Title
Safety Evaluation of Vehicle Following Operations by Fault Tree and Sensitivity Analysis

Author
Chan, Ching-Yao

Publication Date
2000-09-01

CALIFORNIA PATH PROGRAM
INSTITUTE OF TRANSPORTATION STUDIES
UNIVERSITY OF CALIFORNIA, BERKELEY

Safety Evaluation of Vehicle Following Operations by Fault Tree and Sensitivity Analysis

Ching-Yao Chan

California PATH Research Report
UCB-ITS-PRR-2000-18

This work was performed as part of the California PATH Program of the University of California, in cooperation with the State of California Business, Transportation, and Housing Agency, Department of Transportation; and the United States Department of Transportation, Federal Highway Administration.

Report for MOU 325

September 2000

ISSN 1055-1425

PARTNERS FOR ADVANCED TRANSIT AND HIGHWAYS (PATH)

Safety Evaluation of Vehicle Following Operations by Fault Tree and Sensitivity Analysis

Final Report of MOU 325

submitted by

Ching-Yao Chan

EXECUTIVE SUMMARY

This document is the final report for the project of MOU325 for the fiscal years of 1997-2000, which is a continuation of MOU253 from the years of 1995-1997. The major accomplishments from this project can be categorized into the following two areas:

(1) Development of fault tree models for safety evaluation of advanced vehicle control and safety systems.
(2) Utilization of an off-the-shelf fault tree tool to conduct fault tree analysis, such as cutest generation and sensitivity studies.

This research project utilizes commercially available software tools, CAFTA for Windows, to evaluate the safety issues in automated vehicle operations. This software tool was developed by Science Applications International Corporation (SAIC) to evaluate complex and safety-critical systems. The model allows the evaluation of sensitivity study by constructing a fault tree model with corresponding database for failure events.

Operational safety is one critical objective in the development of vehicle technologies. Automation of vehicle control actions in Advanced Vehicle Control Systems (AVCS) or Automated Highway Systems (AHS) are proposed to meet or improve traffic safety over the status quo. In order to do so, it is necessary for the AHS concept selection and design process to include an evaluation of the possible failure modes or events, an estimation of failure probabilities, and an analysis of risk management strategies.

One technique for performing a safety evaluation of a complex and critical problem is fault-tree analysis. In this study, we conducted a fault-tree analysis of vehicle following performed under automatic control. With the readily available tool, CAFTA, we evaluated fault-tree analysis of two different scenarios:

(1) Vehicle-following collision fault tree analysis
(2) Lane-keeping fault tree analysis

In the project, we have attempted to establish a procedure and a method to evaluate the safety aspects of AVCSS by building fault tree models and using tools. We have adopted

a commercial off-the-shelf software tool to allow such endeavors. We demonstrated the process of conducting fault tree analysis by using longitudinal and lateral control systems as examples in this project.

The selection of the top fault event and the degree of sophistication of the modeled systems dictate the results of fault tree analysis. The variations in components and implementation strategies can lead to considerable complexity of the fault tree models and the countermeasures in dealing with the safety hazards of interests. Even though the examples shown in this study only reflect the chosen architecture and limited component details of AVCSS, it does provide us a solid foundation to pursue further studies of this nature.

It is important to possess a reasonably accurate reliability database for the evaluation of fault tree models, which may still be lacking for many subsystems of AVCSS. It is also essential to investigate the performance characteristics of a system in a real-world environment to assess the criticality and consequence of failure events. These challenges and issues will be the topics of future studies in providing a methodology for estimating and evaluating the safety nature of such systems.

KEY WORDS:

Safety Evaluation
Fault Tree Model
Fault Tree Analysis
Advanced Vehicle Control Systems

1.0 INTRODUCTION

This document is the final report for the project of MOU325 conducted over the fiscal years of 1997-2000, which is a continuation from MOU253 in the years of 1995-1997.

This project is focused on the development of modeling tools to evaluate safety issues in vehicle following operations. The project consist of the following efforts:
(1) Use of the SAIC CAFTA fault tree analysis tool for safety analysis.
(2) Evaluate the effects of failure in radar, communication, and speed sensor on the likelihood of intra-platoon collisions.
(3) Sensitivity analysis of selected variables and failure events and fault tree models.

This research project utilizes commercially available software tools to evaluate the safety issues in automated vehicle operations. The Fault Tree Tool was CAFTA for Windows, developed by Science Applications International Corporation (SAIC). This software tool was developed to evaluate complex and safety-critical systems. The model allows the evaluation of sensitivity study by constructing a fault tree model with corresponding database for failure events.

1.1 Past Work on Safety Analysis at PATH

Over the years, there have been many projects conducted by PATH researchers on the analysis the safety benefits and hazards of AVCS and AHS. The following is a brief summary of related work in this area.

Hitchcock conducted fault-tree analysis of an AHS [1,2]. He considered a set of hazards that might result in high delta-velocity collisions. The fault trees are built for each hazard to scrutinize the conditions that could cause such hazards. He asserted that there is one system that provides for much lower casualties than its rivals and he proposes the layout, design and operation of such a system [3,4]. He also described a configuration and operating principals for an AHS that is believed to have safety advantages [5].

Garg and Hedrick examined the issues of fault-tolerant control of vehicle following systems [6]. They discussed potential fault modes among sensors and actuators used in cars in vehicle-following experiments, as well as issues of fault detection of sensors and actuators. Patwardhan studied the fault detection and fault-tolerant control for lateral guidance of vehicles traveling on an automated highway [7]. He addressed the failure modes of tire bursts and sensor faults and the importance of slip angle control to enhance vehicle safety.

Tongue and his students embarked on a series of studies to analyze the behaviors of platoon systems [8-11]. They investigated the effects of selected parameter variations on the response of a platoon, the response of a platoon under different control algorithms, and platoon behavior during non-nominal operations, in particular emergency braking conditions.

Chan used a two-dimensional simulation model to analyze vehicle trajectories in vehicle-following collisions [12,13]. The effects of operational variables, such as lateral offset, initial spacing, speed and vehicle sizes, on the outcome of the collisions were evaluated. It was found that large offsets and large delta-velocity in collisions could cause greater path deviations and early lane departure. Chan and Tan further illustrated the feasibility of applying steering control to stabilize vehicle trajectories in vehicle-following collisions [14-16].

Michael developed a process model and architecture for incorporating the products of fault-tree analysis and other safety evaluation techniques into safety cases for AHS [17]. A specific example is provided of the linkage between the fault-tree model of the lateral control system component of an AHS-equipped vehicle and the safety argument for an AHS.

Michael, Segal, and Patwardhan developed a technique known as Monte Carlo black-box testing [18]. The technique is used to generate a state-space performance characterization of software. They demonstrated its use for evaluating the software for PATH's experimental lateral control system. The results of the testing are described in the context of lateral control system failure modes.

Segal, Chan, and Michael [19] demonstrated the use of the tools by building a fault tree model for automated vehicle control systems (AVCS). They conducted an analysis of the critical variables in AVCS operations, the product of which forms the basis from which to derive system safety requirements for vehicle following operations.

1.2 Research Focus and Approach

Operational safety is one critical objective in the development of vehicle technologies. Automation of vehicle control actions in Advanced Vehicle Control Systems (AVCS) or Automated Highway Systems (AHS) are proposed to meet or improve traffic safety over the status quo. In order to do so, it is necessary for the AHS concept selection and design process to include an evaluation of the possible failure modes or events, an estimation of failure probabilities, and an analysis of risk management strategies.

One technique for performing a safety evaluation of a complex and critical problem is fault-tree analysis. A fault-tree model is a logical representation of a system from a hierarchical structure of failure or malfunction events. In this study, we conducted a fault-tree analysis of vehicle following performed under automatic control. With the assistance of commercially available tool, CAFTA, developed by Science Applications International Corporation (SAIC), we modeled potential modes of failure or component malfunction, which are recorded in a database. In this report, we evaluated fault-tree analysis of two different scenarios:

(3) Vehicle-following collision fault tree analysis
(4) Lane-keeping fault tree analysis

Although the structure of the fault trees and the representation of events and components are general, they do contain elements and terminology described by the PATH architecture of AHS and AVCSS. [20,21,22] The fault-tree models presented in this report centers on the physical and the regulation layers. The model is extensible; it can be expanded to incorporate the upper three layers: coordination, link, and system.

2.0 WHAT IS A FAULT TREE MODEL?

A fault analysis can be described as an analytical technique, by which an undesired state of a system from a safety point of view is specified, and the system is then analyzed in the context of its environment and operation to find all credible ways in which the undesired event can occur. [23,24]

2.1 The Fault Tree Model

The fault tree itself is a graphic model of the various parallel and sequential combinations of faults that can be events associated with component hardware failures, human errors, or any other pertinent events that can lead to the undesired event. A fault tree thus depicts the logical relationships of basic events that lead to the undesired event, which is the top event of the fault tree.

It is important to understand that a fault tree is not a model of all possible system failures or all possible causes for system failure. A fault tree is tailored to its top event, which corresponds to some particular system failure mode, and the fault tree thus includes only those faults that contribute to this top event. Moreover, these faults are not exhaustive; they cover only the most credible faults as assessed by the analyst.

It is also important to note that a fault tree is not in itself a quantitative model. It is a qualitative model that can be evaluated quantitatively. The qualitative nature is true of almost all varieties of system models. The fact that a fault tree is particularly convenient model to quantify does not change the qualitative nature of the model itself.

A fault tree is a complex of entities known as "gates" that serve to permit or inhibit the passage of fault logic up the tree. The gates show the relationships of events needed for the occurrence of a "higher" event. The "higher" event is the "output" of the gate; the "lower" events are the "input" to the gate. The gate symbol denotes the type of relationship of the input events required for the output event. Gates are analogous to switches in an electrical circuit or valves in a piping layout.

2.2 Faults and Failures

A distinction should be made between "failure" and "fault." Consider a relay. If the relay closes properly when a voltage is applied across its terminals, we can call this relay a success. If, however, the relay fails to close under the circumstances, we can call this relay a failure. Another possibility is that the relay closes at the wrong time due to

improper functioning of some upstream components. This is not a relay failure, but untimely relay operation may cause the entire circuit to be in an unsatisfactory state. We will call all occurrences like this a "fault."

Generally speaking, all failures are faults but not all faults are failures. Failures are basic abnormal occurrences, whereas faults are "higher-order" events. The proper definition of a fault requires a specification of not only what the undesirable component state is but also when it occurs. These "what" and "when" specifications should be part of the event descriptions that are entered into the fault tree.

A fault may be repairable or not, depending on the nature of the system. Under conditions of no repair, a fault that occurs will continue to exist. In a repairable system, a distinction must be made between the occurrence of a fault and its existence. This distinction is of importance in fault quantification. In constructing a fault tree, we are only concerned about the phenomenon of fault occurrence.

