U.S. Department
of Transportation
**Federal Railroad
Administration**

Verification Methodology for Fault-Tolerant, Fail-Safe Computers Applied to Maglev Control Computer Systems

National Maglev Initiative
Washington, D.C. 20590

DOT/FRA/NMI - 92/26

May 1993

This document is available to the U.S. public
through the National Technical Information Service,
Springfield, Virginia 22161

[illegible]

NOTICE

This document is disseminated under the sponsorship of the Department of Transportation in the interest of information exchange. The United States Government assumes no liability for its contents or use thereof.

Technical Report Documentation Page

1. Report No.
DOT/FRA/NMI-92/26

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle

VERIFICATION METHODOLOGY FOR FAULT-TOLERANT, FAIL-SAFE COMPUTERS APPLIED TO MAGLEV CONTROL COMPUTER SYSTEM

5. Report Date
May 1993

6. Performing Organization Code

7. Authors (s)
Jaynarayan H. Lala, Gail A. Nagle, and Richard E. Harper

8. Performing Organization Report No.
CSDL-R-2491

9. Performing Organization Name and Address

The Charles Stark Draper laboratory, Inc.
555 Technology Square
Cambridge, MA 02139-3563

10. Work Unit No. (TRAIS)

11. Contract or Grant No.
DTFR53-91-C-00043

12. Sponsoring Agency Name and Address

US Department of Transportation
Federal Railroad Administration
Office of Research and Development
Washington, DC 20590

13. Type of Report and Period Covered
Final Report
July 1991 - May 1993

14. Sponsoring Agency Code

15. Supplementary Notes

Contracting Officer's Technical Representative: Mr. George Anagnostopoulos, USDOT/VNTSC

16. Abstract

Operation and control of the United States Maglev transportation system requires high levels of automation. Maglev control systems are required to make many decisions in real-time. The consequences of an incorrect or late decision could be catastrophic. The Maglev control computer system should therefore be designed to verifiably possess high reliability and safety as well as high availability to make Maglev a dependable and attractive transportation alternative to the public.

A Maglev control computer system has been designed using a design-for-validation methodology developed earlier under NASA and SDIO sponsorship for real-time aerospace applications. The present study starts by defining the maglev mission scenario and ends with the definition of a maglev control computer architecture. Key intermediate steps included definitions of functional and dependability requirements, synthesis of two candidate architectures, development of qualitative and quantitative evaluation criteria, and analytical modeling of the dependability characteristics of the two architectures. Both candidate architectures are variations of a distributed hierarchical architecture consisting of vehicle on-board computers, wayside zone computers, a central computer facility, and communication links between these entities.

Finally, the applicability of the design-for-validation methodology was also illustrated by applying it to the German Transrapid TR07 maglev control system.

17. Key Words
Maglev Control Computer
Fail-Safe Design
Fault-Tolerant Computer
Design for Verification Methodology
Safety and Availability

18. Distribution Statement
This document is available to the U.S. public through the National Technical Information Service, Springfield, Virginia 22161

19. Security Classif. (of this report)
Unclassified

20. Security Classif. (of this page)
Unclassified

21. No. of Pages
206

22. Price

Form DOT F 1700.7 (8-72) Reproduction of completed page authorized

METRIC/ENGLISH CONVERSION FACTORS

ENGLISH TO METRIC

LENGTH (APPROXIMATE)

1 inch (in.)	=	2.5 centimeters (cm)
1 foot (ft)	=	30 centimeters (cm)
1 yard (yd)	=	0.9 meter (m)
1 mile (mi)	=	1.6 kilometers (km)

AREA (APPROXIMATE)

1 square inch (sq in, in^2)	=	6.5 square centimeters (cm^2)
1 square foot (sq ft, ft^2)	=	0.09 square meter (m^2)
1 square yard (sq yd, yd^2)	=	0.8 square meter (m^2)
1 square mile (sq mi, mi^2)	=	2.6 square kilometers (km^2)
1 acre = 0.4 hectares (he)	=	4,000 square meters (m^2)

MASS - WEIGHT (APPROXIMATE)

1 ounce (oz)	=	28 grams (gr)
1 pound (lb)	=	.45 kilogram (kg)
1 short ton	=	2,000 pounds (lb) = 0.9 tonne (t)

VOLUME (APPROXIMATE)

1 teaspoon (tsp)	=	5 milliliters (ml)
1 tablespoon (tbsp)	=	15 milliliters (ml)
1 fluid ounce (fl oz)	=	30 milliliters (ml)
1 cup (c)	=	0.24 liter (l)
1 pint (pt)	=	0.47 liter (l)
1 quart (qt)	=	0.96 liter (l)
1 gallon (gal)	=	3.8 liters (l)
1 cubic foot (cu ft, ft^3)	=	0.03 cubic meter (m^3)
1 cubic yard (cu yd, yd^3)	=	0.76 cubic meter (m^3)

TEMPERATURE (EXACT)

[(x - 32) (5/9)] °F = y°C

METRIC TO ENGLISH

LENGTH (APPROXIMATE)

1 millimeter (mm)	=	0.04 inch (in)
1 centimeter (cm)	=	0.4 inch (in)
1 meter (m)	=	3.3 feet (ft)
1 meter (m)	=	1.1 yards (yd)
1 kilometer (km)	=	0.6 mile (mi)

AREA (APPROXIMATE)

1 square centimeter (cm^2)	=	0.16 square inch (sq in, in^2)
1 square meter (m^2)	=	1.2 square yards (sq yd, yd^2)
1 square kilometer (kn^2)	=	0.4 square mile (sq mi, mi^2)
1 hectare (he)	=	10,000 square meters (m^2) = 2.5 acres

MASS - WEIGHT (APPROXIMATE)

1 gram (gr)	=	0.036 ounce (oz)
1 kilogram (kg)	=	2.2 pounds (lb)
1 tonne (t)	=	1,000 kilograms (kg) = 1.1 short tons

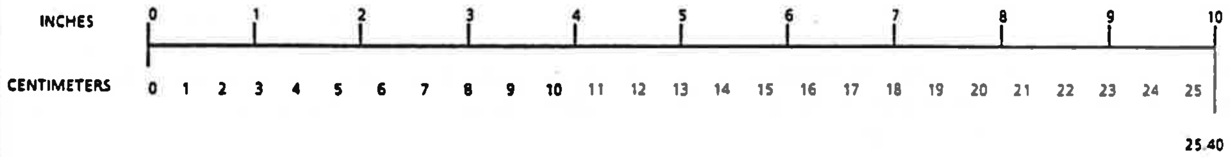
VOLUME (APPROXIMATE)

1 milliliter (ml)	=	0.03 fluid ounce (fl oz)
1 liter (l)	=	2.1 pints (pt)
1 liter (l)	=	1.06 quarts (qt)
1 liter (l)	=	0.26 gallon (gal)
1 cubic meter (m^3)	=	36 cubic feet (cu ft, ft^3)
1 cubic meter (m^3)	=	1.3 cubic yards (cu yd, yd^3)

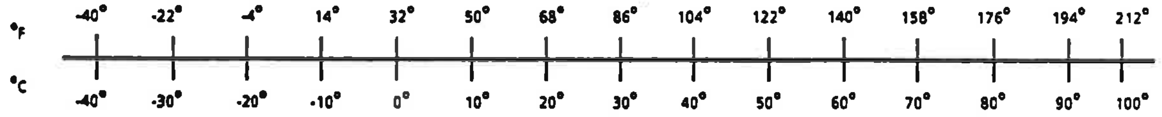
TEMPERATURE (EXACT)

[(9/5) y + 32] °C = x °F

QUICK INCH-CENTIMETER LENGTH CONVERSION



QUICK FAHRENHEIT-CELSIUS TEMPERATURE CONVERSION



For more exact and or other conversion factors, see NBS Miscellaneous Publication 286, Units of Weights and Measures. Price $2.50. SD Catalog No. C13 10286.

EXECUTIVE SUMMARY

Magnetically levitated (Maglev) vehicles operating on dedicated guideways at speeds of 500 km/hr are an emerging transportation alternative to short-haul air and high-speed rail. These systems which are levitated at a height of 8-200 mm above the guideway may be uniquely characterized as low flying vehicles constrained to guiderails. They have the potential to offer a service significantly more dependable than air and with less operating cost than both air and high-speed rail.

Operation and control of this unique transportation system requires high levels of automation. The control system will be comprised of sensors, actuators, communication links, and computers whose collective activities will be coordinated and directed by control and decision making software. Maglev control systems are required to make many decisions in real-time. The consequences of an incorrect or late decision could be catastrophic. Since human safety is at risk, the automated control systems must work correctly and in a timely manner under all expected operating conditions.

Since hardware and software are expected to fail at some time during the life of the system, the system must be able to tolerate these faults while maintaining normal operations or, in the exceptional, worst case scenario, fail in a safe manner. That is, the system should continue to function correctly as a whole even when parts have failed. This is referred to as fault-tolerant operation. If this is not possible, the system must be able to systematically shut down in a safe manner, i.e., fail-safe operation.

A design-for-validation methodology for the development of fault tolerant computer control systems for real-time operation was defined by the Draper Laboratory under sponsorship of the National Aeronautics and Space Administration and the Strategic Defense Initiative Office. This methodology has been used to design the computer system for the United States Maglev Transportation System. A key to the successful application of this methodology is the early involvement of the computer system designer in the overall conceptual design of the subject vehicle to be controlled. Development of performance specifications such as throughput, memory size, response time; safety requirements such as reliability or maximum acceptable probability of failure; and operational requirements such as availability or probability of fail-safe and/or fault-tolerant operation etc. must be defined *concurrently* with the design of the vehicle itself.

The present study starts by defining the maglev mission scenario and ends with the definition of a maglev control computer architecture. Execution of key intermediate steps in the design-for-validation methodology bridges the gap between these end points. After defining the mission scenario, all the functions that fall in the maglev command, control, and communication domain were identified. Their performance requirements were then

quantified. The functions were grouped into three categories: vehicle control, protection, and supervision. The dependability requirements for the maglev transportation system including safety, reliability, availability, and maintainability, were defined and quantified.

A distributed hierarchical architecture consisting of vehicle on-board computers, wayside zone computers, a central computer facility, and communication links between these entities was synthesized to meet the functional and dependability requirements of the maglev. Two variations of the basic architecture were developed: in the Smart Vehicle Architecture (SVA) the on-board computer has the primary responsibility for train control and the wayside zone computers provide backup and consistency checking; in Zone Control Architecture (ZCA) these roles are reversed. In both cases, the central computer facility performs the route planning, scheduling, dispatching and other system level functions.

A set of qualitative and quantitative evaluation criteria for the maglev control computer system were developed. The dependability characteristics of the proposed architecture were modeled analytically. An empirical test and evaluation plan for the verification of a prototype of the computer system was also developed.

The verification methodology developed during the course of this study was applied to the control computer system of the German Transrapid TR07. The computer functional requirements, the overall computer architecture used in TR07 and the redundancy management strategies employed were discussed. A critical failure modes and effects analysis (FMEA) was performed on the TR07 control system. Analytical models for two selected subsystems were then developed. Finally, we presented some open issues which were raised by the FMEA and the discussion of redundancy management strategies. We were able to show that the verification methodology would be a valuable and effective tool in verifying the design of the Transrapid control computer system.

In a future study, the mission scenario and functional requirements should be refined in the context of the System Conceptual Definition studies performed by the four maglev contractor teams. Their impact on the proposed architecture should be investigated and the architecture should be refined, updating the performance and dependability models. Detailed trade-offs between SVA and ZCA should also be performed. These steps will lead to the prototyping of the key parts of the maglev control computer architecture which can then be subjected to empirical test and evaluation and a validation of the safety, reliability and other dependability characteristics.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

Table **Title** **Page**

1. Introduction

The Magnetically Levitated (Maglev) Transportation System will use vehicles which are suspended a short distance above a dedicated guideway by magnetic forces. These vehicles will also use magnetic forces for propulsion and guidance and will travel at speeds up to 500 kmph (311 mph). While the use of a guideway invites comparison to railway travel, in many respects the operation of Maglev vehicles more closely resembles the operation of aircraft than that of trains, since under normal operating conditions the vehicles make no contact with the guideway and travel at high speeds. Like air travel, the operation and control of this new means of transportation will, of necessity, be highly automated. However, Maglev is a unique mode of travel, different from both rail and air transportation systems.

1.1. Verification Methodology

The automatic control system for Maglev transportation will comprise sensors, actuators, communication links, and computers, whose collective activities will be coordinated and directed by control and decision making software. These components of the control system will be distributed among the Maglev vehicles, along the guideways at wayside zone controllers, and in central control facilities. Since human safety is at stake, the automatic control system must work correctly and in a timely manner under all expected operating conditions. Since hardware and software are expected to fail at some time during the life of a system, the system must be able to tolerate these faults while maintaining normal operations or, in the exceptional, worst case scenario, by failing in a safe manner. That is, for the Maglev transportation system, fault-tolerant operation will be the norm and fail-safe operation will be the acceptable, worst case exception. Furthermore, in order to avoid the delays and disruptions associated with unscheduled maintenance, the Maglev control system must have sufficient redundancy to allow vehicles to be reliably dispatched with failures. Only in this way can the system provide the high level of availability which its ridership will demand. It is expected that the Federal Railroad Administration, in order to ensure safety, will require the verification and validation of the operation of the entire Maglev system.

Since a Maglev transportation system will be critically dependent upon its automated controls, the safety, reliability, and availability of the control system must be critically examined in a systematic and methodical manner. To this end, a comprehensive design and verification methodology for the development of fault-tolerant, real-time control systems, which has been developed at Draper Laboratory, Inc., is being assessed for its applicability to the US Maglev System. The key steps in this methodology are outlined in Figure 1-1.

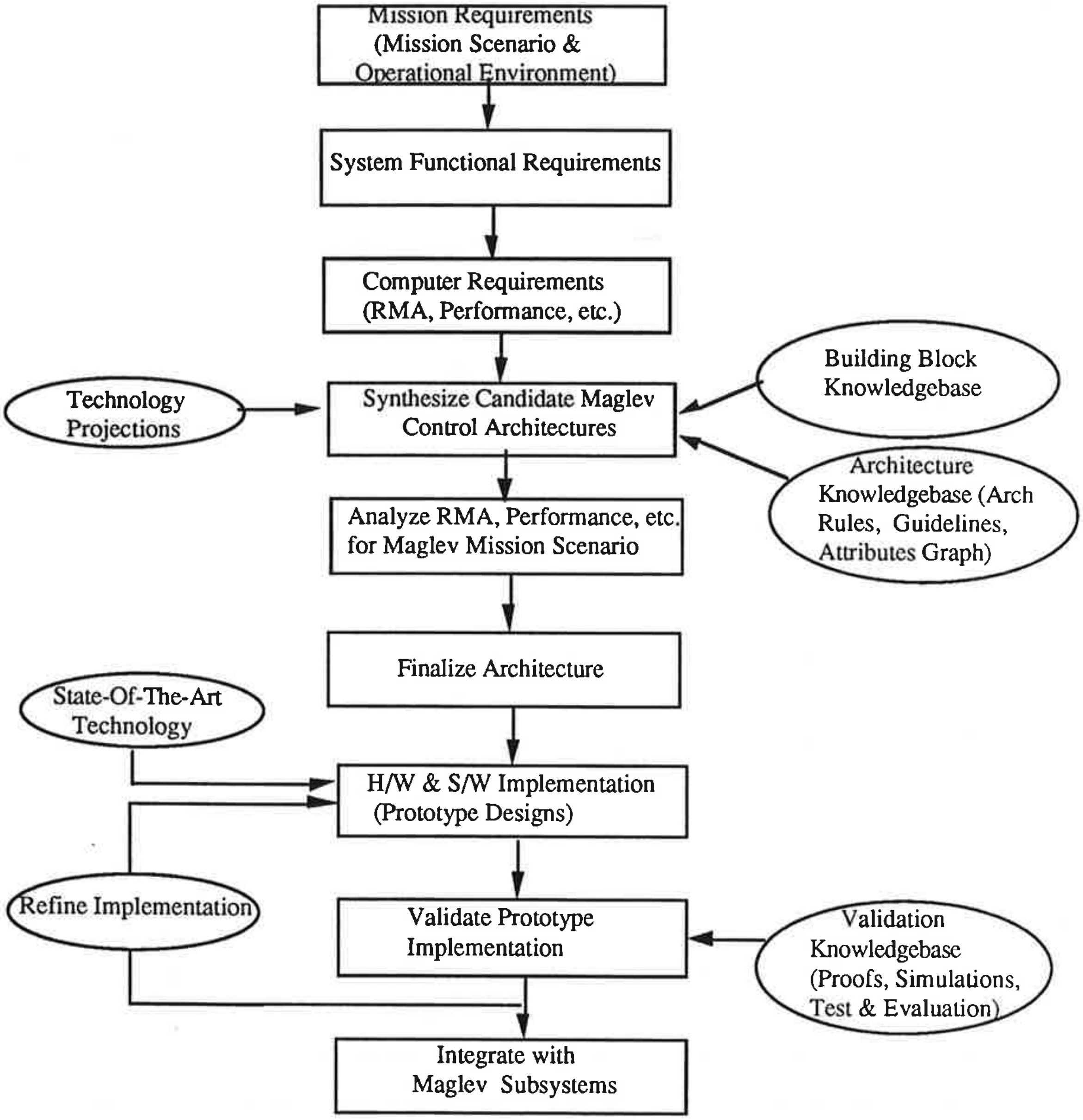


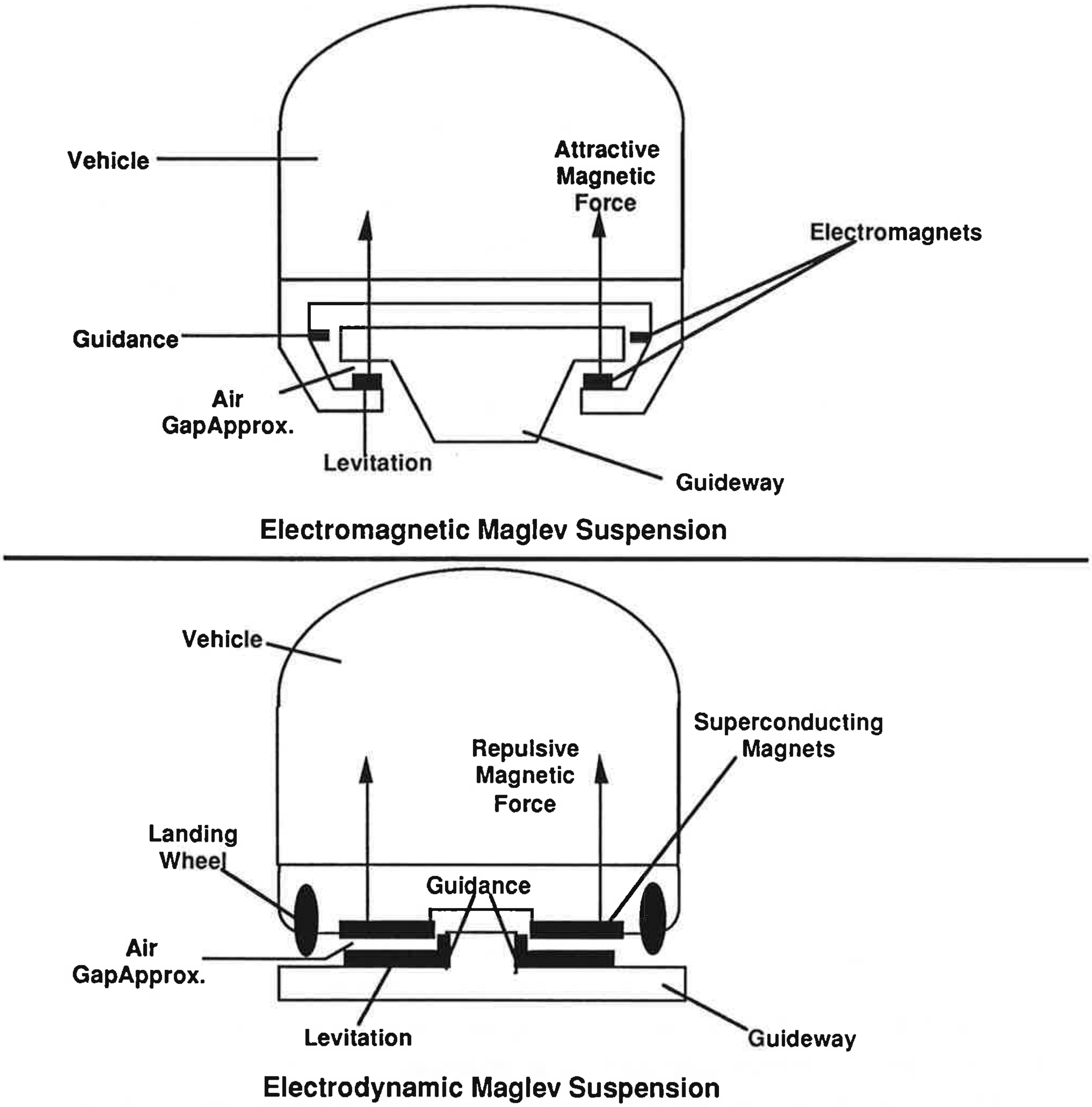
Figure 1-1. Design and Verification Methodology for Maglev Fault Tolerant Computers

The first step in this methodology is to develop a concept of operations (Con-Op) which defines the mission scenario, including the method of operation, the operational environment, and specifies the computer control functional requirements. While it is intended that the concept of operations presented here be considered as a strawman for the purpose of assessing the applicability to Maglev of the verification methodology, every effort has been made to present a realistic, accurate, and detailed picture of the operation of a Magnetic Levitation Transportation System. Furthermore, it is intended that the Maglev control system architecture itself be able to perform its many required functions, with generous margins for growth, for the most rigorously demanding Maglev mission scenario, method of operation, and system level requirements for safety and reliability. The concept of operations contained herein is based on information collected from several sources including material provided by the United States Department of Transportation [1], [2], [7], [9], published details of the operation of the Transrapid Maglev System developed by the Federal Republic of Germany [12], interviews with knowledgeable DOT personnel, and information provided by participants in the National Maglev Initiative who were awarded contracts under the Broad Agency Announcement 90-1, sponsored jointly by the DOT/FRA, the U.S. Department of Energy, and the U.S. Army Corps of Engineers.

1.2. Maglev Suspension Systems

The three primary functions basic to Maglev technology are levitation, propulsion, and guidance. No consensus currently exists on an optimum design to perform these functions [2]. For example, there are two principal means for providing levitation, electromagnetic suspension (EMS) and electrodynamic suspension (EDS), illustrated in Figure 1-2. EMS technology makes use of the attractive magnetic force created by the interaction of electromagnets on the vehicle and ferromagnetic rails on the guideway. EDS technology makes use of the repulsive magnetic force created when superconducting electromagnets on a moving vehicle induce currents in the guideway which in turn generate a magnetic field. Each technology has inherent advantages and disadvantages.

EMS requires a very small gap (approximately 8 mm or 3/8 in) between the vehicle and the guideway. Since the system is inherently unstable, i.e. the magnetic forces get stronger as the gap gets smaller, thereby further closing the gap, an EMS system requires active control of its gap size. Furthermore, guideway construction for EMS requires very fine tolerances to accommodate the small gap. However, EMS vehicles can hover in a stationary mode and do not require shielding from ambient magnetic fields in the passenger compartment.



Electromagnetic Maglev Suspension

Electrodynamic Maglev Suspension

Figure 1-2. A comparison of EMS and EDS suspension systems

On the other hand, EDS systems are inherently stable since the repulsive force increases as the gap size decreases, thereby maintaining the separation of the vehicle from the guideway. Furthermore, EDS systems are lighter than EMS systems of comparable capacity and performance and are, therefore, more energy efficient. In addition EDS systems can operate with a relatively large gap (6-10 in). However, since EDS does not provide levitation forces in stationary vehicles, EDS systems do not operate in a hover mode and therefore must use wheels or some other mechanical aid for takeoff and landing. Furthermore, EDS systems generate large ambient magnetic fields which require shielding of the passenger compartment. Both systems require some form of secondary suspension to provide a satisfactory ride quality. Superior ride quality characteristics can best be achieved by an actively controlled (i.e. computerized) secondary suspension system rather than by simple mechanical means.

Similar tradeoffs exist for both guidance and propulsion systems currently under consideration. Although, the levitation, propulsion and guidance technology which is ultimately chosen for the U.S. Maglev System will impact the control system, the intent in this concept of operations is to specify a system which is adequate for the most demanding of these alternatives. In this way a design for a robust control architecture for Maglev will emerge and the verification methodology can be assessed for its ability to assist in making design tradeoffs during the preliminary design phase and to reduce the burden of validating the completed control system.

1.3. Safety, Reliability, and Availability

The terms safety, reliability, and availability are fundamental concepts used by the verification methodology. Their meaning and significance are discussed at length in Section 4. However, since a precise definition of each is useful to the following discussion, the terms are briefly defined here as well. Safety $S(t)$ is the probability that a system will either perform its functions correctly or will fail in a way that does not disrupt other systems or jeopardize the safety of people associated with the system. The reliability $R(t)$ is a function of time, defined as the conditional probability that a system will perform correctly throughout the interval $[t_0,t]$, given that the system was performing correctly at time t_0. Availability $A(t)$ is a function of time, defined as the probability that a system is operating correctly at the instant of time t.

The control computer systems of both Maglev vehicles and aircraft have high safety requirements. However, subtle differences exist between their respective reliability and availability requirements. First, consider availability. Like aircraft, Maglev vehicles must maintain a safe separation from other vehicles. However, one malfunctioning aircraft does not pose an unavoidable obstacle to another in mid-flight, whereas a malfunctioning

Maglev vehicle, because of its attachment to the guideway, can only be circumvented by another vehicle at a station or a passing area. This imposes a more stringent availability requirement on the Maglev control computer system. Next, consider reliability and safety. The catastrophic failure of the control computer in a fly-by-wire aircraft would result in the loss of the vehicle because a fail-stop mode of operation does not exist. Hence, the reliability and the safety requirement of an avionics control system are equivalent. However, in the event of a failure of the control computer, Maglev vehicles do have a fail-stop mode of operation. Hence, the reliability and safety requirements of the Maglev control computer system are not the same. Although the safety requirement of the Maglev control system is equivalent to that of an aircraft, its reliability requirement is lower. Regardless of the level of these requirements, the need still exists for a methodology to verify that the control computer system actually achieves them.

Maglev vehicles will travel at a much higher speed than conventional trains and at a higher speed than conventional high speed rail. Although the higher speed increases transportation productivity, it implies more serious consequences if the control system fails to maintain safe train separation. Thus, the response time of the control system must be on a par with the response times of aircraft control systems. Furthermore, achieving a passenger load which makes the system economically viable, requires the shortest headways possible within the limits of safe train separation. This conflict between capacity and safety can be mitigated by the use of a fully automated and validated control system which offers significantly faster response times than one using human operators.

2. Maglev Mission Scenario and Operational Environment

In order to define the Maglev control computer system architecture and perform reliability, maintainability, availability and safety (RMAS) modeling of the computer system, a mission scenario must be defined for the Maglev transportation system. The mission scenario will describe the various operating modes for a fleet of Maglev vehicles. The RMAS analysis will describe the transitions from one mode of operation to the other, assigning conditions and probabilities to each transition.

2.1. Mission Scenario Overview

An Example: The ALS Mission Scenario

An example of a mission scenario is provided by comparison with another transportation system such as the Advanced Launch System (ALS) [4]. The ALS mission is to place heavy payloads into low earth orbit on a regularly scheduled basis. This example has been chosen because the design and verification methodology was successfully applied to the design of the guidance, navigation, and control computers for this unmanned, heavy lift rocket. The ALS vehicle consists of both reusable and non-reusable components. After a launch, the reusable components are recovered, refurbished, and re-assembled with a new set of expendable components and a new payload. The non-reusable equipment burns up on re-entry into the earth's atmosphere. A mission scenario state diagram for ALS is presented in Figure 2-1.

The operational scenario consists of seven phases: (1) the integration and checkout phase, during which the vehicle is assembled and its payload is installed in the vehicle assembly building, (2) the pre-launch checkout and tanking phase, during which the vehicle waits on the launch pad while it undergoes extensive pre-launch diagnostic testing and is fueled for its mission, (3) the pre-launch maintenance phase during which necessary repairs are made by a field maintenance crew, (4) the mission or boost phase, during which the vehicle performs the guidance, navigation and control needed to place the payload in orbit, (5) the on-orbit phase during which the payload performs its scheduled operation, (6) the recovery phase during which the reusable avionics and propulsion system components are retrieved after a controlled descent to earth, and (7) the restoration phase during which the reusable components are refurbished for another mission.

The RMAS analyses of a computer system are intimately connected to the failure rates of the components which make up the system. Failure rates are most often estimated by the standards established in the United States Department of Defense Military Standardization Handbook: Reliability Prediction of Electronic Equipment, usually referred to as MIL HDBK-217, which was originally published in 1965, but has been revised in 1974 (MIL

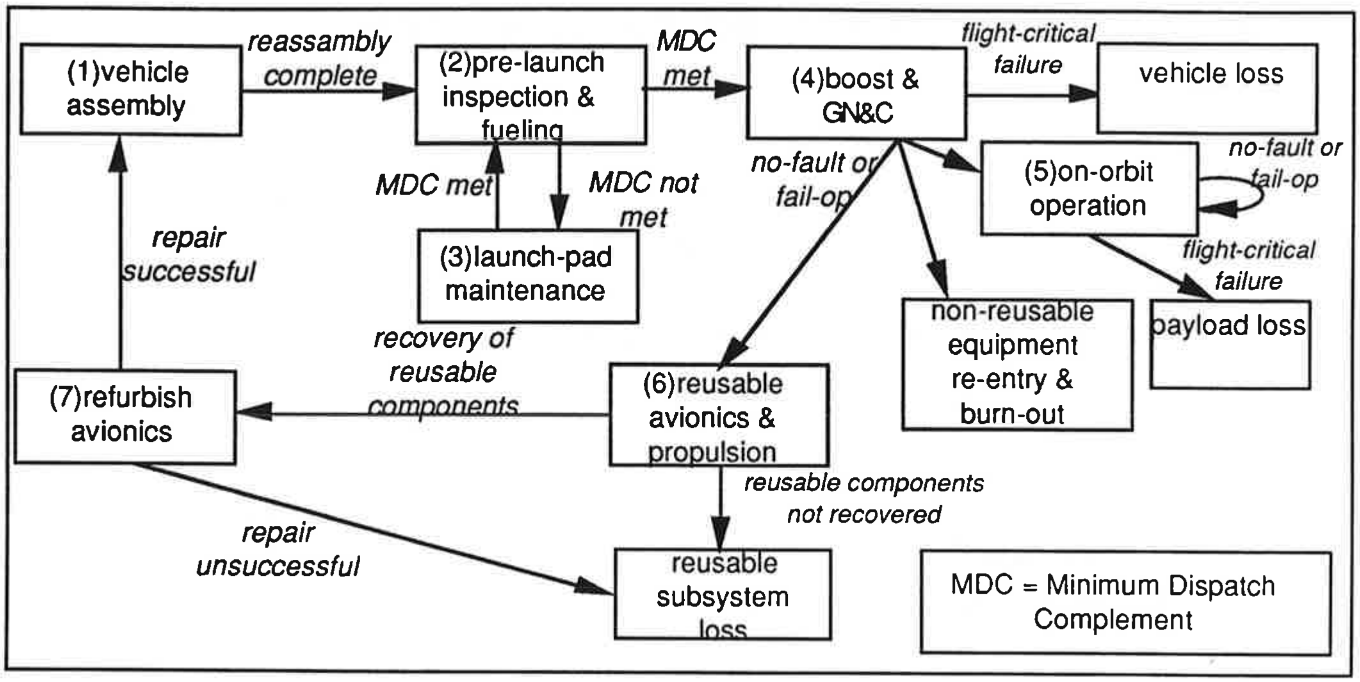


Figure 2-1. Advanced Launch System Mission Scenario State Diagram

HDBK-217B) and 1979 (MIL HDBK-217C). The most recent revision, dated January 2, 1990, is MIL HDBK-217E. MIL HDBK-217B uses the following relationship to predict the constant failure rate of an integrated circuit (IC):

$$\lambda = \pi_L \pi_Q (C_1 \pi_T + C_2 \pi_E) \pi_P \text{ failures per million hours}$$

where π_L is a learning factor, π_Q is a quality factor, π_T is a temperature factor, π_E is an environmental factor including temperature variability, π_P is a pin factor, and C_1 and C_2 are complexity factors. The learning factor represents the overall maturity of the fabrication process used to produce the IC and ranges from 1(proven technologies) to 10 (newer technologies). The quality factor represents the amount of device screening that is performed by the manufacturer, and ranges from 1 to 300. The temperature factor is a function of the device technology, packaging technology, operating temperature, and power dissipation. The environmental factor is a measure of the harshness of the environment. For example, components located in an air conditioned computer room have an environmental factor of 0.2. Components in a launched missile have an environmental factor of 10.0. The pin factor is a function of the number of pins on the IC package. Finally, the complexity factors reflect the number of gates for logic circuits and the number of bits for memories.

For purposes of assigning a value to π_E, the operational environment for phases 1, 2, 3, 6 and 7 of the ALS is assumed to correspond to the Ground, Fixed environment, described in MIL-HDBK-217E. The operational environment for the mission or boost phase is assumed to correspond to the Missile Launch environment and the operational environment for the on-orbit phase is assumed to correspond to the Space, Flight environment, both of which are also described in MIL-HDBK-217E.

The Maglev Mission Scenario

The following strawman Maglev mission scenario describes a day-in-the-life of a typical Maglev vehicle. In this scenario, there are five phases of operation for a Maglev vehicle: (1) stationary pre-run inspection, (2) complete routine daily maintenance, (3) normal operation consisting of non-stop travel between stations, (4) station stops and routine built-in-test (BIT) maintenance checks, and (5) extensive depot maintenance and repair. These are shown in Figure 2-2.

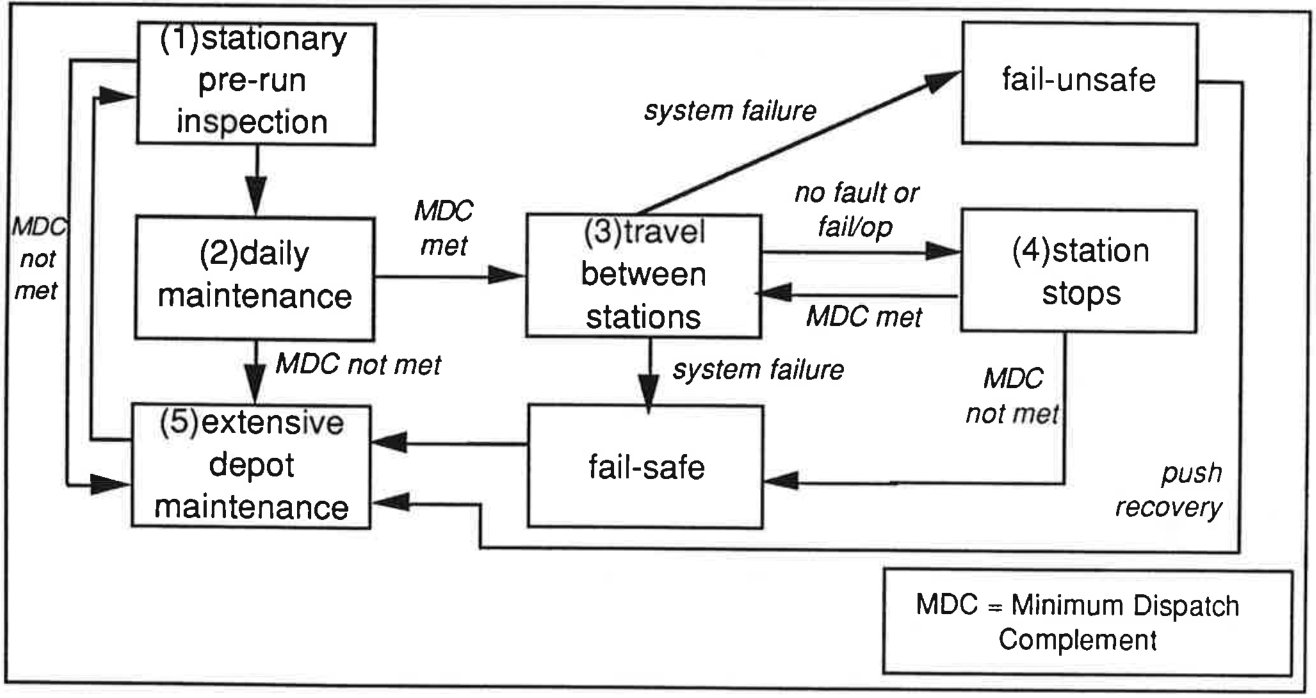


Figure 2-2. Maglev Mission Scenario Vehicle State Diagram

During the stationary pre-run inspection phase the vehicle may be powered or unpowered, and either resting on its wheels or skids or in a safe-hover mode, depending on whether it has an EDS or EMS suspension system, respectively. During this time it is performing internal self-checks on all its vital systems, as well as undergoing thorough interior and exterior cleaning. The vehicle is in this phase for one half hour.

During the complete routine daily maintenance phase, the maintenance crew inspects the vehicle, reviews fault logs and conducts the scheduled diagnostic tests called for by the vehicle life-cycle maintenance program. When necessary, the crew replaces components (Line Replaceable Modules or Line Replaceable Units) identified as faulty. In addition to maintenance using on-condition monitoring procedures which identify and replace failed components, normal maintenance also includes the scheduled replacement of non-failed components whose probability of failure has increased due to age and level of use. This preventive or dynamic maintenance, which is possible for non-electronic components, is based on measured conditions of individual vehicles and subsystems. If the testing and normally scheduled maintenance procedures do not identify any major repairs which are beyond the scope of this maintenance operation, the crew advances the vehicle to the normal operation phase. However, if major maintenance procedures are required, the vehicle is taken off-line and moved to a bay or a central repair facility for more extensive depot maintenance. The complete routine daily maintenance operation lasts for up to two hours and is conducted during a 24 hour cycle. Periodically, the vehicle is sent to the depot for an extensive inspection and overhaul. During the daily two hour time period (from 2:30 to 4:30 A.M.) allowed for phases 1 and 2 of vehicle maintenance, use of the guideway is restricted to travel by specialized maintenance vehicles which inspect and repair the guideway. Necessary maintenance operations are conducted on the guideway, central computer facilities, communications equipment, stations and wayside power stations as required by either on-condition monitoring or routine scheduled maintenance. Note that this maintenance schedule imposes a high degree of modularity on the design and construction of the guideway . By its very nature, a guided rail system has many single points of failure along the track. However, a rigorous program of preventive maintenance conducted with frequent and thorough monitoring and the enforcement of conservative criteria for replacement upon a modular framework will preclude the necessity of shutting down a guideway for a major repair.

During normal operation, the vehicle travels on its guideway using Maglev technology for both levitation and propulsion. BIT maintenance operations are conducted in the background and any detected faults are logged in non-volatile mass memory. The average duration of vehicle travel is two hours and twenty minutes, including station stops and post-mission maintenance, for a total of twenty-two hours daily. There are an average of three off-line station stops per mission. Station stops last an average of three minutes, during which time passengers board and leave the train, baggage and freight is loaded and unloaded, canteen and sanitary supplies are processed, and additional BIT maintenance is performed. An additional service at the end of a mission requires ten minutes. It is assumed that the vehicle has sufficient redundancy in all of its systems to allow it to be dispatched with faults, should those faults occur during normal operation. Of course, there is some minimum complement of components which must be functioning to allow the system to maintain the required level of reliability for a given mission. This is called the minimum dispatch complement (MDC). If faults accumulate such that the MDC is not functioning, the vehicle is not allowed to continue operation as this would expose the passengers and crew to an unacceptable risk of injury. At this point the vehicle would operate in a degraded mode, so as to be able to arrive at the next station for repairs. Redundancy which exceeds the system requirements for fault tolerant operation increases the availability of the system by allowing dispatch in the presence of failed components. Furthermore, additional redundancy also enhances the maintainability of the system by deferring repairs until they are normally scheduled to take place in a maintenance facility.

An example of a maintenance plan for a high speed ground transportation system is provided in [11]. The plan includes: (1) continuous performance monitoring of critical components, notably the braking system, via the train diagnostic and reporting system, (2) inspection of all vital safety functions every time the train arrives at a principal station, and (3) a regular schedule of shop visits where more extensive inspections take place. This aggressive maintenance plan, as well as sufficient sparing of vital components to allow dispatch in the presence of some failed parts[0], has contributed to the high availability of this line. In 1988, failures causing delays of over ten minutes occurred only 3.2 times per million train-km.

"When a degraded mode of operation cannot be maintained safely and the vehicle stops on the guideway, push recovery procedures will be put into effect which will allow the following vehicle to be brought up to the failed vehicle, coupled, run a final series of checks, push the failed vehicle onto the next station to drop off passengers of both vehicles, and continue onto maintenance. Before push recovery is allowed, central control, in conjunction with the zone controller, will interrogate the vehicle to determine if push recovery procedures can be put into effect. If not, a maintenance vehicle will be dispatched. Central control will also dispatch two empty vehicles from maintenance to move the delayed passengers to their final destination." [13]

During extensive depot maintenance, i.e. phase 5 of the mission scenario, the vehicle is powered down and is completely off-line, resting on its wheels or skids. The operational environment for this phase corresponds to Ground, Fixed, i.e. less than ideal conditions (ideal being a laboratory environment), such as "installation in permanent racks with adequate cooling" [8], etc. During the other phases of operation when the vehicle is hovering or traveling along the guideway, the operational environment is assumed to correspond to Airborne, Inhabited, i.e. an environment "without ... extremes of pressure, temperature, shock, and vibration on long mission aircraft" [8].

[0] For example, speed reduction tables call for lower maximum operating speeds for various combinations of brake failures. However, the combined failure of one passenger car truck's friction brake and one power car truck's rheostatic braking system does not require any speed reduction. Furthermore, a run can be initiated with this level of inoperative brakes [11].

2.2. Route Selection

Maglev transportation routes are being considered for several different corridors in the United States [1]. Since "it is extremely important that the first Maglev system be introduced in a high density corridor" [3], this report will use a generic, high density route for the Maglev mission scenario. The hypothetical route connects two major U.S. cities, located 800 kilometers (500 miles) apart. Since Maglev systems are most likely to evolve on a regional basis [9], this hypothetical route is intended to model one section of a typical regional network.

Off-line stations are spaced at 100 kilometer intervals. A typical run makes three stops, including the stop at the end of the line. However, during peak travel periods, direct origin to destination service is provided between all station pairs, with the scheduled frequency of service dependent on demand. The off-line stops take three minutes and the end-of-the-line stop takes ten minutes to allow for some additional routine service of the vehicle. Passengers will be assigned reserved seats and will exit from one platform and board from another. The capacity of each vehicle is forty passengers, with a luggage capacity similar to that of the airlines, i.e. each passenger is allowed one carry-on bag and two suitcases which are transported as freight. These vehicles can be grouped as two or three car trains. The line speed between stations is 500 km/hour [13]. This results in an average speed of 110 m/sec [13]. They will be dispatched so as to allow a volume of 4,000 to 12,000 passengers per hour to travel between the two cities. This means that the headway for each vehicle is approximately 36 seconds or 4 kilometers at 400 km/hour as shown by the following calculations.

$$\frac{1 \text{ vehicle}}{40 \text{ passengers}} * \frac{4000 \text{ passengers}}{\text{hour}} = 100 \text{ vehicles/hour}$$

$$\frac{100 \text{ vehicles}}{\text{hour}} * \frac{1 \text{ hour}}{3600 \text{ seconds}} = 1 \text{ vehicle/36 seconds}$$

$$\frac{400 \text{ km}}{\text{hour}} * \frac{1 \text{ hour}}{3600 \text{ seconds}} = 1 \text{ km/ 9 seconds}$$

$$\frac{1 \text{ km}}{9 \text{ seconds}} * \frac{36 \text{ seconds}}{\text{vehicle}} = 4 \text{ km/vehicle (2.4 miles)}$$

Some attention must be paid to the tradeoffs between on-line and off-line stations. On-line stations do not require guideway switching for their use. However, a train stopped at an on-line station blocks traffic for the duration of its stop. Off-line stations offer significant advantages in terms of capacity and flexible schedules [3]. They allow origin-to-destination service without degradation in performance. An off-line station is essentially a section of guideway built parallel to the main line. A specialized section of the guideway, called the switch, allows a vehicle to be diverted to either the main line or the off-line sections. Navigation of the switch may be accomplished in two ways, referred to as active or passive switching. In so-called passive switching, the guideway simply forms a Y-shape and the vehicle negotiates a route down one fork or the other. In an active switch, a flexible section of guideway is moved from one position to the other, thereby altering the route the vehicle follows. As a fail-safe measure, passive switching may be assisted by magnetic forces from the guideway. For the U.S. Maglev Transportation System, off-line stations will be used.

It is interesting to compare the intercity routes planned for the Transrapid Maglev System, currently being readied for construction in the Federal Republic of Germany with the model used in this concept of operations for the U.S. Maglev Transportation System. The German system will use on-line stations. Trains comprise five vehicles, with each vehicle carrying 98 passengers. The minimum headway is 10 kilometers. Use of off-line stations in the U.S. and on-line stations in Germany is based on the unique demographics of each country. The German population is clustered in large cities. The demand for high-speed travel is between these large, widely spaced urban centers, with little demand for intermediate stops. The U.S. population is more uniformly distributed along long strips which radiate outward from urban centers, forming corridors. Although these urban centers remain frequent *destinations* for travelers, especially business travellers, the starting points for most users of U.S. Maglev trains are not clustered in central locations. Therefore, the demand for intermediate stations is significantly greater. These different demographics result in different travel patterns. To be economically viable, the U.S. Maglev system must serve the needs of its traveling public. This requires off-line stations and smaller capacity trains.

For Transrapid trains traveling between on-line stations, switching is an infrequent activity, used primarily to allow trains to be removed from the guideway. Vehicles must also slow down to negotiate the switch. The setup time for the switch is very long and the wear and tear on these high precision mechanical parts is very low. The Transrapid design uses a movable, single track mechanical switch. To allow a vehicle to reach an off-line station, a movable section of guideway would be temporarily switched over to the parallel section, thereby guiding the vehicle to a section of the track parallel to the main line. Switching of the guideway in the Transrapid system can require up to several minutes to effect, when all factors affecting this time are considered. These include the actual time of positioning the switch for the vehicle leaving the main guideway and then repositioning the switch for a through vehicle, the speed reduction required to safely traverse the switch, and the extra headway required between vehicles to allow enough time to stop safely in the event of a mechanical switch failure. Proper alignment of the switch is necessary to ensure safe passage of the vehicle. Since the guideway is very massive, the switching operation must be done slowly and carefully.

For the U.S. system, trains will be traveling between off-line stations. For a movable switch, the setup time must be very short, much less than the worst case headway of 36 seconds, and the wear and tear on the switch would be very great. Thus, for the U.S. system, route selection is accomplished by means of a passive, i.e. non-movable, high speed switch mechanism [10]. In this case, active redundant propulsion coils would be located in the guideway along both sections of the switch. The coil which lies along the intended line of travel would be activated, directing the vehicle in the correct direction. The benefits of a passive switch for safety, availability, and system capacity warrant the engineering effort required to design such a mechanism.

For the U.S. market, off-line stations and smaller capacity trains are a necessity. Since this concept of operations results in shorter headways and frequent switching, a safety conflict immediately arises. This conflict must be resolved by the designers of the U.S. Maglev system. The present report will address these issues for the Maglev control computer system.

2.3. Additional Route Specific Information

To complete the mission scenario, it is necessary to define various other parameters for the hypothetical route described in Section 2.1. These parameters are the frequency of departures, the number of cars per train, the comfort and ride motion environment, and the operating temperature range.

To accommodate the typical commuting patterns of the business traveller, Monday through Friday departures include two peak periods of travel. During the morning rush hours, i.e. from 7:00 to 10:00 A.M., and the evening rush hours, i.e. 3:00 to 7:00 P.M., trains comprise three Maglev vehicles (i.e. articulated sections) , with a capacity of 120 passengers per train, departing every 36 seconds so as to be able to carry 12,000 passengers per hour per direction. During non-peak commuting hours, the number of departures remains constant, however trains consist of one or two Maglev vehicles, so as to be able to carry 4,000 to 8,000 passengers per hour. Exclusive passenger service is maintained only during peak periods; freight service is interspersed with less frequent passenger car departures during other operating hours. From 10:00 P.M. to 2:30 A.M. and from 4:30 A.M. to 7:00 A.M. the frequency of departures drops to one train every 72 seconds. The shutdown period for maintenance of the guideway and wayside facilities is from 2:40 to 4:30 A.M. While these figures may seem high, they are intended to represent the maximum capacity, utilization and stress of the system. Experience has shown that other successful transportation systems, e.g. airline service and interstate highways, were designed for maximum capacities which greatly underestimated the demand and eventual usage of these modes of travel. However, here our objective is to examine the upper boundary of dependability.

The operating temperature range is from -40°F to 120°F (-40°C to 49°C).

The level of ride comfort for normal, non-emergency operations will meet the two hour reduced comfort limits stipulated by ISO-2631.

It is also necessary to define trip specific safety parameters for this hypothetical route. These parameters are maximum speed, number and distance between wayside zone controllers and safe stopping areas, and weather constraints.

The maximum vehicle speed is 500 km/hour.

Between any two stations there are approximately forty wayside zone controllers, distributed so as to meet the requirement that every train be able to transmit 40 GHz radio data to at least two wayside stations at all times. Since this frequency requires line-of-sight contact between the transmitter and the receiver, repeaters will be positioned wherever necessary. The distance between these wayside zones is 2 kilometers. The site of each wayside zone controller also serves as a safe stopping area between stations.

For emergency situations, a deceleration range is allowed, depending on the severity of the situation. When conditions allow, a deceleration of not more than 0.25 g will be used. For extreme situations, for example, when necessary to decelerate to prevent or mitigate a sudden stop, decelerations of up to 1 g are allowed. Furthermore, the vehicle bodies are designed to absorb the shock of an impact by controlled buckling and thereby to protect passengers and crew from extreme deceleration forces. Furthermore, while the vehicle is in motion, passengers remain seated, except when necessary to use rest facilities. They also wear seat belts designed for both safety and comfort. Each seat is also equipped with an airbag. Airbags are especially effective in this uni-directional mode of travel where all sudden stops are in the direction of motion, i.e. head-on. Each seat is also provided with a video color monitor, personal computer, and keyboard for work or relaxation, an individual telephone hookup. By keeping passengers seated and seat-belted, acceptable ranges of acceleration, deceleration, curve radius, curve speed, tilt, line speed, and safe headway can be extended to increase capacity, and reduce trip times without compromising safety.

Table 2-1 specifies the allowable degraded performance of a Maglev vehicle for a range of weather conditions.

Event	Normal Operation (V_{max} = 500 km/hr)	Degraded Operation (percentage of V_{max})			
		100%	67%	33%	0%(No Op)
Wind:Maximum Sustained Speed	≤ 45 mph	45 mph	55 mph	65 mph	>75 mph
Rainfall	≤ 2 inches/hour	2 in/hr	3 in/hr	4 in/hr	>5 in/hr
Snowfall	≤ 12 inches/hour	12 in/hr	28 in/hr	44 in/hr	>60 in/hr
Ice/Sleet	≤ 1 inch/hour	1 in/hr	2.3 in/hr	3.7 in/hr	>5 in/hr
Hail (diameter)	≤ 0.125 inches	0.125 in	0.25 in	0.375 in	>0.5 in

Table 2-1. Environmental Constraints.

2.4. Safety Issues Related to Automatic Controls

The Federal Railroad Safety Act of 1970, as amended, Section 202(e), gives the Federal Railroad Administration (FRA) jurisdiction over "all forms of non-highway ground transportation that run on rails or electromagnetic guideways, including ... any high speed ground transportation systems that connect metropolitan areas" [7]. However, FRA regulations and guidelines have evolved over the years for a railroad system composed largely of relatively slow (100 km/hr) vehicles for which automatic controls were not required. However, the high speed of Maglev trains (500 km/hr) requires automatic control of all Maglev vehicles. Indeed, the draft Maglev system parameters, prepared by a Federal inter-agency working group in July 1990, state as a minimum requirement that "All controls must be fully automated and fail-safe" [3]. Part 236 of the code of Federal Regulations, Transportation 49, deals with "Rules, Standards, and Instructions Governing Installation, Inspection, Maintenance, and Repair of Signal and Train Control Systems, Devices, and Appliances." A major area not covered by Part 236 is the software and hardware design and operational regulations for microprocessors utilized in vital sections of the signal and control system. For Maglev, the microprocessors involved with levitation control for both moving and hovering vehicles (in the case of EMS), speed control, route integrity, braking, train location, train scheduling and coordination, and data communication are vital to the safe operation of the system. The final report for the Verification Methodology for Maglev Control Computer Systems in part will address this uncovered area of regulation by demonstrating design standards and a verification methodology, and by specifying fault tolerance and redundancy management requirements, and testing and maintenance requirements for reliable real-time control systems employing microprocessors.

3. Functional Requirements

In order to determine the detailed computational requirements of the complete Maglev application, a set of functional requirements must be specified. Computational requirements include throughput, memory, processing lag, function iteration rate, order dependencies among functions, and I/O and inter-function communication rates. To fully specify the computer architecture for Maglev, additional requirements such as weight, power and volume constraints, and operational environment must also be specified.

3.1. ALS Functional Requirements: An Example

An example of the functional requirements of a transportation application is provided by the Advanced Launch System (ALS), discussed in Section 2, which is being designed for NASA and the Department of Defense [4]. The purpose of ALS is to launch heavy payloads into low earth orbit at one tenth the cost of current launch vehicles. The functional requirements for the ALS application are shown in Figure 3-1 as a three tiered hierarchy, with functions at the third level assumed to be the equivalent of executable, independently dispatchable tasks. Nine top-level ALS functions have been identified. The requirements of each top-level function are computed by summing the numerical requirements of each of its lower level constituent functions. The following numerical data are required for each dispatchable task:

(1) Frame rate
(2) Throughput (or instructions per execution)
(3) Throughput margin
(4) Processing lag
(5) Scheduling requirements (e.g. preemptible or nonpreemptible)
(6) Task execution order dependencies
(7) Inter-function communication requirements (bits per iteration, latency)

Given this data, it is possible to construct a distributed schedule for the task suite and quantitatively perform system sizing and determine performance parameters.

These requirements were then used to specify a candidate computer architecture which could be modeled and analyzed to determine whether or not it met ALS requirements for reliability, availability, performance, etc. The candidate architecture was assembled from the hardware and software building blocks of the Advanced Information Processing System (AIPS) which has been designed to provide specific fault tolerant attributes to an embedded control computer system. These include Byzantine resilient fault tolerance, a simplex programming model, and rigorous separation of redundancy management and application software. Each of these attributes simplifies the validation of the final system.

