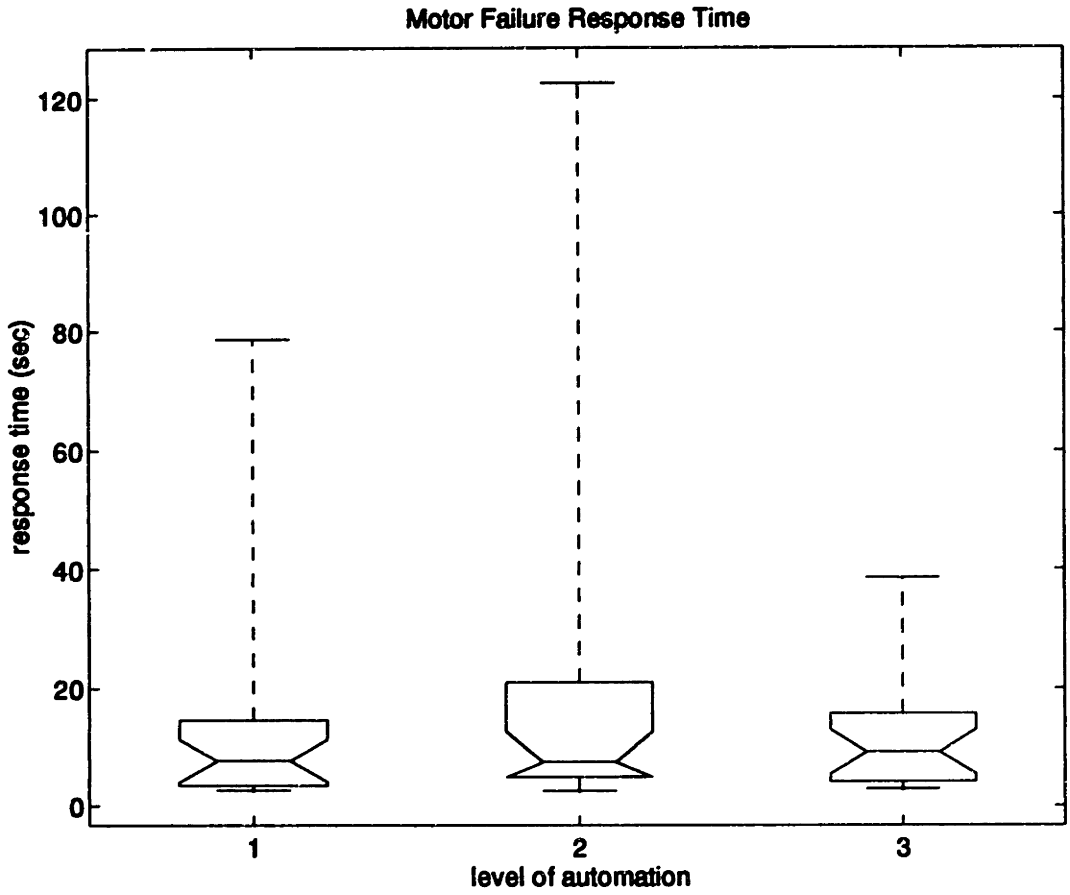

Figure 5-64: Box-Plot Display of Motor Failure Response Time

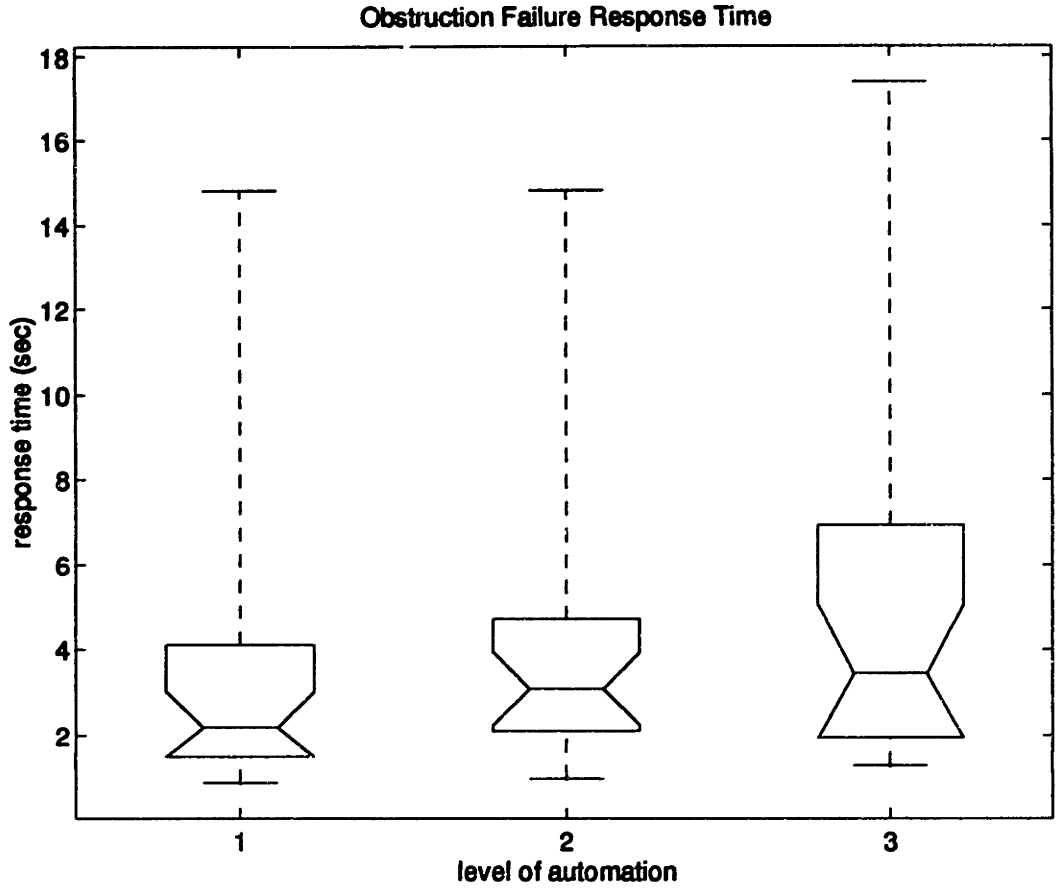

Figure 5-65: Box-Plot Display of Obstruction Response Time

ANOVA test was then applied to the obstruction data. The resultant F value is 0.4415, with degrees of freedom $\nu_1 = 2$ and $\nu_2 = 69$, leading us to the conclusion that we cannot refute the null hypothesis—the differences were not significant at the 5% level. Thus, the observed differences cannot be attributed to differences in the treatments but are rather attributed to noise in the data. Visual inspection of a graphical representation of the data leads us to a similar conclusion (figure 5-65).

Measure of response accuracy was also tabulated. Responses were classified as either *correct* (the proper response procedure was followed), *mistake* (the first response was not correct, but did not pose any danger), or *bad* (the response included actions which could be dangerous to the operator or passengers). The results are shown in tables 5.13 through 5.15.

These data show that the relative error rate is low across all of the failures and all of the treatments. A χ^2 test was performed across both the *mistake* and *bad error* rows of the three failure types. This test allows determination of goodness-of-fit to

	manual	partial	full
mistakes	1	2	0
bad errors	0	0	0
% mistakes	4.2	8.3	0.0
% bad errors	0.0	0.0	0.0

Table 5.13: Summary of Brake Failure Response Accuracy Data

	manual	partial	full
mistakes	2	2	2
bad errors	0	1	0
% mistakes	8.3	8.3	8.3
% bad errors	0.0	4.2	0.0

Table 5.14: Summary of Motor Failure Response Accuracy Data

	manual	partial	full
mistakes	0	0	0
bad errors	0	0	1
% mistakes	0.0	0.0	0.0
% bad errors	0.0	0.0	4.2

Table 5.15: Summary of Obstruction Response Accuracy Data

test	χ^2 value	significant?
brake test, mistake	0.667	no
brake test, bad error	0.000	no
motor test, mistake	0.000	no
motor test, bad error	0.667	no
obstruction, mistake	0.000	no
obstruction, bad error	0.667	no

Table 5.16: χ^2 Test of Uniform Error Rates

a prescribed behavior model. The null hypothesis used states that the errors occur at a uniform rate across all automation modes. The resultant χ^2 values are shown in table 5.16. The level of significance selected was 5% ($p = 0.05$), and the corresponding χ^2 value is 3.84. Thus, the computed value of χ^2 must exceed 3.84 in order to refute the null hypothesis. Based on the results listed in table 5.16, the null hypothesis could not be refuted in any case; therefore, we cannot conclude that there is any effect on response accuracy as a result of control automation.

Finally we consider the subjective data taken in the exit questionnaire. This questionnaire asked the subjects to rate the different automation modes in terms of perceived situation awareness and personal preference. In the case of perceived situation awareness, the subjects were asked to assign a '1' to the automation variant in which they felt most aware, and a '3' to the variant in which they felt least aware. Similarly, they were asked to assign a '1' to the variant which they most preferred, and a '3' to the variant which was least preferable. Eighteen subjects completed the questionnaire, with the results as listed in table 5.17.

It is clear from the responses to the awareness question that the subjects perceive a correlation between automation mode and awareness, with the manual mode having the highest 'awareness' rating and the autopilot having the lowest 'awareness' rating. However, despite this, there is a wide discrepancy between the preferences of the subjects. There are some subjects who prefer to use an automation mode in which they perceive a lower awareness, implying, in addition to perceived situation awareness, there are other factors involved in the selection of automation mode.

	preference			awareness		
	manual	partial	full	manual	partial	full
high	4	5	9	14	4	0
med	6	8	4	4	10	4
low	8	5	5	0	4	14
% high	22.22	27.78	50.00	77.78	22.22	0.00
% med	33.33	44.44	22.22	22.22	55.56	22.22
% low	44.44	27.78	27.78	0.00	22.22	77.78

Table 5.17: Subjective Ratings of Control Automation Modes

5.2.5 Discussion

From the results obtained, it is clear that the use of automation has an impact on the operator response to unexpected emergency scenarios. These effects differ between the on-board and wayside failures.

The brake and motor failure scenarios represent the on-board failures. In both, the partial automation mode shows a significant increase in variance when compared with the manual mode. This result supports the theory that the operator is less inclined to monitor the vehicle systems when an automatic controller is given responsibility for the low-level control tasks.

However, there was also a difference between the variance observed in the partial and full automation modes. In both the brake and motor failures, the variance is greater in partial automation than full automation, suggesting that there might be a change in the operator observance strategy—the operator pays more attention to the instrument panel in the full automation mode than in the partial mode.

It is interesting to note that the response time to obstructions in the guideway does not differ significantly among the three automation variants. The consistent nature of the response suggests that the operators are alert to the risk of an obstruction when they are approaching the grade crossing regardless of the level of automation.

With regard to response accuracy, the data show no statistically significant difference between the three treatments across all of the tested failure modes. This result suggests that the control automation has a negligible impact on the accuracy

of operator response to failure scenarios.

Overall, these results can be explained by the following logical analysis. The task of the rail vehicle operator involves observing the wayside as well as observing the in-cab instruments. Based on the observed state of the "world" and the operating rules, the throttle and brake are controlled to achieve a desired speed. In the manual control case, the speed regulation sub-task involves monitoring the speedometer as well as monitoring the wayside for brake point landmarks. When partial automation is added, the requirement for monitoring the speedometer is relaxed, but the operator must still carefully monitor the wayside for brake point landmarks. In the full automation variant, the necessity to monitor the wayside is relaxed—the autopilot automatically brakes the train at the correct point, without input from the operator.

So, as the amount of control automation is increased, the monitoring requirements of the operator are progressively reduced. As a result, the operators bias their observational attention to the area which is believed to be most important.

Unfortunately, this line of reasoning has a potential negative consequence, suggesting that a higher level of control automation, while easing the observational demands on the operator, has the potential for luring the operators into a higher level of complacency over the course of time. In other words, the ability of a vehicle operator to respond to a truly novel emergency situation may be impaired by higher levels of control automation. Testing the effects of complacency and response to novel emergencies requires very long subject test sessions, which were beyond the scope of this experiment.