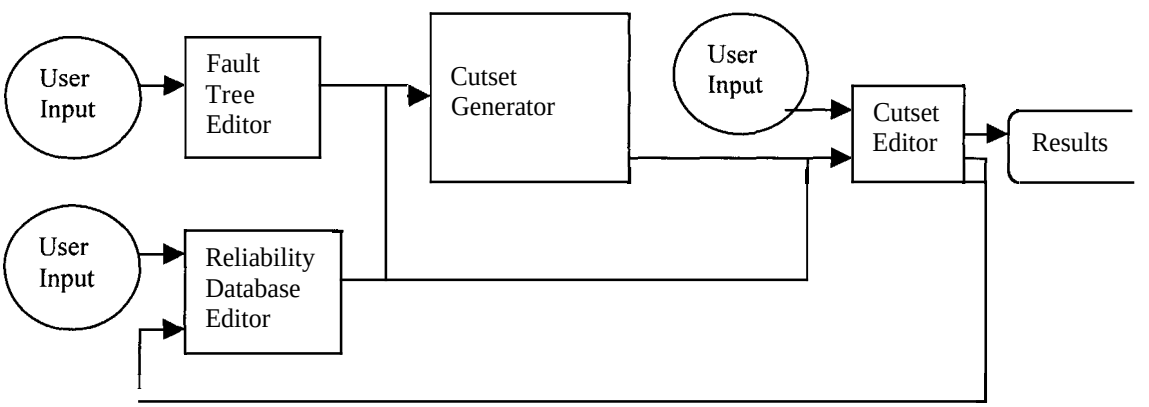


Figure 1. Diagram of Major Blocks in CAFTA

3.0 WHAT IS CAFTA?

CAFTA for Windows [25] is a microcomputer-based fault tree analysis workstation, providing the tools necessary to model and analyze complex systems. It is a product of Science Applications International Corporation. The purchase of CAFTA by PATH was sponsored by the budget of MOU253, a predecessor of the current MOU325.

CAFTA is designed to meet the needs of reliability analysis and to perform fault tree analysis on a system or a group of systems. It includes:

(1) A Fault Tree Editor for building, updating and printing fault tree models,
(2) Reliability database for storing the basic event, failure rate and gate information used in the models;
(3) **A** Fault Tree evaluating processor used to obtain model cutsets, and
(4) A Cutset Editor, which is a valuable tool for reviewing and analyzing cutest results.

Figure 1 shows the major CAFTA for Windows tools that correspond to the steps of fault tree analysis.

4.0 APPLICATION OF CAFTA TO AVCSS

Advanced Vehicle Control and Safety Systems (AVCSS) consist of sophisticated hardware and software, which are assembled and coordinated to execute the designed features and functions. Due to their embedded complexity and their safety-critical nature, they are good candidates for applying fault tree analysis. Their failures may come from a condition of a component, sub-system, or system when the intended design or specific operation is not properly satisfied.

An AVCSS failure can be benign, such as small errors in tracking the preceding vehicle or slight discomfort for the riders. On the other hand, a failure can also cause a serious consequence, such as a fatal crash. In order to build safe AVCSS, it is essential to investigate the failure modes that can be identified. The fault tree model is a step in this direction to evaluate the design of such systems.

Among the many types of potential failures of a longitudinal control system, we selected a vehicle-to-vehicle collision as the top event of the fault tree as the case study for longitudinal control systems. Although the chosen top event cannot include all possible combinations of faults or failure modes in AVCSS, it permits us to establish a fault tree model to reflect the structure and components of the targeted system. It also serves the purpose of exercising various features of fault tree tools to construct models, editing reliability database, and generating cutsets, which are the primary objectives of this project.

A separate fault tree was constructed for a lateral control system with the top event given as the failure of lane-keeping function. Since the PATH experimental vehicles, which are the target s of the models, are equipped and operated with quasi-independent systems for longitudinal and lateral control, it was reasonable to proceed with such division when considering faults or failures in each system. Nevertheless, this does not prevent a possible consolidation if the two models need to merge in the future when either the two systems are inter-connected closely faults or failures from either systems can not distinctly differentiated in their effects on the selected top event.

The two fault tree models and their respective database are listed in the Appendices C and D. The fault tree models are printed using the fault tree editor and the reliability database editor within CAFTA.

4.1 Fault Tree Models and Analysis

For the longitudinal control system, we try to imitate the vehicle-following operation in a platoon scenario. The platoon concept demonstrated by PATH [21] over the years calls

for a tightly spaced formation. The close-spacing formation can increase highway throughput and reduce the speed differential of collisions in failures. The spacing remains relatively constant even when the speed of a platoon changes. The intra-platoon spacing is only adjusted when there are needs for splitting, joining, or lane changing. To maintain the stability of spacing control in a string of vehicles, coordination between vehicles is performed using inter-vehicle communication in either a token-ring or broadcast fashion. Vehicle status and identification are exchanged and the data is used as input to the regulation-layer control system. These elements of the automated vehicle, such as sensors, actuators, and communication systems are reflected in the fault tree model constructed for this project. The created fault tree model is given in Appendix C. Following the fault tree is a list of basic events, gate descriptions and reliability database.

For the lateral control system, we also construct a fault tree to indicate the main building blocks in an automated system that guides the vehicle in a designated trajectory. A magnetic sensing system [22] provides the information need by a vehicle to follow a designated path of a vehicle. **A** steering control system uses the vehicle status and the roadway information to perform lane tracking or lane change maneuvers. The created fault tree model is listed in Appendix D. Following the fault tree is a list of basic events, gate descriptions and reliability database.

4.2 Vehicle Dynamics Simulation

To assess the effects of component failures on vehicle-following operations, a simulation model is being used to investigate the probability and magnitude of collisions. Failure scenarios with single and multiple failures including radar, speed sensor, and communication links are included in the current study. In addition, various control algorithms and operational conditions will be tested in simulations to assess the safety consequences.

In order to understand the cause and effect of events with component failures, we adopt a simulation program based on a certain control law:

$$U = G_{al}*A_l + G_{ap}*A_p + G_{vl}*\_V_l + G_{pl}*\_V_p + G_d*R$$

Where U is the desired acceleration rate, G's are the gains in the control law, and A's are the acceleration data, *V's* are the speed differentials, R is the range (distance to the preceding car), and subscripts 1 and p refer to the leading car and the preceding car.

In a series of simulations, we examine the probability of a collision between the second and the third vehicle in a four-car platoon with faults occurring on the third vehicle. We also tried out different duration of failures in various components. For example, when a radar failure occurs, it is indicated with a wrong value for a period of 1 to 5 seconds or indefinitely. We then continue the simulation to test whether the control laws can avoid or attenuate a potential collision.

The purpose of this simulation exercise was to explore the "what and when" fault effects in the selected longitudinal control system by allowing a recovery when a failure is repairable. It was found that with the selected controller, the platoon was especially vulnerable to the speed sensor failure, which has no supplementary backup or fault tolerance to overcome an erroneous reading or malfunction. It should be noted that this observation is based on a limited number of simulation scenarios and the specific controller. However, it does permit the investigation of particular events when a control law or a system is being validated.

4.3 Fault Tree Analysis

CAFTA for Windows includes a Cutset Editor to conduct an evaluation of a fault tree once it is constructed and the reliability database are in place. The cutest analysis is executed on the two fault trees described above with a probability cutoff of 1.0×10^{-5}. The results of the cutest analysis are shown in Appendices C and D respectively.

Indicated on the top of the cutset pages are the tope event probability. Listed below that is the events that will lead to the top event and their associated probabilities. As can be seen in the results, these cutsets are very straightforward since they all involve single-event occurrence. The basic events that have the highest probability of occurrence naturally show up in the cutest analysis. This is an indication that the propagation of these events from their positions in the tree to the top do not pass an "AND" gate. Nevertheless, the cutest analysis allows the user to see immediately what are the critical events that should be guarded against in the implementation of a system.

For sensitivity studies, the user can vary the numbers in the reliability database and then execute the cutest analysis again to see how the top event probability changes along with that. Also, if the probabilities of events are modified, the order of the critical events in th ecutset results will also be adjusted.

Alternatively, when once sees a sub-system or component in the list of critical events, a modification of the system design or structure can alter or eliminate the criticality of that particular fault. For example, in the cutset list of the longitudinal control system, the radar and communication sub-systems are on the top of the list because of their operating characteristics and high failure probabilities. If we can implement some measures of fault detection and fault tolerance algorithms, the chance of their failure propagating to the top event can be significantly reduced. For example, in Appendix E, we show the portion of a modified fault tree with such measures inserted for the radar and communication portion. The associated gate information and basic events are also attached. When the system is re-evaluated with the cutset tool, the number of critical events was reduced from 14 to 7 with the same cutoff probability of 1.0×10^{-5}. The actual difference in the top event probability was not that significant, but the example demonstrated an approach of searching and correcting the vulnerable points in a safety-critical system.

4.4 Discussions

The values of probabilities given in the reliability database of the fault tree models are only adopted for demonstration purposes. **A** reliable set of numbers can only be obtained by empirical studies or from validated data of existing systems. [A.2] **A** challenging issue of this project was that the fault tree model was constructed for an experimental system, which is continuously evolving and consists of made-for-order components or sub-systems.

In the construction of a fault tree model, it is not necessarily difficult to follow the logic process in re-establishing the architecture of the targeted system. However, it can be quite complicated in assuming or assigning a probability to a certain event. For example, a component failure may lead to a stalled vehicle, which may result in either minor inconvenience or major catastrophes, depending the surrounding traffic or environment at the instant of the failure. Therefore, a thorough failure mode and effect criticality analysis is essential to fully explore the possibility or probability of the failure leading to the top event being considered in the fault tree. To assess the probability of a certain event, a series of simulation or experiments under a diverse set of operating conditions will be necessary to test and verify the occurrence of the targeted or related consequences, as exemplified in the work explained in Section 4.2.

5.0 CONCLUSION

In the project, we have attempted to establish a procedure and a method to evaluate the safety aspects of **AVCSS** by building fault tree models and using tools. We have adopted a commercial off-the-shelf software tool to allow such endeavors. We demonstrated the process of conducting fault tree analysis by using longitudinal and lateral control systems as examples in this project.

The selection of the top fault event and the degree of sophistication of the modeled systems dictate the results of fault tree analysis. The variations in components and implementation strategies can lead to considerable complexity of the models itself and the countermeasures in dealing with the safety hazards of interests. Even though the examples shown in this study only reflect the chosen architecture and limited component details of **AVCSS,** it does provide us a solid foundation to pursue further studies of this nature.

It is important to possess a reasonably accurate reliability database for the evaluation of fault tree models, which may still be lacking for many subsystems of **AVCSS.** It is also essential to investigate the performance characteristics of a system in a real-world environment to assess the criticality and consequence of failure events. These challenges and issues will be the topics of future studies in providing a methodology for estimating and evaluating the safety nature of such systems.