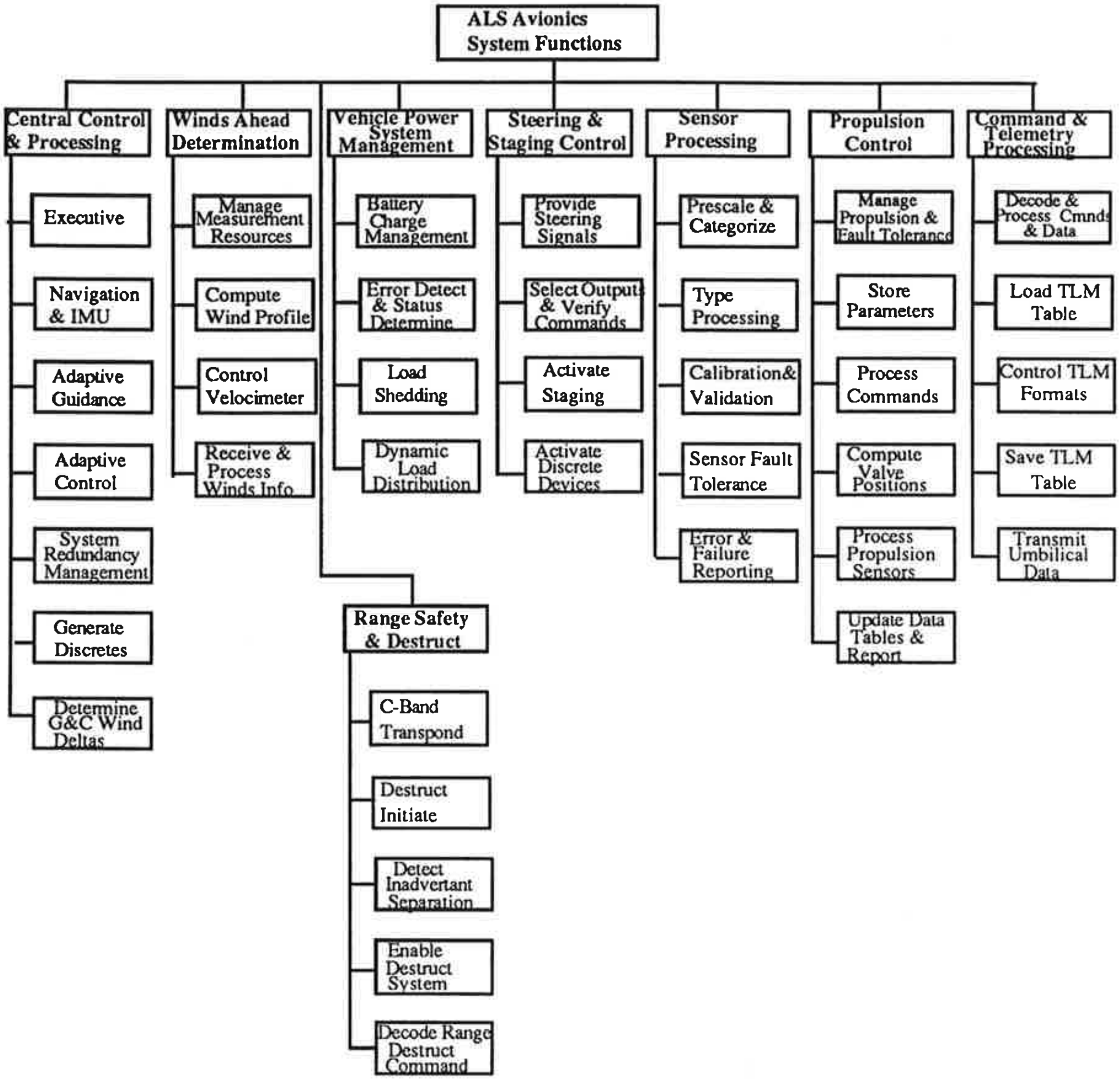

ALS Avionics System Functions
Central Control & Processing
Executive
Navigation & IMU
Adaptive Guidance
Adaptive Control
System Redundancy Management
Generate Discretes
Determine G&C Wind Deltas
Winds Ahead Determination
Manage Measurement Resources
Compute Wind Profile
Control Velocimeter
Receive & Process Winds Info
Vehicle Power System Management
Battery Charge Management
Error Detect & Status Determine
Load Shedding
Dynamic Load Distribution
Range Safety & Destruct
C-Band Transpond
Destruct Initiate
Detect Inadvertant Separation
Enable Destruct System
Decode Range Destruct Command
Steering & Staging Control
Provide Steering Signals
Select Outputs & Verify Commands
Activate Staging
Activate Discrete Devices
Sensor Processing
Prescale & Categorize
Type Processing
Calibration& Validation
Sensor Fault Tolerance
Error & Failure Reporting
Propulsion Control
Manage Propulsion & Fault Tolerance
Store Parameters
Process Commands
Compute Valve Positions
Process Propulsion Sensors
Update Data Tables & Report
Command & Telemetry Processing
Decode & Process Cmnds & Data
Load TLM Table
Control TLM Formats
Save TLM Table
Transmit Umbilical Data

Figure 3-1. ALS Avionics System Functions

A similar set of requirements must be specified for Maglev. A preliminary set of functional requirements for the U.S. Maglev Transportation System control computer, based on the functions specified for the Transrapid system and information presented at the Maglev Technology Assessment Symposium[1], is shown in Figure 3-2. This list of functions was extracted from a descriptive, high level list of functions performed by the Transrapid computer system [2], a somewhat more detailed and comprehensive list of functions included in a report whose main purpose is to discuss the safety issues of Maglev

[1] Maglev Technology Assessment Symposium, Hosted by the John A. Volpe Transportation Systems Center, Sponsored by the U.S. Department of Transportation, the US Army Corps of Engineers, and the U.S. Department of Energy, September 26 -27, 1991, Cambridge, MA.

transportation [1], and material from the proceedings of the Maglev Technology Assessment Symposium. Section 3.2 provides a significant amount of additional detail on these functions. Wherever these documents lacked detail or clarity, qualified DOT personnel provided the necessary information to fully specify a given function.

Each function specified in Section 3.2 is assigned a level of criticality. The three possible levels are safety-critical, mission-critical, and non-critical. A safety-critical function is one whose loss results in severe injury or in loss of life. In other words, the loss of a safety-critical function can lead to a failed-unsafe state in the mission scenario state diagram (Figure 2-2). The loss of a safety-critical function is analogous to either a catastrophic or critical event in the "Undesired Event Severity Categories" list [1]. Information about safety-critical functions is used to model system reliability and safety. An example of a safety-critical function is the speed control of Maglev vehicles. A mission-critical function is one whose loss disrupts normal train service and results in a failed-safe state in the mission scenario state diagram. Information about mission-critical functions is used to model system availability and maintainability. An example of a mission-critical function is data transmission between the vehicle and the wayside. A non-critical function is one whose loss does not impact on the welfare of passengers or employees and does not disrupt train service. An example of a non-critical function would be the automatic control of onboard air conditioning[2] or loss of a lighting lamp.

3.2. Maglev Control Computer System: Survey of Automated Functions

This section presents a survey of functions that have been automated in the Transrapid TR07, the French TGV and the German ICE high speed trains, and/or are contained in the draft U.S. Maglev requirements. These are <u>not</u> the functions used to synthesize the control computer architecture in the present study. Section 5 lists the specific functions whose quantitative computational requirements were used as the basis for the architectural alternatives for the U.S. maglev system. It should also be noted that the functions described here have been allocated to a computer on-board the vehicle, a wayside station or a central computer based on a certain underlying distributed computer architecture of the actual systems from which these functions have been derived. This does not imply the same computer architecture for the U. S. maglev system. As a matter of fact, the functional analysis reported in Section 5 is a pure description of the functions needed in the U.S. maglev system without making any assumptions regarding how the functions will actually

[2] This must be qualified by the recognition that the emergency response and evacuation plans may require that the air conditioning system be shut down in the event of a fire to avoid distributing smoke throughout the vehicle. In this case, the ability to disable the air conditioning system would be considered safety critical.

be implemented. Then, a maglev computer architecture is synthesized which can meet the performance and dependability requirements of these functions.

The Transrapid model TR07 has full automatic control [2] which is spatially distributed at three hierarchical levels: onboard vehicle, decentralized wayside, and Central Control Facility (CCF). Data is acquired, transmitted, and processed at all three levels. The Operational Control System (OCS) for the U.S. Maglev Transportation System is based on this model [1]. There are six broad, high level functions which the OCS must perform: protection, control, supervision, data transmission, passenger information, and peripheral systems. Information about the first four categories was extracted from GFI documents [1] and [2]. This information forms the basis of the OCS for the U.S. Maglev Transportation System presented in this Concept of Operations document. The nature of the functions dealing with passenger information and peripheral systems is not yet specified for the U.S. Maglev system and therefore will not be discussed here. Figure 3-2 shows the three OCS hierarchical components.

The automatic, i.e. computer directed, operation of Transrapid is called Automatic Train Control (ATC) [1]. It is intended to be fully automated in the performance of its two basic functions, (1) route integrity and (2) safe speed enforcement. Route integrity is defined as maintaining a safe, unobstructed path for train travel. Safe speed enforcement is defined as keeping the speed of a train within designated operating specifications. In a similar manner the ATC for the U.S. Maglev Transportation System is defined as the functions and installations whose purpose is the safety, control, and supervision of vehicle operations, as well as intercommunication between them.

3.2.1. Onboard Vehicle Functions

The functions performed by the onboard computer are vehicle location, protection, and control, environmental and equipment monitoring, and fail-safe data and voice transmission with the wayside. Although the position of the vehicle is supervised by the central control system, and its speed controlled by the wayside zone controller, the onboard vehicle control system has the functionality and reliability to assure safe operation without input from the central control. Furthermore, it is solely responsible for the level of ride comfort experienced by the passengers and, in the case of Transrapid which uses an EMS suspension system, the onboard system controls the levitation of the vehicle. Under no-fault conditions, the onboard computer collects information about the state of the vehicle, e.g. its speed and location, the status of various onboard systems, e.g. the status of the position sensors used to measure the gap between the guideway and the levitation magnets, and local environmental conditions, e.g. the wind speed, and transmits this information, via the wayside control stations to the central control center.

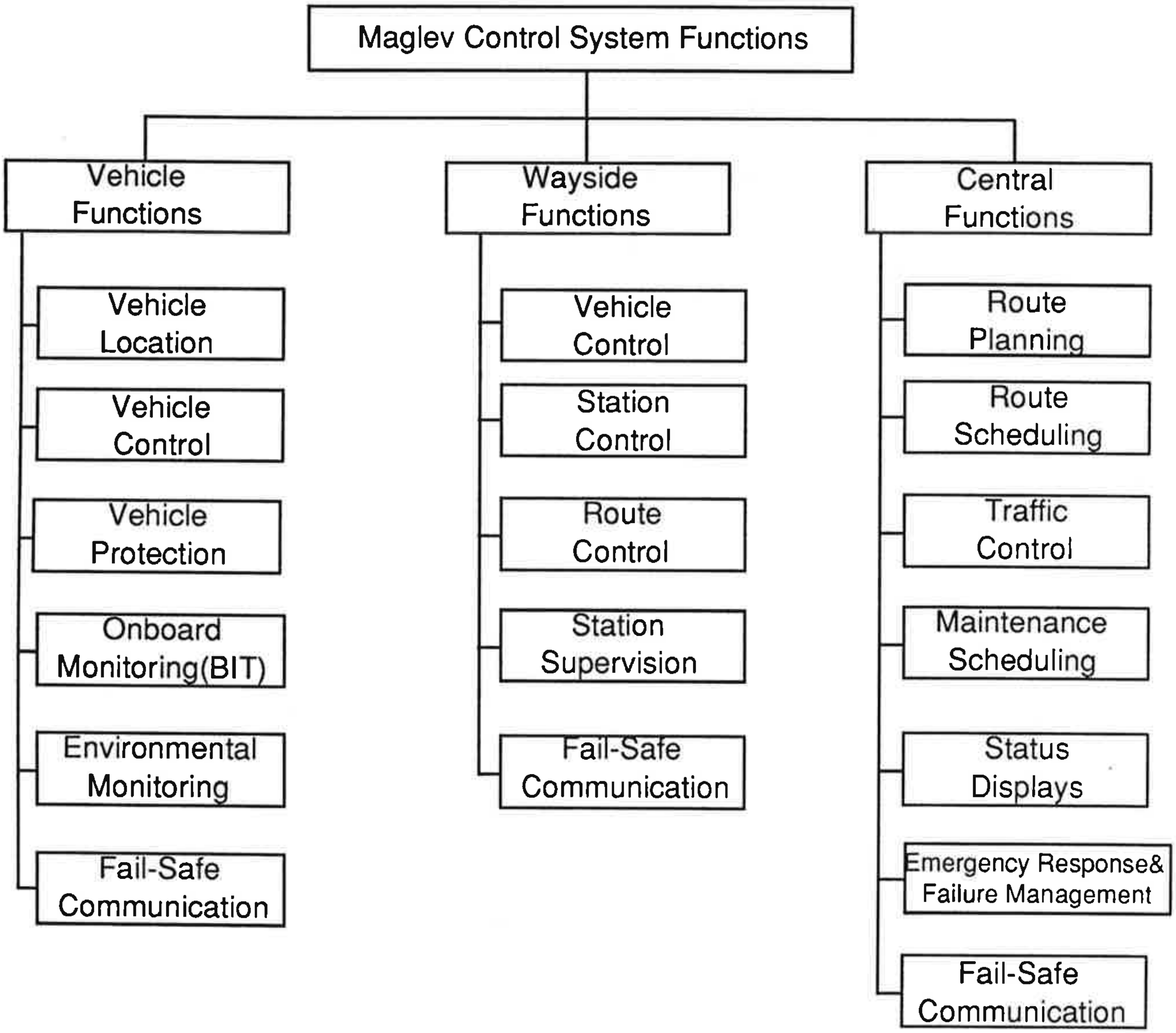


Figure 3-2. Maglev Control System Functions

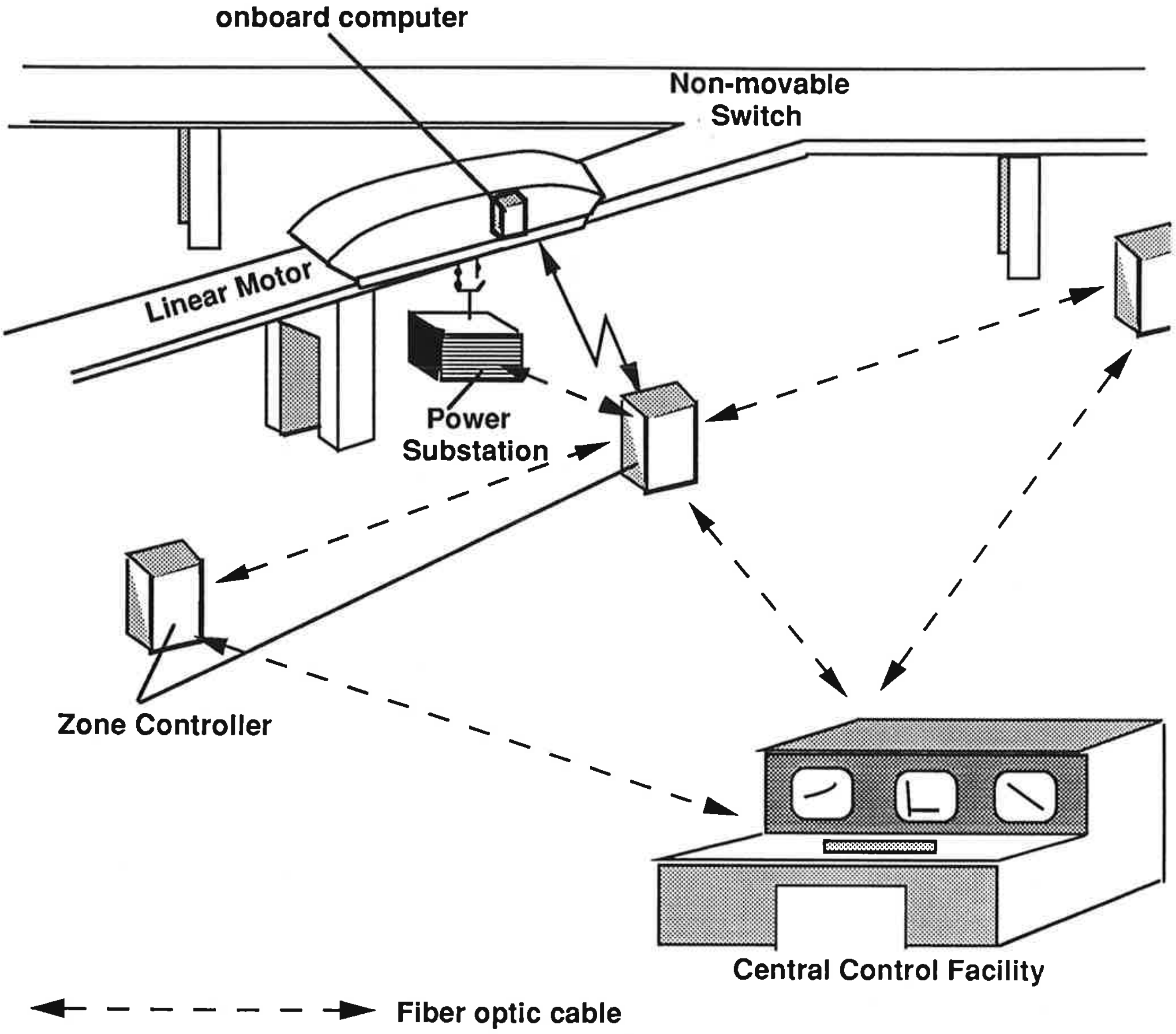


Figure 3-3. Maglev Operational Control System

A multiple articulated train may comprise several cars, each of which is a fully autonomous vehicle in its own right. However, when several independent vehicles are grouped together to form a train, the cars coordinate their control efforts by carrying out a protocol which selects one car to perform the data transmission to the wayside. The other cars now transmit their vehicle-specific data to the primary car for retransmission to the wayside. Each vehicle, however, continues to listen to transmissions *from* the wayside. The vehicles in a train communicate over a network which forms part of the linkage joining the cars together. Each vehicle is assigned a unique, system-wide serial number which identifies the vehicle, each train is also assigned a unique identifier which refers to the train as a logical entity and which is associated with the route and trip profile which this train is to follow. In the case of a multi-vehicle train, each car retains its vehicle serial number but shares the train identifier with the other cars in the train.

3.2.1.1. Vehicle Location (Safety-Critical)

The vehicle location function determines the vehicle position, travel direction, speed, acceleration, and braking capability. These quantities constitute the location state vector of the vehicle. As in the Transrapid system, the onboard computer uses an incremental vehicle location system to determine the absolute position of a train. A passive loop code in the guideway is scanned by four sensors mounted on each vehicle, one pair on each side of the vehicle. Position tags are located on the guideway, about 200 meters apart. The position tags are mounted in pairs, with one member of each pair on opposite sides of the guideway. These tags are mapped to ordered entries in a computer table which yield the "raw" position of the train. More resolution is obtained by using the same sensors to count the stator pack grooves. The vehicle location is considered valid when two of the four readers agree on the value of the tag. The exact position of the vehicle is then calculated by counting stator pack grooves and adding the cumulative length of the stator packs to the last valid tag reading. If the readings disagree, the absolute position is based on the last valid reading and the cumulative stator pack count. By comparing the relative position of two valid consecutive tag readings in the table, the direction of travel is deduced.

Alternatively, vehicles can be equipped with satellite receivers that process signals transmitted from GPS (Global Positioning System) satellites. The GPS, which includes approximately two dozen satellites in low earth orbit, is a U.S. Government sponsored navigation system, which allows a system with appropriate receivers to determine very accurately its geographical location and velocity.

Periodically, the average velocity of the vehicle is calculated by dividing the distance traveled during the period by the exact time elapsed during the period. The average acceleration of the vehicle is calculated by dividing the difference between the initial and final velocities of the vehicle during the period by the time elapsed. These calculations are performed at a frequency of 10 Hz, i.e. every 100 ms, during which time the vehicle may have traveled at most 14 meters (assuming a 500 km/hr maximum speed).

The distance required to stop the vehicle with only aerodynamic braking is a function of its velocity, direction of travel, and the current wind velocity. The capability of the secondary braking system is a function of the velocity of the vehicle, the aerodynamic braking, the eddy current brake and the grade. Finally, the distance needed to stop the vehicle using the primary braking system is a function of the velocity of the vehicle, its acceleration, the aerodynamic braking, grade and the magnetic deceleration force applied by the LSM. These capabilities are calculated every 100 ms.

The vehicle location state vector is transmitted to the wayside every 100 ms.

3.2.1.2. Vehicle Control (Safety-Critical)

The vehicle control function directs the operation of the primary, in the case of EMS, and secondary suspension system and processes status and error messages from the onboard monitoring function. The latter reports the state of the sensors and actuators which control the primary suspension of the vehicle.

3.2.1.2.1. Levitation or Suspension (Safety-Critical)

If levitation of the vehicle is produced by an EMS system, the gap between the guideway and the levitation magnets will require active control. As discussed in Section 1.2, EDS systems do not require active control of the gap between the vehicle and the guideway. However, with EDS systems, the stiffness of the primary suspension may not produce a satisfactory ride quality without actively controlled secondary suspension. Both active controls are demanding applications. However, the control of the secondary suspension for EDS is probably the more demanding of the two, in terms of required throughput and communication bandwidth. The figures given here are based on the EMS system used by Transrapid.

The position of each magnet (i.e. the gap between the magnet and the guideway) is controlled by a standard feedback control loop. There are 30 support magnets and 24 guidance magnets per two section vehicle. The magnets are strung together to form a chain, with adjacent ends arranged to pivot with respect to each other in a hinge-like manner. The field strength on the magnets is actively controlled to maintain an 8 mm gap between the magnet and the guideway. Inputs to this control law are the actual gap size, the nominal gap size, the velocity of the train (obtained from the vehicle location state vector), and the acceleration of the train. The actual gap size is determined by reading and comparing values from two gap sensors. An error is defined as a difference in the two sensor readings greater than 1.5 mm. This error is a normal occurrence when a gap reading is taken over thermal expansion joints. In this case, the lower sensor value is chosen. Otherwise, the average of the sensor values is used. The sensors are attached to the ends of two adjacent magnets, one on each end. The acceleration is determined by a single accelerometer. The current which is to be applied to the magnets to maintain the proper gap is determined by the control law.

The control law is executed by individual digital controllers, each of which obtains gap readings from two sensors. Furthermore, each magnet can be completely controlled by either one of the two controllers associated with it. Sensors and current generators are checked and calibrated on a regular basis by the onboard monitoring function during normal routine maintenance operations at every station stop. If an EDS system is in place, the temperature of the super-conducting magnets is also monitored periodically by the onboard monitoring function.

3.2.1.2.1.1. Safe Hovering (Safety-Critical)

Only EMS can levitate a stationary vehicle, hence this function applies only to EMS systems. The vehicle maintains its own suspension until stopped either by its own internal control or by central control.

3.2.1.2.1.2. Secondary Suspension (Safety-Critical)

The secondary suspension provides an additional level of isolation between the coach body and the guideway. When the velocity of the vehicle is sufficiently high, aerodynamic control may be used to provide a better ride quality. In the EDS design proposed by the Bechtel Consortium, this is accomplished by controlling a flap at the trailing edge of a vehicle to provide secondary suspension. The control algorithm for this application is similar to control algorithms for gust alleviation in an aircraft. For slower speeds, secondary suspension is provided by actively controlling springs which stabilize the roll motion and the vertical (pitch) and lateral (yaw) motion of the vehicle with respect to the magnetically levitated bogey.

3.2.1.2.2. Route Control (Safety-Critical)

For Transrapid, a route is selected by switching a movable section of guideway to one of two possible positions. The vehicle follows the route which is locked in place prior to its arrival. These movable switches, which take 20 seconds lock-to-lock, are probably not fast enough for the high capacity U.S. system. Instead, the vehicle is guided through the switch by other means which do not involve movement of a large section of the guideway. For example, magnetic forces may be used to effectively steer the vehicle through a switch. The route control function energizes the appropriate coils in the guideway which determine the direction of travel through a switch in the guideway. This function is performed principally by the wayside computer. There may be some communication between the on-board computer and the wayside computer to coordinate traversing of the switch such as reducing the vehicle speed, etc.

3.2.1.3. Vehicle Protection (Safety-Critical)

The vehicle protection function ensures that the vehicle behavior conforms to its pre-planned route profile, monitors onboard safety equipment, including the emergency braking subsystem, verifies that communication with two wayside stations is operational, collects information about safe stopping places, and takes corrective action in any emergency situation.

3.2.1.3.1. Route Profile Monitoring (Safety-Critical)

The route profile assigned to a train specifies the correct range of values of position, velocity and acceleration, and the correct direction of travel of the train for each 100 ms time interval for the duration of the trip. This profile is compared with the actual values stored in the location state vector. Deviations from the profile which continue for two consecutive intervals cause the appropriate emergency response to be triggered.

3.2.1.3.2. Emergency Velocity Control (Safety-Critical)

For Transrapid, the vehicle must be able to reach the next safe stopping location by using onboard power only. This means that sufficient energy must be available from 2 of 4 batteries and the linear generators (which operate only while train is moving) to control levitation, braking, and other loads before the vehicle is dispatched from a station. Two batteries can supply all loads (including air conditioning) for 7.5 minutes. Thus the onboard computer determines whether or not the batteries are fully charged, and then report their condition to the wayside control which provides the power to move the train to the next wayside station.

3.2.1.3.3. Emergency Stopping Control(Safety-Critical)

The vehicle must be able to stop itself at a safe stopping area without any input from central control. The location of the safe stopping points is stored in the onboard computer memory. If necessary, the vehicle uses power from onboard batteries to slow down and dock at a safe stopping point

3.2.1.3.4. Wayside Communication Monitoring (Safety-Critical)

The onboard computer sends data to the wayside stations every 100 ms. Similarly, it expects to receive a message every 100 ms from each of two wayside stations. If both wayside communications fail, i.e. if more than two consecutive messages from both wayside stations are not received, the vehicle protection function initiates emergency action and stops the train at the next safe stopping point, using the secondary brake. (Primary braking is controlled by the wayside zone controller. See section 3.2.2.2.2.) The central control facility will coordinate the rest of the rescue effort.

3.2.1.3.5. Secondary Brake Control (Safety-Critical)

Secondary braking is accomplished in two parts. The first uses eddy currents to slow the train to the point where it can safely land on its landing skids or wheels. Friction stops the train from that point on. The eddy currents are induced in the track guide rails by longitudinal vehicle magnets. Each vehicle has two eddy-current brakes, grouped in four autonomous units, separately powered from the four 440 VDC supplies of the onboard power network. Secondary braking is used to slow or stop the vehicle when the primary braking system does not reduce its speed in accordance with the route profile.

3.2.1.4. Environmental Monitoring (Safety-Critical)

Wind speed and temperature are monitored by sensors on the vehicle. These parameters are transmitted to the wayside and may result in the alteration of the route profile.

3.2.1.5. On-Board Systems Monitoring and Control(Safety-Critical)

In addition to the Built-In Tests (BIT) discussed above which are performed at station stops, onboard sensors record the level of the electromagnetic field in the coach. Whenever readings exceed allowed levels, the reading and time are logged for review. These events are also transmitted to the central control facility. In extreme cases, the vehicle may be taken off-line at the next station and a spare vehicle substituted. Finally, the vehicle monitors the track ahead and behind for as great a distance as possible to detect the presence of unexpected large objects, such as another vehicle, which may be on the guideway. When such an emergency situation is detected, the vehicle is brought to an abrupt stop with as little as a 2 second warning, decelerating by means of its secondary braking system at a rate which can cause unrestrained passengers to become dislodged from their seats. Prior to stopping, the wayside is notified of the impending emergency situation.

Furthermore, onboard systems such as onboard lighting, temperature, air flow, door position control, cryogenic systems in the case of an EDS suspension, etc. are controlled by this function.

3.2.1.6. Fail-Safe Data and Voice Transmission (Mission-Critical)

This is conducted by 40 GHz radio transmission using two independent transmitters and is received at all times by two wayside stations. Transmission is bi-directional. Data transmission is "critical" for normal system operations but not a vital, i.e. safety critical, link because the vehicle can always reach a safe stopping point without inputs from external sources.

3.2.2. Wayside Functions

The functions performed by the decentralized wayside computers are route control, vehicle control, station supervision and control, and communications. Between any two off-line passenger stations there are approximately forty wayside stations, distributed so as to meet the requirement that every train be able to transmit 40 GHz radio data to at least two wayside stations at all times. Since this frequency requires line-of-sight contact between the transmitter and the receiver, repeaters will be positioned wherever necessary. The distance between these wayside stations is 2 kilometers (1.2 miles). Each wayside station also serves as a safe stopping area between passenger stations. Each wayside station controls a section of the guideway called an acceleration zone. At most one train may be in a given acceleration zone at any given time. During peak travel periods, the headway between trains is 4 km. By spacing wayside stations every two kilometers, at least one section of unpowered guideway always separates two trains. Thus, to advance a train along the guideway requires a specific action on the part of the controlling wayside computer. If an emergency situation develops, the power can remain off, or a braking action can be initiated.

3.2.2.1. Route Control (Safety-Critical)

The route control function determines whether or not to apply propulsive power to a given section of guideway and determines the direction of travel at switch points along the guideway. It operates in conjunction with the CCF and onboard vehicle computers. The wayside computer is also responsible for controlling the coils in the guideway at each switch and coordinating vehicle traversal through the switch (see Section 3.2.1.2.2).

3.2.2.1.1. Vehicle Position Control (Safety-Critical)

The absolute position of the vehicle, as transmitted from a train to the wayside, is relayed to the CCF after being confirmed by sensors or other detection equipment in the guideway. When the wayside control computer is ready to send a train out of a passenger station, or to power a section of guideway to allow an approaching vehicle to advance, it requests authority to do so from CCF. After CCF has checked its database the route profile of the train and other train positions, it grants the wayside authority to apply propulsion to the cleared section of the route.

3.2.2.1.2. Route Selection (Safety-Critical)

The route selection function controls the direction of travel across a switch. The train may continue on-line or bear right to an off-line station. The route selection function confirms routes and acts in conjunction with the onboard vehicle route selection function to direct the train along the correct section of guideway. The onboard vehicle route selection function requests permission to travel along the fork in the switch which is called for by its route profile. It also indicates the set of steering magnets (left or right) which it wishes to energize. The wayside confirms permission for this route with the CCF, and after receiving an acknowledgment, signals the train to proceed to extend the selected guide wheels. It also enables the propulsion coils in the section of the switch which corresponds to the route selected. After the train signals that its sensors have confirmed proper engagement of the selected guide wheels and the wayside confirms, via sensor readings, the correct propulsion coils are powered, the wayside signals the train to proceed. After the train has passed through the switch, the wayside station signals the train to retract its guide wheels and simultaneously cuts power to the propulsion coils in the switch.

3.2.2.2. Vehicle Control

The vehicle control functional element calculates the desired value for longitudinal position, velocity, and acceleration, and compares the desired (and commanded) values to actual position and velocity, i.e., a normal feedback control loop application. Longitudinal speed control is executed by means of the long-stator, linear propulsion system which is arranged in blocks or sections at the wayside. Separate and alternative power feeding of the left and right long-stator drive sections are implemented to assure fail-safe vehicle control. One vehicle control unit is assigned to one power supply substation area. A vehicle can be fully braked to a standstill within a single substation section, in case the next section is occupied. Substation lengths are between 1.5 and 2.5 kilometers.

3.2.2.2.1. Velocity Control (Safety-Critical)

The wayside receives a predetermined speed profile from CCF for coordination of propulsion and braking of each train which travels along the section of guideway under the control of that wayside station.

The vehicle must attain a sufficient velocity before leaving a substation section to be able to coast to the next allowed safe stopping point. The algorithm is as follows. The acceleration and velocity are checked within each zone. If it is going fast enough, it is allowed to proceed. Otherwise it is stopped at the station or at an auxiliary stopping point provided for that purpose.

The vehicle must maintain a safe distance from adjacent trains. This is executed by the long-stator linear propulsion system. Long-stators are arranged in sections. There are self-consistent, i.e. so that two trains are not scheduled to be in the same place at the same time and a safe headway is preserved at all times between consecutive trains.

3.2.3.3. Maintenance Scheduling (Mission Critical)

Routine maintenance is scheduled for every vehicle, the guideway, power stations, passenger stations and wayside stations in accordance with the manufacturer's or the construction contractor's specifications. On-condition monitoring ensures that the minimum dispatch complement (MDC) of all system components is present before a train leaves a station. Exceptional cases requiring unscheduled maintenance of a vehicle result in the substitution of a spare vehicle in place of the vehicle which does not possess its MDC. To meet the high availability requirement of this system, adequate spare capacity is built in to all parts of the system. Thus, unscheduled maintenance is a very rare event. The maintenance schedule is managed by the central computer facility.

3.2.3.3.1. Route Integrity (Safety-Critical)

Sensors monitor, record and transmit to the central computer facility data about the integrity of the guideway and the propulsion and levitation coils. The guideway must remain properly aligned and free from debris, e.g. ice or litter, which could obstruct the route. Sensor data is processed by adjuncts to the wayside zone computers, condensed, and reported to the CCF. In some cases only exceptional conditions are reported. Exceptional conditions result in the scheduling of the appropriate corrective action.

3.2.3.3.2. Vehicle Integrity (Safety-Critical)

Sensors monitor, record and transmit to the central computer facility data about the integrity of the vehicle including the status of the onboard computer, the adequacy of the EMF shielding, the status of all onboard safety equipment, etc. Sensor data is processed by the onboard computer, condensed, and reported to the CCF. Exceptional conditions result in the scheduling of the appropriate corrective action.

3.2.3.4. Status Displays (Mission-Critical)

Traffic displays show traffic information in a manner "that is conducive to interactive discourse among the staff." These displays use high resolution graphics to present real time data about all aspects of the system. The information which is displayed includes a top level view of the relative positions of each vehicle in the system. This top level view also provides local weather data. The actual route traveled by each train is compared to its route profile by means of a velocity versus time plot. Monitoring information about the status of the guideway, switches, switching propulsion coils, wayside stations, passenger stations, and other specific items of equipment are also displayed. Whenever exceptional conditions are noticed by monitoring software, these conditions are brought to the attention of the operations personnel by means of audible alarms and flashing displays.

3.2.3.5. Traffic Control (Safety Critical)

The traffic control function coordinates the activities of every vehicle in the system. It tracks each train to verify that the train is correctly following its route and speed profile. It maintains safe speed enforcement. It controls the train position by means of switch and speed control. It conducts extensive communication with many parts of this highly distributed system.

3.2.3.5.1. Monitoring (Safety-Critical)

Information which CCF collects includes train positions, train speed, weather conditions, etc. from the vehicles and wayside stations. This information is used to modify and adjust the speed and position of trains as necessary to account for minor variances with the route and speed profiles being followed.

3.2.3.5.2. Supervision of Speed Control (Safety-Critical)

Safe speed enforcement is a primary responsibility of the central control computer. The daily timetable specifies a predetermined speed profile for each train. The CCF uses this information to track each train and verify the adherence of the train to its planned profile. It transmits pertinent sections of this profile to the wayside station in a timely fashion for coordination of propulsion and braking. The wayside computer receives information about a train prior to the arrival of that train in the acceleration zone controlled by the wayside. A series of handshakes and position, velocity, and acceleration confirmations takes places before a train is either propelled through the zone according to plan or slowed down or brought to a stop, depending on the global conditions of the system as seen by the CCF.

3.2.3.5.3. Supervision of Position Control (Safety-Critical)

Wayside stations request route authority from CCF before powering a train out of a station. After CCF has checked its database for the planned switching operation and other train positions, it grants the wayside authority to apply propulsion to the cleared section of the route. The CCF transmits continuous information about safe stopping points to the onboard computers of each train so that the onboard computer can stop at a safe stopping point on its own in the event of failed communications or other emergency situations.

3.2.3.6. Emergency Response and Failure Management (Safety Critical)

The emergency response function responds to emergency conditions by reducing the speed of trains, and if necessary applying emergency braking. Spare vehicles are put into service as necessary and alternate schedules are put into effect.

3.2.3.7. Fail-safe Data and Voice Transmission (Mission-Critical)

The central control facility communicates with the wayside stations over redundant fiber-optic connections.

4. RMAS

The first step in performing detailed RMAS modeling and analysis is to construct a mission scenario state diagram. An example mission scenario state diagram for the Advanced Launch System was presented in Section 2 (see Figure 2-1). The state diagram illustrates various phases in the mission scenario and also describes the events which trigger a transition from one state to another. For example, in order to transition from the pre-launch state to the boost state, the ALS vehicle must pass a Built-In-Test suite which determines whether or not the vehicle has the Minimum Dispatch Complement (MDC) of functioning components. In this example, the MDC does not require every component to be non-faulty, in order to provide better availability but not compromise reliability. The ALS computer architecture has sufficient redundancy to allow mission dispatch with some failed components. A strawman mission scenario for a maglev vehicle was presented in Figure 2-2.

The MDC for maglev vehicles would also allow dispatch with faults present. Since high "dispatch reliability," i.e. availability, high reliability, and absolute safety are maglev system requirements, the control computer architecture will need to have sufficient redundancy to allow dispatch with faults. (Note that qualitative measures of these attributes, such as "high" and "absolute" are not adequate for assessing the adequacy of a given architecture.) The analysis of proposed architectures will determine the actual values of these system attributes and allow the selection of an architecture which meets its requirements.

A significant difference exists between the probabilities which are allowed for arriving at the failed-unsafe state, i.e. for a catastrophic failure, for different applications. For example, the reliability requirement for a commercial transport fly-by-wire system such as the Airbus A-320, stated in terms of the acceptable probability of system failure, is 10^{-10} per flight hour. For military aircraft, the probability of vehicle loss is usually 10^{-7} per hour (presumably because the crew can bail out). The reliability requirement, stated as a probability of failure, for the system control software, and presumably for the hardware upon which it executes, for commercial maglev transportation, as specified in the draft Maglev System Parameters [3], is 10^{-9}.

When fly-by-wire systems experience a catastrophic failure of their control computer, the result is vehicle loss. There is no fail-safe state unless the aircraft is provided with some sort of backup system, which may be an analog computer capable of controlling the flight actuation surfaces to the extent that a safe landing is possible. Mechanical backups do not provide the performance needed to control statically unstable aircraft.

While it is possible to reduce the likelihood of a catastrophic failure of the maglev control computer to an acceptable level, the probability of the occurrence of a common mode fault is not zero. Therefore, a failed-safe state must be provided such that the loss of the control computer does not result in vehicle loss or loss of human life. This can be accomplished in a number of ways and may require the presence of several backup features. For example, the operator may be able to detect the failure of the control computer by observing inconsistent data on his system status display. A mechanical speed governor on the vehicle could prevent excessive vehicle velocities. The wayside computers and central computers could also be used to detect the loss of the control computer on a given vehicle and be able to exert other safety measures, such as remote control operation of the vehicle or control of the guideway. Some combination of these, and other to-be-defined, measures will be necessary to provide proper fail-safe operation of the maglev transportation system.

Dependability is the quality of service that a particular system provides. Reliability, availability, safety, and maintainability, are measures used to quantify the dependability of a system. Precise definitions exist for these metrics. These definitions are included here because subtle nuances sometimes interfere with their accurate usage in discussing various aspects of fault tolerant systems. Although each metric is in theory quantifiable, arriving at a precise value is sometimes very difficult. Hence, these terms also are in wide use as qualitative descriptors of the dependability of a system. However, for the maglev verification methodology, precise numerical values must be assigned to each dependability metric. Fault tolerant design can be used to improve these measurements of dependability for a given system.

4.1. Reliability

The reliability $R(t)$ is a function of time, defined as the conditional probability that a system will perform correctly throughout the interval $[t_0,t]$, given that the system was performing correctly at time t_0. In other words, the reliability is the probability that a system will operate correctly throughout a complete interval of time [5]. The unreliability of a system is the probability that a system will operate incorrectly during a given interval. Hence, the unreliability is often referred to as the probability of failure. Reliability is used to characterize systems in which even momentary periods of incorrect performance are unacceptable, or in which repair is impossible. With space applications, the time intervals may range as long as ten years, while flight control applications may only last several hours but have reliability requirements in excess of 0.999999999, i.e. a probability of failure that is less than 10^{-9} per hour. To achieve these levels of reliability requires a combination of both fault avoidance and fault tolerance techniques. Fault tolerance is used to instantaneously mask the rare faults that do occur.

4.2. Availability

Availability *A(t)* is a function of time, defined as the probability that a system is operating correctly at the instant of time *t*. Availability differs from reliability in that the latter is taken over an interval of time, while the former is taken at an instant of time [5]. Thus a system which is highly available is allowed to experience frequent periods of inoperability as long as each interval is extremely short. Availability is frequently expressed as the fraction of time that a system is available to correctly perform its functions. Systems which cannot be taken offline for repairs, but which can tolerate occasional errors, have high availability as a design requirement. Fault tolerance is used to provide spares or backups when the primary unit responsible for service delivery fails.

4.3. Safety

Safety *S(t)* is the probability that a system will either perform its functions correctly or will fail in a way that does not disrupt other systems or jeopardize the safety of people associated with the system [5]. Safety is a measure of the fail-safe capability of the system. If the system does not fail-operationally, i.e. experiences a fault but continues to perform its functions correctly, at least it will fail in a safe manner. Certain techniques can be used to improve the safety of a system. An example of a fail-safe mechanism is the monitor interlock used on the Advanced Information Processing System (AIPS). This device allows a majority of a redundant group of processors to disable the output capability of a failed channel [5].

4.4. Maintainability

Maintainability *M(t)* is the probability that a failed system will be restored to an operational state within a specified period of time *t*. It is a measure of the ease with which a system can be repaired after it has failed. The repair process includes identifying the source of the problem, physically correcting the problem by replacing the failed module, and returning the system to its operational condition. Systems with a high availability requirement may have to be maintained while the system remains online. Fault tolerant design can contribute to the maintainability of a system, since the fault detection techniques that are necessary to achieve fault tolerance can be used to detect and identify problems for the purpose of maintenance. Since a significant portion of repair time is devoted to identifying the cause of a problem, the automatic diagnoses that are part of fault tolerance also improve the maintainability of a system.

4.5. RMAS Requirements for the Maglev Transportation System

Several quantifiable system level requirements are identified, or alluded to, in the documents comprising the GFI. These are discussed here.

The reliability requirement, stated as a probability of failure, for the system control software, and presumably for the hardware upon which it executes, for commercial maglev transportation, as specified in the draft Maglev System Parameters [3], is 10^{9}. This requirement is based on the commercial transport flight control requirements mandated by the US Federal Aviation Administration (FAA). Those requirements pertain to a 10-hr commercial passenger flight. The reliability requirement for commercial transport, then, may be specified as follows:

"The maximum acceptable probability of failure of the (safety-critical) flight control system is 10^{-10} per flight hour per aircraft."

It should be noted that for aircraft, the terms reliability and safety are used interchangeably as far as the flight-critical controls are concerned. This is due to the fact that the failure of a flight-critical computer is always assumed to result in a catastrophic aircraft failure. In other words, for flight control computers, there is no fail-safe state. Hence, the reliability of the system, i.e. the probability that it will operate correctly over a given time interval, is equal to the safety of the system, which is the probability that it will operate correctly *or* fail in a safe manner. This is not necessarily the case for maglev. If the control computer on-board the vehicle were to fail, it may not always lead to a catastrophic vehicle failure. For example, the computer may fail-stop and the vehicle may coast to a stop on the guideway. Or, the computer may cause the vehicle to exceed the speed limit which may be detected by a wayside computer which, in turn, may not turn the power on to the next section of the guideway resulting in the safe stopping of the vehicle. Thus, there are several alternatives available to bring a maglev vehicle to a safe stop in the absence of a functioning on-board computer which are not available to an aircraft in flight. For these reasons, the safety and the reliability requirements for maglev must be distinguished.

In particular, the reliability requirement stated above for a commercial transport aircraft becomes the safety requirement for maglev vehicles, which then may be specified as follows:

"The maximum acceptable probability of failure of the (safety-critical) computers is 10^{-10} per vehicle per hour of operation."

This requirement applies to the total computer system, including the hardware and software of the onboard vehicle, wayside and central control facility computers.

The reliability requirement for maglev relates to the probability of successfully completing a trip and a reasonable value for not completing a trip due to computer system malfunction is 10^{-6} per vehicle per hour.

The overall reliability and safety requirements for maglev may be illustrated as follows. If 1 billion trips, each of 1 hour duration, were undertaken by a fleet of maglev vehicles, then all except 1000 trips should be completed successfully. Of the 1000 trips in which the vehicles did not arrive at their destination without incident, only 1 would result in a catastrophic accident. If we assume that maglev trains have the same number of scheduled departures per day as planes, i.e. 14,000 per day, and that each trip averages one hour, these 1 billion trips will take approximately 195 years. Over that period of time, in a system which met the stated reliability requirement, there would only be five incomplete trips per year and a total of one catastrophic accident attributable to the failure of the control computer system.

The availability of the maglev transportation system, as defined in Section 1.2, is going to play a very important part in the public's acceptance of this mode of transportation. For the domestic US commercial airlines, the availability of the airliners approaches or exceeds 99 per cent. Less than 1 per cent of the flights are delayed or cancelled due to mechanical, electrical, hydraulic or other aircraft system related failures. As indicated above, there are more than 14,000 regularly scheduled commercial flights a day. It is obvious that the maglev transportation system will have to match or exceed this level of dependability in order to be accepted by the public. A reasonable availability requirement for maglev may be specified as follows.

" The maximum acceptable probability of not being dispatch ready for a trip for each maglev vehicle will be 10^{-2}."

This requirement applies to all the subsystems on-board each vehicle. The unavailability apportionment for the control computer subsystem is assumed to be one tenth of this, or 10^{-3} per vehicle per trip. That is, only one tenth of the unavailable vehicles will be stuck due to on-board control computer system failures.

Since the central control computer is directly involved in the control of each vehicle, a breakdown of this computer will disable maglev transportation in an entire region. Clearly, the availability of the central control computer must be greater that of a single vehicle. Furthermore, the failure of a wayside computer can disable a section of the guideway which in turn will disrupt many scheduled trips which have no alternate route to take. An acceptable level of downtime for an entire region of a maglev network is one hour per year. To realize this level of operation the availability of the central and wayside computers, stated as a probability of failure, is specified to be 10^{-4} per hour.

The maintainability of the maglev system must support the required availability discussed above. In other words, for on-board vehicle computers, unscheduled repairs can result in the disruption of no more than 0.1% of regularly scheduled departures. For the wayside and central computers, maintenance procedures can result in no more than one hour of downtime per year.

To achieve this level of maintainability, the three major computing components of the system, onboard, wayside and central computers, will perform, as part of their normal online operation, continuous built-in tests (C-BIT) capable of diagnosing failures to the level of a line-replaceable unit (LRU). When these tests detect a fault, the information is transmitted to the central computer where it is recorded and displayed for maintenance personnel.

For both the central and wayside computers, only online repairs are allowed. Thus, these computers are never taken offline for repairs. Instead, they are designed to be able to continue operation with a subset of unfailed hardware while failed components are replaced. After the repaired components are brought back into service, C-BIT tests must confirm the success of the repair operation. Furthermore, they must have backup options which allow the system to continue to operate even if a software fault occurs.

For the on-board vehicle computers, offline repair is also possible during the part of the day set aside for maintenance. During this maintenance period, more exhaustive maintenance built-in tests (M-BIT) are performed. If these tests uncover faults, they may be repaired either by replacing an LRU or by requiring the vehicle to return to a depot facility where a shop replaceable unit (SRU) is installed. Following the repair, a power-on series of initial built-in tests (I-BIT) are executed as well as the M-BIT series. These tests must confirm the success of the repair before the vehicle is re-certified for service. For the on-board vehicle computers, 99.9% of all repairs can be performed during regularly scheduled maintenance.

5. Architectural Approaches and Tradeoffs

The control computer system for the Maglev Transportation System must be able to govern train operations both safely and reliably. To design a fully automated control system which meets the safety and reliability requirements of the U.S. Maglev Transportation System requires a thorough understanding of the operation and interaction of the many components which must be controlled. This section presents some historical information on the development of automated train operations, as well as some details of the most advanced train control systems presently in operation or under development. A purely functional decomposition of the control system for Maglev is derived, based on the information contained in the Concept of Operations. A quantitative analysis of the computational requirements of each function is performed. Based on this quantitative analysis, two control computer architectures are proposed. Finally, a qualitative assessment of the advantages and disadvantages of each system is made.

5.1. Historical Background

Train control systems, regardless of the degree of automation, must perform three principal functions: vehicle control, train protection, and scheduling. These functions are performed by three main control system components: onboard control, wayside control, and central control. Each system component performs different aspects of each of the three principal functions. Historically, onboard train control has directed the activity of an individual train as it makes its way along the track. Thus, the onboard system has been used to control train velocity, acceleration, and braking which together make up the vehicle control function. Furthermore, the means to safely stop a train are typically controlled by onboard systems. Safe stopping is a key part of the train protection function. Wayside control has performed most of the functions pertaining to route integrity, another important aspect of train protection. For example, wayside control has been responsible for operating the switching mechanisms which guide a train onto one of several alternate tracks. If an upcoming section of track is occupied, or a routing switch in the track or guideway is not yet engaged in the proper position, the wayside signal indicates either "slow down" or "stop and proceed with caution." On the other hand, if the upcoming sections of track are clear, the wayside signal indicates a "go." Wayside control may also provide to the onboard system the maximum speed at which to run the train for a given section of track. This has provided a reliable means of safe train separation. Traditionally, the central control has performed the high level planning and coordination functions of railway transportation, such as setting up train arrival and departure schedules as well as scheduling maintenance operations. Central control also plays a role in coordinating the response of the system to an emergency situation, another part of train protection. From this discussion, it can be seen that each part of the system is primarily responsible for key aspects of either train supervision or train control, but that train protection is a function in which all parts engage.

In older systems, most of these functions are performed by human operators. Typically, the onboard controller is the engineer, who controls the train velocity manually. The wayside stations provide the engineer with signals, indicating whether or not to stop or proceed, based on information collected from other wayside stations or transponders which mechanically or electrically sense the presence of a train on a given section of track. In fact, wayside signalling function is usually the most automated part of traditional systems. The central control function is performed by a remote central office working in conjunction with a set of dispatchers, each of whom oversees a zone or region of track. The dispatcher uses voice communication to transmit movement authorities to the engineers of trains traveling through his section. It must be borne in mind that some of these systems evolved to permit trains traveling in opposite directions to use a single track. For purposes of this study, this is not a concern for Maglev vehicles, which will travel on uni-directional guideways and not be concerned with head-on collisions.

The modernization of train operations has been an evolutionary process, as must be the case for any equipment intensive industry. Hence, the use of electronic components and microprocessors to automate various functions has occurred in a piecemeal fashion, resulting in a set of federated systems which typically operate in parallel, using a method of checks and balances. This type of system architecture was dictated by the need to upgrade existing equipment. Older systems which worked could be improved in terms of performance and safety by adding on other systems. However, this architecture has proven to be very satisfactory, since many modern systems, built "from scratch", retain this federated control structure.

The drive to further automate the operation of existing and future rail systems is based on the proven ability of automation to deliver safer and more reliable service, with faster operating speeds and increased capacities, while providing lower operating costs. The automated versions of the three principal functions, vehicle control, train protection, and scheduling, which make up the federated train control architecture, are respectively referred to as automatic train operations (ATO), automatic train protection (ATP[1]), and automatic train supervision (ATS). Interestingly, ATO is not usually considered a vital or

[1] To add just a little confusion to the terminology, ATP may also be referred to as automatic train control (ATC). ATP systems are generally considered more sophisticated than ATC safety systems.

safety critical system, since a second, independent safety system, the ATP system, is charged with overseeing train safety. ATP can override ATO and take control of a train which has exceeded some safe operating threshold, usually for excessive speed.

ATO systems have both onboard and wayside components. Speed commands are delivered from the wayside controller to the train and are acted upon by the onboard system with no operator intervention. The Bay Area Rapid Transit (BART) mass transit line in San Francisco is one of the earliest examples of a commercial ATO system. Other commercial systems employing ATO are the Washington, D.C. metro, the Vancouver Skytrain, and the Lille VAL Line 1.

ATC safety systems are usually provided as a backup to a human operator. In these systems, the speed command which is transmitted from the wayside, is displayed to the operator in some manner in the control cab. If the operator does not run the train at or below the indicated speed, the ATC system automatically applies the brakes. Since in all conventional wheel-on-rail train systems, application of the brakes disengages the propulsion system, the train speed is brought under control without competition between the propulsion and braking systems. The French high speed TGV line and the German high speed ICE lines are designed to use this type of ATC. In primitive ATC systems, the wayside can transmit only a very limited amount of data to the train. The only information may be the frequency of the signal itself, which is interpreted as the maximum operating speed. The absence of the signal indicates a command to stop. In these systems, the limit on the amount of information sent to the train is a function of the communication medium employed. Current passing through a track circuit is pulsed at a rate dependent on the number and position of closed relays in the circuit. A closed relay indicates the presence of a train on the track. This signal is picked up inductively by a receiver on the train.

ATP systems, which are more sophisticated versions of ATC, are characterized by a message based communication scheme using radio signals as the communication medium. Hence, not only is it possible for the wayside system to send a great deal more data to the onboard system, but also two way communication is possible. ATP systems have more extensive knowledge of the route and its current condition. A higher degree of control is possible because of the increased quantity and frequency of data which can be exchanged.

Train control and train protection functions, whether manned or automated, have always had an element of feedback in their operation. For example, the speed at which the train should be traveling is compared with the measured speed and an appropriate adjustment, or control, action is taken to correct the difference between the two.

However, scheduling and high level operations of railways have typically lacked this feedback component. In the vocabulary of control systems theory, older train supervision systems are said to run in an "open-loop" manner, i.e. they are loosely coupled, and have minimal information flow for the supervision function among the three control components. An open-loop control system does not make use of current information about the state of a system to direct the current activities of the system. Instead, open-loop systems make use of a pre-defined strategy to control system operations. Figure 5-1 shows a model of an open-loop train traffic control system where the current state of the train and track resources do not have an impact on the central traffic planning function. Instead, decisions about movement authorities, i.e. which train has the right of way over a given section of track, are made on localized data with the primary concern being safe train separation. For example, the Burlington Northern Railroad conducts its current operations in this way with mixed results [11]. Currently, BN operates 60,000 freight cars with 2,300 locomotives over 40,000 kilometers of track, most of which is bi-directional. Although the current system achieves its goal of transporting cargo safely between stations, the operation is inefficient in terms of fuel and other operating costs, and unreliable in terms of delivery times. In an effort to reduce costs and improve reliability, Burlington Northern has studied an ambitious program to upgrade their control system. The Advanced Railroad and Electronic System (ARES) is an integrated command, control, communications and information system which could replace the manual and semi-automated system currently being used.