Chapter 6

Conclusion

The research summarized in this document has led to the successful development of a model for estimating dynamic risk probability in operational systems. This model, based on a discrete finite Markov process, provides a framework for observing and measuring an operational system, and further provides a method for transforming those observations into predictions about the future.

The most significant contribution has been the development of the safety state model. A justification for the model is established, and the mathematics of the model is fully developed in this work. The result is a generalized model for observing the safety-related behavior of a human-machine system, applicable to virtually any such system. The major advantages of the model are generality and comprehensive nature—this model can be used to explore and predict effects and combinations of interactions that might be completely overlooked in other forms of risk analysis. Disadvantages includes scaling issues—the size of the model increases with the square of the complexity. In addition, the observational nature of the model restricts its use to existing operational systems.

In the course of developing the safety state model theory, a closed form solution for calculating the mean time to failure (MTTF) in a discrete Markov model was developed. This solution requires only the state transition matrix, and involves a single matrix inversion. The primary approach uses z-transform analysis, and the result was verified using an alternate method. This derivation and result were not found

elsewhere in reviewed literature on Markov process analysis, and are thus regarded as an original contribution of the research.

The research further contributes to the development of the safety state model by providing experimental demonstration of the concept. Through the use of data gathered in a concurrent behavioral experiment, the safety state model has been shown to identify different levels of risk probability as a function of system state. This result has led to identification of high-risk scenarios and the causal events that lead to them. Thus, the safety state model is of use in systems analysis. In addition, the dynamic risk probability estimation can also be used for analysis of individual operator performance, relative to safety.

Another contribution of this work is the high-speed rail simulation system. Designed and implemented for the purpose of conducting laboratory experiments in human factors, the simulation system represents a step forward in capability for human behavioral studies in high-speed rail. Furthermore, the system provides a testbed for developing new technologies, using a modular design philosophy to allow rapid prototyping of instrumentation and display technologies, among others. Finally, simulation systems in this model are the only acceptable method of performing significant safety studies, which would expose participants to unacceptable risk levels if conducted using an actual system.

In part to generate data for safety state model validation, a human-subject experiment was conducted using the high-speed rail simulation system. The purpose of this experiment was to explore the effects of control automation on operator performance. Test subjects were asked to operate the vehicles, using three different levels of control automation: manual only, partial automation (cruise control and programmed stop), and full automation (autopilot). During the tests, three different types of failures were presented to the subjects. These failures included vehicle system failures (brake failure and motor failure) as well as system emergencies (vehicular obstructions at grade crossings). Responses to these failures were measured and analyzed. The results lead to the conclusion that adding automation has the effect of altering attentional biases. In addition, under the conditions provided in the experiment, there was no signifi-

cant degradation of performance at the grade crossings as a consequence of using the automation.

In summary, the safety state model has proven to be a useful tool for safety-related analysis of operational systems. Potential future developments include application of the methods to operational systems. The author believes that the safety state model has the potential for wide application, both within and without the transportation community. Future enhancements of the theory would include extension of the model to include several failure events, using multiple-chain Markov processes. On the applied technology front, the issue of scaling complexity might be addressed through application of parallel processing computation.

Appendix A

Training Tutorial

Experiment—General

You are participating in an experiment which will answer questions about the relationship between control automation and operator situation awareness in high-speed rail operation. Although the control of rail vehicles and systems can be automated, there are concerns about potential side effects when using automated control in high-speed rail systems. In order to explore these effects, test subjects like yourself are asked to operate a simulated high-speed train in a virtual reality environment. Your actions are measured during the tests, and later analyzed.

Participation in the experiment consists of two phases. The first phase is a period of training, which consists of a three-hour instruction session and a three-hour practice session. Prior to the instruction session, you are asked to review this document, which is a tutorial on the operation of the simulated train vehicle. At the start of the instruction session, you will take a written review quiz, which will gauge your understanding of this material. The written review quiz consists of 25 multiple choice questions, and is closed book. The quiz is graded immediately after completion. You will then be directed by the instructor through a set of training instructions, which will familiarize you with normal operation of the vehicle. You will also be exposed to a set of emergency scenarios, allowing you to learn the proper responses to these situations.

The second half of the training phase consists of a combination practice and test session. This session is conducted like a regular experiment session, in that it is a three-hour session consisting of three round trips between the two stations in the system. The first hour is considered to be the practice portion. During this period, you will be reacquaint yourself with operation of the train simulator. The remaining two hours is considered a "road test" which will further gauge your abilities to operate the train. Your performance with regard to speed compliance, signal compliance, and station stopping accuracy will be evaluated. If you pass this test, you will be ready ("certified") to perform the experiment trials. You will be eligible for payment for the training phase upon completion of the practice and test session.

The second phase consists of a set of experiment trials. These experiment trials will take place in three separate three-hour sessions. Each session is called a shift, and corresponds to a shift of operation in an actual rail operation. During each shift, you will operate the simulated vehicle as if it were part of an actual rail system. Once underway, you will be expected to remain at the simulator controls ("in the vehicle") until the shift is complete. Brief break periods are allowed, as approved by the experimenter (acting as the CTC operator). The three experiment trials will be conducted on three separate days. You will be eligible for payment for the experiment trials upon completion of the third shift.

Payment for the experiment is through the MIT voucher payroll system. The rate of pay is $25 per three-hour session. Therefore, the sub-total for the training phase is $50, and the sub-total for the experiment phase is $75, resulting in a total payment of $125. (Payment for the experiment phase is subject to performance bonuses, as well as penalties that result from illegal behavior—please refer to the section titled "System Operation—Operator Performance Requirements" for details.) Subjects are paid for each phase completed, regardless of performance. However, subjects that do not pass the training phase will not be allowed to continue with the experimental phase. Subjects can elect payment for the training and experimental phases to be separate (resulting in two checks), or payment for the two phases can be lumped together into one check.

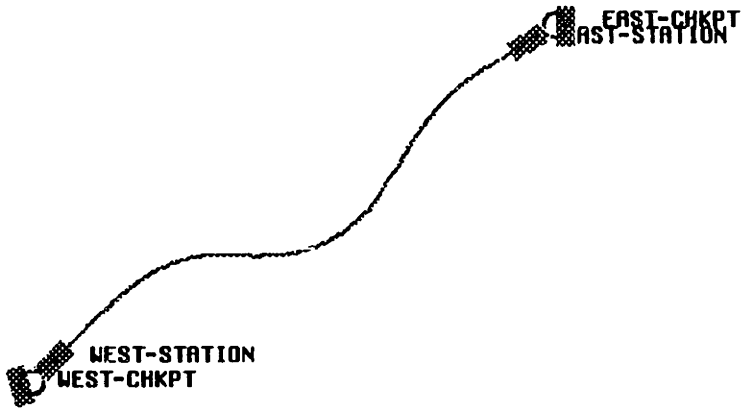

Figure A-1: Track Layout, Simulated Rail System

This tutorial is organized so that the reader can learn the fundamentals of rail system operation in a logical order. Important terms and concepts are highlighted with bold-faced text.

System Operation—General

The rail system used in the experiment is a fictitious rail system connecting two stations, named West Station and East Station (figure A-1). The two stations are connected via a single track which is 50 km in length. At each end of system, beyond the stations, there is a loop of track which is used to turn the vehicles around for the return trip.

The system is operated as a high-speed shuttle between these two stations. There is one vehicle in operation. That vehicle will travel from one station to the other, discharge passengers, loop around to reverse direction, board new passengers, and proceed to the other station. This procedure is followed throughout the duration of

the shift.

Operation of the system is coordinated through a central traffic control operator (CTC). This person is located in a fixed position in the system, and has access to the state of all the vehicles operating in the system. The CTC operator has the task of coordinating the operation of several vehicles that must share resources (such as the track system). To carry this task out, the CTC operator has control over the switches in the system, and can set signal levels manually. In addition, the CTC operator is able to communicate directly with vehicle operators.

The wayside is a general term which refers to all objects in the environment which do not move. This includes items such as the ground, the track, the signal lights, the surrounding trees and buildings, and so on.

System Operation—Block Signal System

Rail systems have traditionally used a system known as block signaling for control of trains in the rail system. With block signaling, the track is divided into fixed length chunks known as blocks. While the length of each block does not change, different blocks are not necessarily of equal length. Typically, shorter block lengths are used in the near vicinity of stations, while longer block lengths are used in regions away from the stations. Block lengths are generally on the order of one mile. In the road system used in the simulation, all blocks between stations are 2 km, and all blocks in the loop sections are 1 km in length.

At the boundaries of each block is a signal light. This signal light displays a color-coded signal, which indicates the maximum speed permitted throughout the block. The signal acts as a dynamic speed limit, and it is the responsibility of the vehicle operator to identify the signal as the block boundary is approached and set the vehicle speed accordingly.

A fundamental rule in block signaling is that no more than one train can occupy a block at any given time. A red signal is used to indicate that the block is currently occupied by another train, and the approaching train is not permitted to enter that block. The blocks that precede the occupied block have signal levels which ensure

COLOR	CODE	ACTION
red	STOP	not permitted to enter block
red/yellow	RESTRICTED	max speed of 80 km/hr
yellow	APPROACH	max speed of 150 km/hr
green/yellow	APPROACH MEDIUM	max speed of 230 km/hr
green	CLEAR	max speed of 300 km/hr

Table A.1: Rail Signal Codes

that the train can be slowed in time to stop before entering the occupied block.

In addition to the speed limits imposed by the block signal system, there are also civil speed limits, which are static. These limits are either memorized or written down by the operator. In all cases, the prevailing speed limit is the lesser of the block signal limit and the civil speed limit.

The exact specification of signals used and speed limits associated with those signals is a design parameter for a rail system, and varies from system to system. In the simulation system, a five- aspect signaling system is used. This means that there are five color codes used in the system, with the codes defined as shown in table A.1.

If a train was occupying block 157, then the signal at the entrance to block 157 would show STOP (red), the signal at the entrance to block 156 would show RESTRICTED (red/yellow), the signal at the entrance to block 155 would show APPROACH (yellow), the signal at the entrance to block 154 would show APPROACH MEDIUM (green/yellow), and the signals at blocks prior to block 154 would show CLEAR (green). The speed limits apply to the entire block, which means an approaching train must reduce speed to the limit before reaching the entrance of the block. So, in this example scenario, another train approaching the train in block 157 must be going slower than 230 km/hr before entering block 154, slower than 150 km/hr before entering block 155, and slower than 80 km/hr before entering block 156 (see figure A-2).

Located throughout the system are position markers known as kilometer posts. The use of these by vehicle operators is discussed in detail in the next section. It is important to note the difference between block signals and kilometer posts. At the

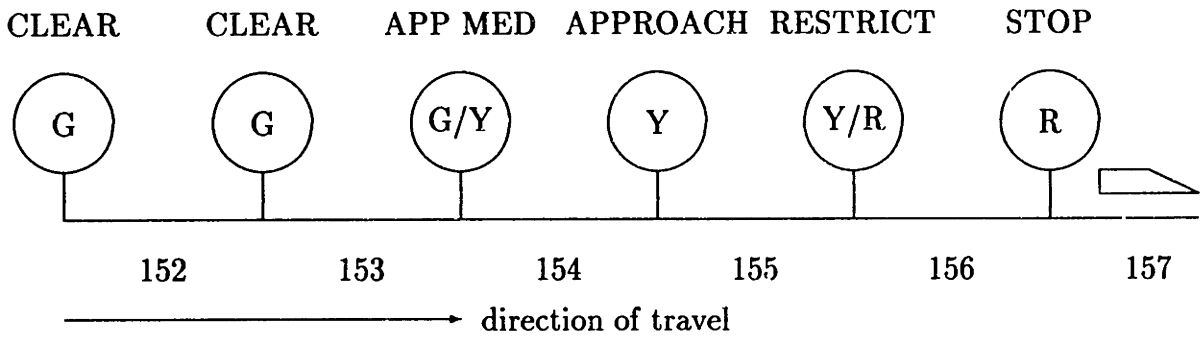

Figure A-2: Block Signaling System

entrance to each block, there is a signal board which identifies the block number and displays the current signal level for that block. Because block boundaries occur at 2 km intervals on the main line in this system, there is also typically a kilometer post at the block boundaries. So, for example, block 13 comprises the distance of track between kilometer posts 26 and 28. This provides opportunity for confusion: When traveling eastbound, the entrance to block 13 is marked by kmpost 26, but when traveling westbound, the entrance to the same block occurs at kmpost 28. Operators must take care to differentiate between block identifiers and kilometer posts, as the relationship between them is not as simple as it might at first appear.