REFERENCES

1. Hitchcock, A., "Fault Tree Analysis of a First Example Automated Freeway," PATH Research Report UCB-ITS-PRR-91-14.
2. Hitchcock, A., "Fault Tree Analysis of an Automated Freeway with Vehicle-Borne Intelligence," PATH Research Report UCB-ITS-PRR-92-15.
3. Hitchcock, A., "Layout, Design and Operation of a Safe Automated Highway System," PATH Research Report UCB-ITS-PRR-95-11.
4. Hitchcock, A., "Entry to and Exit from a Safety-Consciously Designed AHS Configuration," PATH Tech Note 95-04.
5. Hitchcock, A., "Configuration and Maneuvers in a Safety-Consciously Designed AHS Configuration," PATH Research Report UCB-ITS-PRR-95-02.
6. Hedrick, J.K., Garg, V., "Issues in Fault Tolerant Control of Vehicle Follower Systems," PATH Research Report UCB-ITS-PRR-94-11.
7. Patwardhan, S. "Fault Detection and Tolerant Control for Lateral Guidance of Vehicles in Automated Highway," PATH Research Report UCB-ITS-PRR-94-17.
8. Tongue, B., Yang, Y-T, M. White, "Platoon Collision Dynamics and Emergency Maneuvering I: Reduced Order Modeling of a Platoon for Dynamic Analysis," PATH Research Report UCB-ITS-PRR-91-15.
9. Tongue B., Yang, Y-T, "Platoon Collision Dynamics and Emergency Maneuvering II: Platoon Simulations for Small Disturbances," PATH Research Report UCB-ITS-PRR-94-04.
10. Tongue, B., Yang, Y-T, "Platoon Collision Dynamics and Emergency Maneuvering III: Platoon Collision Models and Simulations," PATH Research Report UCB-ITS-PRR-94-02.
11. Tongue, B., Yang, Y-T, "Platoon Collision Dynamics and Emergency Maneuvering IV: Intraplatoon Collision Behavior and A New Control Approach for Platoon Operation During Vehicle Exit/Entry," PATH Research Report UCB-ITS-PRR-94-25.
12. Chan, C.-Y., "Studies of Collisions in Vehicle Following Operations by Two-Dimensional Impact Simulations," ITS American, Sixth Annual Meeting, Houston, Texas, April 1996.
13. Chan, C.-Y., "Collision Analysis of Vehicle Following Operations in Automated Highway Systems," Third World Congress on Intelligent Transport Systems, Orlando, Florida, October 1996.
14. Chan, C.-Y., Tan, H-S, "Lane Tracking Control in Vehicle-Following Collisions,'' in *Proceedings of 1999 America Control Conference,* San Diego, California, USA, June 1999.
15. Chan, C.-Y., Tan, H-S, "Application of a Robust Controller in Emergency Situations," in *Proceedings of 2999 IEEE/IEEJ/JSAI Conference on Intelligent Transportation Systems,* Tokyo, Japan, October 1999
16. Tan, H-S, Chan, C.-Y., "Design of Steering Controller and Analysis of Vehicle Lateral Dynamics under Impulsive Disturbances," *Proceedings of2999 America Control Conference,* Chicago, Illinois, USA, June 2000.
17. Michael, J. B. "Information Requirements for Managing System Safety of Software-Controlled Automated Highways." In *Proceedings of the Transportation*

Management Conference, Maritime College, Throggs Neck, New York: State University of New York, 1994, pp. 476-490.

18. Michael, J. B., Segal, A.C., Patwardhan, **S.,** "Validation of Software Testing Results for Real-Time Vehicle Control Software." In *Systems and Issues in ITS.* Warrendale, Penn.: Society of Automotive Engineers, 1995, pp. 69-72.

19. Segal, A.C., Chan, C.Y., Michael, J. B., "Fault Tree Analysis of Advanced Vehicle Control Systems," ITS American, Seventh Annual Meeting, Washington, D.C., 1997.

20. Varaiya, P., Shladover, **S.,** "Sketch of an IVHS System Architecture," PATH Research Report UCB-ITS-PRR-91-03.

21. Hedrick, J.K., et al, "Longitudinal Control Development for IVHS Fully Automated and Semi-Automated Systems: Phase II," PATH Research Report UCB-ITS-PRR-96-01.

22. Tan, H-S, Guldner, J., Patwardhan, **S.,** Chen, C., and Bougler, B., "Development of an Automated Steering Vehicle Based on Roadway Magnets - A Case Study of Mechatronic System Design," *IEEE/ASME Transactions on Mechatronics,* vol. 4, no. 3, Sept., 1999, pp. 258-272.

23. W.E. Haasl, N.F. Goldberg, Fault Tree Handbook, U.S> Nuclear Regulatory Commission, Report NUREG-0492, January, 1981.

24. U.S. Nuclear Regulatory Commision, "Reactor Safety Study – An Assessment of Accident Risks in U.S. Commercial Nuclear Power Plant,'' WASH-1400, NUREG-75/014, October 1975.

25. *CAFTA for Windows, Fault Tree Analysis System, Version 3.1, User's Manual,* Electric Power Research Institute and Science Applications International Corporation, July 1995.

Appendix A.
A BIREF REVIEW OF FAULT TREE MODEL

A fault analysis can be described as an analytical technique, by which an undesired state of a system from a safety point of view is specified, and the system is then analyzed in the context of its environment and operation to find all credible ways in which the undesired event can occur. [B.1, B.2]

A.l The Fault Tree Model

The fault tree itself is a graphic model of the various parallel and sequential combinations of faults that can be events associated with component hardware failures, human errors, or any other pertinent events that can lead to the undesired event. **A** fault tree thus depicts the logical relationships of basic events that lead to the undesired event, which is the top event of the fault tree.

It is important to understand that a fault tree is not a model of all possible system failures or all possible causes for system failure. A fault tree is tailored to its top event, which corresponds to some particular system failure mode, and the fault tree thus includes only those faults that contribute to this top event. Moreover, these faults are not exhaustive; they cover only the most credible faults as assessed by the analyst.

It is also important to note that a fault tree is not in itself a quantitative model. It is a qualitative model that can be evaluated quantitatively. The qualitative nature is true of almost all varieties of system models. The fact that a fault tree is particularly convenient model to quantify does not change the qualitative nature of the model itself.

A fault tree is a complex of entities known as "gates" that serve to permit or inhibit the passage of fault logic up the tree. The gates show the relationships of events needed for the occurrence of a "higher" event. The "higher" event is the "output" of the gate; the "lower" events are the "input" to the gate. The gate symbol denotes the type of relationship of the input events required for the output event. Gates are analogous to switches in an electrical circuit or valves in a piping layout.

A.2 Fault Tree Construction

A distinction should be made between "failure" and "fault." Consider a relay. If the relay closes properly when a voltage is applied across its terminals, we can call this relay a success. If, however, the relay fails to close under the circumstances, we can call this relay a failure. Another possibility is that the relay closes at the wrong time due to improper functioning of some upstream components. This is not a relay failure, but untimely relay operation may cause the entire circuit to be in an unsatisfactory state. We will call all occurrences like this a "fault."

Generally speaking, all failures are faults but not all faults are failures. Failures are basic abnormal occurrences, whereas faults are "higher-order" events. The proper definition of

a fault requires a specification of not only what the undesirable component state is but also when it occurs. These "what" and "when" specifications should be part of the event descriptions that are entered into the fault tree.

A fault may be repairable or not, depending on the nature of the system. Under conditions of no repair, a fault that occurs will continue ot exist. In a repairable system, a distinction must be made between the occurrence of a fault and its existence. This distinction is of importance in fault quantification. In constructing a fault tree, we are only concerned about the phenomenon of fault occurrence.

A.2.1 Passive versus Active Components

A passive component contributes to the functioning of a system in a static manner, such as a wire carrying current, a steam line transmitting heat, or a structural member transmitting loads. Other examples of passive components are pipes, bearings, journals, welds, and so forth.

An active component contributes to the functioning of a system in a dynamic manner by modifying the system behavior in certain ways. **A** valve, for example, modifies the fluid flow, and a switch has a similar effect on the current in a circuit. Other examples include relays, resistors, pumps, and so on.

A passive component can be considered as the transmitter of a "signal." It can also be thought of as the "mechanism," by which the output of an active component becomes the input to a second active component. The failure of a passive component will result in the non-transmission or partial transmission of its "signal."

In contrast, **an** active component originates or modified a signal. The active component acts as a "transfer function." If an active component fails, there may be no output signal or an incorrect output signal.

From a numerical reliability standpoint, the importance difference between failures of active and passive components is the failure rate values. **As** indicated in [A.2], active component in general have failure rates above $1x10^{-4}$ per demand or above $3x10^{-7}$ per hour and passive components have failure rates below those values. In fact, the difference in reliability between the tow types of components is commonly two to three orders of magnitude.

A.2.2 Component Fault Categories: Primary. Secondary, and Command

It is useful to classify faults into three categories: primary, secondary, and commands.

A primary fault is any fault of a component that occurs in an environment for which the component is qualified. For example, a pressure tank ruptures at a pressure lower than what it has been designed to withstand.

A secondary fault is any fault of a component that occurs in an environment for which it has not been qualified. For example, a pressure tank ruptures at a pressure higher than what it has been designed to withstand.

Because primary and secondary faults are component failures, they are called primary and secondary failures. In contrast, a command fault involves the proper operation of a component but at the wrong time or in the wrong place. For example, a relay in a circuit closes too soon because of a premature or erroneous signal coming from some upstream component

A.2.3 Failure Mechanism, Failure Mode, and Failure Effect

The definition of system, sub-systems, and components are relative, which depends on the context of the analysis. In a particular analysis, definitions of system, sub-systems, and components are generally made for convenience to give hierarchy and boundaries to the problem.

In constructing a fault tree, the basic concepts of failure effects, failure modes, and failure mechanisms are important in determining the proper inter-relationships among the events. When we speak of failure effects, we are concerned about why the particular failure is of importance. When we detail the failure modes, we are specifling exactly what aspects of component failures are of concern. When we list failure mechanisms, we are considering how a particular failure mode can occur and also, perhaps, what are the corresponding likelihood of occurrence. Thus, failure mechanisms produce failure modes, which in turn have certain effects on system operation.

Consider a system that controls the flow of fuel. The sub-system of interest consists of a valve and a valve actuator. Table 1 gives a classification of various events as viewed from the system, sub-system, or component level. For example, "valve unable to open" is a mechanism of sub-system failure, a mode of valve failure, and an effect of actuator failure.

Events	System	Subsystem	Valve	Actuator
No flow from subsystem when required	Mechanism	Mode	Effect	
Valve unable to open		Mechanism	Mode	Effect
Binding of actuator stem			Mechanism	Mode
Corrosion of actuator stem				Mechanism

Table 1. Fuel Flow System Failure Analysis

For the construction of a fault tree, an analyst selects one of the failure events and investigates the immediate causes for its occurrence. These immediate causes will be the immediate failure mechanisms for the particular system failure chosen, and will

constitute failures of certain subsystems. These latter failures will be failure modes for the subsystems and will make up the second level of the fault tree. We proceed, step by step, in this "immediate cause" manner until we reach the component failures. These components are the basic causes defined by the limit of resolution of the fault tree.

A.3 Basic Rules for Fault Tree Constructions

Ground Rule I:
Write the statements that are entered in the event boxes as faults; state precisely what the fault is and when it occurs.

Ground Rule II:
If the answer to the question, "Can this fault consist of a component failure?' is "yes," classify the event as a "state-of-component fault." If the answer is "No," classify the event as a "state-of-system fault."

No Miracles Rule:
If the normal functioning of a component propagates a fault sequence, then it is assumed that the component functions normally.

Complete-the-Gate Rule:
All inputs to a particular gate should be completely defined before further analysis of any one of them is undertaken.

No Gate-to-Gate Rule:
Gate inputs should be properly defined fault events, and gates should not be directly connected to other gates.