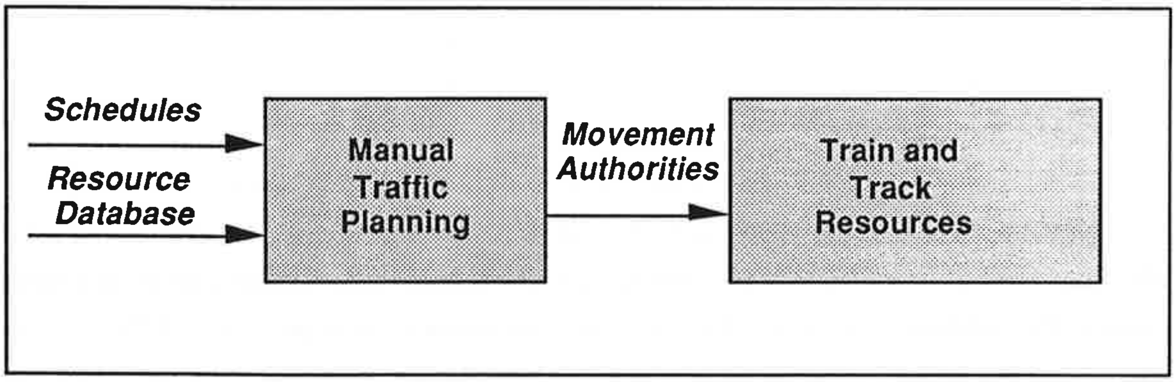


Figure 5-1. Paradigm of an open loop train supervision system

Under ARES, the locations and speeds of all BN trains, derived from the Global Positioning System (GPS), will feed into a central control computer. Using this information and the schedule which the system is trying to meet, a traffic planning program will produce the best plan for operating the system. By using real time data to monitor the system, traffic flow and maintenance of way (MOW) operations can be coordinated so as to improve overall efficiency and reliability. Human operators will continue to make local decisions regarding movement authorities in conjunction with information provided by the central control computer, and trains will still be operated by human engineers. However, it is conceivable that train control could become fully automated, with human operators performing a supervisory function. Figure 5-2 depicts a model of a closed-loop train traffic control system of which ARES is an example.

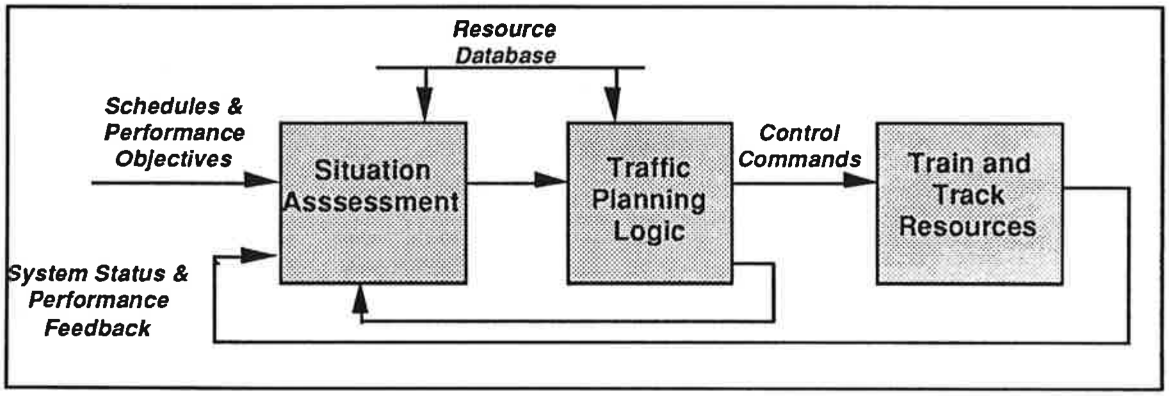


Figure 5-2. Paradigm of a closed loop train supervision system

Other advanced train control systems are under development in the U.S. and Canada. For example, ARINC Research Corporation has developed a system architecture for an Advanced Train Control System (ATCS) [17]. Although ATCS replaces wayside signalling with cab signalling and does all the computation necessary to control the train, the control of the train speed is still performed by a human operator, using data displayed on a terminal in the cab. The designers of this system believe that retaining the human operator in the control loop has led to a considerably more complex system architecture than would be necessary for a fully automated system.

5.2. A Survey of Existing HSGGT Control Systems

One requirement of high speed guided ground transportation (HSGGT) is automated closed-loop control. In Europe and Japan, automatic train control (ATC) and automatic train operation (ATO) have become standard operating practice [14]. Throughout Europe, there is a significant trend, notably in France, Sweden, and Germany, to install ATC and ATO on all principal rail lines in an effort to reduce accidents caused by human error [14].

In France, the French National Railways (SNCF) have been developing several advanced signal and train control systems for high speed and conventional lines. These include: (1) the signalling system for the TGV Sud-Est, Atlantique, and Nord lines, (2) the SNCF "Astree" system, (3) the SACEM (Système d'aide à la conduits et à la maintenance) (system to aid operations and maintenance) of the RER line of SNCF which serves the Paris area, and (4) a subset of the ambitious "Astree" System which can be deployed at an earlier date.

The automated signal and control system presently in operation on both TGV lines is typical of many automated designs. The control strategy is based on adjacent zones or blocks, each of which is 2.1 km long [11], [17]. The system is intended to provide a safe train separation of at least one block between consecutive trains. At the entrance to each block, the train receives data from an ac audio-frequency coded track circuit indicating the speed limit in the current block and the maximum allowable speed permitted at the end of the next block. If the operator onboard allows the train to exceed the specified maximum speed, braking is automatically triggered. A more advanced ATC system is planned for TGV Nord [11]. This system will allow more precise monitoring of train speed by increasing the amount and frequency of data transmitted by the track circuit to the train. The stepped function used to reduce train speeds will be replaced by a continuous function. Thus shorter block lengths can provide the same level of protection to a stopped train, or other obstruction, resulting in a decrease in required headway and an increase in potential capacity. The present four minute headway (8 to 10 km on level track from 300 kmph) could be reduced to three.

TGV also provides some automated verification of route integrity as well as automated surveillance of operator response [11]. On the TGV Atlantique line, various intrusion detection systems are tied to the signaling system. For example, the areas of track which lie under bridges are covered by meshed wires which detect falling objects which have dropped onto the track.

Dispatching and routing of TGV Atlantique trains is controlled from one central location according to a predetermined plan [11]. The system is fully automated. However, manual overrides are available if desired by the dispatcher. This central location also controls the electric power to the entire line. Power to any point on the track can be cut by de-energizing the power to the section in which the point is located.

The onboard control system of TGV Atlantique comprises 18 microprocessors which communicate over a specially developed dual ring network known by the acronym TORNAD. The dual ring automatically reconfigures in the event of a failed link or node. The control and monitoring functions performed by the system include the anti-skid system, the braking system control and monitoring, automatic door control, train lighting and air conditioning, cab signalling and operator display, communication and passenger information. The main computer in the operator's cab is backed up by a separate standby unit. Watchdog functions and data transmission error checking provide some error detection capability. During a run, the cab display provides the operator with the real-time status of vital onboard equipment and provides computerized troubleshooting of failures to help determine the correct remedial action.

The "Astree" System is still under development and many details remain to be finalized. However, the goal of the system is to provide SNCF with system-wide control of train movements in real-time. Each train will continuously calculate its position and speed and transmit this information to a monitoring control center. Technologies to be used onboard vehicles include Doppler radar to calculate distance, electric odometers to designate track positions, and radio beacons to identify trains. The simpler precursor of "Astree", due to come into service in 1994, will use an intermittent, rather than a continuous, ATC system in which data is transmitted to the control center at discrete points along the track, for example, at the start of each block, rather than continuously.

In Germany a continuous automatic train control and track-train communication system called LZB is being applied to new and existing lines [17]. This is a centralized system in which a "vital" train control computer determines authorized speeds and distance to stop and transmits this information to the train. Presently, the high-speed Inter-City Express (ICE) trains operate with about the same degree of automation as the TGV [17]. However, unlike the TGV, the ICE control system allows for full manual control of train speed, with the provision of some automated watchdog protection. The system can also be operated with fully automatic speed control and with manual speed setting and automatic control. Neither manual or semi-automatic modes of operation would be applicable to Maglev. In fact the German maglev system, Transrapid (TR), is fully automated.

The ICE braking system is computer assisted [17]. Although the braking tests are automated and all braking systems are continuously monitored, information about brake failures which are detected by these tests must be entered by the operator who then must calculate and enter, via his console, any reductions in braking capability. Reductions in braking capability are compensated for by a reduction in the maximum speed permitted by the automatic train control system. The interaction of the various braking systems and the anti-skid systems is also under microprocessor control which is programmed to favor dynamic regenerative braking because it is more energy efficient and less maintenance intensive.

The control system of the ICE is highly distributed and requires a sophisticated communication system to coordinate the activities of the various computing elements involved. The various onboard systems communicate over a fiber optic network. This link is used to coordinate propulsion and braking of the individual cars in a so-called consist, or multi-vehicle train, to carry voice communications, and to relay the status of the ongoing diagnostics testing. The cable is looped and networked so that messages can be routed around a failed component. For the transmission of vital information between the train and the wayside stations, inductive loops, 300 m in length, are laid inside the track. Data is transmitted from the wayside to the train at 36 KHz and from the train to the wayside at 56 KHz. However, the inductive cables require a high degree of maintenance and the German Federal Railway is planning to employ radio communications in the near future. Safety information sent from the train to the wayside locations includes the current braking capability of the train, the train identification number, the length of the train, and its speed. This data is transmitted every 14 seconds. The safety relevant data which the wayside transmits to the train includes the distance to the next stopping point and the speed/braking curve to be utilized. This data is transmitted at a frequency of 1 Hz. This information is used by onboard logic to determine the actual train speed to be achieved and the necessary braking or power commands. Thus the speed control loop resides entirely onboard the train. The actual train speed and direction is monitored by redundant pulse generators operating at 16 pulses per revolution. Acceleration readings are provided by an onboard accelerometer.

Information about the integrity of the route is obtained with audio-frequency track circuits. This a microprocessor based interlocking control system developed by Siemens. Broken track can be detected as well as the presence of a train on the track. These systems have been programmed using accepted software engineering principles to obtain a greater degree of software reliability. For example, sections of the code are uniform throughout the system. These sections relate to the basic logic of every interlocking control system. Interfacing to this software is the station-specific software which defines the unique aspects of a particular station.

Since the onboard and wayside computers play a vital role in train safety, a TMR (triple modular redundant) architecture is used to increase reliability of the control system. Three computers operate in parallel. As long as two of the three agree on a result, the system continues to function. If two out of three do not agree, the information is not acted on and the system reverts to a fail-safe, non-automated operating mode. A dual redundant architecture is used for the microprocessor based interlocking control system used to check route integrity. Although both operate in parallel, only one is considered "online." If their results do not agree, an alarm is sent to maintenance personnel and the online and offline computers exchange roles. Clearly, these approaches to fault tolerance suffer from a lack of theoretical rigor in their designs. For example, in the dual redundant system, the redundancy management logic assumes that the two systems disagree because the online module is faulty. However, it could easily be the case that the offline system is the faulty system and yet this is the system which is being switched online. Furthermore, in the TMR system, the fact that two out of three disagree indicates a serious failure in the system. At this point, it is impossible to ensure that an orderly transition to the fail-safe operating mode will occur.

Almost all ICE control functions are implemented by means of software [17]. Hardware circuits are employed only in a limited number of cases to reduce the load on the computer. The high degree of automation also allows the ICE onboard computer system to perform extensive monitoring of all aspects of train control and operations. If a fault is detected, the system conducts an appropriate set of diagnostic tests, logs the results, and alerts both the operator and the maintenance facility [17]. Due to the complexity of the system and the high degree of reliance placed on the automatic controls, automated monitoring and diagnostics are necessary not only to assist maintenance operations, but are also required for train safety and reliability.

5.3. A Functional Decomposition of the Maglev Control System

Each of the three principal functions which must be performed by the Maglev control system, i.e. control, protection, and supervision, can be decomposed into several well-defined sub-functions. This purely functional analysis of the control system may be used as the basis for the design of a control architecture. By partitioning the sub-functions among the various control elements which make up the system, an optimal design for safety and reliability can be achieved. The results of the functional decomposition of the Maglev control system are presented in Table 5-1.

5.3.1. Vehicle Control Functions

5.3.1.1. Vehicle Location (Safety Critical)

The vehicle location function determines the vehicle position, travel direction, speed, and acceleration. The vehicle position is important both in absolute terms of latitude and longitude, as well as in relative terms with respect to the guideway, safe stopping places, and station locations. This function provides the feedback information required by other control and protection functions to determine the actuation necessary to achieve the position, acceleration and speed called for by existing conditions and the mandated travel profile of the vehicle.

CONTROL	PROTECTION	SUPERVISION
Vehicle Location	Safe Vehicle Separation	Route Planning
Velocity Control	Vehicle Position Control	Route Scheduling
Levitation Control	Route Integrity	Dispatching
Lateral Position Control	Emergency Stopping	Maintenance Scheduling
Propulsion Control	Emergency Speed Control	Operator Interface
Secondary Suspension Control	Emergency Position Control	Status Displays
Route Control	Emergency Response	Passenger Supervision
Vehicle Systems Monitoring	Failure Management	
Vehicle Systems Control		
Environmental Monitoring		

Table 5-1. Functional Decomposition of the Maglev Control System.

5.3.1.2. Velocity Control (Safety Critical)

The velocity control function causes the speed and direction of travel of each vehicle to match the speed mandated by its travel profile in accordance with the existing conditions on the guideway. The velocity control must coordinate the activities of the individual propulsion power units in the guideway, each spaced at approximately 2 km intervals. The velocity control communicates directly with the lower level propulsion control to achieve the desired velocity. Inputs to this function are the condition of the guideway in the next zone, the distance to the vehicle ahead, switches to be navigated in the next zone as indicated by the travel profile of the vehicle, and the speed mandated by the travel profile for this zone. Primary braking is controlled by this function. It works by reversing the vehicle thrust in the long-stator propulsion motor. Electrical energy so generated is dissipated in substation load resistors.

5.3.1.3. Levitation Control (Safety Critical)

EMS Version: The small (8 mm ± 4 mm) gap between the levitation magnets and the guideway must be actively controlled at a frequency of approximately 100 Hz. This is accomplished by measuring the gap and controlling the current flow in the onboard electromagnets, which in turn determines the attractive magnetic force between the vehicle electromagnets and the guideway ferromagnets.

EDS Version: The gap does not require active control. However, the temperature of the superconducting onboard electromagnets needs to be carefully monitored. The temperature must be maintained at approximately 5° K. If the temperature rises above this operating temperature, a phenomenon known as quenching will occur, which ultimately causes the magnets to fail. Thus, the onboard cryogenic system must maintain the temperature of the superconducting magnets within their operating range.

5.3.1.4. Lateral Position Control (Safety Critical)

EMS Version: The gap between the lateral guidance electromagnets and the guideway must be actively controlled in the same manner as that of the gap control of the levitation magnets.

EDS Version: The temperature of the superconducting guidance magnets must be maintained in the same manner as that of the temperature control of the levitation magnets.

5.3.1.5. Propulsion Control (Safety Critical)

Propulsion is achieved by the use of a linear synchronous motor made up of conducting windings, installed along the length of the guideway, and variable frequency converters which, together with the necessary switch gear, are located along the guideway at 2 km intervals. Each of two such converters supplies half the power needed. This increases the control requirements, in that two converters must be controlled, but provides some propulsion capability in the event of a failure of one of the converters. The two km sections of guideway are called zones. The speed of the vehicle within a zone is controlled by varying the frequency of a traveling electromagnetic field produced in the stator windings of the guideway by the power electronics system. The speed determination and the signal to direct power to the stators comes from the velocity control function as inputs to this function. This function directs the behavior of the power converters to achieve the indicated speed and direction of the vehicle as it enters the zone corresponding to this power station.

5.3.1.6. Secondary Suspension Control (Safety Critical)

The purpose of the secondary suspension system is to provide a satisfactory level of ride comfort to passengers in the vehicle. EDS systems which possess a fairly stiff primary suspension will require an actively controlled secondary suspension to achieve a satisfactory level of ride quality. For EMS systems, active control may not be necessary but could be provided to enhance the overall smoothness of the ride and offset jerk due to sudden accelerations and lateral motions induced by turns and cross-winds. The secondary suspension provides an additional level of isolation between the coach body and the guideway. When the velocity of the vehicle is sufficiently high, aerodynamic control could be used to provide a better ride quality. This is accomplished by controlling a flap at the trailing edge of a vehicle to provide secondary suspension. The control algorithm for this application would be similar to control algorithms for gust alleviation in an aircraft. For slower speeds, secondary suspension is provided by actively controlling springs which stabilize the roll motion and the vertical and lateral motion of the vehicle with respect to the magnetically levitated bogey.

5.3.1.7. Route Control (Safety Critical)

The route control function guides the vehicle through one of two possible paths at switching points in the guideway. The route profile indicates the correct path to take at every switch. Vehicles may either remain on the main guideway or exit at the switch point to stop at a station. For the purpose of this study, a passive switch design is assumed. Magnetic forces are used to effectively steer the vehicle through the passive switch. The route control function energizes the appropriate coils in the guideway which determine the direction of travel through a switch in the guideway.

5.3.1.8. Vehicle Systems Monitoring (Mission Critical)

During operation, the secondary brakes are monitored at a rate of 1 Hz. This includes a measurement of the charge level of the onboard batteries, as well as a check on their recharging function. Other onboard systems such as onboard lighting, temperature, air flow, and door position control are monitored at a rate of 1 Hz.

5.3.1.9. Vehicle Systems Control (Mission Critical)

Onboard systems such as lighting, temperature, air flow, and door position are adjusted in accordance with the desired level of each parameter, given the feedback information provided by the vehicle systems monitoring function.

5.3.1.10. Environmental Monitoring (Mission Critical)

Wind speed and direction and external temperature are measured periodically. Wind speed is measured at a frequency of 5 Hz and temperature is monitored at 1 Hz. The electromagnetic field at various positions within the coach is monitored at a rate of 1 Hz. Since high field strengths may pose a health risk to passengers, a running average is maintained and readings above a certain threshold are logged. Other means are used at a much lower rate to detect weather conditions of rain, snow, etc.

5.3.2. Vehicle Protection Functions

These functions provide a fail-safe mode of operation. Therefore, they can override the actions of the ATO functions and take control of a vehicle which has exceeded some safety threshold.

5.3.2.1. Safe Vehicle Separation (Safety Critical)

For the purpose of this study, a minimum spacing of 4 km must be maintained between any two consecutive vehicles on the guideway at all times. This is equivalent to a distance of two control zones. Prior to entering a new zone, the distance to the vehicle ahead is calculated. The speed of that vehicle is also determined. If this distance is less than the required 4 km minimum, emergency stopping procedures are activated. For example, a vehicle traveling at the maximum speed of 500 kmph can be decelerated to zero kmph in a distance of 1 km if a deceleration of 1 g is used. If the deceleration rate is 0.25g, the stopping distance increases by a factor of four to approximately 4 km. If the vehicle ahead conforms to the safe train separation requirement, then the position and velocity of that vehicle is passed to the vehicle position control function.

5.3.2.2. Vehicle Position Control (Safety Critical)

This function uses the actual vehicle position and the desired vehicle position to determine the error, if any, between these two values. If the vehicle is not within safe tolerances of its expected position as required by its travel profile, then the vehicle poses a safety hazard to itself and other vehicles which may be exiting or entering the guideway. Therefore, corrective action is taken, typically by increasing or decreasing its speed, with appropriate cautions for existing conditions, to cause it to conform with the expected position called for by the travel profile.

5.3.2.3. Route Integrity (Safety Critical)

Sensors monitor, record and transmit to this function data about the integrity of the guideway and the propulsion and levitation coils. The guideway must remain properly aligned and free from debris, e.g. ice or litter, which could obstruct the route. Sensor data is pre-processed and condensed before transmission. Exceptional conditions result in the appropriate corrective action throughout the system.

5.3.2.4. Emergency Stopping (Safety Critical)

This function is employed when the primary braking capability of the linear synchronous motor provided by the guideway has failed. It is a fail-safe mode of operation. The vehicle must be able to stop itself at a safe stopping area without any input from central control. Secondary braking is accomplished in three parts. The first uses aerodynamic braking from either the trailing flap or a parachute. This is only effective at higher speeds. The second uses eddy currents to slow the train to the point where it can safely land on its landing skids or wheels. Friction stops the train from that point on. The eddy currents are induced in the track guide rails by longitudinal vehicle magnets. Eddy current braking is also effective only for higher speeds. Secondary braking is used to slow or stop the vehicle when the primary braking system does not reduce its speed in accordance with the route profile. This function controls the use of the secondary braking system.

5.3.2.5. Emergency Speed Control (Safety Critical)

This function is employed when the propulsive force provided by the guideway has failed. It is a fail-safe mode of operation. The vehicle must be able to reach the next safe stopping location by using onboard power only. This means that sufficient energy must be available from 2 of 4 batteries and the linear generators (which operate only while train is moving) to control levitation, braking, and other loads. The vehicle is only allowed to stop at areas deemed safe stopping areas. Although entire sections of the guideway may be considered safe for stopping because emergency descent ladders are long enough to allow passengers and crew to descend from the guideway, other sections may be restricted so that the vehicle may only be allowed to stop at certain points. This function acts in cooperation with the Emergency Stopping function.

5.3.2.6. Emergency Position Control (Safety Critical)

This function determines the exact stopping point for a vehicle in an emergency stopping situation, i.e. when the primary propulsion system has failed. Information about safe stopping points is maintained for each zone along the guideway. The entire zone may be deemed a safe stopping area in which case the stop is made as soon as possible within the zone. In restricted areas, the vehicle is constrained to stop at specially designated safe stopping points. These areas may provide turnouts where disabled vehicles can wait for emergency repairs; meanwhile a spare vehicle is dispatched to allow normal operations to resume as soon as possible. This function passes the safe stopping position to the emergency speed and braking functions.

5.3.2.7. Emergency Response (Safety Critical)

The emergency response function responds to emergency conditions by reducing the speed of trains on a regional level. It coordinates the efforts of specially trained personnel and dispatches emergency equipment to the site of the emergency. If push recovery is needed, this function coordinates that effort.

5.3.2.8. Failure Management (Mission Critical)

This function handles failures requiring unscheduled maintenance. Spare vehicles are put into service as necessary and alternate schedules are put into effect.

5.3.3. Vehicle Supervision Functions

5.3.3.1. Route Planning (Safety Critical)

Route planning is performed on the basis of marketing information and resource availability. The detailed information regarding a route, including station stops, switch settings, speed profile, etc. is stored for each route in a file called the route profile. This file is routinely updated with information provided from the maintenance system which may indicate, for example, that sections of guideway can only support degraded speed profiles. When a given train is assigned to travel a given route, the position of the train is tracked and compared to the data in this file. Extensive, detailed data about the guideway is also stored in another file called the global route database. If route changes or additions need to be made, these changes are verified against the global route database for accuracy and consistency. Route profiles are downloaded to individual trains and to wayside zone controllers on a daily basis.

5.3.3.2. Route Scheduling (Safety Critical)

Route schedules are set based on the demand for a given route and knowledge of the upper limits on capacity and safety requirements of a given route. Simulation programs which permit predictions of the effect of alternative timetables allow operators to choose schedules optimized in accordance with the observed demand for service and the safety requirements of the system. Since the operating staff can intervene and change timetable data, changes are automatically verified to ensure that a change does not create a safety problem through the introduction of a human error. Special programs verify that the schedule is safe and self-consistent, i.e. so that two trains are not scheduled to be in the same place at the same time and a safe headway is preserved at all times between consecutive trains. Travel schedules are downloaded to individual trains and to wayside zone controllers on a daily basis.

5.3.3.3. Dispatching and Docking (Safety Critical)

Schedules are particularly sensitive to the need to reduce speed to navigate switch points when exiting or entering the guideway for station stops. Added safety measures are required when vehicles exit or enter a station area. For example, the safe vehicle separation distance is diminished when trains are within a station due to their greatly reduced speeds. This function performs the additional coordination of the activities of vehicles sharing station facilities.

5.3.3.4. Maintenance Scheduling (Mission Critical)

Routine maintenance is scheduled by this function for every vehicle, the guideway, power stations, passenger stations and wayside stations in accordance with the manufacturer's or the construction contractor's specifications. This function works in conjunction with the Route Planning and Route Scheduling functions. All lower level maintenance functions return data to this central function. For example, on-condition monitoring ensures that the minimum dispatch complement (MDC) of all system components is present before a train leaves a station. This function oversees the exceptional cases which require unscheduled maintenance of a vehicle and result in the substitution of a spare vehicle in place of the vehicle which does not possess its MDC.

5.3.3.5. Operator Interface (Mission Critical)

Each vehicle has an operator interface in the cab to display the status of all systems on the vehicle as well as the relative position of the vehicle on its route, its velocity, weather conditions, and the relative positions of other vehicles, safe stopping points, and stations. The operator is responsible for ensuring that passengers board and disembark safely before closing the doors prior to allowing the automatic systems to take control of the vehicle. The vehicle cannot move if the doors are open. Although doors are opened automatically, they are closed under operator control. When the vehicle is moving on its wheels for emergency operation, the vehicle is under manual control. Furthermore, the central facility also has an operator interface to allow real-time changes to schedules and routes in response to unplanned events, such as severe weather conditions and emergencies.

5.3.3.6. Status Displays (Mission Critical)

Central traffic displays show traffic information in a manner "that is conducive to interactive discourse among the staff" [3]. These displays use high resolution graphics to present real time data about all aspects of the system. The information which is displayed includes a top level view of the relative positions of each vehicle in the system. This top level view also provides local weather data. The actual route traveled by each train is compared to its route profile by means of a velocity versus time plot. Monitoring information about the status of the guideway, switches, switching propulsion coils, wayside stations, passenger stations, and other specific items of equipment are also displayed. Whenever exceptional conditions are noticed by monitoring software, these conditions are brought to the attention of the operations personnel by means of audible alarms and flashing displays. Furthermore, the status of all monitored onboard systems is presented to the operator of each vehicle by means of a display in the cab.

5.3.3.7. Passenger Supervision (Safety Critical)

This function directs the onboard activities of passenger and crew regarding restricted travel times during which all travelers must remain seated.

5.4. Candidate Architectures for the U.S. Maglev Transportation System

For the Maglev system described by the Concept of Operations presented in Sections 1-2, train control will be fully automated. The high speed of the vehicles and the short headway between them require the ability to respond to changing conditions which far exceeds the capabilities of a human, however vigilant and well trained. Thus, the three principal functions used to operate the Maglev transportation system, i.e. control, protection and supervision, will be fully automated, tightly coupled, and have significant amounts of information flow between them. Section 5.3 presented the principal functions to be performed by the control system. The various functions can now be mapped to specific computation sites within the overall control computer architecture. Table 5-1 presented the functions which are necessary to perform automatic train control. Alternative ways to partition these functions among the various control computers which make up the control system can now be analyzed. Some tradeoffs between different function assignments can also be made.

The organization of the overall control architecture for the regional line presented in the Concept of Operations is very similar to control architecture of the Transrapid system. The three principal subsystems which make up the whole are the onboard vehicle computer system, the wayside zone computer system, and the central facility computer system. The system described here could easily be expanded to include automated control of many interconnected regional networks of Maglev lines by expanding the role of the central control facility. This would probably require a hierarchical organization similar to that of the U.S. air traffic control system. However, this functionality is beyond the scope of this report.

Two basic architectures, which represent extremes of a continuum, are presented here and are analyzed quantitatively in Section 6. In the first architecture, the primary responsibility for train control rests with the wayside zone control computers, with the onboard system providing backup and consistency checking. Train protection is distributed among the three subsystems. This architecture is referred to as the Zone Control Architecture (ZCA). The function-subsystem mapping for this architecture is shown in Table 5-2. A greatly simplified data flow diagram for the ZCA architecture is shown in Figure 5-3. In the second architecture, the onboard computer has the primary responsibility for train control, with the wayside zone computers providing backup and consistency checking. In this architecture, referred to as the Smart Vehicle Architecture (SVA), the functions relating to vehicle protection are again distributed among the three subsystems. Table 5-3 shows the mapping of each control function to a computer subsystem for the SVA and Figure 5-4 shows the simplified data flow diagram for this architecture.

In both architectures, the vehicle protection subsystem operates independently from the vehicle control functions. and therefore provide a fail-safe mode of operation. In cases where the speed or position of a vehicle exceed safety thresholds, these protection functions can override the actions of the control functions and assume control of the vehicle.

In both the ZCA and SVA architectures, the supervision function is performed by the central facility computer system, which also includes major computing subsystems located in stations. However, the supervisory data which is needed by the primary control computer to adequately perform its function is transferred to that computer on at least a daily basis and may be updated more frequently. This information has been referred to in this document as the travel or route profile for a given vehicle. The format of this information will vary, depending on whether it is to be used by the onboard computer or the wayside computer. This method of anticipating future behavior of the vehicle and confirming its correct behavior in the present supports the method of train scheduling specified by the Concept of Operations. This form of operations views all normal train travel as planned in advance and all passengers riding in reserved seats. However, either architecture could be adapted to a more dynamic method of operations, based, for example, on the airlines shuttle service. This service is demand driven, rather than schedule driven and requires more real-time planning capability. Planning algorithms typically require significant memory and throughput in their computing platforms. Hence, the capacity of the present architectures would need to provide adequate and easy means of expandability if this functionality is to be addressed in the future.

In general, those functions which can, for obvious reasons, best be performed in one site over another, are assigned to those sites. For example, the control of levitation, guidance, secondary suspension, and onboard systems like air conditioning and lighting all require the control of onboard actuators. Furthermore, the sensors needed to obtain feedback information for these systems are also onboard. Hence, the control of these functions should obviously be performed by an onboard computer. Another example of a set of functions which can most easily be performed by an onboard computer is that of emergency stopping, emergency speed control, and emergency position control. The emergency which these functions are intended to address is the failure of the guideway propulsion and primary braking capability. The power for these emergency operations comes from batteries carried onboard for that purpose. The vehicle must be able to reduce its speed, continue onto a safe stopping area and stop there so that passengers can disembark safely from the vehicle and the elevated guideway. Again, these functions can most easily be performed with a minimum of communication overhead by the onboard computer.

The rationale behind the ZCA is that, unlike conventional transportation systems in which the power for vehicle propulsion is onboard, the Maglev system is powered from the guideway. The propulsion control platform must be co-located with the power converters since the required iteration rate is so high as to preclude an allowance for any communication overhead. Hence, by controlling the vehicle velocity from the wayside computers, the communication overhead between the vehicle and the propulsion control can be eliminated. Furthermore, since the wayside zone controllers must communicate with each other to coordinate the speed of the vehicle as it passes form one zone to the next, they can also perform the function of safe vehicle separation and vehicle position which are related to vehicle speed. The communication between wayside systems can be carried out through very reliable media such as redundant fiber optic cables. By locating sensors in the guideway, the vehicle location and route integrity functions can also be performed by the wayside using sensors embedded in or alongside the guideway for these purposes. Again, collecting this information from stationary sensors can be accomplished through secure media. Route control, or the direction of the vehicle through a switch, can also be performed from a wayside computer, especially if the switching mechanism is that of a moveable section of guideway. The ZCA most closely resembles conventional railway control systems without the attendant communication overhead that characterizes systems controlling onboard propulsion from the wayside [18].

The rationale behind the SVA is that an autonomous vehicle can most easily direct its own motion since it is in a position to obtain information about its own state and the state of its surroundings. It must be able to perform these functions for emergency purposes anyway. Hence, it may as well perform them for normal operations as well. It can easily communicate speed commands to wayside propulsion control systems using radio communication. Information about its speed, position, and acceleration are also easily obtained from a combination of GPS (Global Positioning System) and micro-mechanical instruments based on Inertial Navigation Systems, the technology for which will be available by the time the U.S. Maglev Transportation System reaches the prototype development stage. With information about its position and speed, it can perform the functions of safe train separation and vehicle position control by communicating with vehicles both ahead of it and behind it on the guideway. By using onboard sensors, it can also perform route integrity checks both for alignment and obstacle detection. Furthermore, it can easily direct its movement through switches, i.e. perform route control operations, especially if the switches in use do not involve moveable sections of guideway but rather some vehicle-borne steering mechanism. The SVA most closely resembles the most advanced control systems being installed on conventional and high speed rail systems [18].

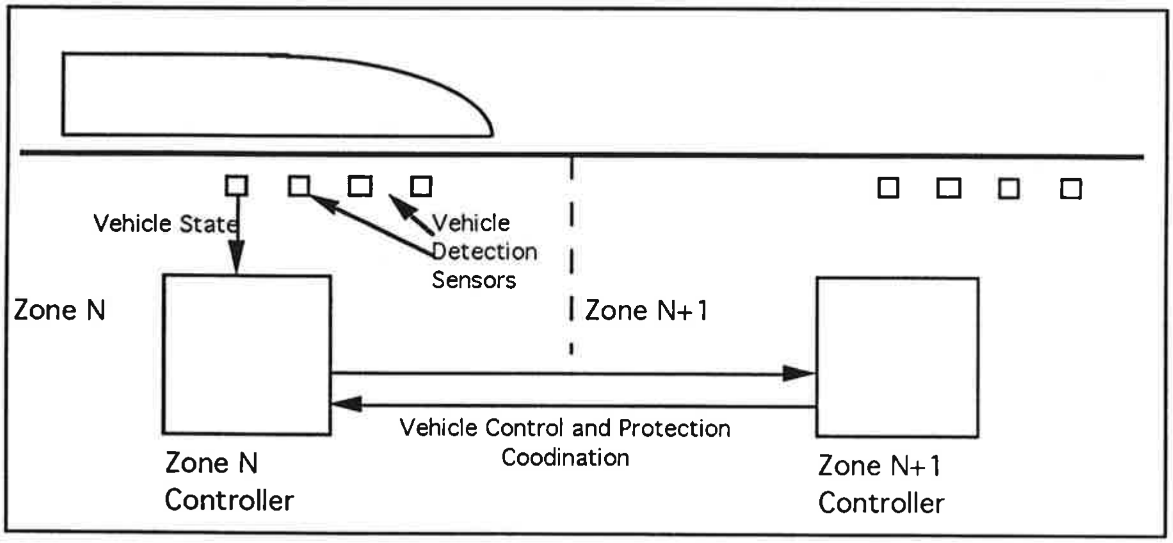


Figure 5-3. Simplified Data Flow Diagram for Zone Control Architecture (ZCA).

In order to determine the relative strengths and weakness of each architecture, a quantitative analysis must be performed. This analysis is discussed in the next section. However, a few qualitative observations may be made at this point. For example, heat and vibration are clearly more of a factor in the design of an onboard, i.e. moving, system than in the design of a stationary wayside system. These considerations show up as an added cost to assure that the system is packaged and shielded to be able to withstand vibrations and temperature extremes. In general, any hardware feature needed by one architecture but not the other and which increases the initial cost of the system is a factor which weighs against that architecture.

Initial cost is only one of many cost related considerations. The final operating cost of a system which is less expensive initially but more expensive to maintain may quickly exhaust any initial savings. For example, an onboard system can easily be taken out of service for maintenance and diagnostic checks without compromising the operation of the rest of the system. However, a wayside computer must be kept online at all times. Although it is possible to design a system which can be serviced during normal operation,

	Onboard	Wayside	Central
CONTROL			
Vehicle Location	B	P	
Velocity Control	B	P	
Levitation Control	P		
Lateral Position Control	P		
Propulsion Control		P	
Secondary Suspension Control	P		
Route Control	B	P	
Vehicle Systems Monitoring	P		
Vehicle Systems Control	P		
Environmental Monitoring	B	P	
PROTECTION			
Safe Vehicle Separation	B	P	
Vehicle Position	B	P	
Route Integrity	B	P	
Emergency Stopping	P	B	
Emergency Speed Control	P	B	
Emergency Position Control	P	B	
Emergency Response			P
Failure Management			P
SUPERVISION			
Route Planning		I	P
Route Scheduling		I	P
Dispatching		I	P
Maintenance Scheduling			P
Operator Interface	P		P
Status Displays	P		P
Passenger Supervision	B		P

P: primary assignment, B: backup assignment, I: access to information in a database.

Table 5-2. Zone Control Architecture (ZCA) Function Assignment.

such a system is more complex and more difficult to maintain. Furthermore, onboard systems which have sufficient spare capacity to be operated in a degraded mode after some component failures can be brought into a depot for maintenance while wayside systems will require visits by the maintenance crews to conduct routine inspections and make repairs.

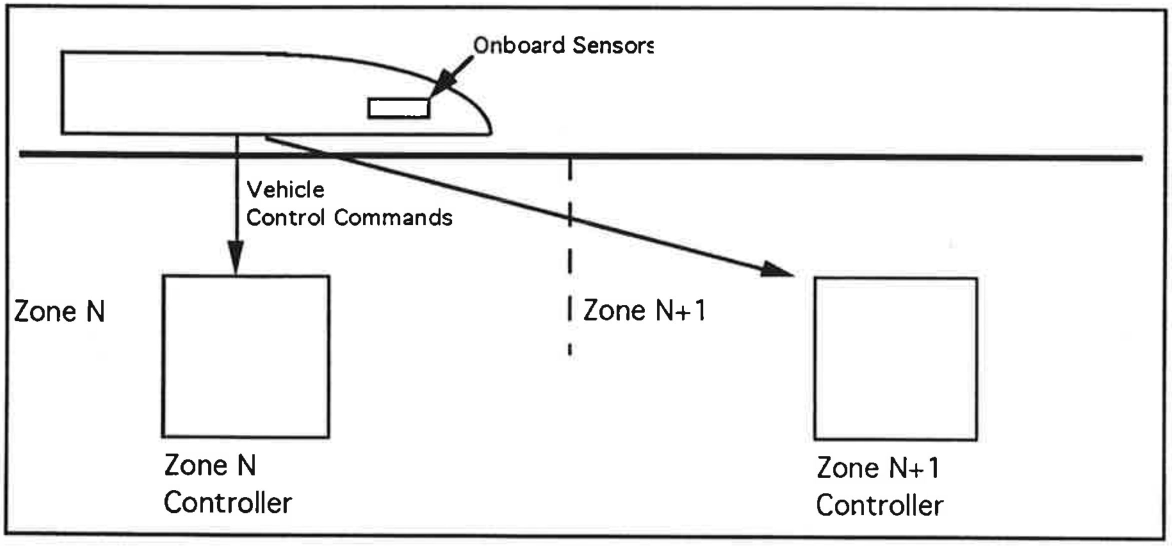


Figure 5-4. Simplified Data Flow Diagram for Smart Vehicle Architecture (SVA).

A factor which bears directly on the initial cost of a system is the cost of hardware replication. For example, since there are many more vehicles in the system than there are wayside stations, it would be more economical to keep the cost of the onboard system to a minimum by keeping its functionality to a minimum. The ZCA architecture offers some initial savings since it transfers several functions to the wayside zone computer. However, the addition of software functions to a system may not directly translate into increased hardware costs. Given the low cost and high throughput offered by current microprocessors, and the anticipated increase in throughput and decrease in cost, the added functionality may add no hardware to a given system, in that the system may not require increased capacity to provide the added service. Since a fully automated system will require onboard computers, this functionality may be provided by the onboard system at no additional cost.

Another significant part of the system cost is the software cost which is directly proportional to the algorithmic complexity of the functions being implemented. This complexity can be reduced in a distributed system by co-locating similar functions in the same computer, thereby reducing the communication complexity and overhead. Thus, algorithmic complexity as well as the required inter-computer communications should be used to evaluate competing architectures.

	Onboard	Wayside	Central
CONTROL			
Vehicle Location	**P**	B	
Velocity Control	**P**	B	
Levitation Control	**P**		
Lateral Position Control	**P**		
Propulsion Control		**P**	
Secondary Suspension Control	**P**		
Route Control	**P**	B	
Vehicle Systems Monitoring	**P**		
Vehicle Systems Control	**P**		
Environmental Monitoring	**P**	B	
PROTECTION			
Safe Vehicle Separation	**P**	B	
Vehicle Position	**P**	B	
Route Integrity	**P**	B	
Emergency Stopping	**P**	B	
Emergency Speed Control	**P**	B	
Emergency Position Control	**P**	B	
Emergency Response			**P**
Failure Management			**P**
SUPERVISION			
Route Planning	I		**P**
Route Scheduling	I		**P**
Dispatching	I		**P**
Maintenance Scheduling			**P**
Operator Interface	**P**		**P**
Status Displays	**P**		**P**
Passenger Supervision	B		**P**

P: primary assignment, B: backup assignment, I: access to information in a database.

Table 5-3. Smart Vehicle Architecture (SVA) Function Assignment.

6. Computer Architecture Synthesis Overview

The next step in the Design and Verification Methodology for Maglev Fault Tolerant Computers shown in Figure 1-1 is to synthesize candidate Maglev Control System Computer Architectures. The inputs are computer requirements, an architecture knowledge base, a building block knowledge base, and technology projections. The computer requirements consist of the RMAS requirements presented in Section 4.5 and the performance requirements presented in Section 6.1. The architecture knowledge base and the building block knowledge base are presented in Section 6.3. Technology projections are presented in Section 6.3.4. Using these inputs, a detailed preliminary design specification for the SVA onboard control computer is presented in Section 6.3.5.

Since the early 1970's, Draper Laboratory has been developing fault tolerant distributed computer architectures suitable for applications which have requirements for ultra-reliability and real-time performance. One of these architectures, known as the Fault Tolerant Parallel Processor (FTPP), has several properties which make it suitable for the Maglev Control Computer System. In addition to ultra-reliability and real-time performance, the FTPP provides performance extensibility and excellent communication interface capabilities. Like other Draper designed fault tolerant computers, it has attributes which make it very desirable for use in complex, safety-critical systems which must be verified and validated to the satisfaction of a Government oversight agency. These include Byzantine resilient fault tolerance, adequate fault and error containment, a simplex programming model, on-line reconfigurability, rigorous separation of redundancy management software and application software, and reliable communication with external devices. Furthermore, these attributes are not dependent on any specific technology of implementation. The FTPP knowledge base provides rules and guidelines for ensuring that the theoretical requirements of Byzantine fault tolerance are met. The building block knowledge base provides a detailed description of the hardware and software components which go into an FTPP.

The process of matching the maglev requirements with the building block capabilities is simplified by considering the performance related requirements and the reliability related requirements as two orthogonal sets, each of which can be mapped independently of the other as a first order approximation. The performance related maglev requirements such as throughput, memory, input/output bandwidth, etc. determine the virtual architecture of the system. The reliability related requirements such as safety and availability determine the physical architecture of the system. The virtual architecture definition includes the number of processing sites, the allocation of control functions among these sites, and the number and type of sensors and actuators and their interconnections to the processing sites. The physical architecture definition includes the redundancy level of each processing site, the redundancy level of sensors and actuators, and the redundancy level of communication interfaces and media.

Two basic architectures for the Maglev Control Computer System were presented in Section 5.4. In the Zone Control Architecture (ZCA) architecture, the primary responsibility for train control rests with the wayside zone control computers, with the onboard system providing backup and consistency checking. In the Smart Vehicle Architecture (SVA), the onboard computer has the primary responsibility for train control, with the wayside zone computers providing backup and consistency checking. In both architectures, the vehicle protection functions are distributed among the three subsystems. This section provides critical analyses of the hardware and software for both the ZCA and SVA.

6.1. A Quantitative Analysis of Maglev Control Functions

The computational requirements of an application are embodied in the scheduled tasks which perform its functions. In order to specify a control computer architecture for the Maglev Transportation System, it is necessary to analyze quantitatively each major function which the system must perform. To perform this analysis, estimates are made of each of the following operating parameters for each major function.

(1) Iteration rate (Hz)
(2) Throughput
(3) Memory requirements (Megabytes)
(4) Input (Kilobits per second)
(5) Output (Kilobits per second)

The results of this analysis are summarized in Table 6-1.

The iteration rate, often referred to as the frame rate, is the frequency at which a control law must be applied to the system being controlled. During each iteration, new sensor readings are obtained, the control law algorithm is executed, and new actuator commands are generated. Some control laws applied to the control of vehicles, especially aircraft, have hard, i.e. inflexible, real-time constraints. These control algorithms are based on the assumption that they will execute in a periodic manner with exactly the same amount of time between the start of each iteration. Any deviation from this periodicity is called jitter. The degree of jitter which can be tolerated varies from application to application. Some control functions do not have demanding real-time requirements. This class of real time applications is able to tolerate jitter. It may be possible to accommodate both classes of control functions within the same computer by allowing low priority applications to execute when time is available and by allowing high priority functions to interrupt the execution of lower priority functions so that they can meet their hard real-time deadlines. Scheduling various control applications so that demanding tasks run on time and complete within their time allotment is a challenging aspect of control system design and implementation. The validation of real-time scheduling algorithms is an especially difficult task. For the maglev control system a wide range of iteration rates and priorities are required.

FUNCTION	**Iteration Rate** (Hz)	**Throughput**	**Memory** MBytes	**Input** KBits/s	**Output** KBits/s
CONTROL					
Vehicle Location (non-GPS)	4-8	<< 1 MIPS	<< 1	~ 2	~ 2
Velocity Control	4-8	<< 1 MIPS	<< 1	~ 4	~ 4
Levitation Control (EMS) (per magnet)	100	<< 1 MIPS	<< 1	10 - 20	2 - 5
Levitation Control (EDS)	1-2	<< 1 MIPS	<< 1	~ 1	~ 1
Lateral Position Control (EMS) (per magnet)	100	<< 1 MIPS	<< 1	10 - 20	2 - 5
Propulsion Control	1000	few instructions	<< 1	~1500	~ 1000
Secondary Suspension Control	200	1.2 MFLOPS	< 1	400	128
Route Control	1	<< 1 MIPS	<< 1	~ 4	~ 4
Vehicle Systems Monitoring	1	<< 1 MIPS	<< 1	~ 1	--
Vehicle Systems Control	1	<< 1 MIPS	<< 1	~ 1	~ 1
Environmental Monitoring	1	<< 1 MIPS	<< 1	~ 1	--
PROTECTION					
Safe Vehicle Separation	1 - 100	< 1 MIPS	< 1	1 - 10	<< 1
Vehicle Position	4 - 8	< 1 MIPS	1 - 2	< 1	<< 1
Route Integrity	1 - 1000	1 - 400 MIPS	1 - 2	1-10,000	1 - 2
Emergency Stopping	on demand	< 1 MIPS*	<< 1	~ 4	~ 4
Emergency Speed Control	on demand	< 1 MIPS*	<< 1	~ 4	~ 4
Emergency Position Control	on demand	< 1 MIPS*	<< 1	~ 4	~ 4
SUPERVISION					
Route Planning	not real time	10 - 20 MIPS	10-40	--	--
Route Scheduling	not real time	10 - 20 MIPS	10-40	--	--
Dispatching	1-10	1 - 5 MIPS	1 - 5	10 - 20	10 - 20
Maintenance Scheduling	not real time	10 - 20 MIPS	10 - 40	--	--
Operator Interface	6-10	< 1 MIPS	1 - 4	< 1	--
Status Displays	5-10	< 1 MIPS	4 - 8	1 - 1000	1 - 1000
Passenger Supervision	1	<< 1 MIPS	1 - 2	1 - 10	1 - 10
Emergency Response	on demand	2-4 MIPS*	1 - 4	1 - 1000	1 - 1000
Failure Management	on demand	< 1 MIPS*	1 - 4	1 - 1000	1 - 1000

Table 6-1. Quantitative Analysis of Primary Maglev Control Functions

The throughput of a computer is the number of instructions which it can execute in a second, often measured in MIPS (millions of instructions per second). The throughput requirement of a task is calculated by using two different methods, with the larger result prevailing. In the first method, the iteration frequency in Hertz is multiplied by the maximum number of instructions which the task could execute during an iteration. In the second method, the maximum number of required instructions per iteration is divided by the maximum allowed processing lag. Processing lag, also called transport lag, is the time interval between the time a sensor, which provides a value to the control law, is read and the time an output value is delivered to an actuator. An application which only uses a few instructions per iteration most likely has a throughput requirement dominated by the size of its processing lag. On the other hand, the first method probably generates a more demanding throughput requirement for an application which requires many computations, i.e. instructions, per iteration.

Throughput, however, is a very coarse measurement to use when sizing a computing platform for a given application. In the first place, internal machine architectures play a significant role in the magnitude of these numbers. Reduced Instruction Set Computers (RISC) typically have very high values of MIPS compared to CISC machines (Complex Instruction Set Computers). However, a single CISC instruction may incorporate many RISC instructions. In the second place, on a given machine, different instructions take different amounts of time to execute. Hence, most manufacturers provide an instruction-mix throughput value which is probably favorable to their machine. Various instruction-mix benchmarks can be used to measure the throughput of a given machine, such as the Digital Avionics Instruction Set (DAIS), Whetstones, Dhrystones,VAX Units Per Second (VUPS), Specmarks, etc., however, these may not accurately reflect the mix in a given application. A more significant assessment of the actual processing lag can be obtained as a design proceeds by benchmarking a given application using a specific programming language, compiler, operating system and processor. Furthermore, it is customary to require an additional margin of throughput from a computing platform to allow enhanced capabilities in a given function to be easily accommodated and also because it is often difficult to obtain accurate quantitative measurements about an application early in the development cycle. The values of required throughput for the various maglev functions presented in Table 6-1 represent generous estimates of various experts working on specific functional areas or from comparisons to equivalent avionic applications and may be used at this stage for preliminary gross system sizing.

The memory requirement of a given function is the amount of RAM needed to perform its calculation. Generally, much smaller amounts of ROM are needed. Applications such as signal processing, which handle large arrays of data, require significant amounts of memory. However, since reliable memory devices in the megabyte range are economically available, these requirements can easily be accommodated. Memory usage is difficult to estimate before a software module exists. At that time, measurements can be taken and more accurate requirements can be established. Furthermore, it is important to allow a generous margin of additional memory to allow for function growth and enhancement in a system. The maglev functions have a wide range of memory requirements. The entries in Table 6-1 are conservative, i.e. generous, estimates garnered from experts working on the various functions or from comparisons to equivalent avionic applications.

Input bandwidth is the number of bits transferred per second from an input device, such as a sensor, to a recipient task. Similarly, the output bandwidth is the number of bits per second transferred from a control function task to an output device, such as an actuator. Since neither transfer can take place instantaneously, each form of communication has an associated latency. Typically, the processor and its sensors and actuators communicate with each other over a network or a bus with a fixed protocol and communication bandwidth. The bandwidth of the communication system must, in general, be much larger than the bandwidth required for any given task because it must be able to service all of its subscribers, have extra capacity for the communication overhead needed to handle protocols and error detection schemes, plus have a margin for growth. The values of input and output bandwidth shown in Table 6-1 for maglev is a summation of values representing the total bandwidth between a given function and a set of possible I/O devices.

Typically, a processor interfaces with sensors and actuators by means of analog-to-digital (A/D) and digital-to-analog (D/A) converters, synchro-to-digital (S/D) and digital-to-synchro (D/S) converters, and I/O controllers, which are micro-programmed interface devices. To obtain greater accuracy, the sampling rate of a sensor may be significantly higher than the rate at which the processor reads the sensor value from the A/D converter. The input latency is the time interval between the sampling of a physical phenomenon by a sensor and the delivery of the data from that sensor reading to the control task which uses it. Output latency is the time interval between the generation of an actuator command by a control task and the delivery of that data to an output device. The execution latency is the time it takes the control task to compute the actuator command, once it has all of its sensor input. The sum of the input, execution and output latencies is equal to the transport lag.

It should be noted that in order to produce a detailed design for the maglev control system, more detailed information than that presented in Table 6-1 is necessary. For example, from the preceding discussion, it is clear that this table does not include values for transport lag, I/O latencies, and execution throughput and communication bandwidth margins. Furthermore, the table also omits bandwidth values for intertask communication, which require a more detailed software specification than is presented here. Finally, the table does not present information on specific scheduling priorities, precedence and dependency relationships between functions, or pre-emption constraints, all of which are needed for a detailed hardware and software design. However, the data is quite sufficient for an initial system specification. This specification would be refined as detailed functional requirements became available.

6.2. Data Flow Analysis of Maglev Control Functions

In order to better understand the interactions among the distributed computing sites, a high level data flow analysis has been performed on the SVA. Figure 6-1 shows the data that would flow between the three major computation sites: the onboard computer, the wayside zone control computer and the central computer. The data which passes between the onboard system and the wayside is very time critical, since this controls the speed of the vehicle. There can be very little latency in this data flow. The communication between consecutive vehicles on the guideway is also very time critical, especially for the short headways (36 seconds) specified in the Concept of Operations. However, the data passing between the central facility and either the onboard or wayside computers is less time critical. Figure 6-2 shows a lower level data flow diagram for the onboard control computer and its associated sensors and actuators.

6.3. Maglev Fault Tolerant Control System Architecture Building Blocks

6.3.1. FTPP Overview

The architecture of the Fault Tolerant Parallel Processor (FTPP) developed by Draper Laboratory was conceived to serve applications with requirements for ultra-high reliability and real-time performance. To satisfy the first requirement, the FTPP is designed to be resilient to Byzantine faults. This terminology is defined below. To satisfy high throughput requirements, the architecture contains many processing sites to provide parallel processing capability. The FTPP architecture is described in references [25], [26], and [27].

The FTPP is designed to provide characteristics which are highly desirable for safety-critical, real-time applications such as the maglev control computer system. Attributes which make it very desirable for use in systems which must be verified and validated to the satisfaction of a Government oversight agency include Byzantine resilient fault tolerance, adequate fault and error containment, on-line reconfigurability, rigorous separation of redundancy management software and application software, and reliable communication with external devices. In addition to ultra-reliability and real-time performance, the FTPP is flexible, easily extensible, and highly programmable. Furthermore, these attributes are not dependent on any specific technology of implementation.

