ɔCUMENTATION PAGE

Form Aporoved
OMB No 0704-0188

mation is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, ompleting and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this r reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson 102, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

2. REPORT DATE
August 1993

3. REPORT TYPE AND DATES COVERED
THESIS/DISSERTATION

4. TITLE AND SUBTITLE
Reliability And Maintainability Of Modular Robot Systems: A Roadmap For Design

5. FUNDING NUMBERS

6. AUTHOR(S)
Capt Dean Leroy Schneider

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
AFIT Student Attending: The University of Texas at Austin

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/CI/CIA- 93-004D

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
DEPARTMENT OF THE AIR FORCE
AFIT/CI
2950 P STREET
WRIGHT-PATTERSON AFB OH 45433-7765

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for Public Release IAW 190-1
Distribution Unlimited
MICHAEL M. BRICKER, SMSgt, USAF
Chief Administration

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

14. SUBJECT TERMS

15. NUMBER OF PAGES
430

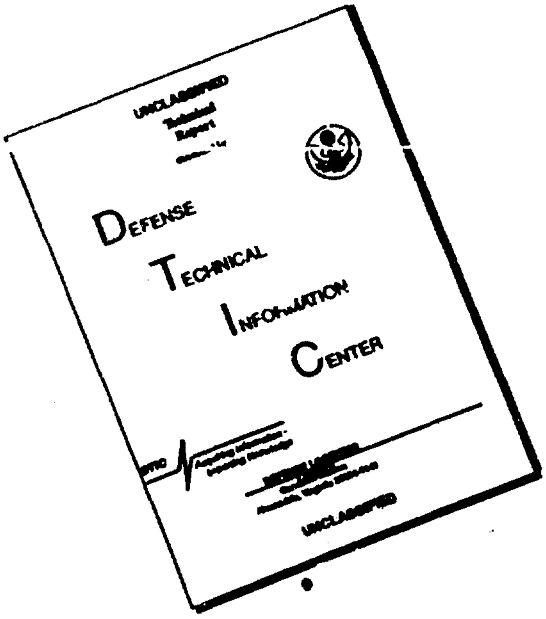
16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT

18. SECURITY CLASSIFICATION OF THIS PAGE

19. SECURITY CLASSIFICATION OF ABSTRACT

20. LIMITATION OF ABSTRACT



CHAPTER 1: INTRODUCTION

1.1. System Availability: Why?

When the National Aeronautics and Space Administration (NASA) began contemplating the design of the Space Station Freedom, it realized that the design requirements for the space station were completely different from the designs NASA had accomplished in the past. The substantial differences were: the space station is to be a multipurpose user facility with a lifetime greater than 20 years, it is to be launched in many different pieces and assembled in space and its design would evolve over time, maintenance and verification will occur in orbit, and perhaps most importantly of all, many of the space station functions will not be safety critical. These changing requirements have allowed NASA to propose the use of autonomous robotic systems to augment the astronaut's abilities in space. In fact, not only will space based robot systems reduce the Extra-Vehicular-Activity (EVA) requirements for the astronauts, NASA suggests it can save approximately $160 million over ten years by using robot systems enhanced by expert systems [47].

The realization of these savings is predicated on the robot systems being available for utilization by space station personnel. Thus, the robot systems must be designed for maximum availability in the space environment. Availability is the ratio between the uptime of a system to the total time of the system where the total time is the sum of the system uptime and system downtime [73]. Over the long term, availability can be expressed in terms of the system Mean Time Between Failure (MTBF) and the Mean Time To Repair (MTTR) as $\frac{MTBF}{MTBF+MTTR}$. The MTBF is generally accepted as a measure of reliability while the MTTR is accepted as a measure of maintainability. To maximize the availability, the designer must maximize the MTBF while minimizing the MTTR.

Maintenance experts generally agree that the best way to make a system easy to repair (thus minimizing the MTTR) is to make the system modular with standardized module interfaces. A prime example of modularity and standardization of architecture is the personal computer (PC). The progress in computer systems over the past two decades has been remarkable and is directly traced to increased reliability of the components (electronics) coupled with an architecture that allows rapid fault isolation and repair [64]. Hsiao traces the early experience International Business Machines (IBM) Corporation had with computer reliability and states that modularity of computer systems was a direct result of reliability improvement via the reduction of the number of required interconnections in integrated circuits. This experience lead to the choice of the modular standard architecture that has made today's PCs highly reliable and immensely flexible.

The same paradigm can be applied to robotic systems. Today's industrial robots are generally of monolithic design, requiring extensive times for trouble shooting and repair. One-off systems carry an even greater burden since the original builder of the system likely will not be able to service the system in the event of a failure (see Figure 1.1). Engelberger documents that the availability of an industrial robot must be greater than 95% for user satisfaction to exist. He notes the current acceptance of robots in the industrial environment and uses industrial reliability data to show this cutoff exists [46]. In an environment such as the space station, the availability of robotic systems must be much higher due to the high cost of operating in space. As noted above, one driver in the availability equation is MTTR. Modularity significantly reduces MTTR and would be a very appropriate choice for robotic architectures in space, especially if robots are planned to repair robots.

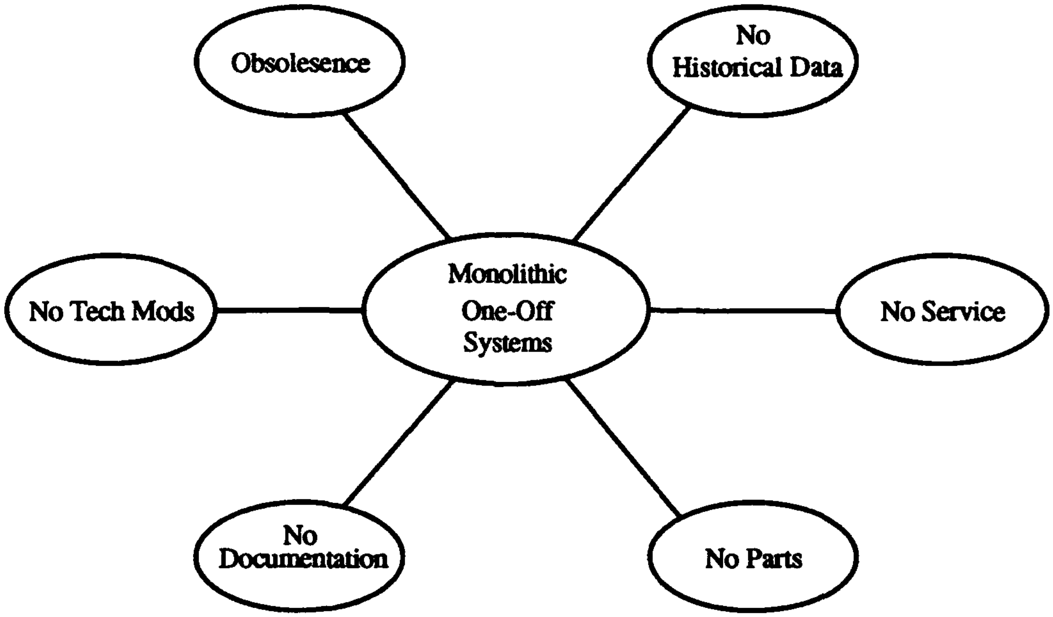


Figure 1.1. Drawbacks of the Monolithic One-Off System

The second part of the availability equation is the system reliability. If the choice of architecture minimizes the repair time, the availability design question reduces to one of maximizing reliability. The study of system reliability dates back to the Second World War where electronics were first being applied to the art of war. At the same time, the United States was developing the first nuclear weapons which further drove development of reliability theory. In the intervening years, reliability theory has matured and techniques have become standardized. Again, an example of this is the success of the PC. In addition to modularity, another reason for the PC's success is the extremely high reliability. This high reliability is a result of technological developments at both the electronic component level and in the system architecture. It was realized early on that for a system to be more reliable than its components, the components must be combined in a way such that the system architecture would minimize the effect of failures at the component and interface level [163]. This realization is generally regarded as the birth of fault-tolerance. Additional advantages of the modular system include standardized support structures, readily available supply of spare parts, etc. (see Figure 1.2). Again, the same argument can be made for robotic systems. To maximize the reliability, the design of a robot system requires the best quality components as well as architecture, both computational and mechanical, that will allow the operation of the system after failure.

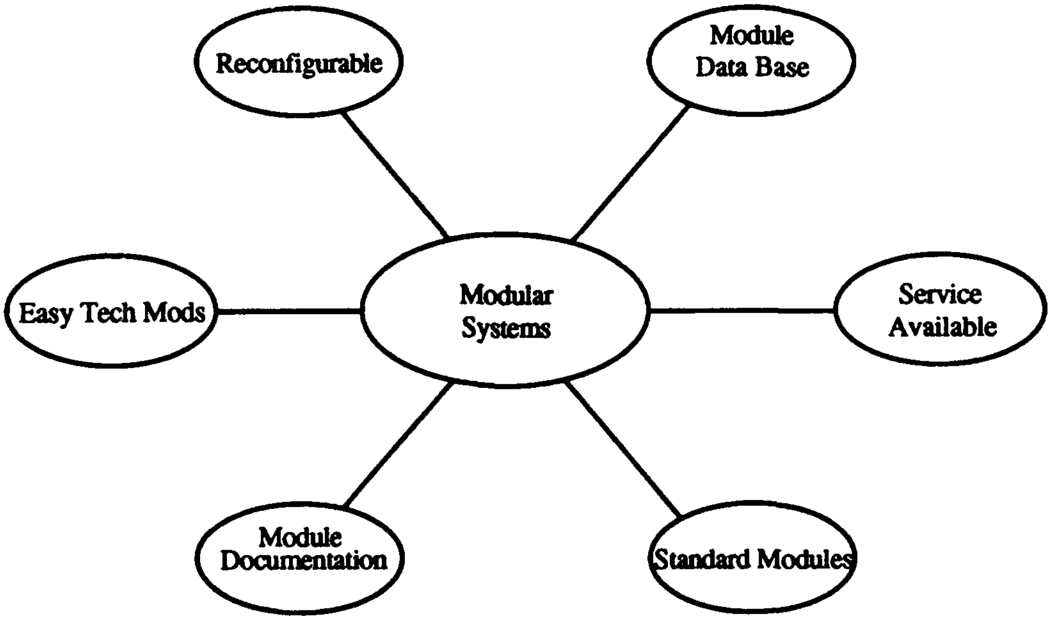


Figure 1.2. Advantages of a Modular System

The flexibility of a modular architecture is worthy of additional discussion. Currently, most successful industrial robot systems are serial monolithic structures. These machines were usually purchased to fulfill a single task or a group of similar tasks. Let's say a substantial change in a process occurs where now instead of spot welding, the manufacturer desires to use robots for part selection and positioning in addition to spot welding. The manufacturer has already invested a large amount of money and time into the acquisition and integration of the welding robots which he now finds to be inadequate for part positioning due to precision positioning requirements. If the original system is modular, the manufacturer could perhaps just add a new set of modules, both hardware and software, to upgrade the system capabilities thus retaining the investment already made in the welding robot. The modular architecture allows this possibility while a system of monolithic design can suffer from obsolescence and the costs associated with it. An additional comparison of modular vs. monolithic system characteristics is shown in Table 1.1.

Returning to the discussion of availability, one can now see that high reliability coupled with good maintainability, i.e. the minimization of repair time, allows for the maximum availability of a system. The modularity of robot architectures, both in the controller, controller software, and mechanical architecture, will allow for fast trouble-shooting and module replacement, minimizing the repair time. The remaining problem is to design the system for high reliability and performance.

1.2. Robot Environments and Reliability Requirements

As alluded to in the previous section, the system reliability is directly related to the environment the system will operate in. Take for instance, a generic power supply system in varying operating environments. If the power supply is installed in an aircraft, the MTBF of the system can vary anywhere between 2,000 and 20,000 hours depending on the usage requirement and flight regime. A similar power supply in a ground maintenance shop can expect to realize a 10,000 to 50,000 hour MTBF, almost an order of magnitude in difference in the MTBF values [130]. Similar situations can be observed in any system examined.

Table 1.1. Modular vs. Monolithic System Characteristics

Monolithic Systems	Modular Systems
Few Interfaces	Standardized Interfaces
Custom Architectures (Task Determined)	Generalized Architectures (User Determined)
Large Amount of Function Sharing	Module are Functionally Independent
Task Specific	Reconfigurable for Different Taskings
Large Repair Times	Many Module Interfaces
Can be Difficult to Troubleshoot due to Function Sharing (Diagnostic Complexity)	More complexity is forced to lower architectural levels (Module Complexity)
Difficult to Use Generic Part Substitutions	Reduced Threat of Obsolescence
Inflexible to Change (Environment, Task, Control)	Task Dependent Structure

The most severe environment is generally regarded to be space. This is reflected in the classes of standard parts quality available for use in designs. The most stringent acceptance requirements are seen in space certified (class S) components. This class of parts represents a 60X increase in component MTBF values over commercial components [130].

For a robotic system to be space certified, it must meet and demonstrate a quality level (which is in part quantified by the reliability) commensurate with space certified components. Again, the problem remains: how to design a robot system for high reliability and performance.

1.3. Modular Robot System Concepts

The concept of modular robots has been around since the mid 1970s when Westinghouse specified a robot for which parts where no heavier than 35 pounds and could fit through a 16 inch manway for maintenance in the steam generator of a pressurized water reactor. Since that time, several research teams have spent a great deal of time in the development of modular robot systems and their associated standards and design problems. Notable among these efforts are the University of Toronto, Carnegie Mellon University, and The University of Texas.

The Toronto effort is best summarized in two papers by Benhabib [13, 14] in which a mechanical design of one Degree-Of-Freedom (DOF) rotary and prismatic joints with a variety of links and adapters can be used to assemble an arbitrary n-DOF configuration. They have shown designs for both a PUMA type articulated robot and a SCARA robot using a proposed set of mechanical modules. They do not address the configuration optimization or system integration necessary for a complete modular robot system.

The Carnegie Mellon modular robot is described by Schmitz [137, 138]. The CMU Reconfigurable Modular Manipulator System (RMMS) has been under development since the mid-1980's. The system consists of two common revolute joints (pivot and rotate) and an array of link modules with standardized interfaces between the modules and automatic sequencing and orientation determination. The researchers at CMU have also developed a software architecture to support the modular system consisting of the feedback control law, path planner, data logger, and an interactive command interpreter. Additionally, methods for selecting the kinematic configuration [119] and actuator models [80] were developed. Paredis and Khosla [119] also describe an iterative design procedure for overall integration of the modular system (Figure 1.3) but do not fully describe the purpose of all the functions enumerated.

Both of the above designs are for the mechanical design of the robot itself. The CMU concept recognizes the need for overall system integration but only addresses component selection. Not mentioned in the literature are the implementation aspects of the modular robot concept. What is missing is a strategy for overall integration of the robot system, optimized to a particular task or other criteria. The University of Texas Robotics Group has conceptualized the complete integration and operation of a modular robot system, from mechanical design and configuration selection, controller selection and implementation, software module selection, through final system integration and operation.

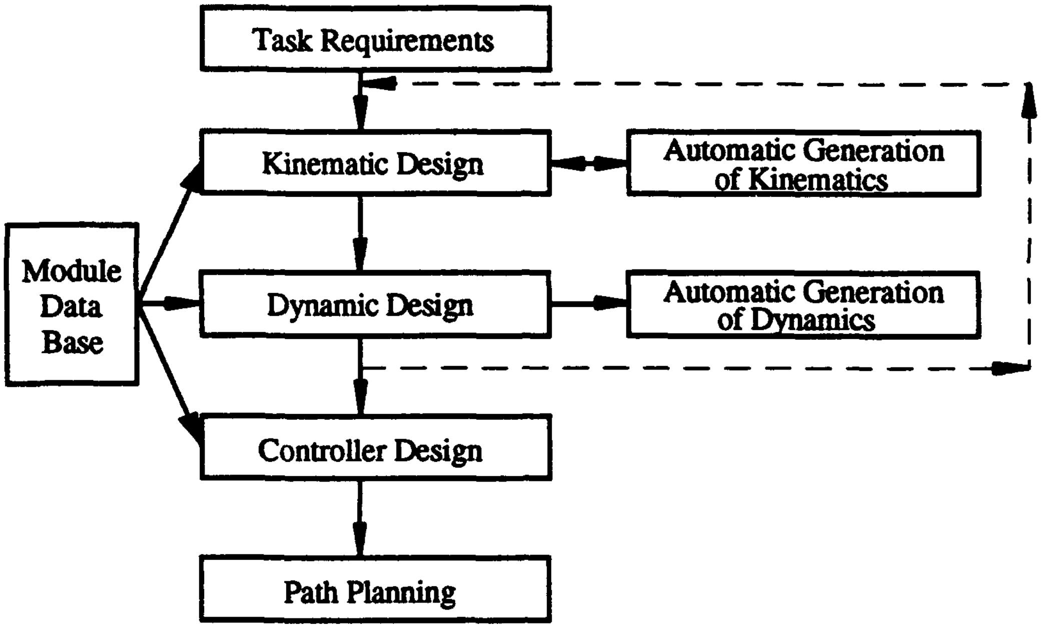


Figure 1.3. Schematic of CMU Iterative Design Procedure [119]

Figure 1.4 is an overall block diagram of a system called the Interactive Synthesis Tool for Advanced Robotics, (ISTAR). The system assumes a finite set of scalable joint and link modules that have already been designed and optimized to a set of operational and performance criteria. The customer would have a (perhaps) loosely defined task with certain performance requirements. The user would input his requirements into the task editor which would then perform a specification transfer to a formal specification language. Based upon the specifications and mechanical modules available to the user, the M_CAD expert system would generate an optimal configuration for the task and generate the physical plant description [63]. This optimality would be based on a set of 30 or more geometric, energy, and load performance criteria as well as operational criteria, such as reliability (see Table 1.2). The S_CAD expert system would then select the appropriate software architectures and low-level interfaces. C_CAD would then select and optimize the controller hardware for proper performance of the task. On-going throughout the process is the interaction with a lessons-learned data base and an overall integration optimization algorithm based on performance and operational characteristics. The output of this integration procedure is a list of modules for the user to assemble in a certain order, the controller hardware needed and how it should be configured, as well as the controller software, ready to run. The entire concept is intended to be a turn-key ope ation for the user [128].

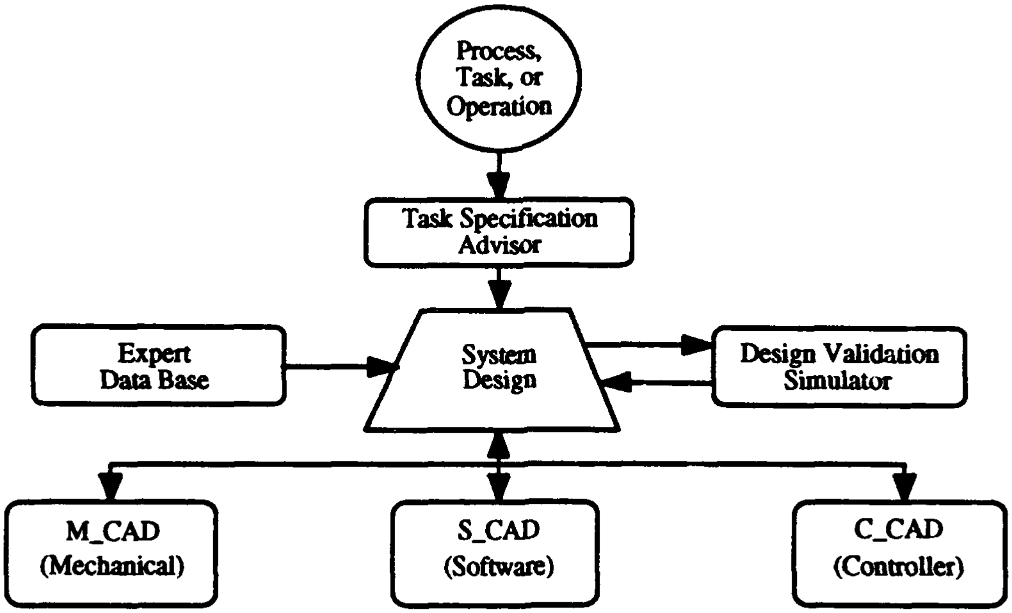


Figure 1.4. ISTAR Expert System Conceptual Diagram [128]

1.4. Contributions of This Research Effort

This research effort represents a unique, systematic, examination of the reliabilities of the underlying component technologies that make up robotic systems.

Since robotics is a multi-disciplinary technology, a review of reliability techniques applied successfully in the various disciplines making up the technology (electronics, mechanisms, software), is made and correlations to the design of robotic systems are made. The unique contribution is the development and presentation of a design guide for reliability during robot system design developed from the "lessons learned" of the constituent disciplines.

Also unique to the reliability field, is the development of a composi'e reliability index, named the Reliability Performance Index (RPI) which allows for the quantification of both a modular robotic system's hardware and software reliability, as well as the performance characteristics of accuracy and repeatability. While shown to be useful in the configuration design of a modular robotic system from a pre-determined suite of modules, it also shows promise as a tool for the quantification of the overall system reliability and performance for any system that has a measured error, such as feedback control systems.

1.5. Problem Statement and Objectives

The reliability of modular robot systems and the technologies that support reliability improvement are varied and multi-disciplinary. Robotics researchers nationwide are trying to come to grips with new technologies and how they may improve robot systems. To make reliability improvements in modular robot systems, designers must be aware of the different technologies and how to use them effectively. The modular robot concept described in the previous section would present the designer with a "shopping list" of available options and technologies he or she could include in the design of their manipulator. Some technologies are more useful for performance characteristics while others effect the operational characteristics of the robot. Part of the integration of the overall system is the inclusion of the proper technologies to meet the specifications described by the user. The expert system will need to choose the proper technologies to use and the priority with which they are to be implemented. This knowledge does not exist. One of the contributing technological aspects requiring consideration during integration is the reliability of the system. In regard to this consideration, the first primary goal of this research is to provide a Reliability and Maintainability (R&M) technology roadmap for modular robot systems. Not only will such a roadmap provide a large portion of the integration expert system's data base, it will also provide a guide for the robotics researcher in reliability and maintainability. This roadmap will provide a prioritized list of technologies that require research emphasis for modular robot R&M and performance. This roadmap consists of literature reviews and technology prioritizations.

Once the technologies are understood, the robot must be designed and integrated. The design of robots can be judged by criteria based on many possible sources. Criteria are especially useful when decision-making (such as that needed for intelligent control) is required. Table 1.2 is a listing of possible criteria that can be used for decision-making for the mechanical manipulator system. A similar list is possible for the controlling software itself.

Most of the criteria identified in Table 1.2 are useful for module design, system configuration and trajectory or task generation, individually. There has not been a criterion that has been explicitly developed for the overall integration and operation of a modular system as described in the previous section. This top level criterion requires both performance and operational criteria embedded in it. The operational criteria at the system level include the Reliability and Maintainability (R&M) of the system. Note these are the components of the availability equation presented earlier. If one assumes that the modularity of the system will allow minimization of the repair time, the remaining operational criterion is the system reliability. A need exists to provide for the application of reliability principles at the modular level and to quantify the module reliability in a systematic way. Many of the other performance related measures are subsumed in the configuration of the modules to satisfy the specifications such as load carrying capacity and workspace requirements. Not explicit are the performance characteristics of accuracy and repeatability, both of which are mechanical in nature. As a practical matter, criteria should be as simple as possible while providing as much relevance as possible. One possible combination is to use the hardware and software reliability along with a probabilistic definition of error on the end effector to develop a system level reliability performance index which can be used in the objective function for overall system integration and optimization.

One good method for the probabilistic description of end-effector error was suggested by Bhatti and Rao [18]. By defining the reliability of the robot system as the probability of the end effector being within a certain error in both position and orientation, all of the underlying characteristics of the system, both traditional hardware and software reliability, and the kinematics and dynamics of the system are included. This also implies a new definition for failure of a robot system, i.e. the end effector being outside a certain error bound in position and orientation. Once these definitions are applied to robot systems, reliability theory can be applied to the system at the module and system level to provide a normalized design and optimization criteria.

Table 1.2. Criteria for Redundant Manipulators [35]

Singularity Avoidance	Manipulability Measures
Totian (High Output Velocities)	Obstacle Avoidance
Potential Fields (Various Sources)	Dexterity Measures
Joint Range Availability	Matrix Condition Number
Minimum Singular Value	Manipulator Velocity Ratio Norms
Energy Minimization	Minimization of Kinetic Energy
Minimization of Actuator Torques	Torque Distribution Criteria
Manipulator Mechanical Advantage	Manipulator Precision
Fundamental Natural Frequency	Generalized End-Effector Spring
Load Carrying Capacity	Speed of Operation
Dynamic Response	Sampling Rate
Computational Effort	Reliability Measures
Maximum Usable Workspace	Kinetic Energy with Vibrations
Distribution of Kinetic Energy	Command Shock Issues

The objectives of this research are to examine R&M technology for modular robot systems and develop a top-level design and optimization criterion:

1. Provide a review of current R&M technologies for robotics and modular systems in general.

2. Provide an overview of how reliability design methods and tools can be applied to robot systems.

3. Rank the technologies applicable to improved R&M in modular robot systems and make suggestions for further research into technologies with potential for high R&M returns in modular robot systems.

4. Develop the mathematical framework for a Reliability Performance Index (RPI) including the hardware and software reliability of a modular system as well as the performance characteristics of accuracy and repeatability to enable the quantification of both modular robotic reliability and performance in a design criterion.

5. Validate the Reliability Performance Index through sensitivity and optimization studies and determine the usefulness of the RPI during design synthesis.

6. Demonstrate the use of the RPI through a case study for the configuration design of a modular robot manipulator.

7. Provide usage guides for both the R&M Technology Roadmap and the Reliability Performance Index.

1.6. Outline

The remainder of this dissertation is outlined as follows. Chapter 2 presents the literature review in both current robot R&M and modular systems R&M. Chapter 3 is a description of reliability application during design. It presents the various tools and methodologies developed over the years and provides a summary of reliability design principles and where to apply them during robot system design. Chapter 4 contains descriptions of the component technologies that affect the R&M of robot systems. Reliability analyses are carried out on generic robot configurations to assess the impact of the component technologies on the overall system reliability. These technologies are then ranked according to the application and payback to guide future research in the R&M of modular robot systems.

To provide for the application of reliability principles to modular systems, the reliability of each module must be quantified. Chapter 5 presents the conceptual development for the framework of the Reliability Performance Index (RPI) to allow achievement of this goal. Following this development, Chapter 6 presents a case study in the configuration design of a modular robot system using the RPI and considers sensitivity and optimization studies of the RPI. Statistical testing for the significance of the differing effects on the RPI are also found in this chapter as well as research recommendations for further validation and refinement of the RPI. The overall framework for the R&M Roadmap for Modular Robotic Systems is found in Chapter 7. This chapter presents the conclusions and discussions of the results of this work along with a list of recommendations for further study.

This dissertation contains three appendices. Appendix A is an overview of reliability theory with explanations of the specific methodologies to be used when applying the various reliability analysis tools. A knowledge of basic statistical analysis is assumed. Appendix B is a set of design checklists for the R&M of modular robot systems. This guide includes general design principles for modular design and suggested technologies that when implemented, will increase the R&M of a modular robot system. Appendix C documents the extensive search for reliability data from industry and supports the major finding of this research regarding the need for robotic reliability data.

CHAPTER 2: LITERATURE REVIEW

2.1. Introduction

2.1.1. The Purpose of the Literature Review

This literature review has two explicit purposes. The first is to present the past efforts in the reliability analysis of robotic systems, list the methods used and the results obtained, and show whether or not the work described was strictly analysis or if it can also be applied to the design of robotic systems. The second is to identify design guidelines or principles that can be applied to modular systems in the hopes of increasing the reliability of the modular system. These design principles can then form the basis of a modular robotic system design paradigm including the various reliability tools and techniques developed in other domains as well as the technology aspects of reliability improvement. This design paradigm is represented in the Roadmap for Design and is presented in Chapter 7 as an overall design approach (using both programmatic and technology thrusts) to improve modular robotic system reliability.

An additional point made while discussing each work is the presentation (or lack thereof) of the actual data used during the research. The basis for any conclusion must be data and to make conclusions about the reliability of a technology, such as robotic systems, reliability data must be available for examination. As can be seen from the description of each work and from Table 2.6, not many researchers actually published their reliability data. An extensive efforts was also made to gather any data from industry to adequately represent the state of the art (See Appendix C). This effort produced no corporate organization willing to divulge reliability data on their systems or on their components. This result severely handicapped this research in making a in-depth review of the state of reliability of the robotics industry. This fact resulted in the first recommendation of the reliability roadmap being the establishment of a robotics industry reliability data base to allow researchers the ability to obtain the global view of the industry and allow industry-wide suggestions to improve their product.

2.1.2. The Importance of Robot Reliability

Reliability: The probability that, when operating under stated conditions, the system will perform its intended function adequately for a specified period of time [73]. One simply has to review the body of literature devoted to the design of robotic systems to appreciate the importance designers give to reliability. Most of the time, however, their treatment of reliability during design is generally lip service if mentioned at all. While exceptions can be found to this statement, researchers usually approach the problem from an analysis point of view. Not much, if any, work has been done to insure the satisfaction of reliability design constraints during the design of a robotic system. This is due to the immensity of the robot system design problem.

To demonstrate the importance designers place on the reliability of robotic systems, it is appropriate to go back to the early history of industrial robotics and examine what is perhaps the first reliability investigation of industrial robots performed at Unimation, Inc. and reported by Engelberger [45]. Engelberger's purpose in writing his paper was to convince industry that robot systems were reliable enough to be utilized in a typical factory environment. Through a complete combinatorial analysis (see Appendix A), Unimation concluded their Unimate robot had a design MTBF of 500 hours. Their experience showed a typical profile of product reliability growth, starting out with a MTBF of 145 hours and increasing over three years to 415 hours, achieving their goal of 400 hours and nearly achieving the theoretical design MTBF. This reliability growth did not occur through design changes, but by implementing a reliability/quality control system during the manufacture and operation of the robots. By examining a typical Unimate installation, they concluded that they met their availability goal of 97%. Engelberger's motive was to convince industry to invest in robot systems by demonstrating the achieved availability and cost effectiveness of his company's robots. He did this after Unimation robots had accumulated over 3 million hours of operation. He makes no mention of reliability during design but he does seem to imply concern for reliability during the early production years.

A note to the reader must be made at this point. Life testing and data analysis are extremely important aspects of the reliability analysis of a system. It provides the most accurate and in-depth review of the system's durability. However, the system must have already been designed and built for any life testing to occur. This can help identify design deficiencies and allow retrofits, but by far the highest payoff in system reliability is the application of good reliability design principles during the design of a system to maximize the reliability and minimize the repair time. The so-called "rule of tens" can be applied here. For each $1.00 flaw not found in the factory or during the design phase, as the level of assembly grows:

components, to assemblies, to unit, to the system, the cost of finding the flaw increases by an order of magnitude of at each assembly level. Reliability methods applied during design to find these design flaws pay off immediately.

Other measures of the importance of reliability during design are contained in several research reports performed at the University of Texas. The first, by Butler and Tesar, lists fourteen design criteria that are used during the design of robot systems, of which, reliability is one [22]. The analysis performed in this report ranks these design criteria in order of their importance in 6 different and diverse application areas from industrial automation to human augmentation to planetary surface operations. Overall, reliability emerges as the most important design criteria for robot systems in five out of the six categories and comes in second only to precision in the sixth (precision assembly tasks).

The second report, by Cox and Tesar, examines technologies and software decision-making criteria as they affect dual-arm operations [35]. It suggests that ultra reliable robots would achieve 150,000 hours MTBF via fault-tolerant design. Again, reliability and diagnosis ranked 9 out of 10 in the technology needs for dual-arm operations overall and for the long term. This report recommends making robot reliability, diagnosis, and safety very high research priorities.

The third University of Texas report by McAndrew and Tesar assesses microelectronic assembly systems and states prioritized rankings for performance in assembly tasks [89]. Out of ten performance requirements, reliability was third, ranking 9.7 out of 10 behind accuracy and resolution. It is interesting to note here that reliability includes both failure free operation and confidence in positioning and accuracy. Both of these components of "performance reliability" are included in the formulation of the Reliability Performance Index for modular manipulator systems presented in Chapter 5.

Other mentions of robot reliability are made and will be discussed next in Section 2.2. As stated before, most concentrate on analysis and not directly on reliability during the design process.

2.2. Robot Reliability

This section will discuss past experiences noted from the literature on the analysis of robot reliability and the methods involved. For a more in-depth description of reliability theory and the mechanics of the methods described, see Appendix A and its references.

2.2.1. Robot Hardware Reliability Models.

The reliability of robot systems has been analyzed in many different ways, using all the popular techniques of reliability theory. Not only hardware failures but also performance aspects have been addressed (see Section 2.2.3). This section addresses only the hardware aspects of robot reliability. Those articles dealing with reliability during the design of robot systems are highlighted.

2.2.1.1. Robot Reliability Data and Life Testing

An important view of the reliability of existing systems is obtained through life testing and the gathering of reliability data throughout the production and use of the system. See Appendix A for a discussion of data analysis and life testing. As mentioned in Section 2.1.1, Unimation Inc. collected data on its Unimate robots for over three years which showed an increase in MTBF from 145 hours to 415 hours [45]. Sugimoto and Kawaguchi also report the results of life testing and operational life and failure data (see Section 2.2.1.2 for their results). Jones and Dawson reported an average of 43 hours mean time to robot related problem from data collected from 37 robots over 21,932 robot hours [72]. The robot systems averaged 75% availability, which is far less than the 97% Engelberger suggested as the minimum availability required to satisfy the users of robot systems. A comparison is hard to make since Jones and Dawson do not specify the types and makes of the robots examined in their study.

Grundmann reports on the frequency-of-failure based reliability testing of an Automated Storage And Retrieval System (ASARS) which handles vials of plutonium samples for chemical analysis in a glove-box environment at the Rocky Flats Plant in Colorado [61]. The system consists of a storage carousel, pneumatic transfer tubes, a scale and bar code reader, and a stepper motor driven 2 Degree-of-Freedom (DOF) transport robot. The robot went through 12,500 cycles of its program without failure during reliability tests. This resulted in a reliability estimate of 0.99992. However, while having a very high reliability, the transport robot had by far the longest repair time of the system, which caused the robot to dominate the downtime of the system causing the authors to declare the robot to be the "weak link" of the entire ASARS. This example illustrates why availability is a better measure of system durability than reliability alone. Grundmann's study also provides insight into the immediate gains that can be made in availability by modularization.

A modular robotic system in this application would alleviate the long repair time seen by the transport robot as well as the report of the robotic "weak link."

A more recent addition to the field is a book by Dhillon named *Robot Reliability and Safety* [42]. Dhillon provides typical failure rates for both mechanical and electrical robot components based on a constant failure rate assumption. He states that the failure rates are obtained from MIL-HDBK-217 and data from the Rome Air Development Center on Non-Electronic Parts Reliability. Dhillon suggests industrial robot system design goals of 400 hours MTBF and 8 hours MTTR which are far less than any current manufacturer admit to. He also gives a great deal of information regarding the collection of reliability data on robot systems but has no actual history of any robot system and as such is of very little use when dealing with practical considerations. Dhillon's book also suggests the best methods for reliability analysis of robot systems are Fault Tree Analysis and Reliability Block Diagrams.

2.2.1.2. Fault-Tree Analysis and Failure Mode and Effects Analysis

Fault Tree Analysis (FTA) and Failure Modes and Effects Analysis (FMEA) [68] are the methods most used for robot hardware reliability analysis. The reasons for this are ready availability of evaluation software and ease of use of the methods. FTA is a top-down methodology where a top level failure event is defined and the design is investigated to isolate the possible causes of that failure. FMEA is a complementary approach to FTA starting at a bottom level failure mode and investigating the propagation of the failure throughout the design. Both of these techniques are described in detail in Appendix A.

Serious interest in robot reliability analysis began in the early 1980s as an adjunct to concerns for safety. Robot safety became a very important area after reports of human injury caused by robot systems were reported in the late 1970s. One survey in Sweden reported a rate of one accident per 45 robots per year [152]. While not seemingly excessive, any injury to a human by a robot is cause for concern and several researchers began investigations into the causes of robot failures. Sugimoto and Kawaguchi developed a fault-tree analysis of hazards created by robots with the top level event beiιg a robot accident caused by five different categories of energy: potential (falling objects), kinetic (movement, flying objects), chemical/biological, thermal, and radiant [152]. They did not perform any probabilistic analysis of their fault-tree, only suggesting causes of accidents and ways to prevent them. One of the conclusions they reached is that adequate robot reliability has not been assured as of 1983. Quoting data from a Japanese study, they found 28.7% of the robots have under 100 hours MTBF and 75% under 1000 hours MTBF. Sixty-seven percent of the failures were attributed to the control systems (microcomputers and electronics), 23% failures in the robot body (probably actuators and sensors), and 20% in programming. Their main conclusion was that the development of intelligent machines which can detect the presence of humans in the workspace and respond to avoid accidents is the only way to assure complete safety in the robotic environment. This work represents one of the only reports of actual reliability data available in the literature and is the most referenced by other authors. However, this data was generated in the late 1970s and very early 1980s and represents extremely dated technologies. That it continues to be so highly regarded (Dhillon places a large emphasis on it [42]) is a disappointment.

Another group of researchers in Britain was interested in this same topic :or the same reason. Khodabandehloo, Duggan, and Husband formulated a FTA model of a five joint hydraulic robot and a generic electric robot considering safety through reliability (the fault trees are reprɔduced in Appendix A, Section A.2.1.1) [76, 77]. They assumed random sources of failure and developed probabilities for each tree event using MIL-HDBK-217 failure rate data for the electric model. Only the model was developed, the overall reliability was not quantified. Their work extends that of Sugimoto and Kawaguchi in that they recognize the importance of design improvement by reliability analysis. Their experience suggests that FTA and FMEA be performed as early as possible in the design of robot systems. Khodabandehloo also performed a FTA and Event Tree Analysis (ETA) for a safety analysis for a PUMA 560 robot [78]. In this paper, they showed that high reliability does not necessarily mean good safety (see also [110]). The results of the reliability analysis for the robot controller (using standard reliability prediction methods in MIL-HDBK-217) was 71% probability of failure at 10°C to 90% probability of failure at 50°C per 1000 hours. This does not include the robot itself. Their study resulted in a main design suggestion to include a monitoring system that would sense robot error and correct for it (an adaptive controller or fault-tolerance scheme). The addition of this system would reduce the probability of failure at 3C°C to 4.7% per 1000 hours, an extraordinary improvement. These results are similar to Sugimoto and Kawaguchi where the main source of robot failure was the controller. Again, these results were obtained using technology that is almost a decade old.