References:

A. 1. W.E. Haasl, N.F. Goldberg, Fault Tree Handbook, U.S> Nuclear Regulatory Commission, Report NUREG-0492, January, 1981.
A.2. U.S. Nuclear Regulatory Commision, "Reactor Safety Study – An Assessment of Accident Risks in U.S. Commercial Nuclear Power Plant,'' WASH-1400, NUREG-75/014, October 1975.

Appendix B.

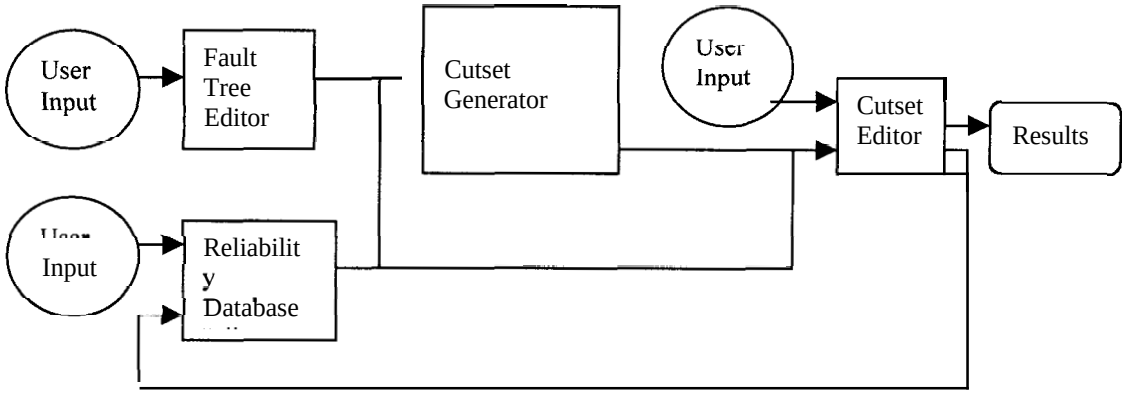
A BREIF REVIEW OF CAFTA [B.1]

CAFTA for Windows is a microcomputer-based fault tree analysis workstation, providing the tools necessary to model and analyze complex systems. It is a product of Science Applications International Corporation. The purchase of CAFTA by PATH was sponsored by the budget of MOU253, a predecessor of the current MOU325.

CAFTA is designed to meet the needs of reliability analysis and to perform fault tree analysis on a system or a group of systems. It includes:

(5) A Fault Tree Editor for building, updating and printing fault tree models,
(6) Reliability database for storing the basic event, failure rate and gate information used in the models;
(7) A Fault Tree evaluating processor used to obtain model cutsets, and
(8) A Cutset Editor which is a valuable tool for reviewing and analyzing cutest results.

The following Diagram shows the major CAFTA for Windows tools that correspond to the steps of fault tree analysis.



B.l Fault Tree Editor

The development of detailed fault tree models involves the construction of fault tree logic based on the analyst's understanding of the system being modeled. The CAFTA editor allows the analyst to check the models as it is being built. The editor also provides a link to reliability database, allowing the user to input the supporting reliability and descriptive data while building the logic model.

B.2 Reliability Data

The reliability data is required for every basic events being modeled in the fault tree. The basic events represent the probability of failure or unavailability of a component in the system, an operation or maintenance error, or an initiating event. Failure probabilities may vary from system to system because of different operating commitments, exposure or mission times. In general, the failure rate is a function of the type of event being modeled, whereas the exposure time is function of how the component in the system performs. The exposure time may be the mean repair time, test interval, or mission time.

CAFTA for Windows provides two centralized databases to manage reliability data to ensure that the most up-to-date values are used. The Basic Event Database contains the basic event names and the system-dependent exposure data contributions to the failure probabilities, e.g. mission time. The Failure Rate Database contains the failure rate or demand failure probability data for each type of event.

B.3 Fault Tree Evaluation

The fault tree can be reduced to the form of cutsets using the CAFTA for Windows cutest generator. A cutest is a minimal set of basic events that lead to the failure of the fault tree top event. These cutsets provide insights to the functionality of the system, as well as a quantification of the top event probability.

Any fault tree of reasonable size will have thousands of cutsets. In order to provide a reasonable list of cutsets, CAFTA for Windows allows one to truncate the cutsets, removing cutssets that have a very small probability of occurring, or cutsets that contain failures of more than some number of events.

B.4 Cutset Editor

The review of cutest results is very important. The review process depends on the type of projects and the complexity of models being evaluated. At the simplest level, cutsets need to be reviewed to check the system model.

At a higher level, cutsets need to be reviewed to take into account operation or maintenance effects not included in the fault tree model. Often adding these types of events to the fault tree models adds a large degree of complexity to the evaluation process that can be more easily handled during the review of cutsets. An example might be a cutest containing two maintenance events, which canot occur simultaneously according to maintenance procedures.

At an even higher level, the cutests need to be reviewed to determine if recovery events are needed and possible. For example, if a given set of component fails, the operator may take an action that will bypass the problem. These types of actions depend on exactly what fails, and therefore are cutset dependent.

In addition to supporting the review of cutest results, the Cutset Editor provides a tool for qualitative analysis. The editor can be used for ranking events by their importance to the top event and for sensitivity studies.

B.4.1 Important Measures

Important measures let you rank events by their importance or contribution to the top event probability and are used to provide quantifiable ranking of this contribution. These measures let one identify which components are most likely to fail.

B.4.2 Sensitivity Studies

Sensitivity studies can be performed by looking at both model structures and event failure probabilities. Key issues and assumptions are often analyzed by modifying the models or results and noting the effects of these changes. The Cutset Editor lets the user change probabilities, add or remove events to cutsets, and add or delete cutsets. All these approaches are used in sensitivity studies.

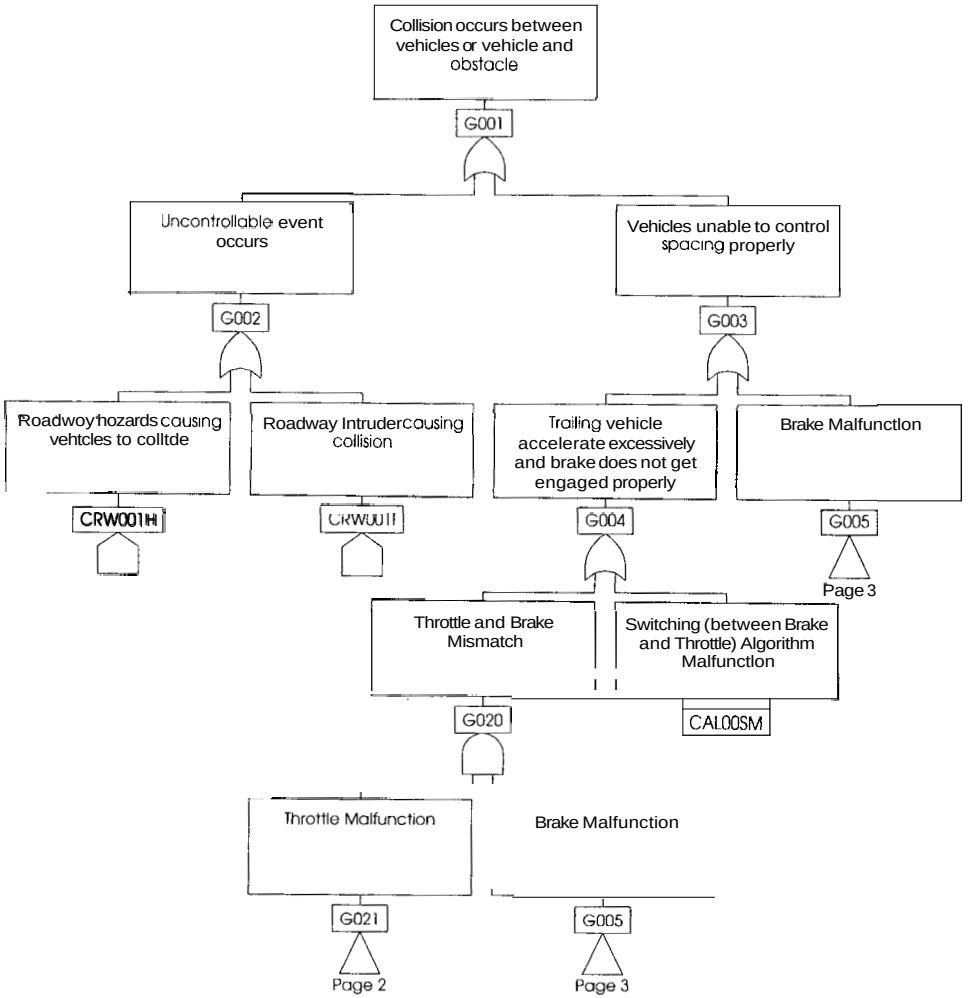
Reference

B.1. *CAFTA for Windows, Fault Tree Analysis System, Version 3.1, User's Manual,* Electric Power Research Institute and Science Applications International Corporation, July 1995.

Appendix C.

FAULT TREE MODEL OF A LONGITUDINAL CONTROL SYSTEM

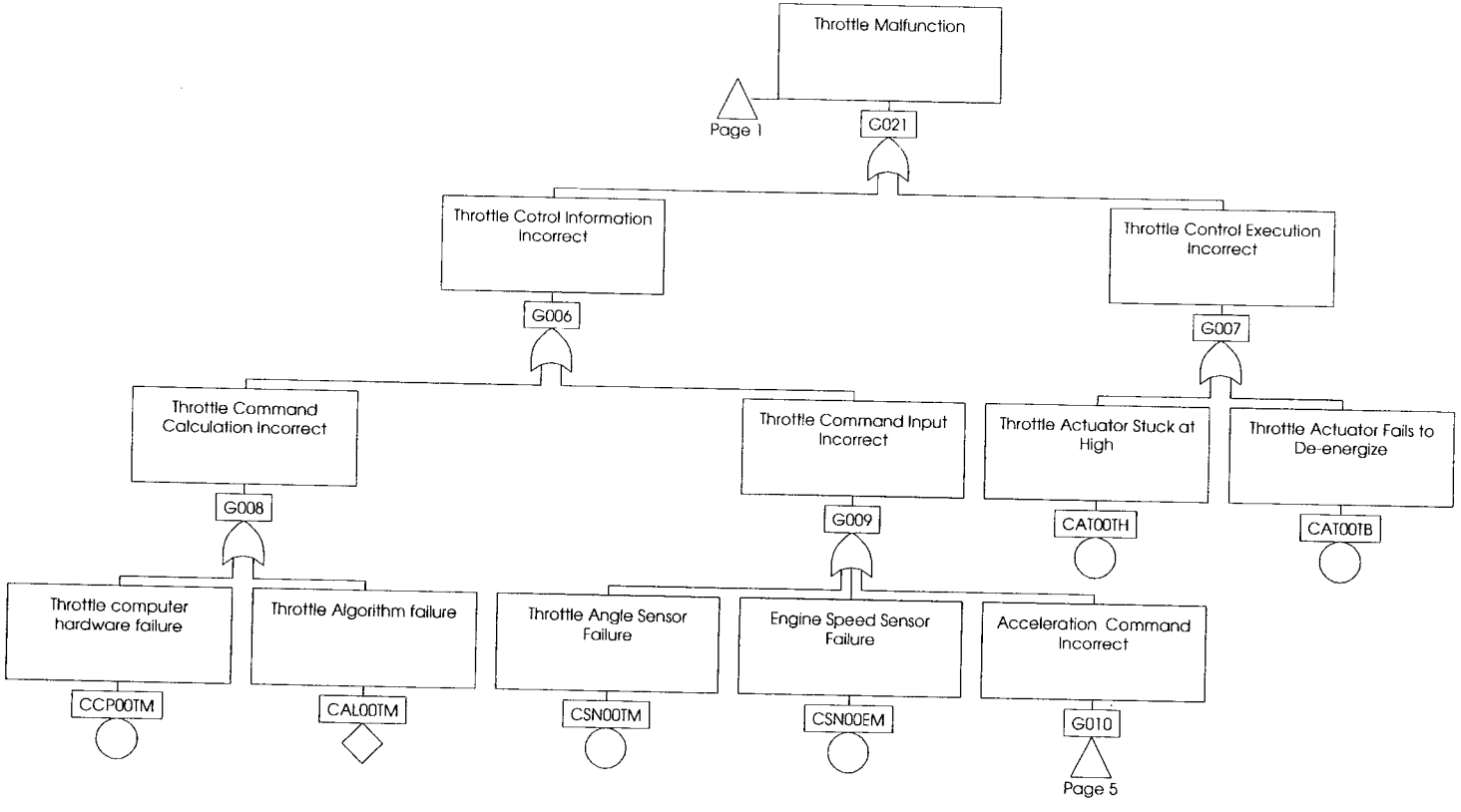
1. Fault Tree Model, pages **1-6**
2. Fault Tree Model, cross reference, page **7**
3. Gate Descriptions
4. Basic Events
5. Reliability Database
6. Cutset Sample