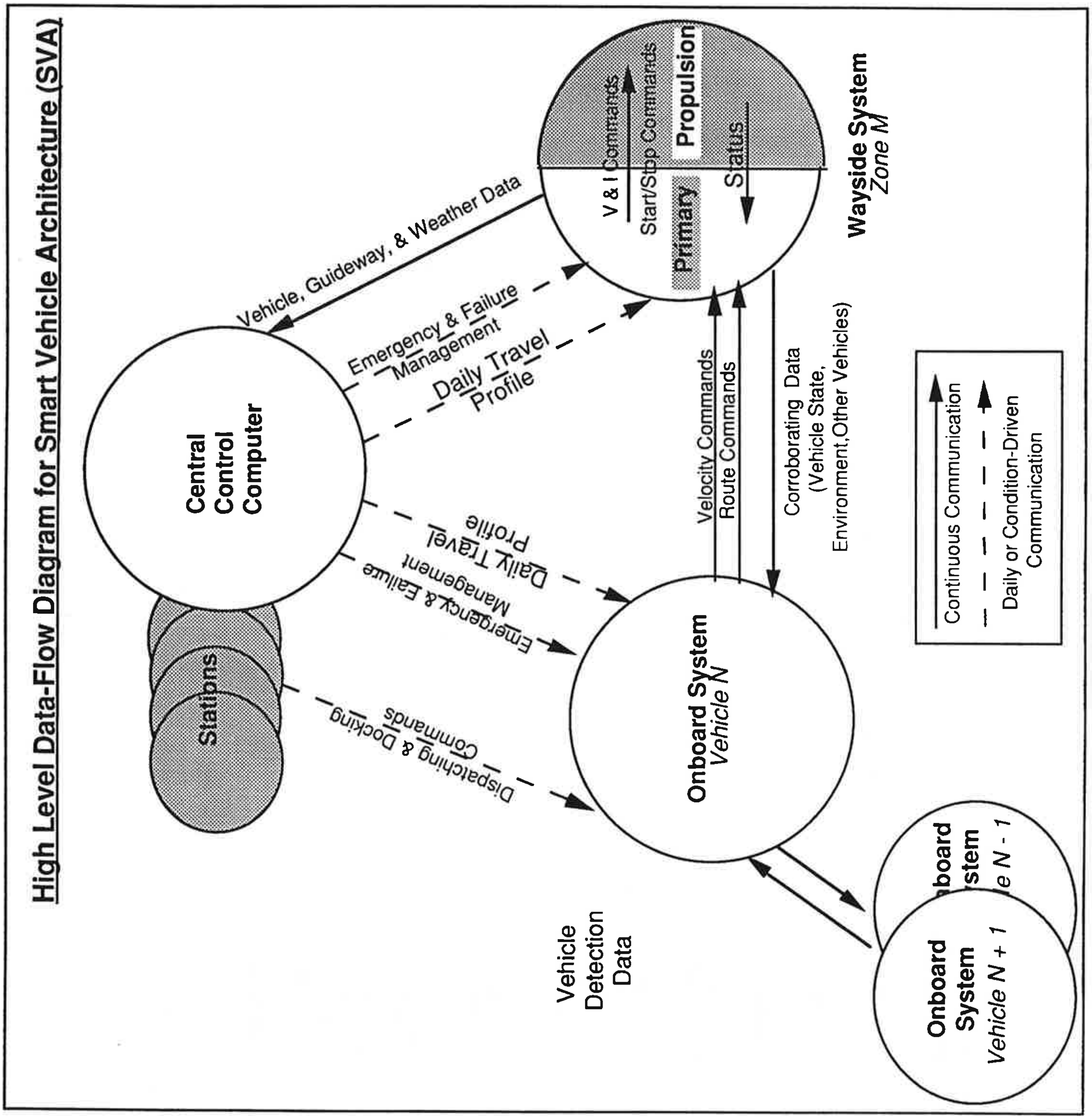


Figure 6-1. Data Flow Diagram for Smart Vehicle Architecture.

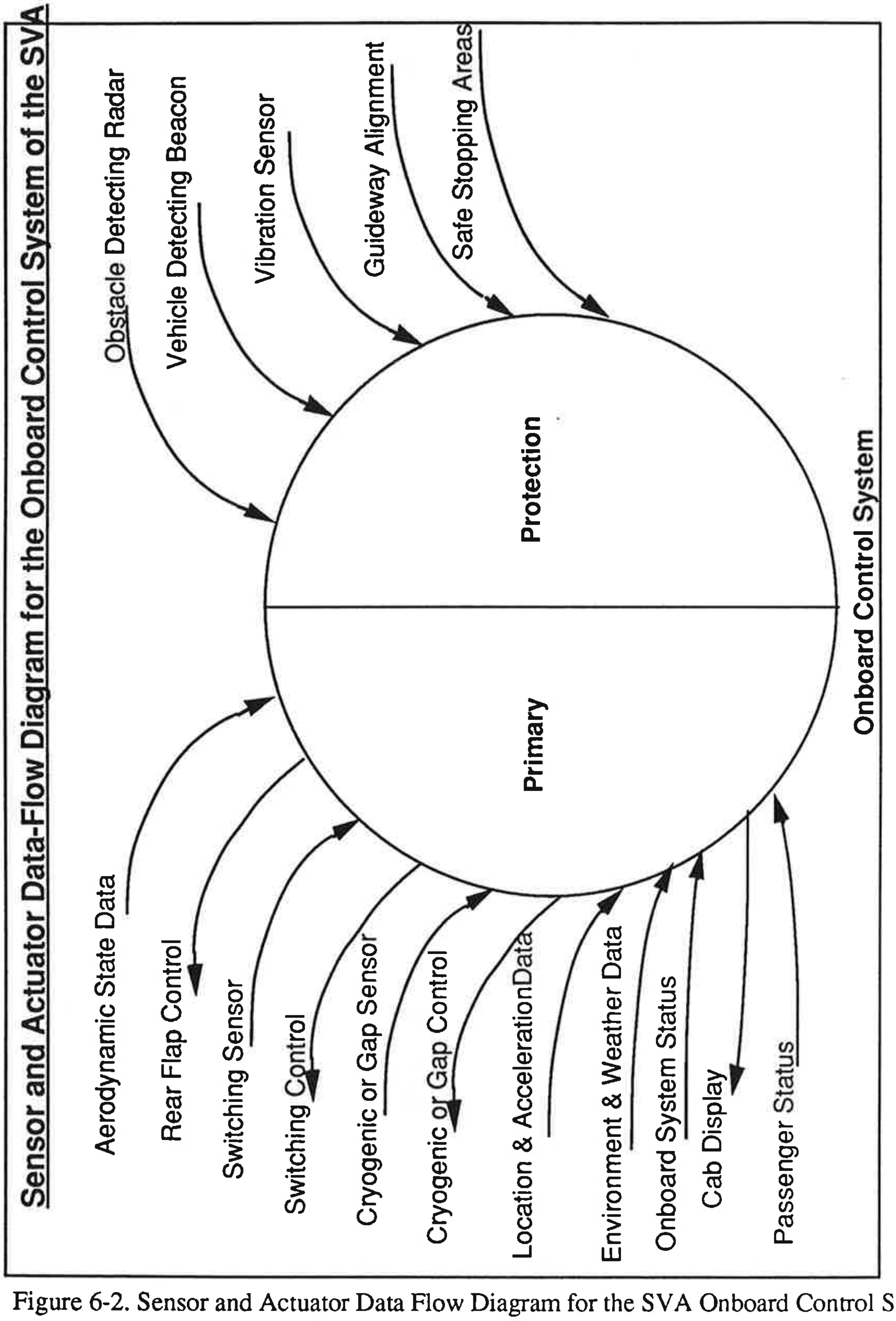


Figure 6-2. Sensor and Actuator Data Flow Diagram for the SVA Onboard Control System.

The FTPP is extremely reliable. It is capable of providing a system loss probability as low as $1X10^{-10}$ per hour. Furthermore, its reliability is not based on specific models of component failure behavior. Instead, it has been designed in strict accordance with the theoretical principles of fault tolerant computing and is therefore able to provide coverage for arbitrary faults in the computing mechanism, the communications mechanism and the clocking mechanism. In addition, a three-pronged approach of fault avoidance, fault removal, and real-time fault detection and recovery has been taken to reduce the probability of common-mode failures in the FTPP due to software faults, hardware design faults and numerous other sources of common-mode failures [30,46].

The FTPP is a highly available system. This property is due to the ability of the system to replace failed components in real-time with existing spares and to differentiate between transient and permanent faults so that the supply of spares is not depleted rapidly.

Due its parallel processing architecture and flexible communication interface, the FTPP can provide very high computational and communication performance. For example, an existing prototype system can provide a computational throughput of over 200 MIPS and an input/output communication bandwidth of 100 Megabits per second. Furthermore, these values do not represent upper bounds, but serve to show a nominal, easily achievable capability.

From a cost viewpoint, the FTPP provides its high levels of dependability and performance efficiently. This is attributable to the fact that the computational overhead for operations which provide fault tolerance is very low since these are primarily implemented in hardware and the high bandwidth fault tolerant communication network is shared by many processing sites. Furthermore, since the FTPP supports mixed levels of redundancy, a given application can apportion more critical functions to sites with more redundancy and less critical ones to sites with less redundancy, thereby fostering efficient allocation of system resources.

Of importance to a system which must be certified for operation by an oversight agency, such as the FRA in the case of maglev, is the ease with which it can be validated. Validation is the process whereby a system demonstrates that it meets all the requirements imposed upon it by the task it is charged to perform. This process is greatly facilitated by the architecture of the FTPP which adheres strictly to theoretical requirements of Byzantine resilience. This confers on a system the ability to tolerate a certain number of arbitrary random hardware faults. The architectural validation process then becomes simply a straightforward demonstration that the FTPP meets these theoretical requirements rather than a painstaking and ultimately futile attempt to demonstrate that every possible combination of hardware faults is covered by some heuristic approach. Furthermore, the strict separation of redundancy management and application software make the validation process more tractable on an FTPP because it can be divided into separate architectural, implementation and application specific analyses.

The clear separation of redundancy management functions from application functions which make systems developed on the FTPP more readily validatable also render the system more readily programmable. The fault tolerant operation of the FTPP is completely transparent to an applications engineer who is then free to develop applications software without having to deal with the cognitive overhead of redundancy management of the underlying compuational platform.

Finally, the FTPP architecture provides a high degree of flexibility, extensibility and upgradability to an application. During an initial design, flexibility may be the most important of these related yet distinct characteristics. Because the FTPP is a highly reconfigurable system, it is easily adaptable to a wide range of dependability and performance requirements. Tradeoffs among cost, throughput and reliability are relatively easy to analyze. To provide extensibility later in the system life cycle, the fault tolerant hardware is specifically designed to accommodate growth. Furthermore, since there is provision for mutual interconnection among groups of FTPPs, there is no absolute upper limit to the extensibility of an FTPP-based system. Upgradability to components fabricated with advanced technology is facilitated by the use of both standard interfaces to the specialized fault tolerant hardware and computing sites built from off-the-shelf hardware and software.

6.3.1.1. FTPP Virtual Architecture

The programming model of the FTPP used by the applications engineer is that of a standard uni- or multi-processing architecture, using familiar operating system calls and constructs. As seen by the applications programmer, the FTPP supports a virtual architecture of a number of computing tasks which may execute in parallel, subject to preemption and data and control flow dependencies. When the underlying operating system is implemented in the Ada® programming language, the programmer's model is that of a number of communicating Ada tasks, as shown in Figure 6-3. The tasks communicate using message passing. Applications are constructed by the applications engineers without regard for the parallel and redundant nature of the underlying system. Moreover, when redundant sites are used, parallel applications can be developed under the greatly simplifying assumption that all processing sites are reliable. Thus there is no need to consider the possible effects of faulty component behavior upon an algorithm.

The high degree of programmability exhibited by the FTPP is due in part to the fact that its fault tolerant behavior is transparent to an application. This is in marked contrast to the operation of many other fault tolerant systems. The programmability of the FTPP translates into greater software productivity for application engineers. It also contributes to the final verification and validation of the system since the application software and the fault tolerance attributes can be verified and validated independently.

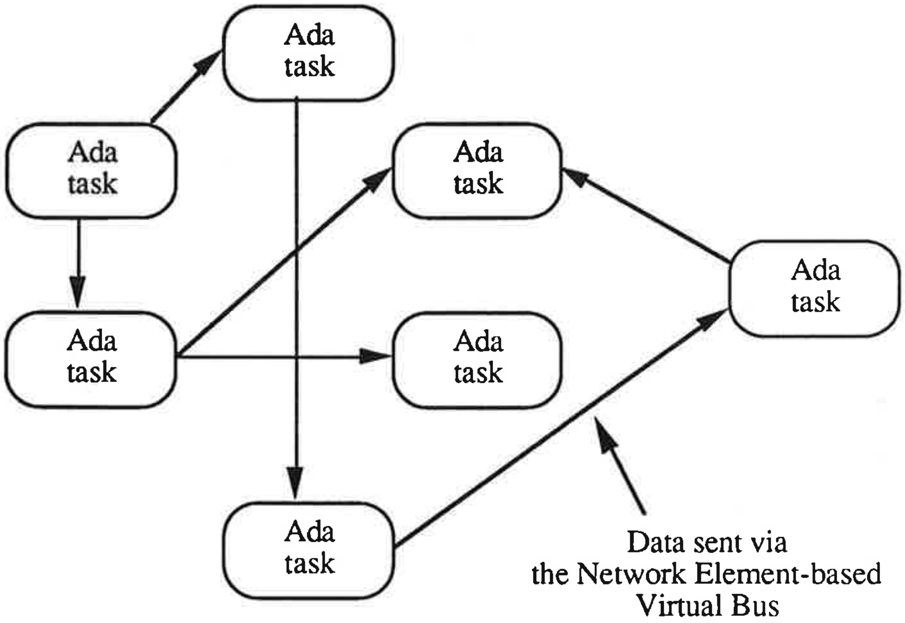


Figure 6-3. FTPP Programming Model for the Ada® Programming Language.

6.3.1.2. Approach to Fault Tolerance

The reliability requirements of systems whose failures could result in the loss of human life are very demanding; the probability of surviving such a failure must approach unity. Typically, these are systems which cannot tolerate any errors in the computer system outputs, because such errors can result in irreversible behavior by the system being controlled. These computer systems are used to control modern aircraft, spacecraft, and weapons systems, to monitor nuclear power plants, and to direct various medical procedures. The control of maglev vehicles is another such life-critical application.

The traditional approach to the design of a reliable system is to perform a Failure Modes and Effects Analysis (FMEA). Using this approach, a supposedly exhaustive list of likely failure modes of components is compiled. An estimate of the extent and effects of each component failure mode is predicted. For each likely failure, a fault tolerance technique is devised. This is clearly an impossible task for digital computers due to the extremely large number of possible component failures and combinations that must be analyzed.

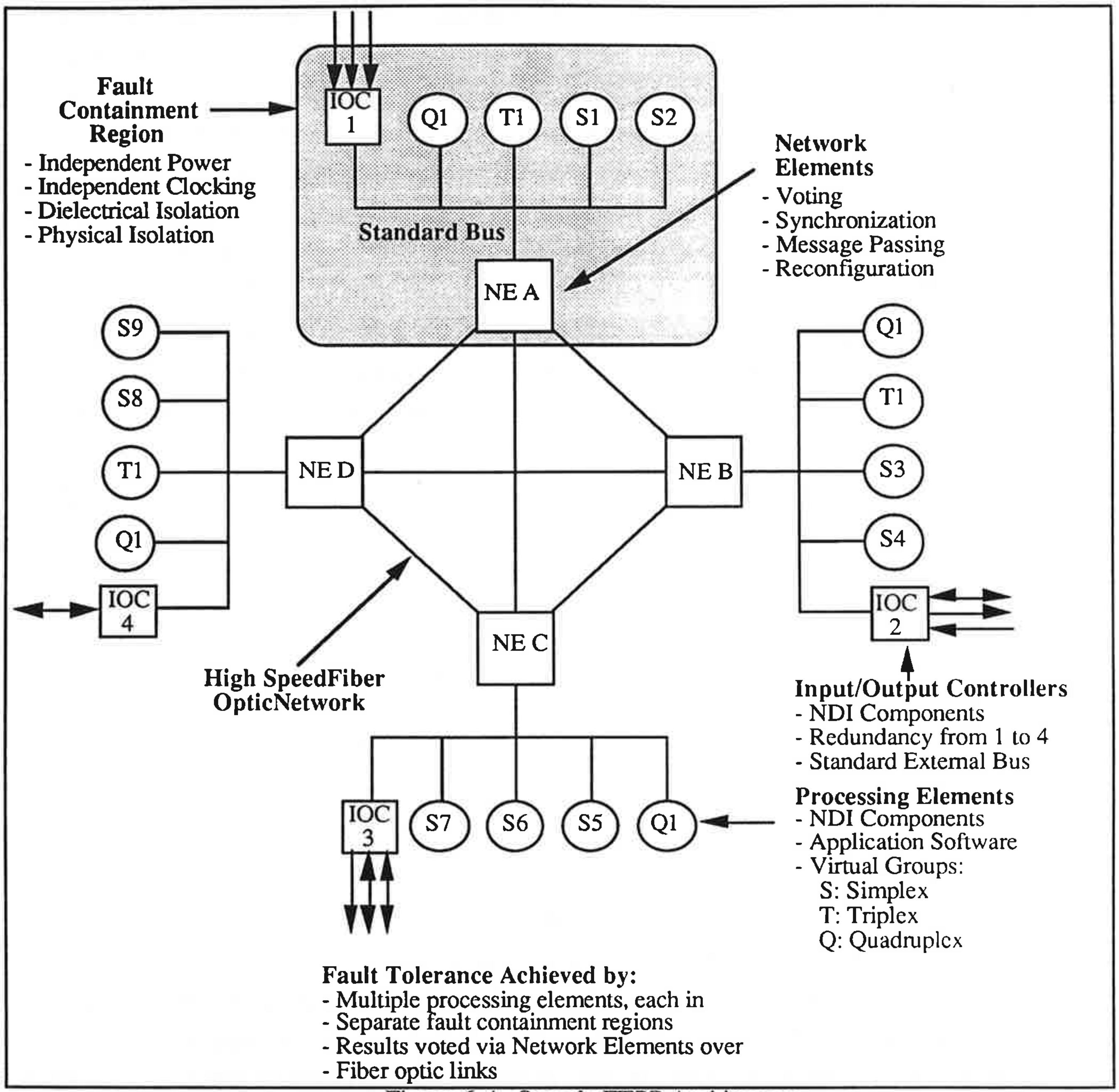


Figure 6-4. Sample FTPP Architecture .

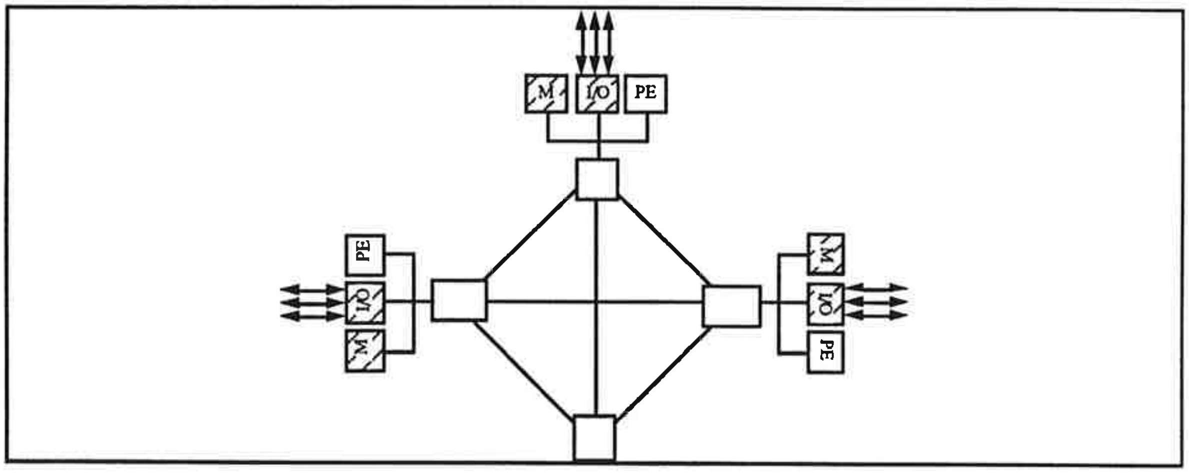


Figure 6-5. Minimal FTPP Configuration.

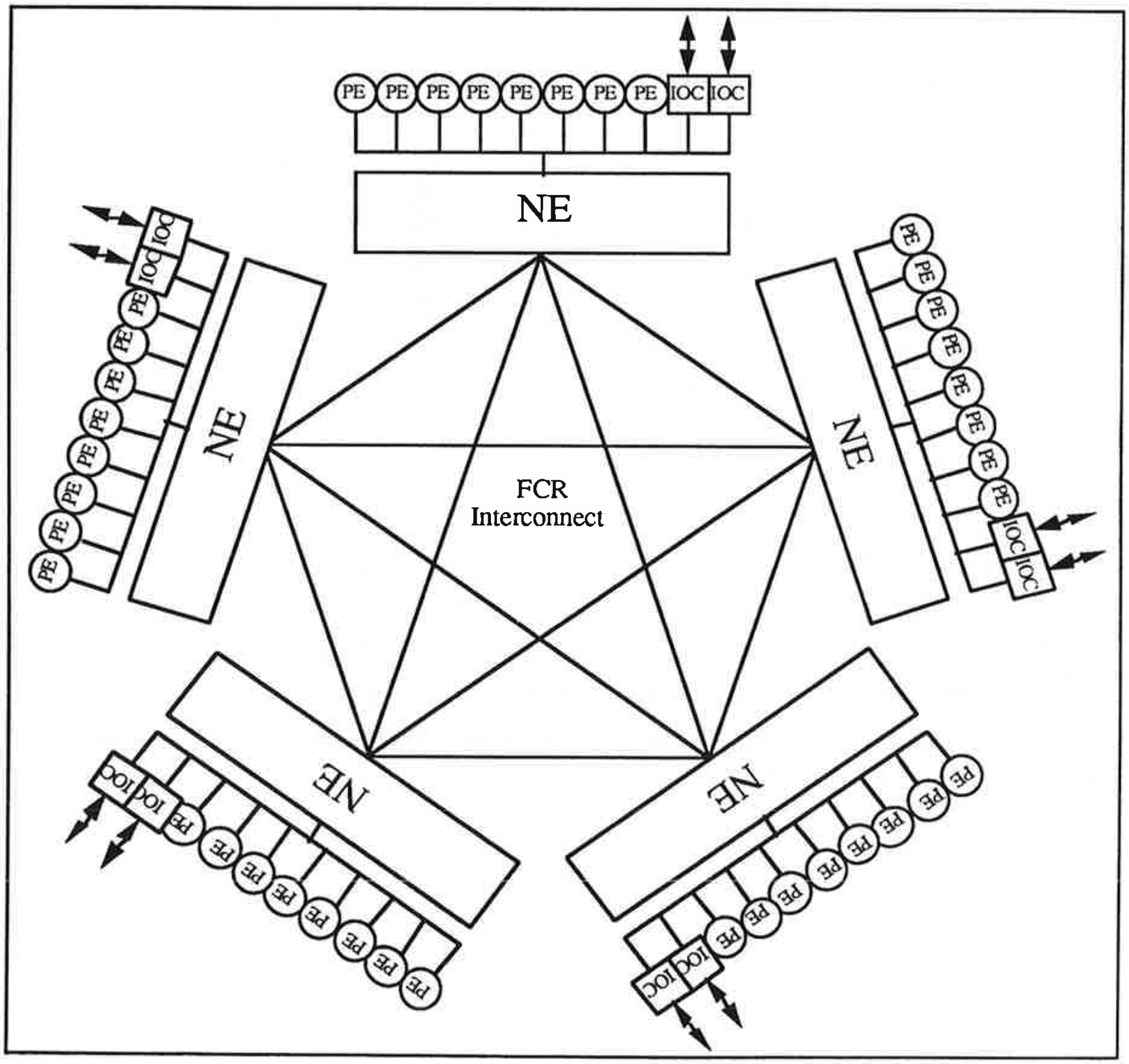


Figure 6-6. Full 40-PE Cluster.

6.3.2.1. System Designer's View of the FTPP

As discussed in Section 6.3.1.1, the FTPP's redundant nature is hidden from the view of the applications programmer. However, the system designer who is configuring an FTPP as the computing platform for a given application is very much concerned with this view, especially as regards the required safety, reliability, and availability of the system at hand. The FTPP's virtual bus topology showing several example virtual processors is shown in Figure 6-7. From this perspective a system designer can use the quantitative performance and reliability requirements of the application to determine the number of required processing sites, the redundancy level of each site, and the number of any necessary spare simplex processors. Furthermore, the bandwidth of the Network Element Virtual Bus can be considered in assigning tasks to processors. Any of the VPs can be used as Input/Output Controllers with access to external devices, such as sensors and actuators.

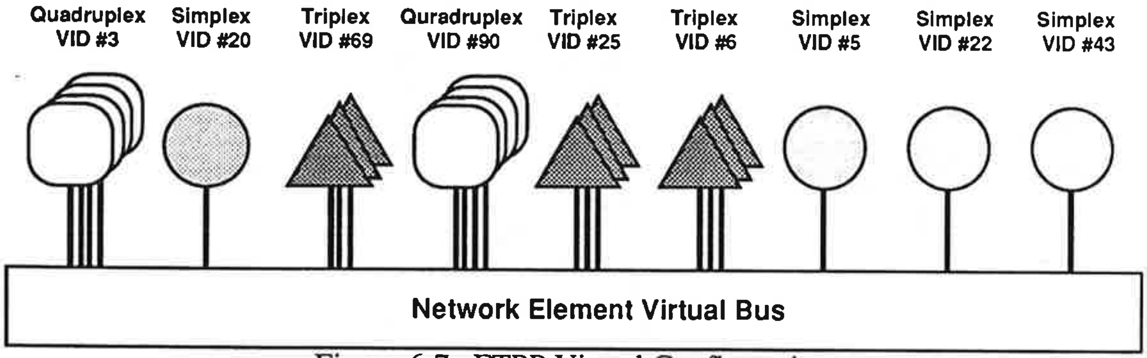


Figure 6-7. FTPP Virtual Configuration

6.3.2.2. Interprocessor Communication on the FTPP

A major drawback of many parallel processing systems is the inability to know with certainty that data sent from one processor is delivered in a timely and orderly manner to another processor. The FTPP is designed to meet the requirements, established by rigorous mathematical theory, which eliminate this uncertainty associated with interprocessor communication. The message passing properties of the FTPP are concisely enumerated in Table 6-3. This is a guarantee made to the applications programmer on interprocessor message ordering and validity which holds in the presence of Byzantine faults, and relieves the programmer from consideration of faulty behavior when designing a distributed application. It should be noted that these guarantees on totally ordered and timely delivery are not typically made by parallel processors even in the absence of faults.

1.	Messages sent by non-faulty, redundant members of a fault tolerant processor are correctly delivered, bit-for-bit, to the non-faulty members of all destination processors.
2.	Non-faulty, redundant members of a fault tolerant processor receive messages in the order sent by the non-faulty, redundant members of the sender.
3.	Non-faulty, redundant members of a fault tolerant processor receive messages in identical order.
4.	The absolute times of arrival of corresponding messages at the redundant members of receiving fault tolerant processors differ by a known upper bound.

Table 6-3. Rules of Operation for FTPP Interprocessor Communication.

6.3.3. Operating System Architecture

As with most complex computing systems, the FTPP is best viewed as a layered system, where each layer highlights a different aspect of the system. As seen in Figure 6-8, the top layer consists of applications programs. The middle layer consists of the FTPP System Services. Certain services are visible and may be invoked by the applications programmer; these include input/output calls, task scheduling, and intertask communication services. This layer is intended to mask the complexity of the FTPP's lower layers from the programmer, making the system functions for fault tolerance, such as voting, interactive consistency protocols, and fault detection and masking transparent to the user. Other important functions of the FTPP Operating System (OS) are not directly accessible by the applications programmer and are performed in a manner which is transparent to the application. These include the delivery of intertask messages to both local and remote tasks, the synchronization of redundant PEs, preemptive scheduling functions, input/output functions, such as periodic sensor readings and the disassembly and re-assembly of long messages into packet format, Built-In Testing (BIT) of the hardware, fault logging, and fielding software exceptions.

The bottom layer performs such functions as fault detection, identification, and recovery (FDIR); configuration of the parallel resources into redundant computing sites; and interfacing to the interprocessor communication network hardware. The application tasks and FTPP System Services/OS execute on Processing Elements, as indicated in Figure 6-8.

The foundation of the operating system for the FTPP consists of a vendor-supplied Ada Run-Time System and Draper-supplied extensions based on recommendations made by the Ada Run-Time Environment Working Group (ARTEWG). Additional features are required to manage the plurality of FTPP resources in a manner appropriate to the mission requirements. Services are provided in the following areas:

- Task management, i.e., task identification, scheduling, dispatching, suspension and termination

- Inter-task and interprocessor communication
- Communication with external input/output devices
- Management of parallel and redundant resources according to specified mission reliability, availability, and performance objectives
- Fault detection, identification, transient discrimination, and recovery throughout the maintenance and operational modes of the system

Additional services include software exception handling and time services, i.e., initialization of date and time from an external source, dissemination of current time and date, manipulation of times, and conversion of the internal time representation to character format.

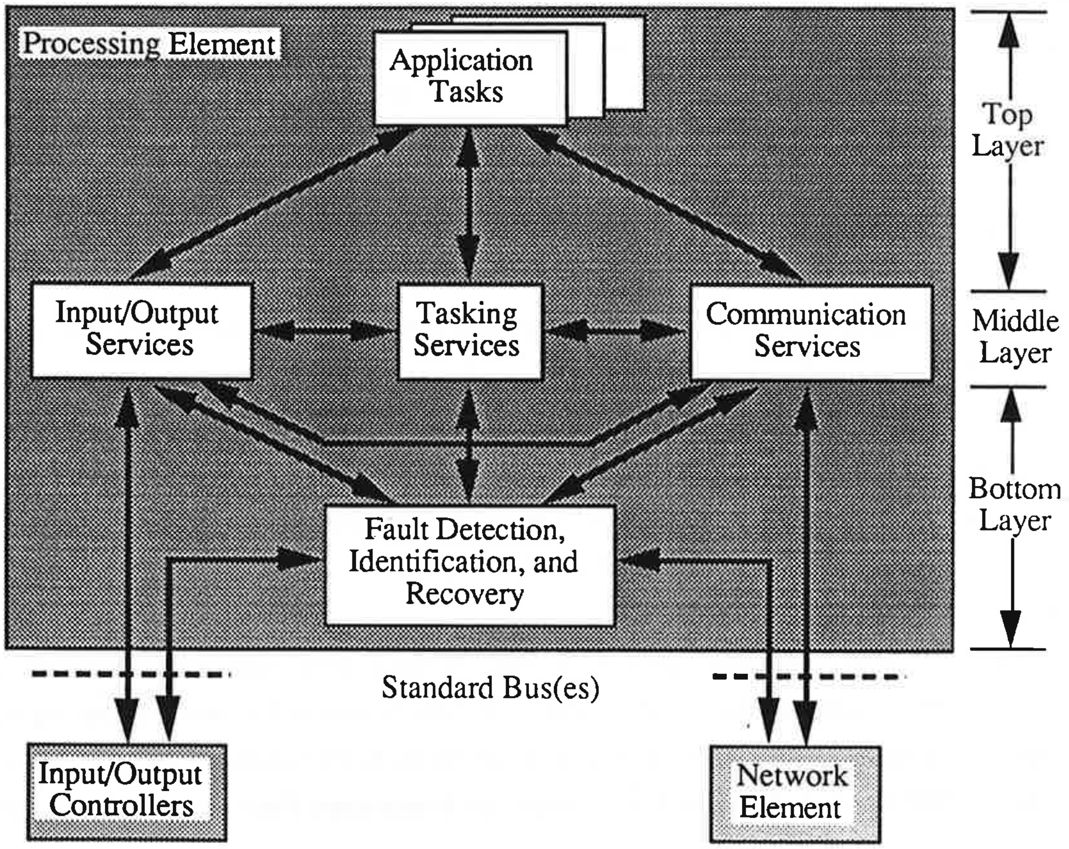


Figure 6-8. Hierarchical View of FTPP Functional Layers.

6.3.3.1. Scheduling

The FTPP supports two different styles of scheduling, each of which is suited for different application domains. The first, known as rate group scheduling, is suitable for task suites in which each task has a well-defined iteration rate and can be validated to have an execution time which is guaranteed to not exceed its iteration frame period(the inverse of its iteration rate). It is particularly useful for control applications which require well defined intervals between consecutive applications of the control law as well as between sensor readings and actuator commands. The second style of scheduling, known as aperiodic scheduling, is necessary when the iteration rate of a particular task is undefined or unimportant. Validation of the temporal behavior of such tasks may be difficult. The FTPP supports task suites consisting of a mixture of rate group scheduling and aperiodic scheduling. Aperiodic tasks are not allowed to perturb the critical timing behavior of rate group tasks.

6.3.3.1.1. Rate Group Scheduling

The basic executive of the FTPP is a Commercial off-the-shelf (COTS) Ada kernel enhanced by Draper. One of these enhancements is a multi-rate group scheduler layered upon the Ada run time executive. In such a paradigm tasks executing on each VP are characterized by an iteration rate, which is typical of most control applications. The frequencies and number of rate group frames are readily changed as the application dictates. Frames executing on different VPs need have no particular phase relationship with each other and may be of different periods. However, on a given VP, frame periods are multiples of the period of the highest frequency group. Within a particular rate group frame, tasks of that rate group are executed using a non-preemptive static schedule.

To achieve multi-rate group execution on a VP, lower frequency rate group tasks are interrupted on a periodic basis to allow the higher-frequency rate groups to execute. The interruption process is transparent to the application programmer. For example, the secondary suspension system of a maglev vehicle uses a periodic control law with a frequency of 200 Hz to improve the passenger ride quality. Control of the linear synchronous motor is also a periodic control application.

6.3.3.1.2. Aperiodic Scheduling

Aperiodic tasks are executed after all rate group tasks have been executed. These tasks may be scheduled in either a round-robin fashion or in a preemptive priority-based fashion. Several non-rate group tasks may run concurrently on different members of a VP, and a given VP may execute aperiodic tasks exclusively by assigning that VP no rate group tasks to execute. For maglev, monitoring of several non-safety-critical onboard systems may be handled as aperiodic tasks.

6.3.4. Redundancy Management on the FTPP

The physical architecture of the FTPP supports a very powerful redundancy management scheme called parallel-hybrid redundancy, which is a combination of both static and dynamic redundancy techniques. Static redundancy provides the ability to mask faults instantaneously as they occur. For maglev, this means that a failed processor cannot provide incorrect commands to the propulsion system or to an aerodynamically controlled secondary suspension system, since the incorrect outputs of the faulty processor are masked by the correctly functioning majority members of its VP. Dynamic redundancy refers to the ability of the system to reconfigure itself in response to failures. Without dynamic redundancy, the reliability of the system would degrade as faulty components accumulate. Dynamic redundancy allows the reliability of the FTPP to be restored following a failure. For example, for the configuration shown in Figure 6-4, if the quadruplex VP labeled Q1 lost the channel from Network Element A, the spare processor labeled S1 could be used as a replacement. This spare capacity increases the availability of the system, especially important for a transportation mode like maglev whose market depends on timely service not interrupted by unexpected failüres, by allowing repairs to be deferred to the normally scheduled maintenance period.

Parallel-hybrid redundancy is a redundancy management scheme that is ideally suited to the Maglev Control Computer System. Like fly-by-wire avionics applications, the maglev system performs many control functions which cannot tolerate an erroneous output command to an actuator. Hence, fault masking at the outputs in real time, as provided by the FTPP, is absolutely essential. There is no performance penalty to be paid for this type of protection, since it is part of the normal behavior of the FTPP. However, reconfiguration does require some additional computing overhead. Therefore, during certain modes of operation requiring high throughput, it may not be desirable to reconfigure the system. However, a maglev vehicle frequently changes modes of operation as it moves from a station to a high-speed section of the guideway. Similarly, the wayside zone controllers also have frequent operational mode changes as vehicles enter and leave their zones. During less critical operational modes, e.g. when a vehicle is stopped in a terminal or when a zone is idle, it is possible for the system to undergo a reconfiguration, by bringing in a spare PE to replace a failed PE. This reconfiguration is effected automatically, i.e. without the need for the intervention of a maintenance crew. Typically the reconfiguration process is accomplished in less than one second.

The reliability and availability of an FTPP implementation is clearly dependent on the fault recovery options used. In addition to instantaneous fault masking and a reconfiguration which replaces a faulty component with a spare, another recovery option which can be used as an intermediate recovery strategy is graceful degradation. This recovery method eliminates the slight overhead penalty paid in the NE message passing protocol by allowing a failed member of a VP to remain online. In this case, a faulty component in a redundant VP or an NE is immediately disabled upon detection, with no lengthy fault recovery or reconfiguration attempted. No effort is made to discriminate between transient and permanent faults for the purpose of performing on-line recovery, in effect treating all faults as permanent until a more relaxed operational regime is entered. This option has the advantage of taking only a very small amount of time to effect and therefore has the least impact on normal system operation. The net effect of a recovery made by graceful degradation is that the redundancy level of a VP is reduced or degraded. Thus a quadruplex becomes a triplex, and so on. The main drawback with this approach is that resources are depleted more quickly. Since transient component failure rates usually predominate over permanent rates, this recovery method is most useful during short missions and critical operational modes, especially when these are interspersed with less critical operational modes which allow retesting of components which produced errors. Thus, if FTPP onboard a maglev vehicle undergoes a graceful degradation to recover from a fault which occurred during a normal trip between two stations, a permanent loss of resources need not result. Once safely in the station, the system can perform extensive self-diagnosis and restore a component whose fault is deemed transient, thereby restoring system resources and further enhancing its maintainability.

6.4. Expected Performance

The expected performance of the FTPP is expressed in terms of the "delivered throughput." The delivered throughput made available to the application programmer is equal to the raw throughput of the PE in the FTPP minus the temporal overhead due to the real-time operating system and the redundancy management software. Based on prior Draper experience, a conservative figure of 10% for each is typical. The delivered throughput is in turn a function of the redundancy level into which the PEs are grouped. For example, a configuration in which all processors are operated in a non-fault tolerant, simplex mode is roughly four times that when the processors are operated in a fault tolerant, quadruplex mode. Note that the formulation presented herein represents an upper bound on the throughput obtainable by a parallelized application which maps to the parallel hardware with 100% efficiency. In most cases mapping and intertask communication inefficiencies and irreducible serial tasks will reduce the speedup obtained by a given application.

The raw throughput is obtained from manufacturer's literature or actual benchmarks. From the Motorola literature, the throughput of the 68040 processors is 20 MIPS. If the horizontal axis of Figure 6-9 were re-labelled as Number of Fault Tolerant Virtual Processors, then the throughput of the system can be obtained from the graph for simplexes.

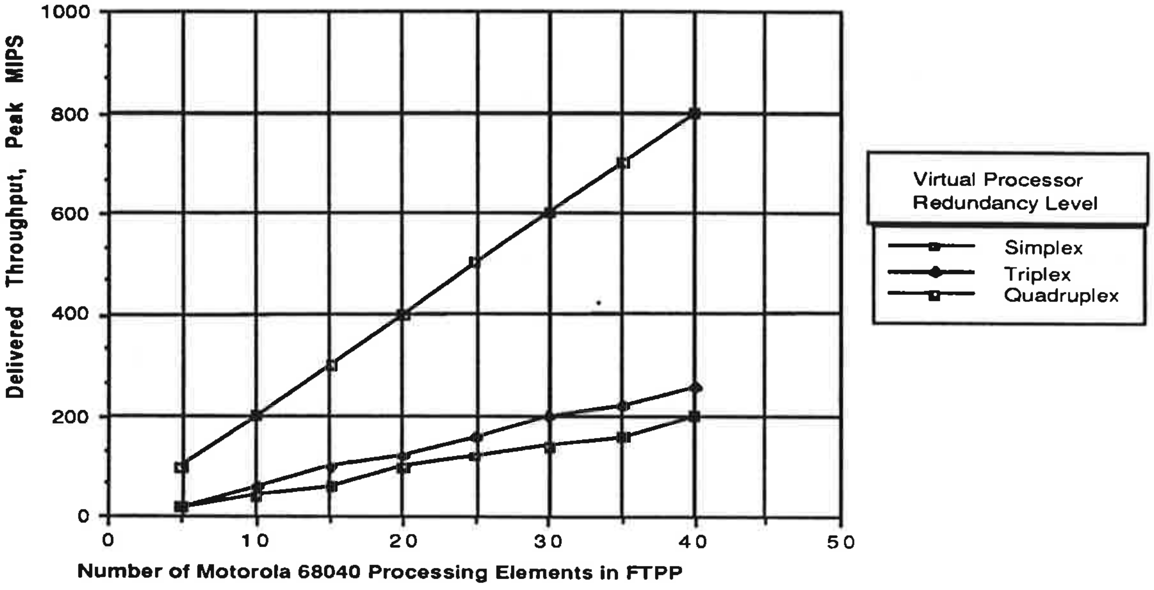


Figure 6-9. Expected Performance of Motorola 68040-based FTPP.

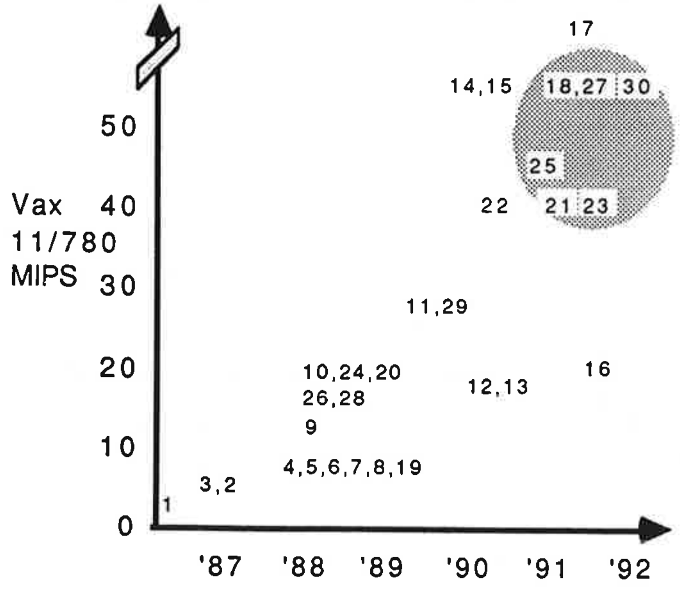
6.5. Technology Projections [38,31]

Since the anticipated development time of a prototype Maglev system will not take place until sometime after 1996, it is important to take note of projected advances in technologies relevant to computing platforms which can benefit the design of the Maglev control computer system. Of particular interest are technology advances which can increase the computational throughput, reliability, and functional density of computer systems. Aside from their own intrinsic value, these advances also reduce the cost of the computer system.

Advances in three technologies can favorably benefit the implementation characteristics of the FTPP architecture. These are VLSI, microprocessor, and packaging technology. Improvements in VLSI will favorably impact the performance characteristics of the NE in terms of increased speed due to higher clock rates, reduced weight and volume due to higher functional density, and more reliability due to improved design tools and increased integration. Clock rates have been increasing at approximately 10 MegaHertz per year since 1988. If this trend continues at only one third the present rate, VLSI devices will be available by 1996 with clocking frequencies in the 70 to 100 MHz range [29]. Greater functional densities are being achieved by reducing feature sizes and by increasing the number of devices which can be incorporated into a single Application Specific Integrated Circuit (ASIC). By 1996, an entire NE may be producible as a single ASIC. Such a device would have greater reliability due to the reduced failure rate that is associated with a reduction in IC interconnections.

Microprocessor performance is measured by taking timing measurements during the execution of specialized benchmarking programs. A benchmark may attempt to provide a general purpose performance metric or to highlight a specific aspect of the general computational capability of a processor. For example, the Dhrystone benchmark focuses on the ability of a microprocessor to perform computations involving integers. The Whetstone is a more general purpose mix of operations involving integers and floating point data, function calls, conditional jumps, and array indexing. The memory access patterns of the Whetstone favors processors with an effective cache design. The Systems Performance Evaluation Cooperative has produced a suite of ten benchmarking programs to evaluate specific computational capabilities. The unit of performance in the SPEC test suite is the VAX 11/780 MIP. As Figure 6-10 shows, the performance of Reduced Instruction Set Computers (RISC) far outstrips the performance of Complex Instruction Set Computers (CISC).

Figure 6-10 [29] graphs processor performance versus year. In the lower left, we see various CISC processors (1,2,3,4,5,6,7,8,19). The new 68040 and 80486 processors (12,13) will have faster descendents (23). Slightly above the lower left is the current grouping of RISC processors (9,10,20,24,26,28). Higher performance versions have followed (11,29). More will follow (22,25,18,27).



Key	Processor	Key	Processor
1	68020/16MHz	16	RH32/25MHz
2	80386/16MHz	17	DARPA MIPS
3	68020/25MHz	18	R4000/70MHz
4	68030/16MHz	19	i80960MC
5	68030/25MHz	20	Mil-R3000/25
6	80386/25MHz	21	Mil-R3000/40
7	68030/33MHz	22	R3000/40MHz
8	80386/33MHz	23	68040/50MHz
9	R3000/16MHz	24	29000/25MHz
10	R3000/25MHz	25	29000/55MHz
11	R3000/33MHz	26	C300/50MHz
12	80486/25MHz	27	SPARC/Lightning
13	68040/25MHz	28	SPARC/25MHz
14	R6000/67MHz	29	SPARC/40MHz
15	SPARC/80MHz	30	Mil R4000/70MHz

Figure 6-10. Processor Performance Projection: 1987-92.

Although present FTPP implementations have used Motorola CISC processors from the 680X0 family, the FTPP architecture is independent of the microprocessor technology used for the PEs. Thus, the FTPP will be poised to take advantage of improvements in microprocessor technology as commercial off-the-shelf (COTS) products become available later in the decade. Minimal changes in the proposed architecture for the Maglev control computer system will be needed to take advantage of these technology advances.

Advances in micro-electronic packaging technology have drastically reduced the weight, power and volume needed to implement a given function. Two advanced methods for packaging electronic components have resulted in computers which can provide sophisticated functionality while consuming only minute amounts of scarce system resources. These electronic packaging technologies are monolithic and hybrid wafer scale integrated circuits. With monolithic wafer scale integration, the microprocessor, memory and other peripheral computer devices are etched onto a single square of silicon, thereby eliminating the wire connections between separate ICs. With hybrid wafer scale technology, individual ICs are not separately packaged, but are mounted as a die directly onto a wafer substrate in very close proximity. The hybridized approach produces a functional entity called a multi-chip module (MCM), which can be as simple as a memory board or as complex as a fully functional microcomputer.

These fabrication techniques provide several benefits. First there is a performance benefit. Interconnection delay represents a significant percentage of total signal propagation delay, with gate delay making up the balance. As device feature size decreases, interconnection delay begins to dominate the mix [29]. With wafer-scale technology, electronic signals have much shorter distances to travel, and therefore this form of packaging produces very fast devices. Their small size provides significant weight and volume reductions when compared to the weight and volume of traditional PC board computer fabrications. Power consumption is also reduced, primarily because most of the power on a PC board is dissipated in driving the capacitive load of external lines. Because CMOS technology can be used throughout, there is no need to drive PC card traces. Finally, this type of packaging provides a reliability improvement. A recent failure rate analysis of the design of some special purpose communication hardware, based on data from MIL-HDBK-217E, indicated that fifty percent of the hiatus failure rate was attributable to the wire connections between the ICs and the PC board [30].

There are several issues to consider when comparing the two types of circuit miniaturization. Multichip modules result in a higher yield when compared to monolithic circuit fabrication since smaller regions of the silicon wafer must be perfect. Each die can be tested before it is attached to the substrate. Monolithic devices can only be repaired by making or severing connections with laser or electron-beams. However, more extensive repair capability is claimed for MCMs in that faulty dies can be replaced by functional ones.

MCMs can support a very high interconnect density because these interconnections are part of a physically separate substrate. This means that the traces which connect signals on MCMs can be routed over and under the devices. Monolithic fabrication requires traces to be routed around the silicon which form the ICs themselves, thereby increasing module size. At the present time, both technologies are more expensive than standard PC board technology. However, because of their superior performance capabilities, their use will become more widespread and it is expected that their cost will diminish significantly in the future. These improvements in packaging technology can be used to improve the performance characteristics of both the PEs and the NE of the FTPP. By the time a prototype implementation of the Maglev control computer system is underway, these advances should be able to provide cost-effective performance gains.

Table 6-4 [31] compares the values of weight and power consumption of standard PC board and MCM implementations of three typical electronic system components: read/write memory (8 MByte SRAM), read-only memory (512 KByte EPROM) and a microcomputer board. All of the MCM products are manufactured by White Technology, Inc. of Phoenix, Arizona. The MCM EPROM (M4194E) comprises several dies mounted with thermally conducting adhesive to a ceramic base. Its PC board counterpart is composed of two standard 128 KByte EPROM devices (X28C010) manufactured by Xicor of Milpitas, California. The power and volume data for the PC board EPROM do *not* include the power and volume of necessary auxiliary devices, such as decoders. However, the figures for the MCM device is all inclusive. The MCM 8 MByte SRAM is an advanced product which has only recently become commercially available. Its PC board functional equivalent is composed of sixty-four 128 KByte SRAM devices (IDT71024) manufactured by Integrated Device Technology, Inc. of Santa Clara, California. Again, the data in the table for the PC board SRAM do not include measurements for auxiliary devices, but the values for the MCM SRAM are all inclusive. The MCM microcomputer is based on a Motorola 68020 microprocessor die. The module includes 128 KByte zero wait state EEPROM and SRAM, a DUART (RS-232), a counter/timer, a wait state generator for external devices, an optional MC68881/68882 floating point coprocessor and various other features. Its PC board counterpart, the Motorola MVME135 32-bit monoboard microcomputer, is not device for device equivalent. The most notable differences are that the Motorola board has more memory (1 MByte of DRAM), a memory management unit, two serial ports, two timers, and a VMEbus controller. As the data in the table shows, the MCM technology shows significant improvement in power and size over PC board fabrication, especially for the SRAM and microcomputer modules. These technical advances mean that systems which need high reliability, high throughput, and low resource usage can be designed to meet all their requirements. Even special purpose hardware needed to implement fault tolerant functions can be implemented with a cost effective, low volume-low power design using MCM technology.

	Power Consumption (Watts)			Volume (Inches)		
•Measurement excludes auxiliary devices	MCM	PC Board	Ratio (PC/ MCM)	MCM	PC Board	Ratio (PC/ MCM)
8 Mbyte SRAM	0.6	2.3*	4	3x 3.5 x 0.14	9 x 5 x 0.16*	5
512 KByte EPROM	0.4	0.8*	2	1.9x 2.1 x 0.16	1.3 x 3.4 x 0.43*	3
Microcomputer	3	25	8.3	2.4 x 2.4 x 0.2	9.2 x 6.3 x 0.8	40

Table 6-4. Quantitative Comparison of Wafer Scale and PC Board Technology.

Wafer scale technology greatly reduces the power consumption and weight and volume overhead of special purpose hardware to perform those functions. In fact, it is a valid principle of design that hardware specifically designed to perform a given function will always have a lower gate count, and therefore a lower weight and power drain and be able to operate at a faster speed than a more general purpose device which possesses a vast superset of the needed functionality [32]. Therefore, the FTPP uses special-purpose, efficient hardware, namely, the Network Element, to implement fault tolerant operations instead of using a microprocessor to implement them less efficiently in software.

6.6. SVA Onboard Control Computer Preliminary Design Specification

For the US Maglev Transportation System Onboard Control Computer (OCC), a baseline system using a Fault Tolerant Parallel Processor (FTPP) with one triplex Virtual Processor (VP) and one simplex spare has been selected. A minimal system has been selected based on the minimum redundancy level needed to mask hardware faults. It may or may not meet the quantitative maglev requirements for reliability, maintainability, availability and safety (RMAS) presented in Section 4.5. If the baseline system falls short of the Maglev RMAS requirements, additional processing elements (PEs) and/or network elements (NEs) will be added as necessary.

For the purpose of the present analysis, the processing elements will be Motorola 68040s which are directly compatible with the VMEbus-based network elements . It is understood from the discussion of technology advances in Sections 6.4 and 6.5 that the FTPP architecture is sufficiently modular and independent of given technologies that upgrades made possible by these advances will not necessitate significant changes in the proposed architecture. This baseline system will be modeled for safety, reliability and availability in Section 11.4.

6.6.1. Definitions

A brief review of the definitions of safety, reliability availability, and related terms for the onboard computer is presented here.

Mission: A *mission* is defined as a trip from one station to another, including departures and arrivals at stations.

Safety: A mission completes *safely* when one of two conditions is met: either (1) the mission completes successfully, i.e. the vehicle arrives at its destination without incident, or (2) the mission is not completed successfully but no one onboard is injured, i.e. when the vehicle stops safely at some intermediate point along the guideway. The probability of completing a mission safely must approach unity.

Fail-Safe: The ability to bring the vehicle to a safe stop under all possible failure conditions is denoted as the *fail-safe* feature of the system.

Reliability: The probability that a trip completes successfully is defined as the *reliability* of the system. The reliability requirement is not as stringent as the safety requirement because of the existence of the fail-safe mode of operation.

Availability: The *availability* of the system is defined as the probability that a given vehicle is ready to take-off for a mission on time, i.e. it has a minimum dispatch complement (MDC) of components in working order.

The quantitative RMAS requirements were discussed in Section 4.5.

6.6.2. OCC Functionality and Dependability Requirements

The functions performed by the onboard control computer include monitoring and control of all onboard systems including the HVAC system, the smoke and fire detection system, the doors, the cryogenic system for cooling the superconducting magnets, the onboard power generation system, the communication system, the operator interface and status displays, the onboard mechanism which directs passive switching, and the secondary suspension system which includes both active vehicle banking operations for coordinated turns and smoothing out the ride. Although velocity and position of the vehicle are directly controlled by the wayside zone control computers (ZCCs), the vehicle itself monitors its precise position and velocity and determines its own speed profile based on existing conditions onboard, in its present zone, and in the system as a whole. The requested velocity is sent as a command via a radio communication link to the wayside zone control computer for the zone in which it is traveling. The iteration rates, throughput, memory, and I/O bandwidth requirements of this ensemble of functions are well within the capabilities of the MVME-68040 processor board as specified by Motorola with adequate margins allowed to perform operations for fault tolerance and redundancy management.

Since most of the functions performed by the onboard computer are safety-critical, the inability to perform these functions reliably constitutes an unsafe condition and requires that the vehicle be brought to a stop. However, the vehicle has no onboard mechanism which it can apply directly to stop itself in an emergency. Thus, it must communicate the command to stop to the wayside zone control computer, which in turn controls the braking mechanism that can bring the vehicle to a stop. Every unsafe condition must be detected and cause a stop command to be issued. However, due to the redundant nature of the system, every failure does not create an unsafe condition. Each onboard safety-critical system has a certain minimum number of working components for safe operation. Failures which degrade the system below this level trigger a fail-safe stop. For example, as long as at least two channels of the OCC are operating correctly, safe operation of the onboard control computer is assured. Thus, if one of three channels of a triplex OCC were to fail, the remaining two channels can continue to operate the vehicle safely.

A failure mode which poses a special problem for reliability (as distinct from safety) is the potential ability of a failed channel to generate an unnecessary stop command, a so-called false alarm. If safety were the only consideration, false alarms would not present a problem. However, false alarms drive down the reliability of the system by increasing the number of unsuccessful missions.

In addition to reducing the reliability of each vehicle, false alarms also reduce the availability of the system as a whole. A disabled vehicle on the guideway renders the guideway unavailable to other traffic until it is removed. Such an event reduces the availability of the guideway. This adverse effect results from the nature of any form of guided ground transportation, including maglev. Hence, for maglev systems, the reliability of the onboard control computer system becomes a factor in analyzing the availability of the system as a whole.

The only conditions which can result in the vehicle entering an unsafe state are coincident or common mode faults. When a fault occurs in a second channel before the first one is detected and accounted for, a coincident fault results. Since the nominal repair time for a fault is less than 20 ms, coincident faults are near simultaneous faults and are therefore extremely rare, although the probability of a coincident fault is clearly a function of the failure rate. A common mode hardware or software fault occurs when a design error causes all channels simultaneously to fail or take an incorrect action. Not every common mode fault will result in an unsafe condition. Thus, the probability of a vehicle entering an unsafe state is extremely small, given both fail-operational modes and fail-safe stopping mechanisms which cover all non-coincident, non-common mode faults.