As mentioned in the Section 2.1, the most important point at which to consider reliability is during the design, since changes can be made most easily during the design phase. This philosophy was demonstrated by Gordon and Curry and reiterated by Weaver and Deininger through FTA/FMEA analysis performed on the RRV-1 robot designed and built by Carnegie-Mellon University for the Three Mile Island Reactor Building [59, 164]. This effort resulted in the development of a reliability improvement strategy for future robot developments. The study recommended three concepts to improve robotic system reliability during design: (1) redundancy of components to improve reliability, (2) diversity of the system to reduce the possibility of common cause failures, and (3) component assurance by using high quality components. These recommendations can be applied to the reliability improvement of any system regardless of configuration and domain. They also observed that the failure rate for the system is not constant and tends to increase with time.

Another example of FTA applied to a robot system is a study by Wells and Krishnaswami [165]. They use FTA to generate probabilities of failure of a remotely operated deep-sea vehicle with a manipulator attached. Their analysis, based on admittedly sparse reliability data, suggested that robot position limit switches had the highest failure rate, followed closely by motor failures. The researchers arrive at the same recommendations as Gordon and Curry of improved component reliability, the redundancy of critical components, and the elimination of single point failures.

2.2.1.3. Reliability Block Diagram Models

Another popular method of reliability analysis is the Reliability Block Diagram (RBD) method [73]. This method is particularly amenable to modular systems since it consists of combinations of "black-boxes" to generate the overall reliability structure of a system. Each component is represented by a block in the block diagram which is generated along functional lines (see Figure 2.1). The reliability study performed by Unimation, Inc., [45] discussed in Section 2.2.1.1 used a RBD model to compute the reliability of PUMA robots. Dhillon [42] also conducted a RBD analysis on the electric and hydraulic robots described by Khodabandehloo, Duggan, and Husband which were presented in Section 2.2.1.2 and Appendix A [76, 77]. The Grundmann ASARS reliability analysis described earlier in Section 2.2.1.1 was also based on a reliability block diagram model for the system [61].

An important point to make at this juncture is that the numerical results for each method, RBD or FTA, are the same. The difference is in the description of the model. The RBD approach models the system at the modular level while FTA models effects. A fault tree can be generated from a reliability block diagram and vice versa. It has been shown that a RBD is equivalent to a fault tree when the components have statistically independent lives [42]. Consider the RBD of Figure 2.1. Using the formulation shown in Appendix A, the reliability function of the system represented by Figure 2.1 can be written as

$$R_S(t) = R_1(t)\{1-[1-R_2(t)R_5(t)][1-R_3(t)][1-R_4(t)]\} \tag{2.1}$$

where $R_i(t)$ represents the component reliability functions. Using the methodology presented in Appendix A, an equivalent fault tree can be generated with the top level event of system failure and the low level events are failure of the components. This fault tree is shown in Figure 2.2

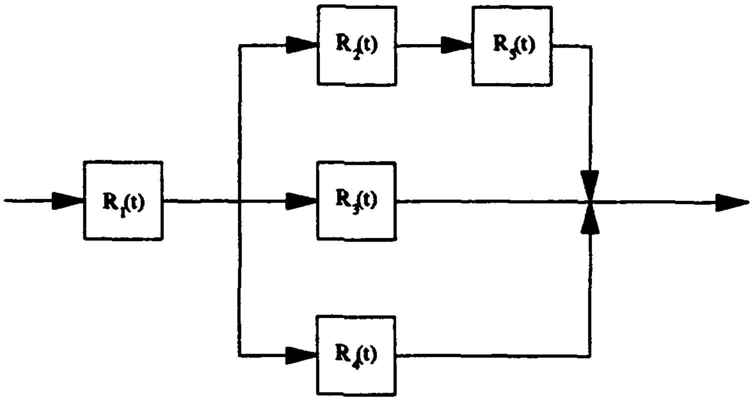


Figure 2.1. A Simple Reliability Block Diagram.

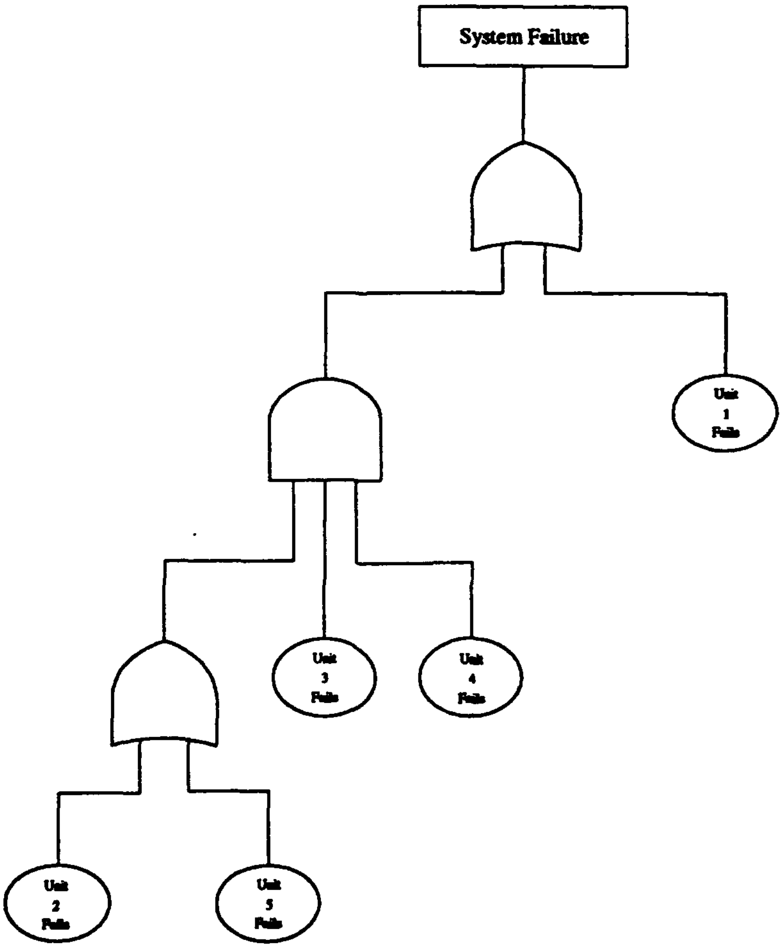


Figure 2.2. Equivalent Fault Tree Representation of the System of Figure 2.1.

2.2.1.4. Markov Reliability Models [133]

While FTA/FMEA and RBD models are useful in describing the robot system's reliability and failure characteristics, they are not amenable to a description of the repair and availability of the systems (the operational characteristics of the system). To be able to describe these characteristics, a mechanism to include repair rates into the model must be used. Traditionally, reliability practitioners have resorted to a Markovian model to describe this behavior. The difference between the combinatorial models and the Markovian model is in the description of the system. FTA and RBD are centered about the component of the system itself and the behavior of those components. The Markovian model is concerned with the operating states of the system and the transitions between those states. The Markovian model is especially useful in describing fault-tolerant systems where one wishes to model the degradation of a system not a strict operational or failed condition [133]. The states of the system can be defined by the number of components operational with other variations, such as component degradation, included as well. The fundamental assumption of the Markov model is that the transition to a new state is dependent only upon the current state of the system and not on the past history of the system. This is also referred to as the "memoryless property" of the Markov model (See Appendix A, Section A.2.1.2). This assumption is fairly restrictive since it requires constant transition rates between states. This also means the life distributions of the components, as well as the repair distributions must be exponential implying a constant failure or repair rate. While generally true for electronic systems, mechanical systems and software generally do not have constant failure rates due to wearout and the correction of software errors, respectively. This problem can be avoided by the use of a Semi-Markov model which allows general transition distributions. Once the model has been solved, the reliability of a system can easily be found by summing the time the system spends in operational states. Availability of the system is determined by the average proportion of time the system spends in operational states. Appendix A contains the mathematical derivations for this model.

Markov type models are generally the most flexible reliability models that can be generated. The drawback of Markov and Semi-Markov models is the number of states required for the model. For instance, for a system with n components each with only an operational and failed state, the Markov model requires 2^n states to model the system. For a system with many components, this quickly becomes very computationally expensive. Consider the system of Figure 2.1. With two states per component the number of states in the model would be $2^5 = 32$. If we wish to model degradation, we can add more conditions to each component, say fully operational, 75% operational, 50% operational, and failed. In this case the model would require $4^5 = 1024$ states. The model is very flexible, but carries a large price in terms of computational complexity since each state represents a state variable in a system of linear differential equations that must be solved to generate the system reliability (see section A.2.1.2).

The first use of a Markov model to describe the reliability of a robot work cell is provided by Cohen and Chandra [31]. In their example, the authors include two processing machines, two robots used for loading and unloading the machines, and an interstage buffer between the machines which allows the machines to operate independently. The authors assume that machine failures occur very infrequently and just model machine failures as wear of the tool in stages from no wear to tool failure. They define robot failure as a degradation in repeatability. This degradation is described by the interference between a part being inserted into a jig. It is measured by insertion force at the end effector. The model describes each machine and robot as a separate Markov chain and the reliability of each component (machine or robot) is evaluated from the model of the machine. If no interstage buffer is included, the system becomes serial, with the failure of any component causing a system failure. With the buffer, a queue is added to the system which slightly alters the reliability since the process can still complete work if a failure occurs after processing the part on a failed machine. The simplicity of the workcell model is beguiling since only 16 states are required for the model. However, the addition of just one more machine doubles the number of states to 32 and the state space can get extremely large very quickly especially when considering the components of the robots themselves. Another problem is with the Markovian property itself, as described earlier. Only the model is shown, the methods needed to obtain the failure rates are not addressed.

Dhillon also uses a Markov model to describe a robot system including human error. His three state model consists of an operational state, a state for the robot system failed due to human error, and a state for the robot system failed by other than human error [42]. Dhillon generates the model assuming constant failure and repair rates and develops expressions for the operation of the system as well as for its reliability. He also provides several sources for human reliability data that can be used in this model. While illustrating the principle of the Markov model, he does not show how the modeling technique can be applied to the robot system itself to provide improvement during design nor doe he provide any pertinent data or numerical examples of the model's solution.

2.2.1.5. Simulation Models

Analytical models are usually adequate for small or simple systems, however, as they increase in size and complexity, these analytical techniques become extremely difficult to apply and resolve. In these instances, simulation becomes necessary. Simulation is described as the use of a model for experimentation to describe the behavior of systems, construct hypotheses to account for the behavior, and to predict future behavior of the system [121]. Any of the models described in the previous sections can be used for simulation purposes. There are several classifications of simulation models: deterministic or stochastic, static or dynamic, and continuous or discrete. Deterministic simulation models ignore the randomness in the world while stochastic simulation models acknowledge it. Static models only represent the system at a particular instant of time while dynamic models describe the system's behavior through time. Discrete models examine changes in the process or model at discrete time intervals while continuous models treat continuously changing phenomena.

Simulation models for reliability must be stochastic in nature since reliability and availability are measures of probability. Such stochastic simulations are termed Monte Carlo simulations. Monte Carlo simulation is widely used to determine the reliability of very complex systems where analytic techniques become difficult to apply or where the analyst doesn't need an analytical form of the reliability function.

An underlying distribution is assumed (or chosen on past experience) and a sample is drawn and combined with other variates based on any of the above models to provide a sample response for the system (see Appendix A, Section A.1.2.5). As an example, assume that a system response function having two independent random variables, **X** and **Y**, can be written as

$$g(\mathbf{X},\mathbf{Y}) = 2\mathbf{X} + \frac{3\mathbf{X}}{\mathbf{Y}} \tag{2.2}$$

Each random variable possesses a probability distribution $F(\mathbf{X}) = P[\mathbf{X} \leq x]$ where x is a certain realization of the random number **X**. A sample can be generated from this distribution and the distribution of each of the other variates and substituted into the system response function. This provides a sample from the distribution of the sample response function. This avoids the problem of having the determine the distribution of the system, response function (this problem is addressed in Section 3.5). As can be observed from Equation (2.2), different values of the samples of the variates of **X** and **Y** will give different samples from the function $g(\mathbf{X}, \mathbf{Y})$. If we define a certain criteria for success such as $g(\mathbf{X}, \mathbf{Y}) > z$, we can judge each sample a success or failure.

The system reliability is then determined from the long term average of the simulation output; i. e., the number of system successes divided by the total number of system trials [133, 116]. As a result, simulation results are very dependent upon the underlying models which may not be representative* (Section A.1.2.5 of

[*] Section 2.2.3.1 shows the results of simulations of a four-bar mechanism and the effect of differing probability distributions.

Appendix A addresses distribution selection). Also, simulation techniques generally require long computer runs which are not attractive to on-line reliability estimation.

The stochastic simulation approach has had only a few instances of application to robot system performance, most notably in the determination of accuracy characteristics [18, 34, 54]. The topic of accuracy and repeatability will be addressed in Section 2.2.4. However, robot (and continuous system) performance has been evaluated using deterministic, continuous and discrete simulations for many years [36].

2.2.2. Robot Hardware Reliability Requirements

Design can be described as the solution of a problem by the application of physical principles subject to specifications. Those specifications must deal with the operational characteristics of the system which are reliability, maintainability, and availability. These specifications usually are difficult to develop and the transfer of the specification to constraints on the design such as strength and dimensional properties, and complexity of the system, is also a difficult task. R&M specification development is addressed in Section 3.2.2.2. Nevertheless, reliability, maintainability, and availability are an important part of the system requirements, usually some of the most important, since if the system is broken or inoperative, it doesn't matter how well it performs (see Section 2.1).

2.2.2.1. NASA: Space Based Robotic Systems.

The space environment is perhaps the harshest environment that is contemplated for robotic uses. The National Aeronautics and Space Administration

(NASA) is interested in the use of robotic systems in space due to the labor intensive construction and maintenance of Space Station Freedom (SSF). A report issued from NASA Johnson Space Center estimates a total of 32,746 hours of human Extra-Vehicular Activity (EVA) to maintain the station [52]. Based on the report, NASA is expected to require up to 86% of this maintenance to be performed by robots. The space environment consists of cosmic rays, solar particles, micrometeoriods, space debris, etc. (see Table 2.1). The types of failures by these environmental factors range from software and logic errors to material degradation as well as the causes of complex systems in less hostile environments such as random electronic failures, etc. The difference in the space environment is that space-based robotic systems cannot afford to lose their operational status since they will be performing tasks critical to the success of the missions involved. These systems will be required to perform reliably for years without maintenance if on a planetary fly-by or unmanned orbital mission. This requires the systems to be fault tolerant, since failures will inevitably occur. Current space system reliability is generally not specified in terms of reliability measures such as MTBF, but in the number of failures a system can tolerate [123]. However, Erickson suggests that space-based robot systems must possess greater than 99% reliability and greater than 90% availability with fault-tolerance provided although no time constraints by which to measure the reliability was given [48].

The subject of robotic fault-tolerance is of relatively recent interest. NASA has funded a study at the University of Texas that centered upon fault-tolerant techniques in the design of robot systems [148]. The study developed four levels of fault-tolerance in robot manipulators: Dual actuators, parallel structures with excess actuators, excess overall degrees of freedom, and dual arm manipulators. The study goes on to recommend the use of Level I (dual actuators throughout) and Level III (extra degrees of freedom) in the near term with dual arm manipulators for space deployment. These design recommendations were made to meet the specification of a manipulator to be two-failure tolerant. This means the system would be able to tolerate the failure of one or more component with limited degradation in operation (see also Section 4.4).

Table 2.1. Space Environmental Characteristics [48]

Characteristic	Failure Effect
Cosmic Ray and Solar Particles	Software failure due to memory bit alterations Altering/Weakening of materials Electronic failure due to particle bombardment
Micrometeoriods Space Debris	Structural damage and erosion Impact damage on electronic components
Temperature Fluctuations	Temperature induced electronic instability Lead failures due to thermal stresses Vacuum bonding of metallic unions
Vibrations g-Forces/Launch	Electronic lead failure Metal crack propagation Bending and torsional stresses and fractures

2.2.2.2. Nuclear Power Industry

Another harsh environment where robots have been extremely useful is nuclear fuel handling and reactor maintenance. The main reason for using robot manipulators in the nuclear industry is the reduction and avoidance of human radiation exposure and secondarily to achieve significant cost savings. Current manipulator systems are used for reactor fuel handling and transportation. The tasks contemplated for a mobile manipulator system for plant maintenance would be filter changes, valve maintenance, and radioactive waste drum handling [44]. Additional tasks envisioned can be non-destructive testing of plant components and reactor vessel inspection and repair. The EPRI report describes the environment the robot must be able to withstand and that it possess "an ability to be readily available for operations." This environment has an ambient radiation dosage rate from 0.5 millirad/hr to 50 rad/hr, for filter and valve replacement, to 500 rad/hr for waste handling. The humidity will range from 15% to 100% and temperatures will be up to 140°F. The specifications also include a lifetime dose (30 years life) of 63,000 rad. This is obviously a very harsh environment requiring and intensive reliability design strategy.

2.2.2.3. Industrial Robot Systems

As mentioned in the introduction to this section, it doesn't matter how good a machine is if it is often broken. This is the driver for high reliability in the industrial environment. Engelberger suggests that while dependent on the application, a robot system availability of 97% is required for customer satisfaction in usage of the robot [45]. This requires both high reliability and fast time to repair. Some may argue that the industrial environment is the harshest of all; the only reason nuclear and space applications are considered more extreme is inaccessibility to the robot system for repair in those environments. One thing that can be said for the industrial environment is that it has the widest diversity. Munson examines the industrial robot environment with respect to reliability during design [111]. The industrial robot environment includes extremes in temperature and humidity, atmospheric contaminants, heating, shock, vibration, noise, etc. Consider the use of robots for handling billets during a heat-treating process. The end-effector must survive repeated exposure to temperatures up to 2000°F during normal processing. For design purposes, Munson suggests that availability be used as the measure of dependability since the economics of the robot installation can be incorporated into the design process. In other words, it may be more acceptable to have a robot with 500 hours MTBF and 4 hours MTTR than one with 5000 hours MTBF and a 40 hour MTTR.

2.2.3. Robot Kinematic Reliability

Ideally, a designer would like to have a single number (or performance index) that has all the system's design criteria (specifications) embedded in it. This number could be used to select between design alternatives and during the operation of the system, could be used to help monitor the system condition to allow detection of system degradation. An extensive amount of work in the development of these decision criteria has been performed at the University of Texas [16, 29, 28]. The criteria investigated to date are largely associated with the structure and workspace of manipulators. What we would like is a way of including as many of the operational characteristics discussed above with the performance characteristics of the manipulator to give a system level performance index. The University of Texas report by McAndrew and Tesar [89] actually suggested the need for such a measure combining hardware reliability and precision in the microelectronics assembly applications. To do this, we have to find compatible ways of measuring performance and system reliability. Fortunately, reliability theory can be used to quantify several important performance characteristics: accuracy and repeatability (termed kinematic reliability in the following sections). This compatibility will allow us to develop this system level performance criteria in Chapter 5.

Thus far, our discussion has been relegated to discussing the reliability of the hardware associated with robot system: joints, links, motors/actuators, modules, controller hardware and/or software. Up to now, we have implicitly defined failure as the lack of proper operation. Let us now propose a more explicit definition: Robot failure is the failure to achieve a certain position or orientation of the end effector within a certain error margin [18]. This section will explore a definition of "kinematic reliability" based upon this definition of failure.

2.2.3.1. Closed Chain Kinematic Reliability and Synthesis

The definition of kinematic reliability revolves around a stochastic description of the kinematics of mechanisms. From the theory of random variables, a function of random variables is a random variable [95]. To generate a sample from the probability distribution described by a function of random variables, samples from the individual random variables that make up the function can be generated and substituted into the function. If the functions resulting from a kinematic analysis of a mechanism are examined, one finds they are made up of trigonometric functions of the kinematic variables of the mechanism such as link lengths and joint angles. If these variables can be described with probability distributions, then any point in the kinematic chain can be described by the function of random variables implicit in the kinematic formulation. This stochastic description is the fundamental basis for the formulation of the *Kinematic Reliability* [17] and has been used extensively in the design and analysis of linkages and the errors in the path described by those linkages.

Interest in the stochastic description of mechanisms can be traced back to the early 1970s with the development of a stochastic model for the analysis and synthesis of mechanical error in four-bar linkages [40]. In the mid 1980s, Crawford and Rao defined the reliability of a function generating mechanism as "the probability of generating the desired function with a specified accuracy in spite of the detrimental effects of production tolerances and clearances" [37]. Using this definition, they generated the reliability of a four-bar straight line mechanism using Monte-Carlo simulation. By writing the vector loop equations, Crawford and Rao developed an expression for the output point position in terms of the input angles. Simulations performed over an input range allowed them to determine the reliability of the mechanism via the frequency interpretation of the ratio of number of successes to total number of simulations. They used the normal, Weibull, and beta distributions to describe the kinematic variable distributions and showed the general utility of the simulation approach. Simulation was used due to the complex non-linearities in the transcendental solutions of the vector loop equations.

Another definition of reliability for closed loop path generating functions has been proposed by Sukhija and Rao in two papers [150, 151]. Sukhija and Rao where interested in the synthesis of four bar path generators. Specifically, they defined an error based reliability index (not a definition of reliability) to measure the error of the actual path generated by the mechanism from the desired path. The error was assumed to be normally distributed and the reliability at each point in the path was represented as a ratio of the actual squared error to the maximum permitted squared error. The reliability index was the average of the reliabilities of the points along the path. They then performed an analytical synthesis of the link lengths of the four bar mechanism by allocating the error tolerance on link lengths. The synthesis route described was deterministic but can allow for stochastic errors in the definition of reliability along the path. This effort showed how stochastic error can be used to create deterministic bounds on the design variables (link lengths in this case). The advantage is that optimization by forming an objective function and then minimizing is not necessary; the synthesized reliability index is minimized by the error tolerance allocation.

Table 2.2. Simulation Results for Four-Bar Function Generator [37]

Distribution	Reliability
Weibull	0.919
Normal	0.9514
Beta	0.9875

2.2.3.2. Open Chain Kinematic Reliability

Closed loop kinematic chains have characteristic loop equations that can be used to solve for specific input-output relationships. Open loop kinematic chains (such as serial manipulators) pose a more complex problem since the position and orientation of the chain terminus can be arbitrary. The problem of a probabilistic analysis of open kinematic chains was addressed by Bhatti and Rao [18]. Bhatti and Rao define the reliability of a manipulator as "the probability of end-effector position and/or orientation falling within a specified range from the desired position and/or orientation." We choose to designate this definition of reliability as the "*kinematic reliability*" of the system, since it is dependent upon the kinematic variables of the manipulator system, and to differentiate it from the hardware and software reliability of the system. Bhatti and Rao designate three types of positional reliability and two types of orientational (depending on the constraints one wishes to place on the end effector position and orientation). They then describe two methods of solution, analytic and simulation. By comparing values obtained for a simple two-link manipulator by the analytic and simulation methods, they conclude that the analytic method suffers from accuracy problems caused by truncation of Taylor's series expansions and numerical integration and the Monte-Carlo simulation method is preferable. Further discussion of this work will be presented in Chapter 5 during the development of the Reliability Performance Index.

In a series of three papers, Gao and Wells develop a simulation model for robot accuracy and repeatability based on Bhatti and Rao's simulation method. They develop input data for a PUMA 560 robot and perform simulations to generate accuracy and repeatability statistics [54, 55, 56]. They also characterize the accuracy and repeatability statistics and perform factorial tests for significance of contributions to the errors. Gao and Wells found that the orientation errors (errors in the Euler angle rotations) are the most significant effects on position error. The errors in the joint rotations have 60% effect of the orientation errors and link dimensional errors have an effect that is 25% of the orientation. These characterizations are then put to use when the reliability of the position for assembly is examined. This reliability is defined in the same way as Bhatti and Rao where the error specification is the allowable offset which permits a successful assembly operation to occur. They show how this reliability can affect the choice of configuration for particular operations by comparing the results of the PUMA to a SCARA robot. They also provide a design chart to select assembly clearances based on the robot to be used for the assembly operation. Figure 2.3 shows their results for the position reliability as a function of radial distance of the end-effector from the robot bases. This shows the PUMA is better for work closer than 10 inches from the base and the SCARA has better positional reliability further out.

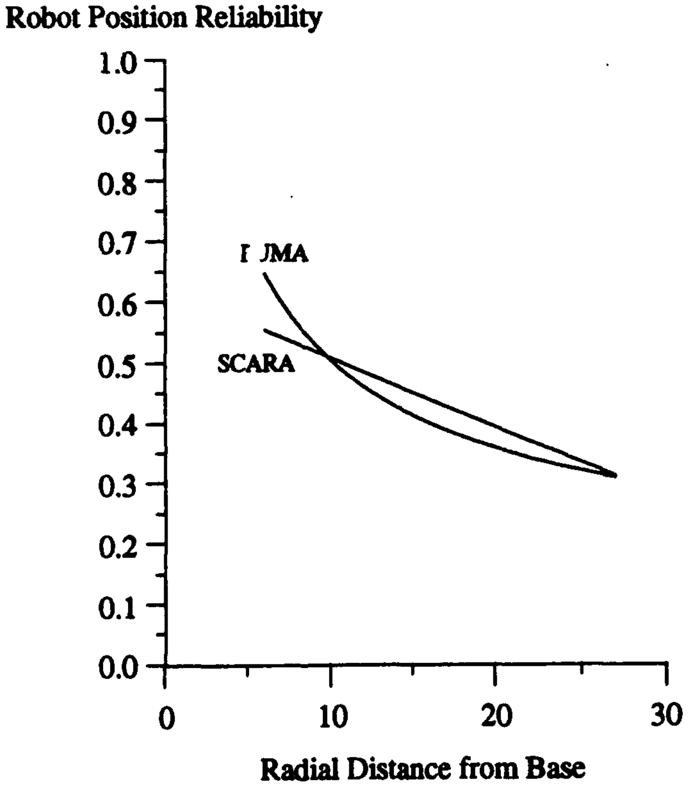


Figure 2.3. Robotic Positional Reliability v. Radial Distance [56]

2.2.4. Error Specifications

If the performance specifications of industrial robots are examined, one of two categories are always stated: accuracy or repeatability. This section examines the various definitions of accuracy and repeatability and how they are quantified.

2.2.4.1. Accuracy and Repeatability

One of the most in-depth examinations of accuracy and repeatability was accomplished by Colson and Perreira [34]. They rigorously defined accuracy and repeatability and examined the statistics associated with these performance measures. Accuracy is defined as the difference between an achieved task and a desired task with no prior knowledge of task performance. In other words, a measure of how well the manipulator follows its programmed route. Repeatability is described as the measure of the ability of a manipulator to arrive at the same location over many separate trials. Colson and Perreira describe three classes of accuracy, two classes of repeatability and examine sensitivity of each characteristic with respect to payload, path, and location within the work volume. Absolute accuracy is defined as the difference between a desired task and the task actually achieved. Relative accuracy is defined as the difference between the achieved task and desired task when the desired task is defined with respect to some reference frame. Palletizing accuracy is described as the difference between an achieved task and a desired task that is interpolated between two reference tasks. Unidirectional repeatability refers to the repeatability of a manipulator when the task(s) performed by the system do not vary from cycle to cycle. Omnidirectional repeatability refers to a changing task such as a robot switching between processes of assembly lines. Unfortunately, Colson and Perreira do not provide any numerical examples, they only present an Analysis of Variance (ANOVA) model that provides sensitivity measurements for the different performance parameters involved in the experiment such as accuracy and repeatability. They also do not discuss how this data should be obtained.

Mooring and Pack describe a method of collecting robot repeatability data and how to produce a specification of robot repeatability [108]. Their definition of repeatability is implied to be the same as Colson and Perreira, identifying positional and orientational repeatability. Their concern is that robot repeatability varies as the robot moves in the workspace. They describe a special end-effector and fixture used to collect repeatability data and suggest that the data should be collected at two positions in the workspace, at the midrange of the joint positions and at the joint limits. This gives the best and worst case of repeatability values that can be included in specifications. This technique can only be applied after construction of the robot and cannot provide design related information except for modification. Using a manipulator with spherical geometry (similar to the Stanford Arm), they measured the standard deviation of the raw sensor data and by inverting the manipulator Jacobian, determined the standard variations of the joint motions (see Table 2.3). For the example manipulator, the roll axis of the wrist was the largest contributor to the variation of the repeatability.

Table 2.3. Standard Deviation of Joint Motions [108]

Joint Motion	Standard Deviation (Degrees)
Waist	0.00107
Shoulder	0.00166
Arm	0.00153
Roll	0.05989
Pitch	0.01792
Yaw	0.01523

Bhatti and Rao also address accuracy and repeatability. They show that kinematic reliability can be used directly as a measure of accuracy and repeatability by specifying which kinematic variable to include in the kinematic reliability formulation [18]. If all the kinematic variables are included, the reliability value indicates the accuracy of the manipulator. If only joint angle variability is included, the reliability value becomes the repeatability. They showed how the reliability (accuracy is all the kinematic variables are used) varies over the workspace of a robot. Their specific results for the Stanford arm is presented in Table 2.4. It is this property of kinematic reliability that makes it attractive as a performance criteria. During the top level integration of a robotic system, operational characteristics must be traded off against performance characteristics to arrive at the overall solution. A combination of kinematic and hardware and software reliabilities may be able to provide a useful tool for performing these top level trade-offs. Chapter 5 begins the development of this tool.

Table 2.4. Reliability of the Stanford Arm [18]

Case	Mean Joint Variable						Reliability (Accuracy)		
	θ_1	θ_2	d_3	θ_4	θ_5	θ_6	Position	Orientation	Overall
1	30	90	10	45	0	60	0.9739	0.9613	0.9428
2	-45	45	10	0	30	0	0.9849	0.9147	0.9095
3	90	30	10	45	-60	-30	0.92387	0.8530	0.8346
4	60	-30	15	-75	90	120	0.9888	0.9748	0.9696
5	-135	20	12	60	-50	120	0.9229	0.7962	0.7792

2.2.4.2. Analysis and Design for Accuracy and Repeatability

The statistics Colson and Perreira gathered are based upon the mean and variance of samples of an error vector which describes the position and orientation error of the achieved end effector with respect to the desired position and orientation. A complete description of the method used to describe this error using an equivalent rotation vector to describe the end-effector frame orientation can be found in [33]. They then present a sample Analysis of Variance (ANOVA) experiment to determine the sensitivity of the performance measures to sources of error. The ANOVA experiments suggested are factorial to reduce the amount of testing required. They suggest sources of error such as algorithm inaccuracies, digital sample time, servo system deadband, actuator inaccuracies, and system compliances, but as stated before, no numerical results are provided. They then make several design recommendations such as using improved robot calibration techniques (this refers to the "teach" method of robot programming), including anti-backlash mechanisms, and improved models of the system, including the effects of gravity loads and machining errors.

Bhatti and Rao also suggest the use of kinematic reliability in the design and planning of a robot workstation. They suggest the development of design charts based upon the manipulator's kinematic reliability at critical points in the workspace and the standard deviations of the joint variables (which are directly proportional to the actuator tolerances) Figure 2.4 shows the effect of the standard deviation of the joint variables. As the allowable variation increases, the kinematic reliability of the manipulator drastically decreases as expected. These charts can then be used to select the actuator specification needed to achieve a desired performance from the manipulator and can be useful in prescribing manufacturing tolerances. They also propose the use of kinematic reliability to choose between configurations of the manipulator. These suggestions were made for workcell design, but one can easily see how they readily apply to the modular robotic system problem as well.

The measurement of errors in robot systems is often called *robot calibration* or *robot metrology*. The University of Texas has extensively studied the contribution of measurement errors and compliance to accuracy of a robotic system by examining the model parameters [66, 144]. They have developed a global compliance model with about 125 parameters measured from the robot system and have shown that deformations in the physical structure of the manipulator can cause position error that is 50 times the robot repeatability. The model is based upon measurements of the joint and link compliances which can then be aggregated into an overall compliance model at the end-effector. Techniques to measure and construct the global end-effector compliance matrix are presented but no statistical data reduction is presented to develop the sensitivity of the model to measurement or inherent component variability. As presented, the model is deterministic. This model information can be used during the design of a robotic system to predict the accuracy and repeatability of the system as well as improve performance by acquiring precise knowledge of the actual, installed system. Since the deformations can cause extremely large errors, this information should be included when quantifying the overall "performance reliability" of a manipulator system. In the kinematic reliability sense, compliances will increase the variability of the joints (and perhaps the links) and will decrease the level of reliability achieved.

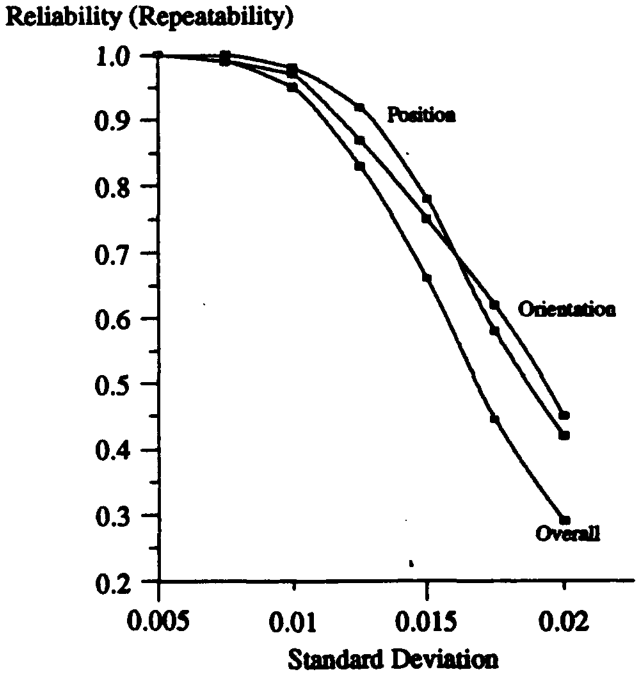


Figure 2.4. Reliability vs. Standard Deviation for the Stanford Arm [18]

2.2.4.3. Control System Reliability

As mentioned before, robot system design does not occur in a vacuum. The control system directing the robot system must also be designed to meet the performance specifications. The durability specifications apply to the whole system, of which the control system is an integral component. Several ways to analyze the reliability of control systems exist. Control engineers usually consider control systems to be reliable if stability can always be guaranteed [158]. This is an important aspect of control system design, but not particularly germane to this discussion. The reason is that this failure definition provides a very narrow definition of reliability for a control system that is not quantifiable for use in an overall reliability model. Robots rarely fail due to lack of stability: this is always addressed during the control algorithm design. We are interested in those aspects of control systems that can be quantified via reliability theory. One way to quantify control system reliability is to perform a standard hardware reliability analysis on the controller components. While providing important information about the dependability of the control system's components, it doesn't tell the whole story since the controller also includes software components that affect the system reliability as well. The software usually is the major contributor to the reliability of any system since it is usually the most complex. Many software models exist and most definitions of software reliability are completely compatible with the definitions of hardware reliability. This means that software can be considered as just another component of the system with its contribution to an overall failure rate [113]. In cases such as these, the appropriate type of model for the control system (controller and software) is a Markov Chain. McGough, Reibman, and Trivedi provide an excellent overview of Markov modeling of control systems [90]. Software reliability is discussed in Section 3.3.4 and in Section 4.3.9 with a numerical example of a modular software model in Section 5.3.3.

While modeling the reliability of a control system is important in the determination of the system reliability, it only provides an analytical view of the design. One would like to address reliability during the design. One way to improve system reliability is to add redundancy (see Section 2.3.5, Section 4.2.1 and Appendix A). This strategy is also successful in the design of reliable control systems. Siljak [143] describes a method of designing reliable control systems by using multiple control systems to provide redundancy. He describes a parallel combination of control systems that provides redundancy at the controller level. While he presents satisfactory results, his assumptions do not allow fault tolerance to be designed into the controllers. This problem is addressed by Cho and Biem [25] who propose the redundancy be actively adaptive, which allows for a large measure of fault tolerance to be included in the controller design. Cho and Biem also take the step of actually quantifying the controller reliability to show the increase in reliability due to their design methodology. This quantification is performed via a Markov model of the adaptive system. By solving the Markov models for the system with and without adaptive redundancy, they form a ratio between the Mean-Times-to-Failure (MTTFs) and show analytically that this ratio is greater than one. Mathematically,

$$\frac{\mathrm{MTTF}_{\mathrm{Adaptive}}}{\mathrm{MTTF}_{\mathrm{Standard}}} \geq 1 \qquad (2.3)$$

They do not, however, say how much greater than one this ratio will be. It is dependent upon the various failure rates within the model. Another question unanswered is how to evaluate the worth of the improvement: Will the reliability improvement be worth the cost of implementation?

An area related to the stability issue mentioned above is the "correctness" of the algorithm used in the control computer. The control designer may, develop the control law correctly such that it stabilizes the system and provides adequate performance, but the code written to implement the control law (the control algorithm) may have errors due to human programming. This is a software reliability issue and can be addressed through good software engineering techniques. However, there exists a way to quantify this software reliability by an analysis of the logic structure of the program. An overview of this technique, called Logic Structure Reliability Analysis (LSRA), can be found in [84]. This analysis is dependent on the structure of the program, path length, execution time, reachability, and connectivity of a tree developed from the algorithm. Failure probabilities are developed from this tree for connection, reachability, and execution and can be combined to give an overall control logic reliability very easily. This is an analysis tool rather than a design aid, since the algorithm must exist before this analysis can be performed. However, it does allow the for the prediction of software reliability based upon the software logic rather than an assumed "bug" count.