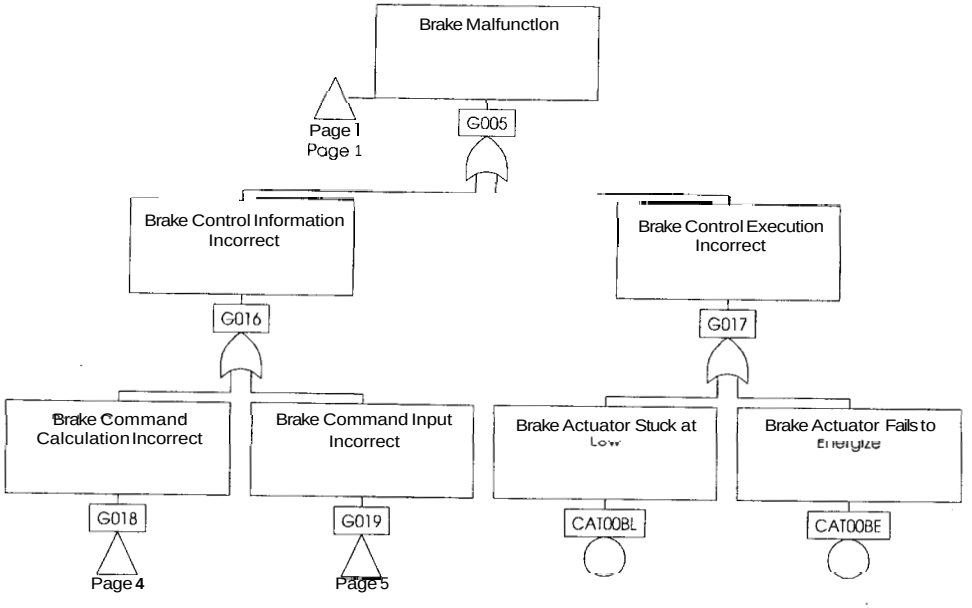
TITLE

Fault Tree for Longitudinal Contro Systems - MOU 325 Flnal Report

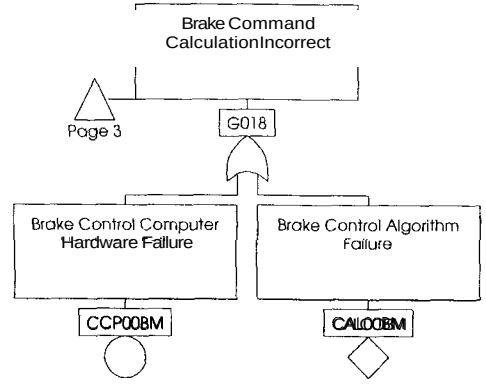
DRAWING NUMBER	DATE
Page 1	6/10/2000



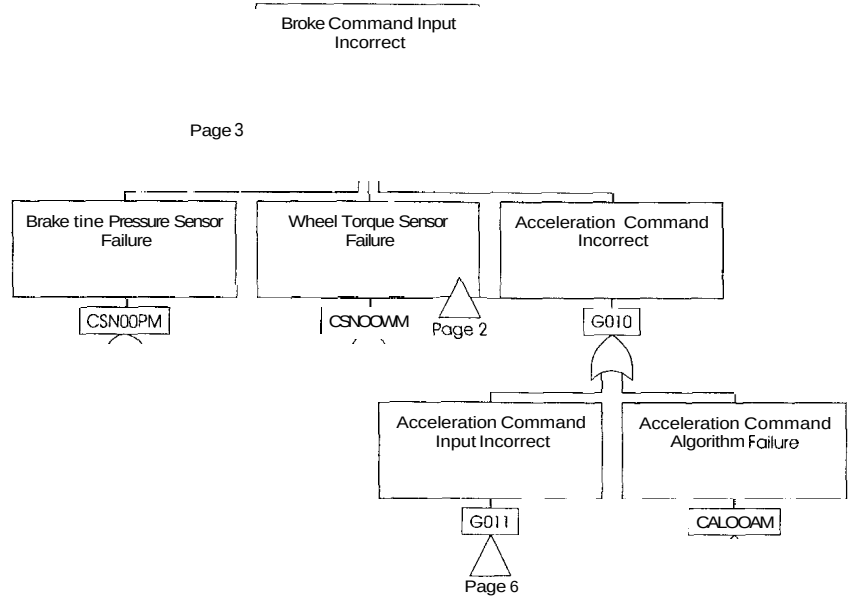
TITLE	Fault Tree for Longitudinal Control Systems - MOU 325 FInal Report
DRAWING NUMBER	Page 2
DATE	6/10/2000



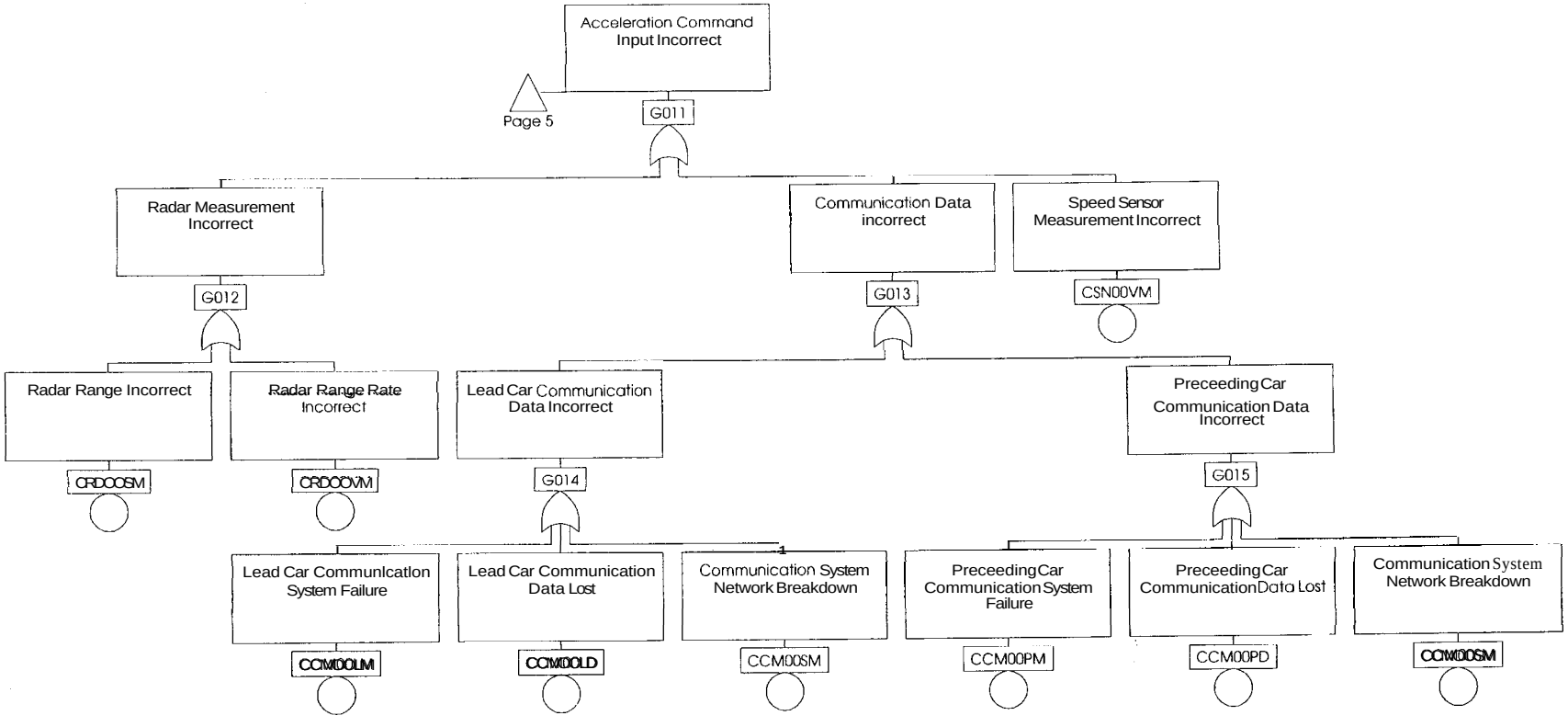
TITLE	Fault Tree for Longitudinal Control Systems - MOU 325 FInal Report
DRAWING NUMBER	Page 3
DATE	6/10/2000



TITLE	Fault Tree for Longitudinal Control Systems - MOU 325 Flnal Report
DRAWING NUMBER	Page 4
DATE	6/10/2000



TITLE: Fault Tree for Longitudinal Control Systems - MOU 325 FInal Report	
DRAWING NUMBER: Page 5	DATE: 6/10/2000



TITLE

Fault Tree for Longitudinal Control Systems - MOU 325 FInal Report

DRAWING NUMBER	DATE
Page 6	6/10/2000

Gate/Event Name	Page	Zone
CALOOAM	5	3
CALOOBM	4	2
CALO0SM	1	3
CAL00TM	2	2
CATOOBE	3	4
CATOOBL	3	3
CATOOTB	2	6
CATOOTH	2	5
CCMOOLD	6	3
CCMOOLM	6	2
CCMO0PD	6	6
CCMO0PM	6	5
CCMOOSM	6	4
CCMO0SM	6	7
CCPOOBM	4	1
CCPOOTM	2	1
CRDO0SM	6	1
CRDO0VM	6	2
CRW001H	1	1
CRWOOlI	1	2
CSNO0EM	2	4
CSNOOPM	5	1
CSNOOTM	2	3
CSNO0VM	6	5
CSNO0WM	5	2
G001	1	2
G002	1	1
G003	1	3
G004	1	3
G005	1	3
G005	1	4
G005	3	2
G006	2	3
G007	2	5
G008	2	1
G009	2	4
G010	2	5
G010	5	3
G011	5	2
G011	6	3
G012	6	1
G013	6	4
G014	6	3
G015	6	6
G016	3	1
G017	3	3
G018	3	1
G018	4	1
G019	3	2
G019	5	2
G020	1	2
G021	1	2
G021	2	4