What is needed is the guaranteed ability to stop safely when continued operation would be unsafe, as well as the ability to prevent false alarms from triggering unnecessary stops. The design discussed below operates in just that way.

6.6.3. OCC Architecture

Figure 6-11 shows a block representation of the onboard FTPP. Three processing elements, designated T_A, T_B and T_C in the figure, form the fault tolerant virtual processor (VP) which conducts the onboard control functions. The fourth PE, designated S_1, acts as a spare. T_I is referred to as channel I. For simplicity, the four NEs are not shown in the figure. Each PE is connected to an I/O bus through a specialized interface called the monitor interlock. The operation of the monitor interlock is described below. The redundant I/O busses are identical. The sensors and actuators needed to perform all of the onboard control functions are attached to these busses. One of these devices is the radio transmitter used to send velocity/stop commands to the wayside zone controller.

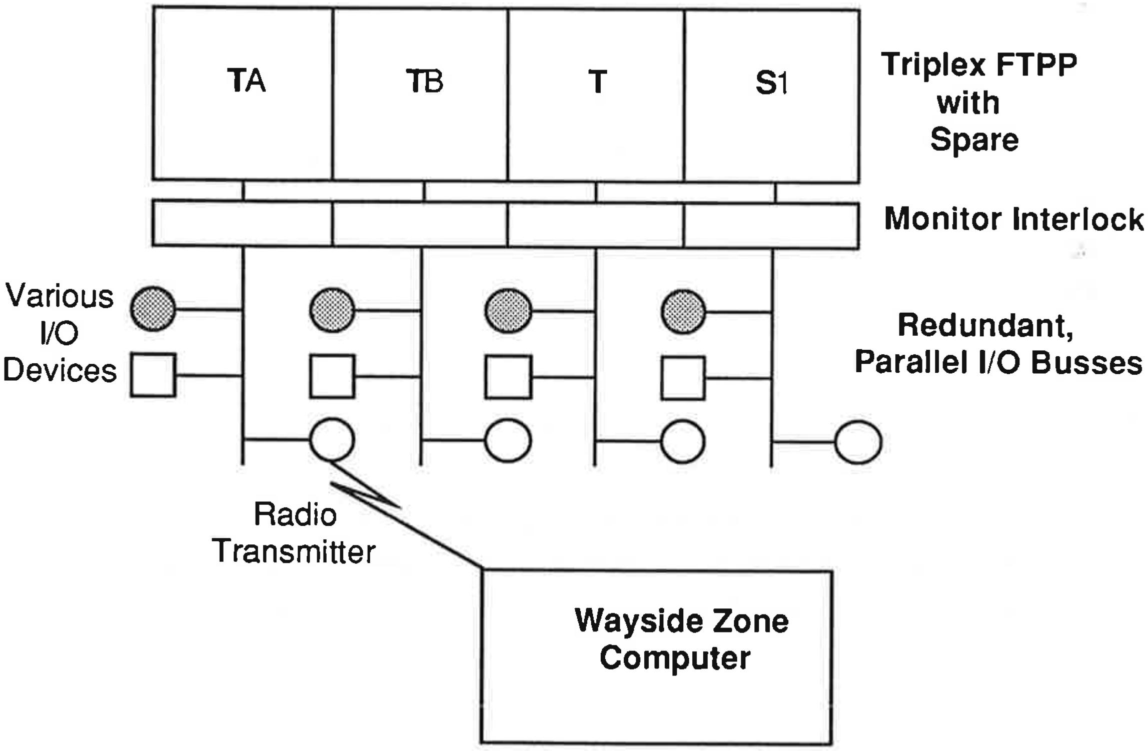


Figure 6-11. Block Diagram of Onboard FTPP Architecture

The purpose of the monitor interlock is to prevent a channel or transmitter which has failed in an active manner from flooding the system with false messages, i.e. the monitor interlock transforms active faults into passive ones. It operates by turning off the power to the failed channel or to its I/O bus. Each operational channel in a FMG is allowed to communicate with its attached I/O bus; only failed channels are depowered. The actions of the monitor interlock are determined by a protocol similar to that of authentication, to be described below. As long as an FMG is active in the system, they cause the monitor interlock to require a majority vote to depower a failed component or to activate the connection to an I/O bus. However, when only a duplex remains, neither channel can cause the other to be shut down.

Although all working channels are actively connected to their I/O busses, the channels which actually communicate with I/O devices on the bus are determined by the type of communication being conducted. For certain I/O functions, it may be desirable to allow all channels or a subset of channels to actively drive their I/O busses. For others, it may be desirable to use only one bus. Message transmission to the zone controller is in the latter category. Thus the VP decides which channel actually transmits messages to the zone controller.

Spare PEs or NEs can be used in the OCC to increase availability. Activating a spare is a form of automated repair called reconfiguration. Although the time required to effect this type of automatic maintenance is only on the order one second, it is still a relatively long period of time to suspend the vehicle control application which typically must execute every 5 to 10 milliseconds. Hence, for this analysis, it is assumed that reconfigurations of this type only occur when a vehicle is stopped at either a station or on the guideway following an emergency stop. If modelling shows the reliability of the triplex system is inadequate, the baseline system will be upgraded to a quadruplex FTPP (4 PEs) with 5 NEs and one spare PE.

6.6.4. OCC-Zone Controller Communication Protocol Timeline

The communication protocol followed by the onboard and wayside zone computers is designed to prevent false alarms while guaranteeing that every real alarm condition in the OCC results in the transmission of a stop command. It operates as follows. Periodically, the vehicle transmits a *well-formed message* (WFM) to the wayside zone controller. The specification of a well-formed message is presented below. The maximum allowable time period between a vehicle's transmission of two consecutive well-formed messages is denoted τ_1 in Figure 6-12. If the zone controller receives a well-formed message from the vehicle, it replies with an acknowledgement within a bounded period of time. The maximum allowable time between a zone controller's reception of a WFM and the OCC's reception of the acknowledgement is denoted by τ_0 in the figure. Since the time between the transmission of a message by the OCC and its reception by the ZCC is very small relative to the values of τ_1 and τ_0, these events can be considered as occurring simultaneously.

These time-out periods are important because the absence of an expected message or acknowledgement by either side within the time-out period results in a corrective action. When the corrective actions do not produce the required response, a potentially unsafe condition exists and results in an emergency stop. Thus, if the ZCC does not receive a WFM within a small multiple of τ_1, it assumes that the OCC is in an unsafe condition and so performs an emergency stop. Similarly, if the OCC does not receive an acknowledgement from the ZCC within τ_0 after transmitting a WFM, it assumes that the outgoing message has been corrupted and performs some recuperative action such as switching to another transmitter and retransmitting the message.

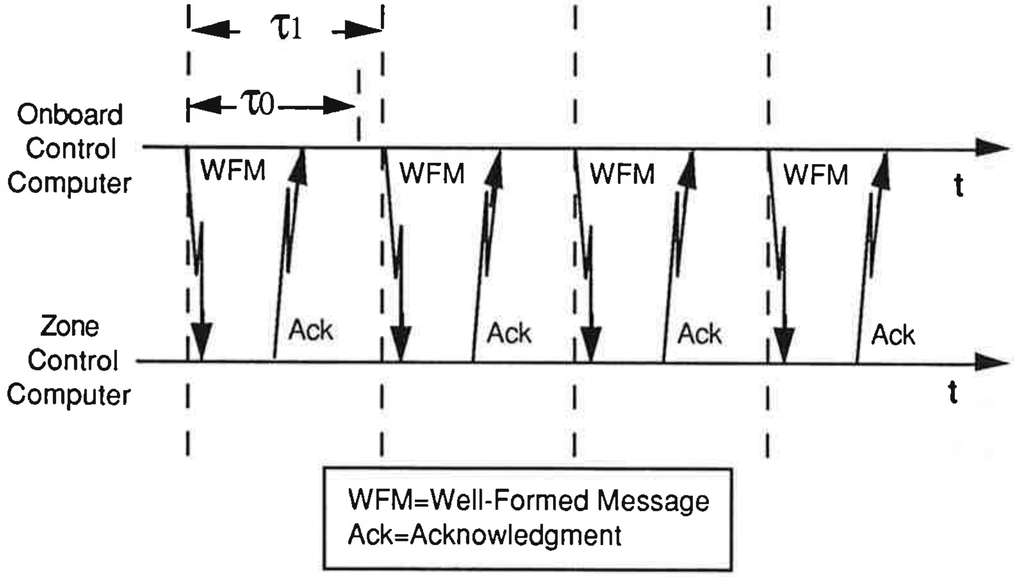


Figure 6-12. Timing Relationships for Fail-Safe Communication Protocol.

6.6.5. OCC Well-Formed Message Format

The format of a well-formed message transmitted from an OCC to a ZCC is shown in Figure 6-13. The message consists of a data field and an authentication field. The data field carries the velocity command and other information which the zone controller uses to propel the vehicle at the indicated speed as well as some information specific to the fail-safe communication protocol. The authentication field consists of N 64-bit subfields whose value is uniquely determined as a function of the current message and the channel of the FTPP which generated the value. Each subfield is called the signature of its channel. Each channel of the FTPP is able to generate a unique unforgeable signature for use by the wayside zone controller in authenticating a message, and, for a given message, no channel can generate the signature of another channel. Prior to transmitting the WFM to the ZCC, each non-faulty channel of the OCC signs the outgoing message and delivers it to the designated transmitter over the OCC's interchannel communication links. Since N is the redundancy level of the onboard VP, $3 \leq N \leq 5$. Thus for the baseline system $N = 3$.

There are three subfields in the data field which relate to the communication protocol: the emergency stop command, the authentication mode, and the message sequence number. The emergency stop command is a unique bit pattern indicating that the vehicle has entered a state whereby continued operation would expose the passengers to unsafe operating conditions and therefore the vehicle must be brought to an immediate stop.

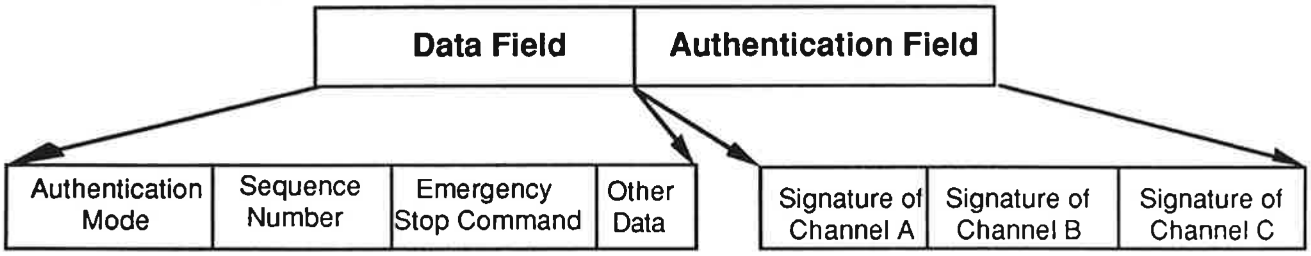


Figure 6-13. Format of Well-Formed Message for Fail-Safe Communication Protocol.

The authentication mode is a key to the authentication field of the *next* message. It indicates which channels are to be used for authentication of the next message to be transmitted by the OCC. For a fully working triplex, the authentication mode would have the value ABC, indicating that a well-formed message must contain at least two out of three valid signatures. To validate a message, the wayside zone controller verifies the signature for each channel based on the authentication mode from the previous message, as detailed in Table 6.5. In general, for a message to be considered valid, it must be authenticated by at least a majority of the channels specified by the authentication mode. For example, for an authentication mode of ABC, at least two signature fields must be authentic. However, for an authentication mode of AB a special rule applies. In this case, if both fields are not authentic, an emergency stop is required. This rule follows from the fact that safe operation of the vehicle requires at least two working channels in the onboard computer. If only one channel has correctly authenticated the message, then a failure in the second channel can be inferred by the ZCC.

The message sequence number is a sixty-four bit integer value which uniquely identifies the message. If the value of the current sequence number is n, then the previous sequence number was $n\text{-}1$ and the next sequence number will be $n+1$. Since the zone controller knows the upcoming value of n, it will ignore a message with an incorrect sequence number. This numbering scheme prevents a failed channel from re-sending a message which has been authenticated but which is no longer current. As a vehicle leaves one zone and enters another, the zone controller of the previous zone sends its counterpart in the next zone a handover message indicating both the initial value of the authentication mode and the message sequence number for the approaching vehicle.

Authentication Mode	WFM Requirements
ABCD	WFM if and only if (a) three out of four or (b) all of the signatures are authentic.
ABC, ABD, ACD, BCD	WFM if and only if (a) two out of three or (b) all of the signatures are authentic.
AB, AC, AD, BC,BD, CD	WFM if and only if all of the signatures are authentic.

Table 6.5. Requirements for WFM as a Function of Authentication Mode

6.6.6. OCC-Zone Controller Communication Protocol Operation

One further aspect of the protocol involves the number of onboard radio transmitters and receivers which participate in message passing. While each I/O bus contains a radio transmitter, at any given time only one onboard transmitter is used to send a message. Hence, the wayside zone controller only needs to process a single message. To provide the designated transmitter channel with the capability to append the appropriate authentication fields to the message to be transmitted to the ZCC, each channel independently calculates its signature and provides it to the designated transmitter over the OCC's inter-channel communication links. However, the radio receivers in all onboard channels listen for the acknowledgement. If an acknowledgement is not received by a majority of the channels' receivers within a specified timeout period after transmission of a WFM, the message is retransmitted from another transmitter. Other fail-safe mechanisms are in place to deal with failures of the ZCC. These include the ability of neighboring ZCCs to act as backups for each other as well as the ability of the central control computer to bring any vehicle to a stop anywhere along the guideway. The details of these mechanisms are beyond the scope of this analysis but must be specified before the fail-safe design of the system can be considered complete. Similarly, if the zone controller does not receive a WFM from the onboard system within a specified timeout period (such as some multiple of τ_1 seconds), it initiates an emergency stop procedure.

The authentication scheme described above is designed to prevent false alarms while ensuring that failures which create safety problems result in a fail-safe stop. An FTPP with a redundancy level of three or more is called a fault masking group (FMG) because any single failed channel can be both detected and masked by the resultant majority of non-failed channels. As long as an FMG is present, the authentication mode requires that a majority of channels issue an emergency stop command to the zone control computer. Thus, a single failed channel is prevented from sending a false alarm which can be validated by the zone control computer. However, when failures have accumulated to the point that only a working duplex remains operational, the next failure must trigger an emergency stop in a fail-safe manner.

To see how this works consider the following scenarios. Suppose that the onboard computer considers that channels A, B and C are working, and that the transmitter connected to channel C is designated to transmit radio messages to the ZCC. Thus, the message contains a authentication mode of ABC. Now suppose that C fails and transmits a stop command. The wayside zone controller would detect this as a false alarm since at most one authentication field (C's) would be valid. (Recall that C cannot forge the signatures of other channels since the probability of a random 64 bit pattern being correct approaches zero.) This message will not be acknowledged by the ZCC within the timeout τ_0, causing the onboard computer to retransmit a message from another channel, channel B for example, which is not failed and which therefore does not contain a false emergency stop command. Note that this message contains valid digital signatures from channels A and B, thus meeting the requirements for a WFM emanating from a triplex OCC. Thus, the failure of one channel does not trigger an emergency stop. As long as two channels continue to operate, messages are authenticated with the valid signatures of the two working channels.

Furthermore, the OCC's local fault diagnosis function can now update the authentication mode in the first WFM which is sent after the failure is detected to indicate that the OCC is now operating in a duplex mode. Thus the authentication mode field now contains the value AB. Next suppose that channel B, the designated transmitter in this scenario, fails. Either B attempts to transmit messages that are not well-formed or ceases transmission altogether. In either event the ZCC will shut down the vehicle after expiration of the timeout. Alternatively, A can detect B's failure and send an emergency stop message. Since this message has only one authentic signature, that of channel A, the vehicle is brought to a stop by the ZCC as required by the communication protocol. If, for some reason, A's message does not get through, the ZCC will not get a WFM, the timeout period will expire, and the vehicle will also be brought to a safe stop. Finally, if B fails such that it sends an "false-alarm" emergency stop message which has only one valid authentication signature and hence is not a WFM, the vehicle is again safely brought to a stop by the ZCC. Stopping at this point is a correct action since, by definition, B has failed leaving only a working simplex, namely channel A, in the system.

7. Qualitative and Quantitative Evaluation Criteria

Various criteria exist to evaluate control computer systems. A complete evaluation incorporates both qualitative and quantitative characteristics of the system. In practice, many major decisions are made on the basis of qualitative rather than quantitative comparisons. These qualitative criteria include flexibility, programmability, reconfigurability, topology, and validatability. The quantitative measures include reliability, availability, safety, and time-based measures of performance related to fault tolerance. This section will discuss these criteria and indicate how the FTPP and the Smart Vehicle Architecture (SVA) are capable of providing these desirable attributes to the onboard, wayside zone and central computers of the Maglev control computer system.

7.1. Qualitative Evaluation Criteria

Flexibility is a broad term encompassing several other attributes and which takes on slightly different nuances over the life cycle of a system. A flexible system will generally demonstrate a high degree of modularity. This property is especially important in the early phases of a design when the number of concurrent computers and their level of redundancy are being selected to meet the throughput and reliability requirements of the system. A robust architecture allows easy customization to meet the needs of many applications, without costly system software modifications or hardware add-ons. Modular design minimizes the interdependence of system components, facilitating change by eliminating ripple effects cascading through the system.

Flexible systems are both expandable and capable of being upgraded at later stages in their life cycle without incurring additional developmental costs. Fielded systems are not static; the user will want to add functionality as necessary and will want to take advantage of improved technologies which give better performance with lower power consumption. When expansion and technology upgrades occur, the dependability of the system must not degrade, nor must the system require extensive or costly redesign to deliver the same level of service.

Flexibility is enhanced by the use of standards in a design. Special purpose hardware needed to support system functions, such as redundancy management and inter-processor communication, should interface to the rest of the system by means of widely used standardized protocols. Systems which adhere to standards can use off-the-shelf components to reduce development costs.

Based on the discussions in Section 6.3, it is easy to see that the FTPP has a high degree of flexibility. The design is highly modular in that it supports, without modification to the communication hardware or to the system software, a virtually unlimited variety in the number and redundancy level of virtual processors (VP). By using a standard VME bus interface, technology upgrades to accommodate improved processors, both general and special purpose, are greatly facilitated. In a similar way, the special-purpose fault tolerant communication hardware (the NEs) can be upgraded and expanded without requiring a change in the processing elements (PEs). The SVA takes advantage of the flexibility of the FTPP in proposing a different configuration of the FTPP for each principal computing site in the system. The SVA design itself, by virtue of its decentralized, distributed nature, is highly modular, allowing the number of wayside zones and vehicles to grow without limit to meet the needs of an expanding Maglev transportation system. This design can be compared to centralized systems whose performance will degrade as the system expands until finally a costly and complex redesign of both hardware and software is required to accommodate growth.

Furthermore, the US Maglev Transportation System will include several applications, integrated to achieve a dependable and safe mode of travel. For example, it will employ separate applications for route planning, velocity control, propulsion control, ride quality, passenger services, etc. These applications may not have the same level of criticality and therefore the only cost effective way to support them is with a computing platform, like the FTPP, which can accommodate mixed levels of redundancy. For example, the passengers may be provided with a telecommunications link which is considered to have a very low level of criticality and may be suitable for a simplex processing site while ride quality may have safety related aspects which require it to run in a triply redundant site. Since one type of processing element may not be ideally suited for each application, the ability of the FTPP to integrate different types of special purpose processors is desirable. For example, GPS is a signal processing application. If GPS is used to determine the position of the vehicle, special purpose signal processor can be incorporated to perform that function with no hardware or software modifications required by the basic FTPP architecture. Furthermore, these diverse applications need to run concurrently and communicate reliably with each other, thereby utilizing the support for parallel processing and highly reliable communication among processing sites provided by the FTPP.

Furthermore, a rigorously adhered to policy of separation among various application components and between application and system components for both software and hardware also promotes flexibility. When a well-defined interface exists between various entities in the system, changes in one component do not impact another as long as the interface specification is honored. The use of layered protocols in the FTPP and among the various subsystems in the SVA foster this mutual independence.

Programmability is an informal measure of the ease of implementing a given application on the system. For redundant systems, it is especially important that the system functions for fault tolerance, such as voting, interactive consistency protocols, fault detection and masking be transparent to the user. Non-standard features that require special system knowledge, additional programming overhead, or extra code for special purpose devices adversely affect the system's programmability and reduce the productivity of the application programmer. Worse still is the danger of increased software errors due to added programming complexity, or inconsistently programmed special purpose co-processors. Ideally, the programming model used by the applications programmer is that of a standard uni- or multi-processing architecture, using familiar operating system calls and constructs. Support for such features as concurrent programming constructs, multi-version software techniques, and standard programming models enhance the programmability of a system. For example, in a distributed system, one task may wish to communicate with another task by means of a logical rather than a physical identifier using a standard Ada rendezvous technique. The details of the communication are supported by the tools and the system software delivered with the system, including the compiler, linker, and communication and scheduling software. Support for testing, debugging, and data retrieval in a multiprogramming environment are also essential. A system that is difficult to use will be doomed by end users' complaints.

The FTPP provides an application programmer with a standard multi-tasking, parallel processing Ada-based programming model. All of the system functions for fault tolerant operation and recovery are transparent to the application. Furthermore, the hierarchical organization of the SVA allows a clear mapping of functions onto computer subsystems, fostering the development of a modular design and well-defined interfaces. Instead of being faced with a complex problem, the architecture has already subdivided the system functions which significantly reduces the overall complexity.

Reconfigurability refers to both the ability to assume a variety of initial configurations and the ability to regroup resources to restore reliability and availability levels to critical functions after a failure has occurred. Initial configurations which are supported by the FTPP include distributed processing capability with mixed levels of redundancy and varying types of input/output interfaces for communication with sensors and actuators. The SVA/ZCA approaches also allow some redistribution of functionality to various sites within the overall Maglev system to achieve the best system performance and reliability. Reconfiguration in response to a fault should be handled on two levels: logical and physical. For a fault tolerant computer to be suitable for Maglev, it must, like the FTPP, have the ability to instantaneously mask errors at the system outputs. Furthermore, once the error is detected, a logical reconfiguration must take place, in which the faulty channel is logically removed from the redundant group to which it belonged. At this point, the reliability of the system is reduced, and over time, would degrade to an unacceptable level without outside intervention. However, in a well-designed system like the FTPP which supports parallel-hybrid redundancy management strategies, a physical reconfiguration can take place. During a physical reconfiguration, a spare simplex processor can be used to replace the faulty channel in a triplex or greater redundancy group. Thus, the reliability of the critical functions in the system is maintained and the availability of the system is extended. Furthermore, the fail-safe modes of the SVA are actually fail-operational modes, where the failure of one system causes functions in the other to detect the problem and take over control of the system, albeit in a degraded mode, so that a graceful recovery is possible.

The topology, or the communication scheme, of a distributed, fault tolerant architecture plays a key role in determining the values of several system metrics, including reliability (Byzantine Resilience requirements must be met), power consumption, weight and volume, and transport delay (the efficiency of the communication system determines this metric). In the FTPP, the topology supports the requirements of a Byzantine Resilient design, without excessive power, weight and volume requirements. It can be reconfiguration in real time without massive overheads for computation of message routing paths. In addition, the NE interface supports system expansion at some future date without undergoing major software or hardware redevelopment.

Finally, an often overlooked attribute of a fault tolerant system is its ability to become mission qualified by the regulating agency charged with the oversight of its function. In the case of Maglev, this agency is the Federal Railroad Administration (FRA). A validatable system is highly observable. The system will have been designed for testability, since this remains the primary means for validating software. Furthermore, the fault tolerance functions will meet the theoretical requirements for achieving interactive consistency among redundant channels which allows a straightforward assessment of its reliability claims. If parts of the system can be validated separately, this simplifies the overall task. It should be possible to separate fault tolerance functions from application functions for purposes of analytical modelling and for testing. Sufficient data must be available about the failure rates of various components, including the software, to accurately model the system. Ideally, the system will have been designed with the final validation effort in mind, so that proper use of standards and conservative design practices result in test measurements that fall comfortably within the required ranges for performance and reliability. From the preceding discussion, it is clear that the FTPP and the SVA satisfy these requirements for the Maglev control computer system.

7.2. Quantitative Evaluation Criteria

Dependability is the quality of service that a particular system provides. Reliability, availability, safety, and maintainability, are measures used to quantify the dependability of a system. Precise definitions for these metrics were presented in Section 4. In addition, this section describes some basic definitions used in various dependability analyses, the concept of fault coverage, and some performance metrics related to fault tolerance. These are further expanded in Section 11 which discusses analytical modeling.

It should be noted here that the quantitative analysis discussed here is being applied to the hardware, by taking into account the component failure rates, and to the architecture, by taking into account the arrangement of the redundant hardware elements, their interconnections and the redundancy management strategy. In the quantitative analysis, for simplicity, it is assumed that all the software is perfect. The software is analyzed critically according to a different set of criteria discussed in Section 9.

The **failure rate** of a component, typically denoted by λ, is the expected number of failures of that component per unit of time. Experience has shown that the failure rate of electronic components follows a "bathtub" curve as shown in Figure 7-1 [5]. The high failure rate in the early part of the curve is due to the quick burnout of substandard or weak components. During their useful lifetime, most electronic components exhibit a constant failure rate. Towards the end , the failure rate increases again as components near the end of design lifetime. This distribution explains why it is impossible to establish a preventive maintenance program for digital systems. An "old" component which has not yet failed has the same probability of failure as a "new" component which has not failed during the flat part of the curve.

The **mean time to failure (MTTF)** is the expected time that a system will operate before the first failure occurs. Although the definition of the **mean time to repair (MTTR)** a system is simply the average time required to repair the system after a failure, it is an extremely difficult parameter to estimate. It is often determined empirically, by injecting a set of faults into a system and observing the time required to repair the system. The MTTR may be stated in terms of a repair rate μ, which is the average number of repairs that occur in a time period. MTTR = 1/μ. The repair rate is typically specified for several levels of repair [5]. The first level is called the organizational level, and consists of all repairs that can be performed at the site where the system is located. For the Maglev transportation system, an organizational repair to the onboard computer system could be effected on the guideway. The key to being able to perform an organizational repair is the ability to locate the fault. At the organizational level, fault location depends on built-in test procedures and self-diagnostics and automated recovery. The second level of repair is called the intermediate level, and consists of repairs that can be performed in the immediate vicinity of the system. For the Maglev transportation system, an intermediate repair to the onboard control computer could be performed in a station. The final level of repair is called the depot level. Depot level repairs are performed at a factory or major facility where extensive diagnostic and test equipment and personnel are available. For the Maglev transportation system, this may be a specialized central repair facility for vehicles which is accessible from many regional routes. Clearly, the length of time taken to effect a repair varies with the level at which it can be accomplished, with organizational repairs being the fastest and depot level repairs taking the most time.

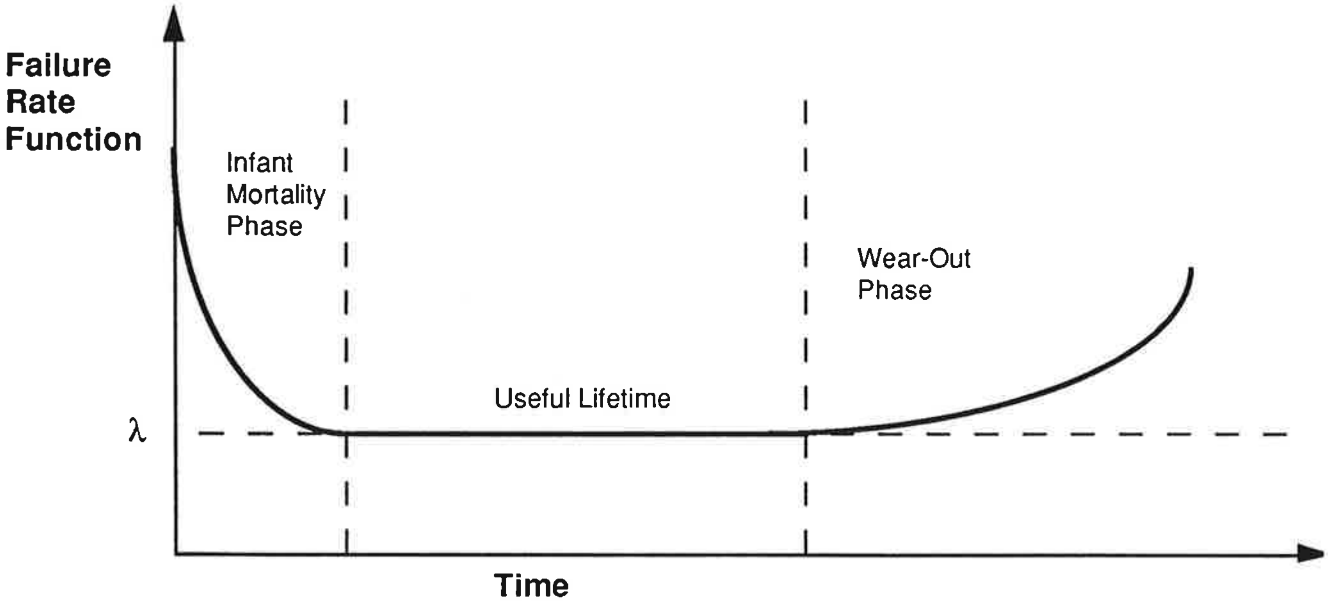


Figure 7-1. Failure Rates of Electronic Components.

The **mean time between failures (MTBF)** is equal to the sum of the MTTF and the MTTR. That is, MTBF = MTTF + MTTR. Since the MTTR is usually very small compared to the MTTF, the value of the MTBF is often very close to the value of the MTTF. For redundant architectures, MTTFs and MTBFs are not good measures of reliability, since the added hardware components typically make the MTBF of a redundant system smaller than the MTBF of a non-redundant computer. However, the probability of a catastrophic failure of a redundant system is much less than that of a simplex system.

Fault coverage has two definitions, one being more intuitive, the other more mathematical [5]. Intuitively, coverage is a measure of the system's ability to correctly perform fault detection, fault identification and recovery (FDIR). It also is a measure of the ability of the system to contain a fault to a local region of operation and not to allow the fault to propagate to non-faulty system components. In other words, fault coverage is a measure of the system's ability to recover from faults and maintain operational status, thereby demonstrating fault tolerance. Mathematically, fault coverage is defined as the conditional probability that, given the existence of a fault, the system recovers. Clearly, fault coverage is extremely difficult to calculate. In the past, when electronic systems were much simpler than they typically are today, a common approach to coverage was to develop a list of all the faults that could occur in a system and then a second list of all the faults from which the system could recover. Fault coverage is then the fraction of faults from which the system can recover. Faults were often postulated as stuck-at-0 or stuck-at-1 faults. They were assumed to be permanent, despite the fact that many faults are transient or intermittent. These approaches are not appropriate for complex systems. Fortunately, the theory of Byzantine Resilience has provided a means to provide coverage for any arbitrary fault, without the need for specific enumeration. For the complex digital systems which will be used in the Maglev control computer system, various types of modeling techniques are the only practical way to quantify fault coverage.

The performance-related quantitative parameters of interest for a fault-tolerant system have to do with the overheads of fault tolerance. Overheads typically penalize throughput, useful memory, and I/O transaction speeds as compared to a simplex system performing the same functions.

In the past, consideration of the overhead for memory utilization was extremely important since memory devices required significant amounts of power, weight and volume. However, modern technology has reduced the size and power consumption of these devices to the extent that memory utilization only becomes a concern for applications requiring relatively large amounts of memory. For example, vision systems, with their heavy dependence on array structures, require a very large memory allocation. However, the software which performs redundancy management and fault tolerance for a control computer system requires only small to moderate amounts of memory, typically on the order of 100 kilobytes.

In contrast, throughput penalties for conducting fault tolerant operations are a more serious design consideration for real-time control applications, especially for applications in which the failure to meet a scheduling deadline poses a safety problem. Some throughput overheads are constant, such as built in tests to detect latent faults. These tests execute in the background when periodic application tasks have completed and processing time is available before the application tasks' next scheduled execution time. Other overheads can be proportional to the utilization of resources, e.g., number of I/O transactions, CPU workload, etc. An example of a proportional overhead is the source congruency communication and voting load for each input. This is proportional to the number of inputs and their frequency.

I/O transaction speed is a source of concern in real-time applications which have constraints on the allowable transport lag, i.e. the time which elapses between the reading of a sensor and the command sent to a control actuator. In general, this amount of time needs to be minimized. For example, the control law for dynamic suspension to improve the ride quality of a Maglev vehicle bases the computation of the actuator command on the difference between the sensor-supplied reading of an acceleration parameter and the desired value of that parameter. The actuator command is intended to reduce this difference. The corrective action needs to be applied before the value of the parameter, which is itself a function of time, changes. Any communication overhead increases the transport lag and therefore, this increase must not only be minimized but be a deterministic and measurable quantity. Even simplex systems must deal with this phenomenon. The requirements of maintaining interactive consistency among the various computing channels in a fault tolerant system merely exacerbate the problem. However, well-designed hardware is capable of reducing this additional overhead to a minimum.

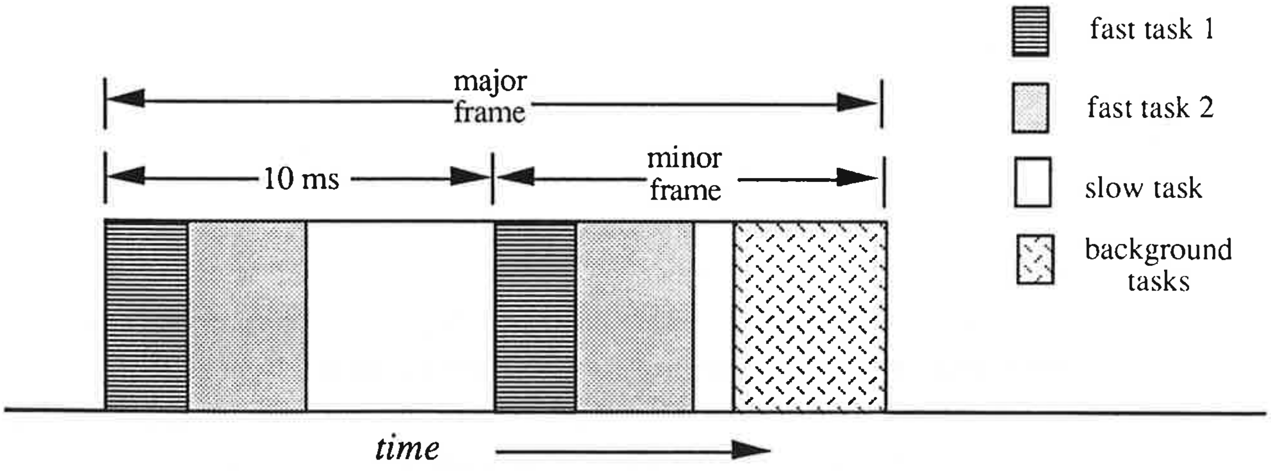


Figure 7-2. Rate Group Scheduling.

The secondary suspension control application and the control of the linear synchronous motor which propels Maglev vehicles forward along the guideway are examples of hard real-time tasks which are designed to execute in a frame-based, rate-group schedule. Each task is expected to start and complete within some window of time, called a minor frame, and all tasks are expected to complete in a larger window of time called a major frame.

Consider a simplified example of a schedule with only two rate groups which has two tasks in the fast group and one task in the slow group. Figure 7-2 illustrates this example. The fast tasks have a minor frame period of 10 ms and the slow task has a minor frame period of 20 ms. Thus, the major frame is also 20 ms, since all three tasks must complete in this amount of time. However, the fast tasks should each have executed two times within that 20 ms interval. Background tasks are allowed to execute in the time remaining between the point at which the slow task completes and the start of the next minor frame. Real-time control applications are characterized by "hard" real-time constraints, i.e. the safe operation of the system is endangered if tasks do not meet their scheduling deadlines. A task which does not complete in its allotted time period is said to have overrun. This overrun in turn causes other tasks to miss their deadlines. Overruns pose serious safety problems for Maglev control software which is written with the assumption that regular, periodic execution of a task on evenly spaced intervals is guaranteed.

The overhead for performing redundancy management in a fault tolerant system becomes a major concern if it interferes with these hard, real-time scheduling requirements. Indeed one of the high priority tasks, which must execute in the fastest rate group in a fault tolerant system, performs fault detection, isolation, and recovery (FDIR) for the redundant hardware. This scheduling requirement is imposed to meet the system requirement for fault coverage. This requirement means that the system must detect and recover from a fault before the occurrence of a second fault. Hence, fault detection must take place as soon after a failure occurs as possible. For rate group scheduling, this means the FDIR task must run at least as often as the most frequently scheduled task. If voting, synchronization and the operations needed to provide interactive consistency are implemented in hardware, as they are on the FTPP, the performance overhead for fault tolerance is extremely small, generally between 5 and 10% of the total throughput. This includes the overhead need to provide interactive consistency as well as run the routine FDIR task. Since the absolute time needed by these operations is a measurable quantity, ensuring that they do not cause task overruns is relatively straightforward. However, a parameter of more concern in characterizing the effects of fault tolerance on performance is the transient in performance caused by actual fault occurrence and fault handling. The temporary overload, if not handled properly, can result in delays in application tasks being executed or, in the worst case, outright suspension of the application tasks.

Figure 7-3 shows the sequence of events which follows the occurrence of a fault in a fault tolerant system. The time between the occurrence of the fault and its manifestation, i.e. the occurrence of an error, is highly unpredictable. However, this unpredictability is relatively unimportant since a fault is not an active problem until it causes an error.

Background tests, such as those run on the FTPP to locate memory faults or faults in other devices, uncover faults that have not as yet caused errors and thereby make the FTPP more robust by increasing the fault detection coverage of the system. Faults which are detected by actively looking for them cause the least disruption to system performance, because their source can be identified and a recovery action taken before the error is manifested. Once an error has occurred, the amount of time required to detect the error, t_d, is bounded by the period of the fastest running, i.e. highest frequency, rate group since the FDIR task is then scheduled to run. For example, assume that the error occurs immediately after the FDIR task has completed an iteration. It will be detected during the next iteration of the FDIR task. In the previous example, t_d is 10 ms. Of course, if the error results in an incorrect output calculation by the faulty channel, that error is instantaneously masked in real-time by the output voting conducted by the FTPP system software.

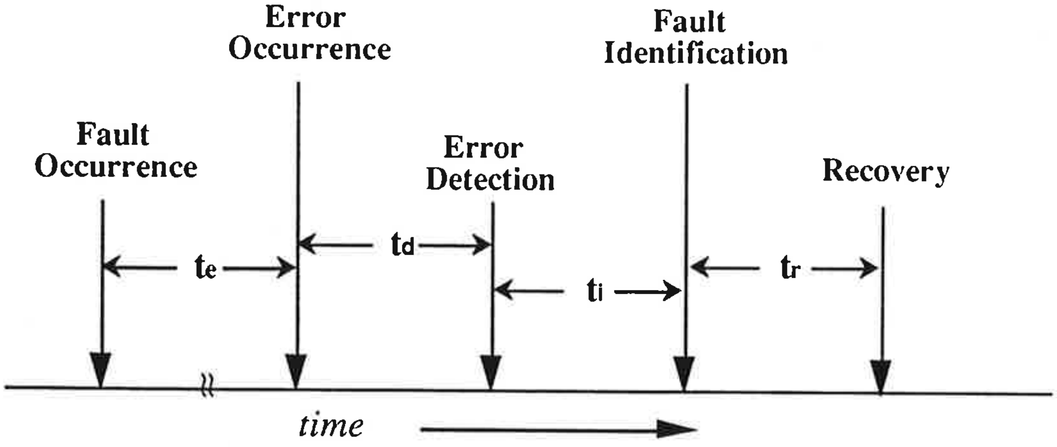


Figure 7-3. Sequence of Events Following Fault Occurrence.

Once the error is detected, the source of the error must be identified. This generally means that the fault containment region to which the faulty hardware belongs must be identified. The time required for this logic, t_i, is a non-deterministic quantity, but in general, worst-case values for a given system can be obtained empirically. After the fault is identified, an appropriate and pre-determined recovery action is taken. There are two basic recovery strategies from which to choose: passive reconfiguration or dynamic reconfiguration. Passive reconfiguration is simply a form of graceful degradation of the system. Using this approach, the faulty component is excluded from the system and the redundancy level of the system is reduced by one. For example, if one channel of a quadruplex configuration fails, a passive reconfiguration causes the quadruplex to be degraded to a triplex. The changes which this approach requires are minimal, for example the mask used by voting hardware must be reset. Hence, this is the strategy which would be used when a vehicle is travelling along the guideway. A dynamic reconfiguration is possible if the fault is determined to be a transient or, if spare capacity is available, a non-failed channel can be substituted for the faulty one. In either case, a memory alignment of the new member is necessary to ensure that all channels of the fault tolerant computer have identical states and identical data. Memory aligned is a time-consuming process, and this strategy is therefore suited to station stops or other relatively idle phases. When this type of reconfiguration takes place during an idle phase, it restores the availability of the system. When it takes place during a mission phase, it restores its reliability.

8. Redundancy Management for the Maglev Control Computer System

Redundancy by itself does not in general guarantee high reliability. Correct management of redundancy is crucial to transforming a redundant system into a fault tolerant one. The following checklist can serve as a starting point for a critical analysis of redundancy management in a fault tolerant architecture.

Single Point Failures: Despite redundant hardware and software, it is possible that failure of a single component can cause the whole system to fail catastrophically. Someone well versed in the fault tolerance field can examine an architectural specification and the redundancy management approach and determine by inspection whether the architecture has the fundamental mechanisms for protection against single point failures. The absence of a single point failure, as far as one can tell from an architectural inspection, does not, of course, imply that none exists. Eventually, prototypes must be fabricated and subjected to various tests to assure none exist.

Fault Containment: How are the faults in one computer element, e.g. a processor, confined to that element and stopped from propagating to the redundant copies of that element? In other words, does the system enforce fault containment boundaries?

Error Containment: How are errors which result from faults in one computer confined to that computer and stopped from propagating to redundant computing elements? In other words, does the system enforce error containment boundaries?

Real-Time Error Masking: Can the system mask the effects of a fault in real time or does it use some other non error-masking strategy such as cold or hot spares that need to be switched in after detection of an error?

Degree of Synchronization: For active redundancy architectures, are the redundant processors operated synchronously or asynchronously. If synchronously, are they synchronized in frames, microframes or only loosely?

Degree of Consensus: Do redundant processes arrive at exact bit-wise consensus or do they only approximately agree with each other under no-fault conditions?

Failure Modes Covered: Is the system designed to cover permanent hardware faults, transient faults, and intermittent faults?

Common Mode Faults: Is the system designed to cover common mode faults, i.e., faults that affect multiple fault containment regions simultaneously?

Output Errors: Are errors in the computational core contained within its boundaries or can they propagate out to actuators, displays, and other subsystems?

Graceful Degradation: Can the system reduce its workload in a prioritized fashion after a fault occurrence or does it stop completely?

programming techniques, formal proof of correctness techniques, and architectural considerations.

Automated design aids include formal specification languages and other metalanguages. They are useful in tracking the software development process, especially as changes are introduced at the specification level, and in supporting documentation of the system software. Computer aided software engineering (CASE) tools are now commercially available which provide automatic code generation directly from a design specification. In the past, compilers supported the development of more complex, more reliable and more maintainable programs, by increasing the level of abstraction which a programmer could express. In a similar manner, CASE tools support the development of more sophisticated and more reliable programs by providing further support for data and control abstraction, and by increasing visibility into a complex software system at a higher level, and by greatly reducing the time spent on coding and thereby allowing increased effort to be expended on the error prone specification phase of development. Studies have shown that use of formal specification languages result in programs which have far fewer errors uncovered by the testing process. CASE tools impose both the discipline of formal specification and automated code generation. As the code generation products become more widely used, the level of confidence in their ability to generate correct code from the stated specification will grow in the same way that confidence in compilers has grown. When it is time to begin the software specification for the prototype Maglev control system, CASE tools which have demonstrated their dependability should be commercially available.

A principal cause of design faults is design complexity. Computer architectures which are designed to reduce this complexity will reduce the incidence of design flaws, since human errors are more likely when dealing with complex systems and unconventional concepts. The designers of the FTPP have addressed this issue on two fronts. First, the platform itself, that is the system hardware and software, are designed around the simple, precise, and formal requirements for Byzantine Resilience and fault containment. These leave no room for ambiguity or confusion as to what the system is required to do. Second, the user interface to the system, that is the programming model that an application programmer deals with, is conventional and unobtrusive. To reduce cognitive overhead to a minimum, the FTPP architecture is designed to make the mechanics of fault tolerance transparent to applications programmers. Furthermore, the distributed nature of the FTPP is designed to keep the complexities of interprocessor communication hidden behind appropriate abstraction barriers.

The fault avoidance techniques discussed above are by no means a comprehensive list of the methods used to provide software fault avoidance. However, no set of improved programming techniques can guarantee complete confidence in the correctness of a program. Therefore, methods have been developed to extend the approaches of fault tolerance to software. Unfortunately, at the present time, these methods do not have the benefit of the theoretical rigor of Byzantine fault tolerance to random faults to aid in their validation. Nevertheless, strategies do exist which can reduce system failure as a result of software errors.

Fault tolerance always involves some redundancy and therefore increases the resource expenditure to perform a given function. Two different techniques have been discussed in the literature to achieve software fault tolerance [31], each using a different form of redundancy. The recovery block method, originally proposed a heuristic solution to hardware fault recovery, utilizes temporal redundancy. N-version programming which is based on the redundant component scheme for hardware fault tolerance, utilizes spatial redundancy. In this context, temporal redundancy is the execution of a different version of a software module following the detection of an error. Spatial redundancy is the concurrent execution of different versions of the software followed by a comparison of the results to mask faults.

Recovery blocks and other rollback recovery schemes were first introduced to deal with transient hardware errors which were extremely common in early electronic components. These schemes range from a simple instruction retry to the laborious preservation of the entire state of a module so that a program segment could be re-executed in the hope that the hardware fault would not manifest itself a second time. More recently, the technique has been applied to software fault tolerance [33]. The algorithm is quite simple:

```
Execute a software module, $P$, the primary module.
Run the results of P through an acceptance test, $T$.
If $P$ fails $T$, then execute an alternate, hopefully independent, module, $Q$, designed to
produce the same result.
If $Q$ fails the acceptance test, $T$, go to fail-stop mode.
```

For real-time applications, the acceptance test is augmented by a watchdog timer to protect against infinite loops. In practice, care must be taken that P not alter data until T has been ensured. The acceptance test is key to the entire operation. It must be thorough but not burdensome, since it executes every time P does, but is considered an overhead function. For applications dealing with physical phenomena, the acceptance test usually takes the form of a reasonableness check. Individual values can be examined to determine whether or not they are in range. Differences between successive values of a given variable can be compared to determine if an unexpected shift has occurred. Or values of different variables or their successive increments can be correlated. Another opportunity to use recovery is provided by computer run time checks. These include hardware implemented detection of anomalous states such as divide-by-zero, overflow, underflow, attempts to execute undefined or privileged instructions, and attempts to write into protected memory areas. The Ada language facilitates recovery methods by providing the formal syntax for exception handlers. In multiprocessor systems, rollback or recovery techniques can quickly become too complex to be useful if many machine states are interrelated and a domino rollback effect can occur. One study used this technique in implementing a Naval Command and Control System and reported a coverage factor of 74%. Of 222 potential software failures, 165 were masked by recovery techniques [35]. Clearly, recovery blocks can be part of a software fault tolerance strategy. They work best when a manageable amount of code can be checked by highly dependable means and when sufficient time for processing is available. The SVA for the Maglev Control Computer System provides a fail-safe operational mode based on the recovery block paradigm. The method employed is even more robust since the primary module *P* executes on the onboard computer subsystem and the acceptance test *T* and the recovery module *Q* execute on the wayside zone control computer. The mechanism would work as follows. Data for vehicle velocity and position control are developed on the onboard system and transferred to the wayside system with the current velocity command. In the above model, this constitutes the module *P*. The wayside zone controller would execute the acceptance test *T* on this data/command pair and carry out the command when the conditions of the acceptance test are met. When the command and data do not pass the acceptance test, the wayside would execute the recovery module *Q*.

In N-version programming, multiple, independent copies of a software module are executed concurrently on a multi-processing machine, with a majority vote of the redundant results used to determine the final output of the system. There are several benefits with this approach. The issue of who checks the checker, i.e. what coverage is provided for errors in the acceptance test, does not arise. For systems with real-time performance requirements, the overhead is very small since the versions run concurrently. And faults are masked instantaneously. The drawbacks to this technique are the degree to which programs written by different people are truly independent, the cost of developing multiple versions of a program, and the overhead caused by the need for a specialized voting algorithm since different versions of a program, compiled by different compilers, may produce correct, but not bit-for-bit identical results. One quantitative experimental study [36] has shown only modest coverage (an improvement of 10%) with this technique, although the programmers were undergraduate students from the same programming class, not the best way to achieve programmer independence. Although the FTPP architecture supports N-version programming, however, the SVA architecture does not intend to use this technique to obtain greater software reliability.

Another area with potential problems for system reliability which critically impacts software development is the human interface to the system. The SVA is designed to provide fully automated routine operation for the US Maglev transportation system. A provision is made to allow human operators to take control when an emergency situation has arisen and travel speed is significantly reduced (less than 80 km/hr) or when a vehicle must travel on its wheels to return to a terminal for repairs. Including a human interface for routine operations greatly complicates the design of the system software. This complexity increases the amount of code which must be written and increases the likelihood of software errors. Furthermore, the human in the loop adds a level of unpredictability in the inputs to the system. Unpredictable inputs are a chief cause of program crashes, since programs are designed to handle a finite number of combinations of circumstances. Full automation means that all inputs and outputs can be fully specified and will be visible to system designers who can then plan responses to a known set of factors.

9.2 Software Fault Removal

Faults that slip past the design process can be found and removed at various stages prior to the computer system becoming operational. Fault removal techniques and tools include design reviews, simulations, testing, fault injection, and a rigorous program of discrepancy reporting and closure. Traditionally, these techniques have been relied on almost exclusively to deal with common-mode faults. Most of these techniques, with the exception of fault injection, are well developed and well known. We will, therefore, limit the discussion to the use of fault injection for fault removal.

Insertion of faults in an otherwise fault-free computer system that is designed to tolerate faults is a powerful technique to exercise redundancy management hardware and software that is specialized, error-prone, difficult to test and not likely to be exercised under normal conditions, i.e., likely to stay dormant until a real fault occurs. Fault insertion techniques can also be used to operate the system in various degraded modes which are expected to be encountered in operational life of the system. Degraded mode operation stresses not only fault handling and redundancy management aspects but also task scheduling, task and frame completion deadlines, workload assignment to processors, inter-task communication, flow control, and other performance-related system aspects. Fault insertion exposes the weaknesses in the hardware and software design, the interactions between hardware and software, and the interactions between redundancy management and system performance. It is an accelerated form of testing the hardware, software and the system, analogous to "shake and bake" testing of hardware devices.

9.3 Real-Time Software Fault Tolerance

Common-mode failures that are not removed prior to operational use of a computer system may eventually manifest themselves in the field as the coincident failure of multiple components of a redundant system. At this point the only recourse is to detect the occurrence of such a failure and take some corrective action. These are fault tolerance techniques and following is an unprioritized list of such methods.

9.3.1. Real-Time Software Fault Detection

Before a recovery procedure can be invoked to deal with common-mode faults in real time, it is necessary to detect the occurrence of such an event. Many ad hoc techniques have been developed over the years to accomplish this objective. Most of these techniques can also be used prior to operational use of the system to eliminate faults. The difference is that in the fault removal phase, detection of a fault leads to some trap in the debugging environment while in the operational phase it will lead to a recovery routine. Similarly, fault removal techniques discussed above can also be used to aid in the task of detecting faults in real time, albeit with a high penalty in performance.

a. Watchdog Timers

Watchdog timers can be used to catch both hardware and software wandering into undesirable states. They are typically used in the Processor Element but can also be employed in the Network Element of the FTPP. Neither hardware watchdog nor task timers unambiguously indicate the occurrence of a common-mode failure. The syndrome in the failed channel of a physical fault is no different from that of a common-mode failure. The syndromes across redundant channels must be compared in real time to determine the cause.

b. Hardware Exceptions

Hardware exceptions such as illegal address, illegal opcode, access violation, privilege violation, etc. are all indications of a malfunction. Again, syndromes across redundant channels must be correlated to distinguish between physical and common-mode failures.

c. Ada Run Time Checks

Ada provides numerous run time checks such as type checks, range constraints, etc. that can detect malfunctions in real time. Additionally, user can define exceptions and exception handlers at various levels to trap abnormal or unexpected program/machine behavior.

d. Memory Management Unit

The Memory Management Unit can be programmed to limit access to memory and control registers by different tasks. Violations can be trapped by the MMU and trigger a recovery action.

e. Acceptance Tests

This is a very broad term and can be applied to applications tasks and various components of the operating system such as the task scheduler and dispatcher. The results of the target task are checked for acceptability using some criteria which may range from a single physical reasonableness check such as pitch command not exceeding a certain rate to an elaborate check of certain control blocks to ascertain whether the operating system scheduled all the tasks in a given frame.

It should be noted again that a physical fault can trigger any of these detection mechanisms just as well as a common-mode failure. Therefore, it is necessary to corroborate the syndrome information across redundant channels to ascertain which recovery mechanism to use.

f. Presence Test

Presence test is normally used in the FTPP to detect the loss of synchronization of a single channel due to a physical fault. However, it can also be modified to detect a total loss of synchronization between multiple channels of an FTPP. This is an indication of a common-mode failure.

g. System Virtual Group

The System VG is a redundant VG composed of formally specified and verified PEs running a small formally specified and verified kernel. It is responsible for detecting random and common-mode failures in itself and other VGs. A typical technique is to require a periodic "heartbeat" message to be sent from each VG in the FTPP to the System VG. Failure of a redundant VG to correctly transmit its heatbeat to the System VG implies that the VG has suffered a common-mode failure. This technique also provides some system-level coverage for faults in simplex VGs.

9.3.2. Real-Time Recovery

The recovery from CMF in real time requires that the state of the system be restored to a previously known correct point from which the computation activity can resume. This assumes that the occurrence of the common-mode failure has been detected by one of the techniques discussed earlier and that its source has been identified.

a. Exception Handlers: If a common-mode failure causes an Ada exception or a hardware exception to be raised, then an appropriate exception handler that is written for that abnormal condition can effect recovery. The recovery may involve a local action such as flushing input buffers to clear-up an overflow condition or it may cascade into a more complex set of recovery actions such as restarting a task, a virtual group or the whole system.

b. Task Restart: If the errors from CMF were limited to a single task and did not propagate to the operating system, then only the affected task needs to be restored and/or restarted with new inputs. The state can be rolled back using a checkpointed state from stable storage. Recovery is then effected by invoking an alternate version of the task using the old inputs assuming that the fault was caused by the task software. This is termed the backward recovery block approach. If the fault is caused by a simultaneous transient in all redundant hardware channels then the same task software can be re-executed using old inputs. This is termed temporal redundancy. Alternatively, forward recovery can be effected by restarting the task at some future point in time, usually the next iteration, using new inputs. This assumes that the fault was caused by an input sensitive software that will not repeat with new and different inputs.

c. Virtual Group Restart: In case the CMF resulted in the loss of synchronization, then redundant channels must be re-synchronized before rollback can begin. Furthermore, the state of the virtual group must be restored before resuming computational activity. This is assisted by system VG concept.

d. System Restart: Finally, if all else fails the whole system can be restarted in real time and a new system state established with current sensor inputs.