Vehicle Operation—User Interface

The user interface for the train simulation consists of two graphics displays, a computer keyboard, and the combined control lever. The two graphics displays are placed side by side on a table, with the computer keyboard between them and the combined control lever to the right.

The display to the right is the instrument panel for the train. A schematic drawing of the instrument layout is shown in figure A-3. In this figure, we can locate the speedometer, the secondary gauges (brake tank pressure, wheel bearing temperature,

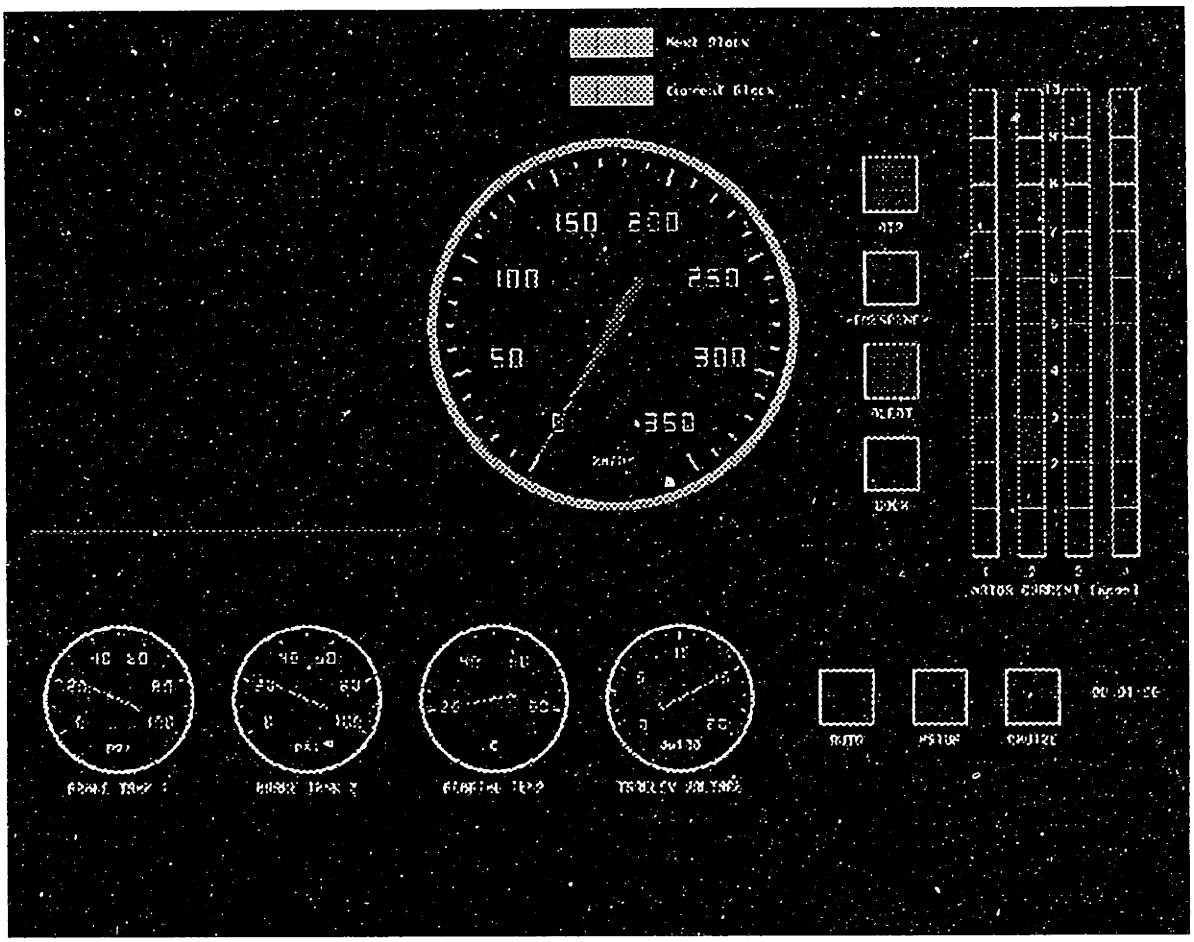

Figure A-3: Instrument Panel Layout

and trolley voltage), the mode indicators, the warning indicators, the motor current meters, the system clock, and the communications display.

The graphics display on the left is the out-the-window viewport. This is also known as the windscreen. In this view, the operator is able to view the world outside the vehicle. The scene presented is a night scene, with wire-frame objects along the wayside. This port also provides a head-up display for assistance in station stopping (described in a later section), as well as an indication of collision with a highway vehicle in a grade crossing (described in a later section).

The combination control lever is a joystick-like lever to the right of the keyboard (beside the instrument panel display). Although there are two handles on the control, only the right black- handled lever is used. This lever is used to control both the thrust

and braking commands, with the forward (up) direction for thrust and the backward (down) direction for braking. The center position (coast) is notched for reference.

The computer keyboard is used for both control buttons and alphanumeric communications input. The top row of function buttons are assigned to specific operator control functions, which are described in the following sections. The main part of the keyboard is used for entry of communications messages. These messages are typed in, and are displayed in the communications area of the instrument panel. While a message is being composed, it appears in the lower portion of the communications display, and is visible to only the operator. When the carriage return is depressed, the message is "sent" to all other operators on the system, including the CTC operator. The message then appears in the "receive" area (top portion) of the communications display. The messages that appear in the top portion are visible to all operators in the system.

Vehicle Operation—Head-Up Display for Station Stopping

Because of the braking dynamics of rail trains, one of the more challenging aspects of train operation is stopping the vehicle at an appropriate point in the station. This becomes very important when the vehicle must be stopped at a precise point, for example, to allow proper alignment of handicapped ramps.

Because of the limited perspective through the out-the-window viewport, a driving aid is provided in the train simulator to assist the vehicle operator in this task. The aid takes the form of a head-up display (or HUD),which is a graphical icon superimposed on the windscreen. (Head-up displays are commonly found in military aircraft, where a set of instrument displays is projected on the windscreen so that the pilots can monitor the vehicle status while concurrently performing other visual tasks.) The HUD in the train is a yellow rectangular box, which appears when the train is within the confines of the station. When the HUD is aligned with the back wall of the station, the train is at the ideal stopping point in the station.

In the event that the vehicle comes to a stop before it is in the station (i.e., before the HUD appears), the operator can slowly bring the vehicle into the appropriate

position. If the vehicle comes to a stop after the station is passed (overshoot), the operator must contact the CTC to report the situation and request further instruction. Under no circumstances should the vehicle operator open the passenger doors if the vehicle is not within the boundaries of the station.

Vehicle Operation—Speed and Position Control

The most important task required of a train vehicle operator is the control of vehicle speed. The operator uses the position of the vehicle to obtain the current speed limit, through a combination of civil speed limits (which are memorized or written down) and block signal states (which are observed on the wayside). The operator then uses the traction motor system and the braking system to adjust the speed of the vehicle accordingly.

The vehicle operator gets information about vehicle speed through the speedometer, which is located on the instrument panel in the locomotive cab. In the train simulation, this speedometer is implemented as a round dial gauge. The units shown are kilometers per hour (km/hr), and the available range of speeds is from 0 to 350 km/hr. The major increments of the gauge display are 50 km/hr, with minor increments each 10 km/hr. The red pointer indicates the current speed, while the smaller yellow pointer (underneath the red pointer) indicates the set speed (used by the automation systems).

Another important task of the vehicle operator is monitoring the position of the vehicle in the rail system. This is done by monitoring the out-the-window view. Along the wayside, distance is marked through the use of "mileposts." Typically, at one mile intervals, a post is placed on the wayside with numbers indicating the mile marker. Vehicle operators use the difference between posts to measure distances along the road. Because the simulation system is implemented using the metric system, these posts are referred to as "kilometer posts," and are located at one kilometer intervals.

When approaching a stopping point, such as a station, the vehicle operator uses out-the-window cues to identify points at which the brakes should be applied to stop at a particular position. Because of the high mass of the train (with resultant high

momentum when in motion), accurately braking the vehicle requires relatively long lead times for the control commands. As a result, it is common for train operators in real systems to use stationary objects on the wayside as braking point markers. Learning the proper braking points represents a significant part of the training process for vehicle operators.

To assist in learning and remembering the appropriate braking points, there are significant landmarks placed in the wayside environment at points which help locate the braking points. During the training phase, the subject is provided with three-page chart titled "Brake Point Specification Worksheet." This worksheet shows a schematic diagram of the track, and identifies the speed restricted areas, as well as the braking points for these areas. The chart also identifies the landmarks which are placed near each braking point.

Blocks 0, 1, 11, 12, and 13 all have speed restrictions, due to the grade crossings present in these blocks. When traveling from West Station to East Station, the first major braking point occurs at kilometer post 17, to slow from 300 km/hr to 100 km/hr before reaching block 11. At km 17, on the right, there is a group of 5 red columns which serve as a landmark. The next major braking point is at kilometer post 44, for stopping at East Station. The landmark at this point is a large red building on the left, surrounded by a blue fence. When traveling from East Station to West Station, the first braking point occurs at kilometer post 33, to slow in time for the speed restriction at block 13. In this case, the landmark is a red pedestrian bridge over the track. Finally, at kilometer post 9, there is a large red building with a green pointed crown on its roof, located to the left side, signifying the braking point for entrance to block 1. These landmarks are noted on the Brake Point Specification Worksheet.

The skill of stopping the train at the station is a critical component of vehicle operation in passenger rail systems. Tables A.2 and A.3 are provided to assist in learning this skill. In the simulation system, the ideal stopping point in the station is defined as the point at which the front of the train is just underneath the block signal sign. As described in the previous section ("Vehicle Operation—Head-Up Display for Station Stopping"), there is a driver aid to assist in locating this position. Table A.2 is

a summary of the relationship between the position in the station and the visual cues in the out-the-window view. Table 3 provides a summary of the braking distances from low speed, using full service braking. From the information provided in these tables, the following strategy for accurate station stopping can be determined:

1. Enter the station at about 35 km/hr.
2. Apply full service braking as the vehicle passes through the entrance doorway.
3. When the vehicle slows to 10 km/hr, ease off the brakes to coast.
4. When the middle of the block signal sign intersects the top of the windscreen, re-apply full service braking.

In addition, table A.3 shows that entrance to the station at a speed in excess of 40 km/hr will result in overshoot, and may result in a missed station stop.

Vehicle Operation—Control Modes

There are four basic control modes available with the high-speed rail simulation vehicle: a) manual mode, b) cruise control mode, c) programmed stop mode, and d) autopilot mode. The latter three are considered automatic control modes because part of the vehicle control task is performed by a computer-based control system.

In manual mode, the vehicle operator is responsible for all aspects of vehicle control. Using the combination control lever, the vehicle operator provides all thrust and brake commands required to achieve speed and position control of the vehicle.

In cruise control mode, the automatic control system applies the appropriate level of thrust force to maintain a constant speed setting. The vehicle operator is responsible for determining the proper speed for the conditions, and then setting the cruise control system for that speed.

In programmed stop mode, the automatic control system applies the appropriate level of brake force to stop the vehicle at a specific position. The vehicle operator is responsible for determining the appropriate position to invoke the programmed stop mode.

position (relative to stopping point	visual cue
-110 m (undershoot)	entra..ce doorway is just visible in windscreen
-100 m	HUD (yellow rectangle) appears on windscreen
-55 m	HUD top bar at top of block signal sign
-40 m	HUD top bar across middle of block sign
-28 m	HUD top bar at bottom of block sign
-20 m	HUD top bar midway between bottom of block sign and top of back wall
-12 m	top of block sign at top of windscreen
-9 m	middle of block sign at top of windscreen
-6 m	bottom of block sign at top of windscreen
0	HUD aligned with back wall
+30 m	top of HUD halfway down upper portion (purple) of back wall
+57 m	HUD top bar at top of exit doorway
+73 m	HUD side bars at sides of exit doorway
+90 m	exit doorway is just visible in windscreen
+100 m (overshoot)	HUD display disappears

Table A.2: Relationship Between Vehicle Position and Out-the-Window View

speed (km/hr)	stopping distance (m) (full service braking)
40	110
35	85
30	62
25	44
20	28
15	16
10	8
5	2

Table A.3: Summary of Low-Speed Braking Distances

In the autopilot mode, the automatic control system applies the appropriate level of thrust and brake forces to follow a predetermined speed trajectory. The vehicle operator is responsible for invoking the autopilot control mode. Once vehicle control has been assumed by the automatic control system, all necessary vehicle control commands are provided by the control system. The task of the operator is reduced to monitoring the vehicle and wayside systems, looking for potential problems in operation.