Gate/Event Name	Page	Zone

TITLE

Fault Tree for Longitudinal Control Systems - MOU 325 Flnal Report

DRAWING NUMBER	DATE
Page 7	6/10/2000

Free Format Report

NAME	DESC
G001	Collision occurs between vehicles or vehicle and obstacle
G002	Uncontrollable event occurs
G003	Vehicles unable to control spacing properly
G004	Trailing vehicle accelerate excessively and brake does not
G005	Brake Malfunction
G006	Throttle Cotrol Information Incorrect
GO07	Throttle Control Execution Incorrect
G008	Throttle Command Calculation Incorrect
GOO9	Throttle Command Input Incorrect
G010	Acceleration Command Incorrect
G011	Acceleration Command Input Incorrect
G012	Radar Measurement Incorrect
GO13	Communication Data Incorrect
G014	Lead Car Communication Data Incorrect
G015	Preceeding Car Communication Data Incorrect
G016	Brake Control Information Incorrect
G017	Brake Control Execution Incorrect
G018	Brake Command Calculation Incorrect
G019	Brake Command Input Incorrect
G020	Throttle and Brake Mismatch
G021	Throttle Malfunction

Free Format Report

NAME	UNITS	DESC	SYM	PROB
CAL00AM	N	Acceleration Command Algorithm Failure	u	1.00E-06
CAL00BM	N	Brake Control Algorithm Failure	u	1.00E-06
CALO0SM	N	Switching (between Brake and Throttle) Algorith	u	1.00E-06
CALOOTM	N	Throttle Algorithm failure	u	1.00E-06
CAT00BE	H	Brake Actuator Fails to Energize	b	1.00E-05
CATO0BL	H	Brake Actuator Stuck at Low	b	1.00E-05
CATO0TB	H	Throttle Actuator Fails to De-energize	b	1.00E-05
CATO0TH	H	Throttle Actuator Stuck at High	b	1.00E-05
CCMOOLD	H	Lead Car Communication Data Lost	b	1.00E-04
CCMO0LM	H	Lead Car Communication System Failure	b	1.00E-04
CCMO0PD	H	Preceeding Car Communication Data Lost	b	1.00E-04
CCMO0PM	H	Preceeding Car Communication System Failure	b	1.00E-04
CCMO0SM	H	Communication System Network Breakdown	b	1.00E-04
CCPO01M	H	Throttle computer hardware failure	b	1.00E-04
CCPOOBM	H	Brake Control Computer Hardware Failure	b	1.00E-04
CCPO0TM	H	Throttle computer hardware failure	b	1.00E-04
CRDO0SM	H	Radar Range Incorrect	b	1.00E-04
CRDOOVM	H	Radar Range Rate Incorrect	b	1.00E-04
CRWO01H	H	Roadway hazards causing vehicles to collide	e	1.00E-06
CRWOOlI	H	Roadway Intruder causing collision	e	1.00E-05
CSNO0AM	H	Throttle Angle Sensor Failure	b	1.00E-05
CSNOOEM	H	Engine Speed Sensor Failure	b	1.00E-05
CSNO0PM	H	Brake Line Pressure Sensor Failure	b	1.00E-05
CSNO0TM	H	Throttle Angle Sensor Failure	b	1.00E-05
CSNO0VM	H	Speed Sensor Measurement Incorrect	b	1.00E-05
CSNO0WM	H	Wheel Torque Sensor Failure	b	1.00E-05

Free Format Report

TYPE	RATE	UNITS	DESC
AL M	1.0E-6	N	Algorithm Malfunction
AT B	1.0E-5	H	Actuator Fails to De-energize
AT E	1.OE-5	H	Actuator Fails to Energize
AT H	1.0E-5	H	Actuator Stuck at High
AT L	1.0E-5	H	Actuator Fails Low
CM D	1.0E-4	H	Communication Data Lost
CM M	1.0E-4	H	Communication System Malfunction
CP M	1.0E-4	H	Computer Malfunction
RD M	1.0E-4	H	Radar Failure
RW H	1.0E-6	H	Roadway Hazard
RW I	1.0E-5	H	Roadway Intruder
SN M	1.0E-5	H	Sensor Malfunction

Cutsets with Descriptions Report

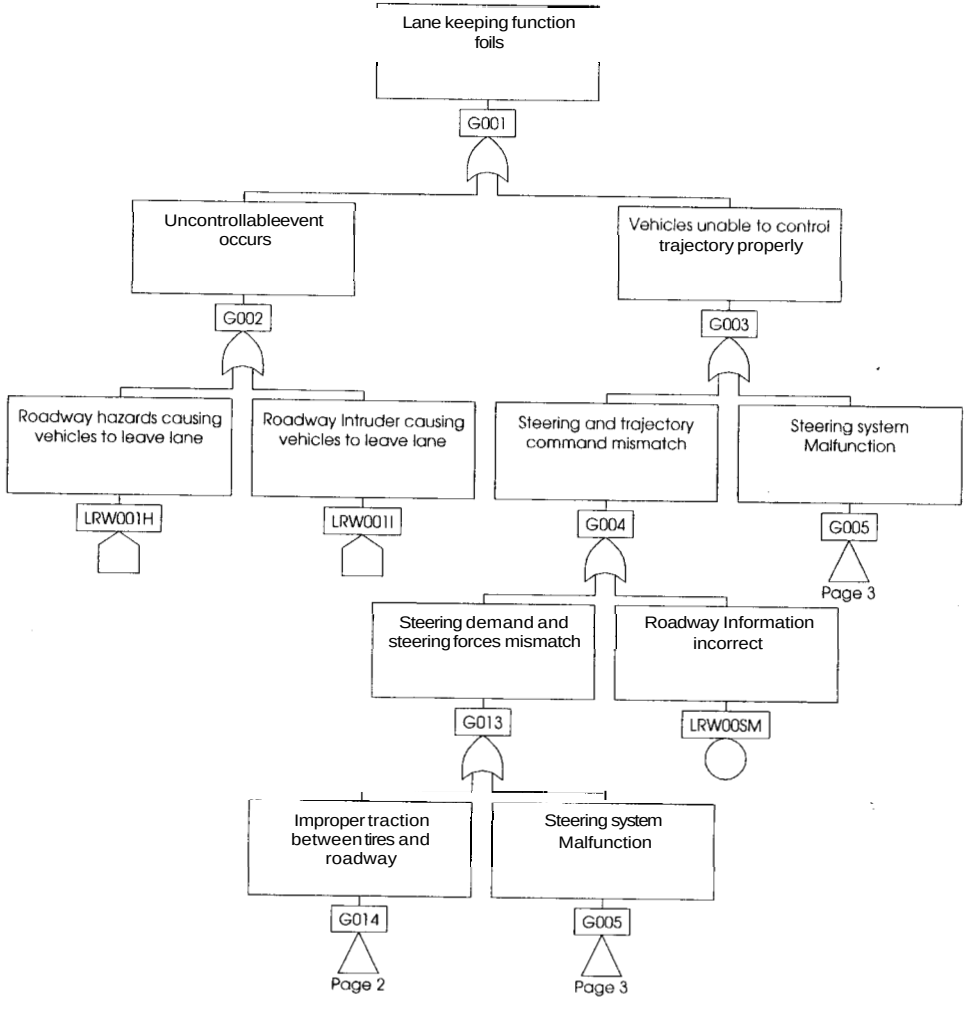
G001 = 8.60E-04

#	Inputs	Description	Rate	Exposure	Prob
1	CCMOOLD	Lead Car Communication Data Lost	1.00E-04	1.00E+00	1.00E-04
2	CCPOOBM	Brake Control Computer Hardware Failure	1.00E-04	1.00E+00	1.00E-04
3	CRDO0VM	Radar Range Rate Incorrect	1.00E-04	1.00E+00	1.00E-04
4	CCMOOLM	Lead Car Communication System Failure	1.00E-04	1.00E+00	1.00E-04
5	CCMO0SM	Communication System Network Breakdown	1.00E-04	1.00E+00	1.00E-04
6	CRDO0SM	Radar Range Incorrect	1.00E-04	1.00E+00	1.00E-04
7	CCMOOPM	Preceeding Car Communication System Failure	1.00E-04	1.00E+00	1.00E-04
8	CCMO0PD	Preceeding Car Communication Data Lost	1.00E-04	1.00E+00	1.00E-04
9	CRWOOlI	Roadway Intruder causing collision	1.00E-05	1.00E+00	1.00E-05
10	CSNO0VM	Speed Sensor Measurement Incorrect	1.00E-05	1.00E+00	1.00E-05
11	CSNOOWM	Wheel Torque Sensor Failure	1.00E-05	1.00E+00	1.00E-05
12	CSNOOPM	Brake Line Pressure Sensor Failure	1.00E-05	1.00E+00	1.00E-05
13	CAT0OBE	Brake Actuator Fails to Energize	1.00E-05	1.00E+00	1.00E-05
14	CATOOBL	Brake Actuator Stuck at Low	1.00E-05	1.00E+00	1.00E-05

Appendix D.