In order to achieve ultrareliable systems today, some combination of software fault avoidance and software fault tolerance techniques will be necessary in their design. In order to validate these systems, methods for analyzing software reliability must be improved.

10. A Hierarchical Approach to Validation of Fault Tolerant Claims

Another important and expensive area of the Maglev control computer system development is the cost to verify and validate the system. Some means must be defined to provide a reasonable degree of assurance that a given system implementation will in fact perform the required mission functions: these means include the validation and verification processes.

Validation refers to the process of demonstrating that an implemented system correctly performs its specified functions under all reasonably anticipated operational scenarios, fault conditions, computational loads, etc. [30]. This is usually accomplished by an extensive series of online tests during which the system logs many hours of correct operation and during which any latent faults are eliminated. Since human life will depend on the ability of the Maglev control computer system to perform its intended function, the validation process is justifiably complex and arduous.

The verification process demonstrates only that an implemented system meets its specification [30]. Verification is relatively more straightforward than proving that an implemented system performs its intended functions in a real-life setting, because, whereas a mission environment incorporates the innumerable vagaries, uncertainties, and complexities of real life, a specification is ultimately a list of requirements that can be enumerated and checked off during the verification process. Verification is a process that can be performed piecemeal on different parts of a system, often under laboratory conditions, showing that the system meets its specification, and thereby ensuring that the validation process has some functionality with which to begin. However, nobody would want to ride in a Maglev vehicle which had never been "flight" tested but which, we are assured, can be formally proven to meet its specifications. Hence, both verification and validation are important in establishing that the Maglev system actually does what it is designed to do.

Their limitations notwithstanding, specifications are written which serve as a mutually understood representation of the mission designer's understanding of what characteristics the computer system must have to perform the mission's intended functions, and the computer designer's understanding of what requirements the computer system must meet.

When writing a system specification, it is useful to bear in mind the process which will eventually lead to the verification of that specification. A three-tiered approach to system specification can simplify the verification process [31]. The first level includes statements which specify the system architecture and which are independent of the final implementation of the system. The second level includes general statements about the system which are implementation dependent, but which are independent of the application executing on the system, and which can only be verified once a system has been designed and a brassboard implementation of the system is available for testing. The final level includes statements which are application specific and furthermore may depend upon interactions between the application and the system itself. These can only be verified by testing the application specific software and hardware in conjunction with the brassboard system hardware.

The system attributes which form the architectural specification of the system are listed in Table 10-1. These are properties which are usually visible on a high level block diagram of the system, and their presence can be verified by simple inspection. The architectural features of the design which must be minimally present to ensure that the system will have the stated attribute are also shown in the table.

Attribute	**Inspection Checklist**
Byzantine Resilience for *f* faults	3*f*+1 FCRs[†], 2f+1 Inter-FCR Connectivity, f+1 Round Input Distribution, FCR Synchrony
Damage Tolerance	Physical Dispersion of FCRs
Graded Redundancy	Inter-processor communication between FTPs with different levels of redundancy

Table 10-1. Architectural Attributes of Fault Tolerant Control Computer Systems

The attributes which specify properties of the system which can only be verified by testing and model-based analysis once the system has been implemented are listed in Table 10-2. These features are independent of the application which will execute on the control computer system. The features of the design which must be minimally present to ensure that the system will have the stated attribute are also shown in the table.

The attributes which specify properties of the system which are application dependent can only be verified by testing and model-based analysis once the application has been hosted by the control computer system. During verification, these tests can be conducted in a laboratory environment with high fidelity simulations of the system to be controlled. Testing of fault tolerant systems needs to include tests under fault-free conditions as well as in the presence of faults. Fault insertion can be accomplished by designing simulation software which is capable of fault insertion as well as generating "real" hardware faults by resetting processors, turning off power supplies, or designing special fault-insertion

[†] Fault Containment Region

devices to insert faults at the pin and device level. Validation requires extensive field testing of the entire system. The application dependent attributes are listed in Table 10-3.

Attribute	Inspection Checklist
Fault Containment	Dielectric Isolation, Independent Clocking, Independent Power
Error Containment	Majority Vote Error Masking, Monitor Interlock, Watchdog Timer, Timer-Based Pre-emptive Scheduling
Simplex Programming Model	Redundancy Management Independent of Application, RM transparent to application
Low Operating System Overhead	Fault Tolerant Operations Implemented in Hardware (Voting, Synchronization/FT Clock, Communication and Fault Detection)
Interactive Consistency	Identical Processor Design, Bitwise Identical Code, Voted Bitwise Identical Inputs, Voted Bitwise Identical Outputs, Real-Time Error Masking
Reliability, Availability	Low Component Failure Rate, Reconfigurability, Hardware N-fail-op, Graceful Degradation, Variable Number of Processing Sites, Flexible Function Allocation, Function Migration
Maintainability, Low Life Cycle Cost	Built-in Testability, Automated Diagnosability, LRM/LRU Repairability
Common Mode Fault Avoidance	Software Design Methodology, Adherence to Standards, Commercial Processors

Table 10-2. Implementation Dependent Attributes of Fault Tolerant Computer Systems.

Attribute	Inspection Checklist
Real-Time Operation	Low Communication Transport Delay, Delivered Throughput, Effective I/O Bandwidth, Effective Intertask Communication Bandwidth, Task Iteration Rate, Pre-emptive Rate-group Scheduling
I/O Requirements	Use of Standard Interfaces and Communication Protocols
Environmental Requirements	Implementation Technology, Shielding
Physical Constraints	Power Consumption, Weight, Volume
Provisions for Growth	Expandable Memory, I/O Bandwidth, Throughput

Table 10-3. Application Dependent Attributes of Fault Tolerant Computer Systems

In summary, validation is the process of demonstrating that the Maglev control computer system correctly performs its intended functions. Verification is the process of demonstrating that it meets its specifications. Although application dependent specifications, such as throughput, cause validation process to be repeated for each new application, there are certain logical statements that can be made about fault tolerant system attributes that hold true independent of applications, such as Byzantine Resilience. In that sense, once these attributes have been demonstrated, one can say that a system is partially validated independent of the intended application requirements [30].

Due to the impossibility of reproducing all possible scenarios under which a Maglev vehicle will be required to operate, testing alone does not produce estimates of reliability which can satisfy validation requirements for human safety. However, modeling and analysis bridge that gap. Predictive verification [30] is a methodology which combines modeling and analytically obtained results with empirically obtained data to corroborate the predictions produced by the model. First, the system's verifiable attributes are enumerated. In the predictive phase of the verification, these attributes are predicted via performance, reliability, availability models, and cost models. In the corroborative phase, critical model inputs are verified via empirical test and evaluation, or sensitivity studies are performed to obtain bounds on the effects of unverifiable parameters. In addition, quantities predicted by the models which can be empirically verified are measured to corroborate the models' accuracy.

11. Dependability Modeling Techniques

Dependability of the maglev control computer system includes such attributes as reliability, safety, maintainability, and availability. These terms were defined in Section 4. This section discusses the techniques which may be used to model these attributes.

11.1. Reliability Modeling

Reliability modeling plays an important role in the design of fault tolerant systems. When designing a fault tolerant architecture and specifying its a redundancy management approach, a large number of tradeoffs must be made to achieve all the requirements of the system. These requirements include power, weight, volume, real-time performance, and cost, as well as reliability. Since the impact of different design alternatives on reliability is not always intuitively obvious, models have been devised which allow quantitative assessments to be made, thereby allowing designers to make tradeoffs intelligently [37].

The reliability model also makes it possible to determine the sensitivity of the system's reliability to a particular parameter. For example, one can determine the impact of the failure rate of a given technology on the overall reliability of the system. Thus, more effort can be spent on those aspects of a design which have the greatest impact.

Once a fault tolerant architecture and redundancy management approach have been chosen, reliability evaluation becomes a primary concern. A high degree of confidence that a fault tolerant system will meet its reliability goal usually cannot be gained through testing alone, since cost may not allow a sufficient number of test systems to be built and since the failure rates of these systems is so low that it is not possible to subject these systems to a sufficiently lengthy interval of testing prior to their actual use. In such situations, a combination of testing and analysis is necessary to obtain the desired degree of confidence in the reliability of a design. Testing may be used on the system components to obtain failure rates which can then be input to the reliability model which can estimate the reliability of the entire system. Additional test data can be gathered on the redundancy management of the system by injecting faults artificially and observing the response of the fault tolerance mechanisms. Thus, the reliability model can analyze both the system architecture, i.e. how the system's components are interconnected, and its redundancy management approach, i.e. how component failures are detected, identified, and how the system is reconfigured to accommodate these failures.

There are three common methods used to analytically determine system reliability: Monte Carlo simulation, combinatorial models, and Markov models. Each have weaknesses and strengths.

A Monte Carlo simulation [38] can be used to determine reliability by failing components at times distributed according to their failure rates. These simulations are repeated until statistically significant reliability measures are accumulated. An advantage of the simulation approach is that very little knowledge, other than that pertaining to the system to be analyzed, is needed. However, a key difficulty of this method is that for highly reliable systems, and whose reliability can only be estimated analytically, a very large number of simulations are needed to obtain statistically meaningful results. For example, if a system is designed to have a failure probability of 10^{-6} for its mission time, then there would only be one failure out of one million simulations. Many more simulations would be required to obtain a statistically meaningful reliability value. Further, consider the case of a comparison study. If one million simulations are performed and both systems have only one failure, then it is not correct to assume that they both have the same reliability. All that can be said is that their reliabilities cannot be distinguished from each other by a one million sample set. A more subtle difficulty with simulation is that the results are highly dependent on the detail of information used in building the simulation model. During the design phase, details of the operation of the system that are required to develop a meaningful simulation model are usually not available. Therefore, reliability information from the simulation model cannot be used during design of the system.

Historically, combinatorial reliability models [39] have been widely used. Fault-tree analysis [40], for example, has become a standard analytical method for reliability prediction in a wide variety of applications. This analytical technique statistically combines component failure probabilities, based on the system architecture and redundancy management approach, to determine the system reliability. Since there is not explicit simulation of system operation, the combinatorial technique avoids the deficiencies of the Monte Carlo simulation. There are, however, three limitations to this approach. First, the fault tree is constructed to predict the probability (reliability) of a certain event. If it is desired to investigate a different event, for example the reliability of a variety of operating modes, new fault trees have to be constructed. Secondly, it is difficult to include events that have sequence dependencies, such as repairs and explicit modeling of reconfiguration strategies. Even in simple systems, there are often sequence dependencies which are quite subtle. Finally, the nature of the combinatorial analysis requires that all combinations of events for the entire time period must be included. For complex systems, this results in a complicated fault tree that is difficult to validate.

More recently, Markov modeling techniques have been used for reliability prediction [41]. A Markov reliability model calculates the probability of the system being in various states as a function of time. A state in the model represents the system status as a function of both the failed and unfailed components and the system's redundancy management strategy. Transitions from one state to another occur at given transition rates which reflect component failure rates and redundancy management performance. Elements in the model's state vector represents the probability of being in each state at a specified time. Since the Markov model traces the evolution of state probabilities based on the probability of component failure, it is not explicitly simulating the system, and therefore, does not have the associated deficiencies that are found in the Monte Carlo technique. The Markov model represents a system of differential equations. Hence, order dependencies such as repairs and redundancy management processes are included naturally. Further, the differential nature of the model means that all event combinations for a long time period do not need to be explicitly considered; it is only required to model events that happen in a discrete step in time. A drawback to the Markov method is that the state space grows exponentially with the number of components. Techniques have been developed to deal with this problem, resulting in models that are both numerically and conceptually tractable.

The flexibility of Markov models and their ability to model the sequence-dependent events that occur in the redundancy management process inherent in fault-tolerant systems makes them an appropriate analytical tool for evaluating these systems. The modeling of the redundancy management processes can be decomposed from the fault-occurrence model and dealt with at a variety of levels of fidelity. An additional benefit of using Markov models is that the entire state vector, and hence, the reliability of all operating modes is available. The efficiency of solution of the Markov model also makes it possible to determine the sensitivity of the system's reliability to each of the myriad of parameters upon which that reliability depends. These sensitivity analyses enable the system's designers to focus their limited resources on those aspects of the design which have the greatest impact on system reliability.

Given the inherent link between accumulated failures and the eventual degradation of performance in a fault-tolerant system, a Markov reliability model can be used to evaluate state-dependent system performance on a probabilistic basis as a function of time. For example, an evaluation of the probability of a minimum throughput level over an entire mission can be obtained by operating on the model's state vector. This combination of reliability measures with performance measures is called "performability".

11.2. Safety Modeling

The fundamental concept of safety analysis is that a system can fail in one of two ways: one way is designated as safe and the other is designated as unsafe. Each designation is uniquely defined for a given application. A simple system with one hardware model which has self-diagnostics can illustrate failed safe and failed unsafe states. Safe failures are defined as those faults which are detected by the self-diagnostics before they manifest themselves during normal system operation. Unsafe failures are those which occur during normal operation and result in system loss. Safety can be modeled using Markov models by splitting the system failed state into two separate states. The first state is labelled failed safe and the second failed unsafe. The safety of the system, S(t), is defined as the probability that the system will either perform correctly or fail in a safe manner. For the simple system described above, the safety of the system can be written as

$$S(t) = p_O(t) + p_{FS}(t)$$

where $p_O(t)$ is the probability of being in the operational state at time t and $p_{FS}(t)$ is the probability of being in the failed safe state at time t. If the failure rate of this system described above is given as λ, and the self-diagnostics provide a coverage of C, the safety of the system can be shown to be given by

$$S(t) = p_O(t) + p_{FS}(t) = C + (1 - C)e^{-\lambda t}$$

Some interesting observations can be made about this result. At time t = 0, the safety of the system is 1, i.e. the system is perfectly safe. As time approaches infinity, however, the safety approaches

$$S(\infty) = C$$

In other words, the safety of the system is highly dependent on the level of fault detection coverage. If the fault detection coverage is non-existent, i.e. C = 0, then the system will fail in an unsafe manner. If the coverage is perfect, i.e. C = 1, then the system is perfectly safe. This dependence on fault detection coverage typically extends to more complex systems as well. In general, the safety of a system will depend on the fault detection coverage mechanisms which are designed into the system. This concept has been applied in the development of the onboard control computer architecture and the fail-safe communication protocol described in Section 6.6.

11.3. Availability Modeling

Availability is defined as the probability that a system is able to perform its tasks at the instant of time t. Availability can be determined empirically, once a system has been in operation for a sufficient amount of time and has experienced some failures and repairs. This calculation is simply the ratio of the total time that a system has been operational to the time elapsed since the system was put into operation, i.e. the percentage of time that a system is available to perform its expected tasks. Unfortunately, this method is not useful for forecasting the availability of a system under development. Other techniques exist which allow availability considerations to be factored into the design process. Two methods are commonly used to estimate the availability of a system. The first yields the steady-state availability and depends on the MTTF and MTTR. The second method uses the failure rates and repair rates in a Markov model to calculate the availability as a function of time.

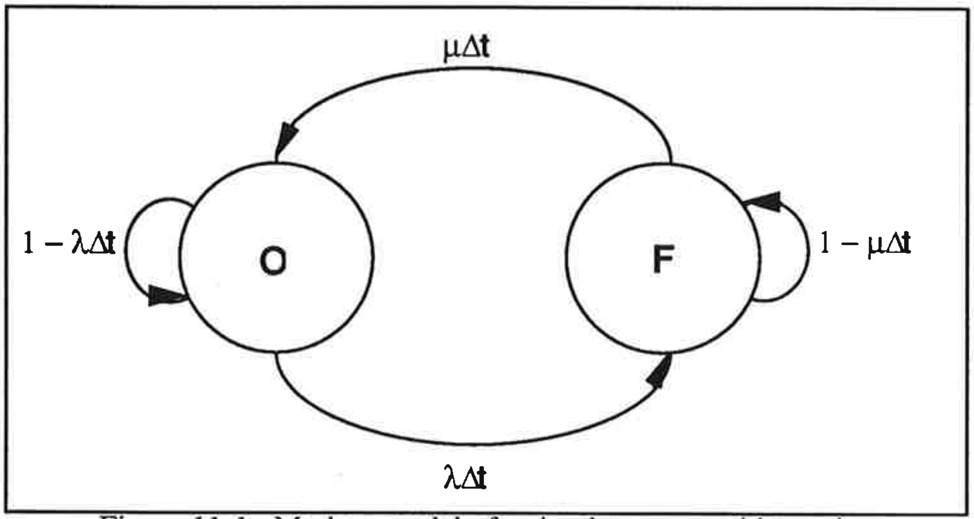


Figure 11-1. Markov model of a simple system with repair.

If the average system experiences N failures during its lifetime, then the total time that the system is operational is N(MTTF) hours, and the non-operational or down-time is N(MTTR) hours. The average, or steady-state availability, A_{SS}, is

A_{SS} = N(MTTF)/(N(MTTF) + N(MTTR)) = MTTF/(MTTF + MTTR)

If the failure rate of a system is λ and the repair rate is μ, then MTTF = $1/\lambda$ and MTTR = $1/\mu$. Then

$A_{SS} = (1/\lambda)/(1/\lambda + 1/\mu) = \mu/(\lambda + \mu)$

The discrete time Markov model of a system with repair is a two-state model, with one state representing the system in its operational mode and one state representing the system in its failed mode. The Markov model for this system is in fact the model required to calculate the availability of a system. Consider, for example, a single computer which has a constant failure rate λ and a constant repair rate μ. During the time interval Δt, the computer will have a probability of failure of $\lambda\Delta t$ and a probability of repair of $\mu\Delta t$, as shown in Figure 11.2-1. The figure also shows that the probability of an operational computer remaining operational during the interval Δt is $1 - \lambda\Delta t$ and the probability of a failed computer remaining in a failed state is $1 - \mu\Delta t$. The continuous time solution for this Markov model for $p_O(t)$, i.e. the probability that the system is operational at time t, is given by

$p_O(t) = \mu/(\lambda + \mu) + \lambda/(\lambda + \mu)e^{-(\lambda+\mu)t}$

As time approaches infinity, $p_0(\infty)$ approaches the constant value of $\mu/(\lambda + \mu)$, which can be seen to be the steady state availability of this simple system.

11.4. Maintainability Modeling

As defined in Section 2.2, maintainability, represented by M(t), is the probability that a failed system will be restored to working order within a specified time, given that it failed at time t = 0. In other words, M(t) is the probability that the system can be repaired in a time less than or equal to t. Clearly, the repair rate μ, that is the average number of repairs which can be performed per unit of time, is an important parameter for calculating maintainability. The inverse of μ is the MTTR, which is the average time required to perform a single repair. An expression for the maintainability of a system related to μ can be derived as follows. Suppose we have N systems into each of which we inject faults and begin repairs both starting at time t = 0. Then M(t) is given by

$$M(t) = N_r(t)/N = N_r(t)/(N_r(t) + N_{nr}(t))$$

where $N_r(t)$ is the number of systems that have been repaired at time t and $N_{nr}(t)$ is the number of system which are not yet repaired, and $N = N_r(t) + N_{nr}(t)$. The probability that the system will *not* be repaired in time t is given by

$$1 - M(t) = N_{nr}(t)/N$$

If we differentiate M(t) with respect to time we obtain

$$dN_r(t)/dt = N\, dM(t)/dt$$

An expression for μ can be based on the derivative of $N_r(t)$ as follows. The derivative of $N_r(t)$ is the instantaneous repair rate of the system. At time t there are $N_{nr}(t)$ systems which have still not been repaired. If we divide $dN_r(t)/dt$ by $N_{nr}(t)$ we obtain an expression called the repair rate function which is assumed to have a constant value of μ,

$$\mu = (1/\, N_{nr}(t))dN_r(t)/dt$$

Substituting the value of $dN_r(t)/dt$ from a previous equation, we obtain

$$\mu = (N/\, N_{nr}(t))dM(t)/dt \quad \text{or} \quad dM(t)/dt = \mu\, (N_{nr}(t)/N)$$

However, $N_{nr}(t)/N$ is (1 - M(t)). Substituting, we can now write

$$dM(t)/dt = \mu\, (1 - M(t)).$$

The solution to this differential equation is

$$M(t) = 1 - e^{-\mu t}$$

This relationship for M(t) has the characteristics needed in a maintainability function. First, if the repair rate is zero, the maintainability is also zero since the system cannot be repaired in any length of time. Second, if the repair rate is infinite, the maintainability is perfect and can be accomplished in zero time. Another value which can be obtained from this function is the maintainability at time t = MTTR.

$M(t = MTTR) = 1 - e^{-\mu 1/\mu} = 1 - e^{-1} = 0.632.$

which means that there is a 63 percent probability that the system will be repaired in a time less than or equal to MTTR.

Thus, it can be seen that the repair rate plays a crucial role in the maintainability of a system. Various levels of repair will have widely differing repair rates. Repairs which can be made on site will have a faster repair rate than those which require depot or factory maintenance. Clearly, for a system to be highly available, it must either have a very high level of maintainability, or have sufficient spare capacity, i.e. redundancy, to allow operation to continue in the presence of faults. When this is the case, maintenance can be carried on in an orderly fashion, without disruption of service. Although maintainability is still important, it will not be the driving parameter in the ability of a system to perform its function in a dependable way.

12. SVA Onboard Control Computer Dependability Analysis

The onboard vehicle control computer (OCC) for the Maglev Transportation System is described in great detail in Section 6.6. Since the onboard system performs functions which are safety-critical, e.g. velocity control, the probability of this system failing in an unsafe manner must be less than 10^{-9} per hour of operation. Furthermore, since a failure to complete a mission, i.e. breaking down between stations, even in a safe manner, can have a devastating impact on total system availability, and a consequent negative impact on ridership, the unreliability of the OCC must be less than 10^{-6} per hour. Finally, the unavailability of the OCC must be no more than 10^{-3} in order to compare favorably with the availability achieved by the onboard control systems of today's commercial aircraft. The OCC discussed in Section 6.6 has a simple but elegant communication protocol which ensures safety but guards against false alarms thus directly protecting the reliability of the vehicle during a mission and indirectly maintaining the availability of the system as a whole.

The primary objective of this section is to produce a first order estimate of the reliability, availability, safety, and maintainability of the OCC. In addition there are two secondary objectives. The first of these is to specify a system which is cost-effective by not allocating more redundancy than needed to meet the RMAS requirements. Thus, it is necessary to determine the correct redundancy level of network elements, processors, and I/O busses which must be fully operational when a vehicle leaves a station to ensure that it will safely and reliably reach its destination. This level of redundancy is called the minimum dispatch complement or MDC. The second objective is to determine the number of spare components needed to ensure the necessary availability of the OCC in keeping with reasonable maintenance schedule.

In order to meet these objectives, it is necessary to specify a mission state diagram for the Maglev vehicle. This diagram identifies the various states the OCC can occupy and the events which drive it from one state to the other. Based on this diagram a detailed Markov model for reliability and safety can be constructed and analyzed. The state diagram for the OCC is shown in Figure 12-1. The state diagram in this figure is based the one shown in Figure 2-2. It has been streamlined to focus on the states of greatest interest for safety, reliability and availability, and modified to accommodate automatic recovery from the fail-safe state.

The mission state diagram of the OCC shows five principle states. Of special interest are the fail-safe and fail-unsafe states. There are only two events which can drive the system to a fail-unsafe state: coincident hardware faults or common mode software faults. The probability of the former is addressed below. The probability of the latter cannot be stated with certainty, however, this probability can be reduced by various programming and testing techniques. A system failure which results in a fail-safe state does not necessarily require push-recovery to rescue the vehicle. If the failure occurs as a result of a transient fault, which is by far the dominant failure mode, the system may be restarted and the mission completed, albeit not on schedule. It has been noted that the public can accept outage times of short duration (a few minutes), but outages longer than an hour have a very negative impact. Hence this ability to recover from transient faults is very important to public acceptance of this mode of travel.

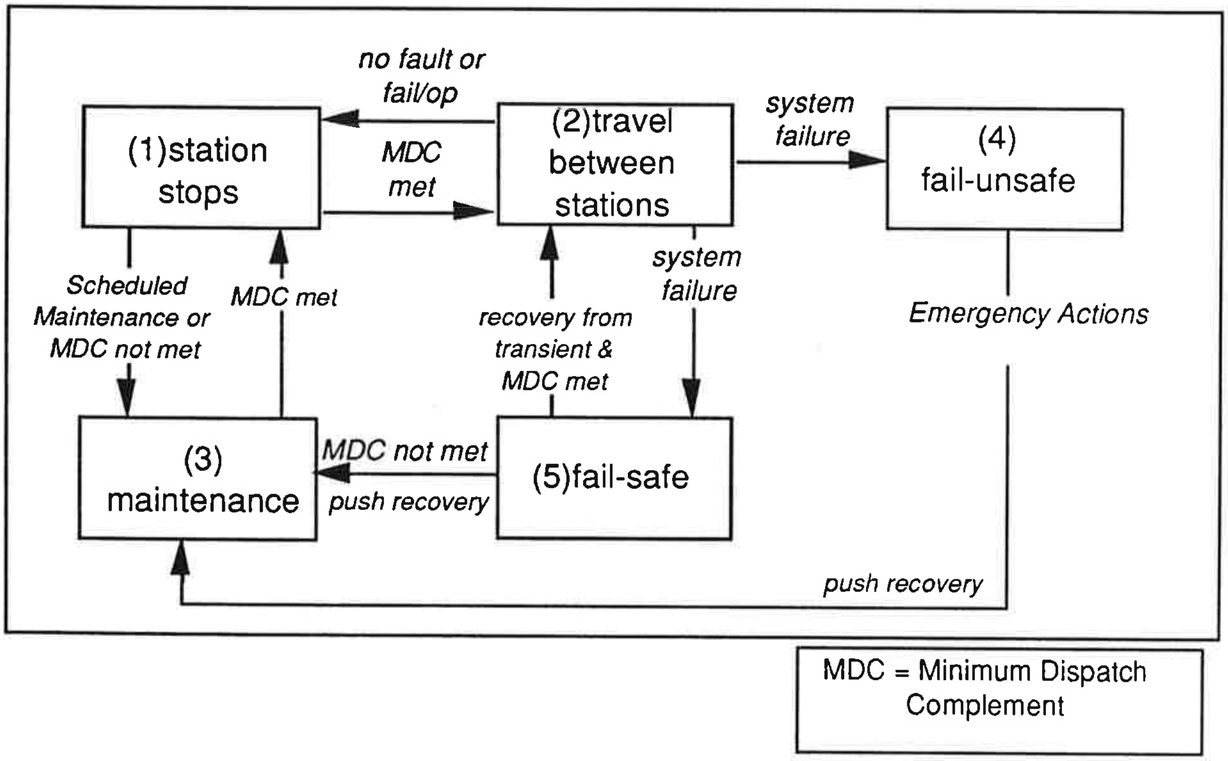


Figure 12-1. Mission State Diagram of the Onboard Control Computer (OCC).

12.1. Approach

The objective of the analysis presented below is to estimate the reliability, safety, and availability of the SVA Onboard Control Computer (OCC). First, a Markov model of the OCC is constructed to determine OCC reliability and safety as a function of Minimum Dispatch Complement (MDC). This model can be used to select an MDC which is sufficient to meet the Maglev reliability and safety requirements. Next, a Markov model of the OCC is constructed to determine OCC availability as a function of MDC and spare components. Once an appropriate MDC has been determined, the availability model can be used to determine the number of spare components required to meet OCC availability requirements.

For reasons of clarity as well as to stress an approach which uses only as much redundancy as required in order to remain cost-effective, a minimal FTPP configuration will be modeled and analyzed first, followed by more a complex configuration. Thus, a reliability and safety analysis of a triplex OCC is developed first, followed by the model of a quadruplex OCC. Subsequently, an availability model applicable to both triplex and quadruplex OCCs is developed and analyzed. It should be noted that all models presented below are a subset of one large model. However, in this document we present them separately for clarity of exposition.

12.2. Assumptions

Several assumptions made to facilitate the analysis are listed below.

1. The components of the OCC exhibit constant failure rates which are obtained from MIL HDBK-217E, "United States Department of Defense Military Standardization Handbook: Reliability Prediction of Electronic Equipment."
2. The OCC is able to reconfigure from faults with a constant reconfiguration rate. This assumption results in conservative estimates of reliability and availability.
3. Any failure in a channel will cause the channel to be taken offline. Since some failures may not require a channel to be taken offline, this assumption also results in conservative estimates of reliability and availability.
4. The FTPP provides unity coverage for failures which result in a quadruplex transitioning to a triplex, a triplex transitioning to a duplex, or a duplex entering the simplex fail-stop mode. This is a reasonable assumption for Byzantine resilient systems.
5. The reliability of the zone controller is not modelled.
6. The OCC begins the mission with an MDC which is known with unity probability to be healthy. This is also a reasonable assumption for Byzantine resilient systems, which are accurately diagnosable.

12.3. OCC Reliability and Safety Model

12.3.1. Triplex OCC

This section analyzes the reliability and safety of a triplex OCC (3 PEs and 3 NEs) for a single mission under two policies of recovering from transient faults. In the first policy, the OCC attempts no recovery from transient faults. The advantage of this policy is that exposure to coincident faults during lengthy transient recovery is avoided. The disadvantage of this policy is that unreliability due to exhaustion of redundancy is not minimized. In the second policy, the OCC attempts recovery from transient faults. The advantage of this policy is that unreliability due to exhaustion of redundancy is reduced. The disadvantage of this policy is that exposure to second fault while recovering from a transient fault (a process requiring roughly one second) increases the probability of unsafe failure. A second disadvantage is the fact that during the recovery period normal execution of real-time control processes may be suspended. However, this effect can be mitigated for critical functions by using a special recovery process.

12.3.1.1. Triplex OCC with No Transient Recovery

The states which the OCC can assume fall into four broad categories. In the "0 faults" category, the OCC has not yet suffered any faults. In the "1 fault" category, the OCC is either in the process of recovering from a fault, or has successfully recovered from a single fault and is operating in a degraded redundancy mode. In the "2 faults" category, the OCC is either in the process of executing a safe shutdown, or has performed a safe shutdown. Two kinds of safe shutdowns are possible under a no-transient-recovery redundancy management policy: “soft” shutdown and “hard” shutdown. In a soft shutdown state, although the vehicle has been shut down by an OCC failure, the OCC faults are not permanent. Therefore, after the vehicle has been safely shut down, the OCC can re-initialize itself and resume operation as a fault tolerant controller, and the vehicle can thereafter be restarted without external assistance. Thus this state does not contribute to vehicle unreliability, although it will contribute to a late arrival at the destination station. In the hard shutdown states, the OCC has shut down due to permanent faults and can not re-initialize itself and resume operation as a fault tolerant controller, so this state does contribute to vehicle unreliability. External activity such as a tow or physical replacement of an OCC component is required before the vehicle can be cleared from the guideway. Finally, in the "unsafe failure" category, the OCC is unable to safely control or shut down the vehicle.

The Markov model of the triplex OCC managed under the no-transient-recovery policy is shown in Figure 12-2. Inside each circle denoting a possible OCC state is either an ordered pair or a textual comment. In states denoted by ordered pairs (x,y), x refers to the number of transient faults which have occurred and y refers to the number of permanent faults which have occurred. In states containing textual comments, RP indicates that it is a state in which the OCC is reconfiguring from a permanent fault, and RT indicates that it is a state in which the OCC is reconfiguring from a transient fault. The model’s states and transitions are described in Table 12-1.

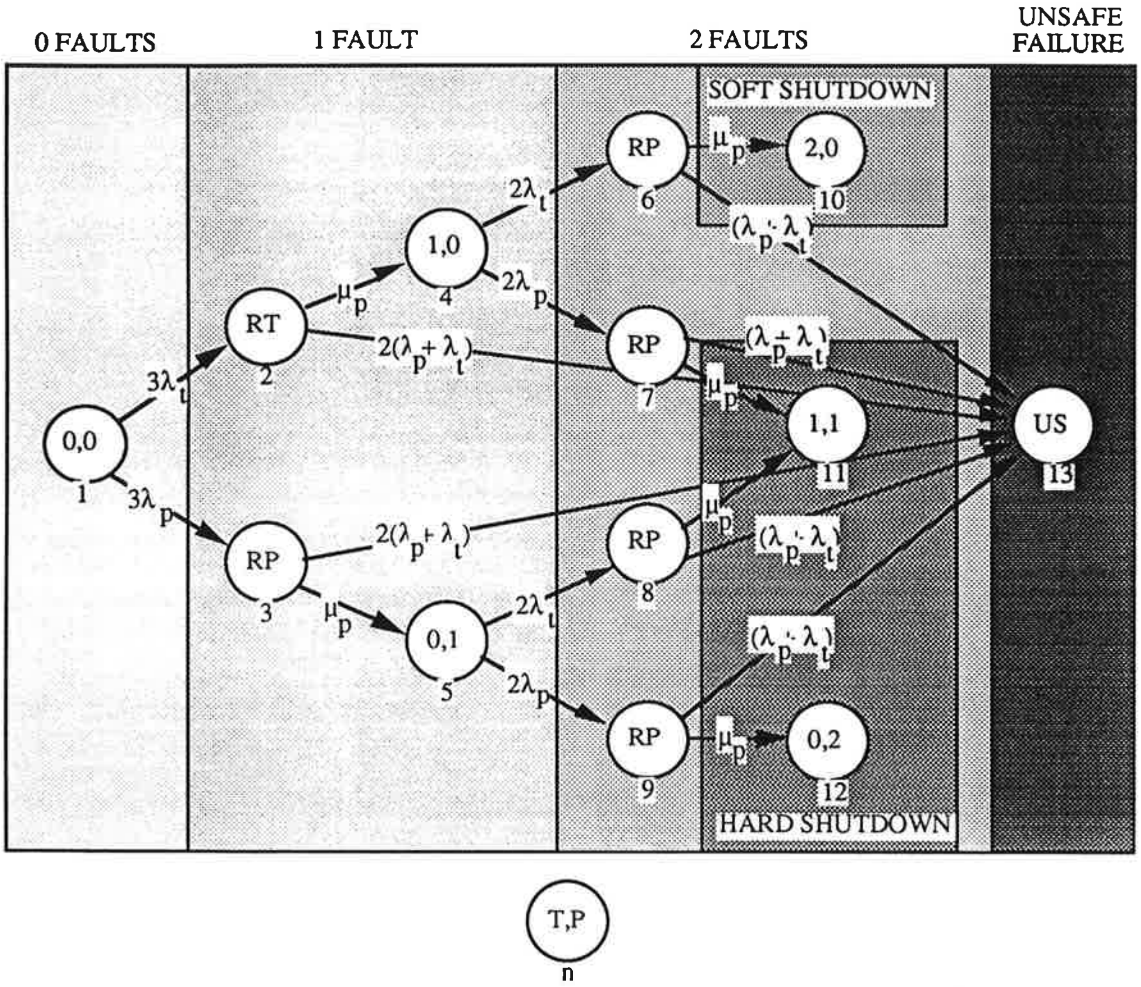


Figure 12-2. Markov Model of Triplex OCC With No Transient Recovery

State	Description	Transitions From State
1	No faults. Fully operational triplex OCC.	-Transient faults occur at rate $3\lambda_t$ to State 2. -Permanent faults occur at rate $3\lambda_p$ to State 3.
2	Reconfiguring from transient fault.	-Successful reconfiguration from transient faults occurs at rate μ_p to State 4. -Faults occur during reconfiguration at rate $2(\lambda_p + \lambda_t)$to State 13.

3	Reconfiguring from permanent fault.	-Successful reconfiguration from permanent faults occurs at rate μ_p to State 5. -Faults occur during reconfiguration at rate $2(\lambda_p + \lambda_t)$to State 13.
4	Operational duplex OCC containing one channel suffering from one transient fault. This state can only be entered if transient recovery is not employed.	-Transient faults occur at rate $2\lambda_t$ to State 6. -Permanent faults occur at rate $2\lambda_p$ to State 7.
5	Operational duplex OCC containing one channel suffering from one permanent fault.	-Transient faults occur at rate $2\lambda_t$ to State 8. -Permanent faults occur at rate $2\lambda_p$ to State 9.
6	Duplex containing one transiently faulted channel has suffered a second transient fault and the duplex is in the process of executing the fail safe procedure.	-Successful fail-safe procedure occurs at rate μ_p to State 10. (Note that this does not take as long as transient recovery since a duplex can not recover from a transient fault.) -An additional fault occurs during the fail-safe procedure at rate $(\lambda_p + \lambda_t)$ assumed to cause the fail-safe procedure to fail, thus causing transition to state 13.
7	Duplex containing one transiently faulted channel has suffered a second, permanent, fault and is in the process of executing the fail safe procedure.	-Successful fail-safe procedure occurs at rate μ_p to State 11. -An additional fault occurs during the fail-safe procedure at rate $(\lambda_p + \lambda_t)$ assumed to cause the fail-safe procedure to fail, thus causing transition to state 13.

8	Duplex containing one permanently faulted channel has suffered a second, transient, fault and is in the process of executing the fail safe procedure.	-Successful fail-safe procedure occurs at rate μ_p to State 11. -An additional fault occurs during the fail-safe procedure at rate $(\lambda_p + \lambda_t)$ assumed to cause the fail-safe procedure to fail, thus causing transition to state 13.
9	Duplex containing one permanently faulted channel has suffered a second permanent fault and is in the process of executing the fail safe procedure.	-Successful fail-safe procedure occurs at rate μ_p to State 12. -An additional fault occurs during the fail-safe procedure at rate $(\lambda_p + \lambda_t)$ assumed to cause the fail-safe procedure to fail, thus causing transition to state 13.
10	The OCC has successfully executed the fail-safe procedure as a result of two consecutive transient faults, and has entered a "soft shutdown" state.	-None
11	The OCC has successfully executed the fail-safe procedure as a result of one transient fault followed by one permanent fault (or vice versa–the model does not differentiate between the two possible orders of permanent and transient fault arrivals), and has entered a "hard shutdown" state.	-None
12	The OCC has successfully executed the fail-safe procedure as the result of two consecutive permanent channel faults, and has entered a "hard shutdown" state.	-None

13	In this state the OCC has suffered an uncontrollable failure due to the arrival of a second fault while the OCC is busy recovering from a previous one. In this state the OCC can not reliably control the vehicle nor execute the fail-safe procedure.	-None

Table 12-1. Markov Model of Triplex OCC With No Transient Recovery –State Description

12.3.1.2. Triplex OCC with Transient Recovery

The description of a Markov model of the triplex OCC with transient fault recovery follows. Note that it is a subset of the previous model in that states 4, 6, 7, and 10 are missing. Also note the absence of the soft shutdown failure mode, since, because the OCC can recover from transient faults, it can not be shut down by a series of transient faults. The model is depicted in Figure 12-3 and the model's states and transitions are described in Table 12-2.

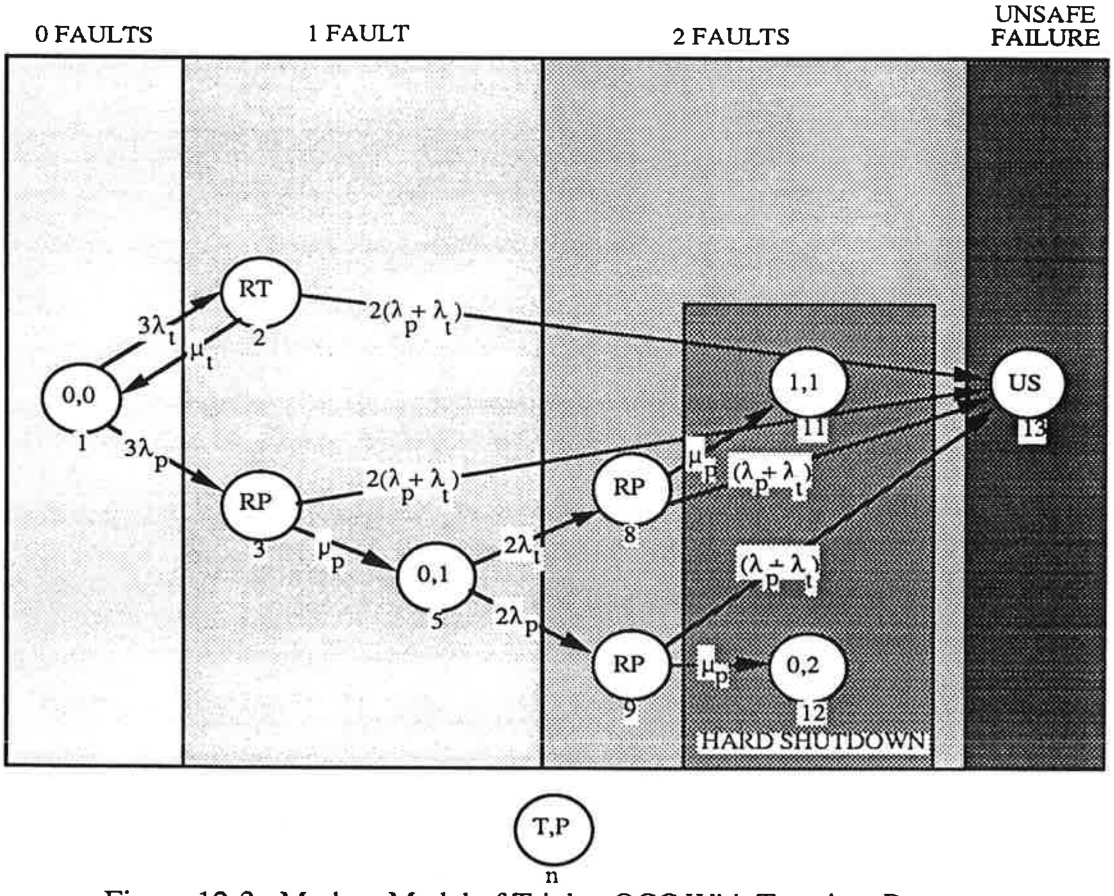


Figure 12-3. Markov Model of Triplex OCC With Transient Recovery

State	Description	Transitions From State
1	No faults. Fully operational triplex OCC.	-Transient faults occur at rate $3\lambda_t$ to State 2. -Permanent faults occur at rate $3\lambda_p$ to State 3.
2	Reconfiguring from transient fault.	-Successful reconfiguration from transient faults occurs at rate μ_t to State 1. -Faults occur during reconfiguration at rate $2(\lambda_p + \lambda_t)$to State 13.
3	Reconfiguring from permanent fault.	-Successful reconfiguration from permanent faults occurs at rate μ_p to State 5. -Faults occur during reconfiguration at rate $2(\lambda_p + \lambda_t)$to State 13.
4	This state cannot be occupied under transient recovery policy.	-None.
5	Operational duplex OCC containing one channel suffering from one permanent fault.	-Transient faults occur at rate $2\lambda_t$ to State 8. -Permanent faults occur at rate $2\lambda_p$ to State 9.
6	This state cannot be occupied under transient recovery policy.	-None.
7	This state cannot be occupied under transient recovery policy.	-None.
8	Duplex containing one permanently faulted channel has suffered a second, transient, fault and is in the process of executing the fail safe procedure.	-Successful fail-safe procedure occurs at rate μ_p to State 11. -An additional fault occurs during the fail-safe procedure at rate $(\lambda_p + \lambda_t)$ assumed to cause the fail-safe procedure to fail, thus causing transition to state 13.

9	Duplex containing one permanently faulted channel has suffered a second permanent fault and is in the process of executing the fail safe procedure.	-Successful fail-safe procedure occurs at rate μ_p to State 12. -An additional fault occurs during the fail-safe procedure at rate $(\lambda_p + \lambda_t)$ assumed to cause the fail-safe procedure to fail, thus causing transition to state 13.
10	This state cannot be occupied under transient recovery policy.	-None.
11	The OCC has successfully executed the fail-safe procedure as a result of one transient fault followed by one permanent fault (or vice versa–the model does not differentiate between the two possible orders of permanent and transient fault arrivals), and has entered a "hard shutdown" state.	-None
12	The OCC has successfully executed the fail-safe procedure as the result of two consecutive permanent channel faults, and has entered a "hard shutdown" state.	-None
13	In this state the OCC has suffered an uncontrollable failure due to the arrival of a second fault while the OCC is busy recovering from a previous one. In this state the OCC can not reliably control the vehicle nor execute the fail-safe procedure.	-None

Table 12-2. Markov Model of Triplex OCC With Transient Recovery–State Description

12.3.1.3. Analytical Results for Triplex OCC

The above models were evaluated for mission times ranging from 1 to 5 hours using the values for the failure and reconfiguration rates shown in Table 12-3.

Rate	Value
Permanent Failure Rate, λ_p	1 x 10^{-4} per hour (10,000 hours MTBF)
Transient Failure Rate, λ_t	1 x 10^{-3} per hour (1,000 hours MTBF)
Reconfiguration Rate From Permanent Faults, μ_p	5.5 x 10^{6} per hour (20 msec)
Reconfiguration Rate From Transient Faults, μ_t	2.8 x 10^{4} per hour (1 sec)

Table 12-3. Failure and Reconfiguration Rates

The results from the model evaluations are presented in Table 12-4 for mission times of 1, 2.5, and 3.5 hours.

Mission Time, hours	Probability of Soft Shutdown (State 10)	Probability of Hard Shutdown (States 11 and 12)	Probability of Unsafe Failure (State 13)
No Transient Recovery			
1	7.5 x 10^{-7}	1.5 x 10^{-7}	6.6 x 10^{-13}
2.5	1.2 x 10^{-5}	2.5 x 10^{-6}	2.6 x 10^{-12}
3.5	2.7 x 10^{-5}	5.7 x 10^{-6}	3.9 x 10^{-12}
With Transient Recovery			
1	0	8.3 x 10^{-8}	1.2 x 10^{-10}
2.5	0	1.3 x 10^{-6}	4.7 x 10^{-10}
3.5	0	3.0 x 10^{-6}	7.1 x 10^{-10}

Table 12-4. Reliability and Safety Results for Triplex OCC

When the triplex OCC is operated without transient fault recovery, the dominant failure mode is soft shutdown. Recall that, because soft shutdowns are caused by transient fault accumulation, the OCC can re-initialize from the transient faults after shutting down the vehicle, and subsequently restart and safely control the vehicle. Hard shutdowns are less likely than soft shutdowns, and are caused by hard fault accumulation. Thus, if the OCC experiences a hard shutdown, the vehicle must stop and can not be restarted under OCC control, even after OCC re-initialization. Finally, the probability of unsafe failure is much less likely than either of the two safe shutdown modes. In this case, unsafe failure is primarily caused by a second fault occurring while the OCC is in the process of diagnosing and reconfiguring from a previous fault.

When the triplex OCC is operated with transient fault recovery, the soft shutdown failure mode is nonexistent. In addition, the hard shutdown failure probability is reduced to a lower value than in the no-transient-recovery case, because in the former, hard shutdown is caused in part by combinations of unrecovered transient faults and hard faults, whereas in the latter, there are no unrecovered transient faults and hard shutdown thus occurs solely due to hard faults. Finally, because transient recovery consumes a significantly longer period of time than a policy which does not recover transient faults, the period of exposure to coincident faults is increased. This significantly increases the probability of unsafe failure.

The general engineering tradeoff highlighted by these models is as follows: the triplex OCC can be operated without transient fault recovery to achieve extremely high safety levels, at the cost of increasing the occurrence of soft and hard shutdowns. Alternatively, the OCC can be operated with transient fault recovery to eliminate soft shutdowns and reduce hard shutdowns by an order of magnitude, at the cost of increasing the probability of unsafe failure.

12.3.2. Quadruplex OCC

Markov models for quadruplex OCC reliability and safety were developed for the two transient fault recovery policies described above for the triplex OCC.

12.3.2.1. Quadruplex OCC with No Transient Recovery

The Markov model for a quadruplex OCC with no transient recovery is shown in Figure 12-4. The quadruplex state descriptions and transitions are sufficiently similar to those of the triplex model that they are not presented in tabular format.

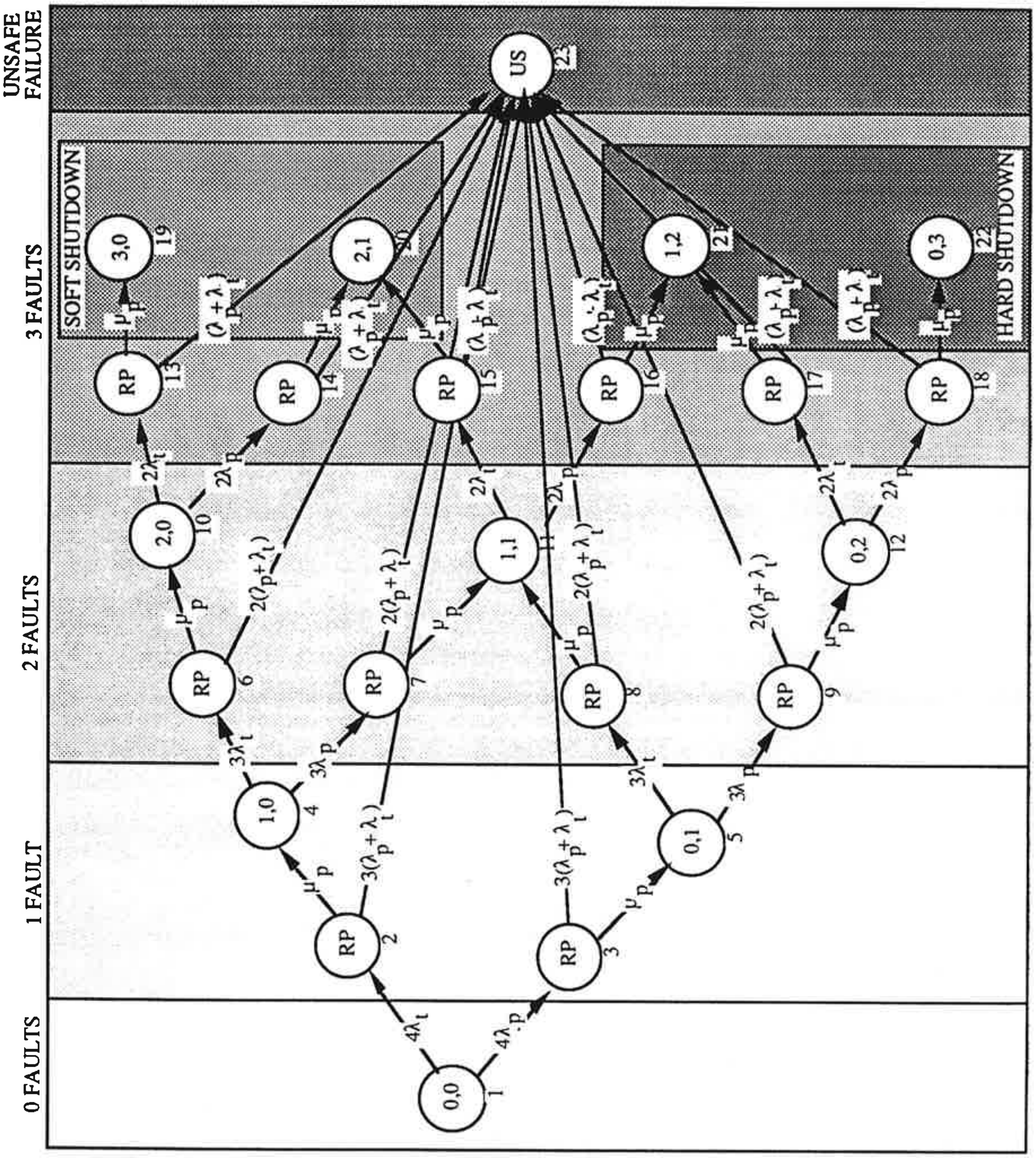


Figure 12-4. Markov Model of Quadruplex OCC With No Transient Recovery.

12.3.2.2. Quadruplex OCC with Transient Recovery

The Markov model for a quadruplex OCC with transient recovery is shown in Figure 12-5.

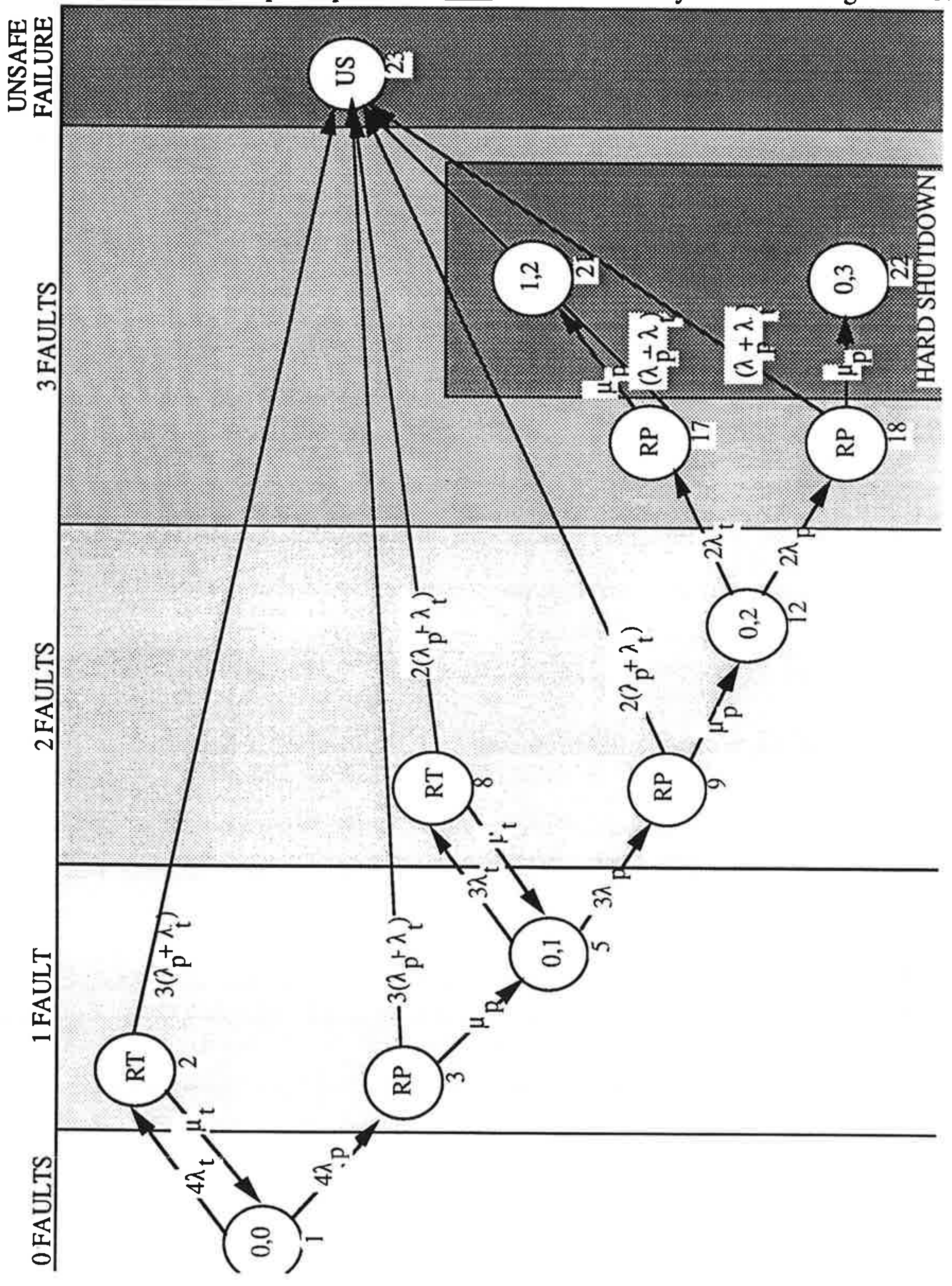


Figure 12-5. Markov Model of Quadruplex OCC With Transient Recovery.