The example given in this work is a controller for a heat exchange system for a nuclear reactor. This control system has the directed graph of which a small piece is shown in Figure 2.5. Each vertex (only three are labeled) in the graph represents a decision or transfer point in the program, the start or end of an iteration sequence, or the beginning or end of the program or subroutine. The arcs connecting the vertices are known as edges. The connectivity of the directed graph is represented by a matrix showing the connections between the vertices. In this case, the connectivity matrix is 33 by 33. The number of paths through the logic of the control system was calculated to be 1,898 with 15 direct paths. There are several types of reliability measures that can be generated from the graph. They are connection, reach, execution, logic structure, program, and overall adaptive control logic reliabilities. Connection reliability is based upon the probability that two vertices are connected. Reach reliability is the probability that a vertex can be reached from the entry vertex. The execution reliability is a measure of the ability of the code to correctly execute a given path. The logic structure reliability is a combination of connection, reach, and execution reliabilities. The program reliability is a "bug" estimate (see Section 3.3.4 for software reliability "bug" models) and the control logic reliability is the overall reliability prediction for the code. The example given in Liang, et. al. resulted in the values reported in Table 2.5. The exact formulation and methods for calculations of the values in Table 2.5. can be found in [84].

It is interesting to note that control theory is based on the description and removal of error through feedback of sensor data. The reliability of anything that is capable of being measured by an error (a difference between desired and observed values) can be treated in a manner similar to that of the kinematic reliability of a manipulator. The stochastic error model may differ, but the basic idea is the same. McInroy and Saridis have developed a method for analyzing the reliability of control algorithms with respect to a desired task and specifications [91, 92, 93]. A group of feasible algorithms (called plans) are constrained by probabilistic performance constraints (formalized through entropy distributions). A reliability index is defined for each specification (based on a Maximum Likelihood Estimator (MLE) for the mean of each specification) and combined to give the overall probability of the algorithm meeting the specifications. These reliabilities can be compared and the best algorithm can be selected. The process described is not trivial, the description of probabilistic constraints (called reliability performance functions) can be exceedingly difficult. However, the method does allow for a description of the reliability of a control system very similar to the kinematic reliability of a manipulator.

2.2.5. Results of the Robot Reliability Literature Review

The review of literature pertaining to robotic reliability shows that there has not been an adequate job of addressing the reliability of a robotic system during design. Dhillon [42] had an opportunity to do this but he did not correlate the tools with the design process. Also, any studies presented in the literature (except for the three noted), do not present any acquired life data and only presented the underlying model used for analysis.

The results of this review is presented in Table 2.6. Each work is ranked in from 0 to 10 in the categories of data contained, depth of presentation, complexity of the problem, applicability to modularity, currency of technology, and applicability to design given in the paper.

Table 2.5. Reliability Measures for the Control System of Figure 2.5 [84]

Reliability Type	Value
Connection	0.3404
Reach	0.0351
Execution	0.0851
Logic Structure	0.3636
Program	0.9980
Adaptive Control Logic	0.3629

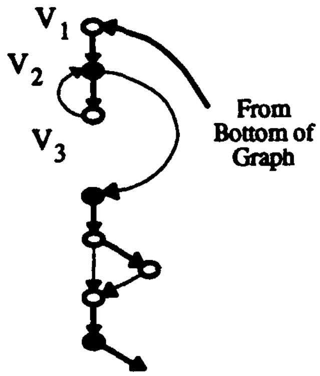


Figure 2.5. Directed Graph of a Sample Adaptive Control System Algorithm[84]

Table 2.6. Robotic System Reliability Literature Rankings

Work and Comments	Model	Data	Depth	Complexity	Modularity	Currency	Design	Total
Engelberger (1974) [45]	RBD	9	7	7	0	0	1	24
Butler and Tesar (1992) [22] Design criteria rankings	None	0	0	2	2	9	5	18
Cox and Tesar (1992) [35] Suggests 150,000 hours MTBF with no environmental statement	None	0	0	2	2	9	5	18
McAndrew and Tesar (1991) [89] Suggests definition of robot reliability including performance	None	0	0	2	2	9	5	18
Jones and Dawson (1985) [72] Strong data analysis but cannot make conclusion since technologies are not stated	None	7	5	2	0	2	3	19
Grundmann (1989) [61] Presents extremely strong argument for modularity	None	9	6	6	10	5	1	37
Dhillon (1991) [42] Gives goals: 400 hr MTBF and 8 hr MTTR based on [72] and [152]. Not realistic for current technology.	Many	2	2	2	0	0	5	11

Table 2.6 (Continued). Robotic System Reliability Literature Rankings

Work and Comments	Model	Data	Depth	Complexity	Modularity	Currency	Design	Total
Sugimoto & Kawaguchi (1983) [152] Quotes Japanese MTBF and failure rates. Suggest Intelligent Control	FTA	2	5	5	0	0	2	14
Khodabandehloo, et. al., (19840 [76,77]. Used MIL-STD-217 generic data.	FTA	5	8	6	0	0	3	22
Gordon and Curry (1985) [59] Weaver and Denninger (1991) [164] Took data but did not present. Recommended design changes	FTA/ FMEA	0	0	4	0	0	9	13
Wells, et. al., (1988) [165] Analysis only. Recommended design changes	FTA	5	0	5	0	0	9	19
Cohen and Chandra (1984) [31] Robot failure = performance degradation	Markov	0	5	5	0	0	0	10
Crawford and Rao (1987) [37] 4-bar simulation at Purdue	R_K	0	5	5	0	0	0	10
Sukhija and Rao (1986, 1988) [150, 151]. Tolerance allocation	None	0	5	5	0	0	2	12
Bhatti and Rao (1988, 1989) [17,18] Provided original definition of kinematic reliability	R_K	3	8	9	0	5	6	31

Table 2.6 (Continued). Robotic System Reliability Literature Rankings

Work and Comments	Model	Data	Depth	Complexity	Modularity	Currency	Design	Total
Gao and Wells (1990) [54-56] Positional data for PUMA	R_K	3	8	8	0	5	3	27
Colson and Perreira (1985) [34] No examples. Just method.	ANOVA	0	7	7	0	2	5	21
Mooring and Pack (1987) [108] Showed repeatability to vary over workspace	Stat.	3	9	9	0	5	6	32
Hudgens and Tesar (1992) [66] Deterministic measurements on in-depth compliance model (*)	*	5	9	9	5	7	9	44
McGough, et. al., (1989) [90] Markov model of control systems	Markov	0	8	7	0	5	6	26
Cho and Biem (1989) [25] Adaptive Control Reliability	Markov	0	8	8	0	6	4	26
Liang, et. al. (1989) [84] Software Logic Structure Reliability Analysis (LSRA)	LSRA	8	8	8	0	7	0	31
McInroy and Saridis (1991) [91-93] Very difficult to apply. Examples only showed control system algorithm design	None	0	6	9	0	8	6	29

2.3. Reliability During the Design of Modular Systems

Most people would agree when one states that the computer is the device that has had an enormous impact on all aspects of technology. We find computers everywhere: banks, supermarkets, entertainment devices, cars, etc. The question can be posed: why? Besides the obvious answer of being able to perform calculations at incredible speeds and quantity, why do people accept and trust computers to control so much of their lives? One partial answer can be found by addressing the question of reliability and modularity. Consider today's Personal Computer (PC). One of the main reasons for its popularity is their availability; they operate for long periods of time without failure and when they do fail, they are extremely easy to troubleshoot and repair. The reason is the maximization of reliability coupled with modularity.

2.3.1. Modularity: A Design Characteristic

The design process can be divided into several stages: problem definition/task clarification, conceptual design and feasibility studies, embodiment design, and detailed design [117]. During the problem definition stage, the designer must define the functions and operational characteristics of the system or process. It is at this time the advantages and disadvantages of modularity must be addressed. One clear advantage of modularity is the standardization required to implement interfaces between modules as well as stabilizing the module inventory. Modularity allows for the possibility of future expansion and upgrade of the system. Another advantage is the ability to provide a functional separation between modules. This functional separation can be employed for independent designs of modules, or modules that perform the same function but to different specifications. This will allow optimization of the module selection to meet the specifications. This approach is not generally applicable to mechanical systems since during mechanical design, functions are shared, but it does apply to a robot system when separated into joint/actuator and link modules [4]. As a matter of fact, the success of VLSI design can be attributed to the ability to functionally separate the system into sub-functions which are designed completely independently to a form, fit, and, function specification [127]. Modularity does carry some penalties. It can increase the complexity, since shared functions reduce the number of components needed. It can also increase the inventory costs of a system by allowing a wide range of modules to be available [74]. While not necessarily a drawback, this point must be taken into *account during the design feasibility study.* Modularity also increases the number of connections which always are likely failure points.

Thus, when examined from the perspective of the overall design problem, modularity can be seen to be a design characteristic or criterion. In other words, a definition of modularity (i.e. the level of modularity) is specified and the design proceeds according to the designer's preferred methodology. Specifying modularity adds design constraints to the system or process under consideration. The first of these constraints is the necessity for some kind of specification (Rinderle calls them design rules [127]) that allow the functional blocks to be successfully combined to form the system. This specification must describe interfaces (mechanical and electrical), communications protocols, and any other interactions between the modules. Another constraint alluded to previously is the system size or complexity. As the complexity increases, the design problem can become intractable, which can be ameliorated by the specification or design rules referred to earlier. Modularity also reduces the number of design possibilities by restricting the number of design configurations that are possible. For instance, it generally prevents the sharing of functions common in many mechanical devices. A monolithic system may take on any form (an infinite number of configurations) but will have only one configuration implemented at the end of the design. However, from a set of four revolute joint modules and nine link modules, over 7000 kinematically unique robots can be assembled [4]. Even though modularity is only a characteristic of design, there are some design principles that can be associated with modularity.

2.3.2. Modular Design Principles

A design principle can also be considered to be a design "rule-of-thumb" which if followed, will generally result in a design superior to a design not following the design principle. By searching case studies considering modular system designs (from all different domains such as electric power distribution systems, avionics systems, etc.) be can glean the different design principles for a modular system. The searches resulted in the following four modular design principles. The question being asked while we consider these principles is: How do we create an analytical process to take advantage of these modular design principles. Specifically, what analytic process can we use to increase the reliability of a modular robotic system? These answers are elusive, as discussed in Sections 3.5 , 5.3.1, and 5.4.2.

1. *Module Commonality and Functional Separateness.* The design methodology of modular systems is the same as for other design problems, but must include some principles that are driven by the modular characteristic. The first is

module commonality. The modular system is designed along functional divisions with a minimum of function sharing. Modules must be functionally separate from each other to allow for an explicit description of their function and use in the system configuration. Thus, each function must be separately designed and tested. Notice, the word function in this sentence. It does not mean each individual place in the design the function is used. Commonality means you only have to design, test, validate, and certify for the common design [154]. This will reduce the costs of design development as well as in production and logistic support of the system.

2. *Minimize interfaces and provide for interface control (standardization).* Most system failures can be seen to occur at an interface of some sort, such as a connector, solder joint, coupling, bolt, etc. This is because interfaces are generally *the places of maximum stress* [130, 23]. Thus, to provide for maximum resistance to failure and minimization of complexity, a modular design must minimize the interfaces required. Also, to provide for functional independence and independence of module design, a systematic, standard, set of design rules must be established for the interfaces [112]. This governs the information that is transmitted across informational interfaces and describes the load and force transmissions on mechanical interfaces. It also allows for independent improvement in modules, as long as the interface standard is maintained (a concept similar to form-fit-function replacement and improvement).

3. *Reduce Life Cycle Costs (LCS) through increased reliability of modules.* As alluded to in the previous design principle discussion, modular systems are capable of assuming many different configurations dependent upon the selection of modules to make up the system. Modularity also allows for quick diagnosis and

replacement of failed modules from the system. However, the point must be made that as the modular system becomes larger, the complexity of the system will be contained in the modules themselves. It is a generally accepted premise in the reliability community that more complexity (parts) creates higher failure rates and lower reliability [130]. Recall the availability equation from Chapter 1. Availability is a combination of uptime (quantified by MTBF) and downtime (quantified by MTTR). Availability is maximized by maximizing reliability and minimizing the repair time. Modularity insured low MTTR, the other component of availability left to the designer is reliability. Of concern also is the cost to support the modular system. This is generally referred to as the Life-Cycle-Cost (LCS) of the system. The LCS is made up of the cost of design, development, production (the manufacturing costs), replacement, disposal, and inventory costs (see Section 3.1.3). The design costs for a modular system can generally be regarded as higher and the physical cost of the modules will be higher due to complexity. They will have higher failure rates. The advantage in LCS occurs if the downtime costs outweighs the fixed cost in each up/down cycle, and modularization minimizes the downtime [74]. Modularization can actually reduce costs through the increase in reliability of the modules.

4. *Level of modularity chosen to provide maximum flexibility.* One of the most important advantages of modular systems is the ability to rapidly reconfigure, either to incorporate new technologies or adapt to new tasks and missions. The ability to adapt is directly driven by the level of modularity. If the level is too low, at the component or sub-assembly level, the ease of troubleshooting and repair disappears since you have many components to search through. The alternate is

choosing the level of modularity too high. When this occurs, the ability to rapidly adapt is abrogated by the architecture and the system becomes monolithic [128]. The challenge is to provide the level of modularity that minimizes the LCS of the system. Support costs must be evaluated early in the design and based upon module failure rates, inventory levels must be estimated to develop the proper inventory. Performance issues must be considered as well, but the cost of performance is hard to quantify to allow comparison to inventory (see also Section 2.3.5).

2.3.3. Reliability as a Modular Characteristic

The reliability of a modular system can also be considered a modular characteristic of the system. The traditional method of reliability analysis is a combinatorial method using a block diagram of the reliability structure of a system. These diagrams are called Reliability Block Diagrams (RBD) (see Section 2.2.1.3). In a static sense, the reliabilities of system components can be combined to provide the system reliability [73]. If the components in the RBD are modules, then the reliability model that describes the system is modular as well. In this way, one can see that system reliability is a modular characteristic. This is convenient in our consideration of the reliability of a modular system. If the modules are designed and tested independently and the specification of the interfaces between modules creates independence in life distributions, the module failure rates can be used in a RBD model directly to evaluate the reliability of the modular system. In other words, the modular characteristic has no intrinsic effect on the system reliability except in complexity, however, there is an effect of module reliability on the reliability of a system.

2.3.4. Modularity as the Result of Reliability Improvement

Recall the earlier discussion of the success of the personal computer. This success can be directly attributed to a modular architecture that allowed independent design of modules for different purposes. The modular architecture in computers can be traced back to improvements in reliability and fault-tolerant architectures [64]. During the design of IBM computer systems, the designers found that when transitioning to transistors, they were able to dramatically reduce the number of interconnections (interfaces and sockets) which provided an increase in reliability. At the same time, the failure rates of the logic and storage elements were being reduced. This allowed the systems to be designed for serviceability, which evolved directly into a modular architecture. In the early 1960s, IBM developed the 7030 computer which had standard modular cards as field replaceable units. This allowed IBM to completely revamp its maintenance concept, removing the "customer engineer" from the customer's location allowing the customer to troubleshoot and repair their system themselves. This provided higher availability and customer satisfaction and allowed easy upgrades and improvements in their systems. All of their computer systems (including PCs) designed since then have incorporated a modular fault-tolerant architecture for these reasons. Thus, one can see that the reason for the success of the PC is directly attributable to high reliability driving a modular architecture.

2.3.5. Level of Modularity: The Effect of Reliability

Modularity is supported by high reliability, but can be detrimental when module reliability is low [74]. A life-cycle-cost problem is introduced; the cost of the module must be cheap relative to the services provided by the system, determined by the cost-to-reliability ratio of the module and the system. It also has a direct effect the other way around. The level of modularity has a direct effect on the reliability. In fact, Upadhyaya, Pham, and Saluja have demonstrated that for fault-tolerant systems using module redundancy, system reliability can be used to determine the optimal module size (granularity or level of modularity) [156]. Their approach was applied to a fault-tolerant computer system that relied on module redundancy to provide the high reliability required for the system. They show how to increase the ability of a system to tolerate faults by partitioning the system at the submodule or component level. The technique involves adding additional voters into an *n*-module redundancy system providing additional fault coverage if a subcomponent fails. They then optimized the number of partitionings required to maximize the reliability of the system with respect to the number of components (there is a point of diminishing returns in this method; see Figure 2.6). This approach can be used for systems where the circuits are amenable to further partitioning, such as logic circuits. The problem arises when applying this argument to a mechanical system such as a robot manipulator that is not hierarchical in architecture (it is generally serial). This type of system is usually monolithic and not amenable to partitioning or the addition of redundancy. Redundancy in this case means duplicating components to provide backup during a failure, not excess degrees of freedom. However, if the robot architecture is modular, this type of partitioning can be attempted (Robotic Fault Tolerance Levels I through IV from [148] (see Section 4.4)).

Consider a robot joint module with two actuators such as that described in [148]. The duality of the module is complete down to the encoders that record shaft position. Voters can be provided for these sensors in the controller, and if the reliability of the actuator components are known, this method can be applied to determine if the module should be reparable at a lower level, such as separate motor or encoder replacement. This would be a very valuable tool during the initial design phase of a modular robot, but has the limitation of requiring life data on the components.

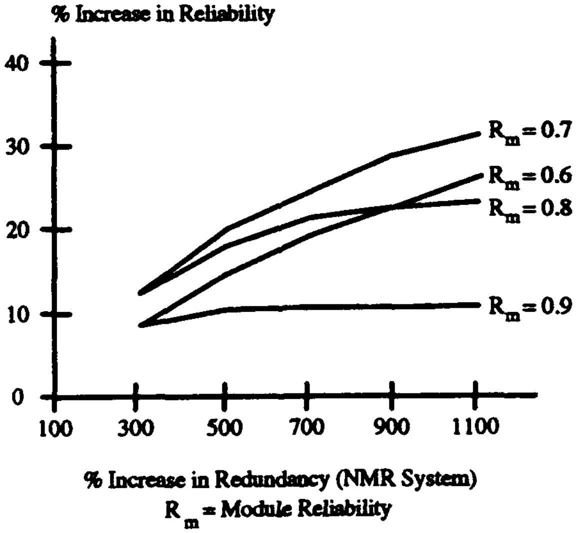


Figure 2.6. Percentage Increase in Reliability vs. Redundancy

2.4. Summary

This chapter has presented a review of the current literature on robot system reliability and modular system design with an emphasis on reliability. Most robot system reliability studies where undertaken to support safety analyses with relatively little done on robot reliability in recent years. Additionally, no published reports of specific reliability data exists and that which does [42, 45] is either extremely dated or so non-specific as to be useless for specific applications. There have been several types of reliability analysis applied to robot systems, but none published utilizing current technologies. Thus, any analysis pertaining to modular robotic systems reliability currently must use generic data as found in MIL-HDBK-217 [96] and the *RADC Non-Electronic Parts Reliability Data Handbook* [131].

As shown in the reviews of the papers, the reliability of robotic systems during their design has not been adequately addressed. No author presented a case study of the design of a robotic system showing how reliability was addressed during that design. This may be due in large part to the extremely high competitiveness that currently exists in the robotics industry. Manufacturer's feel that reliability data is proprietary, and guarding this data is essential to their business. While it is clear that this attitude hinders industry-wide improvements in technologies and design, it is also clear that these concerns must restrict access to this critical competitive data.

This chapter also examined the design principles of modular systems by reviewing design case studies of modular systems. Four modular design principles where identified: module commonality and functional independence, minimization of interfaces and interface standardization, reduce life-cycle-cost through module reliability enhancement, and choosing the level of modularity to provide the maximum flexibility. These principles form the basis of a paradigm for the improvement of reliability of robotic systems as described in Chapters 3 and 4.

Modularity does not, in general, effect the reliability of a system one way or another. Modularity is only a specified characteristic of a system that imposes some functional and commonality constraints on the design. The system reliability does not depend on whether the system is modular; it depends upon the system architecture (such as redundancy) and upon the failure rates of the component parts of the system. These constraints can effect the reliability of the system, but this effect is application specific. In general, complexity degrades the reliability of a system and a modular system may be more complex due to the lack of function sharing and interface requirements. However, this degradation may be offset by the use of standardization and commonality which can increase the reliability of the modules, increasing the system reliability. This coupling will be unique for each system.

CHAPTER 3: DESIGNING FOR RELIABILITY

3.1. Reliability and the System

The user of a system will generally not be concerned with the reliability of components or even the overall system reliability. However, he or she will have a keen appreciation of the availability of the system. As discussed in Chapter One, availability can be described for a system as the ratio of operating time to the total time of a system, or

$$A = \frac{\text{Operating Time}}{\text{Operating Time} + \text{Downtime}} \tag{3.1}$$

All of the user's criteria for measuring the effectiveness of his systems will be based on profit and availability of the system is one of the prime components in the economic analysis of any process. If the assumption of constant system failure rate and repair rate can be made[1], Equation (3.1) can be expressed as

$$A = \frac{\text{MTTF}}{\text{MTTF} + \text{MTTR}} \tag{3.2}$$

where MTTF is the Mean-Time-Between-Failure and MTTR is the Mean-Time-To-Repair. This formulation of availability shows the direct relationship to reliability and maintainability. The system user, while not being directly aware of the system

[1] The time to failure distributions for large complex systems can be shown to generally follow the exponential distribution which implies a constant failure rate [73]. A constant failure rate assumption is usually made for convenience but, fortunately, the same steady state solution holds if a constant repair rate cannot be assumed. Another important assumption requiring examination is the independence of life distributions of the components (See Section 3.5).

reliability or maintainability measures, is directly effected by them. This makes it in the best interest of the system designer to be aware of all the techniques and technologies available to maximize the reliability of a system while minimizing the downtime of the system.

There are two sides in the achievement of high reliability in systems. The first side pertains to planning, program management, and contractual guarantees to insure the contract specifications are achieved. One can argue this is just protecting the people that make the specifications and pay the bills. The second side is technology. To maximize the inherent reliability of the system, the judicious use of new and proven technologies in the design must be encouraged. The next chapter (Chapter 4) will provide the technology thrust for modular robotic systems.

The main purpose of this chapter is to fill the gap in the robotic design process by showing how R&M tools and techniques can be applied to the problem of robot design. Thus, this chapter is devoted to examining those techniques that help designers include reliability and maintainability concepts during their designs. It discusses the meaning of reliability and how reliability is included during design. This chapter also presents and discusses the history and application of reliability concepts during the design of systems from other domains such as aircraft and electronics.

3.1.1. Inherent vs. Operating Reliability

Part of the responsibility of the designer of a system is to minimize the chance of failure over the entire life of the system. Therefore, it is helpful to examine the components of the system reliability and their contributions toward the entire life of the system. The operational reliability of a system can be thought of as having two components: inherent and operating [107]. The inherent component of system reliability is the reliability attainable in the design itself. This is the reliability that is affected by the design parameters selected during the design process. Inherent reliability also includes the manufacture of the system: the quality of the parts used, the processes, materials, and production controls used during the fabrication process. The operating component considers how the system is supported, maintained, and operated. This component is directly under the control of the user, not the designer. It includes the system environment in storage, shipment, and use; the maintenance concept, spare parts, personnel training; and the operation of the system. The designer can influence the operating reliability by designing the system so that it is easy to operate and maintain. This is the goal of good maintainability design: to increase the overall reliability of the system by allowing easy and fast repair of malfunctions and maintenance. An extremely important point is that the reliability inherent in a design will never be exceeded. In other words, the operating component of reliability will always reduce the system reliability, and the reliability of a system will never be improved beyond the inherent design reliability bv changing the operating environment. Mathematically, the system reliability can be considered a serial reliability structure and can be expressed as

$$R_S = R_I R_O \tag{3.3}$$

where R_S is the system reliability, R_I is the inherent reliability and R_O is the operating reliability. This formulation immediately shows the relationship between the operating and inherent reliabilities of the system.

3.1.2. The Designer's Responsibility to the Customer: Maintainability and the Ability to Keep What Was Delivered.

When examining the design process from the systems point of view, the designer discovers that meeting the performance specifications is not all that is required for a "quality" design. Philosophically, one can argue a design can perform a function better than any other alternative, yet still be unsatisfactory because it would cost the customer more to maintain than would a less sophisticated, lesser performing model that is inexpensive to maintain and support. Upon re-examination of the specification process, the designer will discover the need to include specifications to enhance the "operability" of the system after delivery. It means the designer is responsible for designing for the entire life cycle of the system. This realization must take place early in the conceptual phase with the designer taking into account the user's ability to maintain the system, the maintenance concept, spare parts allocations, shipment, storage, etc. The system must be designed from the user's point of view.

A specific item that needs to be addressed at this point is consideration of the user's environment. A system being designed for deployment into space will generate much different requirements and specifications than one being planned for an electronics assembly plant. The first difference the system designer will note is the inaccessibility of space-based systems compared to a system installed on a shop floor. This immediately impacts the level of reliability that must be specified for the system and will drive the maintenance concept of the system. Another difference is the operating environment. A space based system must be able to tolerate high cosmic radiation levels, extremely wide ranges of thermal fluctuations, launch vibrations and forces in the orbital environment. A system on a shop floor will see a much more benign environment in general.

3.1.3. Reliability and the Life-Cycle-Cost

The Life-Cycle-Cost (LCC) of a system is made up of several different components. The first component is the cost of the materials used to build or fabricate the system. Another component is the cost of design (the Research and Development (R&D) costs). This includes the salaries paid to the designers, the materials and services purchased during the design, and the administrative overhead involved in the design. By adding the companies profits to the material costs and amortizing the R&D costs over the time the product will be sold, the price of the product is determined. Up until the early 1970's, the price of the product was generally the only factor considered when purchasing products and equipment. Prior to this time, managers and engineers started to realize that the costs involved in supporting a system over its lifetime can far exceed the acquisition cost of the system. In 1971, the Department of Defense began requiring that the acquisition of major defense system be based on the life-cycle-costs of the system [41].

The LCC of a system includes the acquisition costs described above, as well as the costs of maintaining the system and disposing of it when it reaches the end of its useful life. The maintenance costs of a system include the labor to perform periodic maintenance and repairs as well as the cost of spare parts. Both the labor costs and the cost of spare parts are a direct result of the system's reliability and maintainability. Examples of these R&M related costs can be found in Table 3.1. These costs can be included into the overall cost models in several ways, depending upon the model and which costs are included. There have been many LCC models proposed and used over the years for many different applications. A listing of 10 general LCC models and 14 specific LCC models are presented by Dhillon [41].

Table 3.1. Examples of Reliability Cost Categories [68]

Prevention costs	Hourly costs and overhead rates for personnel: engineers, designers, etc.
	Hourly costs and overhead rates for reliability screens
	Cost of preventative maintenance program
	Per capita cost of annual reliability training
Appraisal Costs	Hourly and overhead costs for R&M evaluations and testing
	Average cost of assembly testing, screening, etc.
	Vendor assurance costs: qualification testing, audits, etc.
	Cost of test result reports
Internal failure costs	Hourly and overhead costs for troubleshooting, repair, testing, etc.
	Replaced parts cost
	Spare parts inventory costs
	Administrative costs
External failure costs	Cost to repair a failure
	Service engineering hourly and overhead costs
	Replaced parts and service kit costs
	Spare parts inventory costs
	Cost of failure analysis
	Warranty administration and reporting costs
	Cost of liability insurance

One model suggested in [41] bears examination for the modular robotic system viewpoint. This LCC model is concerned with estimating the LCC of modules. It was originally for modular electronic systems, but can easily be applied to any system composed of distinct modules. The LCC is defined as

$$LCC = \sum_{j=1}^{5} CM_j - T_v \tag{3.4}$$

where CM_j is the jth cost: $j = 1$ (cost of initial spare modules per system)
$j = 2$ (cost of pipeline assets per system)
$j = 3$ (cost of replenishment spare modules per system)
$j = 4$ (cost of initial modules used per system)
$j = 5$ (cost of module repair per system)

and T_v is the value if the modules at the end of their useful lives. If we can assume that $T_v = CM_2 = CM_5 = 0$ then Equation (3.4) can be expressed as

$$LCC = \sum_{j=1}^{m} C_j M_j \left[1 + \frac{NIS_j}{M_j} + \lambda_j (1 - FMR_j) SL \right] \tag{3.5}$$

where SL is the system life in hours
FMR_j is the proportion of failed modules of type j which can economically be repaired
m is the module types in the system
λ_j is the hourly failures per module of type j
C_j is the cost associated with a module of type j
NIS_j is the initial spares of type j needed per system
M_j is the quantity of modules of type j

3.1.4. Reliability and Maintainability Program Management

One of the best ways for a manufacturer to insure he incorporates the life cycle into his design process is to implement a comprehensive Reliability and Maintainability (R&M) program in the company. This program consists of two distinct components, the overall R&M management emphasis within an organization and product specific R&M programs.

3.1.4.1 Organizational R&M Program Management

To provide for an overall high level of quality to all the products developed by the design organization, an overall approach to reliability management must be adopted. As discussed previously, the operating reliability of a system is dependent upon many factors besides the inherent reliability of the design. The reliability management approach must address all of the contributors to the system reliability, and thus must be promulgated over the entire life of the design, from conception through disposal.

The first step in the establishment of a reliability management approach is the development of corporate goals for reliability [68]. These goals should be established at each organizational level commensurate with the responsibility at that level. Product reliability goals are generally established for each product and are based on either contractual requirements levied by customers or on desired in-house reliability levels. These goals must be relevant, attainable, supportable, compatible, acceptable, and most importantly, measurable. Once the goals are defined based on desired results and payoff, planning must occur to ensure the organization will actively progress towards the meeting of the goals. An important aspect of the goal setting exercise is the leadership required. Total commitment to the established goals must be readily apparent from the highest levels of management to ensure active pursuit of the stated goals. To ensure this management support, reliability policies must issued from the highest management levels and the deeds of management must support those written policies. Policies should be action oriented and supportive of organizational goals. They should also be consistent with other policies, be credible, and authoritative but allowing flexibility, specific to provide unambiguous direction and focus, relevant to the situation, and relatively stable over a period of time [68].

Along with establishing the reliability goals and policies, a company must also establish responsible agencies for various basic reliability-related functions within the organization. The *Handbook of Reliability Engineering and Management* [68] describes the basic reliability related functions listed in Table 3.2. Functions One through Five represent the long term leadership and policy functions required by management. These functions, except for program management, will have an indirect effect on product reliability while the rest of the functions will have a high impact on the product reliability. The ownership of these functions is dependent upon the organizational structure. There are many different organizational forms in use today ranging from functional arrangements as in Figure 3.1 to product arrangements as in Figure 3.2 [43].

The function oriented organization (Figure 3.1) tends to be preferred by companies with long-term stable jobs and by the larger organizations. This structure allows for excellent utilization of resources with high efficiency, provides for even workload distribution, larger experience bases, and direct supervision. However, this organization can cause conflicts to arise between departments and coordination and scheduling problems can occur. This organizational structure is not as flexible when handling new technologies since communication between departments may be difficult and decisions regarding implementation of new technologies are difficult to reach [43].

Table 3.2. Basic Reliability-Related Functions [68]

1. Corporate Planning	11. Design Engineering
2. Corporate Reliability Leadership	12. Component and Materials Engineering
3. Requirements Definition	13. Supplier Reliability Assurance
4. Program Planning and Development	14. Reliability Information
5. Program Management	15. Reliability Methods and Standards
6. Functional Administration	16. Reliability Testing and Evaluation
7. Technical Operations Control	17. Production Reliability Assurance
8. Reliability Facilitation	18. Failure Analysis and Reporting
9. Reliability Analysis and Statistics	19. Field Engineering
10. System Engineering	20. Customer Service

The product oriented structure (Figure 3.2) is much more adaptive and creative than the functional structure since all of the personnel working on a single product (or goal) are combined in the same place and organization. It allows for rapid decision making and the minimization of bureaucracy. Schedules are managed effectively and flow of work is efficient The main problem with this structure is it requires much duplication in manpower and facilities. It also does not allow wide diversification of experience base by allowing personnel to be involved in many products over time. Since this organizational structure is more flexible, it seems to be best in fields where technologies tend to be very fluid and markets are unpredictable [43].

A matrix organization is an alternative organizational structure combining both the functional and product oriented structures (Figure 3.3). This approach can be used to provide effective use of a smaller company's resources and provides great flexibility as products appear and disappear, however, several drawbacks must be noted. The first and probably most important problem is that during a shortage of development activity, a personnel surplus can occur which will increase overhead during these times. Another drawback is that personnel are on loan from the departments to the product managers which can result in personnel having two supervisors as well as diminishing the authority of the product manager compared to his responsibility.

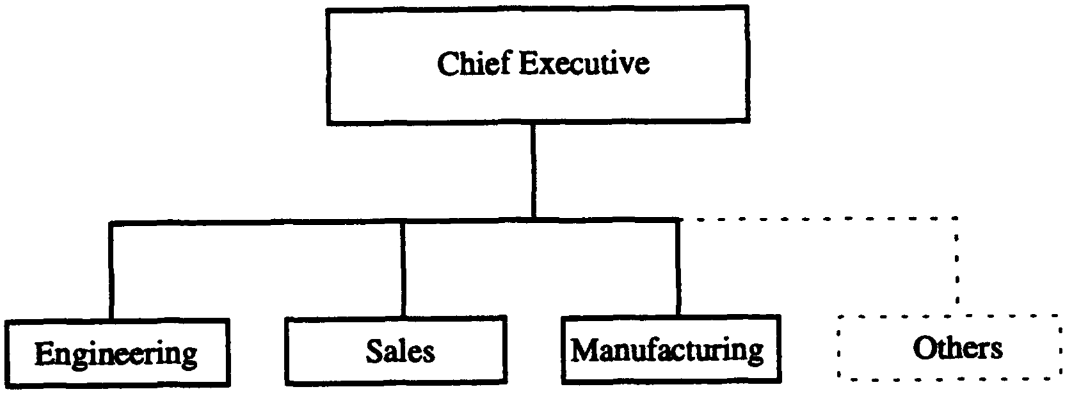


Figure 3.1. Function Oriented Organization [43]

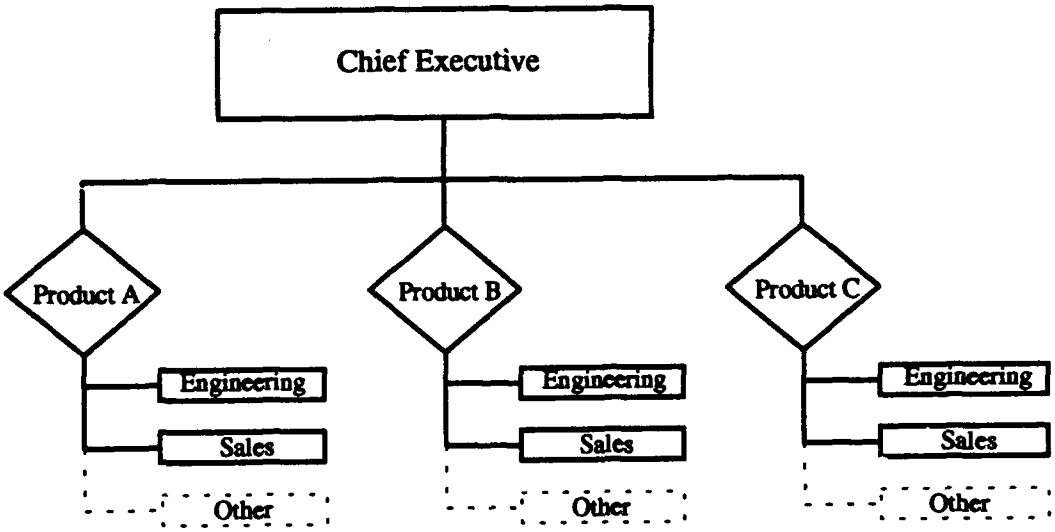


Figure 3.2. Product Oriented Organization [43]

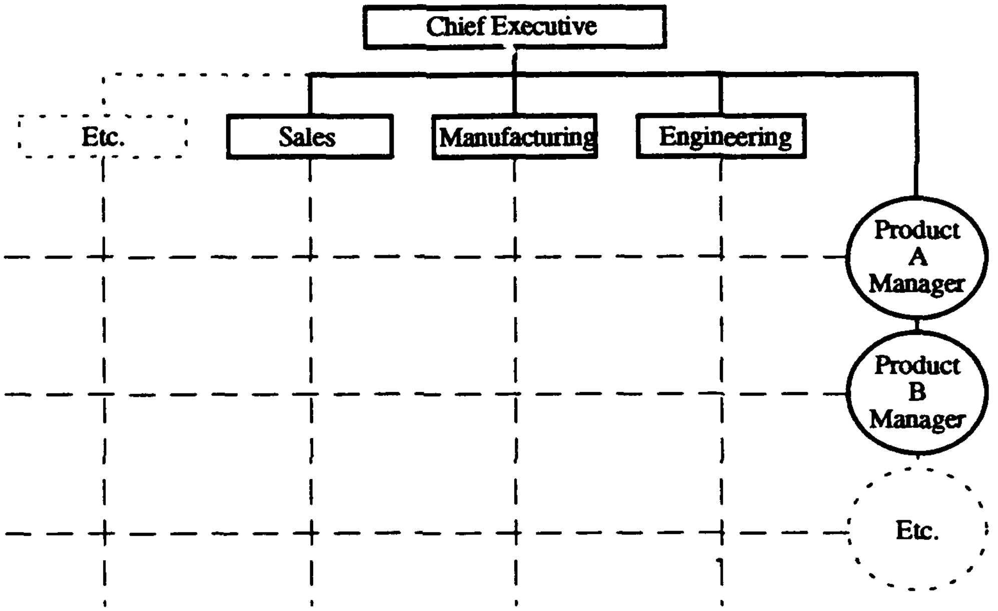


Figure 3.3. Matrix Organization [43]

Many companies and government agencies have adopted "Total Quality Management" (TQM) methodologies which have as one of their central tenets the "empowerment" of the employee. Another way of thinking about this is making the employee responsible and accountable for the portion of any process in which he or she is involved. This approach can significantly effect the reliability and maintainability of a design by ensuring the person(s) responsible for the design is aware of the R&M requirements and takes an active part in the development and satisfaction of these specifications. TQM recognizes the importance of satisfying the customer which is directly tied to the durability of the product. Many texts exist on TQM and how to implement the methodologies. An example is [15].