FAULT TREE MODEL OF A LATERAL CONTROL SYSTEM

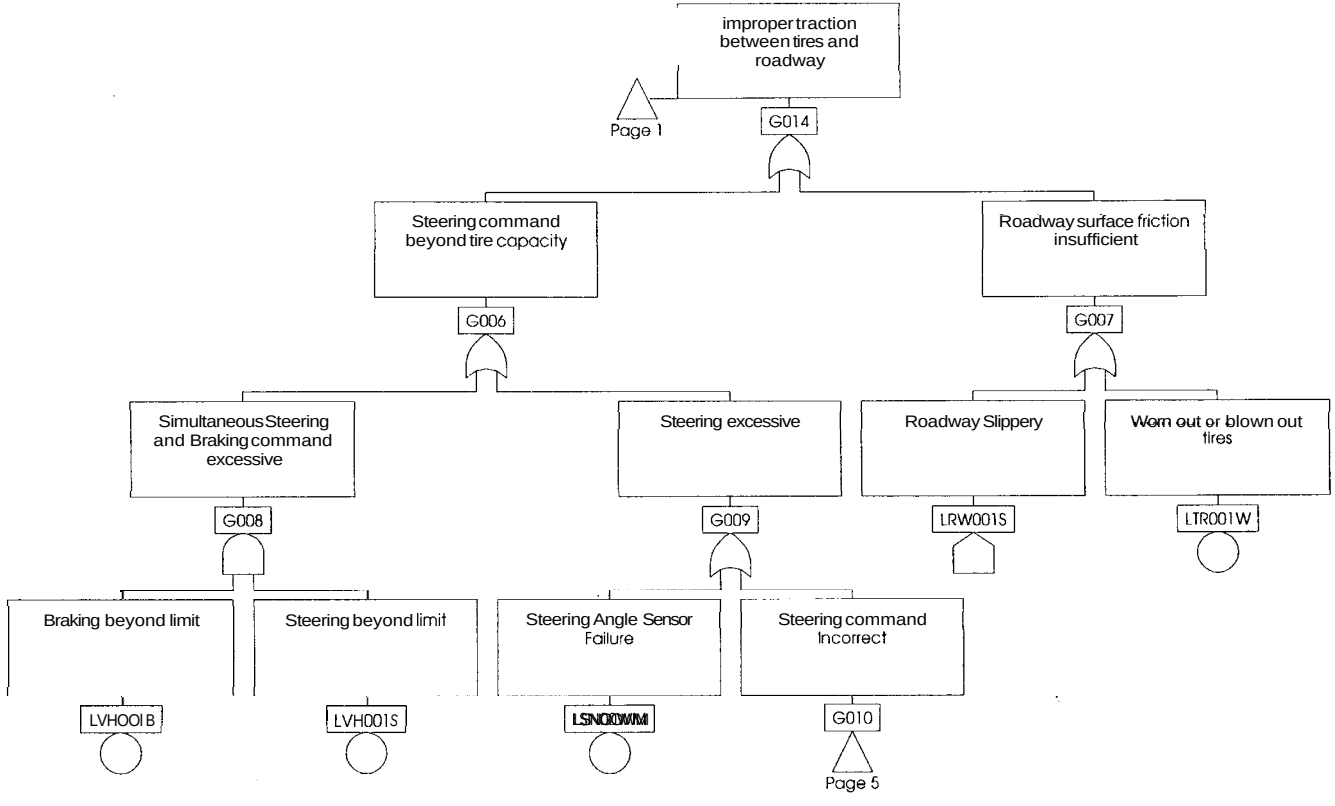
1. Fault Tree Model, pages 1-6
2. Fault Tree Model, cross reference, page 7
3. Gate Descriptions
4. Basic Events
5. Reliability Database
6. Cutset Sample



TITLE

Fault Tree for Lateral Control Systems - MOU325 FInal Report

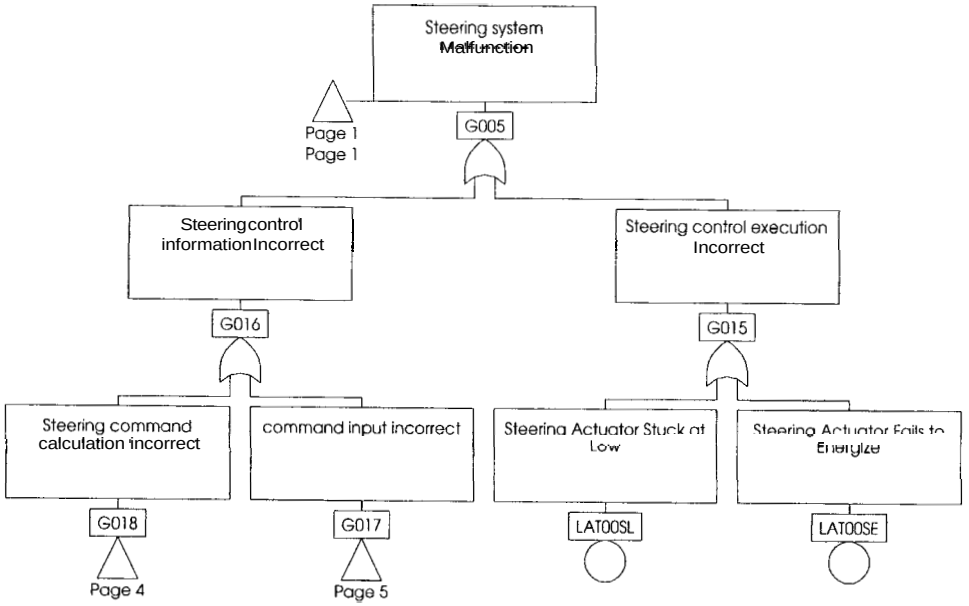
DRAWING NUMBER	DATE
Page 1	6/10/2000



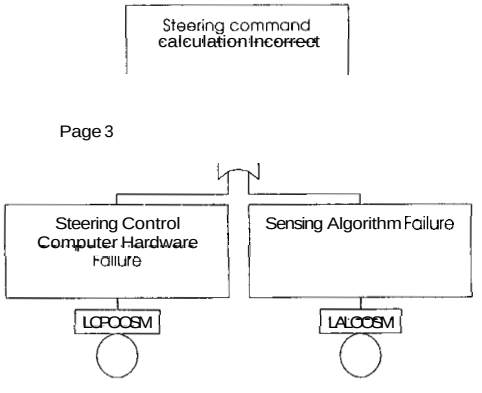
TITLE

Fault Tree for Lateral Control Systems - MOU325 Flnal Report

DRAWING NUMBER	DATE
Page 2	6/10/2000



TITLE	
Fault Tree for Lateral Control Systems - MOU325 Flnal Report	
DRAWING NUMBER	DATE
Page 3	6/10/2000

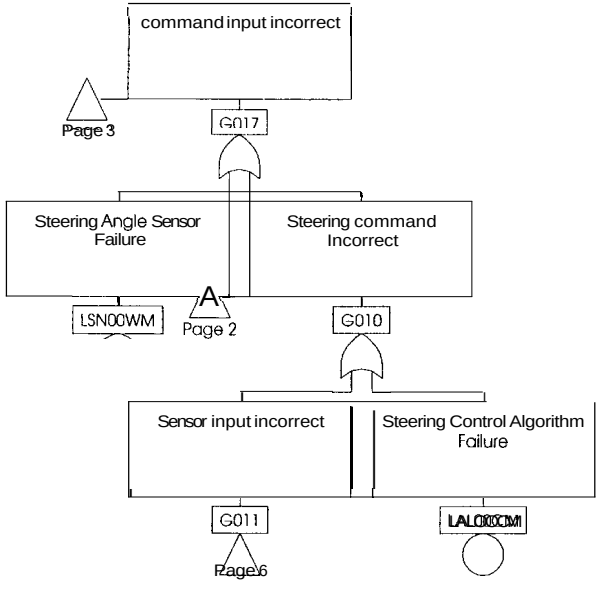


Page 3

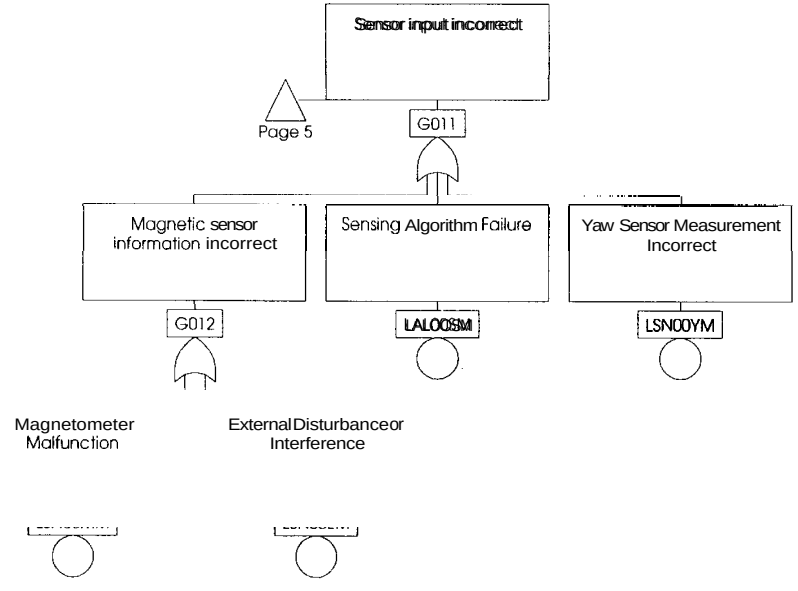
TITLE

Fault Tree for Lateral Control Systems - MOU325 Final Report

DRAWING NUMBER	DATE
Page 4	6/10/2000



TITLE	Fault Tree for Lateral Control Systems - MOU325 FInal Report
DRAWING NUMBER	Page 5
DATE	6/10/2000



TITLE	Fault Tree for Lateral Control Systems - MOU325 FInal Report
DRAWING NUMBER	Page 6
DATE	6/10/2000

Gate/Event Name	Page	Zone	Gate/Event Name	Page	Gate/Event Name	Page	Zone
G001	1	2					
G002	1	1					
G003	1	3					
G004	1	3					
G005	1	3					
G005	1	4					
G005	3	2					
G006	2	2					
G007	2	5					
G008	2	1					
G009	2	3					
G010	2	4					
G010	5	2					
G011	5	1					
G011	6	2					
G012	6	1					
G013	1	2					
G014	1	2					
G014	2	4					
G015	3	3					
G016	3	1					
G017	3	2					
G017	5	1					
G018	3	1					
G018	4	1					
LALOOCM	5	2					
LALOOSM	4	2					
LALOOSM	6	2					
LATOOSE	3	4					
LATOOSL	3	3					
LCPOOSM	4	1					
LRWO01H	1	1					
LRW001I	1	2					
LRWO01S	2	4					
LRWO0SM	1	3					
LSNOOEM	6	2					
LSNO0MM	6	1					
LSNOOWM	2	3					
LSNOOWM	5	1					
LSNOOYM	6	3					
LTROOlW	2	5					
LVH001B	2	1					
LVHOOlS	2	2					

TITLE

Fault Tree for Lateral Control Systems - MOU325 Final Report

DRAWING NUMBER	DATE
Page 7	6/10/2000

Free Format Report

NAME	DESC
GO01	Lane keeping function fails
G002	Uncontrollable event occurs
G003	Vehicles unable to control trajectory properly
G004	Steering and trajectory command mismatch
G005	Steering system Malfunction
G006	Steering command beyond tire capacity
G007	Roadway surface friction insufficient
G008	Simultaneous Steering and Braking command excessive
GOO9	Steering excessive
G010	Steering command Incorrect
G011	Sensor input incorrect
G012	Magnetic sensor information incorrect
G013	Steering demand and steering forces mismatch
G014	Improper traction between tires and roadway
G015	Steering control execution incorrect
G016	Steering control information incorrect
G017	command input incorrect
G018	Steering command calculation incorrect

Free Format Report

NAME	UNITS	DESC	SYM	PROB
LAL00CM	N	Steering Control Algorithm Failure	b	1.00E-06
LAL00SM	N	Sensing Algorithm Failure	b	1.00E-06
LAT00SE	H	Steering Actuator Fails to Energize	b	1.00E-05
LAT00SL	H	Steering Actuator Stuck at Low	b	1.00E-05
LCP00SM	H	Steering Control Computer Hardware Failure	b	1.00E-04
LRW001H	H	Roadway hazards causing vehicles to leave lane	e	1.00E-06
LRW001I	H	Roadway Intruder causing vehicles to leave lane	e	1.00E-05
LRW001S	H	Roadway Slippery	e	1.00E-06
LRWO0SM	H	Roadway Information incorrect	b	1.00E-05
LSNO0EM	H	External Disturbance or Interference	b	1.00E-05
LSNO0MM	H	Magnetometer Malfunction	b	1.00E-05
LSNOOSM	H	Steering Angle Sensor Failure	b	1.00E-05
LSNO0WM	H	Steering Angle Sensor Failure	b	1.00E-05
LSNO0YM	H	Yaw Sensor Measurement Incorrect	b	1.00E-05
LTR001W	H	Worn out or blown out tires	b	1.00E-05
LVH001B	H	Braking beyond limit	b	1.00E-05
LVHOOlS	H	Steering beyond limit	b	1.00E-05