12.3.2.3. Analytical Results for Quadruplex OCC

The results from the model evaluations are presented below for mission times of 1, 2.5, and 3.5 hours.

Mission Time, hours	Probability of Soft Shutdown (States 19 and 20)	Probability of Hard Shutdown (States 21 and 22)	Probability of Unsafe Failure (State 23)
No Transient Recovery			
1	6.5×10^{-10}	1.5×10^{-11}	1.3×10^{-12}
2.5	4.2×10^{-8}	9.9×10^{-10}	5.3×10^{-12}
3.5	1.4×10^{-7}	3.3×10^{-9}	7.9×10^{-12}
With Transient Recovery			
1	0	5.5×10^{-12}	2.4×10^{-10}
2.5	0	3.5×10^{-10}	9.4×10^{-10}
3.5	0	1.2×10^{-9}	1.4×10^{-9}

Table 12-5. Reliability and Safety Results for Quadruplex OCC

The probability of unsafe shutdown for a quadruplex OCC is slightly higher than that of the triplex OCC because the quad's added hardware increases the overall failure rate and, consequently, the probability of coincident faults. However, the probability of both hard and soft shutdowns is significantly reduced due to the increased redundancy of the quadruplex. In addition, the quadruplex OCC analysis yields general conclusions similar to those of the triplex OCC analysis. A quadruplex OCC can be operated without transient fault recovery to achieve extremely high safety levels at the cost of increasing the occurrence of soft and hard shutdowns, or it can be operated with transient fault recovery to eliminate soft shutdowns and reduce hard shutdowns by an order of magnitude at the cost of increasing the probability of unsafe failure.

12.3.3. Recommended Minimum Dispatch Complement for the SVA OCC

To assist in selecting an MDC appropriate for the Maglev SVA, portions of the reliability and safety analysis are reproduced below for a representative mission time of 1 hour (different mission times in the same range yield similar conclusions).

Both the triplex and quadruplex OCC configurations exceed the safety requirement of 10^{-9} per hour. However, because the transient-recovery redundancy management options for both configurations result in unsafe failure probabilities which are uncomfortably close to the OCC's safety requirement, this analysis implies that the no-transient-recovery option should be preferred for safety reasons.

Redundancy Level (MDC)	Probability of Soft Shutdown	Probability of Hard Shutdown	Probability of Unsafe Failure
No Transient Recovery			
3	7.5×10^{-7}	1.5×10^{-7}	6.6×10^{-13}
4	6.5×10^{-10}	1.5×10^{-11}	1.3×10^{-12}
With Transient Recovery			
3	0	8.3×10^{-8}	1.2×10^{-10}
4	0	5.5×10^{-12}	2.4×10^{-10}

Table 12-6. Comparison of Triplex and Quadruplex OCC for 1-Hour Mission.

The triplex OCC without transient recovery marginally meets the OCC's reliability specification (i.e., probability of hard or soft shutdown) of 10^{-6} per hour, while the additional redundancy of a quadruplex OCC enables it to exceed this requirement with a wide margin. Because the triplex OCC only marginally meets the reliability requirement, this analysis leads to the selection of a quadruplex OCC operated without transient fault recovery as the baseline OCC configuration.

12.4. OCC Availability Model

The availability achieved by a given OCC configuration depends on the number of FCRs in the OCC after a maintenance action, the MDC, and the time between regularly scheduled maintenance actions. A maintenance action restores the OCC to a configuration denoted the *nominal configuration* by replacing all faulty FCRs. The nominal configuration includes at least as many FCRs as the MDC, and can include spare FCRs to increase availability. The previous reliability and safety analysis indicates that the MDC of the OCC must consist of four FCRs in order for it to meet the reliability and safety requirements. The vehicle can not begin a trip and meet these requirements unless the OCC meets its MDC requirements. Thus an OCC nominal configuration can contain either four or five FCRs. The time between regularly scheduled maintenance actions is a free variable which can be determined based on the OCC's maximum allowable unavailability as determined by the model below.

The Markov model for the OCC availability is shown in Figure 12-6. It is assumed that no maintenance is performed except at regularly scheduled periods. Multiple missions will typically take place between regularly scheduled maintenance events. It is also assumed that the failure rates for the availability and reliability models are identical since the OCC is in an identical environment in both cases. The model is used to compute availability over the time period beginning immediately after a regularly scheduled maintenance action, at which time it is brought up to its nominal configuration of either four or five FCRs. If the nominal configuration contains five FCRs, the model is initialized with unity probability in state 1 and zero probabilities in states 2-6. If the nominal configuration contains four FCRs, the model is initialized with zero probability in state 1, unity probability in state 2, and zero probabilities in states 3-6. After restoration to its nominal configuration by a maintenance action, the OCC is put into operation, during which transient and permanent faults may occur. For the purposes of availability modeling, it is assumed that recovery from transient faults is always successful since recovery can always be accomplished either between missions (i.e., at a scheduled stop) or after a transient-induced soft shutdown. The OCC continues operation until either the next regularly scheduled maintenance action or until it becomes unavailable by being unable to meet the MDC requirement.

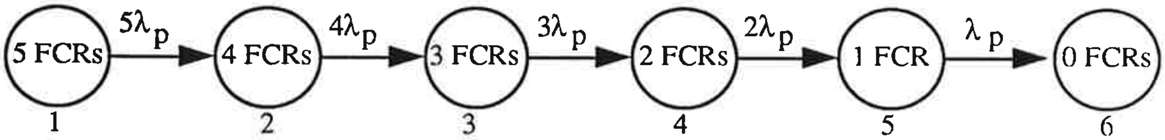


Figure 12-6. Availability Markov Model of OCC With Transient Recovery.

The Markov model states and transitions are defined in Table 12-7.

State	Description	Transitions From State
1	OCC containing 5 FCRs is operational. OCC is available if MDC = 3, 4, or 5.	-Permanent faults occur at rate $5\lambda_p$ to State 2.
2	OCC containing 4 FCRs is operational. OCC is available if MDC = 3 or 4.	-Permanent faults occur at rate $4\lambda_p$ to State 3.
3	OCC containing 3 FCRs is operational. OCC is available if MDC = 3.	-Permanent faults occur at rate $3\lambda_p$ to State 4.
4	OCC containing 2 FCRs is operational. OCC is not available.	-Permanent faults occur at rate $2\lambda_p$ to State 5.
5	OCC containing 1 FCR is operational. OCC is not available.	-Permanent faults occur at rate λ_p to State 6.
6	No operational FCRs in OCC. OCC is not available.	-None.

Table 12-7. Availability Model of OCC With Transient Recovery –State Description.

12.4.1. Analytical Results from OCC Availability Model

The results from the OCC availability model evaluations are presented in Table 12-8 for nominal configurations of four and five FCRs and for triplex and quadruplex MDCs. Although the OCC reliability and safety analysis eliminated the triplex MDC from consideration, results for a triplex OCC are included below for completeness. Representative availability results are illustrated for a periodic maintenance interval of 200 hours of wall-clock time (approximately 8 days). The OCC availability for other repair intervals can be easily analyzed using the model described in this section.

Nominal Configuration	MDC	Unavailability after 200 hours
5 FCRs	3	7.5×10^{-5} (States 4, 5, and 6)
5 FCRs	4	3.7×10^{-3} (States 3, 4, 5, and 6)
4 FCRs	3	2.3×10^{-3} (States 4, 5, and 6)
4 FCRs	4	7.5×10^{-2} (States 3, 4, 5, and 6)

Table 12-8. Availability of OCC.

12.4.2. Recommended Spares Complement for the SVA OCC

For the quadruplex MDC selected from the reliability and safety analysis, a nominal configuration of either four or five FCRs can be used. The availability analysis indicates that 1 spare FCR yields an unavailability after 200 wall-clock hours of 3.7×10^{-3}, which slightly exceeds the OCC maximum unavailability requirement of 10^{-3}.

Three general strategies can be used to enable the OCC to meet this requirement.

First, the maintenance interval can be decreased from 200 wall-clock hours to a shorter interval while maintaining a nominal configuration of 5 FCRs. The availability model presented above can be used to determine the effect of varying the maintenance interval on OCC unavailability. Table 12-9 illustrates this tradeoff, which indicates that the maintenance interval must be 100 wall-clock hours or less in order for the 4-FCR MDC, 5-FCR nominal configuration OCC to meet its availability requirement. The life-cycle costs of more frequent periodic maintenance must be compared to the savings accrued due to lower FCR hardware procurement costs.

Maintenance Interval	10 hours	100 hours	200 hours
Unavailability	8.1×10^{-6}	9.6×10^{-4}	3.7×10^{-3}

Table 12-9. Maintenance Interval vs. Unavailability for 5-FCR Nominal Configuration

Second, an additional FCR can be added to increase the nominal configuration to six FCRs. Table 12-10 shows the effect of increasing the nominal configuration to 6 FCRs. This increase would allow the maintenance interval to be stretched to 400 wall-clock hours. The acquisition cost of the additional FCR hardware must be compared to the savings accrued by deferring periodic maintenance in a life cycle cost analysis to determine the cost-effectiveness of this strategy.

Maintenance Interval	10 hours	100 hours	200 hours	400 hours
Unavailability	1.6×10^{-8}	1.9×10^{-5}	1.5×10^{-4}	1.1×10^{-3}

Table 12-10. Maintenance Interval vs. Unavailability for 6-FCR Nominal Configuration

Finally, the failure rate of the OCC components can be decreased in a number of ways, such as reducing their complexity (while still meeting the required OCC functionality), exploiting advanced packaging (which may increase the OCC's cost), utilizing higher-quality components (which may also increase the OCC's cost), or housing the components in a more benign environment in order to reduce their failure rates. Table 12-11 shows the effect of halving the FCR failure rate for the 4-FCR MDC, 5-FCR Nominal Configuration OCC. If this can be achieved, then the 4-FCR MDC, 5-FCR nominal configuration OCC can meet the availability requirements with a 200-hour maintenance interval.

Maintenance Interval	10 hours	100 hours	200 hours
Unavailability, FCR Failure Rate = 10^{-4}/hour	8.1×10^{-6}	9.6×10^{-4}	3.7×10^{-3}
Unavailability, FCR Failure Rate = 0.5×10^{-4}/hour	2.0×10^{-6}	2.5×10^{-4}	9.6×10^{-4}

Table 12-11. FCR Failure Rate vs. OCC Unavailability

13. Empirical Test and Evaluation

The requirements of extremely high safety, reliability and availability for the maglev transportation system precludes validation of the control computer dependability exclusively by any single technique, tool, or approach. A balanced validation plan that uses analytical models, formal proofs, empirical test and evaluation, and architectural attributes that enhance the "validatability" of the system [42] can be cost effective and feasible in achieving a validated fault tolerant computer system architecture for maglev. A set of analytical models has been developed and discussed in Section 12, to characterize the dependability attributes of the maglev on-board vehicle control computer. To appreciate the role of empirical evaluation in the validation process, we quote from [43] as follows:

"Design-for-validation concept consists of the following:

1. The system is designed in such a manner that a complete and accurate reliability model can be constructed. All parameters of the model which cannot be deduced from the logical design must be measured. All such parameters must be measurable within a feasible amount of time."

The design of the maglev control computer system, which includes the Fault Tolerant Parallel Processor (FTPP) as one of the major subsystems, has adhered to this precept of the "design for validation" methodology. For example, by complying with all the known theoretical requirements for Byzantine resilience, the reliability of the FTPP can be modeled analytically with just a few parameters. It is not necessary to exhaustively enumerate failure modes and show that each mode is covered with the requisite probability. The analytical models were discussed in Section 12 for the maglev Concept of Operations defined in this study. The models are, however, general enough so that by changing a few parameters one can predict the safety, reliability and availability for other mission scenarios as well. The analytical models use three types of parameters: component failure rates, fault response times (detection and reconfiguration times), and fault coverages (detection and reconfiguration coverages). The component failure rates are estimated using the MIL-HDBK-217F. The other parameters, however, must be deduced from the design or measured experimentally.

Once engineering models of the maglev control computer have been fabricated, it is feasible to measure some of the parameters experimentally by artificially introducing faults and errors in the system and observing the system response. This process is generally known as the fault injection process and will be elaborated on later in this section. The analytical modeling and empirical characterization of the control computer complement each other. Analytical models are abstractions of physical reality. Test and evaluation on the engineering model can help verify model assumptions, determine unknown parameters and increase overall confidence, and hence claims of validation, in the system.

Apart from gathering data for reliability parameter estimation, fault injection plays another important role in the overall system validation. Fault injection can be used to obtain feedback for fault removal from the design implementation. Again, the role of fault injection in finding and fixing design errors should be kept in the proper perspective. One cannot rely solely on fault injection to uncover design, specification and implementation errors. Fault injection is not a substitute for the design-for-validation methodology. However, it *is* a component of the methodology just as specifications, design reviews, analytical models, and formal methods are.

If the fault injection process does not uncover a single flaw in the system under test, it does not imply that there are no flaws in the system, only that the system is correct with respect to the fault set to which it was subjected. But what if some design flaws *are* uncovered? Does that mean the exercise was useless? On the contrary, a utility of the fault injection technique is in uncovering shortcomings in the system. One gains a deeper understanding of the fault tolerance design, a more fundamental appreciation of the cascade of events triggered by a fault, including complex interactions between hardware and software elements and the timing relationships between various events.

With the above discussion in mind, the goals of the empirical test and evaluation, in general, and fault injection, in particular, of the maglev control computer system can be summarized as follows :

1. To test the system design specification for fault tolerance.
2. To obtain feedback for fault removal from the design implementation.
3. To obtain statistical data regarding fault detection, isolation, and reconfiguration coverages and latencies.
4. To obtain data regarding the effects of faults and the fault-tolerance mechanisms on system performance.

The remainder of this section describes the parameters that must be varied to create a comprehensive set of fault injection tests for the architecture selected for the maglev control computer system, i.e., the Fault Tolerant Parallel Processor. Pin-level hardware faults using a hardware fault injector, software-injected memory mutations, disruptions of power supply, and physical disconnection of boards, optical links and connectors are some of the ways to inject faults and errors to test the system.

13.1. Fault Injection Test Cases

Any given fault that is injected occurs within a particular context which consists of the hardware environment, the software environment, and the fault injection environment. Thus, there are four major parameters that can be varied when creating test cases:

- Hardware environment
- Software environment
- Fault injection environment
- The actual fault

Hardware Environment

The hardware environment consists of the hardware elements that make up the system during the particular test. For the FTPP, these include the Processing Elements (PEs), the Network Elements (NEs), and the I/O Controllers (IOCs). These hardware building blocks may be arranged in varying configurations. The parameters that describe the possible configurations are:

- Redundancy level of the victim Virtual Group, i.e. PEs,
- Configuration of other Virtual Groups (# of simplexes, triads and quads)
- Redundancy level of NEs
- IOC configuration

Software Environment

Similarly, the software environment consists of the software building blocks that compose the system during the particular test. These software building blocks may be arranged in varying configurations. The parameters that describe the possible configurations are:

- Combinations of system functions and applications, and support functions
- Iteration rates of system functions
- Number of application tasks
- Computational loads of application tasks (throughput)
- I/O requirements of application tasks
- Memory requirements of both application and system tasks
- Language specific parameters (e.g. Ada pragmas)

Fault Injection Environment

The fault injection environment consists of attributes of the faults to be injected. These attributes may be varied and arranged in different configurations. The attributes are:

- Number of Fault Containment Regions (FCRs) affected

- Number of simultaneous faults
- Duration of fault
- Random placement vs. selected placement
- Hardware injection vs. software injection
- Scope of fault (i.e., FCR, Board, Chip, Pin)

A graphical representation of all combinations of these parameters for the Fault Injection environment is shown in Figure 13-1. For clarity, the variations of the parameters under the top level are shown only once.

13.2. The Injected Fault

Finally, each individual hardware fault to be injected is determined by examining the possible places at which to inject a fault and the region of application. The affected region can be varied hierarchically from an FCR to a single board or link to a chip to a pin. As an example, a whole FCR can be faulted by turning off its power supply or corrupting its clocking source. A board or a link can be faulted by physically disconnecting it from the backplane or the connector, respectively. A chip can be faulted by disconnecting its power or ground connection. A pin can be faulted by intercepting the correct signal on the pin and replacing it with the desired faulty signal. A special hardware fault injector has been designed and built at Draper to perform this function and has been successfully used to test several fault tolerant computers [44]. It is possible to inject stuck-at faults as well as externally generated arbitrary signals as the output of a selected output pin of a device or as the input of an input pin of a selected device. Multiple simultaneous faults, permanent faults, and faults of a limited specified duration can be applied to emulate transient faults. Some of the possibilities with a software-injected memory fault include corruption of data or instructions in memory and alteration of an outgoing message from a processor or an incoming message to a processor.

To reiterate, a test case consists of an actual fault within a fault injection configuration within a software configuration within a hardware configuration. The total number of possible test cases is the product of the number of hardware configurations, software configurations, fault injection configurations, and faults, i.e.,

$$N_{Tot} = N_{HW} * N_{SW} * N_{FI} * N_{F}$$

If all possible variations of the hardware environment, software environment and fault injection environment were exercised with only one fault each, the number of test cases would exceed any practical test and evaluation budget and schedule constraints. In order to limit the test cases to a number consistent with the time and financial constraints of a prac-

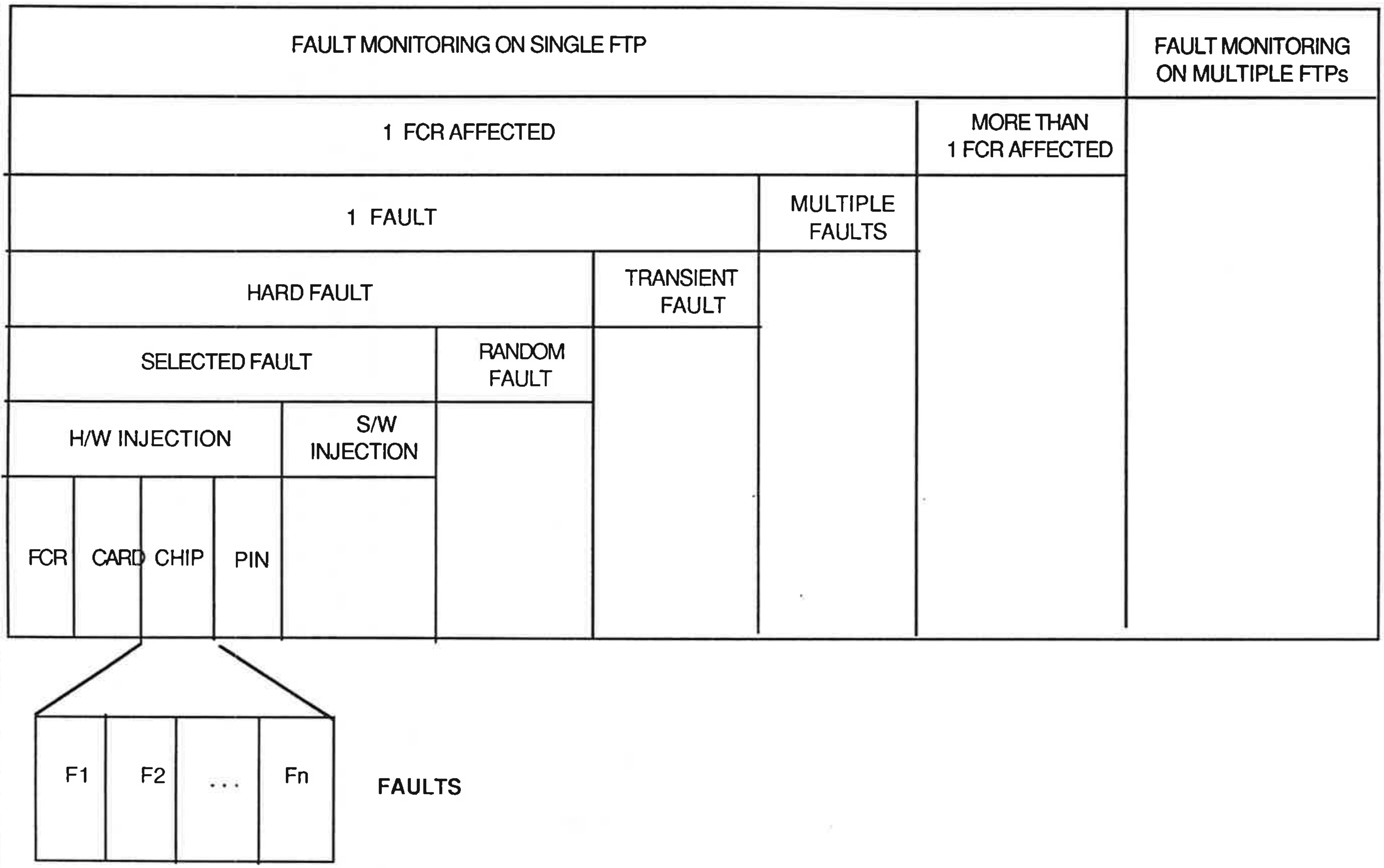


Figure 13-1. Fault Injection Configuration

tical program, the test cases must be selected to be a subset of the exhaustive combination described here. A set of guidelines are needed to select the test cases such that the objectives of the test and evaluation program are still achieved. With regard to the maglev control computer architecture, the following guidelines can be used to reduce the set of faults for empirical evaluation.

1. NDI vs Custom Hardware: Custom designed hardware should be more heavily tested than Non-Development Items (NDI), i.e., off-the-shelf hardware. The obvious reasoning behind this guideline is that the NDI hardware has already been thoroughly tested by its manufacturer, it's design is more mature and because of the widespread use of such hardware, other users have discovered the anomalies which have been corrected by the vendor. By contrast, the custom designed hardware has totally opposite characteristics. For the FTPP, examples of NDI are Processing Elemènts and I/O Controllers. Custom hardware includes Network Elements and their interfaces to processor bus and the Fiber Optic Data Links interconnecting NEs.

2. NDI vs Custom Software: The same logic applies to software as well. For the FTPP, the NDI software includes the Ada Run Time System. The custom software includes the FDIR (Fault Detection, Isolation and Reconfiguration) routines, task scheduler, I/O System Services, and real-time extensions to the Ada Run Time System. Depending on the maturity of the application, the application software may or may not be considered off-the-shelf and then tested accordingly.

3. Redundancy Management Elements vs Other Elements: Within the custom hardware and software, the redundancy management related elements should be stressed, tested and evaluated more thoroughly than the rest. The RM hardware and software tends to be more complex and therefore more likely to contain design errors. Most of RM software is used only under abnormal conditions, i.e., when an error is observed. As such, it is invoked less frequently than, say, the task scheduler or the I/O services. Similarly, the scoreboard, one of the major hardware element within the NE, is normally used to process message requests under no-fault conditions. It rarely has to deal with processor time-outs, full buffers or other error and flow control situations. It is important therefore to exercise the RM-related hardware and software elements much more thoroughly.

To test the Redundancy Management hardware and software in a systematic manner, a strategy needs to be developed for guiding the process of fault injection. One such strategy has been developed and applied recently to a system in Europe [19]. This, along with the fault injection experiments conducted on the Advanced Information Processing System

[46], are good starting points for developing the process for the empirical test and evaluation of the maglev control computer system in the next phase of this study.

14. Application to Transrapid System

In this section, the verification methodology developed during the course of this study is applied to the control computer system of the German Transrapid system designated TR07. While a complete and thorough application of this methodology to TR07 is beyond the scope of the present work, it is possible to illustrate how such an effort would be conducted given sufficient time and, most importantly, access to the design and implementation details of the Transrapid control computer system. Despite our best efforts, in the time available for this study, we were unable to obtain answers to key questions regarding the operation of the automatic control system used in TR07. Hence, we have based our analysis on material which has been published in various references available to the public at large. Based on this material, we describe the mission requirements and the functional requirements of the TR07 control computer architecture. Next we discuss the overall computer architecture used in TR07 and, to some extent for selected subsystems, and the redundancy management strategies employed to meet the stated mission requirements. Using the approach presented in Section 10, a critical failure modes and effects analysis is performed on the TR07 control system. Due to limited information, this qualitative analysis raises more questions than can be answered. Nevertheless, it provides a useful starting point for design verification. These questions may in part be answered by developing analytical models and solving those models to obtain quantitative results. Parts of the TR07 onboard control system have been modeled analytically. Finally, some open issues are raised and some general conclusions are drawn.

14.1. Mission Requirements

In order to effectively design a control computer system for any application, several quantifiable system level requirements must be specified. These include the dependability requirements, i.e., the safety and reliability requirements of the control computer system stated respectively as a maximum probability of system loss and the maximum probability of not completing a trip. Other dependability requirements include the availability and maintainability of the system which relate to the ability of the system to deliver its service in a timely and predictable manner. If these requirements are not specified quantitatively at the outset, the system designers will not have clear goals to steer toward nor will system evaluators have clear yardsticks by which to measure the adequacy of the final design. Our research did not uncover clear, quantitative references to these requirements for the control computer system of TR07.

Although quantitative requirements are lacking, there are various qualitative statements in the reference materials about the required level of dependability. For example, the

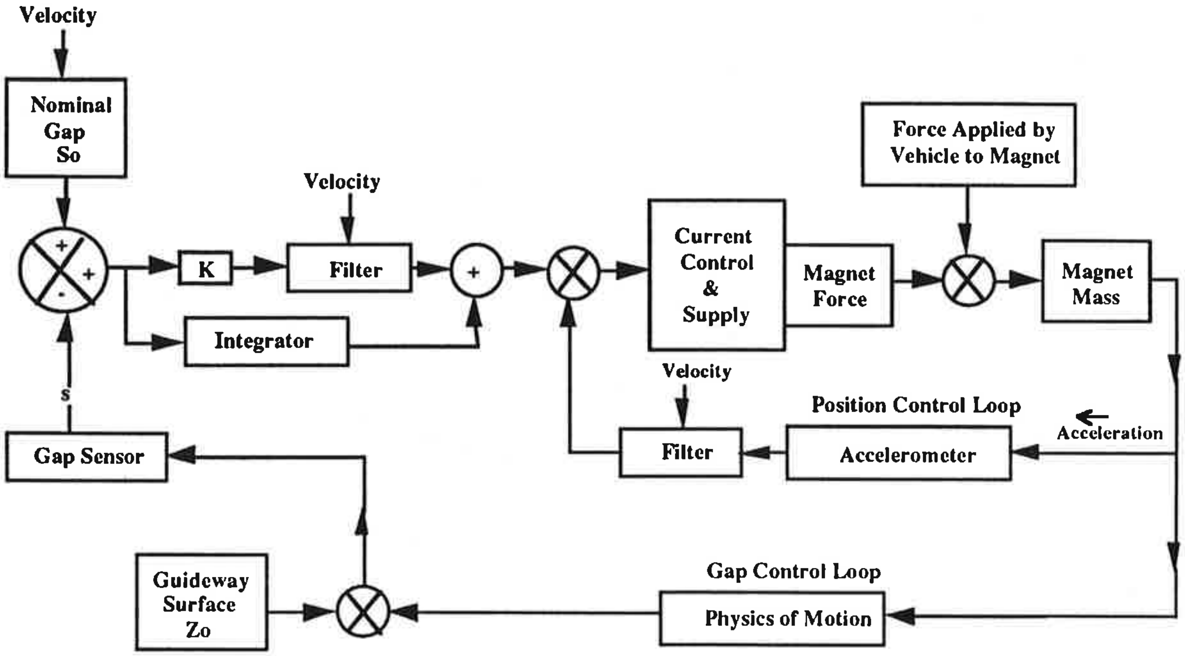


Figure 14-1. TR07 Levitation and Guidance Magnet Control Loop

Onboard Power Regeneration and Storage

Many onboard systems require power continuously for proper operation. While the vehicle is in motion and has a speed greater than 100 km/hr, power is supplied by linear generators which provide non-contact power collection to the vehicle by induction with the magnetic flux from the guideway-mounted long stator motor sections. This power is also used to recharge four electrically isolated 440-volt batteries. When the speed of the vehicle falls below 100 km/hr, power to onboard systems is provided by the onboard batteries. These must provide enough power to bring the vehicle to the next safe stopping point while maintaining the operation of all onboard systems, such as the OCC and the HVAC equipment, without interruption even in the event of various multiple faults.

Control of Onboard Systems

Many onboard systems require automated control. These include monitoring the safety related processes of train operation, operating the door controls, operating the location system, operating and monitoring the diagnostic systems, displaying current operating information on the onboard operating console, operating the auxiliary brake, monitoring the passenger emergency signal, and handling communications to the WCC and CCF. It must be able to detect when the vehicle has entered an impermissible state which occurs, for example, when data transmission is lost.

Onboard (Secondary) Brake

This braking system must be located on the vehicle and operate independently of wayside communication and operator input. It must also operate independently of the drive, i.e. propulsion, system. The power for the secondary brakes is supplied by onboard batteries. When a current is supplied to several onboard braking magnets, an eddy current is induced in the solid steel guidance rail which exerts a magnetic drag on the vehicle provided that the vehicle is traveling with sufficient speed. As the secondary brake slows the vehicle below some threshold velocity, the brake becomes ineffective and the final braking mechanism is a mechanical skid brake. This requires that the safe hover mechanism be overridden, allowing the vehicle to land on its skids on the guideway. Several independent braking circuits are required. Two key requirements for the proper operation of the secondary braking system are (1) the ability of the onboard control system to determine when an impermissible operating state has occurred and to then initiate safe emergency braking and (2) the ability to reliably transmit to the CCF the fact that an emergency braking action has been taken. The former is needed to ensure that a vehicle which cannot be stopped by the primary brake can stop itself. The latter is needed to ensure that collisions do not occur as the result of an emergency stop, i.e. that vehicles following the disabled vehicle are safely brought to a stop in a timely manner.

Safe Stopping

The safe hover concept discussed above requires that vehicles only be allowed to stop at designated safe stopping areas. To ensure that a vehicle can always reach a safe stopping area, Transrapid imposes the following five requirements on its system: (1) The vehicle must develop sufficient velocity before leaving a zone so that it can reliably coast to the next allowed stop location. (2) The vehicle must be able to reach the next stop location independently of the wayside power system, i.e. by use of onboard power only for levitation and other onboard systems. (3) The vehicle safe hover and safe stopping systems must have the required reliability, with electrical and physical autonomy, to limit the risk of multiple failures to an acceptably low level. (This level is not specified quantitatively by the developer.) (4) The vehicle must be able to bring itself to a safe stop at a safe stopping location without any input or guidance from the central control system. (5) The vehicle control system must have the reliability to assure safe operations independent of the central control system. (This reliability is not specified quantitatively by the developer.)

The onboard computer is continuously provided with adequate information (such as vehicle location and safe stopping locations) via its data link to central control so as to permit stopping of the vehicle at any time during the trip at the next available safe stopping point independent of further outside information from either the central or wayside control.

Propulsion/Brake (Primary)

Propulsion is not considered to be a safety critical function on TR07 since the vehicle must continue to levitate and maintain sufficient momentum to coast to the next safe stopping point in the event of a propulsion system failure. The propulsive force is provided by the guideway-mounted linear synchronous motor which is controlled from power substations spaced at intervals along the guideway under the direction of the CCF. These substations effectively divide the guideway into zones. Each zone is powered by two substations, each of which can provide power independently to propel a vehicle, albeit at reduced speed, through its zone. An operator at CCF selects a particular speed profile and operating scenario for a given vehicle from a data bank of these files maintained by the CCF. This profile is then transferred over an optical communication link to the various WCCs for the coordination of propulsion and braking of the vehicle. Presumably, there is a significant amount of coordination and communication between adjacent WCCs and the CCF to effectively and safely propel the vehicle smoothly along the guideway. For example, before a vehicle is allowed to enter a zone, authority must be granted by the CCF to the previous WCC to advance the vehicle into the next zone. If the authority is not granted, the vehicle is slowed and possibly stopped in its present zone, and the power in the next zone is not turned on.

The primary brake is initiated by the central control system which controls the long stator propulsion motor to reverse vehicle thrust. Once the command is given by the CCF to apply the brake, the actual braking action is carried out by the WCCs which have direct control over the power converters which propel the vehicle.

The propulsion system must have a highly reliable shutdown capability to allow correct operation of the vehicle braking system during a propulsion failure.

Route Control

Route control refers to the process whereby a vehicle is advanced to the next zone on its pre-planned travel profile or route. Before a vehicle is allowed to enter a zone, it must receive clearance or authority from the CCF to proceed. Clearance is requested by the WCC of the preceding zone. The CCF only allows vehicles to proceed after it has verified that certain conditions have been met. These include a verification that no other vehicle occupies the zone, no other vehicles or technical installations can enter the guideway zone in question, information about the local permissible maximum speed and location of safe stopping points has been distributed by the CCF and received by the OCC and WCC, and that all switches, i.e. moveable guideway elements, in the zone are locked into their proper position. Furthermore, the CCF only requests permission to advance the vehicle when certain other conditions are met. These include a verification that the vehicle has sufficient velocity to coast to the next safe stopping area, that the batteries onboard the vehicle have sufficient charge to stop the vehicle at the next safe stopping area without outside power supply, and that the communication system is operating properly. Presumably, this list of checks in not complete since other failures could require the vehicle to stop at the next safe stopping point.

Route Integrity and Switch Control

Transrapid uses mechanical switches to change routes from one section of guideway to another. A special section of guideway is moved from one point to another to effect the change. The switch position sensors must be able to accurately determine the switch position within a required +/- 1.5 mm tolerance. Once the switch is in the correct position, it is held in place by hydraulic locks. The switching operation is very demanding, requiring precision movement of mechanical parts under electronic controls. There are many independent components which are subject to failure. Evidently, the inspection and maintenance plan for switches is not yet part of the published Transrapid documentation.

The position of every vehicle in the system is tracked continuously by the CCF. Since Transrapid uses online stations, has fairly slow acceleration and deceleration capability and uses long multi-vehicle consists of 2 to 10 cars to increase its capacity, it requires fairly long headways of approximately 5 to 7 minutes between consecutive trains. Hence, there is adequate time to allow the CCF to perform its function of route integrity and switch control. In addition, reference is made to providing a means of ensuring that no foreign obstacles or debris are on the guideway. Finally, given the very tight tolerances between the vehicle and the guideway, some means of ensuring the guideway alignment is necessary. The guideway alignment is checked periodically by a maintenance vehicle which compares actual geometry with the desired specifications.

Vehicle Control

The Transrapid system is highly automated. Central control (CCF), wayside facilities (WCC) and the onboard control systems (OCC) interact to maintain automated control of train operations during normal conditions and most emergencies.

The CCF initiates and controls maglev train operations according to demand or selected schedules. Despite the automated nature of most system operations, constant monitoring by central control personnel is required to ensure that abnormal conditions are identified and corrected before they escalate to emergencies. Loss of communication is a major safety concern, however the present documentation does not describe how this condition is detected or how a failure would be dealt with.

Vehicle speed range is location dependent. It is generated by the CCF with consideration given to guideway operational readiness, vehicle internal data, status of levitation, guidance, and braking installations.

Station Supervision and Control

Passenger stations provide the normal means for passengers to board and exit the maglev train. Passengers have about 2 minutes to disembark and about 8 minutes to board the train. Since the Transrapid system uses the CCF to direct all train movements very little in the way of special automated control functionality is delegated to stations.

Safe Speed Control

The CCF determines the speed profile for every vehicle in the system. The speed profile is the allowed range of speed for each zone through which the vehicle must pass as it travels from its origin to its destination. The maximum speed may not be exceeded and the minimum speed must be achieved. This information is transmitted over a secure data link to the various WCCs along the route which in turn control the power electronic equipment which directly controls the velocity and acceleration of the vehicle. The OCC is also given this safe speed data for use in emergency situations during which time it may be required to control its own speed by means of a secondary brake.

Environmental Monitoring

Weather conditions such as wind, snow and rain are monitored either by the vehicle or the WCC. EMF test and measurement equipment monitor level of electromagnetic fields in areas populated by passengers and personnel. Information on how Transrapid plans to monitor internal and external noise was not available. Each vehicle is equipped with an automatic fire alarm. Whether this alarm is triggered by smoke, heat, or both is not discussed.

Route Planing and Scheduling

Route planning and scheduling is performed by the CCF is conjunction with input from human operators. The capability to simulate planned schedules is provided by software hosted by the CCF computers.

14.3. Control Computer Architecture and Redundancy Management Approach

To meet the requirements for "high levels of reliability" the designers of the control computer system for Transrapid have developed several computer architectures which are intended to provide fault tolerant operation. They have made a provision for hardware redundancy in their design to allow the system to continue to operate in a normal, or possibly degraded, manner even if a component fails. Whether or not uninterrupted, full capacity operations are allowed after a fault depends on the type of fault and the coverage provisions made for that fault in the initial design. However, in all cases the intent is to protect passengers and employees from harm in the event of a fault even though the level of service may be reduced, resulting, for example, in late arrivals and/or delayed departures. Thus the intent of the design of all the control computers is to make the overall system fail-safe. References [1] and [45] provide some details about the architectures of the OCC, the WCC, and the CCF. The various components of the OCC are described in relatively great detail. A good deal less information is provided on the architectures of the WCC and CCF. However, the amount of information on any part of the system is sketchy at best. The discussion below is based on data obtained from these references.

The OCC comprises two main subsystems: the computer which performs functions related to Automatic Train Control (ATC) operations and the controllers which perform levitation and guidance. A fair amount of detail is available for the design and operation of the latter; more limited detail is provided for the former.

Operational ATC technology is defined as "the functions and installations whose purpose is the safety, control, and supervision of vehicle operations, as well as intercommunication between them." Of the functions described in Section 14.2, the onboard ATC computer is expected to perform the following functions:

1. Monitor and control the safety-related processes of train operation in conjunction with the CCF (which entails comparing control and actual values pertaining to the safety process, accepting data from and sending data to either the "data transfer computer" or the vehicle operating console.) Note: The data transfer computer is not further identified nor are its functions specified.
2. Control the door.
3. Obtain location information from and monitor the correct operation of the location detection system.
4. Initiate and control emergency braking of the vehicle after the detection of an impermissible operating state which is defined in various places as data transmission loss, failure of one part of the quad-redundant vehicle location installation, or the failure of the onboard ATC computer. (This is not an exhaustive list of impermissible states, only those expressly identified in the references.) A key aspect of the design of the emergency braking system is that after a failure has

been detected, and braking initiated, the components which are still operational must not fail before the vehicle is brought to a stop.

5. Monitor the charge level and correct operation of the four onboard batteries.
6. Display information on the operating console.
7. Monitor the passenger emergency signal.
8. Conduct communications with the CCF.

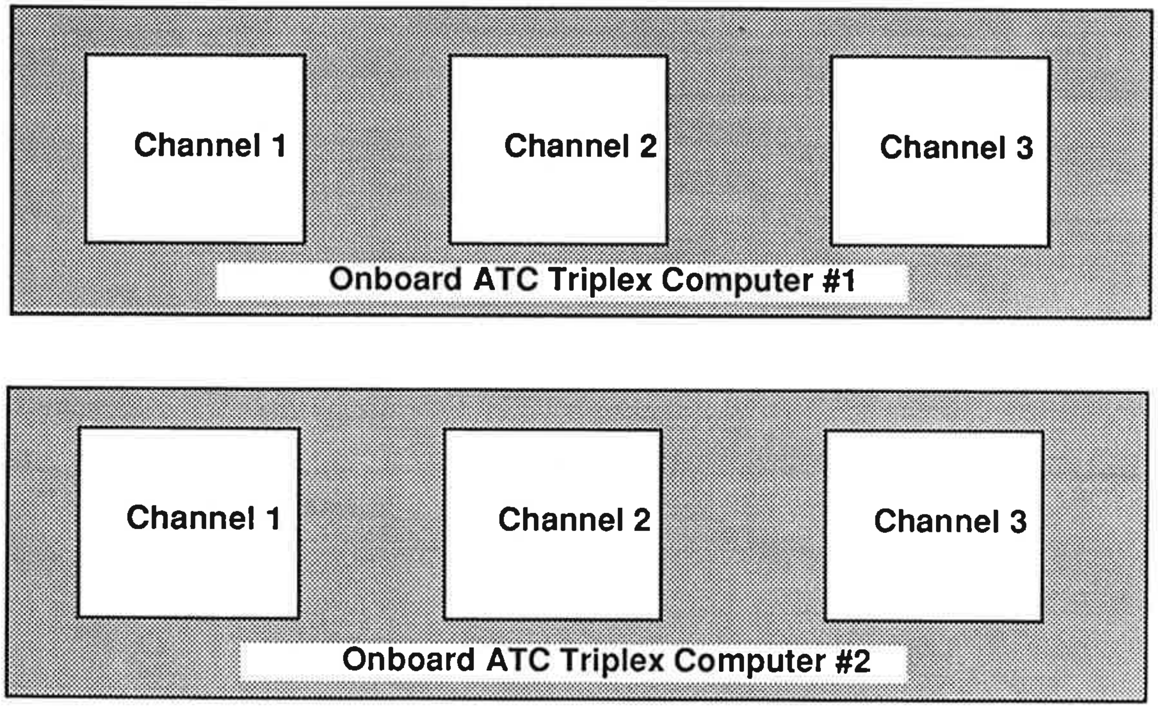

Channel 1
Channel 2
Channel 3
Onboard ATC Triplex Computer #1
Channel 1
Channel 2
Channel 3
Onboard ATC Triplex Computer #2

Figure 14-2. The dual TMR architecture of the OCC for Automatic Train Control

The onboard ATC computer employs two separate virtual computers each comprising three channels. A channel consists of a microprocessor and its associated hardware components. The three channels act as a TMR (Triple Modular Redundant) architecture to carry out the functions of a single fault-tolerant computer. A block diagram of this architecture is shown in Figure 14-2. The interchannel and intercomputer connections are not shown because the references provide no information as to their interconnectivity or the means by which they share or exchange information. This aspect of the redundancy management scheme is very important and can potentially contain a number of single point failures if not designed in accordance with the fundamental theory of fault tolerance.

Loss of one channel (i.e. one microprocessor and/or its associated components) in either TMR system is tolerated and the system continues to operate with no degradation in capability. A second channel failure in either computer requires the vehicle to stop at the next safe stopping point. If either or both TMR systems lose one channel, normal operations are allowed to continue. However, if one of the TMR systems sustains two failures, even if the other has three fully functioning channels, a fail-safe mode of operations commences during which time the vehicle must be brought to a stop before any other failures occur. Thus, at least two channels in one TMR system must be operational to safely bring the vehicle to a stop. The literature does not describe how failures are detected, or how long it takes to detect a failure, or how the two TMR computers communicate reliably so that each is fully aware of the other's status. Furthermore, there is no information on how the system reconfigures itself after a failure is detected so that a failed channel cannot create an unsafe situation by, for example, stopping the vehicle at an unauthorized time or place. Presumably either of the two TMR computers is capable of controlling the emergency braking operation. However, the literature does not indicate whether the remainder of the above functions are distributed between the two TMR computers or whether both computers perform all of the functions all the time.

The other information which is given about the ATC computers is that the software which executes on these computers must be validated through "comprehensive checks and tests." However, the ATC computer hardware is exempt from these validation efforts since the hardware has already been approved by the German Federal Railways and hence the German certification agency (TÜV Rhineland) does not deem it necessary to re-certify it for the Transrapid application. The hardware is a Siemens Corporation system based on its SIMIS control computers and known by the German acronym BLM. The literature which Siemens provides about the SIMIS system only describes a dual channel architecture. In this system, two Siemens MES 80 computers are connected by a comparator circuit for detecting errors. An error is defined as a disagreement between compared values from the MES 80 processors. A disagreement triggers a pulse from another circuit which activates lock-out circuitry and prevents any outputs from being sent to the outside world. Their literature claims that the SIMIS "reacts to every conceivable hardware fault by shutting down." Clearly, this is not the operation that is intended for the Transrapid onboard ATC computers. But the relationship between the manufacturer's description of its SIMIS system and the use of SIMIS in Transrapid is never explained.

The second main subsystem of the OCC is the levitation and lateral guidance system. In the Transrapid TR-07 design, the vehicle is supported and guided by a logitudinally distributed set of magnets, each three meters long, supported by brackets which link the magnets together in a manner which produces a kinematic hinge between the magnets. Electronic control circuits are used to control the forces acting on the magnets so as to maintain on average a constant distance between the hinge points and the levitation surfaces.

The position of each hinge point is controlled by two independent control circuits. Either circuit is capable of controlling the hinge position and can independently provide adequate force to maintain levitation and guidance of the vehicle within the fixed guideway. Hence, the design provides dual redundancy at each hinge point where two magnets are linked. However, the leading and trailing magnets do not appear to benefit from the redundancy of this design. A separate magnet set, consisting of 15 magnets each, is used to support each side, left and right, of the vehicle. Thus, there are 14 hinges on each side for a total of 28 per vehicle, with each hinge controlled by a dual redundant gap control system. Section 14.5 describes an analytical model of the reliability of the gap control system. Figure 14-3 shows a simplified cross sectional view of the TR07 vehicle highlighting the key components of the electro-magnetic suspension and guidance system.

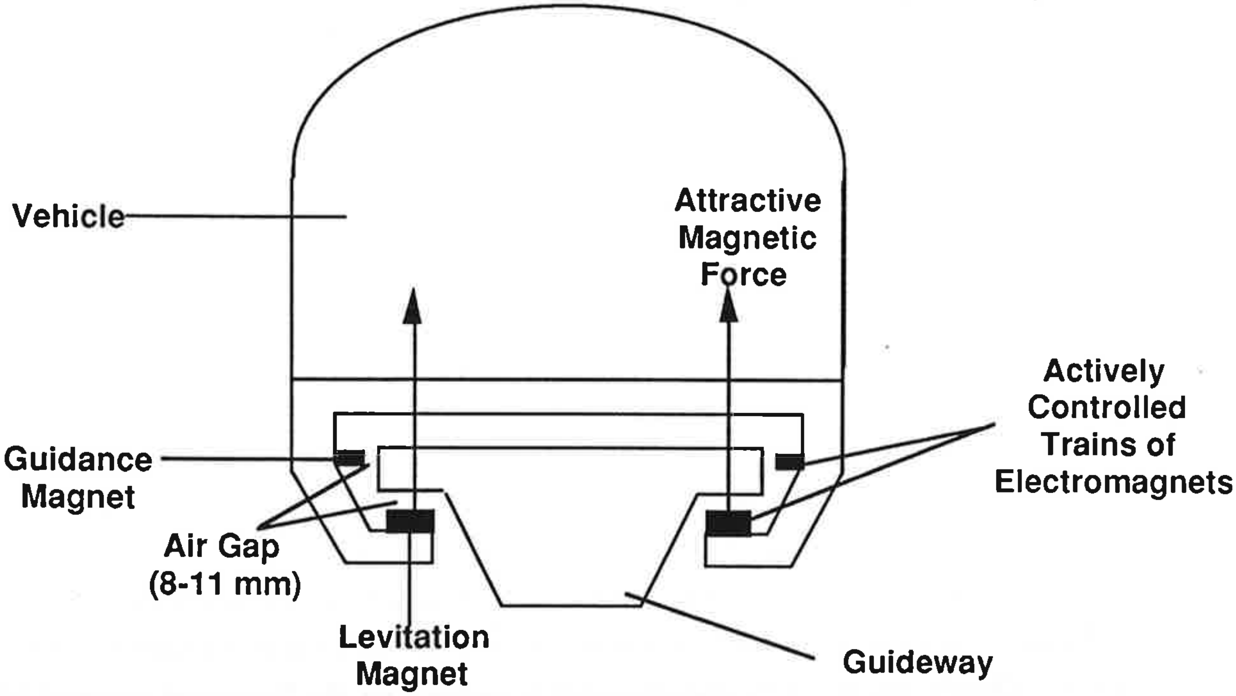

Vehicle
Attractive Magnetic Force
Actively Controlled Trains of Electromagnets
Guidance Magnet
Air Gap (8-11 mm)
Levitation Magnet
Guideway

Figure 14-3. TR07 Electromagnetic Suspension and Lateral Guidance System

Each magnet is divided into two separate units. These units are controlled by separate controllers. Rather than controlling a single magnet, a controller controls a hinge point which is the connection of two magnets. In effect, a controller controls the position of the endpoints of two magnets. Each such hinge has two independent controllers. Figure 14-4 shows this arrangement. The gap at the hinge is controlled by controllers 2 and 3. The gap is measured by four gap sensors. Each controller receives a reading from a sensor on either side of the gap. For example, controller 2 takes gap measurements from gap sensors A-3 and B-4. The acceleration signal is measured by two accelerometers. The references do not make clear whether or not the accelerometers are read by only one controller or by both controllers.

The physical separation of the two gap sensors permits the gap control to be maintained over the thermal expansion joints in the support and guidance rails. A large gap signal occurs at a sensor when it passes over an expansion joint. Normally, the gap signal that is used in the control algorithm is the average of the two gap sensor readings. However, these gap readings are subjected to a range check before being used in calculating an average value. If the difference in the two measurements is greater than 1.5 mm, the larger value is discarded and only the smaller value is used as input to the control algorithm. Since only one sensor at a time encounters an expansion joint, this algorithm provides an accurate measure of the gap provided that neither sensor has failed. Presumably, the ill-effects of a failure are counter-balanced by the actions of the second controller on the gap. The gap reading obtained here is compared to the desired gap to provide the error measurement used in the control loop shown in Figure 14-1. Controller 2 and its associated chopper provide the current to magnetic unit 2A to generate magnetic forces to reduce the gap error.

The location detection system, or location installation, is not part of the onboard ATC computer but it nevertheless performs a safety critical function by providing that computer with the exact location of the vehicle. Without the correct operation of this system, the safety-critical programmed brake function can fail. The location installation is known by the German acronym INKREFA, which can be translated as the Incremental Vehicle Location System. There are three parts to the system: position markers attached to the guideway at intervals of approximately 200 meters, an active vehicle mounted sensor system which scans the guideway to read the markers, and a mechanism to count stator pack grooves between markers to more accurately determine the position of the vehicle. The marker gives the raw vehicle position. This marker is used by the ATC computer as an index to an internally stored table which provides the absolute position of the tag. Finally, more accuracy in the position measurement is obtained by counting stator grooves between markers. How these grooves are detected and counted is not explained in the reference. Also not mentioned is the degree of accuracy required to support the safety-critical function of the secondary brake. The system is redundant in that (1) tags are placed on both sides of the guideway and (2) four sensors are mounted on the vehicle, two per side. Contradictory requirements are provided in the references as to how many of these four sensor readings must agree before some action is taken. In one case at least two must agree. In the other, at least three must agree. In the former case, the only action taken is that the most recent

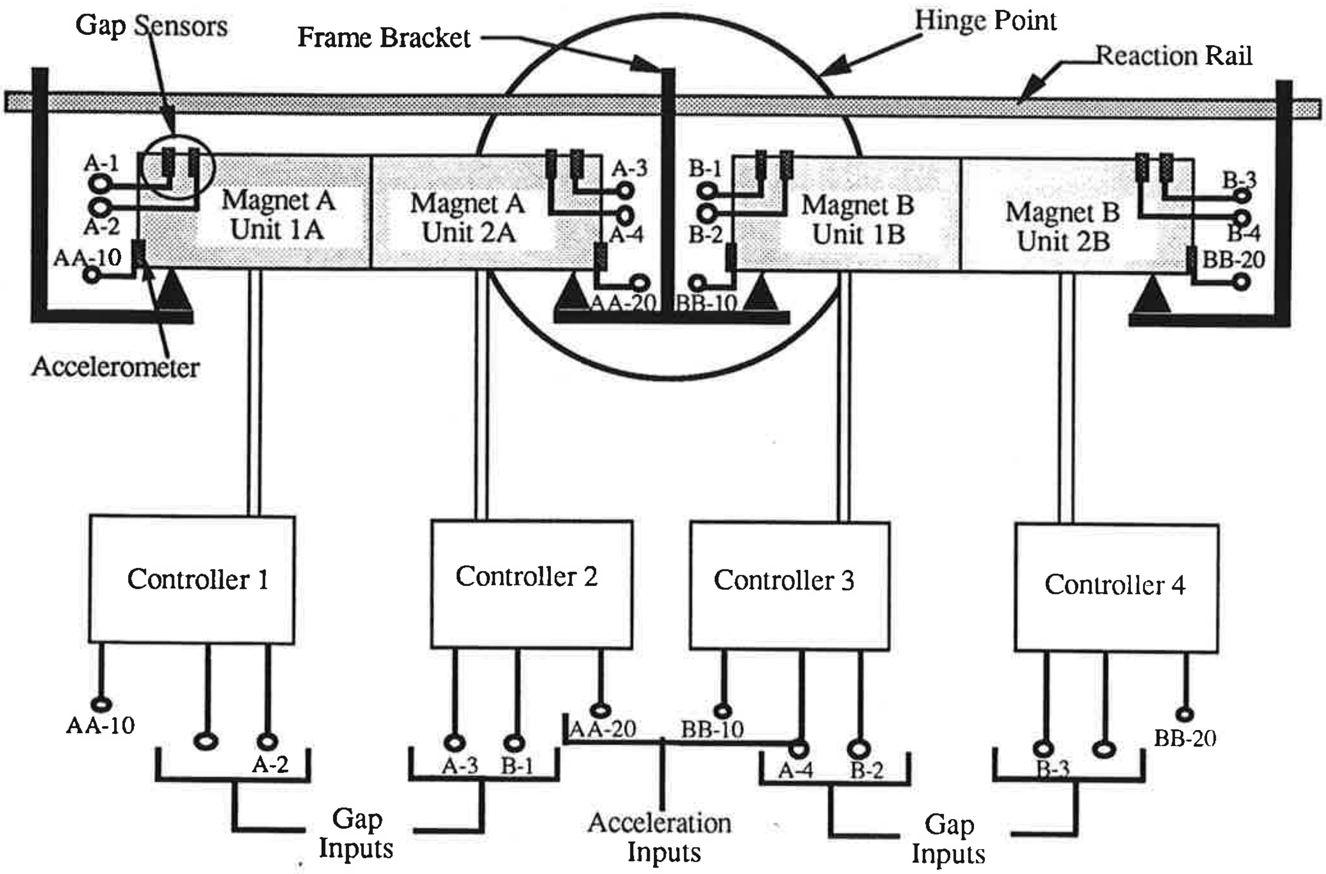


Figure 14-4. Levitation and Lateral Guidance Controller Circuits

successful location reading is used to extrapolate the correct position until a successful reading is obtained. In the latter, if less than three channels agree, an emergency stop must be initiated. Furthermore, in this case, no other failures in the system must occur until the vehicle is brought to a safe stop. Unfortunately, the manner in which these four tag reading sensors are connected to the ATC computers is not discussed. This connectivity is crucial to the reliability analysis of the system.