Vehicle Operation—Traction System

In general, high-speed trains are propelled by electric motors, called traction motors. Power for the traction motors is fed from a high-voltage line, usually on overhead wires. The power is then passed through a motor controller, which governs the amount of power supplied to the traction motors based on the control command of either the vehicle operator or automatic control system.

In the manual operating mode, the vehicle operator is in direct control of the traction power via the combination control lever. Moving the lever forward increases the level of power, and, consequently, the acceleration of the vehicle. When the control lever is in the center position, no tractive power is provided, and the vehicle coasts. Moving the lever back increases the braking force.

In the three automatic modes (autopilot, cruise control, and programmed stop),

the level of tractive power is determined automatically by the control system, and a control command is provided to the traction motor controller.

When tractive power is commanded, either manually or automatically, the motor controller provides electrical power to the traction motors. The tractive force provided by the motor is proportional to the current through the motor windings. The dashboard display includes four current meters (ammeters), which display the level of current through each of the four traction motors. In manual mode, these displays will respond directly to the input at the combination control lever, while in automatic mode, they provide a mechanism for observing the operation of the automatic systems.

The traction motors are protected by circuit breakers, which will interrupt the flow of electrical power to the motors if a failure condition is detected. Each of the four motors has a separate circuit breaker. Under certain circumstances, the circuit breaker for a single traction motor can be tripped, which results in no power being provided to the traction motor. The occurrence of this event can be observed through the ammeters—when one (or more) of the ammeters does not respond with the others, the circuit breaker for that traction motor has tripped and must be reset.

The procedure for resetting the circuit breaker is as follows: a) Remove all power from the other traction motors, by moving the combination control lever to a coast or brake position. b) Depress the appropriate traction motor reset switch (F1 through F4 on the control panel). c) Apply tractive power manually, using the combination control lever. d) Resume the control mode previously in use. If any of the traction motor reset switches are depressed while power is applied, a safety system causes all of the traction motor circuit breakers to trip, preventing motor overload. In this event, all of the circuit breakers must be reset to resume proper operation.

Vehicle Operation—Brake System

Train brakes utilize air pressure, which is stored in tanks on the locomotive. Under non-braking circumstances, the pressure in the tanks prevent the brakes from engaging. When the brakes are applied, pressure is released from the tanks, causing the

brakes shoes to contact the rotating surfaces and resulting in a force which slows the vehicle. An important variable to be monitored by the vehicle operator is the brake tank pressure.

The braking system has two separate modes of operation, service braking and emergency braking. During normal operation, the vehicle operator uses service braking to apply various levels of braking force to the vehicle. In this mode, the level of service brake application is controlled by the combination control lever. The level of braking force can be varied continuously throughout the available range. Application of the maximum available braking force is known as full service braking.

In the emergency braking mode, all of the pressure in the brake system is released, resulting in the maximum possible brake force. In general, this is not a desirable event, as the forces generated result in severe deceleration, which can damage equipment and can cause discomfort or injury to passengers. The vehicle operator can command application of the emergency brake via a control switch on the instrument panel. Also, in certain operational modes, the emergency brake will be applied as a result of dangerous conditions or improper control actions. Such application of the emergency brake is known as a penalty application.

During application of the emergency brakes, the emergency brake indicator will be lit (red). Once the emergency brakes have been applied, they cannot be released until the vehicle comes to a complete stop. When the vehicle is stopped, the control lever is pulled back to a position which results in application of the service brake, and the emergency brake reset switch is depressed. At that point, the emergency brake indicator light will be extinguished, and the vehicle will be ready to continue with normal operation.

The pressure in the brake tanks is indicated by the brake tank pressure gauges, which are located on the instrument panel. In the train simulation, there are two brake tanks, one each for two separate halves of the brake system. The corresponding gauges are round dial gauges, calibrated in units of pounds per square inch (psi), with a range from 0 to 100 psi. The normal reading when the brakes are not applied (i.e., the nominal high pressure) is approximately 98 psi. When full service braking is

applied, the pressure drops to approximately 22 psi, and the pressure further drops to 0 psi under emergency braking.

If there is a failure in the braking system, one or both of the tanks may show a reduction in tank pressure. This situation will result in the brakes being applied without being commanded by either the operator or the control system. The procedure for rectifying this situation is to switch to an alternate brake compressor. This is accomplished by depressing the brake compressor switch (F10 key). The pressure in the faulty tank will then rise to the appropriate level.

Vehicle Operation—Cruise Control Operation

The function of the cruise control system is to maintain a constant vehicle speed. The vehicle operator invokes the cruise control by depressing the cruise control enable switch (F5 key) while the vehicle is traveling at the desired speed.

When the cruise control mode is enabled, the cruise control indicator light (green) is illuminated, and the yellow pointer on the speedometer indicates the set speed. When the cruise control is first selected, the control system will adjust itself to determine the proper level of thrust force required to maintain the selected speed. As a result, there is a small amount of "hunting" around the set speed at first. The system then settles down to the set speed, and small fluctuations in the motor current indicate that the control system is operating.

The vehicle operator can alter the set speed by depressing the "up-arrow" and "down-arrow" keys on the keyboard. With the depression of each key, the set speed is adjusted up (or down) by 1 km/hr. This feature allows the operator to "fine-tune" the set speed. When the operator adjusts the set speed down to a lower speed, the vehicle will coast down to the lower speed.

From cruise control mode, the vehicle operator can either select manual mode or programmed stop mode. By pulling back on the combination control lever, the braking system is actuated, and the vehicle returns to manual mode. The vehicle operator can also directly select programmed stop mode from cruise control mode.

For reasons of safety, application of the brakes will always disengage the cruise

control system. As a result, it is not possible to engage the cruise control system when the brakes are in use. If the operator attempts to engage the cruise control system while the brakes are applied, the system will not respond to that mode command, and the vehicle will remain in manual mode.

Vehicle Operation—Programmed Stop Operation

The function of the programmed stop system is to bring the vehicle to a smooth, controlled stop at a predetermined location (specifically, at the end of the currently-occupied block). The vehicle operator invokes the programmed stop function by depressing the programmed stop enable switch (F6 key) while the vehicle is traveling at a steady speed which is less than 80 km/hr.

The programmed stop mode must not be invoked when the vehicle is traveling at a speed greater than 80 km/hr. If the vehicle speed is greater than 80 km/hr, activating this mode will result in a penalty application of the emergency brakes. In addition, the programmed stop controller must keep track of the distance between the train and the stopping position (the end of the current block). If the programmed stop mode is invoked while the vehicle is too close to the end of the block to be stopped using full service braking, the emergency brakes will be applied in an attempt to stop the vehicle before the end of the block.

When the programmed mode is enabled, the programmed stop indicator light (orange) is illuminated. The vehicle operator can return to manual mode by pulling back on the combination control lever to activate the braking system.

For reasons of safety, application of the brakes will always disengage the programmed stop control system. As a result, it is not possible to engage the programmed stop mode when the brakes are in use. If the operator attempts to engage programmed stop operation while the brakes are applied, the system will not respond to that mode command, and the vehicle will remain in manual mode.

Vehicle Operation—Autopilot Operation

The function of the autopilot system is to perform all thrust and brake commands required to follow a pre-determined speed trajectory. The vehicle operator invokes the autopilot function by depressing the autopilot enable switch (F7 key) while the vehicle is in motion. For best performance, the vehicle must be traveling at a speed greater than 10 km/hr when the autopilot is activated.

When the autopilot mode is enabled, the autopilot indicator light (blue) is illuminated. In addition, the pre-determined speed setting is indicated by the yellow pointer on the speedometer. The vehicle operator can return to manual mode by pulling back on the combination control lever to activate the braking system.

For reasons of safety, application of the brakes will always disengage the autopilot control system. As a result, it is not possible to engage the autopilot mode when the brakes are in use. If the operator attempts to engage the autopilot system while the brakes are applied, the system will not respond to that mode command, and the vehicle will remain in manual mode.

Vehicle Operation—Alerter System

The alerter system is a safety system on the train, which reduces the risk of accidents which are due to operator incapacitation or inattention. The principle behind the system is the requirement for periodic input from the vehicle operator, to determine if s/he is still functional at the controls. If the operator does not respond in a reasonable amount of time, the system assumes that s/he is incapacitated, and applies the emergency brakes to stop the vehicle.

The alerter system, as implemented in the high-speed rail simulation, is similar to those used internationally in actual rail systems. If the operator does not depress the alerter response button ("escape" key) within a period of 42seconds from the last depression of that button, the system issues a warning reminding the operator to do so. The warning consists of a flashing yellow indicator light and an audible chime. If the operator does not respond within 10 seconds of the onset of the warning, the

system assumes that the operator is incapacitated and applies the emergency brakes. In this scenario, both the alerter warning light and emergency brake light will be illuminated.

One type of alerter system, known as a smart alerter, will acknowledge other command actions by the operator as a response to the alerter system. For example, if the alerter issues a warning and the operator pulls back on the combined control lever to apply the brakes, the alerter system interprets that braking command as a response, and ceases to issue the warning. This type of system presents less of a workload to operators under hectic conditions. The alerter system implemented in the simulation system is a smart alerter system.

Vehicle Operation—ATP System

The automatic train protection system (ATP) is a safety system on the train. Its function is to reduce the risk of accident by preventing an overspeed condition to occur. An overspeed condition is defined as operation of the vehicle at speeds in excess of either the civil speed limit or the signal speed limit.

The ATP system continuously monitors the speed and position of the vehicle. It also identifies the state of the block signal when a block is entered by the vehicle. Based on the position of the vehicle and the block signal state, the maximum allowable speed is determined. If the vehicle exceeds that speed by no more than 15 km/hr, a warning is issued. The warning consists of a flashing yellow indicator light and an audible chime. If the speed is not reduced to a level less than the limit within a period of 20 seconds, or if the speed is greater than 15 km/hr over the limit, a penalty application of the emergency brakes is invoked. In this scenario, both the ATP warning light and emergency brake light will be illuminated.

Vehicle Operation—Door Control

The vehicle operator is responsible for controlling the state of the passenger doors. The doors are to be opened when the vehicle is stopped in the station. In principle, the doors must not be opened at any other point in the system, for the protection of

the passengers.

Door control is accomplished through the door control button (F8 button). The state of the doors is indicated by the door indicator light (red) on the instrument panel. When the light is illuminated, the doors are open. Depressing the door control button while the vehicle is stopped will cause the state of the doors to change—if the doors are open, they will be closed, and if the doors are closed, they will be opened.

A safety system prevents the doors from being opened while the vehicle is in motion. Door control commands while the vehicle is in motion will be ignored. If the vehicle is stopped with the doors open, any attempt to move the vehicle will cause a penalty application of the emergency brake.

Vehicle Operation—In-Cab Signal System

In many rail systems using block signal technology, the signal information is available only on the wayside. This type of system presents two distinct problems: the vehicle operator must remember the state of the signal after the vehicle has passed the block boundary, and there is no indication of the signal level of the next block to allow the operator to make preparatory control commands.

Many contemporary locomotive cab designs include in-cab signals, which are devices in the cab which display the signal level of the block currently occupied. While this solves the problem of remembering the signal state of the current block, it still does not solve the preview problem.

At true high-speed operation (speeds in excess of 200 km/hr), it is virtually impossible to see wayside signals in time to take appropriate corrective action. Therefore, such operations require some form of signal preview as part of the in-cab signal system. In the high-speed rail simulation vehicle, the in-cab signal system is implemented with two color-coded light bars, each containing three lights. The bottom light bar displays the color code of the signal state of the currently-occupied block when the vehicle entered it. The top light bar displays the color code of the signal state of the next block.