3.1.4.2 Product Oriented R&M Program Management

The United States Armed Forces have long recognized the need for effective R&M management during the acquisition of new systems and the design of modifications to systems already in service. MIL-STD-785, *Reliability Program for Systems and Equipment Development and Production*, was developed to identify the important elements of a reliability program during the system acquisition phase [104]. Many of these program elements can and should be applied in the commercial environment. These elements are listed in Table 3.3. Maintainability has a similar program document: MIL-STD-470, *Maintainability Program for Systems and Equipment* [102]. These elements are listed in Table 3.4.

The economics of this level of R&M activity must be carefully weighed against the anticipated return on investment. For instance, weighting the importance of reliability to the product, the Apollo space program rated reliability 0.7 on a 1.0 scale, performance was rates 0.25, and price was rated 0.5, while a commercial consumer electronic product is rated at 0.1 for reliability, 0.15 for performance, and 0.75 for price [68]. This shows that a reliability program for the space program was highly cost effective due to the severe consequences of failure, while in consumer electronics, price competition is paramount and the implementation of reliability controls may not be indicated. The economics of reliability are unique for each product, but it is generally accepted that reliability is expensive.

Table 3.3. Elements of Reliability Program from MIL-STD-785B [104]

101 Reliability Program Plan	205 Sneak Circuit Analysis
102 Monitor/Control of Subcontractors	206 Tolerance Analysis
103 Program Reviews	207 Parts Program
104 Failure Reporting, Analysis, and Corrective Action System (FRACAS)	208 Reliability Critical Items
105 Failure Review Board	209 Effects of Functional Testing, Maintenance, and Packaging, Handling, Storage, and Transportation
201 Reliability Modeling	301 Environmental Stress Screening
202 Reliability Allocation	302 Reliability Development/Growth Testing
203 Reliability Prediction	303 Reliability Qualification Test
204 Failure Modes, Effects, and Criticality Analysis (FMECA)	304 Production Reliability Acceptance Testing

The most important part of both the reliability and maintainability program is Task 101, Reliability or Maintainability Program Plan. This is a document that is developed at the start of a project or design that identifies system reliability specifications, a description of each of the tasks listed in Tables 3.3 and 3.4, and how they will be managed. The Program Plan also describes the reliability responsibilities of the designers and consultants and how the reliability efforts will be integrated into the design and into the logistics support analysis tasks. The design schedule is related to the tasks outlined in the program plan. Known and/or anticipated reliability and maintainability problems are identified along with an assessment of each problem and proposed or planned methods of resolution when they arise. The final segment of the plans describe the data sources for the analyses performed and analysis methods. The maintainability plan will also describe the planned or proposed maintenance and support concepts.

Table 3.4. Elements of Maintainability Program from MIL-STD-470 [102]

101 Maintainability Program Plan	203 Maintainability Predictions
102 Monitor/Control of Subcontractors	204 Failure Modes, Effects, and Criticality Analysis (FMECA)
103 Program Reviews	205 Maintainability Analysis
104 Data Collection	206 Design Criteria
201 Maintainability Model	207 Inputs to Maintenance Plan and LSA
202 Maintainability Allocations	301 Maintainability Demonstration

Task 102 for both MIL-STD-785 and MIL-STD-470 is Monitoring and Control of Subcontractors and Suppliers. The necessity of controlling is common sense since if the reliability and maintainability specifications for the components delivered by a vendor for use in a system are not met, the system under design will not meet its reliability and maintainability specifications. It must be noted that this is not strictly an engineering task and management of the subcontractor or vendor must be actively involved in their own reliability program, ɼeliability program reviews, and in the providing of reliability data to the prime contractor.

A task closely related to the monitoring and control issue is Task 102, Program Reviews. The purpose of Task 102 is to require formal design reliability and maintainability reviews at key points during the design of a system. These reviews allow management to ensure that the reliability and maintainability programs are proceeding as expected and that the reliability specifications are being met. A convenient time for the R&M reviews to take place is during the preliminary and critical design reviews. Issues that are addressed at R&M reviews include statistical tasks of the 200 series and the plans and results of reliability and maintenance testing from the 300 series tasks. The entire design teams should participate in these reviews to prevent isolating R&M issues to R&M engineers and to allow crossfeed between design groups and the customers. As mentioned before, these reviews should also be required at the subcontractor and vendor level to insure that the progress of their R&M programs is acceptable.

A very important aspect of customer satisfaction is an impression that a vendor is responding to customer problems and correcting design problems (if necessary) when they arise. MIL-STD-785 Task 104, Failure Reporting, Analysis, and Corrective Action System (FRACAS) is a formal way to address those customer concerns. The FRACAS provides for the reporting of failures that occur during design, qualification and acceptance testing, and parts screening. This task is conducted in conjunction with the 200 series tasks and provides a closed-loop reporting system which requires analysis of each failure and the proposal of corrective action to prevent new occurrences. An additional benefit of FRACAS is it allows for the validation of design modifications made to address failures. An integral part of a FRACAS is the use of a Failure Review Board, MIL-STD-785 Task 105. This board is normally composed of design engineers, reliability engineers, system safety, maintainability, manufacturing, and quality assurance personnel. The board reviews failure trends, failure analysis and ensures that corrective actions are taken. The Failure Review Board can also be the approval authority for corrective actions and the resulting engineering changes.

MIL-STD-470, Task 104, Data Collection, Analysis, and Corrective Action System, is similar to a FRACAS except the emphasis is on the maintenance data and analysis methods. The corrective actions resulting from this analysis will be changes in maintenance procedures and tech manuals. The maintainability data can be used for comparison to maintainability predictions, and for inputs to the customers maintenance planning function.

The 200 series tasks of MIL-STD-785 and MIL-STD-470 are modeling and analysis tasks during the design of the system. The reliability tasks will be discussed first. The data resulting from these tasks should actively be used during the design of the system to enhance the reliability of the system. MIL-STD-785, Task 201 is the Reliability Modeling task. Since the study of reliability is statistical in nature, a mathematical model is the first requirement for reliability evaluation of a system. It is usually best to formulate a simple model to begin with, then as the design evolves, new components can be added to the model. The first reliability model proposed is usually a Reliability Block Diagram with static failure rates assumed. As testing is done, the data for the model is updated and the model becomes more accurate. Reliability models come in many different types: Reliability Block Diagrams, Fault

Trees, Markov Models, Interference Models, etc. For a discussion of the various ways to model the reliability of a system or component, see Appendix A.

Task 202, Reliability Allocation, is performed in the earliest design stages and provides reliability specifications for each component. The overall reliability specification is broken down and allocated to the various sub components according to the reliability model developed in Task 201. A general representation of the allocation problem is a reliability optimization problem where the objective function is

$$f(R_1, R_2, \ldots, R_n) \geq R^* \tag{3.6}$$

and *f* represents the functional relationship between the components and the system, R_i is the reliability allocated to the *i*th subunit, and R^* is the specified system reliability [73]. Once the functional relationship is determined, Equation (3.6) can be solved using dynamic programming approaches, or non-linear optimization techniques.

During the preliminary design, these allocations are usually expressed in terms of failure rates or MTBF, and are subject to the constant failure rate assumption mentioned earlier. The simplest allocation method makes the failure rates additive with the allocations made through historical data trends or engineering judgment [71]. A similar method is equal apportionment of reliabilities [73]. To illustrate, let R^* represent the system reliability specification and R_i the subsystem (or component) reliability. Then

$$R^* = \prod_{i=1}^{n} R_i \quad \text{and} \quad R_i = (R^*)^{\frac{1}{n}}, i = 1, 2, \ldots, n \tag{3.7}$$

These two methods have serious drawbacks when considering redundancy and fault-tolerance in the system, and are invalid when the failure rates are not constant.

Another technique is the ARINC method which also assumes a series reliability structure, constant failure rates, and equal operation time of all components [73]. Equation (3.6) is expressed in component or subsystem failure rates λ_i* and the specified system failure rate λ*, e.g.

$$\sum_{i=1}^{n} \lambda_i^* \leq \lambda^*, \quad i=1,\ldots,n \tag{3.8}$$

A relative weighting of importance of each subassembly can be calculated or estimated from previous failure data. If the past failure rate is represented by λ_i, then a weighting factor ω_i can be determined by

$$\omega_i = \frac{\lambda_i}{\sum_{i=1}^{n} \lambda_i}, \quad i=1,\ldots,n \tag{3.9a}$$

where ω_i represents the relative vulnerability to failure of subcomponent *i* and

$$\sum_{i=1}^{n} \omega_i = 1 \tag{3.9b}$$

Equation (3.8) is then evaluated as an equality to determine the subsystem failure requirements based on the prior failure vulnerability as

$$\lambda_i^* = \omega_i \lambda^*, \quad i=1,\ldots,n \tag{3.10}$$

Another well known allocation method is more sophisticated than those previously discussed but operates with the same assumptions as the ARINC method just examined. The AGREE allocation method [2, 73] uses a relationship between the component and system failure and the complexity of the sub-component or subsystem. This method was originally developed for the allocation of reliability for electronics but has found wide use in all areas or reliability allocation. Fault-tolerance (which is intrinsically determined by the importance of a unit to the failure of the system) can be included in the model by the assignment of an importance factor $\omega_i = P(\text{System Failure} | \text{Subsystem } i \text{ Fails})$. The model also represents the fact that reliability is generally a time dependent phenomenon and changes over time. The allocation formula will provide accurate allocations for those subsystems with an importance factor near one but will distort the allocation if fault-tolerance is high. The allocation formula is

$$R_i(t_i) = 1 - \frac{1 - [R^*(t)]^{N_i / \Sigma N_i}}{\omega_i} \tag{3.11}$$

where t = Mission time or required system operation time
t_i = Time units for which the ith subsystem must operate ($0 < t_i \leq t$)
N_i = Number of modules in the ith subsystem
ω_i = Importance factor for ith subsystem
$R^*(t)$ = Specified system reliability at time t

Task 203 of MIL-STD-785 is the Reliability Prediction Task. The mechanics of reliability prediction vary with the type of system being designed, i.e. electronic or mechanical, as well as the stage the design is in. Electronic systems are particularly amenable to the Parts Count Reliability Prediction method since the failures in electronics are usually considered to be random with constant failure rates and a large generic data base exists for estimation purposes. Parts count reliability prediction is outlined in MIL-HDBK-217F [96] and uses generic parts failure rates, modified by environmental and quality factors to generate composite failure rates for the system. Equation (3.12) governs the Parts Count Prediction Method.

$$\lambda_{Equip} = \sum_{i=1}^{n} N_i (\lambda_G \pi_Q) \tag{3.12}$$

where λ_{Equip} = Total equipment or system failure rate (failures/10^6 hr)

λ_G = Generic failure rate for the *i*th generic part (failures/10^6 hr)

π_Q = Quality factor for the *i*th generic part

N_i = Quantity of the *i*th generic part

n = Number of different generic part categories

This method can be used for mechanical design using generic mechanical failure data such as found in the *RADC Non-Electronic Reliability Notebook* [129, 131]; however, mechanical systems are highly susceptible to changes in the environment and usage, and do not have constant failure rates. This can cause large errors if the environment the data was obtained in is different from the environment for which the prediction is being made. Other methods, such as those used in the *Handbook of Reliability Prediction Procedures for Mechanical Equipment* [23], develop failure models for the mechanical components to estimate a failure rate that can be used in Equation (3.12).

MIL-HDBK-217F also contains formulas used to perform a Part Stress Analysis Prediction. This method is specifically for electronics based on the thermal and electrical stresses experienced by the components in a circuit. This method generates a basic part failure rate that is used in place of the term $(\lambda_G \pi_Q)$ in Equation (3.12). Similar methods can be used on mechanical systems but the failure rate assumption again is invalid. Stress-Strength Interference theory can also be used to predict the reliability of mechanical components if the stresses on the components are known. See Appendix A for a description of Interference Theory and its application.

It must be noted that prediction methods are only statistical estimates of failure rates and the actual failure rates achieved will not have the same values [71]. Care must be exercised when using these numbers for design. A good philosophy to follow is reliability prediction during design should only be used to choose between design alternatives. Testing must be accomplished to ensure the satisfaction of the design specifications. Reliability predictions will give a hint of the level of reliability that may be achieved, but should not be relied upon to support values for warranties or specification satisfaction.

MIL-STD-785 Task 204 is the Failure Modes, Effects, and Criticality Analysis (FMECA) (Also called the Failure Modes and Effects Analysis (FMEA)). This analysis is a bottom-up failure analysis technique that provides the effect of low level failures of single components on the system level. This is a standard analysis performed during reliability analysis of systems and is described in detail in MIL-STD-1629. The methodology is described in Appendix A. The information provided by the FMECA includes percentages of component failures attributed to each part failure mode, the effect the failure will have on system capability, indications of failure (a great help in the development of troubleshooting and maintenance manuals), fault isolation procedures, and corrective actions. This analysis is particularly valuable during the design process since it identifies possible catastrophic failure modes and identifies single point failures. It can be performed at any stage of design and can have a tremendous impact in the early design phases since it can point out bad designs at a very early point. The difference between FMEA and FMECA is that FMECA also quantifies the relative criticality of the failures with respect to system operation.

Sneak Circuit Analysis (Task 205) is a complex analysis task that is applied to electronic circuitry to determine if hidden faults or functions exist in the design that can limit reliability or performance. A complete set of engineering drawings and schematics are required and it is the most time consuming analysis in MIL-STD-785. Since it can only be effectively applied towards the end of the design process, any recommended design changes are very expensive to implement [71]. It is advisable to perform this analysis on the critical paths identified in the FMECA or where safety critical functions are involved. Sneak circuit analysis is not especially germane to mechanical systems. However, when seen from the system level; mechanical failures may have unexpected results and will impact the safety and reliability of the system. In non-electrical cases, the system is examined for sneak paths, or conditions of the system that will allow unplanned states to be attained. These paths may also be present in software, where the code allows unexpected data transfer or operations [68]. The recommendation is that for any critical path, electrical or mechanical, a sneak circuit analysis is appropriate.

Task 206, Parts/Circuits Tolerance Analysis, is also aimed specifically towards electrical components. This analysis consists of simulations to alter the operating environment of the electrical components and using known data on the temperature behavior of the device, the performance specifications are examined to see if degradation occurs. A good mechanical design will automatically include thermal expansion tolerances in the design and match the materials appropriately.

As alluded to in Section 3.1.1, the inherent reliability of a system is dependent upon the quality of the design and of the components used in that design. MIL-STD-785, Task 207, Parts Program, specifically addresses the issue of standardized quality parts. It requires the selection and use of a standard parts list that is used by all contributors to the design. The advantage is that a consistent level of reliability can be achieved in the parts used in the design. MIL-STD-965 is an excellent guide for establishing and conducting a parts control program.

A task that can help the design team to improve the design is Task 208, Reliability Critical Items. This is a list of projected high failure rate components from the reliability prediction task as well as those with high logistic impact from the FMECA. Program management can place emphasis on the design of these components to remove them from this list, thus improving the overall reliability or supportability of the system. Task 209, Effects of Functional Testing, Storage, Handling, Packaging, Transportation, and Maintenance, provides data to the designers on the maintenance aspects of the system. By providing information on the durability of the system during testing and logistical handling, design trade-offs can be accomplished to increase the long term reliability of the system. Packaging and handling concerns, such as static sensitivity, should be addressed. This is the only reliability task that addresses the effect of non-design related problems while design is still in progress [71].

Many of the Maintainability Design and Analysis tasks of the 200 group in MIL-STD-470 correspond directly to tasks in MIL-STD-785 and should be carried on concurrently. The same data is required and the same type of procedure is followed. Planning for concurrent task accomplishment allow the efficient completion of all the reliability and maintainability tasks [71].

MIL-STD-470, Task 201 is Maintainability Modeling. Maintainability models are usually based on procedures found in MIL-HDBK-472, *Maintainability*

Prediction [99]. Five different modeling and prediction techniques can be found here, the simplest being the Mean-Time-to-Repair (MTTR) prediction model (Procedure II). This model uses average task completion times to repair the failed component. This model is shown in Equation (3.13).

$$MTTR = \overline{T}_P + \overline{T}_{FI} + \overline{T}_{FC} + \overline{T}_A + \overline{T}_{CO} + \overline{T}_{ST} = \sum_{M=1}^{m} \overline{T}_M \tag{3.13}$$

where $\overline{T}_P$ = Average preparation time
$\overline{T}_{FI}$ = Average fault isolation tin.e
$\overline{T}_{FC}$ = Average repiacement time
$\overline{T}_A$ = Average alignment time
$\overline{T}_{CO}$ = Average checkout time
$\overline{T}_{ST}$ = Average startup time
$\overline{T}_M$ = Average time of the *M*th element of MTTR

Other maintainability models include thc synthesis of repair time distributions for modular replacement policies (based on testing or design estimates), downtime estimation, preventative maintenance estimation, and parameter estimation based on time standards for elemental maintenance actions. The selection of this model is based upon the data available to the designer and the complexity of the system being designed. The models can be built incrementally, but all must be representative of the tasks required to maintain the equipment.

The Maintainability Allocation Task (Task 202) can be performed in much the same way as the reliability allocations discussed previously. The model of Equation (3.13) can easily be used to partition the MTTR goal of the system to each subsystem and component. The allocation can also be accomplished by level, i.e. system, assembly, subassembly, and component. The difficult portion of maintainability allocation involves setting the system maintainability specification or goal. The maintainability specification may be on the MTTR for each assembly level or can be expressed as a ratio of mean maintenance hours to operating hours (MMH/OH). This second method of specification implicitly specifies a mission length and represent the maximum mean time that can be allowed at each level of maintenance [71]. MIL-HDBK-472 also contains conversions between the different types of specifications. An example of this is expressed by Equation (3.14) [71].

$$MMH/OH = \frac{MTTR \cdot C \cdot F}{MTBF} \tag{3.14}$$

where C is the maintenance crew size and F is the operation service ratio.

Task 203, Maintainability Prediction, is used to determine if the design will meet its maintainability goals. These predictions, based on the model developed in Task 201, should be used during the design to improve those areas of the design that do not meet the established goals and allocations. The frequency of maintenance actions is directly related to the reliability of the components (implied by Equation (3.13)), which means this task should be performed concurrently with the reliability prediction task from MIL-STD-785.

MIL-STD-470, Task 205, is the Maintainability Analysis Task. This task identifies maintainability features of the system which allows the system to meet its maintainability goals, evaluate the design alternatives, and provide inputs for the maintenance planning process [71]. Part of this analysis consists of the development of design criteria to help meet the maintainability goals (Task 206, Maintainability Design Criteria). These criteria can include requirements such as modularity, tool kit standardization, connector and fastener specifications, and procedure standardization. Other criteria may be access and clearance specifications, prohibitions against practices that require scheduled maintenance, etc. It is important to note that this task will be a standard part of any good design methodology for any type of system or equipment. A good practice is to consult maintenance personnel with preliminary designs to determine similar criteria and to judge the current criteria.

The last maintainability design and analysis task is Task 207, Preparation of Inputs to the Maintenance Plan and Logistics Support Analysis (LSA). The results of the maintainability analyses are important to the development of the maintenance and logistics planning for the support of the system. Knowledge of projected failure rates and maintenance concepts allow advance planning for spares production and maintenance facility development.

The last portion of MIL-STD-785 and MIL-STD-470 is the Evaluation and Testing Section (Task Section 300). There are four reliability related tasks and one maintainability task in this section. MIL-STD-785, Task 301 is Environmental Stress Screening (ESS). This is a non-destructive production screen that applies temperature and vibrations screens at the part and sub-assembly level to stimulate failures of latent defects such as weak components and workmanship defects which would fail later in the field. ESS does not increase the inherent reliability of a design, but allows the inherent reliability level to be reached by removing the "infant mortality" problems from the population. ESS is generally considered for application only to electronic components such as piece parts and circuit card assemblies with only a limited application to mechanical components.

Reliability Development/Growth Testing (MIL-STD-785, Task 302) is the testing one generally sees on television in the automotive commercials promoting how well the cars are tested; slamming doors, shaking the suspension, etc. The object of this task is to test a production item to failure and, using a test-analyze-fix process, modify the design to withstand the unexpected failure modes found during this testing.

The testing used to verify if the system meets its reliability specification is called Reliability Qualification Testing (RQT) (MIL-STD-785, Task 303). If an assumption of an overall constant failure rate can be applied to a system, MIL-STD-781, *Reliability Design Qualification and Production Acceptance Tests: Exponential Distribution,* can be used to test for the reliability specification. This is usually performed by an independent auditor if contractual constraints exist. After acceptance of the design, usually by passing RQT, Production Reliability Acceptance Testing (PRAT) (Task 304) is used to insure that the design continues to meet its reliability requirements. This can also take the form of statistical process control [60] if PRAT is not required by a contract.

A test similar to RQT is required by MIL-STD-470, Task 301, Maintainability Demonstration. This task is a demonstration that the system physically can be maintained. This test induces failure into the system and tests if the system can be repaired using only the technical manuals and support equipment that will be available to the technicians responsible for maintaining the system. This is a true operational test of the system since all aspects of the design will be exercised during this demonstration.

While this discussion has been long and rather tedious, it is necessary to understand how each portion of a product reliability program fits together and the overall program should be managed. Not all of the tasks described are necessary to the successful design of a system, however, careful consideration of the tasks will lead to a better design process and insure R&M is given proper consideration. One should not ignore the possibility that the programmatic aspect may be overburdening the designers with too much bureaucracy. The emphasis should not be to protect the program office (although this is important) but to design and build the highest quality system possible within the economic constraints.

3.2. Reliability and the Design Process

Up to this point we have discussed R&M philosophy and design in generalities. In this section, we will present a general design methodology and show how reliability and maintainability concepts can be addressed during design and some of the impacts R&M constraints can place upon a design.

3.2.1. The Engineering Design Process

It has been argued that design is a structured endeavor with a very specific process involved [49, 50, 117]. Pahl and Beitz identify four main phases of the engineering design process: Problem Clarification, Conceptual Design, Embodiment Design, and Detail Design [117]. Aslaksen and Belcher split the design process into five phases: Definition, Analysis, Design, Implementation, and Verification [10]. One can immediately see correspondences in the design processes; what is important is that a methodology exists. A method of design insures that important design related questions are asked and implications of designs considered.

Problem Clarification or Definition refers to the process a designer goes through to clearly establish a problem statement. The specifications are developed along with ways to test if the specification is satisfied. Conceptual Design considers the functions required by the system and if the functions are physically possible. Financial and economic analysis are performed at this point. Functional diagrams can be made and tested to see if the requisite functions of the system can be accomplished. Once the functions required are decided upon, solutions that provide the specific functions are then examined for suitability and implementation. This moves the design into the Embodiment phase where the solution variants are integrated into an overall system configuration. Once the system configuration is decided, Detail Design proceeds to firm up the individual components and interfaces as well as the manufacture of the system.

It does not really matter which particular brand of design methodology one subscribes to; the important point is reliability and maintainability must be ranked equally with cost, schedule, and performance to allow for the development of a quality product. To insure this is taking place during the design process, reliability and maintainability specifications must be established at the same time as the other system specifications and the appropriate design penalties must be established for violating R&M constraints.

3.2.2. Specifying R&M and R&M Measures

Specification starts with a need [117]. A good, reliable design must start with a thorough understanding of the customer's needs that have been translated into clear, well-defined requirements the designers can understand [68]. Specifications are used as the primary communication tools from the user to the designer, and allow the designers to measure how well the design will meet the customer expectations. A danger implicit in this statement is that one must insure the specification is actually what the customer desires; that it adequately represents the constraints the user has defined for the system.

3.2.2.1. Specification Characteristics

It is a good idea to first illustrate some desirable characteristics of specifications and the uses of these specifications. Aslaksen and Belcher [10] state that a specification must fulfill two conditions, one of content, and one of style. The content condition is that the specification must be complete and treat all areas of concern of the user. The style condition is that the specification must be understandable to the designer, allowing a clear conveyance of ideas from the user. Pahl and Beitz also have given some characteristics of specifications: they must be differentiated between demands (which must absolutely be satisfied) and wishes (which should be considered on an economic and functional basis) [117]. Requirements (demands) should always be quantified in some way to provide for testing to determine if the demand is satisfied. Related to being quantifiable, a specification must also be measurable since unless a specification can be tested for satisfaction, it does not have a use. Also, any special procedures, important influences, or intentions must be clearly defined and explained. These characteristics are summarized in Table 3.5.

Table 3.5. Characteristics of Specifications.

Clearly stated and very specifically defined, treating all areas of concern
Demands and wishes are differentiated
All requirements are quantifiable
All requirements are measurable
Special requirements and procedures are clearly identified

How do we specify reliability and maintainability? This is a complex question and depends upon the domain of interest. The first question that must be addressed when considering the specification of R&M for a system is the operating environment and intended use. For instance, the R&M requirements for a remotely operated, space based system will be completely different from a easily maintained system on a production floor. Once the environment is understood and the tasks the system will be required to perform are understood, the system definition process can proceed. It is extremely important to understand the environment since the environment and system usage will define the stresses the system will experience and will drive the failure modes.

The second question must be how a failure is defined. This is an extremely important question from two points of view. Contractually, failure must be defined in terms that will prevent misunderstanding during product acceptance testing, and also prevent wrongful charges of failure during testing. From the system point of view, failure must be defined as to how it effects the operation of the system. For instance, the failure of a transducer on a robot arm may or may not cause any degradation on the operation of the system depending upon the redundancy available in the system. However, an error in the controlling software may cause the restart of an executable module or may halt the system altogether. The user might not even notice the former problem while the latter problem may cause a shutdown of the entire process the robot system is being used in. As stated before, this definition is extremely important during the testing phases and must also take into account the effects on service, maintenance concept, and spare parts levels [68].

Once the overall system reliability specification is determined, the required reliability levels must be allocated down to the component level using a reliability allocation method (see Equations (3.7) and (3.8) and associated discussion). This allows the designer of the components and subsystems to have realistic reliability specifications to meet on their design.

3.2.2.2. R&M Specification Methods

The method used for specifying R&M depends upon the assumptions the designer is willing to make in his reliability modeling of the system and in the particular allocation method preferred. The most popular assumption to make is a constant failure rate for all components in the system or that the system failure rate is constant (see footnote for Equation (3.2)). This is equivalent to assuming an exponential time to failure distribution (see Appendix A) and allows the use of Mean-Time-Between-Failure (MTBF) as the measure of reliability. For the exponential distribution, the MTBF is the inverse of the failure rate and is the expected life of the system.

$$MTBF = E[t] = \int_0^\infty R(t)dt \tag{3.15}$$

The use of the exponential distribution is preferred since the mathematics involved in the development of acceptance tests is tractable and is well known. An excellent overview and procedural guide for exponential reliability testing can be found in MIL-STD-781, *Reliability Testing for Engineering Development, Qualification and Production* [103] and its companion document MIL-HDBK-781, *Reliability Test Methods, Plans, and Environments for Engineering Development, Qualification, and Production* [100]. These documents give tests in terms of the goal MTBFs and determine if the system actually meets those goals.

Another similar measure for general distributions is the Mean-Time-To-Failure (MTTF). The mathematical definition of MTTF is the same as for MTBF, however, MTTF must be used for those life distributions that do not have a constant failure rate since the time to failure (the expected life) is not the time between failures. This is a consequence of the Markovian Property discussed in Appendix A, Section A.2.1.2. One must also realize that the value of MTTF and MTBF is dependent upon the underlying time to failure probability distribution. The same value of MTTF will give different reliability levels. For instance, a system having a normal failure distribution will have a reliability at the MTTF of 0.5 while a system having a exponential failure distribution will have a reliability at the MTTF of 0.368 [73]. As you can see, the choice of distribution and reliability measure can make a large difference in the reliability actually predicted.

Another reliability measure frequently used is the failure rate, λ. This measure is usually expressed in units per million hours $(Failure\ Rate/10^6\ hours)$. Electrical systems and components usually are considered to have a constant failure rate, while mechanical systems have failure rates which change over time. Another way of expressing a failure rate is through the use of the *hazard function* (see Appendix A). The hazard function is a representation of the "instantaneous" failure rate at a particular point in time and is derived from the time to failure probability distribution.

Maintainability is generally specified and measured using Mean-Time-To-Repair (MTTR). The MTTR can also be measured probabilistically, using repair distributions. A typical repair distribution is either the log-normal or the Weibull distribution. These distributions do not have constant repair rates since the time remaining on a repair is generally dependent upon how much time has already been spent on the repair. An alternate way of specifying and measuring maintainability is the maximum repair time. Setting a maximum repair time is an attractive method since statistics governing the actual repairs are not needed, and time standards (which are compiled from statistical data) for tasks are used. Other maintainability indices are Mean-Time-Between-Maintenance (MTBM), Mean-Time-Between-Replacements (MTBR), Maintenance Downtime (MDT) and Mean-Maintenance-Time (MMT), Logistics or Supply Lead-time, and Maintenance Administrative Time. These indices can be used to specify and measure specific portions and components of the maintainability of a system [10].

Availability is another frequently used R&M measure and it imbeds both the reliability and maintainability of the system in a single criterion. Availability is a steady state measure of the system operating state. Equation (3.2) is generally termed the inherent availability, while the achieved availability of a system must include preventative and failure related maintenance as well as administrative processing time [10].

One other way to specify reliability is to specify the operability in the presence of failure. The underlying assumption here is that failure will always occur, how does one prevent the failures from affecting system operation. This leads to the realization that systems must be fault-tolerant. In some applications, such as space-based systems, the reliability requirements at the system level are so high that the required apportioned reliability levels in the sub-systems and components cannot be achieved. In these cases, reliability specifications take the form of "tolerate failure." A specification of this type would specify the operational condition and critical sub-functions of the system. The system would then be required to operate, say, in the presence of any single component failure with no degradation, and with a "graceful" degradation to a "safe" system state upon a second component failure. Such a system would be considered "two-fault" tolerant [123]

3.2.3. Reliability-Based Design

The treatment of uncertainty in design is usually through the application of a "Factor-of-Safety." This is a deterministic approach, and while usually producing acceptable designs, severe overdesign can occur. A reasonable alternative to this method is to acknowledge the variability occurring in the world, model it, and base the design methodology on it. This design approach is called probabilistic design. A subset of probabilistic design, is the probabilistic prevention of failure or Reliability

Based Design. An overview of the methodology is presented here. For more in-depth information, the reader is referred to [62, 93, 124, 141]

The classical design criterion can be expressed as [62]

$$\mathcal{S} > s \cdot (SF) \tag{3.16}$$

where $\mathcal{S}$ represents the deterministic component or system strength, s represents the single valued applied load stress, and SF is the safety factor. The equivalent probabilistic design criterion is

$$P(\mathcal{S} > s) = P(\mathcal{S} - s > 0) = R \tag{3.17}$$

where $\mathcal{S} \sim (\bar{\mathcal{S}}, \sigma_{\mathcal{S}})$ is the strength random variable, $s \sim (\bar{s}, \sigma_{s})$ is the applied stress (or load) random variable, and R is the reliability which is the probability of success. Equation (3.17) is the classic statement of stress-strength reliability modeling, i.e. the probability that the strength exceeds the stress greater than a stated reliability level.

The design problem is then formulated probabilistically by developing the design equations and relationships and applying the algebra of random variables to see if the constraints of Equation (3.17) are met. A brief presentation of this subject is made in Appendix A. One of the difficulties of the probabilistic design approach is the complexity of the issue. For instance, to design a solid circular shaft to survive a certain torsional stress, one can use the equation [73]

$$\tau = \frac{2T}{\pi r^3} \tag{3.18}$$

where τ is the shear stress in pounds per square inch, T is the applied torque in inch-pounds, and r is the shaft radius in inches. The design problem is to find the required shaft radius to survive the applied torque stress $T \sim N(\overline{T}, \sigma_T)$ constrained by the allowable shear stress (a material property) $\tau_y \sim N(\bar{\tau}_y, \sigma_{\tau_y})$, such that a certain reliability level R is reached. The design variable is *r*, subjected to the stress T and the constraint on τ which are assumed independent. An unknown that must be assumed is the standard deviation of the shaft radius since expressions for both the mean shaft radius and the standard deviation of the shaft radius must be used. The method of solution consists of finding expressions for the mean stress, $\bar{\tau}_y$ and the standard deviation of the stress, σ_{τ_y}. The equations are then solved such that $P(\tau > T) = P(\tau - T > 0) \geq R$.

This methodology can also be applied to design optimization. Design optimization problems attempt to find the best values of the design parameters rather than just adequate values subject to constraints placed on the problem. The standard optimization problem can be stated as [124]

$$\text{Find } \mathbf{X} = \{x_1, x_2, \ldots, x_n\}^T$$

$$\text{which minimizes } f(\mathbf{X}) = f(x_1, x_2, \ldots, x_n)$$

such that

$$g_j(\mathbf{X}) \leq 0; \quad j = 1, 2, \ldots, m$$

and

$$h_k(\mathbf{X}) = 0; \quad k = 1, 2, \ldots, p \tag{3.19}$$

where x_i is the *i*th design variable, $\mathbf{X}$ is the vector of design variables, f is the objective (or cost) function, g_j are the m inequality constraint functions, and h_k are the p equality constraint functions. This optimızation problem is readily adaptable to reliability allocation problems (See Section 3.1.4.2) as well as probabilistic design problems. The reliability-based design formulation can be stated as *f*(**X**) = *System*

Reliability Function, where **X** = {*Vector of Design Variable Means*} subject to constraints on the means (based upon the design formulation) and constraints on variability (based upon the combinations of the random variables as well as the required reliability) [124].

Probabilistic design methods prevent overdesign by planning for the variability in the design variables. Several problems are inherent in this approach. The first is the lack of good quantifiable statistical data for the design variables. There is a wealth of data available for the determination of material properties, but not for the stresses a system will see. This is due to the uniqueness of applications and configurations among different designs. One good rule of thumb is design tolerances can be taken to be three standard deviations (σ_i = tolerance/3). Another problem lies in mathematical intractability due to the typically nonlinear design equations. It can be very difficult to build the probabilistic constraints of Equation (3.19) as well as evaluating their satisfaction. Many times, the systems can only be solved numerically.

3.3. Design Applications and Lessons

There is extensive experience in designing for reliability and maintainability in many different domains. The purpose of this section is to review this experience and identify "lessons learned" for translation to the robotic design environment. Unfortunately, the lessons learned are always published in generalities and the specific levels of reliability improvement are almost always unpublished, especially for commercial products. Appendix C documents the extensive effort that was made to find reliability data with which to attack this problem. This presented a great handicap for this investigation since we would like to have shown a positive relationship between reliability and modularity. As a result, these design lessons gleaned from the literature do not have the associated documentation in reliability improvement that one would like to see and to be able to say "Yes, this approach did work." However, these tools and techniques described in this section have been almost universally accepted in the reliability community as being valuable to the improvement of reliability during design and operation, and this acceptance allows us to put our faith in the community as to the effectiveness of these tools.

3.3.1. Electronics and Computer Systems

The earliest experiences in the development of reliability theory comes from the development of electronic systems during World War II. In 1947, 70% of Naval electronics were not operating properly. During the war, 50% of all stored airborne electronic equipment became unserviceable before they were even used [3]. As a result, the military services began intensively sponsoring the development of reliability theory and reliability practices. The Advisory Group on Reliability of Electronic Equipment (AGREE) was formed by the DoD in 1952 with its first report published in 1957 [2]. The early emphasis was on testing, resulting in the development of MIL-STD-781, *Reliability Demonstration - Exponential Distribution*, issued in 1967. It was this early devotion to testing electronics for reliability that has generated the wealth of data available for failure modes and rates for electronic equipment.

During this time, new electronic technologies were being introduced, such as the transistor and the integrated circuit. Enormous advances in electronic reliability were made by the introduction of solid state components (vacuum tubes were infamous for their failure rates). The main reduction in failure rates was achieved through the reduction of interconnections and modularization of systems. In fact it was the increased reliability of electronic components coupled with fault-tolerant computer architectures that allowed IBM to build a modular computer [64] (see Section 3.3.1.3).

3.3.1.1. Electronic Failure Modes

Over the years, many different failure modes for electronic components, equipment, and systems have been identified. The earliest involve chemical and mechanical processes and exhibited strong dependence on temperature. Other effects include electrical and thermal stresses. The Arrhenius Reaction Rate Model is an empirically determined rate model which can be used to predict exponential failure rates in electronic devices. A simple characterization of the Arrhenius model is [20]

$$\lambda = Ae^{\frac{-E_a}{kT}} \tag{3.20}$$

where λ is the temperature related failure rate, A is a normalization constant, E_a is the activation energy of the degradation process (a measure of device power dissipation in many instances), k is Boltzmann's Constant, and T is the absolute ambient temperature. The allure of this failure model is the mathematical simplicity as well as a good correspondence to empirical data. This failure rate model is the basis for all of the MIL-HDBK-217 failure prediction methodologies with different variation for each different component part [96]. The Arrhenius model predicts an exponential relationship between the failure rate and the environmental temperature and power dissipation of the device. This allows the statement of a fundamental electronic system reliability guideline: Minimize temperature variations and distributions throughout the devices [20].

Another related failure mechanism is in the thermal behavior of the materials themselves (also known as power breakdown). These failures occur whpn the device temperatures get high enough (on the order of several hundred degrees centigrade for silicon devices) to change the electrical properties of the devices. Silicon devices will generally not operate above 350- 400 °C since the junctions in the silicon disappear (this is called the intrinsic temperature [20]). Thermal expansion coefficients of the different materials used in electronic devices are also of extreme importance. Differences in the thermal coefficients can create mechanical stresses in the components which over time will lead to cracks and fractures in the components. This is especially important when considering bonding materials (both to hold the device in the package as well as in electrical contacts).

A failure mechanism that is power related is known as current breakdown or hot-spot melting. It is well known that the power dissipation in devices is dependent on the current density, the conductor volume and the conductor resistivity. Higher resistivity will cause higher power dissipation which will cause high local increases in temperature. Dislocations in the crystal matrix of the material, impurities, as well as sharp bends in conductors will cause locally high resistivity values. If the thermal dissipation and heat flow is not taken into consideration during the design of the device or circuit, the temperature increases may cause melting of the conductor with catastrophic results.

A well known electrical overstress failure mechanism is known as high-voltage breakdown, dielectric breakdown, or punch-through. This failure mode occurs during periods of high electric field strength when current flows through an insulating layer of material. This mechanism is dependent upon the strength of the electric field and temperature. As the ambient and device temperatures increase, the voltage level required for the breakdown decreases. While this failure is not necessarily catastrophic, it generally causes other catastrophic failures (such as high-current and power dissipation) as a secondary failure.