The vehicle location obtained and verified by the onboard ATC computer is transmitted to the CCF via a data transmission link which utilizes a 40 GHz radio link between the vehicle and receivers along the guideway and a fiber-optic cable link between the receivers and central control. The data is transmitted in "secured telegrams" with "inverse telegrams" also sent for verification of telegram accuracy. Furthermore, the system is designed so that two receivers are in range at any one time and the vehicle has two autonomous transmitters. Presumably the two TMR ATC computers conduct these communications since there is a requirement that a fail-safe computer perform this function. No explanation is provided as to how these computers reach agreement as to the contents of a telegram, either incoming or outgoing.

The actual movement of the vehicle, although directed by the CCF, is under the direct control of the WCCs which control the power converters which form part of the linear synchronous motor which directly moves the vehicle along the guideway. The speed control required to maintain safe operating distances between vehicles is executed by means of the long-stator, linear synchronous propulsion system which is arranged in sections. The Transrapid design calls for functional redundancy in the distribution and power conditioning system, beginning at the input utility service feeders and terminating at the long-stator motor segments. By separate and alternate power feedings of the left and right sections of the propulsion system windings additional propulsion reliability is achieved since either side can propel a vehicle, albeit at a reduced speed. This design is intended to increase the availability of the power supply, rather than its reliability. Apparently, the Transrapid designers believe that the propulsion system cannot be designed to be fail-safe.

The speed profile is transferred from the CCF to the WCCs using a fiber optic network. Whether or not the WCCs exchange information directly or only through the CCF is not clear. However, the WCCs are responsible for the coordination of vehicle propulsion and braking. Apparently a zone-based system is used. Each WCC requests authority from the CCF to move a vehicle into the next zone. If the permission is denied, it is assumed that the WCC can stop the vehicle in the present zone or that the vehicle can stop itself. Some consideration must have been given to the scenario in which a WCC decides to stop a vehicle when it is not supposed to. Since the vehicle can stop itself, this failure mode could also be caused by erroneous actions on the part of the vehicle. However, this scenario and how it would be handled are not discussed. Furthermore, no information whatsoever is provided by our references about the computer architecture of the WCC. It is not considered part of the "safety-critical" part of the system.

All vehicle movements are directed by the CCF. The CCF supervises vehicle operations, displays traffic information, coordinates vehicle propulsion control between central and wayside elements, and monitors other key wayside- and vehicle-based operational elements. Secure communication is an integral part of a centrally controlled system as proposed by Transrapid, yet the means to ensure this communication is not described in detail. For example, there is a provision for two independent communication installations for voice contact between the control center and the vehicle. However, the means to meet this requirement is not discussed. One passing mention of the architectures of the computers used in the CCF indicates that they must be dually redundant systems. Another description points out that the CCF computers must be high-capacity process computers. Transrapid operational sequencing, i.e. timetables and the maintenance of safe separation between vehicles, are nominally automated. However, the operating staff has a wide latitude to intervene and make modifications to the timetable so as to correct or bypass faults. However, the nature of the man/machine interface is not discussed in the reference materials.

The route integrity portion of the control system is responsible for determining if the route requested by the system operator at the CCF is safe for the requested operation. In particular, the bending beam switch must be able to move between two precisely defined positions on demand and remain in these positions under severe vehicle and environmental loading conditions. Before the switch is deemed in place, all end position and locking sensors must register the correct position. The switch is kept in place by a mechanical lock. The switch position sensors are required to be able to measure switch position to within a +/- 1.5 mm tolerance. There are three such sensors, left, right, and center. Agreement of all three is necessary to conclude that the switch is in the correct position. The movement of the switch is controlled by eight hydraulic cylinders. Each of these is monitored during switch movements. Hydraulic locks are activated and the final position of each lock, which is monitored by a single sensor, must be within 2.5 percent of the required location for that cylinder. Presumably, if any of these criteria are not met, the route integrity software would detect this failure and not allow any vehicles to pass through the section of the guideway containing the switch.

14.4. Critical Failure Modes and Effects Analysis (Qualitative)

Section 8 of the present report pointed out that redundancy itself does not in general guarantee high reliability. Correct redundancy management is crucial in transforming a redundant system into a fault tolerant one. Section 8 presented a checklist which can serve as the starting point for a critical analysis of redundancy management in computer architecture which purports to be fault tolerant. The following discussion demonstrates how this list can be used as a basis for qualitatively evaluating the redundancy management of a redundant computer architecture. The sample analysis is performed on the OCC since the references provide the greatest amount of detail on this part of the Transrapid control computer system. Nevertheless, the level of detail is not adequate to perform a thorough analysis; many open issues and questions remain unanswered.

Single Point Failures:

Without knowing the details of the onboard ATC dual TMR computer architecture, it is impossible to pinpoint its single point failure modes. However, the obvious single point failure in a dual SIMIS microcomputer, upon which the ATC architecture is based, is the comparator circuit. If this circuit fails in such a way that a shutdown command is issued at an inopportune moment, the vehicle could be brought to a stop at an unsafe point, or the transmission of an important message could be cut-off. The SIMIS system also attempts to protect against faults which produce errors in data which is transmitted to peripheral devices by the use of a check word which is "added" to data at the source and "subtracted" from data at the destination. This protocol can detect some bit flips which occur during transmission, but not all such flips since some combinations of errors will still produce the same check sum. Hence, this is another potential single point failure mode of the SIMIS system. Finally, if the outputs of the ATC 2x3 system must pass through some type of comparator, then the comparator circuit again could be a single point failure.

The single point failures in the levitation and lateral guidance control system occur at the unpaired leading and trailing hinge points. There appears to be only one gap sensor, controller, and accelerometer for these magnetic units, of which there are four per vehicle. If any one of these components fail, a race condition could develop, resulting in contact of the levitation or guidance magnet with the guideway.

Fault Containment:

An example of this type of fault is a short circuit in one channel which propagates to another channel. Without more information about the interconnections between the various channels of the ATC, it is impossible to see how they are electrically isolated from one another. It appears that the dual control circuits for each side of a levitation hinge are electrically independent, and therefore electrical faults do not propagate from one to the other.

Error Containment:

According to the Transrapid operating rules, the vehicle must be stopped at the next safe-stopping point when an impermissible operating condition is detected. One such condition is the failure of two channels of one of the onboard TMR computers. But how does the non-failed TMR computer detect this condition? If it must rely on a self-reporting algorithm, where the faulty TMR "knows" it has degraded to only a single channel, then it is possible that the failed TMR system would "lie" about its true status, and thus an error would be propagated to the non-failed channel, resulting in an incorrect and unsafe operating mode.

Real-Time Error Masking:

Without knowing how the six channels of the onboard ATC computer communicate, it is impossible to determine whether faults are masked in real time or not. In the levitation and guidance control system, there are two ways in which faults are dealt with in real-time. When going over a thermal expansion joint, one of the gap sensors is known to report an incorrect gap reading. The system deals with this by only using both sensor readings if they differ by less than 1.5 mm. When the readings differ by a larger amount, the smaller reading is chosen. Presumably, if one of the gap sensors failed outright, this same algorithm would be used throughout and would only be problematic when going over a thermal expansion joint. At this point a very large reading might be reported, causing the controller to increase the magnetic force between the levitation magnet and the guideway. This would be a serious problem if not for the fact that the effect of this error is mitigated by the actions of the other controller which presumably is causing the proper magnetic force to be exerted by the other magnet in the hinge. This is not error masking per se, but the net result is real-time protection against the effect of an error in the levitation control system.

Degree of Synchronization:

If the dual TMR architectures are not operated synchronously, within some bounded skew, there is no way to use comparison to detect a fault. However, the type of synchronization used and the synchronizing mechanism are not described. The redundant controllers in the levitation and guidance hinges are not synchronized; each acts independently of its partner. In this system, fault coverage is provided by, in effect, averaging the magnetic force applied to the hinge point by both controllers.

Degree of Consensus:

In the SIMIS computer an exact bit-wise consensus is required of all compared operations to avoid triggering system shutdown. Presumably, the same requirement is applied within each of the two onboard TMR systems. It is not clear whether the same requirement is applied to the results produced by each TMR with respect to its dual. Clearly, the levitation and guidance controllers do not employ any type of consensus algorithm in executing their fault tolerant operations.

Input Consistency:

One of the requirements for producing bit-wise identical outputs is that redundant channels operate on bit-wise identical inputs. This requirement is often overlooked in the design of redundant computers. No information is available that indicates that such a requirement was addressed for the TMR computers.

Failure Modes Covered:

There is no discussion of fault duration for the onboard ATC computer, or for any effort at recovering a channel which is deemed failed after a transient fault. Hence, it is impossible to say what the system does to handle any other than permanent faults. However, the levitation and guidance control system would operate in the same way whether a fault is permanent, transient or intermittent. What is unclear is how a faulty component is detected, diagnosed and replaced in this system.

Common Mode Faults:

There is no discussion of common mode hardware faults in the references, so presumably no provision for this possibility has been made. However, there is a great deal of emphasis on software fault protection through the standard techniques of structured programming and extensive testing for fault avoidance and fault removal, respectively. There is no discussion of recovery block, N-version software or other means of tolerating common-mode software faults in real-time.

Output Errors:

Apparently, the dual TMR channels of the onboard ATC computer are capable of independently committing output errors since each is required to be able to transmit (and receive) data. The communication is conducted with the CCF which must have some mechanism for detecting errors in these redundant messages and some strategy for handling this situation. It is unclear whether or not one TMR system could lock out the other if it believed the other had failed. If this proviso is not made, presumably one TMR could take an action which the other did not agree to. For example, they could differ on a decision as to whether or not to bring the vehicle to an emergency stop or as to the location of a safe-stopping point. Furthermore, it is not clear as to how information is presented to the various operators in the system. Is there a separate display of data from each onboard TMR computer, or are the results compared automatically and then presented on operator displays? For the levitation and guidance controllers, output errors go directly to the controlled magnets.

Graceful Degradation:

The workload of the onboard ATC controllers is not reduced as a result of accumulated faults. When the number of failed channels exceeds a pre-determined limit, the vehicle is brought to an emergency stop. The levitation and guidance system also has no provision for a graceful degradation. Presumably, a vehicle is not cleared for departure without a full working complement of levitation and guidance equipment since the existence of one faulty component here would introduce a single-point failure mode into the system. However, the propulsion system does include a graceful degradation plan. Power is supplied by two separate power conditioning systems. If either fails, the other can still propel the vehicles at reduced speed.

Operator Intervention:

The possibility for operator intervention is provided for the onboard ATC computer. However, the levitation and guidance control system is completely autonomous.

14.5. Analytical Models and Results

Section 11 of this report described the approach to dependability evaluation of computer architectures using analytical modeling techniques. Using these techniques, one can evaluate the reliability, safety, availability and maintainability of the candidate architectures and compare these to the desired requirements. This section illustrates this step of the design-for-validation methodology by applying it to the gap control system and the Automatic Train Control (ATC) computer both of which are part of the Onboard Control Computer (OCC) system of the TR07.

14.5.1 Gap Control System Model

The gap control system that controls the gaps between the levitation and guidance magnets and the guideway has been described in Section 14.3. Reference [45] has identified a number of failures which can result in loss of levitation or guidance: loss of power supply, faulty drive control, software defect, loss of synchronism followed by set-down, entry into stator short-circuit loop before the neutral point, and magnet gap control loop malfunction. Modeling of all of these failure modes is beyond the scope of this study. Instead, we will illustrate the methodology by modeling one specific failure mode of the gap control system, i.e., magnet gap control loop malfunction, that can lead to loss of safe hover, i.e., contact with the guideway.

The gap control system consists of three types of components: gap sensors, accelerometers and choppers (part of controller). A failure modes and effects analysis performed in reference [45] and reported in the form of a fault-tree [Figure 3.1A of 45] has identified the following component failures that can lead to a loss of safe hover.

If the sensor reports smaller than actual gap, or the accelerometer reports higher than actual speed, or the chopper fails at zero current/voltage, then the magnet gap control loop will malfunction leading to the vehicle dropping on its skids. Alternatively, if the sensor reports larger than actual gap, or the accelerometer reports lower than actual speed, or the chopper fails at full current/voltage, then the magnet gap control loop will malfunction leading to the magnets striking the guideway. The effect of either event is a loss of safe hover.

A markov process model of the gap control system has been developed for a quantitative evaluation of the probability of loss of safe hover during a typical trip. Figure 14-5 shows the markov states and transitions between states. Tables 14-2 and 14-3 describe the markov states and the state transitions, respectively. Table 14-4 defines the parameters of transition rates. The following assumptions were made in constructing the markov model. Transitions in parentheses refer to state transitions in the markov model.

1. The gap control system description and the fault-tree analysis, both described in reference [45], form the main basis for the model.

2. There are 28 hinges per vehicle and a failure of the gap control system at any one of them will lead to loss of hover.

3. There are 2 controllers per hinge (for a total of 56 controllers per vehicle) and as long as at least one controller at a hinge is operating correctly, the gap at that point will be controlled correctly. That is, a correctly operating controller can overcome the forces being applied by the failed controller.

4. As soon as one controller at any of the 28 hinges fails (transition 1-2), the vehicle is brought to a stop at the next safe stopping point (transition 2-30).

5. While the vehicle is moving to the next safe stopping point, with one controller failed, the loss of a second controller at the affected hinge will lead to a loss of safe hover (transition 2-31).

6. While the vehicle is moving to the next safe stopping point, with one controller failed, the loss of a second controller at any other hinge (transition 2-3) is tolerated with 100% coverage.

7. The gap control system continues to tolerate additional controller failures (transitions to states 4 to 29), up to 28 maximum, as long as they are all at different hinge points, i.e., as long as there is at least one functioning controller at each hinge point.

8. The failure of an accelerometer, or one of the two gap sensors per controller, or the chopper circuit, in the manner described previously, will lead to the failure of the associated controller.

9. The fault detection coverage for sensors, accelerometers and choppers is 100%. That is, a failure of any of these components is detected immediately with a probability of 1.0 and a decision is made to stop the vehicle at the next safe stopping point.

10. The leading and trailing edges of the leading and trailing magnets, respectively (a total of 4) apparently only have a single controller. The failure of any of these 4 simplex

controllers will probably lead to a loss of hover. This would normally be the dominant component in the probability of hover failure. However, this failure mode has been ignored in the present analysis since it is not perfectly clear from [45] whether, in fact, there is an active gap control system at the leading and trailing edges of the leading and trailing magnets. Also, it seems inconceivable that the designers of the TR07 would have left 4 single point failures in the gap control system, in the form of four simplex controllers, that could lead to a loss of safe hover.

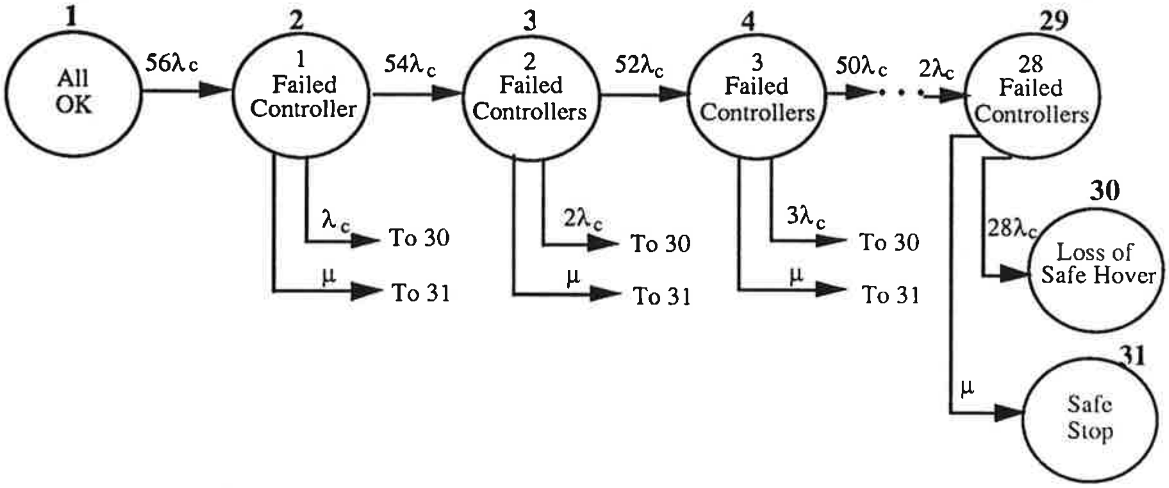


Figure 14-5. Markov Model of TR07 Gap Control System

State No.	State Description
1.	All OK. All 56 Controllers at 28 hinges operational.
2.	One controller at one hinge failed. Gap control system operational. Vehicle to stop at the next safe stopping point.
3.	Two controllers at two different hinges failed. Gap control system operational. Vehicle to stop at the next safe stopping point.
• • •	
29	28 controllers, one at each of the 28 hinges, failed. Gap control system operational. Vehicle to stop at the next safe stopping point.
30	Both controllers at one hinge failed. Loss of safe hover.
31	Vehicle stopped at a safe stopping point.

Table 14-2. Markov States of the TR07 Gap Control System Model

State Transition From - To	State Transition Description
1-2	One of 56 controllers fails. Hinge degrades to a single controller.
2-3	One of 54 controllers at the 27 undegraded hinges fails.
2-30	Second controller at the degraded hinge fails. Loss of safe hover.
2-31	Vehicle reaches a safe stopping point.
3-4	One of 52 controllers at the 26 undegraded hinges fails.
3-30	Second controller at one of two degraded hinges fails. Loss of safe hover.
3-31	Vehicle reaches a safe stopping point.
•	
•	
•	
29-30	One of 28 controllers at the 28 degraded hinges fails. Loss of safe hover.
29-31	Vehicle reaches a safe stopping point.

Table 14-3. State Transitions for the TR07 Gap Control System Model.

Parameter	Description
λ_c	Controller failure rate = $\lambda_a + \lambda_{ch} + 2\lambda_s$
λ_a	Accelerometer failure rate
λ_{ch}	Chopper failure rate
λ_s	Gap sensor failure rate
μ	v/d
v	Average Speed
d	Average distance between safe stopping points.

Table 14-4. Parameters for the TR07 Gap Control System Model.

The markov model can now be solved numerically if the component failure rates were known. However, in the absence of specific failure rate data, the following general comments can be made. There are a total of 31 states in the model. As such, this is a modest size model and any markov modeling package, such as the MARK II package at Draper Laboratory, can be used to obtain the state occupancy probabilities as a function of time. In the initial state, say, when the train is at a station, and the gap control system is operating correctly, the system is in State 1 with a probability of 1.0. All other states have an initial probability of zero. Assuming a typical trip time of 1 hour, the model can be solved numerically to provide the state occupancy probabilities at the end of the typical trip. The probability of a loss of safe hover during this 1 hour trip then is the probability of being in State 30 at the end this time period. One can obtain other dependability measures as well from this model. For example, the overall likelihood of not completing a trip successfully due to a malfunction of the gap control system is the sum of probabilities of States 30 (loss of safe hover) and 31 (unscheduled stop at a safe stopping point). The sum of States 2 to 29 is the probability of operating in degraded mode, i.e., operational with failed components.

Another point that can be illustrated about markov process models is the technique for handling proliferation of states. As can be seen in this illustrative example, even a few simple failure modes can lead to a large number of states. Although solving a 31 state model or for that matter even a 300 state model is quite trivial with today's computer technology, the proliferation of states leads to a cognitive overload. After a certain point, the model becomes too complex for a human being to comprehend. This can lead to errors in defining all the states and transitions between states. Therefore, to reduce the model complexity and increase the confidence in the correctness of the model, it is prudent to keep the total number of states and state transitions to a modest level. Two commonly used techniques to trim the number of states are state aggregation and truncation. In the present example, one can make the simplifying assumption that the probability of 4 or more controller failures in a 1-hour trip is insignificantly small. States 5 to 29, which represent 4 or more controller failures, can therefore be aggregated with state 4. This reduces the total number of states in the model to 6 and furthermore, it provides a conservative estimate (i.e. an upper bound) of the probability of hover failure. Typically, truncating the model at 3 failures or so can be quite prudent for short mission times. A technique for ascertaining the degree of inaccuracy due to truncation is to re-run the model with one more failure and determine if there is a significant change in the results.

14.5.2 Automatic Train Control Computer Model

The functions performed by the Automatic Train Control (ATC) computer as well as the ATC architecture were described in Section 14.3. Unlike the gap control system, reference [45] or other available TR07 literature does not contain any FMEA or fault-tree analysis of the ATC. However, many of the common-mode faults that have been identified as potential causes of the failure of the gap control system are equally applicable here. These include loss of power supply, software defects, loss of synchronism between redundant channels and so on. Once again, modeling of all of these failure modes is beyond the scope of this study. Instead, only one specific ATC computer failure scenario, computer redundancy management, is modeled here for the purpose of illustrating the methodology. The ATC computer markov state model is shown in Figure 14-6. Tables 14-5, 14-6, and 14-7 describe the model states, state transitions, and the model parameters, respectively. The following assumptions have been made in constructing the model. Transitions in parentheses refer to state transitions in the markov model.

1. The ATC computer redundancy management strategy, as described in Section 14.3, forms the main basis for the model.

2. Loss of one channel in one TMR system (transitions 1-2 and 1-3) or both TMR systems (transitions 2-4 and 3-4) is tolerated with 100% coverage, i.e., a probability of 1.0. ATC computer system continues to operate correctly.

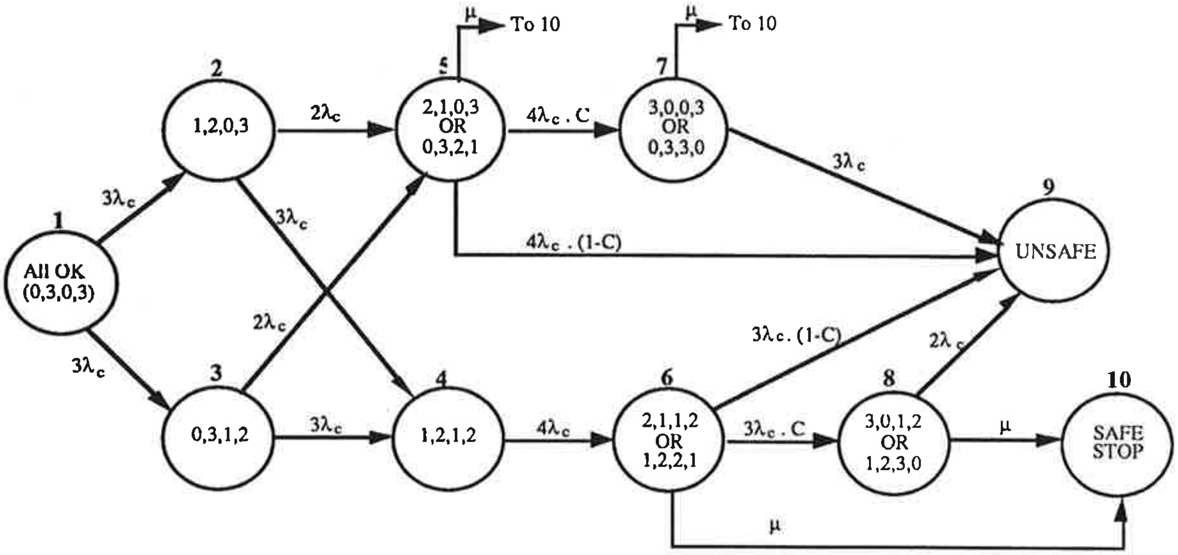
3. Loss of a second channel in a TMR system (transitions 2-5 and 4-6), regardless of the number of operational channels in the other TMR system, leads to a stop at the next safe stopping point.

4. Loss of the second channel in a TMR system is detected with 100% coverage and its effects are masked with 100% coverage.

5. Loss of an additional channel in either TMR system, given that 2 channels in one TMR system have already failed, is tolerated with coverage C (transitions 5-7 and 6-8). The lack of coverage (1-C) of this event leads to the unsafe state (transitions 5-9 and 6-9).

6. Loss of an additional channel, given that all 3 channels have failed in one TMR system, leads to the unsafe state (transitions 7-9 and 8-9).

The ATC computer model can be easily solved numerically, provided the quantitative values of the computer failure rates and other parameters were known. In the absence of such data, the following general comments apply. The initial probability at the start of a trip is 1.0 for state 1 and zero for all other states, assuming that the ATC computer has been diagnosed to be fault-free at the station. The likelihood of being in an unsafe state due to a malfunction of the ATC computer for a typical 1-hour trip is the probability of being in state 9 at the end of a 1 hour period. The sum of states 9 and 10 provides the probability of not completing a typical 1-hour trip successfully. The complement of this, i.e., the sum of states 1 to 8, is the probability of completing the trip successfully, as far as the ATC computer operations are concerned. In the absence of detailed knowledge of the redundancy management (RM) strategy of the ATC computer, certain assumptions (3, 4 and 5 above) have been made regarding the coverage of a channel failure. However, the model can easily be modified to reflect a different RM strategy. For example, if the coverage of a channel failure in states 2 or 3 is non-unity, additional transitions can be



Legend:(# failed chs. in TMR1, # operational chs. in TMR1, # failed chs. in TMR2, # operational chs. in TMR2)

Figure 14-6 Markov Model of TR07 Automatic Train Control Computer

added from these states to state 9 and the transition rates to states 4 and 5 can be reduced accordingly. Similarly, the model can be evaluated using a range of values for the coverage parameter C as discussed in the following.

A point that can be illustrated about the power of analytical modeling is the sensitivity analysis. The impact of any of the system parameters on the likelihood of being in the unsafe state, for example, can be easily calculated. To do this, one simply has to solve the markov model for different values of, say, the channel failure rate. Such an analysis is very useful in the early design stages for a number of reasons. It can help in various design trade-offs by analyzing the impact of different design strategies such as reducing component failure rate versus increasing failure coverage. It can focus designers' attention on the weak links in the system dependability by identifying parameters to which the system is specially sensitive. Another use of the sensitivity analysis is to compensate for the lack of precise knowledge about the system parameters in the early design stages. For example, in the early stages of the computer architecture design, the details of the channel hardware may not be fully developed. Therefore, the channel failure rate data at this stage will only be a gross estimate. In such a case, the various probabilities of interest can be computed for a range of failure rates centered around the estimated failure rate.

State No.	Failed and Operational Chs. in TMR1 & TMR2	State Description
1	0,3,0,3	All OK. All 6 channels operational
2	1,2,0,3	One Ch. in TMR1 failed. TMR2 all OK. ATC operational
3	0,3,1,2	TMR1 all OK. One ch. in TMR2 failed. ATC operational
4	1,2,1,2	One ch. in TMR1 failed. One ch. in TMR2 failed. ATC operational.
5	2,1,0,3 OR 0,3,2,1	Two chs. in TMR1 failed. TMR2 all OK. OR TMR1 all OK. Two chs. in TMR1 failed. ATC operational. Vehicle to stop at the next safe stopping point.
6	2,1,1,2 OR 1,2,2,1	Two chs. in TMR1 failed. One ch. in TMR2 failed. OR One ch. in TMR1 failed. Two chs. in TMR2 failed. ATC operational. Vehicle to stop at the next safe stopping point.
7	3,0,0,3 OR 0,3,3,0	All chs. in TMR1 failed. TMR2 all OK. OR TMR1 all Ok. All chs. in TMR2 failed. ATC operational. Vehicle to stop at the next safe stopping point.
8	3,0,1,2 OR 1,2,3,0	All chs. in TMR1 failed. One ch. in TMR2 failed. OR One ch. in TMR1 failed. All chs. in TMR2 failed. ATC operational. Vehicle to stop at the next safe stopping point.
9	Various	UNSAFE STATE
10	Various	Vehicle stopped at a safe stopping point.

Table 14-5. Markov States of the TR07 ATC Computer Model

State Transition From-To	State Transition Description
1-2	One of 3 chs. in TMR1 fails.
1-3	One of 3 chs. in TMR2 fails.
2-4	One of 3 chs. in TMR2 fails.
2-5	One of 2 chs. in TMR1 fails.
•	
•	
5-7	One of 4 operational channels in TMR1 or TMR2 fails while vehicle is moving to the next safe stopping point. Channel failure is covered.
5-9	Same event as 5-7 except that the failure is not covered. ATC computer fails. Vehicle unsafe.
5-10	Vehicle arrives at a safe stopping point.
•	
•	
•	
8-9	One of two remaining operational channels fails. ATC computer fails. Vehicle unsafe.
8-10	Vehicle arrives at a safe stopping point.

Table 14-6. State Transitions for the TR07 ATC Computer Model.

Parameter	Description
λ_C	Channel failure rate
C	Coverage for successfully recovering from a channel failure given that 2 channels in a TMR are already failed.
1-C	Lack of coverage for the same event.
μ	v/d
v	Average Speed
d	Average distance between safe stopping points.

Table 14-7. Parameter for the TR07 ATC Computer Model.

The sensitivity analysis can also be used to determine the effect of various coverage strategies on the system dependability. In the present analysis, assumptions 4, 5, and 6 describe the coverage of an additional channel failure for three different states of the system.

14.5.3 Overall Vehicle Model

In order to compute the overall probability of a given event, such as the likelihood of an unscheduled stop at a safe stopping point due to malfunction of *any* part of the Onboard Control Computer system, the results of a number of the models such as the two discussed here must be combined. This is accomplished by "adding" the probabilities of relevant states from different models, for example, states 31 and 10 from the previous two models, respectively. The MARK II markov modeling package, mentioned earlier, has the facilities to combine state probabilities according to the laws of probability. The details of this are beyond the scope of this study. Suffice it say that such a computational tool is extremely invaluable since one can model each independent part of the system separately, thus keeping the number of states in each model reasonably low. In the absence of such a tool, one would have to create an all encompassing super model which will have the number of states equal to the product of the number of states in each of the sub-models.

By combining the results of sub-models, one can predict the overall reliability, availability, and safety of the system, taking into account failures of all subsystems.

14.5.4 Basler and Hofmann Evaluation of TR07

In its report on the German high-speed maglev train safety requirements [45], reference is made to a quantitative analysis of the safety of the overall TR07 system performed by Basler and Hofmann, a Swiss consulting firm under contract to the German maglev system developer. They describe 21 measures which could be used to reduce the risk of fatalities. The application of these measures could *reduce the assessed risk level by a factor of 10* to 0.02 fatalities per 62.5 million miles per person. Assuming an average capacity of 100 passengers per vehicle and an average speed of 250 miles per hour, this can be restated as a probability of a fatality of 10^{-5} per vehicle per hour of operation. If the measures are not implemented, the probability of a fatality would be 10^{-4} per vehicle per hour of operation.

For avionics applications, typically the control computer system is allowed to contribute 10 per cent of the total unreliability of a system. The level of automation in TR07 is comparable to or greater than that of most avionics applications. Hence, if all suggested fatality risk reduction efforts were applied, the contribution to the unreliability of the overall TR07 system made by its control computer system could be estimated at 10^{-6} per vehicle per hour of operation. This can be illustrated as follows. If 1 billion trips, each of 1 hour duration, were undertaken by a fleet of TR07 maglev vehicles, then 10,000 deaths or injuries could be expected. If we assume that maglev trains have the same number of scheduled departures per day as planes, i.e. 14,000 per day, and that each trip averages one hour, these 1 billion trips will take approximately 195 years. Over that period of time, 50 injuries or deaths per year could be expected, of which 5 would be due to a failure of the control computer system. However, without knowing the safety and reliability requirements of the TR07 control computer system, we cannot compare the results of the Basler and Hofmann analysis with the system requirements.

14.6. Open Issues

This preliminary assessment of the control computer system for Transrapid has uncovered many open issues and areas where insufficient information has prevented a complete and thorough evaluation. The following discussion summarizes and highlights the open issues which the previous analysis has uncovered.

It should be emphasized here that the purpose of this exercise is to illustrate the application of the Verification Methodology and not to pass judgement on the Transrapid control system. Furthermore, the open issues are just that. They do not necessarily point to deficiencies in the Transrapid system. If enough detailed design information is provided, these issues may be resolved satisfactorily.

The Transrapid contention is that only the software of the onboard ATC is to be fully validated since the hardware has already been verified for other applications by the German Federal Railways. This brings up the issue of verification vs validation. Verification is the process whereby a component or system is shown to meet its requirements. Validation is the process of showing that the requirements are correct. If the onboard ATC system has been verified for other applications, there is no guarantee that it will work correctly when integrated into a new and very much more demanding application. Clearly, the requirements of the Transrapid system are much more demanding than those of any other guided ground transportation system now in existence. Furthermore, if the dual SIMIS computer has been validated, has it been validated in the TMR configuration planned for the onboard ATC? Or is this a completely new SIMIS computer?

Many questions about the architecture of the onboard ATC computer remain unanswered. For example, how do the two ATC TMR computers communicate with each other? How does the onboard ATC prevent a failed channel from trying to stop the vehicle inappropriately? In fact, almost no information exists on the redundancy management of this safety critical computer system.

Similarly, there are several outstanding questions about the fault tolerant properties of the safety critical levitation and guidance control system. For example, what type of redundancy is provided for the controllers of the leading and trailing magnets in the linked magnet trains on either side of the vehicle? These points are not protected by the hinge design of the other magnets in the train. There appears to be only one controller and one gap sensor at this point. What happens if this controller or this gap sensor fails? It is interesting to note that the Transrapid developers have identified several failures which could result in the loss of levitation or guidance. These include a loss of the onboard power supply, a faulty drive control, a software defect, and a loss of synchronism followed by set down. However, they do not consider the failure of an individual guidance or levitation magnet to be a concern because of the requirement, which they apparently believe they have met with their design, for enough autonomous, redundant units to prevent loss of the levitation function during the maximum number of conceivable failures. This may be true of a passive magnet failure, which would result in the vehicle contacting the guideway from above. However, it does not address the "race" condition whereby the attractive force between the levitation or guidance magnets could contact the guideway from below when the attractive magnetic force which levitates the vehicle becomes too great and completely closes the gap between the vehicle and the guideway.

Several potential problems exist with the vehicle detection system. This is not the location used by the propulsion system to move the vehicle along. It is used by (1) the CCF for high level traffic control and (2) the vehicle when it is performing an emergency stop by means of the secondary braking system. Unless the CCF resolves any disagreements between where the vehicle thinks it is and where the propulsion system thinks it is, there could be serious problems. The tags used in the INKREFA system are subject to the obvious problems of falling off, being obscured by dirt or snow, or becoming damaged and therefore unreadable. Finally, it is not clear how the six channels of the onboard ATC computer are guaranteed to have a consistent view of the values reported by the four channels of the INKREFA system. In a redundant computer system, it is imperative that all non-failed channels receive identical copies of all inputs. Otherwise, the channels will come to different conclusions about the state of the system and diverge in their operation, thereby becoming desynchronized. When this happens, how can the system recover sufficiently to ensure that the vehicle can be brought to a safe stop under the control of the onboard ATC system?

Many robust methods of protecting data from transmission errors exist. The method chosen here of "inverse telegrams" and dual transmissions is not discussed in enough detail to evaluate the adequacy of the error protection it provides.

A general observation may be made regarding the requirement that "operational points which record, transmit, or process safety-relevant information...must be reliable signaling technology components" which comply with a certain standard (DIN VDE 0831). However, an exception is made for systems which do not meet one of these requirements. Although the wording is very confusing here, it appears that when two units are required to guarantee a safety-oriented action, three units must be used if the components do not meet the DIN standard. This approach to improving reliability commits a common, serious, and costly reasoning error by assuming that more redundancy alone makes a safer or more reliable system. In fact, greater redundancy per se only increases the overall failure rate! Only with proper redundancy management can increased redundancy actually bring about increased reliability.

It is not entirely clear why, in a system which purports to be fully automatic, there is such a strong need for reliable voice communication and what if any improvement this feature provides to system reliability. Furthermore, the Transrapid design seems to require a great deal of human intervention at the CCF level to handle dispatching and scheduling decisions. The man/machine interface is a very difficult one to manage. It is generally conceded that the software for a fully automated system is easier to specify, implement, test, and validate than software which must accept a lot of human inputs, since the range of these inputs is impossible to predict and control.

There is no discussion of which part of the control system actually commands the switch control mechanism to move from one setting to another. Thus there is no discussion of how a failure of this communication link could lead to a catastrophic movement of the switch at an inopportune time, or what measures are in effect to protect against this failure mode.

There is no discussion whatsoever of the computer architecture of the safety critical CCF computers. The propulsion system has been deemed not to be safety critical and this is used to justify less attention being paid to its fault tolerant properties. This is a very difficult assumption to accept in the absence of any evidence or explanation. It seems intuitively clear that the WCCs which directly control vehicle movements along the guideway have the potential power to cause accidents.

The analytical reliability models of parts of the Transrapid OCC cannot be evaluated numerically without data on component failure rates, reconfiguration times, coverage for fault detection and recovery, etc.

14.7 Conclusions

In this section, the verification methodology developed during the course of this study was applied to the control computer system of the German Transrapid system designated TR07. In accordance with the sequential approach specified by the methodology, we began by describing the mission requirements and the functional requirements of the TR07 control computer architecture. Next we discussed the overall computer architecture used in TR07 and, to some extent for selected subsystems, the redundancy management strategies employed to meet the stated mission requirements. Using the approach presented in Section 10, a critical failure modes and effects analysis (FMEA) was performed on the TR07 control system. Analytical models for two selected subsystems were then developed. Finally, we presented some open issues which were raised by the FMEA and the discussion of redundancy management strategies. Despite the fact that we only had a limited amount of information on which to base our investigation, we were able to show that this methodology would be a valuable and effective tool in verifying the design of the Transrapid control computer system. Furthermore, due to the power of the verification methodology to organize and present information in a clear and coherent manner, the control computer system for Transrapid is presented here as a cohesive, logical, and intelligible totality, which is not the case for its presentation in the source materials upon which this study is based. Given the fact the control system is fundamental to the overall design of a maglev transportation system, the verification methodology needs to be used from the start as the basis of developing a safe, reliable, and tractable system which can be shown to meet its requirements.

This requirement applies to the total computer system, including the hardware and software of the on-board vehicle, wayside and central control facility computers. The reliability requirement for maglev relates to the probability of successfully completing a trip and the probability of not completing a trip due to computer system malfunction was specified to be 10^{-6} per vehicle per hour.

The overall reliability and safety requirements for maglev may be illustrated as follows. If 1 billion trips, each of 1 hour duration, were undertaken by a fleet of maglev vehicles, then all except 1000 trips should be completed successfully. Of the 1000 trips in which the vehicles did not arrive at their destination without incident, only 1 would result in a catastrophic accident. If we assume that maglev trains have the same number of scheduled departures per day as planes in the US, i.e. 14,000 per day, and that each trip averages one hour, these 1 billion trips will take approximately 195 years. Over that period of time, in a system which met the stated reliability and safety requirement, there would only be five incomplete trips per year and a total of one catastrophic accident attributable to the failure of the control computer system.

The availability of the maglev transportation system is going to play a very important part in the public's acceptance of this mode of transportation. For the domestic US commercial airlines, the availability of the airliners approaches or exceeds 99 per cent. Less than 1 per cent of the flights are delayed or cancelled due to mechanical, electrical, hydraulic or other aircraft system related failures. It is obvious that the maglev transportation system will have to match or exceed this level of dependability in order to be accepted by the public. Availability requirement for maglev was specified as follows.

" The maximum acceptable probability of not being dispatch ready for a trip for each maglev vehicle will be 10^{-2}."

This requirement applies to all the subsystems on-board each vehicle. The unavailability apportionment for the control computer subsystem is assumed to be one tenth of this, or 10^{-3} per vehicle per trip. That is, only one tenth of the unavailable vehicles will be stuck due to on-board control computer system failures.

The principal functions to be performed by the Maglev computer control system, control, protection, and supervision, were mapped to specific computation sites within the overall control computer architecture.

A distributed and hierarchical control architecture was defined for maglev. Its principal subsystems are an on-board vehicle computer system for each vehicle, a wayside zone computer system for each zone, and a central facility computer system. Two variations of the basic architecture, which represent extremes of a continuum, were developed. In the Zone Control Architecture (ZCA), the primary responsibility for train control rests with the wayside zone control computers, with the on-board system providing backup and consistency checking. Train protection is distributed among the three subsystems. In the Smart Vehicle Architecture (SVA), the on-board computer has the primary responsibility for train control, with the wayside zone computers providing backup and consistency checking. The functions relating to vehicle protection are again distributed among the three subsystems.

In both architectures, the vehicle protection subsystem operates independently from the vehicle control functions, and therefore provides a fail-safe mode of operation. In cases where the speed or position of a vehicle exceed safety thresholds, these protection functions can override the actions of the control functions and assume control of the vehicle. The supervision function is performed by the central facility computer system, which also includes major computing subsystems located in stations. However, the supervisory data such as the travel or route profile for a given vehicle which is needed by the primary control computer to adequately perform its function is transferred to that computer on at least a daily basis and may be updated more frequently. This form of operation views all normal train travel as planned in advance and all passengers riding in reserved seats. However, either architecture could be adapted to a demand driven schedule which requires more real-time planning capability.

In general, the functions were assigned to a particular computational site based on where it can best be performed. For example, the control of levitation, guidance, secondary suspension, and on-board systems like air conditioning and lighting all require the control of on-board actuators. Furthermore, the sensors needed to obtain feedback information for these systems are also on-board. Hence, the control of these functions should be performed by an on-board computer. Other criteria used to assign functions to particular computational sites include minimization of communication and data latency, minimization of adverse effects of failures of communication links and processors, and ability to validate the architecture. Functions which can be performed by an on-board computer with a minimum of communication overhead is that of emergency stopping, emergency speed control, and emergency position control. The emergency which these functions are intended to address is the failure of the guideway propulsion and primary braking capability. The power for these emergency operations comes from batteries carried on-board for that purpose. The vehicle must be able to reduce its speed, continue onto a safe stopping area and stop there so that passengers can disembark safely from the vehicle and the elevated guideway.

The rationale behind the ZCA is that, unlike conventional transportation systems in which the power for vehicle propulsion is on-board, the Maglev system is powered from the guideway. The propulsion control must be co-located with the power converters, i.e. within the same zone control computers, since the required iteration rate is so high as to preclude any communication latency. Since the wayside zone controllers must communicate with each other to coordinate the speed of the vehicle as it passes form one zone to the next, they can also perform the function of safe vehicle separation and vehicle position which are related to vehicle speed. The communication between wayside systems can be carried out through very reliable media such as redundant fiber optic cables. By locating sensors in the guideway, the vehicle location and route integrity functions can also be performed by the wayside using sensors embedded in or alongside the guideway for these purposes. Route control, or the direction of the vehicle through a switch, can also be performed from a wayside computer, especially if the switching mechanism is that of a moveable section of guideway. The ZCA most closely resembles conventional railway control systems without the attendant communication overhead that characterizes systems controlling on-board propulsion from the wayside.

The rationale behind the SVA is that an autonomous vehicle can most easily direct its own motion since it is in a position to obtain information about its own state and the state of its surroundings. It must be able to perform these functions for emergency purposes anyway. Hence, it may as well perform them for normal operations. It can easily communicate speed commands to wayside propulsion control systems using radio communication. Information about its speed, position, and acceleration are also easily obtained from a combination of Global Positioning System and low cost Inertial Navigation Systems such as those based on micro-mechanical instruments, the technology for which will be available by the time the U.S. Maglev Transportation System reaches the prototype development stage. With information about its position and speed, it can perform the functions of safe train separation and vehicle position control by communicating with vehicles both ahead of it and behind it on the guideway. By using on-board sensors, it can also perform route integrity checks both for alignment and obstacle detection. Furthermore, it can easily direct its movement through switches, i.e. perform route control operations, especially if the switches in use do not involve moveable sections of guideway but rather some vehicle-borne steering mechanism. The SVA most closely resembles the most advanced control systems being installed on conventional and high speed rail systems.

A principal subsystem of the SVA is the on-board control computer (OCC). A baseline of the OCC was developed using the Fault Tolerant Parallel Processor (FTPP) which is one of the Draper fault tolerant computer building blocks. A triplex configuration with an additional spare processor was chosen as the baseline. A fault tolerant interface and communications protocol between the OCC and the zone control computer was also developed. Various dependability attributes of the OCC, including reliability, availability, safety, and maintainability, were modeled analytically and compared with the maglev dependability requirements. As a result of this analysis, the OCC architecture was modified slightly, namely by adding spares, to meet the availability requirement.

This study also addressed the general issues related to the overall validation and verification of the maglev control computer system. Qualitative and quantitative evaluation criteria for the maglev control computer system were developed. Means of increasing the software reliability were discussed. A hierarchical approach to validation of fault tolerant claims was also developed and discussed. Dependability modeling techniques were presented and these were complemented by an empirical test and evaluation plan, including hardware fault and software error injection methods.

The verification methodology developed during the course of this study was applied to the control computer system of the German Transrapid system designated TR07. In accordance with the sequential approach specified by the methodology, we began by describing the mission requirements and the functional requirements of the TR07 control computer architecture. Next we discussed the overall computer architecture used in TR07 and, to some extent for selected subsystems, the redundancy management strategies employed to meet the stated mission requirements. Using the approach presented in Section 10, a critical failure modes and effects analysis (FMEA) was performed on the TR07 control system. Analytical models for two selected subsystems were then developed. Finally, we presented some open issues which were raised by the FMEA and the discussion of redundancy management strategies. Despite the fact that we only had a limited amount of information on which to base our investigation, we were able to show that this methodology would be a valuable and effective tool in verifying the design of the Transrapid control computer system. Furthermore, due to the power of the verification methodology to organize and present information in a clear and coherent manner, the control computer system for Transrapid was presented here as a cohesive, logical, and intelligible totality. Given the fact the control system is fundamental to the overall design of a maglev transportation system, the verification methodology needs to be used from the start as the basis of developing a safe, reliable, and tractable system which can be shown to meet its requirements.

A principal conclusion of this study is that the design for verification methodology developed for real-time mission- and safety-critical aerospace applications can be successfully applied to the maglev control computer system. A key to the successful application of this methodology is the early involvement of computer system designer in the overall conceptual design of the subject vehicle to be controlled. Another major conclusion of this study is that the maglev control computer design is driven primarily by dependability requirements. The performance requirements are well within the current state-of-the-art microprocessors. The dependability requirements, on the other hand, are extremely challenging. In particular, meeting the safety and reliability requirements without sacrificing availability will require the use of sophisticated architectures that possess fault tolerance, graceful degradation and reconfiguration attributes.

16. References

[1] Dorer, R.M. and W.T. Hathaway, "Safety of High Speed Magnetic Levitation Transportation Systems," November 1990.

[2] "Assessment of the Potential for Magnetic Levitation Transportation Systems in the United States, A Report to Congress, Report Supplement," U.S. Dept. of Transportation, Federal Railroad Administration, June 1990.

[3] "Final Report, National Maglev Initiative," Government-Industry Workshop, Argonne National Laboratory, Argonne Illinois, November 1, 1990.

[4] Lala, J.H., et al., "Advanced Information Processing System for Advanced Launch System: Avionics Architecture Synthesis," NASA Contractor Report-187554, Draper Laboratory, Inc., Cambridge, MA, September 1991.

[5] Johnson, B.W., "Design and Analysis of Fault Tolerant Digital Systems," Addison-Wesley Publishing Company, Reading, Massachusetts, 1989.

[6] Lala,J., "A Design Approach for Ultrareliable Real-Time Systems," *IEEEComputer*, pp.12-22, May 1991.

[7] Gaumer, R.L. et al., "Safety Relevant Observations on the ICE High Speed Train (Draft)," U.S. Department of Transportation, Federal Railroad Administration's Office of Research and Development and Office of Safety, August 1991.

[8] MIL HDBK-217E, "United States Department of Defense Military Standardization Handbook: Reliability Prediction of Electronic Equipment," 2 January 1990.

[9] "Assessment of the Potential for Magnetic Levitation Transportation Systems in the United States, A Report to Congress, "U.S. Dept. of Transportation, Federal Railroad Administration, June 1990.

[10] "Final Report, New York State Technical and Economic Maglev Evaluation," prepared for the New York State Energy and Research Development Authority and the New York State Thruway Authority by Grumman Space and Electronics Division, Bethpage, NY, June 1991.

[11] Gaumer, R.L. et al., "Safety Relevant Observations on the TGV High Speed Train (Draft)," U.S. Department of Transportation, Federal Railroad Administration's Office of Research and Development and Office of Safety, August 1991.

[12] Heinrieich, K. and R. Kretzschmar, et al., "Transrapid Maglev System," Hestra-Verlag, Darmstadt, West Germany, 1989.

[13] Anagnostopoulos, G., Personal communication with George Anagnostopoulos, the Contracting Officer's Technical Representative, Volpe National Transportation Safety Center, U.S. Dept. of Transportation, Cambridge, MA.

[14] Smith, M.E. and R.R. Resor, "Keeping Trains on Schedule: On-line Planning Systems for the Advanced Railroad Electronics System (ARES)," July 1990.

[15] Bing, A.J. et al., "Collision Avoidance and Accident Survivability: Interim Report to Volpe National Transportation Systems Center", Volume 1, Reference No. 63058, Arthur D. Little, Inc., Cambridge, MA, September 1991.

[16] Rahn, T. et al., "ICE: High Tech on Rails," Hestra-Verlag, Darmstadt, West Germany, 1991.

[17] "Advanced Train Control Systems: Overview System Architecture," Draft 3, ARINC Research Corporation, Anapolis, Maryland, July 1987.

[18] Peterson, C., "Monorail Control System Safety," Journal of Electrical and Electronics Engineering, vol. 10, pp. 28-35, Sydney, Australia, March 1990.

[19] D. Avresky, et al, "Fault Injection for the Formal Testing of Fault Tolerance," 22nd International Symposium on Fault Tolerant Computing, Boston, MA, July 8-10, 1992, pp. 345-354.

[20] Lamport, L., Shostak, R., and Pease, M., "The Byzantine Generals' Problem," ACM Trans. Programming Languages and Systems, Vol. 4, No. 3, pp. 382-401, July 1982.

[21] Lamport, L., R. Shostak, and M. Pease, "Reaching Agreement in the Presence of Faults," JACM, Vol. 27, No. 2, pp. 228-234, April 1980.

[22] Dolev, D., "The Byzantine Generals Strike Again," J. Algorithms, Vol. 3, No. 1, pp. 14-30, 1982.

[23] Lynch, N. and M. Fischer, "A Lower Bound for the Time to Assure Interactive Consistency," Info. Proc. Lett., Vol. 14, No. 4, pp. 183-186, Apr. 1982.

[24] Dolev, D., C. Dwork, and L. Stockmeyer, "On the Minimal Synchronism Needed for Distributed Consensus," IBM Research Report RJ 4292 (46990), May 8, 1984.

[25] Abler, T., "A Network Element Based Fault Tolerant Processor," MS Thesis, Massachusetts Institute of Technology, Cambridge, MA, May 1988.

[26] Harper, R., Lala, J., Deyst, J., "Fault Tolerant Parallel Processor Overview," 18th International Symposium on Fault Tolerant Computing, June 1988, pp. 252-257.

[27] Harper, R., Lala, J., "Fault Tolerant Parallel Processor," J. Guidance, Control, and Dynamics, V. 14, N. 3, May-June 1991, pp. 554-563.

[28] Harper, R., "Critical Issues in Ultra-Reliable Parallel Processing," PhD Thesis, Massachusetts Institute of Technology, Cambridge, MA, June 1987.

[29] Cole, R., "Advanced Information Processing System for Advanced Launch System: Hardware Technology Survey and Projections," NASA Contractor Report-187555, Draper Laboratory, Inc., Cambridge, MA, September 1991.

[30] Harper, R., et al, "Advanced Information Processing System: The Army Fault Tolerant Architecture Conceptual Study," NASA Contractor Report 189632, Draper Laboratory, Inc., Cambridge, MA, June 11, 1991.

[31] Nagle, G., R. Harper, "A Survey of the State of the Business of Fault Tolerant Computing Relevant to Avionics, Aircraft, and Autonomous Vehicle Applications," CSDL-R-2233, Prepared for Lockheed Missile and Space Company, Sunnyvale CA, Draper Laboratory, Inc., Cambridge, MA, September 1991.

[32] Harper, R., CSDL Intralab Memorandum EJE-91-28. March 13, 1991.

[33] Randell, B., "System Structure for Software Fault Tolerance," IEEE Transactions on Software Engineering," Vol. SE-1, No. 2, pp. 220-232, 1985.

[34] Nelson, V.P. and W.D. Carroll, "Tutorial:Fault Tolerant Computing," IEEE Computer Society Press, 1987.

[35] Anderson, T. et al., "Software Fault Tolerance: An Evaluation," IEEE Trans. Software Engineering , Vol. SE-11, No. 12, pp 1502-1510, December 1985.

[36] Avizienis, A. and J. Kelly, "Fault-Tolerance by Design Diversity: Concepts and Experiments," IEEE Computer, pp 67-80, August 1984.

[37] Babcock, P.S., "An Introduction to Reliability Modeling of Fault-Tolerant Systems," The Charles Stark Draper Laboratory, Cambridge, MA, August 1986.

[38] Rubenstein, R.Y., Simulation and the Monte Carlo Method, J. Wiley, Inc., New YOrk, 1981.

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

[40] Vesely, W.E., et al., Fault Tree Handbook, NUREG-0492, Office of Nuclear Regulatory Research, US Nuclear Regulatory Commission, Washinton, D.C., January 1981.

[41] Howard, R.A., Dynamic Probabilitstic Systems, J. Wiley, Inc., New York, 1971.

[42] Harper, R. E., L. S. Alger, and J. H. Lala, "Advanced Information Processing System: Design and Validation Knowledgebase", NASA Contractor Report 187544, Langley Research Center, Hampton, VA, September 1991.

[43] Johnson, S and R. Butler, "Design for Validation Methodology," 10th DASC, Los Angeles, CA, October 1991.

[44] L. F. Burkhardt, T. K. Masotto, and J. H. Lala, "Advanced Information Processing System: Fault Injection Study and Results", NASA Contractor Report 189590, Langley Research Center, Hampton, VA, May 1992.

[45] Dorer, R.M. et al., "Safety of High Speed Magnetic Levitation Transportation Systems: German High Speed Maglev Train Safety Requirements-Potential for Application in the United States (Interim Report)," Report No. DOT-VNTSC-FRA-92-3, U.S.Dept. of Transportation, Federal Railroad Administration, February 1992.

[46] Lala, J.H. and R.E. Harper, "Fault Tolerance in Real-Time Embedded Systems: Importance and Treatment of Common Mode Failures," Workshop on Hardware and Software Architectures for Fault Tolerance: Perspectives and Towards a Synthesis, Le Mont Saint-Michel, France, June 1993.