Vehicle Operation—Bearing Temperature Display

One of the secondary gauges provided on the instrument panel of the train simulation displays wheel bearing temperature. Train operations have traditionally been concerned about detecting overheated wheel bearings, as an overheated wheel bearing could ultimately seize and lead to derailment. Certain train operations have installed hot box detectors, which are wayside-located boxes that detect a hot bearing as the train passes by.

The wheel bearing temperature system in the train simulation uses sensors on each wheel to measure the temperature. The display then shows the temperature of the hottest bearing among those that are measured. In normal operation, the temperature will rise as the speed of the vehicle increases, and fall to ambient temperature when the vehicle slows down. If a wheel bearing does fail, the wheel bearing temperature gauge will reflect the temperature of the hottest bearing. It does not, however, indicate which bearing is at fault—the vehicle operator must stop the train and locate the faulty bearing (usually through observation of smoke at the wheel).

Vehicle Operation—Trolley Voltage Display

One of the secondary gauges in the instrument panel displays the trolley voltage, which is the voltage available from the power supply grid to the traction motor controllers (nominally 1500 VDC).

The gauge is provided so that the vehicle operator can detect problems with the power supply voltage, which will have implications on the performance of the vehicle. Although the vehicle operator cannot take any action to rectify such an occurrence, s/he would be able to inform the CTC operator, via the communication channel, of the problem. The CTC operator would then contact the appropriate personnel in the power supply department in an effort to rectify the problem.

Vehicle Operation—Acceleration and Braking Performance

Although there are many conceptual similarities between operating a rail vehicle and driving an automobile, the biggest difference occurs in braking and acceleration performance. Because of the very large mass of a rail vehicle (typically in the range of hundreds of tons), the distances required for acceleration and braking are much larger than might be expected. This becomes more of an issue as the vehicle speed grows larger—the stopping distances from a speed of 300 km/hr are on the order of several kilometers.

This situation has several implications. An important safety consideration is that the large stopping distance makes it virtually impossible for an operator to stop in time from high speed for an unexpected track obstruction. The only reasonable solutions to this problem are to reduce the allowable vehicle speeds in risk-prone areas, and to provide additional driving aids, such as obstruction warning systems. In general, operators are not held liable for situations which are beyond their control.

In the simulation system, this issue is addressed by taking care that there are no unexpected hazards that are beyond the operator's control. Grade crossing areas are the only places where track obstructions can occur. These obstructions take the form of highway vehicles crossing the track. In the grade crossing areas, the civil speed limit is reduced to 100 km/hr. These areas include blocks 0, 1, 11, 12, and 13. Within these areas, the vehicle operator is able to detect and react to an obstruction at a highway grade crossing. Highway vehicles that cross the tracks will only do so when there is sufficient distance for them to clear the crossing before the train arrives. Under certain circumstances, however, the highway vehicle may get stuck at the crossing in the path of the train. By design, the point in time at which the vehicle gets stuck will be such that the train operator has sufficient time to detect the obstruction and stop the train before a collision occurs.

Another implication of large braking distances is the necessity for good judgment of braking points. Because the braking distances are so large, proper planning of braking points is essential for a vehicle to be stopped accurately at stations (or other

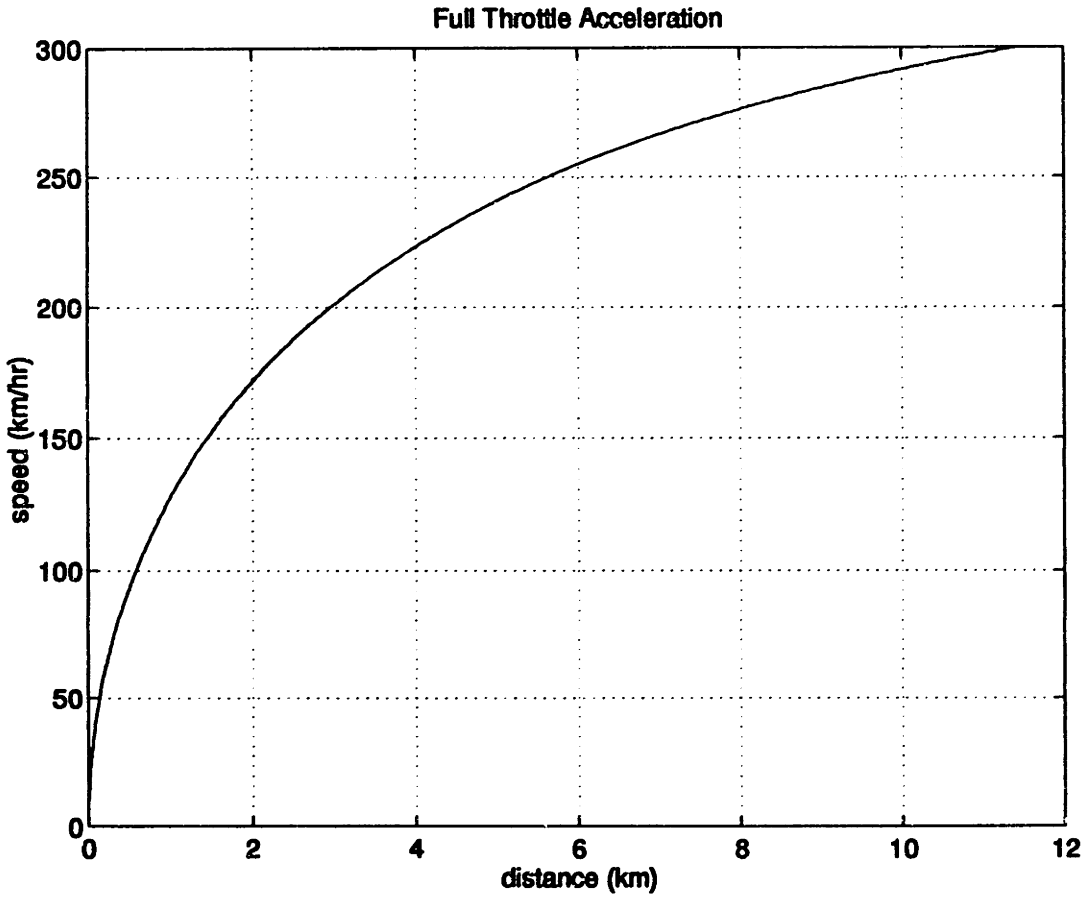

Figure A-4: Full-Throttle Acceleration Profile

stopping points). This is a skill which requires a great deal of practice to master, and represents an important component of operator training programs in actual rail operations. Such training programs typically last for a year or more.

To shorten this learning curve, graphical representations of the vehicle performance curves are shown in figures A-4 through A-6. Figure A-4 displays the full-throttle acceleration profile on level ground. The speed is shown as a function of distance. Form this curve, we can see that the distance required to reach 300 km/hr from a standing stop is approximately 11.5 km.

In figure A-5, the full-service braking profile is shown, again for the case of travel on level ground. From this curve, we can see that the braking distance from 300 km/hr is approximately 5.2 km, which is substantially shorter than the distance required to achieve that speed in the first place (figure A-4). The reason for this difference is twofold: a) peak braking forces are generally higher than peak traction forces, for safety reasons, and b) at higher speeds, the resultant aerodynamic drag works against acceleration, but contributes to deceleration forces.

Note that the braking distance from 100 km/hr is just a little larger than a half kilometer. Obstructions in the grade crossings are visible from the vehicle for a distance of almost one kilometer—therefore, this braking performance provides adequate opportunity for the vehicle operator to detect and react properly to an obstruction in a grade crossing.

The full-service braking profile (figure A-5) can be used to estimate appropriate braking points under manual control. When approaching West Station, there is a civil speed restriction of 100 km/hr for two blocks (4 km) (blocks 0 and 1, which cover the track between kilometer post 4 and the station) prior to the station, to reduce speed for grade crossing safety. Approaching East Station, there is a similar speed restriction for one block (2 km) (block 24, between kilometer post 48 and the station), to ensure that a train does not go through the station at full speed. In addition, there is a speed-restricted region in blocks 11 through 13 (covering the track between kilometer posts 22 and 28), for reasons of grade crossing safety. When approaching these regions, full-service braking from full speed (300 km/hr) should begin 4.5 km