Other electronic failure modes are more insidious since they take place over the long term. These mechanisms are corrosion, electromigration, and secondary diffusion. Corrosion is one of the most well known long-term failure mechanisms. It will affect electronics in non-hermetic packages as well as exposed wiring and other components. Corrosion is highly environmentally dependent requiring a combination of moisture, dc operating potentials, and chlorine or sodium ions. The time-varying and random nature of the causative elements prevents the development of a usable mathematical model. Corrosion will eventually cause open circuits to exist or cause intermittencies. The objective of corrosion control is to remove one of the causative elements of corrosion such as higher operating temperatures to drive out moisture, grounding out dc potentials, and cleaning to remove salts.

Electromigration is caused by the repeated application of high current densities in conductors. The high current densities cause the metal atoms to move in the direction of electron flow causing eventual open circuits or changes in the properties of semiconductors. This is a common failure mode in aluminum conductors in semiconductor devices. This mechanism is also temperature dependent with a commonly accepted failure model being [20]

$$\textit{Mean lifetime} = AJ^{-n}e^{\frac{\Delta E}{kT}} \tag{3.21}$$

where A is a material and geometry constant, J is the current density, n is a constant, ΔE is the power dissipation, k is Boltzmann's constant, and T is the average conductor temperature. The higher the temperature and lower the current density, the less migration will occur.

Another temperature related long-term failure mechanism is that of secondary diffusion. This mechanism is the changing of the dopant levels of semiconductor devices over time. This causes the number of electron carriers in the semiconductor to change over time which dramatically impacts the electronic properties of the device over time. This is also known as "parameter drift" and while it might not cause a failure as defined by the designers, it does cause performance degradation over time. This failure mechanism generally does not occur at room temperatures; an elevated device temperature (due to high power dissipation) over a long period of time is required [20].

3.3.1.2. Failure Prevention Methodologies

In general, susceptibility to operational failure modes of electronic devices (electrical overstress and thermal expansion problems) can be ameliorated by applying parts derating during the design and parts selection process. Derating is defined as limiting the stress applied to parts to levels that are well within their specified or proven capabilities in order to enhance their reliability [53]. The basic idea of part derating can be best understood from the interference theory point of view. As discussed in Appendix A, the reliability of a component can be calculated by integrating the area of intersection between the tails of probability distributions for a component's strength and stress (see Figure 3.9). As the mean value of stress is decreased, the area under the intersection of the tails decrease and the probability of failure is decreased. Parts derating is the practical application of this principle and is one of the most powerful tools available to increase electronic systems reliability during design. MIL-HDBK-217 presents Part Stress Analysis Reliability Prediction methodologies that can be used to choose the correct derating level for the allocated level of reliability. The manufacturers of electronic components generally provide curves of operating parameters vs. temperature, maximum junction temperatures, and thermal characteristics. This information is used in the MIL-HDBK-217 models to choose the necessary parts. Derating should be accomplished on junction temperature and power dissipation. Careful attention should also be given to thermal design. MIL-HDBK-217 also contains data on failure rate vs. stress for most types of electronic parts which is very useful in determining the level of derating required. This type of data is not available for mechanical systems, however, the same principle can be applied [53]. The difficulty in overdesign in mechanical systems can be penalties in cost performance.

Other issues relating to the reliability of electronic systems are not necessarily design related although they can significantly impact the reliability of electronic systems. Packaging is an extremely influential component of the system reliability which must be addressed during the design process. Corrosion can be inhibited by the use of hermetically sealed packages. Stresses on the leads and packages can be reduced by using leadless chip carriers and surface mounting technologies. Very Large Scale Integration (VLSI) technologies embedding fault-tolerant electronic architectures and built-in-test schemes also greatly enhance electronic systems reliability by reducing parts counts and reducing handling and thermal expansion problems.

The majority of electronic failures occur during the very early stages of the component's life. This is called the infant mortality period. The cause of these failures can be any one of the electrical stress related mechanisms discussed above, however, the reason for the mechanism to occur is a latent defect or a marginally performing component. It is desirable to remove these problems before the component is fielded to prevent infant mortality period failures from occurring. This is accomplished using screening of the devices, components and equipment. MIL-STD-883, *Test Methods and Procedures for Microelectronics* is the generally accepted guideline for the screening of electronics [105]. The screening required is determined by the quality level the device is expected to achieve. These screens are listed in Table 3.6.

The screens selected for a specific component should be tailored to the specific weaknesses exhibited by the part since screening is expensive; it must be applied to 100% of the components produced. The average screening cost for Class B quality level devices is one dollar per device [53]. This is a considerable cost considering the millions of devices that can be produced. The payback can be seen from the defects found during screening. The fallout (components that fail during the screening) is almost 35% of the population [98]. This is 35% less device failures that will occur after fielding of the equipment. Screening is also effective at the circuit card and module level. The screen generally employed at this level is an environmental stress screen (see MIL-STD-785, Task 301).

Table 3.6. Types of Screens for Electronic Devices [105].

Internal Visual Screen	Interim Electrical Parameter Testing
High Temperature Baking	Burn-In
Temperature Cycling	Final Electrical at Ambient
Constant Acceleration	Final Electrical at Max. and Min. Temp.
Particle Impact Noise Detection	X-ray Radiograph
Hermiticity	External Visual Screen

Another increasingly important issue in the reliability of electronic systems is the sensitivity of modern electronics to Electrostatic Discharge (ESD). This phenomenon is the result of static charging of non-conductive materials. This is the same phenomenon that occurs when you walk across a carpet and receive a shock when you reach for the doorknob. Non-conductive materials, such a plastic wrappers, styrofoam cups, parts trays, packing materials, waxed floors, etc. can build up static charges in excess of 20,000 volts. The discharges of static electricity can easily damage sensitive electronic components or cause their malfunction. All electronic devices are susceptible to ESD damage, however, the level of susceptibility varies considerably. The susceptibility is inversely proportional to the age of the device's technology. For instance, metal oxide semiconductor (MOS) technology is much more susceptible to ESD damage than bipolar devices.

Regardless of the device technology, the failure modes induced by ESD range from melting of the junctions (avalanche degradation) and conductors (metallization melt) to the destruction of insulating layers (dielectric breakdown). ESD can even cause fractures in oscillator crystals due to excessive forces induced during a high voltage pulse [53].

As a result, ESD control has become an extremely important topic of concern in the electronics industry, having a direct effect upon the operational reliability of electronic equipment. ESD control should be applied during the design, production and repair. Design precautions include using parts that are less susceptible to ESD damage, building-in ESD protective circuitry, isolating ESD sensitive components from the environment, Faraday Shielding and masking, and special grounding and cabling precautions. The use of less susceptible parts poses a great problem, since as stated before, each generation of electronic technologies becomes more sensitive to ESD damage. Therefore, reliance upon the other preventative measures during design and proper ESD control procedures during the manufacture and repair of the equipment are imperative to preserving high electronic system reliability.

The control of ESD during manufacture, repair, and use is straight forward and relatively inexpensive. ESD control requires protection of ESD sensitive components throughout the lifetime of the system. Effective ESD control utilizes ESD-protected workstations, ESD sensitive component identification, and ESD protective packaging. The ESD workstation is designed to keep the technician, component, and work surface at the same potential so static buildup does not occur. ESD packaging materials provides for the prevention of triboelectric charging, equalizes potentials and charges, and provides shielding from electrostatic and electromagnetic fields through the use of Faraday Cages. Additional information on the implementation of an ESD control program can be found in Reference [53], MIL-STD-1686 [106], and MIL-HDBK-263 [97].

3.3.1.3. Computer Systems Reliability

A special area of electronics worth noting is computer systems. One of the reasons that computer systems are now as pervasive as they are is the acceptance in and reliance on computers that people are placing. One of the main reasons for this acceptance is the level of reliability that has been achieved in modern computer systems. Another reason is the ease with which failures may be corrected since computers are modular.

Environment plays a very important part in the reliability of computer systems. Unlike many electronic systems, the main environment for computer systems is the desktop, usually in environmentally controlled area such as an office. Failure rates for computer systems in harsh environments are significantly less than those in the office environment [86].

Another contribution to the reliability of computers is the architecture of computational systems. The prime characteristic of reliable computers systems is *recovery*. Recovery is the reduction of fault occurrence, detection and correction of errors, and efficient repair procedures [64]. Implied is the resumption of operation without any loss of data. Ideally, the recovery of a computational system is completely transparent to the user. The user will have no indication that a problem has occurred and the system has corrected the problem. Techniques to incorporate recovery are generally covered under the topic of *fault-tolerance* which is usually addressed through the use of different computational architectures and software fault detection and isolation schemes. Modem day computers have large scale fault-tolerant architectures to insure that users of the system are not inconvenienced by problems within the system. Small computer systems, such as personal computers (PCs) do not have the larg? scale recovery techniques built in to their systems. The reliabilitv of PCs is attained through the use of high reliability components and completely modular architectures allowing fast isolation and repair of failed components. As can be seen in Figure 3.4, the overall failure rates of integrated circuits have been falling steadily over the past five years [132]. The modular architecture of the PC has been standardized and allows for easy and fast addition of add-on boards to the systems thus enhancing their capabilities.

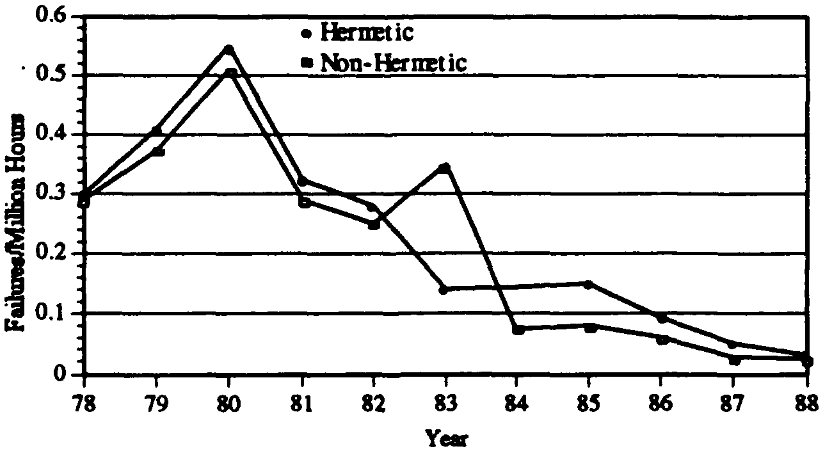


Figure 3.4. Integrated Circuit Failure Rate as a Function of Year [132]

There are two types of modularity: constructional and functional. Good maintainability is promoted by constructional modularity. Constructional modularity is designing a system for easy "snap-in/snap-out" replacement of components. The construction aspects of a system are not directly dependent on the functionality of the system. Functional modularity is the separation of the functions of the system into modules. Constructional modularity does not necessarily imply functional modularity. The tendency has been towards functional modularity, especially since the advent of Very Large Scale Integration (VLSI) allowing large amounts of logic gates in a single device. This allows complete separation of functions. Functional modularity is very dependent upon where the level of functionality is defined. For instance, if we define our function level to be a gate, such as an AND or OR logic gate, we will have an extremely large functional description of the system. If the level is defined at a higher level, such as a clock, the functional description of the system is reduced.

Personal computer systems have enjoyed the advantages of having both high functional and constructional modularity. "he systems are of constructional modular at a high level: the input/output boards, memory boards, disk controllers, disk drives, etc. and functionally at a mid-level: disk drives and controllers are separate modules, etc. This constructional modular structure can be seen in Figure 3.5. This personal computer was recently released to the market and is designed specifically to be modular. All of the system components can be replaced by removing two thumbscrews to loosen the cover and one screw for each component. The disk drives (#1) , expansion card cage (#2), and power supply (#3) are separate modules allowing easy trouble shooting and repair. Specialized functions, such as data acquisition, or special computational boards, are highly functionally modular as well, since the addition of a single module (the expansion board) enables a new, different function. The true value of modularity in this environment is the ability to upgrade new functions or to include new technologies in a module that improves performance overall. As an example, hard disk drives started out with memory densities between 10 and 20 MBytes. Improving technologies have made hard drives possible holding 500 MBytes in five years. The interchange of these drives is fairly simple since each is a self-contained module and dramatically expands the memory capacity of the computer. This is the advantage of modularity the modular robotic system expects to capitalize on.

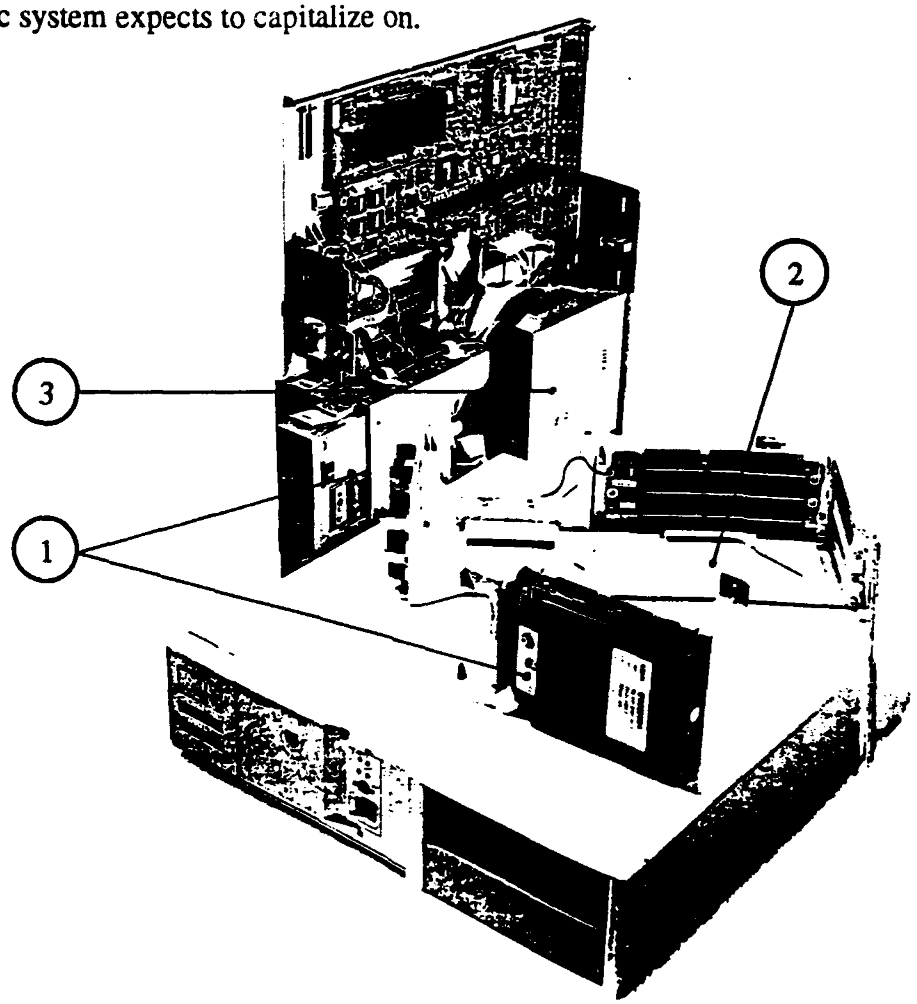


Figure 3.5. Modular Personal Computer [167]

The functional level of modularity found in personal computers can represented as in Figure 3.6. An overall control and timing function is included insuring the proper coordination between the functions. Longbottom [86] suggests the optimal functional modularity would be all of the functions replicated in each module and parallel redundancy employed to give an overall high reliability of the system. The stated reason why this level of functional modularity is not used more is economic. Redundancy is always more expensive and if user satisfaction can be maintained without it's use, then it should not be used.

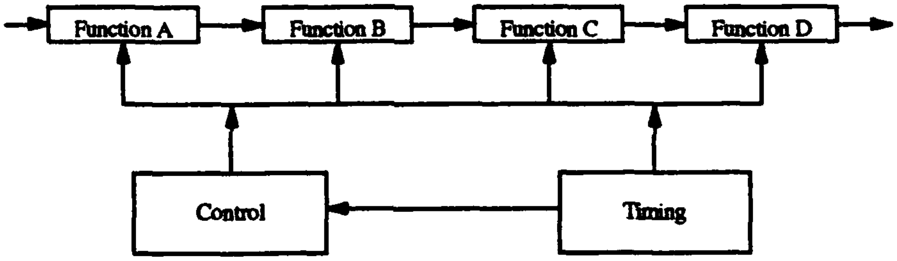


Figure 3.6. Standard PC Functional Modularity [86]

3.3.2. Aeronautical and Mechanical Systems

This section presents current efforts in the quantification of mechanical reliability modeling and failure theories. Historically, mechanical design has centered around a negative approach to design, i.e., the design by failure. The basis for this design has been the development and application of negative criteria such as wear, noise, vibrations, etc. Mechanical design has also not been considered amenable to modular designs since "good" mechanical designs generally seek to minimize parts and promote function sharing in the mechanical structure. What we desire is the opposite: mechanical design in small, modular packages to reduce the system level design burden. This will allow rapid prototyping at the system level and with high reliability modules and interfaces, a high reliability system can be achieved. This is not the typical design approach in robotics, as examined in Section 3.4. These topics are covered to review the current state of mechanical reliability and to present the relationship between the current design and analysis tools to the reliability of robotic systems.

3.3.2.1. Mechanical Component Reliability Prediction

Advances in the reliability analysis and prediction of mechanical systems has not kept pace with the advancement of electronics. There are several reasons for this; the first being the multitude of environments that mechanical systems operate in. Mechanical systems, unlike electronic systems, are subject to friction which causes wear between the mechanical components of the system. Electronic systems generally are not subjected to the physical forces experienced by mechanical systems and do not generally exhibit wear-out. Wear is a time and environmentally dependent phenomenon that is exceedingly hard to quantify. Another problem that wear causes is a non-constant failure rate. The exponential life distribution generally does not apply for mechanical components. Conceptually this does not pose problems, but mathematically, the reliability analysis problem increases in difficulty.

Another difficulty related to the environmental issue is the wide variability in failure data for similar mechanical components and systems. This inconsistency has prevented a generally accepted mechanical reliability prediction method based on failure rates. The reasons for this variability or lack of data are: mechanical components generally share functions and generally are non-standard; a non-constant failure rate requires the recording of all times to failure for all components in addition to operating hours and number of failures, complicating the data collection requirements; sensitivity of mechanical systems to loading, operating mode, and utilization rates; and the definition of failure for mechanical systems is dependent upon the application [23].

These difficulties are being addressed by a project by the Naval Surface Warfare Center's Carderock Division by developing a *Handbook of Reliability Prediction Procedures for Mechanical Equipment* [23]. The proposed prediction methodologies are not based solely on failure rate data, rather they address the problems above by considering material properties, operating environments, and failure modes at the component level. The ultimate objective of the project is to provide a document similar to MIL-HDBK-217 for mechanical components. The failure rate models presented in [23] can be used in system models to generate reliability predictions at the system level for electromechanical systems. The models usually begin from a base failure rate developed from historical data with empirical modifiers for the physical environment, materials geometry, and loading factors as well a different failure modes of the components. Many of these base failure rates are generic and the data presented in Chapter 4 can be used in the Carderock failure rate models. The models included in the handbook are listed in Table 3.7.

Another popular methodology for mechanical component reliability prediction is the use of stress-strength interference theory (See Appendix A). This theory is well suited to mechanical reliability prediction and analysis since it is based on the strength of the component and the stresses the component will experience

[24, 73]. These are design parameters and requirements commonly available in deterministic mechanical design. However, to predict the reliability of a mechanical component, a statistical characterization of the stresses and strengths of the components are required, i.e. the stress and strength probability distributions. The strength distributions are probably available through test data for the materials used in the design modified perhaps by geometrical issues. The stress distributions are chosen based on the design requirements and engineering judgment of the variability. This methodology is good but it is not easily used in conjunction with other methods since it generates a probability of failure, not a failure rate that can be used in a system reliability model. This problem can be overcome by using the probability estimate in an assumed (or estimated) life distribution for the component that can provide a failure rate estimate compatible with the system level reliability model.

Table 3.7. Mechanical Reliability Prediction Model Listing from [23].

Seals and Gaskets	Filters
Springs	Brakes and Clutches
Solenoids	Compressors
Valves and Valve Assemblies	Electric Motors
Bearings	Accumulators, Reservoirs, Pressure Vessels
Gears and Splines	Threaded Fasteners
Actuators	Mechanical Couplings
Pumps	Slider-Crank Mechanisms

3.3.2.2. Mechanical Failure Modes

A failure mode may be defined as the physical process that takes place to produce failure [32]. For mechanical failure, these modes may manifest themselves as elastic or plastic deformation, rupture or fracture, or a change in the material properties. Collins lists 24 of the most common mechanical failure modes. This listing is reproduced in Table 3.8. Most of these failure modes as described in any text on mechanical failure and machine design.

Table 3.8. Commonly Observed Mechanical Failure Modes [32]

Force Elastic Deformation	Wear	Spalling
Temp Elastic Deformation	Impact	Radiation Damage
Yielding	Fretting	Buckling
Brinnelling	Creep	Creep Buckling
Ductile Rupture	Thermal Relaxation	Stress Corrosion
Brittle Fracture	Stress Rupture	Corrosion Wear
Fatigue	Thermal Shock	Corrosion Fatigue
Corrosion	Galling and Seizure	Combined Creep and Fatigue

All of these failure modes require consideration during mechanical design. The handling of failure is implicit in any mechanical design. The normal goal of mechanical design is to prevent the failure of mechanical components from load induced stresses. These failure modes generally include deformations, yielding, rupture, buckling, fracture, and fatigue. The other causes of failure are not as obvious and generally not the first considerations when designing mechanical components. These modes include corrosion, creep, and thermal effects.

While not directly related to robotic systems, the mechanical failure modes and methods for representing loading do present a valid method for determining the reliability of mechanical systems: Stress-Strength Interference Theory (see Appendix A, Section A.2.1.4). These methods for quantifying the stresses of mechanical systems provide the most immediate way of determining the stress levels in mechanical structures. An additional use for the stress and load descriptions presented here is shown in Section 3.5.

Mechanical element design usually begins with a load to be supported of carried. This load then directly determines the minimum strength the member must possess. The design is usually based upon the maximum allowable stress levels within a component. The methods for characterizing mechanical stress levels during design can be found in any machine elements text such as [138]. The design of mechanical components to withstand a maximum stress level is well known, but not of great relevance to the modular robotic reliability problem, what is of interest are the tools that have been developed to retain the inherent reliability of the design.

3.3.2.3. Reliability Centered Maintenance (RCM)

One of the most immediate impacts to reliability when considering aerospace systems is one of scale. These systems are extremely complex and always use the newest technologies to provide the cutting edge in performance. To use these complex systems economically and safely, these systems must be highly reliable and maintainable. The only way to achieve the high R&M required by these systems is to use a total life-cycle program incorporating well planned R&M programs and technologies. Another differentiating characteristic, and perhaps the most important one is the planned life of the system, lasting perhaps decades. This long lifetime requires the system to be maintained to allow for the proper operation of the system over its life. An aircraft is a much more complex system than a robotic system, however, the planned life of a robot installation can be measured in decades in some instances. The expected life of the robotic system makes periodic maintenance a necessity and for the maintenance planning and implementation, a proven technique for the prevention of failure called Reliability Centered Maintenance (RCM) can be used.

One of the main contributions of aerospace to R&M technologies, besides driving the development of reliability theory through improving electronic avionics systems, has been the concept of RCM. The focus of Reliability Centered Maintenance is on maintenance planning and the prevention of failure. It is a decision logic that focuses on the consequences of failure and the actual preventative maintenance tasks. This logic process considers the maintenance tasks related to three conditions: 1) Hard Timed Replacement (HTR), where performing replacement or maintenance functions will prevent a failure, 2) On-Condition Maintenance (OCM) where degradation prior to failure can be detected through inspections, and 3) Condition Monitoring (CM) where degradation is detected in time to prevent failure through sensors and data analysis [7].

The RCM logic process follows four major steps:

1. Performance of a FMECA to identify critical components and end-items.

2. Apply RCM logic to select the optimum combination of HTR, OCM, and CM maintenance task requirements or to determine if redesign is needed.

3. Implement the RCM decisions through specific maintenance tasks and develop the data needed for logistic analysis.

4. Collect and use actual hardware failure data to optimize the RCM process during the life of the system.

RCM recognizes the relationship defined in Equation (3.3) and that maintenance should be performed on critical components only when it will prevent a decrease in reliability and/or performance or when it will reduce life cycle costs. The logic should be applied during the design process to help define the maintenance concept and maintenance requirements for the system.

RCM involves the application of a logical analysis to select the CM, OCM, and HTR maintenance tasks that will be most effective in preventing the system's significant part failure modes. This is *not* the development of a fixed preventative maintenance schedule. The first step in a RCM analysis is the identification of the components that are critical in terms of mission safety and operating system. These are known as the *Maintenance Significant Items*. This is done through and concurrently with the Failure Mode, Effects, and Criticality Analysis (FMECA) described in Appendix A. Based on the system description, a determination is made on the feasibility and desirability of maintenance addressing the critical item. The objective in this analysis is to reduce the scheduled maintenance burden, to eliminate excessive support costs, and to preserve the inherent level of reliability present in the system's design. The actual RCM decision process is dependent upon the domain of the system to which the methodology is applied. For instance, the use of RCM for structures depends heavily on visual inspection to determine the condition of the system, rather than measurement monitoring. The result of the RCM program is a consolidated, fully documented maintenance plan. An additional requirement is that the condition and reliability monitoring of the system must be performed throughout its entire life cycle.

3.3.2.4. Damage Tolerance Analysis (DTA)

One well known problem dealing with the life times of mechanical components and their failure modes is with the ability to detect defects in the components that can lead to failure. Some of these defects are cracks, deformities, stress concentrations, etc. Damage Tolerance Analysis (DTA) is an analysis procedure based on fracture mechanics and the strength of materials that develops inspection and maintenance procedures for structural components. The methodology is mainly applied to aircraft structures, but can be applied to any complex structure subject to fatigue.

The basic assumption of DTA is structural imperfections will always exist. The objective of DTA is to achieve a fail-safe structure by insuring a slow, controlled crack propagation rate in the structure [118]. DTA uses either deterministic or probabilistic crack growth models based upon structural models of the system under consideration. The structural model is usually a finite element model. This structural model is used to generated stress intensities throughout the structure which are used to predict possible crack growth rates thus determining rigid inspection intervals.

3.3.3. Nuclear Power Plants and Power Systems

3.3.3.1. Reliability and the Nuclear Power Generation Industry

To date, there have been two nuclear power plant accidents of note: the Chernobyl disaster in the former Soviet Union, and the Three Mile Island Incident in the United States. The world has a reason to be concerned about the safety of the nuclear industry, considering the possible results of a nuclear plant accident as seen in at Three Mile Island and Chernobyl. As a result, the nuclear industry has the most stringent reliability requirements found outside of the space industry. These requirements are the result of the need for extremely high levels of safety which is directly related to the level of reliability of the nuclear power station. The main focus of the nuclear industry has been on a preventative approach to safety and reliability through the stressing of remedial actions in both the design and operational phases of the plant's life to identified safety and reliability problems. The process is based on life-cycle cost and is oriented towards the use of existing reliability engineering and management techniques [82].

While no R&M techniques are specific to the nuclear industry, it is important to note the effort which is expended after the plant is in operation to preserve the reliability inherent in the plant's design. This is represented by the intensive failure data gathering effort that is required during operation (on pumps, fittings, control systems, electrical components, etc.) and the associated Failure Reporting, Analysis, and Corrective Action System that must be used by all nuclear power plants. Also notable is the effort taken to prevent single point failure from causing accidents. The preferred tool for this design problem is Fault Tree Analysis (See Appendix A).

During the design of a nuclear power plant, the following steps are specifically taken to insure a high inherent reliability level [82]:

1. Parts Selection. Standard high quality parts are specified and used during the construction and repair of nuclear plants. This applies both to the electrical and mechanical components of the plant. Table 3.9 presents a list of some of the factors used in parts selection.

2. Screening Application during Production and Repair. One Hundred percent screening is usually required on all electronic components and many selected electromechanical and mechanical components. These screens are applied according to MIL-STD-883 and MIL-STD-785 for electronic components and according to the latent failure mechanism removal in all other components.

Table 3.9. Part Selection and Control Factors [82]

Does part failure mode impact plant safety (from failure mode studies)?	Does part have a short replacement life?	Does part have long procurement lead time?
Does part require qualification testing?	Is the part a high cost item?	What is the required failure rate?
What derating factor is required (from reliability allocation)?	Is burn-in (or other screens) required to achieve the failure rate?	Is MIL-STD part available from a qualified vendor?
What is the normal delivery Cycle?	Will part be available throughout plant's life?	Is there an alternate standard procurement document?
Are existing procurement specs available?	Is part procurement spec necessary?	Are multiple sources available?

3. Parts Derating. This is the selection of parts for operation at conditions less extreme for which they are designed. Depending upon the class, electronic parts are derated from 45% to 80% of the rated stress (or load) level depending upon the level of reliability required from the part. Derating is also applicable to mechanical and structural components which are usually designed to meet the worst case stress the component will experience. This is another way of expressing the factor of safety approach described by Equation (3.16). In these cases, probabilistic modeling can be used to determine the part strength required for a specific level of reliability.

4. Diagnostics and Modularity. As noted in Section 3.3.1.3., the maintainability of equipment is directly related to the ease with which a fault is isolated and repaired. Diagnostics are used to effectively and quickly locate and isolate a fault condition while modularity allows rapid replacement. The level of modularity here is chosen with cost-effectiveness criteria (see Section 3.1.3). If the repair cost of a module is greater than the cost of a new unit, the module should not be considered reparable and should be discarded. The use of disposable modules causes an increased supply burden since replacement modules must always be available. The size and cost of diagnostic features increase rapidly as the capability for fault-isolation extends down throughout the system. Interconnections should always be minimized and the replacement level should be as high as possible. The Advanced Liquid Metal Reactor (ALMR) planned by the US Department of Energy is to be a modular system, consisting of 9 reactor core modules each with modular cooling and safety systems to provide for additional assurance against failure. In this case, modularity provides for

quick and easy restoration of safety systems as well as providing redundancy and superior availability during power generation.

5. Simplicity and Redundancy. These two design guidelines are contradictory, but each are very useful and together can be extremely powerful. In general, reliability is inversely proportional to complexity and an intense effort to simplify a design can significantly increase the reliability of a design as well as providing substantial cost savings. Redundancy improves the system's resistance to failure by providing insurance against failure. The best reliability can be achieved by using simple, reliable base system design and use redundancy to augment the reliability as indicated by the life-cycle-cost models until the specification is achieved.

3.3.3.2. Power Distribution Systems Reliability

The reason for having nuclear power plants in the first place is to generate electric power at reasonable costs to the consumer. The reliability concerns with nuclear plants is mainly safety related, however, a great deal of concern is placed on the availability of electric service to the customer, regardless of the method of generation. The reliability of electrical distribution systems is usually quantified in terms of the generating system unavailability or *Forced Outage Rate (FOR)*. An availability measurement requires a reliability model that will easily generate times of operation and failure. This is easily provided by a Markov model and indeed, the Markov model is the preferred reliability model in the power industry [19].

A power distribution system can be described as a group of interconnected networks providing a great deal of generating availability redundancy but a distribution system with a serial link between generator and the customer. This results in the typical feature of a power distribution system that the customers connected to the system farthest from the supply point tends to suffer the most outages. A singular characteristic of modern power systems are their size. When examined as a whole, these distribution systems are too large to be analyzed using even the largest computer system available today due to the state explosion problem exhibited by the Markov Reliability Model (see the next paragraph and Appendix A). Fortunately, these systems can be broken down into smaller pieces and analyzed separately [19]. The pieces usually used are generating stations (nuclear power plants are included in this subsystem group), generating capacity models, primary transmission lines, substations and switching stations, and protection systems. This approach provides us with a good rationale for reducing the large size of the design problem faced in robotic systems by imposing a modular structure on the system. The components of the system must have independent operation, and for serial robotic systems, this is true. The modular approach allows power system engineers to understand and design each component separately, providing a standard interface with other components of the system.

Even at the subsystem level stated above, the Markov reliability models generally have many states. From Appendix A, if the components can only be operational or failed, a Markov model will have 2^n states where n is the number of components is the system. This generally prevents the use of analytic solutions for power system networks and forces the reliability analyst to use approximate methods. Some of these methods include the use of approximate availability equations based upon failure and repair rates of the components in various combinations. Also used are network reduction techniques to create equivalent networks that are easier to solve.

Another technique often used in the power industry is the Failure Mode, Effects, and Criticality Analysis (FMECA) (See Appendix A). The failure modes of power distribution systems are generally known and the FMECA can be developed for each different distribution system. FMECA allows the reliability analyst to determine the critical failure mechanisms within the system and allows the designer to address those problems during the design.

Another significant point is the long standing commitment the power industry has had to reliability. This has been a customer driven concern. The significance of this is that the industry has collected large amounts of data on all aspects of the reliability of power systems as maintained by the Electric Power Research Institute (EPRI). The vast amount of data allows engineers to do a very good job of prediction of the reliability of power systems and makes probabilistic design (see Section 3.2.3) much easier since the life, stress, and strength distributions of all the design parameters for the system components can be determined from the available data.

3.3.4. Software Engineering and Software Reliability

Software are the instructions (or programs) that direct the execution of processors within computers. Software includes the programs and the input and output data [139]. It is the software driving the computational hardware that allows the computer to attain its power and flexibility. The software is also the high cost item involved in computer systems today. Consider a personal computer. One can buy a good system complete with data storage media, graphics drivers and screen, large amounts of memory, extremely fast and powerful microprocessors for around $2,000 - $3,000. Operating and applications software can quickly add up to more than the system hardware. Purchasing the basic software necessary to support system operation such as the operating system, graphical interfaces, word processing, data and graphics, spreadsheets, etc. can cost upwards of $2,000. One state-of-the-art statistical package [135] can cost over $1,000 for the basic system. Computer-aided engineering design packages have comparable prices.

Software design is also revolutionary by nature [94]. Software designers usually start from scratch every time while hardware designs usually evolve from previous designs. The result is that there are few standard ways of performing different functions with few examples to follow. Thus, software development is a manpower intensive activity. This allows for the inadvertent introduction of errors during the development of software. These errors can be of several types [79, 139]. The first is simple typographic errors made during the coding. These errors can usually be found during compilation of programs if the syntax of the language is affected, however, the transposition of numbers, indexes, and branch destinations will not be caught. The next error is logic errors. A logic error is the function of the program not following the specification. This is exemplified with an erroneous branching condition, or a mistake in data manipulation. The most serious error is an error in specification. Software failure is often caused by the requirements and specifications being incorrect. If the software is not designed to the correct specification, the software will fail to meet the correct specification.. Thus, the designer must insure the specifications actually represent what the software will be required to do. This includes input and output data, data manipulation, and data storage.

The main difference between software and hardware is that software does not show any of the effects of aging[2]. This is because software does not degrade over time without outside human intervention. Another fact that should be mentioned is the difference between software errors and software failure. Software errors are described in the preceding paragraph. However, software failure does not occur until code that contains an error (or bug) is executed. If the erroneous code is never executed, the software cannot fail due to that error. This illustrates another property of software: software usually has a modular architecture with sequential execution of the different modules.

Good software engineering practice requires that the program under development be modular [139]. This requirement is imposed to force a standard of interfaces betw ;en program components that makes de-bugging and maintenance of the software a tractable endeavor. The software design process is usually top-down with the top level problem being addressed through sub-problems which evolve into modules. Software can be designed from the bottom-up as well. One can develop a set of software modules that perform specific functions such as file input or output, or data manipulation, that have specific input and output specifications. These modules can then be used as building blocks of a larger software system. The advantage of modular program development is that the functional independence of the modules allow for independent testing of the modules. Upon integration, one

[2] The effects of aging do not include computer viruses.

can then be almost assured that errors are caused by the module interactions rather than the software modules themselves. Modular software reliability models are discussed later in this section and evaluated for modular robotic systems in Section 5.3.3.3. Software reliability enhancement techniques are addressed in Section 4.3.9.1.

As a result, most of the software reliability models are based upon the number of errors in the program and upon the rate of removal of errors from the software [94, 139]. The usual approach is to hypothesize an error density (expressed in bugs per instruction) with an exponential model for error removal. The removal rate is determined from past data or experience. This information can be combined into an expression for the number of bugs remaining in a program, $\varepsilon_r(\tau)$ where τ is the debugging time. The assumption is then made that the probability a bug is encountered in a time Δt after t successful hours of operation is $z(t)\Delta t$. If we let t_f be the operating time until the occurrence of a failure then we can express this probability as

$$P(t < t_f \leq t + \Delta t \,|\, t_f > t) = z(t)\Delta t = K\varepsilon_r(\tau)\Delta t \tag{3.22}$$

where K is an arbitrary proportionality constant. Then $z(t)$ is recognizable as a failure rate (or hazard function). Using the relationship between the reliability and hazard functions in Table A.1 results in the software reliability function

$$R_{S/W}(t) = e^{-z(t)t} = e^{-[K\varepsilon_r(\tau)]t} \tag{3.23}$$

For a fixed value of τ, the hazard rate is constant and Equation (3.23) becomes the exponential reliability function. The values of the constants involved with this model can be calculated as shown in [139]. It is important to note that this software reliability model is completely compatible with hardware reliability models and definitions.

An alternative to the bug removal rate reliability model of Equation (3.23) is a structural model treating software modules much as one treats hardware components [85, 81]. This model is based upon a Semi-Markov model description of program control between modules and the failure rates of the modules themselves. Kubat [81] arrives at a failure rate expression for a software system made up of K programs expressed as

$$\lambda_{S/W} = \sum_{k=1}^{K} \lambda_k - \sum_{k=1}^{K} \lambda_k \left(\prod_{i=1}^{M} (g_{ik}^{*}(\alpha_i))^{a_i(k)} \right) \tag{3.24}$$

where λ_k = the arrival rate of calls of the program k, $k = 1, ..., K$.
$g_{ik}(t)$ = the p.d.f. of the time spent in module i of program k, $i = 1, ..., M$.
$g_{ik}^{*}(\alpha_i)$ = the Laplace transform of $g_{ik}(t)$.
α_i = the failure rate in module i.
a_i = the average number of visits to module i in program k.