Free Format Report

TYPE	RATE	UNITS	DESC
AL M	1.0E-6	N	Algorithm Malfunction
AT B	1.0E-5	H	Actuator Fails to De-energize
AT E	1.0E-5	H	Actuator Fails to Energize
AT H	1.0E-5	H	Actuator Stuck at High
AT L	1.0E-5	H	Actuator Fails Low
CP M	1.0E-4	H	Computer Malfunction
RW H	1.0E-6	H	Roadway Hazard
RW I	1.0E-5	H	Roadway Intruder
RW M	1.0E-5	H	Roadway Measurement Information Incorrect
RW S	1.0E-6	H	Roadway Slippery
SN M	1.0E-5	H	Sensor Malfunction
TR W	1.0E-5	H	Tire Worn Out or Blown Out
VH B	1.0E-5	H	Vehicle Braking Limit Exceeded
VH S	1.0E-5	H	Vehicle Steering Limit Exceeded

Cutsets with Descriptions Report

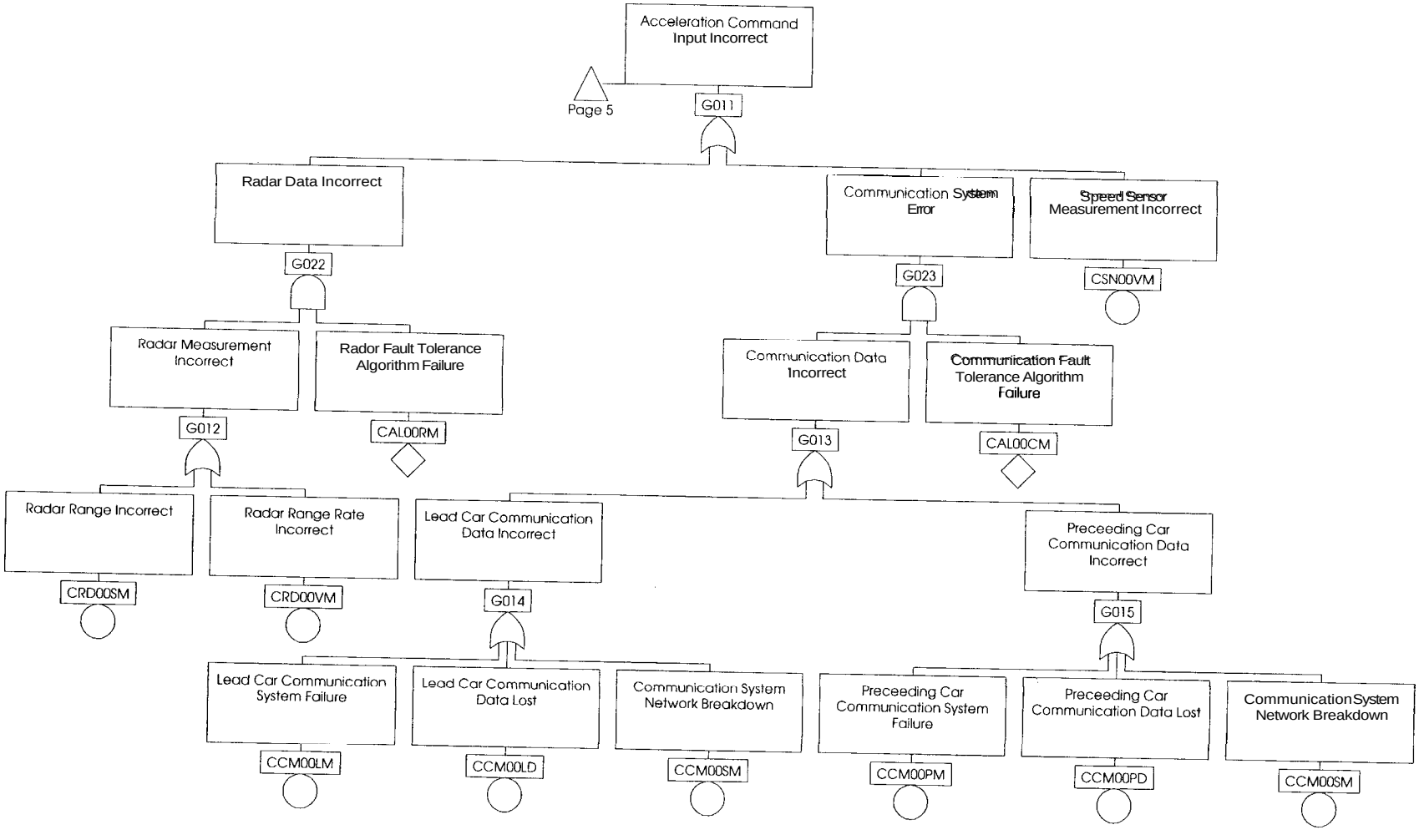
G001 = 1.90E-04

#	Inputs	Description	Rate	Exposure	Prob
1	LCPOOSM	Steering Control Computer Hardware Failure	1.00E-04	1.00E+00	1.00E-04
2	LSNO0MM	Magnetometer Malfunction	1.00E-05	1.00E+00	1.00E-05
3	LRWOOSM	Roadway Information incorrect	1.00E-05	1.00E+00	1.00E-05
4	LSNO0YM	Yaw Sensor Measurement Incorrect	1.00E-05	1.00E+00	1.00E-05
5	LSNOOEM	External Disturbance or Interference	1.00E-05	1.00E+00	1.00E-05
6	LATOOSL	Steering Actuator Stuck at Low	1.00E-05	1.00E+00	1.00E-05
7	LSNOOWM	Steering Angle Sensor Failure	1.00E-05	1.00E+00	1.00E-05
8	LTRO01W	Worn out **or** blown out tires	1.00E-05	1.00E+00	1.00E-05
9	LRWOOlI	Roadway Intruder causing vehicles to leave lane	1.00E-05	1.00E+00	1.00E-05
10	LATOOSE	Steering Actuator Fails to Energize	1.00E-05	1.00E+00	1.00E-05

Appendix E.

FAULT TREE MODEL OF A MODIFIED LONGITUDINAL CONTROL SYSTEM

1. Modified Portion of Fault Tree Model, page **6**
2. Modified List of Gate Descriptions
3. Modified List of Basic Events
4, Modified Cutset Sample



TITLE

Fault Tree for Longitudinal Contro Systems - MOU 325 Flnal Report

DRAWING **NUMBER**

Page 6

DATE

6/11/2000

Free Format Report

NAME	UNITS	DESC	SYM	PROB
CALO0AM	N	Acceleration Command Algorithm Failure	u	1.00E-06
CALO0BM	N	Brake Control Algorithm Failure	u	1.00E-06
CAL00CM	N	Communication Fault Tolerance Algorithm Failure	u	1.00E-06
CALO0RM	N	Radar Fault Tolerance Algorithm Failure	u	1.00E-06
CALO0SM	N	Switching (between Brake and Throttle) Algorith	u	1.00E-06
CALOOTM	N	Throttle Algorithm Failure	u	1.00E-06
CATO0BE	H	Brake Actuator Fails to Energize	b	1.00E-05
CATO0BL	H	Brake Actuator Stuck at Low	b	1.00E-05
CATOOTB	H	Throttle Actuator Fails to De-energize	b	1.00E-05
CAT00TH	H	Throttle Actuator Stuck at High	b	1.00E-05
CCMOOLD	H	Lead Car Communication Data Lost	b	1.00E-04
CCMO0LM	H	Lead Car Communication System Failure	b	1.00E-04
CCMO0PD	H	Preceeding Car Communication Data Lost	b	1.00E-04
CCMOOPM	H	Preceeding Car Communication System Failure	b	1.00E-04
CCMO0SM	H	Communication System Network Breakdown	b	1.00E-04
CCPO01M	H	Throttle computer hardware failure	b	1.00E-04
CCPOOBM	H	Brake Control Computer Hardware Failure	b	1.00E-04
CCPOOTM	H	Throttle computer hardware failure	b	1.00E-04
CRDO0SM	H	Radar Range Incorrect	b	1.00E-04
CRDO0VM	H	Radar Range Rate Incorrect	b	1.00E-04
CRWO01H	H	Roadway hazards causing vehicles to collide	e	1.00E-06
CRWOOlI	H	Roadway Intruder causing collision	e	1.00E-05
CSNO*0A*M	H	Throttle Angle Sensor Failure	b	1.00E-05
CSNOOEM	H	Engine Speed Sensor Failure	b	1.00E-05
CSNO0PM	H	Brake Line Pressure Sensor Failure	b	1.00E-05
CSNOOTM	H	Throttle Angle Sensor Failure	b	1.00E-05
CSNO0VM	H	Speed Sensor Measurement Incorrect	b	1.00E-05
CSNO0WM	H	Wheel Torque Sensor Failure	b	1.00E-05

Free Format Report

NAME	DESC
G001	Collision occurs between vehicles or vehicle and obstacle .
G002	Uncontrollable event occurs
G003	Vehicles unable to control spacing properly
G004	Trailing vehicle accelerate excessively and brake does not
G005	Brake Malfunction
G006	Throttle Cotrol Information Incorrect
G007	Throttle Control Execution Incorrect
G008	Throttle Command Calculation Incorrect
G009	Throttle Command Input Incorrect
G010	Acceleration Command Incorrect
G011	Acceleration Command Input Incorrect
G012	Radar Measurement Incorrect
G013	Communication Data Incorrect
G014	Lead Car Communication Data Incorrect
G015	Preceeding Car Communication Data Incorrect
G016	Brake Control Information Incorrect
G017	Brake Control Execution Incorrect
G018	Brake Command Calculation Incorrect
G019	Brake Command Input Incorrect
G020	Throttle and Brake Mismatch
G021	Throttle Malfunction
G022	Radar Data Incorrect
G023	Communication System Error

Cutsets with Descriptions Report

G001 = 1.60E-04

#	Inputs	Description	Rate	Exposure	Prob
1	CCPOOBM	Brake Control Computer Hardware Failure	1.00E-04	1.00E+00	1.00E-04
2	CSNOOPM	Brake Line Pressure Sensor Failure	1.00E-05	1.00E+00	1.00E-05
3	CAT00BE	Brake Actuator Fails to Energize	1.00E-05	2.00E+00	1.00E-05
4	CATOOBL	Brake Actuator Stuck at Low	1.00E-05	1.00E+00	1.00E-05
5	CSNO0VM	Speed Sensor Measurement Incorrect	1.00E-05	1.00E+00	1.00E-05
6	CRWOOlI	Roadway Intruder causing collision	1.00E-05	1.00E+00	1.00E-05
7	CSNO0WM	Wheel Torque Sensor Failure	1.00E-05	1.00E+00	1.00E-05