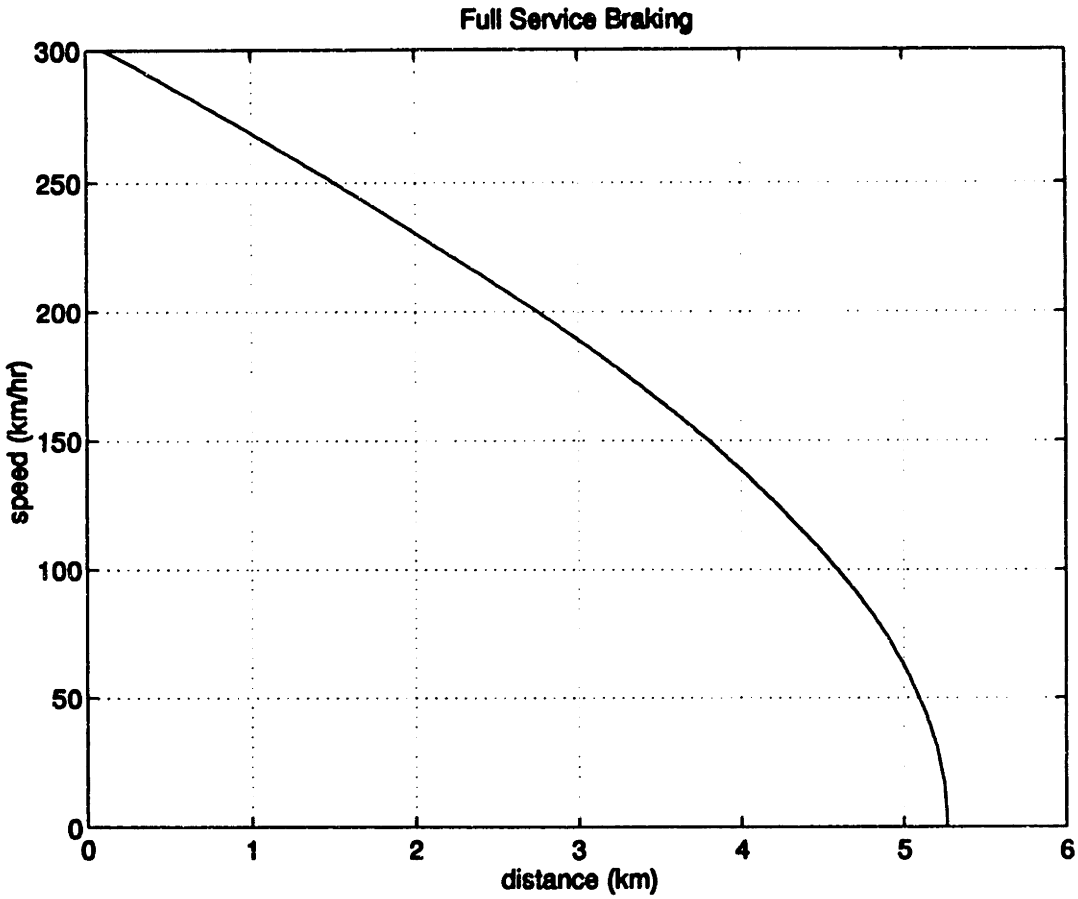

Figure A-5: Full-Service Braking Profile

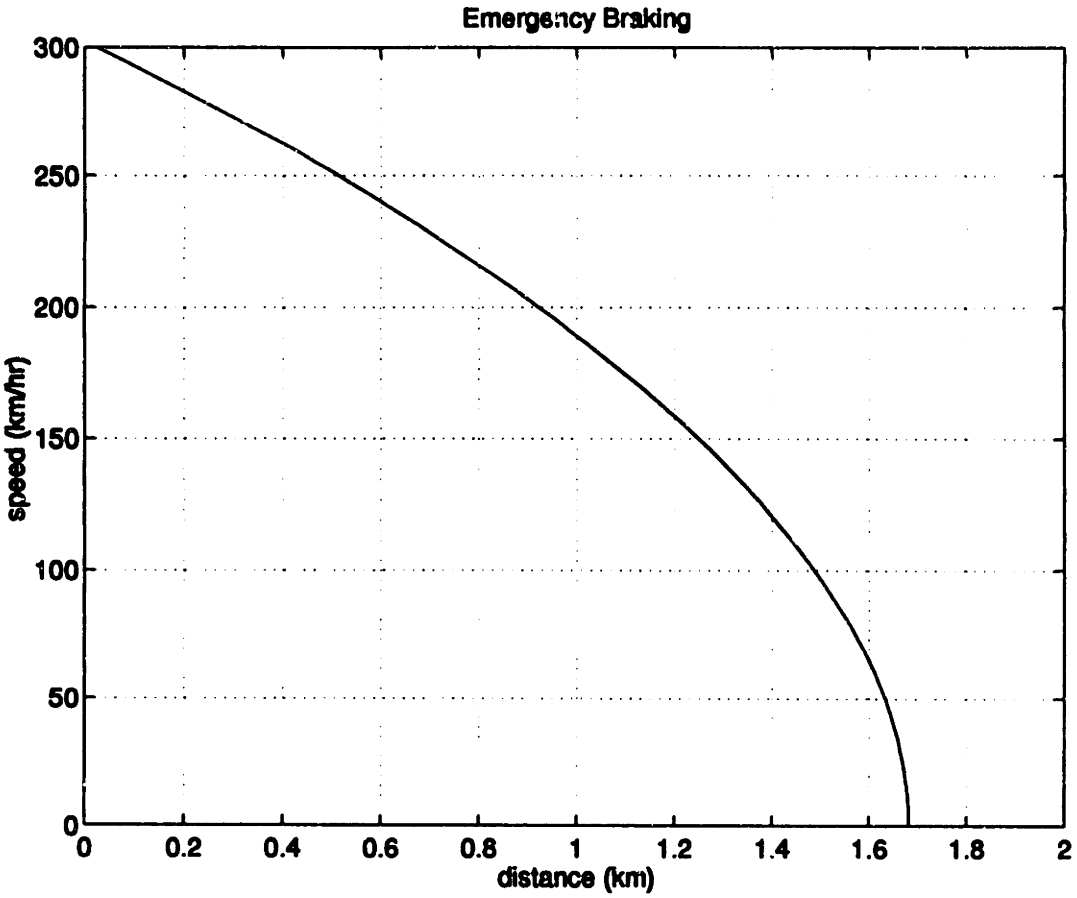

Figure A-6: Emergency Braking Profile

prior to the start of the restricted block, to ensure that the vehicle speed is at or below the restricted speed before entering that block. When approaching the two stations, the vehicle operator can again apply full service braking in the last block when the vehicle is about 1 km from the station, bringing the vehicle to a coasting speed of slightly less than 40 km/hr. As the vehicle passes through the entrance of the station, the vehicle operator can again apply the brakes to stop precisely in the station. The ideal stopping point is the position at which the block signal board is just visible in the windscreen. If the operator stops the vehicle too soon, the vehicle must be brought forward to this point so that the passenger doors may be opened. The operator may legally stop the vehicle forward of this point, as long as the exit doorway and back wall of the station are still visible through the windscreen.

Figure A-6 shows the braking profile under emergency braking conditions. The distance required to brake from 300 km/hr under emergency stop conditions is only about 1.7 km. However, this braking performance has a high cost—the more severe

deceleration experienced during these stopping conditions is the result of severe forces at the wheels which can result in damaged train equipment, as well as damaged track. In addition, in passenger service, the resultant deceleration can cause passengers to be thrown about in the passenger compartments, opening the possibility of injury. Emergency braking procedures are generally considered to be a last resort. For these reasons that penalty applications of the emergency brake, triggered by the alerter, ATP, or programmed stop systems, should be avoided whenever possible.

System Operation—Communications

Communications in the high-speed rail simulation system is via a broadcast text-based system. An operator in each vehicle, as well as the CTC operator, can enter messages through the computer keyboard. When the message is completed (either with a carriage return or when the message length reaches 80 characters), it is sent out over the network and received by all other active simulation elements. Thus, everybody hears everybody.

The protocol used for these communications aids in reducing any confusion in the messages. Specifically, the message sender identifies the intended recipient of the message, adds the body of the message, and terminates with self-identification. All identification is with vehicle identifiers (vehicle ID numbers are issued when the operator starts a shift).

For example, the following exchange might take place between the CTC and a train operator in vehicle E48401:

"E48401, request position, CTC."

"CTC, block 5, kmpost 11, destination East Station, E48401."

In this exchange, the CTC operator requests a position update from operator E48401, who responded with a summary of the current vehicle position. All of the active vehicles in the simulation heard this exchange. The exchange format could be shortened to the following, without any loss of information:

"E48401, req pos, CTC."

"CTC, blk 5, km 11, dest E St, E48401."

Messages Initiated by Vehicle Operator	Message Shorthand Form	Response From CTC Operator
announce departure	"CTC, depart <station>. <vehID>"	"<vehID>, ack depart <station>, CTC"
announce arrival	"CTC, arrive <station>, <vehID>"	"<vehID>, ack arrive <station>, CTC"
inform CTC of vehicle failure	"CTC, <brake -- motor> fail, fixed, <vehID>"	"<vehID>, ack fail, CTC"
inform CTC of obstruction	"CTC, obstruct at <cross num>, <vehID>"	"<vehID>, ack obstruct, CTC"
inform CTC of collision	"CTC, collide at <cross num>, <vehID>"	"<vehID>, ack collide, CTC"
request break	"CTC, req break at <station>, <vehID>"	"<vehID>, break req accept, CTC" "<vehID>, break req deny, CTC"

Table A.4: Standard Communications Message Initiated by Vehicle Operator

Tables A.4 and A.5 contains a summary of the most frequently-used messages and responses used in the system. Although any text is allowable, these tables summarize the set of most frequently used messages and responses, in shorthand form to minimize the necessary typing.

Most of the messages included in these tables represent typical information which might be conveyed between a vehicle operator and a CTC operator in an operational rail system. In a typical rail system, there is very limited provision for exchange of data between vehicles and centralized control. Although this is changing as technology progresses, voice communications over radio still represent a significant method for transferring state information between the vehicles and the CTC.

System Operation—Schedules

During a shift of operation, the vehicle operator is required to make three round trips in the shuttle operation. Each round trip starts at West Station, proceeds to East Station, reverses direction, and returns to West Station. Each trip leg (one-way) is scheduled to take twenty minutes. When the train reaches the destination

Messages Initiated by CTC Operator	Message Shorthand Form	Response From Vehicle Operator
request vehicle position	"<vehID>, req pos, CTC"	"CTC, km <num>, <vehID>"
request vehicle status	"<vehID>, req veh status, CTC"	"CTC, status OK, <vehID>" "CTC, <brake—motor> fail, <vehID>" "CTC, obstruct at <cross>, <vehID>" "CTC, collide at <cross>, <vehID>"
request automation mode status	"<vehID>, req mode status, CTC"	"CTC, man mode, <vehID>" "CTC, cruise mode, <vehID>" "CTC, pstop mode, <vehID>" "CTC, auto mode, <vehID>"

Table A.5: Standard Communications Messages Initiated by CTC Operator

station, the vehicle is stopped and the doors are opened for a period of one minute, to allow the passengers on-board to disembark. The train is then routed around the reversing loop to change direction. The looping procedure requires approximately seven minutes. When the train arrives at the same station in the opposite direction, it is again stopped, and the doors are again opened for a period of two minutes, to allow new passengers to board. The total scheduled round trip time is one hour. A printed schedule is provided to the operator for each shift.

Under certain circumstances, there may be deviations from the prescribed schedule. In this case, it is the responsibility of the CTC operator to adjust the schedules accordingly. The vehicle operator must wait for CTC instructions before departing a station at any time which is not in compliance with the prescribed schedule.

System Operation—Role of CTC Operator

The central traffic control operator (CTC operator) is the coordinating element in the rail system. S/he must monitor the positions and speeds (when possible) of the vehicles in the system, and adjust operating parameters of the system in order to

achieve the system goals. The CTC operator also represents the coordination point between the vehicles, as well as with "outside" agencies (such as fire, police, power, maintenance, etc.).

Given this perspective, the CTC operators have a level of jurisdiction which is higher than the wayside signals, the operating rules, and any other influence. In short, when there is a conflict between influences that govern the action taken by a vehicle operator, the CTC operator is always the highest authority. Therefore, CTC operators are supervisors, with respect to the vehicle operators, and vehicle operators are required to follow all directions given by the CTC operators. This supervisory role has a higher level of responsibility than the task of monitoring.

System Operation—Obstruction Hazards

As mentioned in the braking performance section, the track used in the simulation system has highway grade crossings. These are points in the rail system where highway roads and rail tracks intersect. In total, there are five grade crossings, one each located in blocks 0, 1, 11, 12, and 13. At each grade crossing, highway vehicles (cars) can cross in front of the train from either direction. These vehicles are visible at over a half kilometer distance.

In each of the blocks containing grade crossings, there is a civil speed limit of 100 km/hr. This means that, regardless of the signal level in the block, the maximum speed of a vehicle in the block is 100 km/hr. The ATP system is programmed to detect when these blocks are entered, and will use this civil speed limit in its restriction rules.

Traffic at the grade crossings arrives according to a probabilistic process. A car will proceed across the crossing only if there is sufficient distance to clear the crossing before the train. (In other words, a car will not proceed if there is not enough room.) However, it is possible for a car to become disabled as it is crossing the tracks, which will result in an obstruction for the train. In this event, the train operator must bring the train to a stop before the intersection. If the train is not stopped in time, a collision will occur, which will be indicated by a cracked windscreen.

It is important that the train operator be able to quickly assess the crossing

traffic and determine whether the train must be stopped. On one hand, a collision is a major event, and will result in a significant delay in operation. On the other hand, stopping the train unnecessarily will also cause delays in service. It is up to the vehicle operator to evaluate the situation and determine the best course of action under these constraints.

If a collision occurs, the vehicle operator must stop the vehicle and immediately contact the CTC operator to report the collision. In the case of a collision, the windscreen will appear "cracked." This crack will remain for the remainder of the shift. For each of three collisions that might occur within a shift, the windscreen will become progressively more "cracked."

System Operation—Operator Performance Requirements

In order to assure that each vehicle operator is capable of adequately controlling the train, the performance of each operator (test subject) is monitored throughout the test sessions. As an incentive, there is a bonus system which provides monetary rewards for good performance. If operator performance does not fall within certain minimum criteria, penalties may be assessed. At the end of each experiment session, the subject's performance is evaluated with regard to bonuses and penalties.

By decree of the Federal Code of Regulations, Number 49, Part 240, the performance of rail vehicle operators (locomotive engineers) is regulated such that any willful violation of speed restrictions or signal indications is punishable with both a monetary fine and a loss of certification (which may be permanent or temporary, depending on the circumstances). The monetary fines are substantial, and can range from a minimum of $250 to a maximum of $20,000. In short, these violations are considered serious offenses, and are not tolerated.

Because of these regulations, speed and signal compliance are considered key performance items. In the simulation system, violations are defined by ATP-induced or signal-induced penalty applications of the emergency brakes. During the road test period of the training phase, such a violation will result in disqualification of the subject from further participation. In this case, the subject will be paid for training

period. During the experiment phase, the first violation in a shift will result in a penalty of 100,000 bonus points. If a second violation occurs in the same shift, the subject will be disqualified from further participation in the experiment. In this case, payment will cover the sessions that have been completed to date.

In addition, willful circumvention of the vehicle safety systems, such as the alerter system, will not be tolerated. If a subject is found to have bypassed these safety systems, s/he will not be permitted to complete the experimental tests. In this case, payment will cover the sessions that have been completed to date.

Other key operator performance items include station stopping accuracy, schedule maintenance, and response to emergency (or failure) situations. In general, good performance in these areas will result in the award of bonus points, which result in an increase in payment for that session. The bonus points schedules are shown in tables A.6 and A.7, and the penalty points schedules are shown in tables A.8 through A.10.

Table A.6 shows the bonus points that result from station stopping performance. The stopping point is defined as the first point at which the vehicle stops, in the vicinity of the station. The closer the stop occurs to the desginated stopping point, the more bonus points are awarded. The bonus points are distributed to favor undershoot (i.e., it's preferable to stop before the designated point than after).