Since the model is Markovian, the resulting software system's failure rate $\lambda_{S/W}$, is a constant and also results in an exponential reliability model for the software that is completely compatible with hardware models. The choice of the models presented should be based upon the data available (see distribution selection in Appendix A, Section A.1.2.5). If data on the errors in the code is available, a code based reliability model should be used. If only the specifications (functional and interface) or failure rate information is available, the modular model is the more appropriate choice.

3.4. The Robot Design Process and Reliability

As described in Sectioꞑ 3.2.1., the methodology of design can be partitioned into phases: Problem Definition, Conceptual Design, Embodiment Design, and Detail Design. These phases can be correlated with specific tasks that are required during the design of a manipulator. The Problem Definition stage will include a definition of the task the robot will perform. The robot task will typically include a specific trajectory and orientation through a required workspace along with points where manipulation of an end-effector will be required. The maximum planned load of the work being performed will generally be known, since the requirements of the process itself influenced the selection of a manipulator to accomplish the task. If the process the robot will be integrated into has not been fully developed, some knowledge of the task must be known. Based on the precision and speed requirements of the task to be accomplished, the performance requirements of the manipulator must be determined. The physical characteristics that need to be determined are payload, work volume, accuracy, repeatability, speed, etc. It is also at this point in the design that R&M concepts must first be used. The first concept is the addition of specifications to address reliability and maintainability. A system reliability specification depends upon the operational environment of the manipulator. The maintenance concept must be included in this environment. A typical industrial robot can be expected to perform between 1000 and 10,000 hours before failure occurs. The availability can be very high regardless of the actual reliability if the time to repair is small. Most industrial robot manufacturers claim to specify a Mean-Time-To-Repair of 30 minutes to isolate and repair any failure in the robot system. A space-based or nuclear maintenance robot will not have the luxury of repair upon failure and the reliability must be extraordinarily high to have an acceptable availability. Professor Tesar at the University of Texas reports that the nuclear industry currently experiences failure 1 out of 10 deployments in robots for steam generator repair and desires 1 in 20.

Also at this time, reliability program tasks should be planned and initiated. According to Figure 3.7, robot conceptual design includes specification development, kinematic selection, and selection of the types of actuators to be used. The selection of the kinematics also impacts the reliability since the accuracy and repeatability of the manipulator are determined by the variability on the kinematics. As the configuration is being determined, a reliability block diagram should be started with static reliability estimates used to calculate the system reliability. This includes the mechanical and electrical components of the system. Redundancy in both degrees-of-freedom and functions will augment the reliability of the system. Redundancy in the degrees-of-freedom provides the possibility of task completion using fault-tolerance and reconfiguration schemes. Functional redundancy will immediately improve the reliability of the system. In Chapter 2, it was reported that in the early 1980's, the most robot system failures occur in the controller electronics. Redundancy applications for the controller are the easiest to accomplish and can provide the greatest reliability improvement for the effort.

As the design progresses from the conceptual to preliminary design phase, different configurations are being judged for acceptability according to the established specifications. While the meeting of the reliability specifications are important at this stage, a more important use of the reliability models for each configuration is to allow the designer to view the relative improvement in reliability one configuration has over another. This is where the application of FMECA, Fault Tree Analysis, and component reliability prediction models should be applied.

The resolution of the configuration selection problem and functional concept realization are the main tasks of the preliminary design phase. For a highly reliable system, a singularly important design goal should be the elimination of single point failures. Fault-tree analysis is the tool of choice for identifying single point failures and should be applied at this stage. The primary reliability goals during preliminary design are the identification of critical components and design elements and the identification of uncertainties that have the most profound effect on the design [158]. The FMECA is the best way to find the critical components in a design. It also provides a measure of the criticality of the component under consideration and of each failure mode. The procedures for Fault-Tree Analysis and FMECA can be found in Appendix A.

The identification of uncertainties leads directly to the probabilistic design problem and the quantification of the uncertainties lurking in the design. Part of the answer to this question lies in the experience of the designer to fathom the intricacies of the design and how the different functions combine to make the whole. The solution of the probabilistic design problem will give sensitivities to the constraints imposed on the design. A choice at this point could be to alter the specifications if the design goals seem unreachable with the design concepts at hand.

Once the configuration and preliminary design is selected, the design process proceeds to the detail design phase. This is the phase that the final attributes of the design are developed. At this stage, the reliability analysis becomes intense and builds upon the previous models. It is a this point that predictive models, based on test data generated during the detail design phase, are used to insure the specifications are met. The FMECA indenture level is lowered and the FTA is finalized to insure proper consideration of possible fault modes. It is also at this stage that the parts selection and screening programs are implemented. Maintainability tasks are also considered at this point and MTTR specification confirmed.

After the detail design is completed, the final system integration occurs and system testing is accomplished to prove the satisfaction of specification. It is at this time the appropriate reliability and maintainability tests should be performed. The reliability tests include life testing and production acceptance testing as described in Section 3.1.4.2. The maintainability acceptance testing is performed as well. If the specifications are not satisfied, the failures are analyzed for design modifications to prevent the failures from recurring and to investigate the impact of similar failures. This is a follow-on to the FMECA and FTA.

Once the system reaches the production and marketing stage, the reliability and maintainability program becomes concerned with the maintenance of the inherent reliability of the system. This can be done through the implementation of a process quality control program using statistical process control methodologies [60]. This program reduces the production process variability and insures the systems coming off assembly will meet the specifications.

The major tasks and when they should be accomplished are represented in Figure 3.8. A practical note: Many robot systems are initially prototyped to test out the final concept in preliminary design. This prototype can be a very useful tool during reliability analysis. It is easier to get insight into the possible failure modes during the actual operation of a system than through the examination of schematics and diagrams. Data should be collected on component data to aid in the final design of the system.

3.5. Reliability Implications for Modularity: The Assumption of Independence

During reliability analysis and design, one of the assumptions usually made is that of independent life distributions of the components of the system. By independence, we mean that the life of one component does not influence the life of another component. This assumption is made for mathematical tractability and in many cases is reasonable when the causes of failure are random for each component of the system. In this case, reliability can be considered a modular characteristic of the system components or modules. This section considers the implications that the independence assumption has on the modularity problem, and what is required to attempt to provide for a modular reliability model when the components are not independent.

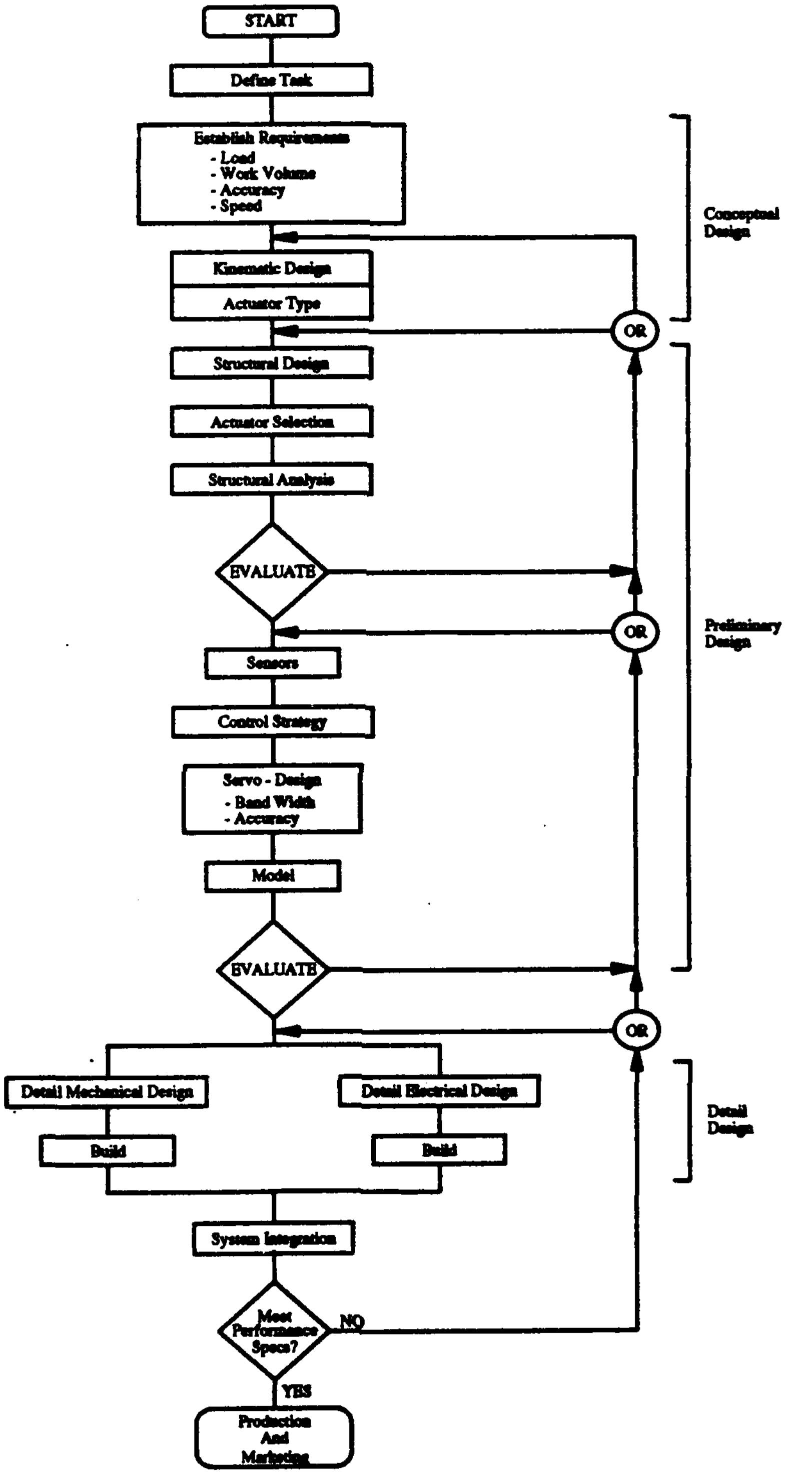


Figure 3.7. Flowchart for Robot Design [6]

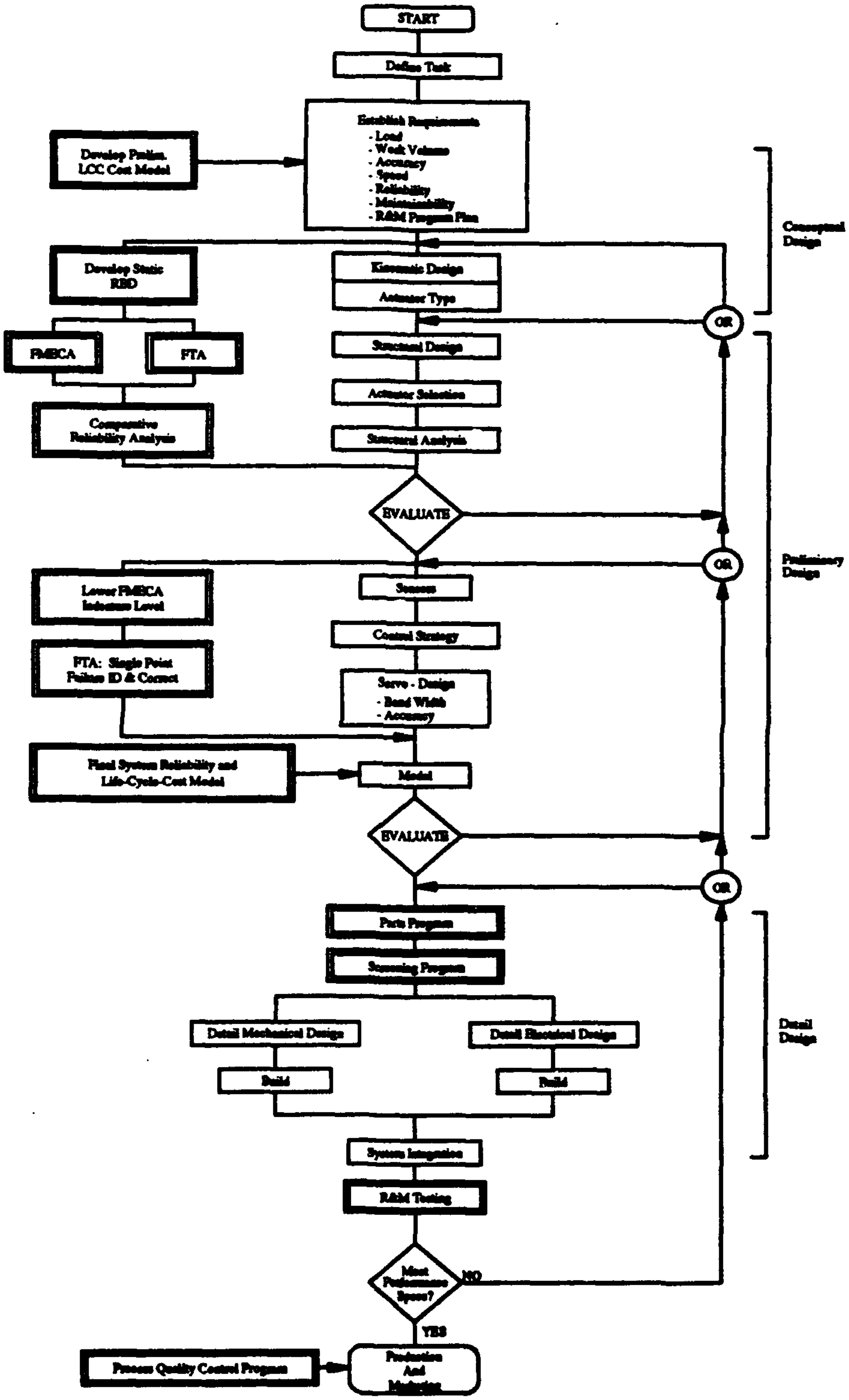


Figure 3.8. Flowchart for Robot Design Including Reliability Tasks

Consider a simple two component system with a series configuration (Figure 3.9).

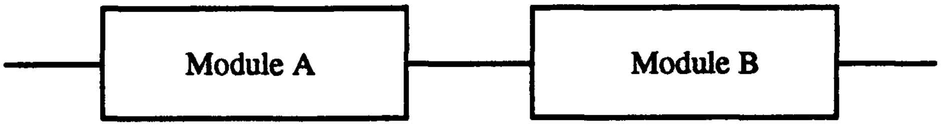


Figure 3.9. Two Component Series Configuration

The probability of survival of the system to time **t** is the intersection of the events of the survival of both components (or modules) to time **t**.

$$P(T \geq t) = P(A \geq t \cap B \geq t) \qquad (3.25)$$

dropping the greater than equal signs for clarity Equation (3.25) can be written as

$$P(T) = P(A \cap B) = P(A)P(B|A) = P(B)P(A|B) \qquad (3.26)$$

where $P(B|A)$ is the probability that the life B is greater than t given that a life of A has occurred for the first component. If the lives of A and B are independent then the realized values of the lifetimes do not depend on the value of the other components and we can write $P(A|B) = P(A)$ and $P(B|A) = P(B)$ and Equation (3.26) can be written as

$$P(T) = P(A)P(B) \qquad (3.27)$$

In terms of reliabilities and generalizing to *n* components

$$R_S = P(E_1)\cdot P(E_2|E_1)\cdot P(E_3|E_1 \cap E_2)\cdots P(E_n|E_1 \cap E_2 \cap ... \cap E_{n-1}) \qquad (3.28)$$

upon imposing the assumption of independence, the system reliability becomes

$$R_S = P(E_1) \cdot P(E_2) \cdot P(E_3) \cdots P(E_n) = \prod_{i=1}^{n} R_i \tag{3.29}$$

Notice the simplicity of the expression of Equation (3.29). If a reliability or reliability function is associated with each module, and the modules are arranged such that the failure of one module results in a system failure (a series reliability structure), Equation (3.29) can be used to calculate the system reliability as each module is added. Thus, if the module lives are statistically independent and follow Equation (3.27), then the reliability of the modules are a modular characteristic of the system. Consider a modular electronic system with the serial reliability structure of Equation (3.29). Say we have five modules each with a reliability of 0.999, then by Equation (3.29) the system reliability would be $R_S = (0.999)^5 = 0.99501$. This illustrates the fact that the system reliability of a system with a serial reliability structure will be less than the reliability of any of the modules.

Now, consider the case of a robot system. The electronic components of the robot system can generally be considered to have independent lives since their operating environment and stresses are constant and the failures are random electronic failures. This assumption might not hold if the robot system is being commanded to a particularly grueling program which may demand extreme torques and velocities from the actuators. In this case, the current amplifiers will be subjected to higher stress and the failure of the amplifier components may be dependent upon the current overstresses received. The first dependency has been introduced.

Another dependency can be examined in the mechanical portion of the robot system. The dynamic loads on the arm are highly variable, both due to the motion program accelerations and velocities, as well as various end-effector loads during the work process. The stresses experienced by the arm components are random variables with large variances. These loads are distributed over the mechanical system and the load distribution depends on the configuration of the system, the motion program, and the end-effector load. In other words, the stresses seen by the arm components are dependent upon the configuration of the other components in the system as well as the motion program and end effector loads. The question now arises: How does one adequately describe the reliability of the mechanical system of the robot and of the amplifiers, to include the dependencies between the components and the stresses on them.

One way to describe the reliability of components is to use Interference or Stress-Strength Theory to describe the relationship between the stresses a component receives and the strength of the component (see Appendix A). Kapur and Lamberson present this theory using the interference area between the stress and strength probability distributions to determine the reliability of a component [73, 107]. Figure 3.10 shows the relationship between the stress and strength distributions. Using the notation $f_{\mathscr{s}}(\mathscr{s})$ to represent the probability distribution function (p.d.f.) of the stress experienced by the component and $f_{\mathscr{S}}(\mathscr{S})$ to represent the p.d.f. of the strength of the component, one can express the reliability of the component as

$$R = P(\mathscr{S} > \mathscr{s}) = P(\mathscr{S} - \mathscr{s} > 0) = P(Y > 0) \tag{3.30}$$

Integrating the interference region

$$R = \int_{-\infty}^{\infty} f_{\mathscr{S}}(\mathscr{S})\left[\int_{-\infty}^{\mathscr{S}} f_{\mathscr{s}}(\mathscr{s})d\mathscr{s}\right]d\mathscr{S} \tag{3.31}$$

Note that we have now introduced the random variable Y = $\mathscr{S}$ - $\mathscr{s}$, which is called the interference random variable. To determine the reliability of a component, we must know the distribution of the random variable Y such that

$$R = P(Y > 0) = \int_0^{\infty} f_Y(Y)dY \tag{3.32}$$

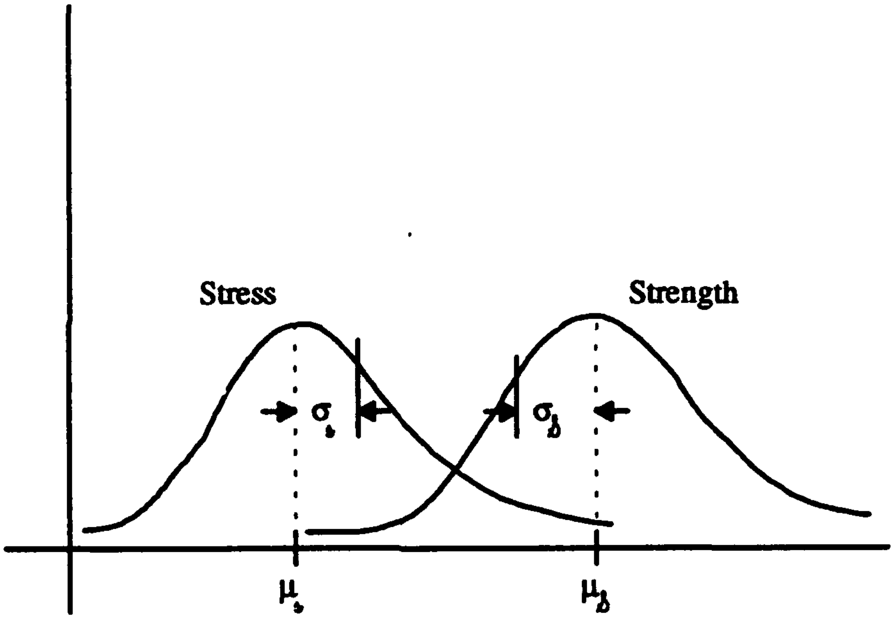


Figure 3.10. Stress and Strength Probability Distributions

If the stress and strength distributions are independent, then we can write the classic convolution integral

$$f_Y(Y) = \int_{\mathscr{s}} f_{\mathscr{S}}(y+\mathscr{s}) f_{\mathscr{s}}(\mathscr{s})\, d\mathscr{s}$$
$$= \begin{cases} \int_0^{\infty} f_{\mathscr{S}}(y+\mathscr{s}) f_{\mathscr{s}}(\mathscr{s})\, d\mathscr{s} & y \geq 0 \\ \int_{-y}^{\infty} f_{\mathscr{S}}(y+\mathscr{s}) f_{\mathscr{s}}(\mathscr{s})\, d\mathscr{s} & y \leq 0 \end{cases} \tag{3.33}$$

This distribution is univariate and represents the interference random variable for one component. To extend this to the entire manipulator structure requires the development of an overall stress and strength distribution for the arm, which requires the development of the joint stress p.d.f. and the joint strength p.d.f. If the component strength distributions are independent, then the joint strength p.d.f. is just the multiplication of the individual strength p.d.f.s together

$$\begin{aligned} f_{\mathscr{S}_0}(\mathscr{S}) &= f_{\mathscr{S}_1}(\mathscr{S}) \cdot f_{\mathscr{S}_2}(\mathscr{S}) \cdots \cdots f_{\mathscr{S}_n}(\mathscr{S}) \\ &= f_{\mathscr{S}_{1,2,\cdots,n}}(\mathscr{S}_1, \mathscr{S}_2, \cdots, \mathscr{S}_n) \end{aligned} \tag{3.34}$$

However, by our previous observations, the component strength p.d.f.s may not be independent which creates a joint p.d.f. with some type of unknown dependency structure. As of the present time, the tools to determine this joint p.d.f. and dependency structure have not been developed for practical use.

The stress p.d.f. presents a similar problem in nature if perhaps not in scope. A good assumption is that the maximum stresses and loads will be estimated during design analysis and development of general motion programs. The difficulty here is that for a modular system assembled upon demand, the stress distribution must be general in nature and not tailored to a specific motion program or profile. A possibility based upon a load characterization of the robot workspace may suffice. Consider that we know that we can determine through inverse kinematics the time domain history of the joint space designated as $\overline{\Phi}(t)$ and through design analysis, determine a generalized load vector $\overline{L}(t) = f(\overline{\Phi}(t))$. A joint stress p.d.f. may be generated from knowledge of the time history by signal processing techniques [20]. Two choices present themselves: to develop the stress p.d.f. about the mean load value L_0 or about the maximum load value L_{max} for each component of the load vector as in Figure 3.11. This technique consists of sampling the load time history of each component and calculating frequency histograms from the time history. Using the maximum load value may result in an extreme value distribution, while the mean load value will be an arbitrary distribution. This histogram can then be used to calculate a discrete signal p.d.f. representing the values of the signal (see Figure 3.12). A continuous p.d.f. can be fit using parameter estimation techniques using a hypothesized p.d.f. with finite range. Currently, this technique cannot explicitly quantify the correlation between the elements of the load vector $\overline{L}(t)$. This technique must be expanded to generate a joint p.d.f. with a dependency structure.

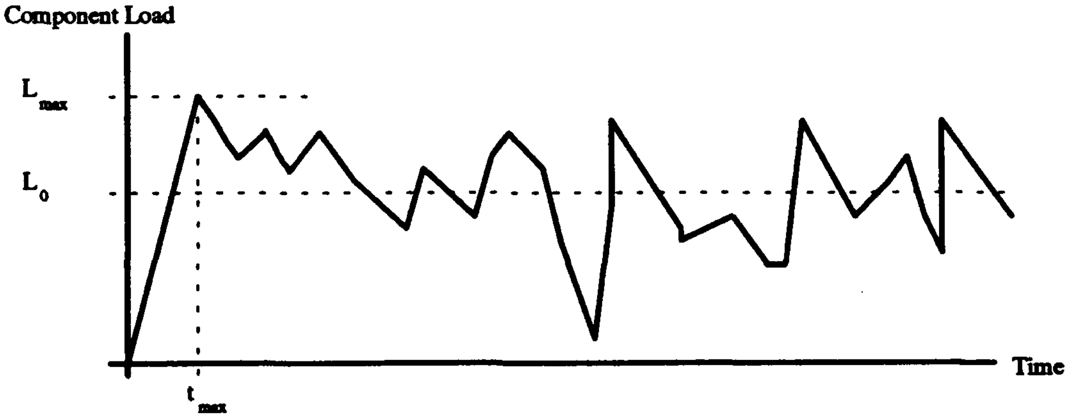


Figure 3.11. Load vs Time.

Once joint dependent p.d.f.s for the strength and stress are obtained, the interference random vector can be formed: $\overline{Y} = \overline{\mathscr{S}} - \overline{\mathscr{s}}$. To evaluate the probability of success (the reliability), the p.d.f. of $\overline{Y}$ must be determined. Since the random vectors for stress and strength have dependency structures, Equation (3.33) cannot be used to determine the p.d.f. of $\overline{Y}$ and more sophisticated techniques must be used.

A technique that shows promise for solving this problem is the use of integral transforms and *H*-functions [75, 146]. An *H*-function is a general distribution that includes most of the special distributions such as exponential, Weibull, etc. as special cases. The algebra of random variables using *H*-functions becomes simplified since manipulations solved for the general *H*-function is valid for all the special cases. A tractable way of handling combinations of random variables is through integral transforms. *H*-functions are particularly amenable to integral transform techniques since the transforms are trivial parameter manipulations of the *H*-functions themselves. Another nicety is that combinations of *H*-functions also follow the *H*-function distribution. *H*-functions have been used to examine specific dependency structures for bivariate dependent random variables [75] and this technique shows great promise for extension to multivariable combinations of random variables with dependency structures. This is the type of tool needed to provide for the reliability analysis of systems with dependent components.

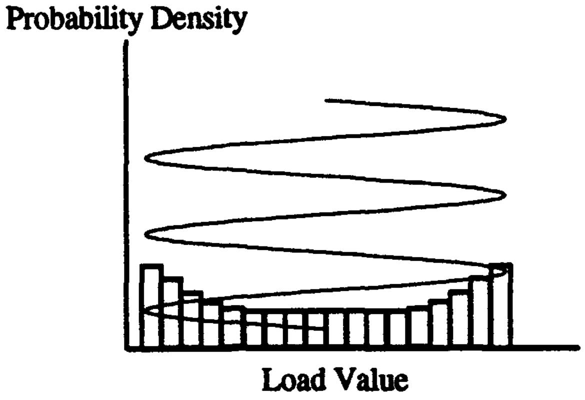


Figure 3.12. Relation of Load p.d.f. to the Load Time History [20].

The following problems must be addressed in order to take advantage of this methodology:

1. *H*-function theory and methodologies need to be developed for n-dimensional random vectors with general dependency structures. This will allow the quantification of the joint stress and strength p.d.f.s and interference random variable p.d.f. This effort will most likely result in the development of software tools to perform the mathematical manipulation necessary for this analysis.

2. Extensive investigation into the dependency structures of components in general modular systems must occur to allow quantification of the life distributions.

3. Finally, data issues must be addressed. An extended metrology effort must be undertaken to quantify the actual life distributions of modular robot systems. This data should then be used to analyze the configuration of modules for reliability optimization.

3.6. Summary

This chapter has presented a review of the use of reliability methods and tools used during the design of components and systems and as such represents the first section of the R&M Roadmap for modular robotic systems: the programmatic aspect. As a summary, Tables 3.10 through 3.12 are presented to show how the tools and techniques used in other domains are applied to robotic systems. The application of these tools were shown in Figure 3.8 which outlined the robot system design process and indicates the proper sequence and timing of the various reliability tasks.

Both the organizational and product related aspects were presented in Section 3.1.4 which outlined various organizational possibilities for R&M management. The product oriented program was illustrated by a overview of the Department of Defense Standards for Reliability and Maintainability programs [102, 104]. While these standards have been useful in the past for the development of reliability control programs, the non-judicious use of these documents without regard to cost or engineering judgment may hinder rather than support the organization.

The hypothesis that modularity inherently enhances the reliability of a system is unprovable with current data sources. The reliability of the system is determined by the system architecture (redundancy, etc.) and the level of reliaiblity of the components. This is entirely system dependent. We find that we must resort to trying to maximize the reliability of each system in design via the reliability analysis tools and predictions based upon data for similar systems or generic data. This is the main reason for this chapter emphasizing reliability techniques during design.

The next chapter presents the other thrust of reliability improvement: the supporting technologies. Chapter 4 examines the robotic system architecture from the reliability point of view and contrasts modularity with monolithic designs. It also presents assessments of the components that make up modular robotic systems and makes suggestions for reliability improvements based on generic parts data. The final chapter presents the full framework of the roadmap with both the programmatic and technology aspects side-by-side.

Table 3.10. Cross Reference for Reliability Tools for Robotic Application (Part 1)

Design Domain and Tools	Modular Robotic Application
1. Electronic Systems	Applies to controller technologies
Device Derating	Can overdesign mechanical components to provide similar effect. Will penalize system in weight and performance.
Screening	Calibration/measurement of module geometries will provide higher quality and reliability for mechanical components
Thermal and ESD Control	Robot controllers
Computer Systems	Modular robot systems take advantage of both functional and constructional modularity. Remains to use high reliability components to maximize availability.

Table 3.11. Cross Reference for Reliability Tools for Robotic Application (Part 2)

Design Domain and Tools	Modular Robotic Application
3. Mechanical and Aerospace Systems	
Factor of Safety	Overdesigned mechanical components will penalize system in weight and performance. Probabilistic design methodologies can reduce overdesign.
Reliability Centered Maintenance	Useful in maintaining operational reliability levels. Design data should impact reliability and maintenance planning.
Damage Tolerance Analysis	Prevents failure using stress analysis of Finite Element Analysis (which will usually be used during robot system mechanical design). Can identify maintenance and inspection strategies which promote good reliability.

Table 3.12. Cross Reference for Reliability Tools for Robotic Application (Part 3)

Design Domain and Tools	Modular Robotic Application
4. Nuclear Power and Power Distribution Systems	
Parts Selection	Insures use of high reliability parts which with modularity will maximize availability
Diagnostics and Modularity	Modularity increases level of maintainability. Positively effects the life-cycle-cost using module cost and time to repair. Impacts the maintenance concept.
Simplicity and Redundancy	Simple systems are more reliable by reducing complexity. Redundancy can be used to increase baseline reliability subject to the return on investment in complexity.
Separation of distribution networks into small parts for design and analysis.	Provides rationale and example for reducing the robotic design problem by modularization. Key assumption is independence of design paradigm for each component or module.

Table 3.13. Cross Reference for Reliability Tools for Robotic Application (Part 4)

Design Domain and Tools	Modular Robotic Application
5. Software Engineering	Direct application to controller
Modular Program Structure	Directly applicable to all robotic system software. Modular structure allows general top-level integration using "black-box" approach. Software Module usage is independent of module internal structure, functional specification only. Modular robotic systems require extensive module data to provide for control and accuracy.

CHAPTER 4: RELIABILITY TECHNOLOGY FOR MODULAR ROBOT SYSTEMS

4.1 Introduction

Industrial robots can be found today in an extremely wide and diverse range of applications. They range from heavy cargo manipulation with gantry mounted systems to the ultra-precise insertion of integrated circuit packages into circuit boards. The requirements and environments have ranges just as wide, from withstanding molten metal spray and high temperature tooling to the clean room. Robot systems find many other uses as well. They are indispensable in the safe handing of nuclear fuels and are rapidly becoming more capable of performing maintenance in the nuclear power plant. Human augmentation and agriculture are seeing large growths in the utilization of automation. Perhaps the area where robot systems are the most required is the space environment, both in on-orbit and planetary functions.

As stated in Chapter 3, there are two components to improving the system reliability during design: program and technology. To illustrate the impact technologies can have on the reliability improvement of a component, we present the following example. The F100 engine powers our top line fighter aircraft: the F-15 and the F-16. These engines are constructionally modular to allow for fast repair under combat conditions. These modules consist of the fan section, two turbine sections, the combustion chamber, and the augmentor. Additional modules (called accessories) attach to the engine to provide for control and power generation functions. These modules are the Unified Fuel Control, the Engine Electronic Control (EEC), and the gearbox. For this particular example, we are considering the EEC. The EEC is a complex hybrid computer that provides for the control of engine operation based upon engine speed and fuel flow. In 1986, the US Air Force identified serious reliability problems with the EEC and decided to institute Environmental Stress Screening (ESS) to reduce the failure rate. This effort was successful reducing the in-service failure rate of the F-15 EEC by 65% and the F-16 EEC by 23% [168]. This provided a significant savings in the repair and overhaul of these units. However, the engine managers identified even better technologies that could be used to provide for even better performance and higher reliability. A new Digital Electronic Engine Control (DEEC) was designed as a form-fit-function replacement for the EEC. This was only possible because the EEC itself was a module of the F100 engine and was constructionally modular allowing an easy replacement with better technology. The DEEC has shown a 97% decrease in failure rate over the EEC in the F-15 and an 89% decrease for the F-16 [169]. This shows the improvement that can be made in reliability by the proper introduction of technology. It is important to note that this improvement could not have occurred except at a much more expensive level had the EEC not been a module of a modular system.

This chapter is devoted to a review of robotic technologies that will provide for the design of highly reliable modular robot systems. The review areas include architectures, component technologies, software, and fault tolerance. These applications and reviews are then compared by subjective rankings to provide a prioritization of technology emphasis and design guidelines for reliability improvement. In all discussions, a constant failure rate is assumed. Where a non-constant failure rate is known to exist, exceptions are noted and the handling of the model is explained. The constant failure rate assumption is usually not valid when dealing with mechanical systems and should be used for comparison purposes only.

4.2. Architectural Impact on Robotic System Reliability and Maintainability

As described in Chapter 3, the system reliability is dependent upon the system design. The component of system reliability effected by the design is termed the inherent reliability of the design. This section describes the various impacts the architectures chosen for the robotic system have on various reliability and maintainability (R&M) measures. The first consideration will be a comparison of monolithic and modular architectures followed by a discussion of redundancy in mechanical systems.

4.2.1. Modular vs. Monolithic Design

Modular systems are always cited for their contribution to ease of maintenance of systems. Once the repair needed is identified, a modular system can rapidly be restored and returned to an operational state. An additional bonus is reconfigurability. The system can be rapidly configured to handle different tasks as well as easily upgraded by the addition of modification of the modules. This section explicitly considers the maintainability effects of a modular robotic architecture as it pertains to the time to repair or reconfigure a robotic system.

This analysis is based on the comparisons of standard times required to remove and replace (R&R) a component from a monolithic robotic system to the times required to R&R a similar component from a modular robotic system. These time standards are available in MIL-HDBK-472, *Maintainability Prediction* [99]. This method consists of breaking a task into elemental maintenance actions, such as removing a screw, attaching an electrical connectcr, etc. The average times required to complete these standard elemental maintenance tasks can be found in MIL-HDBK-472. The appropriate elemental tasks times are added up to give an estimate of the time required to complete the overall task.

The task considered during this analysis is the replacement of a prime mover in a robotic system. Most monolithic robotic systems provide for the easy R&R of motors since they usually have one of the highest failure rates in a robotic system. The task will be analyzed for several monolithic industrial robots as well as several types of modular systems described in the literature. This analysis takes into consideration only the time required to perform the physical assembly/disassembly on the monolithic system, not the time required to turn off and safe the system for maintenance or to insure proper functioning after the maintenance action.

For the monolithic systems, the steps to remove and replace a joint motor are outlined in the maintenance and service manuals. The R&R procedures usually start with the removal of cover plates which require removal of the machine screws holding them in place. They may also require draining oil from a reservoir or other preliminary actions. The next step is usually to gain access to the motor itself, or to the motor's electrical connections. This can be the removal of housings or electrical box covers and the time associated with the removal of screws. The motors are usually connected to wiring harnesses through threaded connectors, normally one for control and sensor signals and one for the power signals. The motor is usually secured by four or five bolts to a frame which must also be removed. Sometimes, couplings to shafts must be loosened to remove the motor from the robot. The installation is generally the same steps performed in reverse order along with alignment and positioning tasks.

The time standards were applied to several different monolithic robotic systems: the PUMA 560, the Cincinnati Milacron T^3-726, and the ASEA IRb-6. The PUMA task was the replacement of the inner link drive motor. This procedure was located in the PUMA Equipment and Programming Manual [155]. This task, assuming no testing or procedural errors would take 15.47 minutes. The T^3-726 task was to replace the upper arm drive motor [27]. Not including times to set and release the robot's brakes and power system cycling, the time required to complete this task is 20.64 minutes. The ASEA robot had the simplest motor replacement procedure of all the monolithic robots considered [9]. The task considered for this robot was the replacement of the α motion drive motor and the time to complete this task was 13.34 minutes.

The modular systems under consideration have a much simpler R&R procedure for the modules. The first assumption made is that the modules are all self-contained with no additional connections to the system except at the module interface. This lends itself to a simple support and detach sequence such as returning a robot to the neutral position, removing power and brakes, disconnecting the mechanical interface, then demating the electrical connectors. The same procedure is followed for assembly. One aspect of modularity not addressed in this analysis is the need for calibration and testing. A modular system would be expected to be fully interchangeable, with automatic methods for informing the control system of module parameters. This would remove any lengthy calibration or testing procedures from the repair process, making the time to return to an operational state even shorter. Many monolithic systems require backlash adjustments when the integrity of the robotic system is violated. This would be overcome easily in a modular system since these adjustments would be internal to the replaced joint modules and would be performed at the factory before delivery.

There are four different module interfaces that have been discussed in the literature. The first one was presented by Wurst in 1985 [166]. This design showed a square interfacial plate with four bolt holes. Both the torsional and bending loads were transmitted through these four interface bolts. Access to the bolt heads are obtained from the actuator module side with the bolt threads tapped into the links. No alignment pins could be seen in the design. Additionally, there were two electrical connectors embedded in the surface of the mechanical interface. The depth was missing from the view so it was assumed that these connectors mated as the mechanical connections are made. That is, the electrical connectors are automatically mated as the mechanical connection is being made. This feature is admirable as the mating of the electrical connectors occur automatically and does not add any time to the task. The time for removal and replacement of a module (a link or joint) from this modular robot would take 9.58 minutes. This figure was determined as described above and does not include any time for testing, power cycling, or other incidental tasks. It only includes alignment, positioning, connecting, tightening, etc. and is arrived at by taking the time to mate and demate one interface and multiplying that number by two since two interface mating/demating is require to replace a module in the middle of the kinematic chain.