Table A.7 shows the bonus points that result from schedule accuracy. Because there may be emergency situations which will affect schedule maintenance (such as an obstruction or collision), the schedule is adjusted for such occurrences. As with station stopping accuracy, the bonus point distribution shows a preference for early arrivals over late arrivals.

Table A.8 shows the bonus point schedule associated with failure and emergency response. In general, you are evaluated in terms of the response time to the failure, as well as the accuracy ("correctness") of the first response action. The actual events used to evaluate the accuracy depends on the specific emergency being evaluated.

Tables A.9 and A.10 show the penalties that can be levied for poor performance. Table A.9 identifies penalties that result from explicit violations of the operating rules.

station stop accuracy	bonus points
more than 10m before stop point deviation < -10.0 (undershoot)	0
$-10.0 \leq$ deviation < -8.0	+400
$-8.0 \leq$ deviation < -6.0	+800
$-6.0 \leq$ deviation < -4.0	+1200
$-4.0 \leq$ deviation < -2.0	+1600
$-2.0 \leq$ deviation $\leq +2.0$	+2000
$+2.0 <$ deviation $\leq +3.0$	+1600
$+3.0 <$ deviation $\leq +4.0$	+1200
$+4.0 <$ deviation $\leq +5.0$	+800
$+5.0 <$ deviation $\leq +6.0$	+400
more than 6m beyond stop point deviation $> +6.0$ (overshoot)	0

Table A.6: Bonus Point Schedule for Station Stopping Accuracy

schedule accuracy	bonus points
more than 26 secs early deviation < -26.0	0
$-26.0 \leq$ deviation < -22.0	+400
$-22.0 \leq$ deviation < -18.0	+800
$-18.0 \leq$ deviation < -14.0	+1200
$-14.0 \leq$ deviation < -10.0	+1600
$-10.0 \leq$ deviation $\leq +8.0$	+2000
$+8.0 <$ deviation $\leq +12.0$	+1600
$+12.0 <$ deviation $\leq +16.0$	+1200
$+16.0 <$ deviation $\leq +20.0$	+800
$+20.0 <$ deviation $\leq +24.0$	+400
more than 24 secs late $+24.0 <$ deviation	0

Table A.7: Bonus Point Schedule for Schedule Accuracy

response time (sec)	bonus/penalty points
less than 2.0	+1000
more than 2.0, less than 5.0	+750
more than 5.0, less than 10.0	+500
more than 10.0, less than 20.0	+250
more than 20.0	0

response accuracy, obstruction	
correct braking action	+1000
incorrect braking action	+500
error (other command action)	0
safety hazard (door open command)	−1000

response accuracy, brake failure	
correct action (switch brake pump)	+1000
incorrect command action	0
safety hazard (estop, door open commands)	−1000

response accuracy, motor failure	
correct action (brake, then reset failed motor)	+1000
partially correct action (depress reset button for wrong motor, or failure to apply brakes first)	+500
incorrect command action (other action)	0
safety hazard (estop, door open commands)	−1000

Table A.8: Bonus/Penalty Point Schedule for Emergency Response

infraction	penalty points
penalty application for speed or signal compliance violation	−50,000
passenger doors opened outside of station bounds (more than 20m undershoot or 100m overshoot)	−10,000
unnecessary application of emergency brake (no emergency present)	−5,000
station overrun	−2,000

Table A.9: Penalty Point Schedule for Violations

The most significant is a penalty application of the emergency brakes, resulting from either a speed violation (via the ATP system) or a signal violation. Both of these are considered serious offenses, and the resultant penalty is substantial. The next penalty is for opening the passenger doors when the vehicle is outside the station. This action is considered a serious compromise of passenger safety, and is penalized accordingly. The third penalty is for unnecessary use of the emergency brake. Use of the emergency brake takes a heavy mechanical toll on the train systems, and results in a very uncomfortable ride for the passengers. Gratuitous use of the emergency brake is inappropriate, and the penalty for unnecessary use is significant enough to outweigh any benefit that might be obtained (such as improving the station stopping or schedule accuracy). The fourth penalty is imposed if the operator fails to stop the vehicle within the station. In this event, the doors must not be opened. In addition, the vehicle operator must notify the CTC operator of the situation, and await further instructions from the CTC operator.

Table A.10 shows the penalty points that occur if there is a collision with a highway vehicle at a grade crossing. The intent of this schedule is to impart a sense that collisions at higher velocities are more serious—in the event that a collision is inevitable, the operator should do as much as possible to reduce the impact of that collision by reducing the vehicle as much as possible.

collision impact speed (km/hr)	penalty points
between 60 and 100	−1000
between 40 and 60	−750
between 20 and 40	−500
between 0 and 20	−250

Table A.10: Penalty Point Schedule for Grade Crossing Collisions

After the total bonus points are computed for a shift, the bonus points are converted into a pay bonus, at the rate of one dollar for each ten thousand points. There are a total of eleven station stops per shift, for a maximum possible 44,000 points per shift for station stopping performance. Because the number of failures and obstructions will vary from shift to shift, it is not possible to determine beforehand the maximum possible bonus points that are available. However, in most cases, there will be sufficient opportunity for an excess of 50,000 bonus points, which will yield an equivalent pay rate in excess of $10 per hour.

Appendix B

High-Speed Rail Simulator Training—Review Quiz

Instructions: For each question, circle the letter of the answer you feel best answers the question.

1. "Kilometer posts" are located where?

   (a) on the wayside, at tenth-kilometer intervals

   (b) on the vehicle instrument panel

   (c) on the wayside, at kilometer intervals

   (d) on a monitor screen in the CTC operations center

2. Which of the following is considered an automatic mode?

   (a) programmed stop

   (b) emergency stop

   (c) ATP warning

   (d) alerter warning

3. If the signal level of the upcoming block is full yellow (YYY), what is the speed limit in that block?

   (a) 15 km/hr

   (b) 150 km/hr

   (c) 80 km/hr

   (d) 230 km/hr

4. What is the expected one-way travel time between East Station and West Station?

   (a) 5 minutes

   (b) 20 minutes

   (c) 15 minutes

   (d) 25 minutes

5. How many signal levels are used?

   (a) one

   (b) seven

   (c) four

   (d) five

6 How many control modes are available on the train?

   (a) none

   (b) four

   (c) three

   (d) ten

7. When full service braking is applied, what is the typical tank pressure?

   (a) 1000 psi

   (b) 100 psi

   (c) 0 psi

   (d) 22 psi

8. Where are block signals located?

   (a) at the entrance to every block

   (b) in the middle of every block

   (c) every 50 m

   (d) on the back of the preceding train

9. What is the expected distance to accelerate to 300 km/hr from a standing stop on level ground?

   (a) 11.5 km

   (b) 21.2 km

   (c) 4.5 km

   (d) 5.7 km

10. Which category best describes the type of system being operated?

   (a) shuttle service

   (b) subway

   (c) commuter rail

   (d) long-haul freight

11. There are four electric traction motors. How many circuit breakers in total are used to protect these traction motors?

    (a) eight

    (b) four

    (c) one

    (d) none

12. How is the programmed stop mode disabled?

    (a) by manually applying the brakes

    (b) by depressing the cruise control enable button

    (c) by depressing the programmed stop enable button

    (d) by waiting until the vehicle comes to a stop

13. During system operation, how many trains are simultaneously in use?

    (a) four

    (b) one

    (c) ten

    (d) two

14. In autopilot mode, the control system maintains the proper speed through which control mechanism?

    (a) thrust only

    (b) braking only

    (c) both thrust and braking

    (d) magnetic levitation

15. How is the cruise control mode enabled?

    (a) by depressing the cruise control button while braking to the desired speed

    (b) by depressing autopilot and cruise control buttons simultaneously

    (c) by holding the vehicle at a steady speed under manual control

    (d) by depressing the cruise control button while traveling at the desired speed

16. What is the relative status of the CTC operator, from the perspective of a train operator?

    (a) supervisor

    (b) subordinate

    (c) monitor

    (d) peer

17. If the vehicle speed exceeds the maximum allowable speed by more than 15 km/hr, the ATP system does what?

    (a) applies the emergency brake

    (b) applies more thrust

    (c) limits the effectiveness of manual thrust commands

    (d) applies the service brake

18. What is the purpose of the block signal system?

    (a) for vehicle operator to determine the number of vehicles allowed in the block

    (b) for vehicle operator to determine the maximum allowable speed in the block

    (c) give vehicle operators something to do

    (d) for vehicle operator to determine the number of passengers allowed on board

19. In a block signaling system, how many trains are allowed to occupy a single block at the same time?

    (a) two

    (b) four

    (c) none

    (d) one

20. How many different braking modes are available to the vehicle operator?

    (a) one

    (b) three

    (c) five

    (d) two

21. If the vehicle operator moves the vehicle when the doors are still open, what happens?

    (a) the motor circuit breakers are tripped

    (b) the passengers are warned over the intercom

    (c) the emergency brakes are applied

    (d) the doors close automatically

22. What is the expected stopping distance from 300 km/hr to 100 km/hr under full-service braking on level ground?

    (a) 21.2 km

    (b) 11.5 km

    (c) 4.5 km

    (d) 5.7 km

23. What warning does the alerter system give the vehicle operator before a penalty application is imposed?

    (a) chime only, for 5 seconds

    (b) electric shock through the seat, for 5 seconds

    (c) flashing light only, for 15 seconds

    (d) flashing lights and chime, for 10 seconds

24. How does the vehicle operator know the speed of the train?

    (a) from the CTC operator

    (b) from the speedometer, on the vehicle instrument panel

    (c) from the brake pressure gauge, on the vehicle instrument panel

    (d) from the block signaling system

25. Where are CTC operators located?

    (a) in a maintenance shed

    (b) in small booths along the wayside

    (c) in a centralized operations center

    (d) in the last car of each train

Appendix C

Train Schedule

The following page lists the train schedule used by the subjects in the control automation experiment (section 5.2). The stations listed in the schedule correspond to the stations shown in figure A-1.

Train Schedule

00:05:00	Depart West Station
00:23:30	Arrive East Station — discharge passengers
00:26:00	Depart East Station — loop around
00:33:00	Arrive East Station — board passengers
00:35:00	Depart East Station
00:53:00	Arrive West Station — discharge passengers
00:56:00	Depart West Station — loop around
01:03:00	Arrive West Station — board passengers
01:05:00	Depart West Station
01:23:30	Arrive East Station — discharge passengers
01:26:00	Depart East Station — loop around
01:33:00	Arrive East Station — board passengers
01:35:00	Depart East Station
01:53:00	Arrive West Station — discharge passengers
01:56:00	Depart West Station — loop around
02:03:00	Arrive West Station — board passengers
02:05:00	Depart West Station
02:23:30	Arrive East Station — discharge passengers
02:26:00	Depart East Station — loop around
02:33:00	Arrive East Station — board passengers
02:35:00	Depart East Station
02:53:00	Arrive West Station — discharge passengers

Appendix D

Exit Questionnaire

The questionnaire on the following page was given to the experimental subjects after the test sessions had been completed. The answers provided on this questionnaire were used to determine the subjective evaluation of the control automation.

Exit Questionnaire

Please answer the following questions.

1. Rate the levels of automation in order of preference (use "1" for the automation level you liked the most, "3" for the automation level you liked the least)?

   ___ full automation (autopilot)

   ___ partial automation (cruise control and programmed stop)

   ___ no automation (manual control)

2. Rate the levels of automation according to level of "awareness" (use "1" for the automation level in which you felt the most aware, "3" for the automation level in which you felt the least aware)?

   ___ full automation (autopilot)

   ___ partial automation (cruise control and programmed stop)

   ___ no automation (manual control)

3. Do you feel that the training process provided adequate preparation for the test task?

   ___ yes ___ no

4. Any other comments? Critical comments are appreciated.

Appendix E

Vehicle Dynamics

A design goal of the high-speed rail simulation system was to provide the highest quality vehicle dynamics possible, within the constraints presented. Both of the simulated vehicles implemented to date have met this goal.

Because the experimental work described in this document is focused on the control automation experiment and the simulation elements which supported that experiment, this appendix is directed mainly on description of the vehicle dynamics implemented in the control-aided vehicle simulation. However, the implementation is based on the previous implementation which is part of the display-aided vehicle, which preced the control-aided vehicle; as a result, a brief review of the display-aided implementation is included here. A more detailed description of the vehicle dynamics implementation in the display-aided vehicle may be found in [1].

Control-Aided Vehicle Simulation

The vehicle dynamics model implemented in the control-aided train simulation is based on a single lumped mass model of the vehicle. Since the vehicle is a passenger train traveling at high speed, this is a reasonable assumption--the complex effects of a long train at lower speed, such as found with freight trains, are considered to be negligible. The basic state equations

$$\frac{dx}{dt} = v(t) \tag{E.1}$$

$$\frac{dv}{dt} = \frac{F_{tot}(t)}{m} \tag{E.2}$$

are used. The total applied force F_{tot} is expressed as

$$F_{tot} = F_{trac} - F_{brake} - F_{aero} - F_{rolling} + F_{grade} - F_{curve} \tag{E.3}$$

where F_{trac} is the force applied by the traction motors, F_{brake} is the braking force, F_{aero} is the aerodynamic resistance, $F_{rolling}$ is the rolling resistance, F_{grade} is the force due to gravity when on a grade, and F_{curve} is the resistive force due to curvature of the track.

The force applied by the traction motor is determined by the equation

$$F_{trac} = \frac{K_m}{R_m}\left[(C_t V_t) - (K_m v)\right] \tag{E.4}$$

where K_m is the motor constant, C_t is the relative throttle position (a value between 0 and 1), V_t is the available voltage, v is the vehicle speed, and R_m is the motor resistance. This model corresponds to the model of a simple electric motor. The coefficients are chosen such that the characteristics of the motor approximate the power curves used in the tractive force look-up table used with the display-aided vehicle. There is no provision for regenerative braking—when the brakes are applied, the motors are open-circuit, and there are no resistive forces.

The braking force is expressed with the equation

$$F_{brake} = m b_{max}\left[\frac{P_{max} - P}{P_{max} - P_{min}}\right] \tag{E.5}$$

where m is the vehicle mass, b_{max} is the full service braking deceleration, P_{max} is the maximum pressure in the system, P_{min} is the minimum operating pressure in the system, and P is the current system pressure. The constants P_{max} and P_{min} are arbitrary, chosen such that the quantity

$$b_{max}(P_{max}/(P_{max} - P_{min}))$$

is equal to the emergency braking deceleration. The system pressure P is determined from the throttle lever position, such that full service braking results in $P = P_{min}$ and no braking results is:

$$P = P_{max}$$

Modulation of the brake lever results in a change in system pressure, with a time delay. In the event there is a brake failure, the pressure of the system is slowly leaked until either the pressure reaches zero or the operator takes corrective action.

The aerodynamic resistance is computed as

$$F_{aero} = C_{aero} v^2 \tag{E.6}$$

where C_{aero} is a coefficient of aerodynamic drag. The rolling resistance is computed as:

$$F_{rolling} = C_{r1} + C_{r2} v$$

Taken together, these correlate to the rolling resistance in the other simulator. The coefficients C_{aero}, C_{r1}, and C_{r2} are chosen such that the resultant combined resistance force approximates the force curve in the look-up table for the other simulator.

The force due to gravity when on a grade is computed as:

$$F_{grade} = -mg \sin(\theta)$$

where g is the gravitational acceleration and θ is the grade angle of the track, with uphill represented as a positive value.

Finally, the resistive force due to curvature of the track is computed as:

$$F_{curve} = C_f m v^2 c$$

where C_f is the coefficient of friction, and

$$c = 1/r$$

is the curvature of the track.

Display-Aided Vehicle Simulation

The vehicle dynamics calculations are based on the following equations:

$$\frac{dx}{dt} = v(t) \tag{E.7}$$

$$\frac{dv}{dt} = \frac{F(t) - R(x,v)}{m} \tag{E.8}$$

$$R(x,v) = R_{rolling} + R_{grade} + R_{curvature} \tag{E.9}$$

$$R_{grade} = mg\sin\theta \tag{E.10}$$

$$R_{curvature} = c\frac{mv^2}{r} \tag{E.11}$$

where $x(t)$ is the distance traveled along the path, $v(t)$ is the vehicle speed, $F(t)$ is the applied tractive and braking forces, $R(x,v)$ is the sum of the resistive forces, $R_{rolling}$ is the rolling resistance, R_{grade} is due to the force of gravity, $R_{curvature}$ is resistance due to the curvature of the track, m is the mass of the vehicle, g is gravitational acceleration, θ is the grade angle, c is the coefficient of friction, and r is the radius of the track curve. The coefficient of friction used is 0.4, while the train parameters are taken from the TGV system. For the purposes of these equations, the vehicle is treated as a lumped mass. This is a resonable assumption for a high-speed passenger train, which is much shorter and faster than a freight train. The tractive and braking forces are determined through use of a look-up table, as a function of the throttle lever position. This look-up table is based on the traction motor and brake system characteristics of the TGV locomotive.

Bibliography

[1] Shumei Yin Askey. *Design and Evaluation of Decision Aids for Control of High-Speed Trains: Experiments and Model.* PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 1995.

[2] Shumei Yin Askey and Thomas B. Sheridan. Safety of Ground Transportation Systems—Human Factors Phase II: Design and evaluation of decision aids for control of high-speed trains: Experiments and model. Technical report, US-DOT/RSPA, 1995.

[3] Philip S. Babcock. An introduction to reliability modeling of fault-tolerant systems. Technical Report CSDL-R-1899, Charles Stark Draper Laboratory, 1986.

[4] A. Chapanis. Human engineering. In C. D. Flagle, W. H. Huggins, and R. H. Ray, editors, *Operations Research in Systems Engineering.* Johns Hopkins, 1960.

[5] Code of federal regulations. Number 49, Part 240.

[6] Alvin Drake. *Fundamentals of Applied Probability Theory.* McGraw-Hill, 1967.

[7] Mica Endsley. SAGAT: A methodology for the measurement of situation awareness. Technical Report NOR DOC 87-83, Northrup Corp., Hawthorne, CA, 1987.

[8] Mica Endsley. Situation awareness global assessment technique (SAGAT). IEEE, New York, 1988.

[9] Mica Endsley. Toward a theory of situation awareness in dynamic systems. *Human Factors*, 37(1), 1995.

[10] Paul M. Fitts. Human engineering for an effective air-navigation and traffic-control system. Technical report, National Research Council, Washington, D.C., 1951.

[11] Paul M. Fitts. Functions of men in complex systems. *Aerospace Engineering*, 21(1), 1962.

[12] Thomas W. Forbes, editor. *Human Factors in Highway Traffic Safety Research.* Wiley- Interscience, 1972.

[13] Bernard Friedland. *Control System Design: An Introduction to State-Space Methods.* McGraw-Hill, 1986.

[14] David M. Gaba, Steven K. Howard, and Stephen D. Small. Situation awareness in anethesiology. *Human Factors*, 37(1), 1995.

[15] David I. Gertman and Harold S. Blackman. *Human Reliability and Safety Analysis Data Handbook.* John Wiley and Sons, 1994.

[16] L. B. Gratt. Risk analysis or risk assessment: A proposal for consistent definitions. In *Uncertainty in Risk Assessment, Risk Management, and Decision Making.* Plenum Press, NY, 1987.

[17] A. Hald. *Statistical Theory with Engineering Applications.* John Wiley and Sons, 1952.

[18] Morris Hamburg and Peg Young. *Statistical Analysis for Decision Making.* Dryden Press, 6th edition, 1994.

[19] K. C. Hendy. Situation awareness and workload: Birds of a feather? In *Situational Awareness: Limitations and Enhancements in the Aviation Environment.* 1995.

[20] Ronald A. Howard. *Dynamic Probabilistic Systems*, volume 1. John Wiley and Sons, 1971.

[21] Nehemiah Jordan. Allocation of functions between man and machines in automated systems. *Journal of Applied Psychology*, 47(3), 1963.

[22] S. Joshua and N. Garber. A causal analysis of large truck accident through fault trees. *Risk Analysis*, 12(2), 1992.

[23] Dean Karnopp and Ronald Rosenberg. *System Dynamics: A Unified Approach.* Wiley-Interscience, 1975.

[24] Roger E. Kirk. *Experimental Design.* Brooks/Cole Publishing Co., 2nd edition, 1982.

[25] P. Kuzminski, J. S. Eisele, N. Garber, R. Schwing, Y. Y. Haimes, D. Li, and M. Chowdhury. Improvement of highway safety I: Identification of causal factors through fault-tree modeling. *Risk Analysis*, 1995.

[26] Edward J. Lanzilotta. Analysis of driver safety performance using safety state model. In *Transportation Research Record*, number 1485, Safety and Human Performance: Human Performance and Safety in Highway, Traffic, and ITS Systems. Transportation Research Board, National Research Council, 1995.

[27] Edward J. Lanzilotta. Using the safety state model to measure driver performance. In *SP-1088.* SAE, 1995. also published as SAE Technical Paper 950968.

[28] Elmer E. Lewis. *Introduction to Reliability Engineering.* John Wiley and Sons, 1987.

[29] William W. Lowrance. *Of Acceptable Risk.* William Kaufmann, Inc., Los Altos, CA, 1976.

[30] David G. Luenberger. *Introduction to Dynamic Systems: Theory, Models, and Applications.* John Wiley and Sons, 1979.

[31] Gilbert Marshall. *Safety Engineering.* Brooks/Cole Engineering Division, 1982.

[32] Norman J. McCormick. *Reliability and Risk Analysis.* Academic Press, 1981.

[33] D. Meister. *Human Factors: Theory and Practice.* Wiley, 1971.

[34] Katsuhiko Ogata. *Modern Control Engineering.* Prentice-Hall, 2nd edition, 1990.

[35] James Reason. *Human Error.* Cambridge, 1990.

[36] N. Rescher. *Risk: A Philosophical Introduction to the Theory of Risk Evaluation and Management.* University Press of America, 1983.

[37] Sheldon M. Ross. *Introduction to Probability Models.* Academic Press, 3rd edition, 1985.

[38] W. D. Rowe. *An Anatomy of Risk.* John Wiley and Sons, 1977.

[39] Gavriel Salvendy, editor. *Handbook of Human Factors.* Wiley-Interscience, 1987.

[40] Mark S. Sanders and Ernest J. McCormick. *Human Factors in Engineering and Design.* McGraw-Hill, 6th edition, 1987.

[41] Nadine B. Sarter and David D. Woods. How in the world did we ever get into that mode? mode error and awareness in supervisory control. *Human Factors,* 37(1), 1995.

[42] J. W. Senders and N. P. Moray. *Human Error: Cause, Prediction, and Reduction.* Erlbaum, 1991.

[43] Thomas B. Sheridan. *Telerobotics, Automation, and Supervisory Control.* MIT Press, 1992.

[44] Thomas B. Sheridan, Edward J. Lanzilotta, and Shumei Yin Askey. Safety of High Speed Guided Ground Transportation Systems—Human Factors Phase I: Function analyses and theoretical considerations. Technical Report DOT-VNTSC-FRA-94-4, U.S. Department of Transportation, Federal Rail Administration, 1994.

[45] David Shinar. *Psychology on the Road: The Human Factor in Traffic Safety.* Wiley, 1978.

[46] Kip Smith and P. A. Hancock. Situation awareness is adaptive, externally directed consciousness. *Human Factors*, 37(1), 1995.

[47] Murray R. Spiegel. *Theory and Problems of Statistics.* Schaum's Outline Series. McGraw-Hill, Inc., 2nd edition, 1988.

[48] A. D. Swain and H. E. Guttman. Handbook of human reliability analysis with emphasis on nuclear power plant applications. Technical Report NUREG CR-1278, U. S. Nuclear Regulatory Commission, Washington, D. C., 1983.

[49] Webster's New Universal Unabridged Dictionary. Barnes and Noble, 1994.

[50] F. Wharton. Risk management: Basic concepts and general principles. In J. Ansell and F. Wharton, editors, *Risk Analysis, Assessment, and Management.* John Wiley and Sons, 1992.