The second concept is from the University of Toronto investigated by Benhabib [14]. This concept presented only the mechanical interface with no provision for electrical connections. The concept provides for cylindrical links and interfaces. The interfaces are aligned via a single alignment pin and attached by four bolts. Access to the bolts is provided by cutouts in the links. Based on the overall concept presented in the paper, three electrical connectors are assumed, one threaded connector for power transmittal and two BNC (Cannon Plug) connectors for signal transmission. The module replacement time for this design was calculated to be 7.64 minutes.

The next concept was developed at Carnegie-Mellon University and is known as the Reconfigurable Modular Manipulator System (RMMS) [138]. This design is cylindrical as well with alignment pins to carry the torsional loads. The mechanical interface is held together by a V-band clamp which provides a quick disconnect. No mention of electrical connectors are made in the description of the RMMS, however, the explanation of the control structure suggests that at least two electrical connectors at each interface is required, one for power (assumed to be a threaded connector) and one for control signals (assumed to be a BNC connector). Another notable feature is an orientation identification feature built into the links using a light emitting diode on one face and four receptors on the other face. This provides the control system with an automatic way of recognizing the orientation of each link on the system. Using these assumptions, the RMMS module replacement time is 3.08 minutes.

A similar approach has been proposed at the University of Texas. This early concept is cylindrical with eight alignment pins to provide torsional load transmittal with a threaded collar to provide for mechanical mating and bending load transmittal. The electrical connections yet to be determined would follow the pattern of Wurst by being permanently mounted in the interface and providing automatic mating during mechanical connection. The time to replace a module using this concept is 2.96 minutes.

After reviewing these different concepts, preliminary characteristics of a standard robot module interface can be suggested. The first characteristic is that mechanical torsional load can effectively be carried by alignment pins in the mechanical interface. The bending load can be carried by an appropriate choice of a union. This union could be by V-band or threaded collar to provide a quick disconnect. The collar could support more load and may be more rigid. The electrical connectors, as well as any pneumatic, hydraulic, and optical connections, will automatically mate upon mechanical connection and an automated method of providing orientation and calibration information to the control system. This module interface would have the mating properties of the Wurst interface, the alignment properties of the RMMS interface and the connection properties of the University of Texas interface. Based on this description, the module replacement time is 3.56 minutes for modules with these suggested standard interface characteristics.

These times for completion of the various tasks are displayed in Table 4.1. The differences in the values of Table 4.1 make a clear statement of the advantage of modularity in robot systems from the maintainability point of view. An alternate point of view is that of availability. If the MTBF of the system was, say, 5 hours and the MTTR was represented as the times in Table 4.1, the average availability of the monolithic robots is 94.8%. A similar reliability for a modular system gives an average availability of 98.1%. This difference becomes less important as the reliability of the systems are increased. If the system MTBF is at 5000 hours (a very reasonable number for current industrial robot systems) the average monolithic availability is 99.995% and the average modular availability is 99.998%. The availabilities quickly approach 100% as the reliability becomes much greater than the repair time. This allows the observation that the main payback of modularity will be in reconfigurability, and the ability to produce "optimal" configurations, not in availability. However, note that the assumption of the motor replacement as the mean repair time is not valid for monolithic robotic systems. This can cause a much larger amount of repair time for other failures while the modular system is already represented by its worst case: the replacement of an entire module.

Table 4.1. Replacement Times for Selected Tasks: Module or Motor Replacement.

Average of Prototype Modular Concepts	Suggested Initial Standard Interface	Average of Existing Monolithic Robots
5.8 Minutes	3.6 Minutes	16.5 Minutes

4.2.1. Serial vs. Parallel Structures

It is a well known fact in reliability theory that adding components in parallel will increase the reliability of the network. From Appendix A, the reliability function of a parallel structure is

$$R_S(t) = 1 - \prod_{i=1}^{n} [1 - R_i(t)] \tag{4.1}$$

This formulation provides for a system reliability that is always greater than or equal to the reliability of the components. This is directly opposite from a serial system where the system reliability is always less than any of the system's components.

Adding parallel redundancy is a preferred method of improving the reliability of electronic systems. This is true since this redundancy may easily and cheaply be incorporated into electronic designs. The usual choice for electronic redundancy schemes is a technique known as *Triple Modular Redundancy* (TMR). The TMR configuration (shown in Figure 4.1) uses a voter to check the outputs of the modules against one another. If one module output does not match the other two, the module is assumed failed and taken off-line and the output of the other two modules is used. This arrangement can only suffer one failure; on the second, the system will fail.

Only rarely will you see more than four components in a parallel arrangement. This is because the payback in higher reliability rapidly decreases as you add more components in parallel, especially as the component reliability increases. It may be more profitable to improve the components rather than add redundancy. This is especially true in mechanical systems where it is extremely difficult to design in a redundancy. An exception is the University of Texas Robotic

Fault-Tolerant Architecture which makes use of mechanical redundnacy to achieve fault-tolerance (see Section 4.4 for a description). At best, mechanical systems can be considered as shared loads (See Section A.2.1.1.1. in Appendix A).

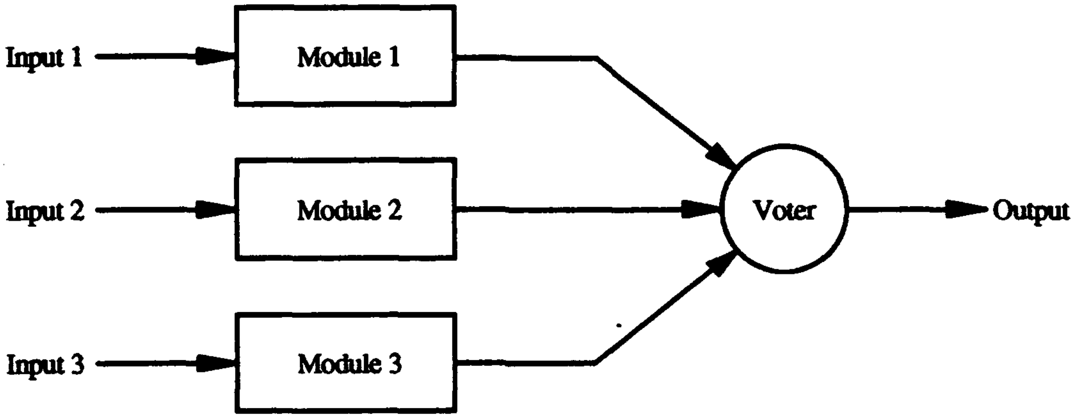


Figure 4.1. Triple Modular Redundancy [70]

One example of mechanical redundancy is seen in bolt configurations. Generally, more than one bolt is used to support a structural member. This represents a shared load configuration as well. Another example directly applicable to robotics is the application of redundancy in the actuators. A design currently undergoing testing at the University of Texas is a prime example of redundant actuation [67]. This design, shown in Figure 4.2, has complete symmetry about the center axis of the module (effectively two complete actuators in one housing). All the normal functions of an actuator are replicated twice: motor, resolver, brake, and gear train. This module has a design specification of 300 hours MTBF. A reliability analysis on the design was conducted according to [23] resulting in a MTBF of 1535 hours for one half of the actuator. This reliability analysis assumes constant failure rates which indicates an underlying exponential distribution. The MTBF above of 1535 hours represents the half load condition while the fully loaded condition (one side failed) can be represented as 2/3 the full load which gives a MTBF of 1025 hours. If we let the half load failure rate be λ_h and the full load failure rate be λ_f, then Equation (A.37) can be written as

$$R(t) = e^{-2\lambda_h t} + \frac{2\lambda_h}{2\lambda_h - \lambda_f}[e^{-\lambda_f t} - e^{-2\lambda_h t}], \quad t \geq 0 \tag{4.2}$$

which is not an exponential failure distribution. Integrating Equation (4.2) to determine the actuator Mean-Time-To-Failure (MTTF) gives

$$\text{MTTF} = \frac{\lambda_f + 2\lambda_h}{2\lambda_h \lambda_f} = 1791 \text{ hours} \tag{4.3}$$

Thus, the actuator reliability specifications were greatly exceeded with a 16% increase in MTTF over a single actuator.

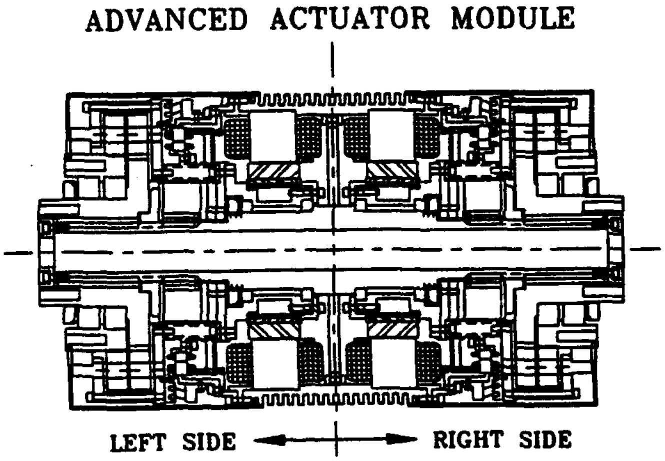


Figure 4.2. University of Texas Actuator Module [67]

An additional type of redundancy exhibited by robotic systems is that of a parallel structure. This can exhibit itself in two ways. First, a link or actuator can be duplicated while the kinematics of the mechanism remains a serial chain. This is the type of redundancy exhibited by the UT actuator module described above. Secondly, a redundancy can be added to the kinematic chain, transforming it to a parallel chain.

One of the simplest parallel mechanisms allowing actuation redundancy is the five-bar linkage. A widely accepted definition of a parallel mechanism is a closed loop (sub)structure with more joints than degrees-of-freedom (DOF). The five bar mechanism shown in Figure 4.3 has two degrees-of-freedom and has five joints.

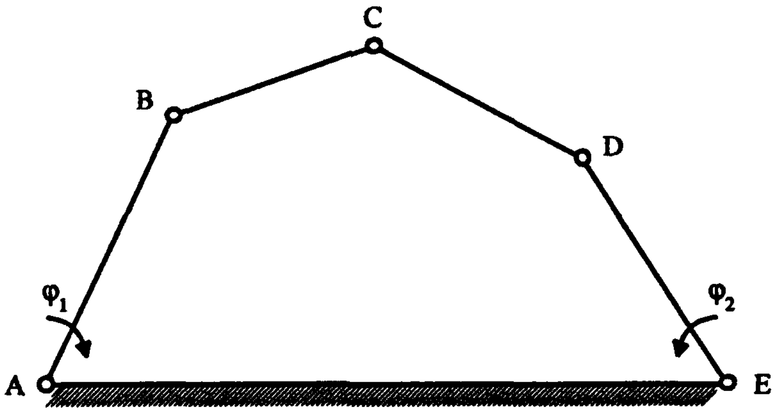


Figure 4.3. Five-Bar Linkage

Parallel structures are attractive for use in robotic systems for a number of reasons [148]: they have high positional accuracy and can carry much higher nominal loads than serial systems due to the load sharing that can occur in the parallel legs. Parallel systems have better structural stiffness and can have less effective moving mass than comparable serial systems. The parallel structure allows for distributed actuation (across the base legs) and distributes the errors and deformations over the entire structure, thus minimizing them. One final positive attribute is the fact that the human anthropomorphic structure is a parallel structural system, with the ability to provide antagonistic actuation and control of forces. So it is with a parallel robotic structure.

The drawbacks of parallel structures are that they possess less workspace and have less dexterity than serial mechanisms and more complicated forward kinematics and dynamics. Nevertheless, parallel mechanisms can provide a great deal of protection from failures when they do occur (fault-tolerance). This is because it is easy to provide for redundant actuation in a parallel mechanism. In the parallel case, a failure of a joint must result in a free motion to allow movement of the mechanism, whereas in the serial case, a failure must result in a locked joint to allow any continued operation after a joint failure.

For the reliability analysis of parallel mechanisms, there have been two methods used. One method is based on the errors in the system and the probability of successfully following a certain path (kinematic reliability). The second method is a structural approach to the prevention of stress-related material failures in the legs. The kinematic reliability is a performance related criteria that can be related to the structure if failure is defined as a loss of precision. If one is only interested in the physical survival of the system, structural techniques can be used to estimate the reliability of the structure and FTA or FMEA use to estimate the reliability of the system.

The kinematic reliability of mechanisms is examined in some detail in Chapter 5 and in [17]. The development of structural reliability is based on stress-strength interference theory discussed in Appendix A. This leads to the definition of a reliability index based on the probability of failure. Recalling the discussion in Appendix A, we can define an interference random variable Y as

$$\mathrm{Y} = \mathscr{S} - \mathscr{s} \tag{4.4}$$

where $\mathscr{S}$ is the strength random variable and $\mathscr{s}$ is the stress random variable. If both the stress and strength are normally distributed and independent, then Y is also normal distributed with mean

$$\mu_{\mathrm{Y}} = \mu_{\mathscr{S}} - \mu_{\mathscr{s}} \tag{4.5}$$

and standard deviation

$$\sigma_{\mathrm{Y}}^2 = \sqrt{\sigma_{\mathscr{S}}^2 + \sigma_{\mathscr{s}}^2} \tag{4.6}$$

The reliability of the structure can then be written as

$$R = P(\mathrm{Y} \geq 0) = \Phi(\beta) \tag{4.7}$$

where Φ is the standard normal cumulative distribution function (c.d.f.) and β is called the reliability (or safety) index and is defined as

$$\beta = \frac{\mu_{\mathscr{S}} - \mu_{\mathscr{s}}}{\sqrt{\sigma_{\mathscr{S}}^2 + \sigma_{\mathscr{s}}^2}} \tag{4.8}$$

In general, structural reliability is not expressed as the reliability but as the reliability index, which is the argument to the standard normal c.d.f. This characterization is based entirely on the moments of the distributions of the random variable. This is the basis for what is known as the *First Order Second Moment* (FOSM) method for estimating structural reliability [38]. The problem is that the joint probability distribution function (p.d.f.) of the design problem are not normal and are generally unknown. The FOSM addresses this problem by deriving a generalized form of

Equations (4.5), (4.6), and (4.8) that are not dependent upon the underlying distributions. This is convenient since the mean and standard dev:ation or variance can always be estimated for any data set used to represent the design variables.

The FOSM method begins with the determination of the ***State Function*** of the design problem $g(\mathbf{X})$ where

$$R = P[g(\mathrm{X}) \geq 0] \tag{4.9}$$

The state function, $g(\mathbf{X})$, can be considered a failure surface that when $g(\mathbf{X}) = 0$, describes the boundary between "safe" and "failed" regions of the design space described by the random vector **X**. By using the properties of combinations of independent random variables and using a Taylor Series expansion we arrive at estimates for the moments of $g(\mathbf{X})$ as

$$\begin{gathered} \mu_{\mathbf{Z}} = g(\mu_{\mathbf{X}_1}, \mu_{\mathbf{X}_2}, \ldots, \mu_{\mathbf{X}_n}) \\ \sigma_{\mathbf{Z}}^2 = \left[\left(\sum_{i=1}^{n} \frac{\partial g}{\partial x_i} \bigg|_{\mathbf{X}=\mu} \right)^2 \sigma_{\mathbf{X}_i}^2 \right]^{1/2} \end{gathered} \tag{4.10}$$

where the random variables $\mathbf{X}_i$ are independent. Then

$$\beta_{\mathbf{Z}} = \frac{\mu_{\mathbf{Z}}}{\sigma_{\mathbf{Z}}^2} = \frac{\mu_{\mathscr{S}} - \mu_{\mathscr{s}}}{\sqrt{\sigma_{\mathscr{S}}^2 + \sigma_{\mathscr{s}}^2}} \tag{4.11}$$

where we have assumed the state function to be the interference random variable. In this case, $\mathscr{S}$ and $\mathscr{s}$ can be from general distributions.

If $g(\mathbf{X})$ is non-linear, significant errors can occur since the expression for the variance in Equation (4.10b) only contains the first term in the Taylor series and higher order terms become important. This method is also not invariant to the mechanical formulation of the problem. Different values for the variance will be obtained for different equivalent state functions. Another difficulty is in the development of the state function. For complex parallel structures, the state function can be quite complex and difficult to determine, and may even be a set of non-linear coupled equations, depending upon the design variables one is interested in. Other more advanced methods of structural reliability estimation address these problems and a good estimate of a static structure can be obtained.

Several conclusions can be reached based on this discussion. The first is that unless probabilistic design methodologies are used, the factor of safety approach described in Section 3.2.3 will provide a reliability approaching unity for the structure under static loading. Dynamic loads are extremely hard to quantify probabilistically since the input loads will change. The second is that the reliability of parallel structures are the result of the configuration and can help in the selection of the configuration, however, since the structural reliability will normally be over designed, performance issues and the desire for fault-tolerance should dominate the selection of the parallel structure with the reliability analysis, either structural or kinematic, used to finalize the actual configuration selected during the detail design.

4.3. Component Reliability Improvement Technologies

The reliability of robotic systems is determined by the reliability of the components used in the system. This section examines the components of robotic systems and how they are applied in the robotic system. Technology improvements are noted as well as general trends in development. Reliability estimates of the components are included when available in the literature. The components reviewed are communications, interfaces, sensors, actuators and motors, controllers, end-effectors, links and structural members, and last but not least, software.

4.3.1. Communications and Interfaces

4.3.1.1. Communications Systems

The heart of a robotic system is the controller which translates the user's commands and desires into the appropriate motions of the robotic mechanism. The arteries that carry the commands from the controller to the actuators in the joints and data from the sensors to the controller comprise the communication system. This subsystem includes the wires that carry the information and power, as well as components that convert or transform signals from one domain to another. It can also be considered to contain the controller itself and the sensors as well. The controller and sensors are discussed in later sections.

A block diagram of a typical industrial robot is shown in Figure 4.4. The system converts from analog signals to digital signals at the interfaces in the figure. These are the Analog-to-Digital and Digital-to-Analog converters used in the computer controller. The communications system we are concerned with here are the actual wires and connectors that connect the control systems to the robot hardware. In a typical industrial robot, over 150 power and signal lines are necessary to operated the system [1]. If there is no reduction in this number, this means all these wires must pass through each module. To provide for fast and easy connection and disconnection, connectors for all these wires must be installed at each module interface. This problem requires the development of methods to reduce the number of signal and power paths through the modules. The answer to this problem is two-fold and each possible solution has its own reliability implications.

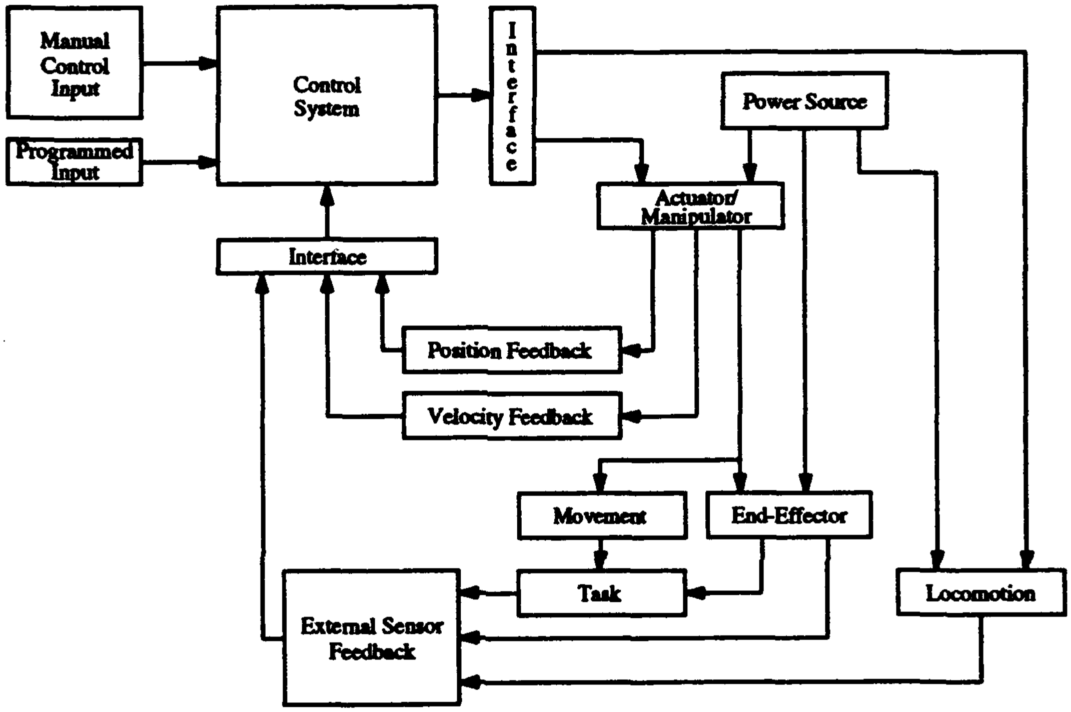


Figure 4.4. Typical Industrial Robot System Control System Block Diagram [83]

The types of communication in a robot system can be divided into two different systems, power and signals. The power system provides the appropriate currents from the motor amplifiers to drive the servo-motors in the joints providing the actuation of the robot. The signal system transmits sensor information obtained on the arm and its surroundings back to the control system which computes the proper currents to send out over the power system. Thus, the problem can be stated in two different parts, first, reducing the number of signal lines in the robot and second, reducing the number of power lines through the structure.

The first problem is addressed by the use of a data bus in the structure of the robot, thus removing the dedicated lines for each sensor. A data bus is a standard configuration of data and control wires (or fiber optic cables) that share the data through a multiplexing scheme. The controller send and receives data from the sensors over the same communication lines. There are two important reliability implications in this method. First, the communication system reliability will be increased by the physical reduction in the number of wires and connections which must be mated and de-mated at each module interface. Also, bus communication protocols have a tremendous impact on the reliability since they govern how well the data is received and sent as well as the error content. This strategy can also reduce the system reliability by the addition of more complexity. The send and receive functions at the sensors must be driven by interface electronics and microprocessors in the modules themselves.

This implies a distributed control system such as one described in [115] where different layers of control are physically distributed throughout the manipulator structure. The highest level of control is designated the Intelligent, Fault-Tolerant Control level. This is the real-time task controller integrating the robot system into its workspace and allowing for task completion despite faults in the robot system and its surroundings. This control level drives the System Control level. The System Control level is what is generally thought of as the robot arm controller. This system is the one that actually commands responses from the arm's actuators. It coordinates the actions of the joints to complete its task. This control level communicates to the intelligent control level and to the actuator control level. The actuator control level is the actual servo loop. This level monitors motor speed, and other functions and adjusts the motor parameters to accomplish the task commanded from the system level. This level of control would be embedded in the joint module itself. The fourth level of control is the sensing level. This is the lowest level of the control hierarchy and processes the sensor information for use by the other three levels.

This control hierarchy adds to the number of components in each module, reducing the reliability. However, a network easily allows for redundancy and error checking which may offset this degradation. A network is an arrangement of components that communicated with each other over a grid of interconnections via a specified protocol.

There are two basic type of multiplexing methods: frequency and time. Frequency multiplexing sends information on different carrier frequencies simultaneously. This allows multiple devices to receive information at the same time. However, the devices on the network must be able to broadcast at all the frequencies while receiving at only one. This adds additional complexity to the network devices in their receiving and transmitting circuitry. Time multiplexing allows each device to have a specific amount of time on the network. Timing is critical in this type of network and is provided from a single source to provide synchronization. This is called a synchronous network. A synchronous network can either have a master-slave configuration or a token passing protocol. The master-slave relationship uses a master controller to control all the data transmission on the network. The master controller can make data transmission to specific addresses or it can broadcast to all devices on the network. Requests for data are usually address specific and are sequential. It is desirable to have the addresses of the devices in the network to correspond to their position in the manipulator system. Token passing protocols create a specific order of the device's access to the network data lines. A device can receive and transmit only when it possesses the control token. This token is passed in a specific order to all of the devices on the network.

Asynchronous time multiplexing allows independent access to a data bus, which can cause problems when specific pieces of data are needed at specific times. One example of an asynchronous network is the ETHERNET. The devices on this network can receive and transmit at any time, but only if the network is not being used by a different device. The devices requiring transmit check to see if the network is busy. If it is busy, the device waits a moment and tries again. If the network is still busy, the device waits a longer time (this is known as exponential backoff). If the data is needed by the master controller, it may not arrive in time causing control problems. The limiting factor for a network is the bandwidth, or speed at which data can be transferred.

One of the largest impacts of multiplexing networks on reliability can be related to the topology or arrangement of the network. The networks can be arranged in stars, trees, and rings (see Figure 4.5). The traditional topology is the star network. Service and control is performed by a central processor. Multiplexing is not necessary in a star network. Failure of the central processor causes the entire network to fail, however, the network can suffer failure of a remote node without affecting the rest of the network. A topology that is inherently fault-tolerant is the ring network. In a ring, the information can flow both directions allowing two distinct data paths to a remote node. This allows error checking and monitoring of the network. Multiplexing is required since all nodes use the same cable. Access logic is more complex for a ring network. A tree network is an open-ended bus that is relatively inexpensive and simple to design and use. This simplicity promotes good reliability and is generally used for broadcast systems. Multiplexing is also required; a preferred method is synchronous token passing protocols.

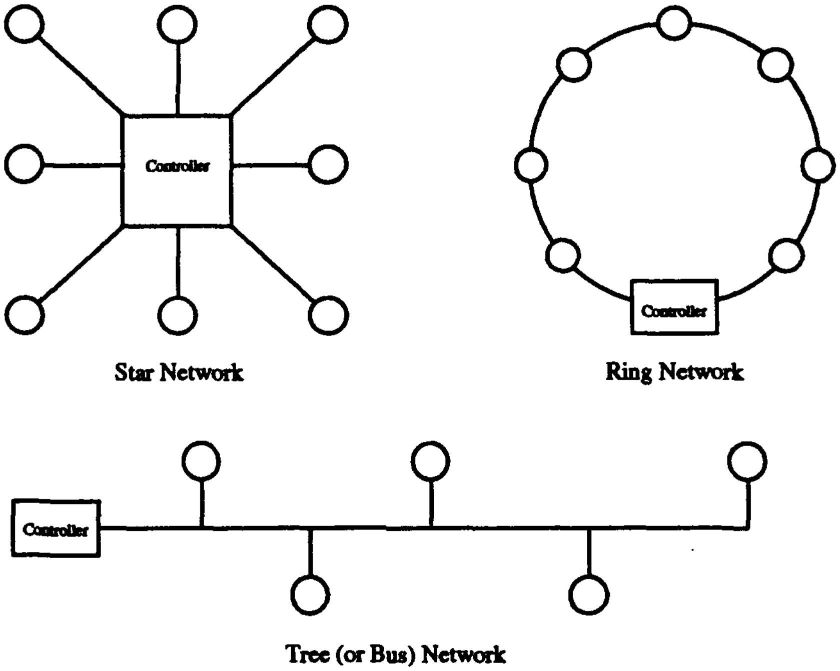


Figure 4.5. Network Topologies [21]

The actual physical connections between the nodes either can be twisted copper wire, coaxial cable, or fiber optic cable. The copper wire is the least expensive and can be obtained in many different forms and capacities. However, copper wire is highly susceptible to noise and interference and is limited in bandwidth without amplification [21]. Coaxial cable is much less noisy and has a much higher bandwidth for signal transmission. Coaxial cable cannot support high power densities. Both copper wire and coaxial cable can be easily tapped and signals inserted in a parallel fashion. The interfaces for these cables are straightforward and do not require the conversion of energy from one domain to another. Fiber optic cables are becoming much more popular for data and communication systems. Optical fibers use light as the carrying medium and can have extremely high bandwidths. An additional mode of multiplexing available for fiber optic networks is a form of frequency modulation using different wavelengths of light to send information. The information carried by each wavelength can be frequency or time modulated, allowing access to many more devices than is possible with conductive mediums. Fiber optic networks are also immune from electrical and radio interference and can transmit data for long lengths without amplification. These networks are also high resistant to corrosion and temperature variations [21]. There are penalties for using fibers. The first is in complexity. The electronic data must be transformed into light for transmitting through the fibers and must be received and converted back. Work is proceeding on optical processing, but to date, it is not a practical technology. This transformation of energy increases the complexity of the devices using the network. Optical fibers are also hard to tap for signal extraction and injection which makes parallel networks difficult to construct. Practical implementation of optic networks require each fiber to terminate and the data re-transmitted to the next node. This can increase the time required for processing and reduce the bandwidth of the network.

A survey of commercial industrial robots was performed as part of this investigation. Questions were asked of the companies about failure and design stresses. The unanimous consensus was the robotic systems components with the highest failure rate were cable assemblies. None of the manufacturers were willing to support this claim with reliability data*, however, this does appear plausible since most of the stress on cabling will come directly from the movement of the robot system itself. This claim is not supported by the generic data presented in Table 4.2, however, the environment can have a severe effect on the system's components and the flexing may not be represented in Table 4.2. Cable failures can manifest themselves in two ways, The first is an open signal or power path. This is probably the easiest failure mode to detect, since it results in the loss of signal or power. Another failure mode is a short. This can provide an alternate signal path or can hold a high or low logic level. This failure can propagate back into the control system allowing erroneous conditions to occur. Both of these modes can occur suddenly, such as a lost connection due a fatigue failure occurring as a result of bending or vibration, or a burnthrough of the insulation caused by an electrical overstress. These problems can also occur slowly, as the result of continued bending stress or corrosion.

There are several ways to prevent these problems. First, one must design strain relief into the cable assembly. This will prevent stress buildups in the conductors from having to support its own weight under dynamic loads. Better cable/connector interfaces can be used, many with built-in strain relief and environmental barriers to prevent corrosion occurring inside the connector housing. The proposal of a modular mechanical architecture addresses most of these problems, given a reduction in the number of required conductors. Since the cabling

* All companies contacted viewed reliability data as proprietary and would not release it.

will be internal to the modules, they can be quite well supported internally and the stress levels on the connectors can be reduced. The bending stress problem still exists in the joint modules, however, they will not be as severe since all of the signal and power paths will be tapped in each joint module. The paths will be bent around the module axis, but it can be well supported.

An alternative to this problem is to traverse the interior of the motor itself. Many motors have a hollow shaft to reduce the motor's inertia. This tube can be used as a path for cabling, allowing the removal of the cabling from the bending mode, but adding a torsional load. If twisted wires, are used, this will substantially increase the life of the cable since twisted wires can tolerate a torsional load much better that bending loads. Additionally, the torsion can be removed entirely if the motor is considered to be the module. Consider attaching the yokes to the links themselves and the motor being the only physical connection between the links. In this case, interfaces will be located at the pivot of the joint itself, requiring the interface connection to be made in the center of the pivot joint. This will require the inclusion of some method of transferring signals and power over the rotating interface. By allowing rotation in the connector, the torsional load is removed from the cable and transferred to the connector, where the wear can be more closely monitored. This will reduce the reliability of the interface since a wear component (slip rings, etc.) has been introduced.

Table 4.2. Estimated Reliability of Cabling Technologies [129, 96, 131]

Technology	Failure Rate (Failures/10^6 hrs)	MTBF (hrs)
General Cable	0.0107	93.5×10^6
Coaxial Cable	0.0014	714×10^6
Fiber Optic Cable	0.0037	268×10^6

The second problem that exists is the reduction in the power signals required to drive the joint motors. As described in the control hierarchy description above, the actuator control level will be embedded in the joint module itself. The power requirements will be the same as current motor amplifiers and if the reduction in power lines is to be accomplished, the power amplification must take place within the joint module itself at the actuator control level. This would allow the use of a simple power bus throughout the structure with the joint modules connected to the bus in a parallel fashion, as in any electrical distribution system. Recent advances in power semiconductors allow the placement of power amplifiers in the robot structure [1] but this raises another reliability issue that must be addressed during design.

The power requirements remain the same, thus the power dissipation in the power semiconductors will be commensurate with current levels. This means a great deal of heat will be generated by the power electronics located in the joint module. A widely accepted and empirically justified opinion is that the higher the operating temperature of electronics, the higher its failure rate and the lower the reliability. The heat generated by the power amplifier at the joint must be adequately removed to preserve the reliability of the module itself. Hence, in the future, it may be necessary to provide active cooling in the joint modules of advanced systems. A trade-off then must to be made over the reduction in reliaiblity of the system caused by including the active cooling system into the module interface (see Table 4.3).

4.3.1.2. Interfaces

The module interface is perhaps the most critical design in a modular robot. The interface will provide for the transmission of forces through the structure, the connection and transmission of power and signal energy, and at the some time, be easy to disconnect and reattach. Some of the current designs under consideration are discussed in Section 4.2.1.1. The design criteria for the mechanical interface include the minimization of bending and torsional deflection as well as quick disconnecting and connecting. These measures are strength related and will be addressed during the structural design (deterministically or probabilistically). As a practical matter, the structure will be overdesigned and will not fail structurally under normal loading. However, deformations will occur affecting the accuracy and repeatability of the system. Thus, the reliability concerns in the interface will focus on the electrical and fluid connectors present in the module interface.

An important question when considering the design of the module interface is which electrical connectors to use. The module connector will need to pass both signals and power. Additional fluid connectors may need to be present to allow for pneumatic actuation of an end-effector or for active cooling of the joint modules. A great deal of information has been gathered on connector reliability, most related to mating cycles. Some failure mechanisms that are intrinsic to electrical connectors are corrosion, loss of normal contact force through stress relaxation, and temperature degradation due to excessive contact resistance. Extrinsic failure mechanisms include contamination, use of connectors outside its range, excess current, and improper mating practices [109]. Other considerations include the amount of crosstalk between signal paths, noise shielding, impedance matching and current capacity.

Electrical connections can be grouped into six different application areas or levels. The first level is the device to package level. This level includes the wire bonds from the IC package to the semiconductor chip inside. The second level is component to circuitry (printed circuit board (PCB)). This level of connection refers to the ways the chip carriers and discrete components are attached to the printed circuit board. These connections can be permanent (the IC or device soldered to the PCB) they can be separable (chip sockets). The third level is the board-to-board level. This level of interconnection is where PCBs are connected to each other inside the cabinet. This includes board to board cabling as well as the connections to a backplane. The fourth level of interconnection is subassembly-to-subassembly. These are the connectors within cabinets such as the power strips. The fifth level is subassembly to I/O ports. This level represents an interface with the outside world both in signals and power. At this level, shielding and filtering become concerns and environmental protection and ease of mating/demating are paramount. The most rugged of all the levels is the sixth level: system-to-system. This level refers to cable assemblies and power cords and their connections. The concerns at this level are electromagnetic and radio frequency interference.

The connections of main concern for the reliability of modular robot systems can be grouped mostly into levels five and six. The connectors within the computer will be in a benign environment comparea to the connectors used in the arm itself, thus having a far lower failure rate than those used in the modules. The connectors of concern will be those at the module interfact and at the motor housings. One connector that is at a lower level that will be present on the arm are the connectors to the sensors and internal to them.

As described in Section 4.2.1.1., the module interface will probably contain the signal and power connectors that automatically mate while the mechanical attachment is being made. This will preclude the use of "cannon plugs" or threaded connectors. There may be some type of spring loading to ensure a slightly higher mating force on the connectors to ensure a good contact and to overcome the mating resistance of self-sealing fluid connectors that may be present in the interface. MIL-HDBK-217 [96] gives a generic rate of failure for friction type connectors at 0.017 failures per million hours. This rate is due to the intrinsic modes of failure described above. Realistically, in the author's opinion, a modular robot in industry will be reconfigured five to ten times over the life of the robot system. Those in an academic or research environment will be reconfigured more than that, however, the environment will be much more benign in the lab than in an industrial facility. This leads to the conclusion that intrinsic failure modes of the connectors will dominate since most commercial connectors are rated for thousand of mate/demate cycles.

The other types of connectors that may be present in the interface are self-sealing fluid connectors for hydraulic and pneumatic systems. The data available in the NPRD suggests fluid coupling failure rates at 5.7 failures per million hours corrected to a ground fixed environment [131]. The configuration of the couplings were not specified, but it probably assumes a static coupling. If dynamic seals are added, the coupling will more closely approximate the self-sealing fluid connectors that will be required. Dynamic seals were stated to have exhibited 3.004 failures per million hours which would be added the coupling failure rate to give a rough estimate of 8.704 failures per million hours for the self-sealing fluid connector.

Table 4.3. Estimated Reliability of Generic Connection Technologies [129, 96, 26]

Technology	Level	Failure Rate (Failures/10^6 hrs)	MTBF (10^6 hrs)
Fluid (Self Sealing)	5-6	8.704	0.115
Pneumatic (Self Sealing)	5-6	8.704	0.115
Printed Circuit Board	3-4	0.094	10.6
BNC/Threaded	5-6	0.017	58.8
Friction/PCB Assembly	3-6	0.024	41.7
IC Socket	2	0.0088	113.6
Coaxial Connector	5-6	0.015	66.7
Fiber Optic Connector	3-6	500-1000 matings	*
Fiber Optic Coupling	5-6	0.141	7.1

4.3.2. Sensors

The technology that makes control feedback possible is sensors. A sensor can be defined as a device that transforms physical energy, such as forces, pressures, positions, etc., into electrical or optical energy so it can be measured in some way [134]. A typical feedback arrangement can be seen in Figure 4.6.

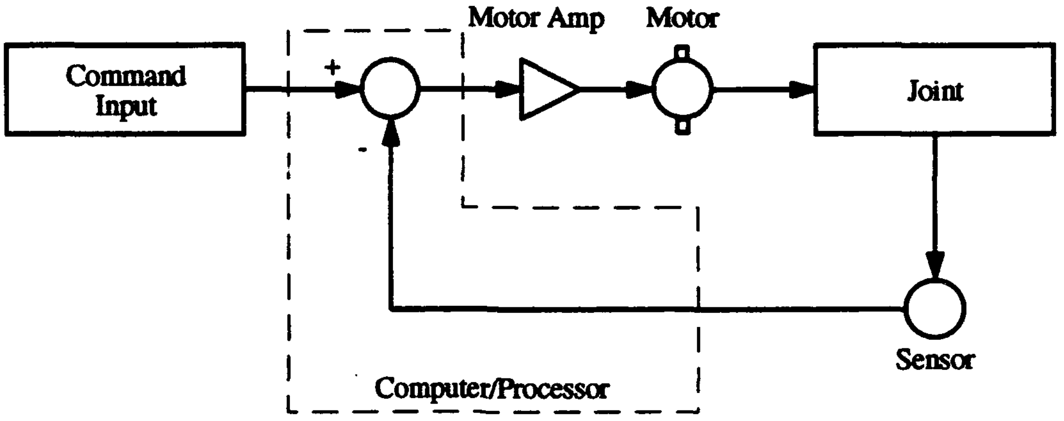


Figure 4.6. Typical Robot Joint Feedback Block Diagram

Not only is sensor information required to control the movement of joints to a commanded value, it is required to integrate the motion of the manipulator into the overall task. Some of the types of measurements required are position, velocity, acceleration, force and torque, tactile, and current. All of these measurements are important to the operation of an integrated system. This section presents reliability assessments of the different technologies involved ii sensing our environment. The application of sensors should be decided by the measurement needed, then the technology examined for reliability. Vision systems will not be examined due to their wide diversity.

4.3.2.1. Position Sensors [83, 146]

There are several different types of position sensors commonly used in robotic systems today. They are potentiometers, resolvers and synchros, and encoders. Potentiometers are simple resistive voltage dividers that output a dc voltage proportional to the linear or rotary position of a device being measured. A rotary potentiometer configuration can be seen in Figure 4.7.

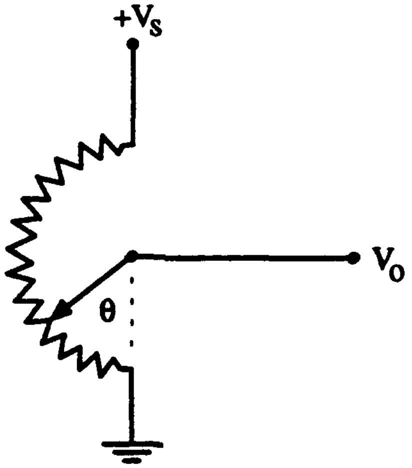


Figure 4.7. Rotary Potentiometer Functional Diagram

The potentiometer senses the angle of the shaft by a voltage division such that $V_o = K\theta$. The voltage output varies as a slider attached to the shaft slides over a coil. This is known as the resistive track which can be made up of many different types of materials such as carbon films, conductive ceramics, or wound wire. The main advantage to potentiometers are their low cost and small size. They do however have several disadvantages that make them unpopular for industrial use. The greatest disadvantage is that wear occurs as the slider moves across the resistive element. Nothing can be done to alleviate this problem since the operation of the potentiometer depends upon this contact. This wear causes potentiometers to have a limited life in comparison to other position sensors. Another disadvantage is that potentiometers are limited in rotation to less that 360°. This can pose problems when measuring position of a motor shaft making multiple revolutions. An additional drawback is that the output of the potentiometer is an analog voltage, which requires additional processing to be used in a computer control system. MIL-HDBK-217 reports failure rates as high as 28 failures per million hours for precision ceramic potentiometers.

A more popular option is the synchro or resolver. This sensor is similar to an electric motor. Both have a single winding on the rotor and the synchro has three stator windings and a resolver has two stator windings. The rotor is energized with a high frequency alternating voltage (50 Hz to 10 kHz) which induces a voltage in the stator windings. The voltages between the windings can be compared to generate the shaft angle. The accuracy and reliability of synchros and resolvers are very good however, they have the same failure modes as electric motors (winding or bearing failure). An additional problem is involved in applying the voltage to the rotor. The usual method is to use slip rings and brushes to transfer the voltage to the rotor, but brush wear is a problem with this method. An alternative to this can be used for direct drive systems which prevent the full rotation of the shaft. In this case, the power can be hardwired to the rotor with slack provided for rotation of the shaft. However, this can produce fatigue in the wire to the rotor if overused or under supported. MIL-HDBK-217 predicts the failure rate of a synchro or resolver to be 0.32 failures per million hours in a ground fixed mode (a factory floor) at 25°

centigrade. The RADC Non-Electronic Part Reliability Data (NPRD) Handbook gives a similar estimate, after correction to the ground fixed environment [131].

Another type of angular position sensor is an encoder. Encoders can be incremental or absolute and can be optically or magnetically operated. In an optical encoder, light is passed through a grating disk attached to a shaft. The grating disk of an incremental encoder is radially striped and the angle the shaft is turned through can be determined by sensing the number of stripes that pass by. The output from the light sensors on the receiving side of the grating disk are pulses corresponding to the number of radial stripes that pass by the photocell. The direction of shaft travel is determined using another light source offset from the first creating a phase lead or lag depending upon which way the shaft moves. The precision of an incremental optical encoder is determined by the number of radial lines on the grating. This imposes a size restriction on the encoder since the detector footprint limits the number of lines that can be sensed which imposes a lower limit on the size of the encoder for a certain accuracy.

Absolute optical encoders work much the same way except that the grating is slotted such that the pattern on the disk is a binary code. A separate light source and receiver is required for each bit on an absolute encoder. The output of an absolute encoder is a binary code so no additional processing is required and the encoder output can be used directly by the controller. Some encoders use a binary *Gray Code*, which is similar to a strict binary system except that only one bit at a time changes level, rather than several at a time. This improves the reliability of the encoder since it reduces the possibility of an erroneous signal, but additional circuitry must be added to the encoder to transform from the Gray code back to strict binary [83].

The magnetic encoder uses the same principle except the pulses counted are generated by magnetic material mounted on the encoder disk and sensed by induction. Optical encoders are generally more precise and are preferred. The optical encoder light sources also vary. If the encoder uses Light-Emitting-Diodes (LEDs), the photocell must be kept very near the light source to prevent light leakage from affecting the photocells. This limits the size of the encoder as well. If a laser is employed as the light source, only one source is required and the coherent light beam can be split into as many separate beams as is required for the encoder's accuracy. This does introduce additional devices into the optical path which can effect the reliability of the device. It definitely affects the ruggedness of the device, since precise optical alignment is needed in a laser encoder.

A clear advantage of encoders is the lack of wear occurring in the device. This means the only failure modes are bearing failure or random failure of the optical or electronic components. Empirical data bear out this relation since the failure data reported in the 1991 NPRD [131] shows a maximum failure rate of 1.5 failures per million hours for commercial optical encoders. This value is much higher than the actual figure since no failures where reported in 686,000 hours of operation which requires a very large confidence limit. This level of reliability is commensurate with resolver reliability.

4.3.2.2. Velocity and Acceleration Sensors [134, 83]

Velocity feedback is used to eliminate resonances in the robotic system by providing additional damping. Velocity information can be obtained by differentiating position information, however, numerical differentiation can produce large errors or even instability if there is noise in the sensor measurement or if the sensor has a digital output (such as an encoder). This forces the use of direct measurement of velocity or the integration of acceleration information (which has a tendency to remove errors). Acceleration is not normally used since accelerations can only be measured with respect to an inertial frame, preventing the determination of relative velocities through integration. For this reason, acceleration sensors are generally not used on any joint except the first. Thus, the only way to get velocity information from distal joints is to measure the velocity directly.

Table 4.4. Estimated Reliabilities of Generic Position Sensing Technologies [129, 96]

Technology	Failure Rate (Failures/10^6 hrs)	MTBF (hrs)
Potentiometer	28	35,700
Resolver (with brushes)	0.32	3.125×10^6
Synchro (with brushes)	0.32	3.125×10^6
Optical Encoder*	<< 1.45	>> 686,000

* This is an upper bound on the failure rate. The NPRD data showed 686,000 hours of operation without failure. The failure rate was zero during this time.

The generally accepted velocity sensor is called a tachometer. Other velocity sensors include tachsyns, resolvers, and optical encoders. A tachometer operates on the same principle that an electric motor uses. In fact, the tachometer is a dc generator which has an output voltage proportional to the rotational speed of the armature. The main drawback of tachometers is their use of brushes to commute the armature windings. There will also be a ripple in the output voltage due to the winding poles on the armature which can cause control instabilities. The life is limited on tachometers due to the use of brushes which will wear. The MIL-HDBK-217 brush factor increases the failure rate of a tachometer by as much as 3 times, depending upon the number of brushes used. The NPRD gives a tachometer failure rate of 15.3 failures per million hours in a ground fixed mode. Another type of tachometer is called the tachsyn [87]. This device is a three phase permanent magnet rectified alternator which is a brushless dc tachometer. This allows at least a three fold reduction in the failure rate over tachometers with brushes. The tachsyn would have the same reliability characteristics as a brushless dc motor.

New devices are also available to the designer for combined velocity measurement. By using additional electronics, velocity information can be obtained using resolvers and encoders. The electronics are internal to the sensor housing reducing the noise and with new filtering techniques, stable velocity outputs are obtained. This technology greatly enhances the reliability of the system by reducing the number of components used to sense the parameters. However, a failure in the sensor will could affect both channels resulting in the loss of two components of information rather than one. Using a single sensor can degrade the redundancy of the system if both sensors were being utilized in a fault detection algorithm.

4.3.2.3. Force Sensors [134]

The basis of the measurement of dynamic forces can be found in the equation $F = ma$. If we can measure the acceleration an object is experiencing and know its mass, we can determine the forces causing that motion. However, this relation only holds for dynamic measurements. To measure static forces (which are the usual forces one wishes to measure when accomplishing a robotic task), we must make use of the fact that some materials deform a known amount to a certain applied force. This is also known as the spring equation, or $F = kx$. In this case, we can calculate the force applied to an object if we can measure the deflection and know the spring constant k. The force sensors in this class are generally known as strain gauges. There are several different types of sensors in this class: metallic strain gauges, piezoresistive (semiconductor) strain sensors, capacitive, and optoelectronic force sensors.

The metallic strain gauge is actually a resistive element bonded to an epoxy or laminate. The resistance at zero deflection is known and the material properties of the gauge are known as well. When a deformation is introduced to the gauge, the resistance changes proportionally to the applied force causing the deformation. The sensitivity of the gauge is called the gauge factor and can be calculated by taking the ratio of the change in resistance to the change in length. The gauge factor can also be calculated from the material properties of the gauge. The governing equation of the strain gauge is

$$F = \frac{EA}{G_f}\left(\frac{\Delta R}{R}\right) = K_T \Delta R$$

where E is the modulus of elasticity, A is the cross sectional area, G_f is the gauge factor, R is the total resistance of the gauge, and ΔR is the change is the resistance due to the deformation caused by the applied force F. The change in resistance, ΔR, can be measured by various means. Metallic strain gauges are very sensitive to changes in the environment since the deformations being measured are in the range of thermal expansion effects. This requires temperature compensation, usually provided by placing additional strain gauges in the same vicinity to provide a base resistance measurement at the temperature of operation. This sensitivity has reliability implications as well, requiring fairly gentle handling procedures to prevent damage to the sensor. The material of metallic strain gauges is usually nichrome filament. Nichrome filament epoxy resistors have a ground fixed failure rate of 0.0118 failures per million hours [96]. The RADC Non-electronic Parts Reliability Data Handbook reports data on metallic strain gauges where they have recorded 26,000 hours without failure.

A similar strain sensor is the piezoresistive or semiconductor strain gauge. This sensor operates on the same princip'e that the metallic strain gauge uses, however, it is more sensitive to deformations and operates at lower temperatures. The sensor is construrted out of doped semiconductor which has very large resistive changes as the sensor deforms. The sensor is more fragile since it has lower mechanical strength, however, there is no generic reliability data available for semiconductor strain gauges.

Table 4.5. Estimated Reliabilities of Generic Velocity and Acceleration Sensing Technologies [129; 96, 131]

Technology	Failure Rate (Failures/10^6 hrs)	MTBF (hrs)
Tachometer	6.26	160,000
Tachsyn (brushless)	3.14	318,200
Resolver	0.32	3.125×10^6
Optical Encoder*	<< 1.45	>> 686,000
Force Balance Accel.	26.6	37,500
Potentiometer Accel.	6.1	165,000

The next type of force sensor is the capacitive strain gauge. This device exhibits a change in capacitance which is proportional to the force applied. This sensor is linear for only very small values of strain. The sensor is very sensitive but has limited application due to its small range of usefulness. Reliability data for the capacitive sensor was not available as well, however, since the device is essentially a variable capacitor, generic data for variable capacitors can give a general idea of the reliability level of the technology. MIL-HDBK-217 gives values from 0.098 to 1.2 failures per million hours.

Optoelectronic force sensors are complex devices that generally finds use in tactile sensor arrays. These sensors use the fact that leakage from an optical fiber is

* This is an upper bound on the failure rate. The NPRD data showed 686,000 hours of operation without failure. The failure rate was zero during this time.

proportional to the radius of curvature of the fiber. This fact can be used to measure the deflection of a fiber array subjected to an applied force. The output of such an array will be a photocell voltage proportional to the amount of light leakage. From the physical parameters of the array, the deflection can be calculated. These arrays are made up of many different sensing elements, i.e. fibers and photo^ells. The reliability of this technology was addressed in Section 4.3.1.

Table 4.6. Estimated Reliabilities of Generic Force Sensing Technologies [129, 96, 131]

Technology	Failure Rate (Failures/10^6 hrs)	MTBF (hrs)
Metal Strain Gauge	0.0118	84.7×10^6
Semiconductor Strain Gauge	Unavailable	Unavailable
Capacitive Force Sensors	0.098 - 1.2	$9 \times 10^5 - 10^7$

Strain gauges are not usually used by themselves on a component or structure. The most common use in robotics is to place a strain gauge array onto a load cell that is attached between the robot wrist and the end effector. The load cell is a small structural device placed in the robot's load path, allowing the strain gauges attached to the structure to measure the transmitted forces and torques. The load cell usually has many strain gauges at various places inside the structure and includes electronics to transform the strain gauge resistances directly to computer compatible outputs. The failure rates of load cells can be calculated using the failure rates of the electronics and from the stain gauge failure rates. The structure of a load cell is compliant, since deformation is necessary to measure static forces. This introduces additional compliances into the robot structure which must be compensated for if the system accuracy is to be preserved. The load cell structure is also designed for certain loads. If those loads are exceeded, the load cell will be damaged and erroneous readings will result. Fail-safe stops can readily be designed into the control system to limit these loads.

4.3.2.4. Hall Effect Sensors [146]

A semiconductor device used to detect magnetic fields rapidly gaining in popularity is called the Hall Effect sensor. When a current carrying semiconductor is placed in a magnetic field perpendicular to the current flow, a force is exerted on the electrons moving in the current and causes a voltage across the semiconductor proportional to the strength of the magnetic field. This is also known as the motor effect and allows the operation of electric motors and generators. Hall Effect sensors are very small and can be employed in many applications with little weight penalty. Hall Effect sensors are commonly used to sense current levels in electric motors, and in proximity and presence detection circuits. The NPRD Handbook states a maximum failure rate of 0.5899 failures per million hours since the data represented 1.7 million hours of operation without failure.

Table 4.7. Estimated Reliabilities of Hall Effect Sensors [131]

Technology	Failure Rate (Failures/10^6 hrs)	MTBF (hrs)
Hall Effect Sensors	0.5899	1.7×10^6

4.3.2.5. Tactile Sensors [135]

As stated above, an important trend in future robotic sensing is the sensing of tactile stimulation. Tactile sensing is finding applications in tasks where vision systems are impractical. The particularly important applications for tactile sensing include the handling of objects that vary in size and shape, such as foodstuffs. Intelligent controllers can also take advantage of the information provided by tactile sensor arrays to make criteria based decisions on task planning and adjustment. When speaking of tactile sensors, one is referring to the sense of touch. This implies measurement of forces, position, and orientation.

Tactile sensors are usually arranged as an array since this allows for a single type of sensor to determine both position and force at the same time. One of the simplest touch sensor is the position switch. An array of switches can be laid out (with microswitches, the array can have a very high density). The switch measures whether or not the applied pressure exceeds the force threshold to actuate the switch. The problem with using switches is that they are generally too large and there is no way to adjust the switching threshold. It is this drawback that most modern tactile arrays try to overcome. The basic principle of tactile arrays is to have an array of contacts (which can be in a semiconductor base) covered by a sheet of some conductive, elastic material. When contact is made with the elastic sheet, the sheet deforms and makes contact with the electrical contacts underneath, creating a circuit path. The amount of force is proportional to the number of contacts energized and the position can be determined by which contacts are energized. Constructing the sensor with semiconductor base also allows the controlling and sensing electronics to be integral to the sensor itself.

Tactile arrays can be optically based as well. A sheet of plastic will guide light and if it is bend or a deformation is introduced into its surface, light will leak from the sheet and emerge from the sheet face. A photodiode array will be able to sense this and the amount of light leaking from the sheet will be proportional to the applied force. A mechanical device can also be embedded in an array that will interrupt a beam of light between an emitter and detector, however, arrays of this nature have a low resolution.

Magnetic sensors, such as Hall Effect sensors can also be used for tactile arrays and can be arranged to be sensitive to applied torques. Additionally, tactile sensors can also be capacitive, much like the capacitive force sensor. An elastic spacer (or elastic dielectric) is used between two rows of electrodes. The two row arrays are rotated 90° with respect to each other and the change in capacitance between the two arrays as a force is applied can be measured and related to the force and position of the object being gripped.

As one can see, there are a wide variety of tactile sensor technology being investigated for application. However, no reliability data has been published for any type of tactile array. An estimate for a particular technology could be made using the component technologies 's failure rate estimates, but this would need to be done for each type of array and would be dependent upon the size of the array and the density of the embedded devices. In general, one should seek to avoid sensors that are subject to mechanical wear, such as switch arrays. Solid state devices usually will have the lower failure rates than combined mechanical and electrical devices.

Depending upon the performance specifications needed for a tactile array, a magnetic or capacitive tactile array should provide better reliability than others.

4.3.3. Actuators, Motors, and Servo-motors

There are three power sources to choose from for the actuation of robotic mechanisms: electrical, hydraulic, and pneumatic. The earliest robot systems where hydraulically actuated [45]. Of these early robots, the hydraulic systems where a major contributor to the overall failure rate of the robotic system. While hydraulic systems can carry large payloads, problems with fluid flow and pumps (which are traditionally high failure rate items) [131] prohibit their use in applications where maintenance will be restricted and high reliability is necessary. Hydraulic actuation is particularly unsuited for modular robot systems since the high pressure hydraulic lines must pass through the module interfaces raising the interface complexity tremendously as well as reducing the real estate available for load transmission and electrical connectors. Pneumatically actuated robots are generally not considered to be precise enough for most industrial applications and do not see much use except in end-effectors or educational robots. For these reasons, this section will only address electrically driven actuators and motors. Additionally, the principles of operation and the performance aspects of motors for robot systems can be found elsewhere [4, 6, 30, 146]. This section will describe the components that make up electric motors and present two reliability models for them. Based on the relative merits of the model parameters, a suitably reliable motor can be selected.

The power to drive the electric motors of a robot can be supplied as direct (dc) or alternating (ac) current. The choice of power supply can have an enormous impact on the reliability of the drive system, since it immediately determines the complexity of the motor and the motor controller. This is because a direct current motor requires commutation (which is the changing of the direction of the magnetic field) to produce the torques creating rotation of the motor rotor. Traditionally, commutation for a dc machine is performed by conductive brushes sliding over a commutator (a slip ring type of configuration) attached to several different windings on the rotor. As the rotor rotates, the windings on the rotor are energized in order creating the torque required to turn the rotor. The brushes are the life limiting aspect of direct current motors. Other components in electric motors are magnets, windings, shafts, bearings, and housings. Additional components usually integral to the motors are shaft position and velocity sensors and current sensors (see Section 4.3.2).

A recognized fact is that electric motors have a non-constant failure rate and exhibit a pronounced wearout phase at the end of their useful lives. This characteristic is widely modeled by the use of the Weibull distribution. MIL-HDBK-217 considers three components governing the life of electric motors: windings, bearings, and brushes. The MIL-HDBK-217 model uses a Weibull model with a specified maximum life for the bearings combined with an exponential model for winding life. Empirical data is used to generate estimators for the bearing Weibull characteristic life and the winding failure rate based upon the ambient temperature of the motor. A good guideline is that a 10° increase in temperature will cut the winding mean life in half [141]. This effect can be seen in Figure 4.8. A brush factor based on the number of brushes present in the motor can multiplied with the failure rate.

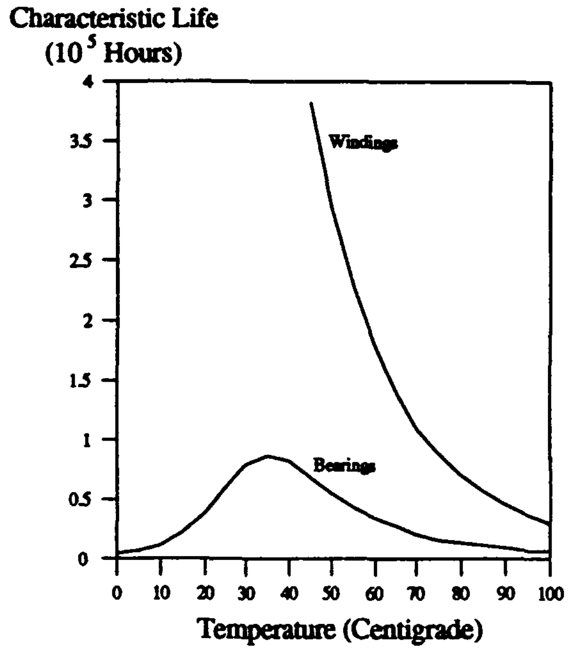


Figure 4.8. Lifetimes by Temperature for Motor Winding and Bearings from MIL-HDBK-217 [96]

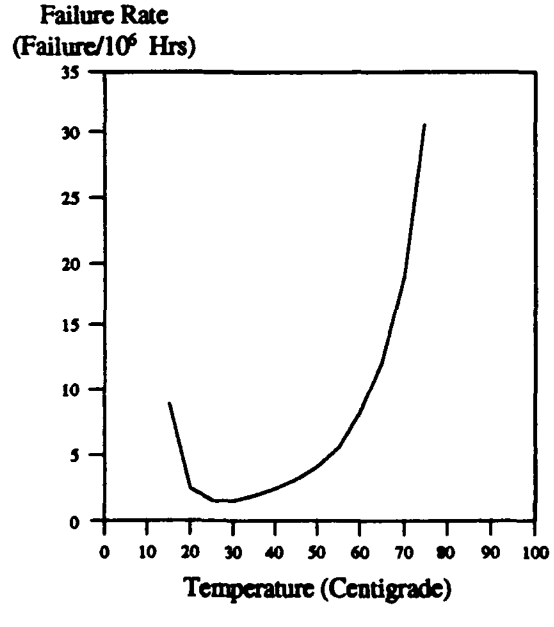


Figure 4.9. Failure Rate vs Temperature for Electric Motor with a Design Life of 10,000 Hours from MIL-HDBK-217 [96]

Table 4.8. Brush Factors from MIL-HDBK-217 [96]

Number of Brushes	Failure Rate Multiplication Factor
2	1.4
3	2.5
4	3.2

An alternate model is presented in the Naval Surface Warfare Center Carderock Division's *Handbook for Reliability Prediction Procedures for Mechanical Equipment* [23]. This model has more components than the MIL-HDBK-217 model, but is equivalent in most respects. This model adds the failure rate of all the different components of the motor to get the overall motor failure rate. The wear factors are included under the component models. This model includes bearings, windings, brushes, shaft, housing, and gearing. The bearing model, unlike the MIL-HDBK-217 bearing model, is based upon the manufacturer's L_{10} life and is expressed in failures per revolution. An average rotational speed must be assumed to convert from revolutions to time. This is roughly equivalent to the MIL-HDBK-217 maximum life model but since it uses the manufacturer's data, it will give a better predication of the bearing life. The failure rate is then modified for load conditions, lubricant viscosity, contamination, and vibration based upon the difference between the manufacturer's specifications and the actual application.

The Carderock motor winding base failure rate model is identical to the MIL-HDBK-217 model and predicts the winding failure rate as a function of ambient temperature. The base rate is then modified for insulation factors, electrical voltage variations, temperature cycling, and altitude effects.

The Carderock model also includes shaft and housing terms. However, unless severe overloading occurs, the failure rates for the housing and shaft will be much less than those of the bearings, windings, and brushes and can be assumed to be zero for practical purposes. An additional factor included is a gear failure rate component. The Carderock gear reliability model is based upon the manufacturer's specified failure rate at a specific speed, load, lubrication level, and temperature. This failure rate is modified for usage differing from the manufacturers specifications.

The last part of the Carderock model is a failure rate for brushes (if used). The developers of the Carderock model say only that a brush model is under development, however, correspondence with them suggested the model will be based upon the material found in ASME's *Wear Control Handbook* [122]. A failure rate model is not presented as such, but factors contributing to the wear and degradation of brush performance are given. In general, brush performance is optimized (low wear, maximum transfer of current, low noise, etc.) if the voltage drop across the brush-commutator/slip ring interface is less than 2.5 volts, and the friction coefficient between the brush and sliding surface is between 0.08 and 0.35. It is best to have the lowest possible values of voltage drop and friction coefficient. Most brushes are made of graphite composites and according to [122], the highest wear mechanism occurs when the graphite brush is very dry, and does not exhibit the natural lubricating effects of graphite. The preventative for this condition is to prevent an extremely dry atmosphere, operating at over 6000 parts per million of water vapor or other vapor, such as hydrocarbon, or by adding a additional component to the brush material that acts as a lubricant in low humidity or high temperature environments. This wear is made worse by operating at high temperatures and the *Wear Control Handbook* suggests limiting the brush temperature to 150 °C if low brush wear rates are required. Additional factors such as spring pressure and contact force and the location of the brushes within the machine also impact the wear rate of the brushes.

An additional failure mode can be encountered in ac and brushless dc motors: demagnetization of the rotor magnet. Both the ac synchronous motor and the dc brushless motor operate on the same principle: an alternating or commutated current in the stator coils produce an electromagnetic force on the rotor, causing rotation. This force is supplied by the interaction of the stator winding magnetic field with the magnetic field of a permanent magnet attached to the rotor. Demagnetization can occur if the motor is placed in an excessive external magnetic field or by operating at temperatures over 100 °C or lower than -40 °C. It can also be caused by allowing or creating excessive current flow in the stator windings by an instantaneous reversal of applied voltage [51].

By examining the technologies available and the generic failure rate data, one is able to conclude that brushless electric motors are the obvious choice for superior reliability of the electric drive. Considerations still need to be made of the complexity of control (inductive ac control or electronic dc commutation), however, this is easily addressed with modern computational capabilities. The survey of robot manufacturers showed a tendency to use ac motors more often than the dc brushless motor. This is because the power densities can be greater for an ac machine for the heat generated and the simplicity of the ac design provides better reliability than dc brushless motors.

Table 4.9. Estimated Reliabilities of Generic Electric Motor Technologies [129, 131, 42]

Technology	Failure Rate (Failures/10^6 hrs)	MTBF (hrs)
1-10 hp AC Motor	1.333	750,200
DC Brushless	0.9812	1,019,160
DC Servo (w/brushes)	14.0	71,400
DC Stepper	6.658	150,200

4.3.4. Gearing and Gear Heads

Unless one is using a direct drive method of actuation (see [8]), the speed of the motor driving the joint must be reduced and the torque capacity enhanced. If the joint is prismatic, the reduction usually occurs in the leadscrew drive. If the joint is a revolute joint (as most robot joints are), a reduction in the revolutions and speed must occur between the motor and the joint. These reductions are usually accomplished through the usage of gear trains. In most current industrial robots, the gear trains are an integral part of the robot system. The motors are fairly easy to remove and replace, but gear replacement is a much more involved procedure. Not only must the robot be completely disassembled to reach certain drives and shafts, but the backlash and alignment of the gearing must be adjusted upon reassembly. This makes for extremely long repair times.

A modular robot will most likely have a combined motor/gear head in the actuator/joint module. The footprint of the gearhead must be kept to a minimum and weight and inertia are of great importance. These design pressures will force the designer of a modular robot joint to use the latest speed reduction technology available. These technologies can be identified as common gear trains (spur gears and pinions) in a variety of configurations, planetary gear systems, harmonic gear systems, and cycloidal drives. All of these technologies are currently available, although some may have limited applications and higher expense than others.

Precision gears are usually not sold individually, but are packaged as gear heads designed for certain applications. It is usually much more cost and time effective to select gear heads from available sources than designing a gear system from scratch. These gear systems all have bearings and internal support structures that effect the reliability of the gear system. Gear train use data suggests that the bearings will limit the life of the structure [23]. This observation is borne out in the examination of manufacturers catalogs which invariable state the life of the gear system is based upon the L_{10} life of the bearings used. The life of the gears themselves are limited by wear, directly effecting backlash and therefore, precision. This high level question of precision over the life of the gear train is not supported by any reliability data.

Standard gear trains used in industrial applications usually consist of drive gears and pinions in standard arrangements. Reliability information on standard gears have been compiled by gear type and cannot be directly compared to gear systems unless all components of the gear train, including bearings, are used to calculate a gear failure rate. These reliability levels are reported in Table 4.10.

Table 4.10. Estimated Reliabilities of Generic Gear Types [42, 131, 129]

Technology	Failure Rate (Failures/10^6 hrs)	MTBF (hrs)
Spur Gears	3.15	317,500
Pinions	0.073-0.22	$4.5 \times 10^6 - 14 \times 10^6$
Bevel Gears	1.33	751,900
Helical Gears	5	200,000
Reduction Gear Box	5	200,000

Planetary gear drive systems have two types of components. They are the planet gears, and sun gears and can be found in many possible arrangements. The speed reduction of the gear train occurs from differences in the planet and sun gear teeth and even larger reductions are obtained as the planets orbit the sun gears. The arrangements can be made very thin; the Ferguson's Paradox planetary gear system used in the UT Modular Actuator Module is less than two inches thick [67]. Current manufacturers catalogs state about 8000 hours average life for a fixed planetary gear system at rated load and speed. This figure increases dramatically when the system is used under less severe conditions. As an example the UT actuator module's calculated failure rate for the gear system (including the bearings) is 15 failures per million hours or a MTBF of 66,700 hours. This low failure rate is due to a great deal of overdesign in the gear system of the actuator module.

Harmonic Drives have become popular due to their high accuracy and efficiency and low inertia. The input shaft is attached to a slightly eccentric cam that is attached thrjugh a friction surface to a flexible spline. This flexible spline engages in a fixed circular spline that has more teeth than the flexible spline. As the input shaft is rotated, the eccentric drives the flexible spline in the opposite direction, meshing with the outer spline. This drives the outer spline at a reduced rate depending upon the difference in teeth between the two splines. The harmonic drive principle depends upon the flexing of an internal component. The design of the flexible spline must be such that the bending stresses induced in the spline is less than the material's endurance limit. The proper choice of materials can make the harmonic drive a very reliable speed reduction technology. Statements by manufacturers put the average life from 10,000 to 15,000 hours at rated loads.

A speed reduction technology similar to the harmonic drive is the cycloidal drive. The basic principle is the same, however, the eccentricity in the drive does not drive a flexible component, it drives a bearing race with eccentric holes. This race is meshed with pins to drive the output disk. The speed reduction is obtained by having fewer balls than slots between the eccentric disk and the output disk. The only stresses felt in a cycloidal drive are due to the torque transmission through the balls between the plates. Some cycloidal drives use pins in eccentric plates to obtain the same results. The life limit in a cycloidal drive is the wear between the plates and the balls or pins. Proper material and lubricant selection can extend the lives of cycliodal drives to incredible lengths. Only one manufacturer stated any life data, and they only stated that the lubricant should be changed every 20,000 hours or every four or five years.

Table 4.11. Estimated Reliabilities of Various Speed Reduction Technolo,₃ies

Technology	Failure Rate (Failures/10^6 hrs)	MTBF (hrs)
Planetary Drives	125	8,000
Harmonic Drives	80	12,500
Cycliodal Drives	< 50	> 20,000

If reliability is a true driver of the design of a modular robotic system, the need for gear reduction should be seriously examined. As one can see from Tables 4.10 and 4.11, gear systems with bearings can have a fairly high failure rate and can exhibit performance degrading behaviors such as backlash or hysteresis. If a direct drive system is used, the failure mechanisms present in gearing are eliminated and the limitation of bearings is significantly reduced by eliminating the gear system bearings. The price one must pay is in the complexity of the control of the direct drive system [8] and, due to torque limitations in the current state-of-the-art, an extremely reduced load capacity.

4.3.5. Clutches and Brakes

In robotic systems, it is usually not desirable to control deceleraiion via mechanical braking. Brakes are included in the actuator of a robot to allow for the locking of joints during maintenance and stiffness requirements, in other words, a parking brake. The locking action can take several forms. One could use a normal brake friction material with brake pads, or the shaft or rotor of the motor could be wedged. In either case, a robotic motor brake will generally be actuated with loss of signal. This means that power must be applied to the brake to prevent it from stopping rotation of the joint. This power-off braking is desirable in robotic applications since it would in effect freeze the robot in case of a power loss. It also has the ability to lock a joint if a motor were to fail in a free motion case. Such brakes are usually electromagnetically actuated. The electromagnets are energized and keeps the brake from engaging.

Since the robotic brake generally will not experience friction wear, as an automobile brake does, the failure modes of the brakes that need to be recognized are a premature closing of the brake (caused by loss of electromagnetic force), or the brake not closing when the electromagnets are de-energized. The second failure mode can be made extremely remote by the use of the springs with very large spring force. This does force the use of large and powerful magnets to keep the brake open, so a tradeoff must occur here during the design. The first failure mode will be the most probable. It can have several root causes, the most likely being a complete loss of power to the robot. This case cannot be considered a failure since it is precisely this event the brake is made to protect against. This failure will most likely occur due to a control system malfunction causing a loss of signal to the electromagnet, or an open or short in the power path to the electromagnet. In either case, the failure will cause the locking of a joint.

The same rationale applies to the use of clutches in robotic systems. Robot systems will not required the transfer of power through a multistage transmission, or the engagement of a shaft to a moving shaft, which is the traditional use of a clutch. In a robotic application, a clutch will be used only to decouple a shaft connecting two motors, as might be found in the UT Actuator Module. This would allow the brake to be applied in one motor upon failure and the clutch decoupling the shaft so the other side of the actuator can continue operation. Thus, the robotic clutch will not require friction surfaces, it will only require a method of removing the coupling between shafts, such as an electromagnetic clutch, or even shear pins. As a practical matter, clutch failure will occur only if another failure already exists requiring the release of the clutch. A release of the clutch under normal operation should not affect operation of a robotic system, since it would be designed to operate with the clutch released for a short time. This reasoning leads to the conclusion that the clutch will not impinge the reliability of a robotic system since the clutch failure will only affect the ability of the robot to recover from the previous failure. This is not to say that the reliability of the clutch itself is not important, only that it will not effect the overall reliability of the robotic system.

The reliability levels given in Table 4.12 are for brakes and clutches using friction surfaces and are not generally applicable to robotic applications. The data is included to make the designer aware of the general levels of reliability expected in commercial and vehicular applications. The failure rates in robotic applications will be far lower.

Table 4.12. Estimated Reliabilities of Generic Clutches and Brakes [129, 131]

Technology	Failure Rate (Failures per 10^6 hrs)	MTBF (hrs)
Electromechanical Brake	10.64	94,000
Friction Clutch	5.32	188,000

4.3.6. End-Effectors [83, 146]

The robotic manipulator is useless without a tool that allows for the accomplishment of the desired task. This task could involve the manipulation of objects: positioning, rotating, inserting, etc., or it could involve the application of finishes: painting, plasma deposition, welding, etc., or perhaps the removal of material: rivet removal, deburring, etc. As can be imagined, the differences in applications prevent the use of general purpose end-effectors, and requires the use of specialized tooling.

One of the most common robotic tasks is the gripping and positioning of objects in the robot workspace. This can be accomplished using jaws that open and close, actuated by any number of different means, depending on the task, or by *expanding* flexible grippers, or by the application of vacuum. A familiar method of handling ferromagnetic materials is by using an electromagnet.

Painting and other material deposition processes require some sort of gun attached to the robot's wrist. The material being deposited also needs to be transported to the end-effector. This requires additional fluid lines or wire feed mechanisms, all of which increase the complexity of the system. Sensors also need to be included to allow the control system to sense the limits of the task.

The removal of material also presents unique problems. If the removal is by mechanical means, such as deburring or shot-peening, mechanical energy must be supplied by the end-effector to the workpiece in a manner that the material is removed. If removing paint by laser, the aiming and sensing devices must be located on the end-effector, but the laser could be remote and the light transferred to the workpiece by optical fiber.

Several observations can be made for modular systems. The first is the power supplied to operated the end-effector probably will be supplied by the modular robot, itself, either from the robot system power bus or by internal fluid lines. The reason for this is that additional lines for the end-effector violates the functional separateness required for the modular system. Secondly, unless a special "end module" is developed, the end-effector will use the standard modular interface. This will place size and positional restrictions on the tool and also effects the maximum capacity of the gripper.

The point of this discussion is not to describe all possible end-effectors or the technology being used. It is to illustrate the enormous diversity among end-effectors and the tasks they are designed for. As a result, there is no realistic way to judge the reliability of end-effector technologies. The choice of end-effector is application specific. All of the reliability methodologies described in Chapter 3 and Appendix A can be utilized in the design and selection of end-of-arm tooling.

4.3.7. Structural Members and Links

An extensive search has been made for any type of failure in the structural components of robotic systems. There have no documented reports of a failure in any structure of robotic systems, such as bending or breaking of links or attachments. There have been instances of performance drift that has been attributed to the deformation of robot structural components, but there have been no published reports of failure. During the survey of manufacturers, there was one incident of cracking of a link, but the company representative stated it was caused by a massive overload at the end effector. The engineers responsible for the process (an automotive assembly plant) knew they where overloading the robot and were expecting to damage it. The deformations occurred over time and they were able to monitor the amount of deformation by frequency with which they had to recalibrate the robot.

Much attention has been given to the performance aspects of robotic systems and how that performance relates to the stiffness of the robot structure. The definition of kinematic reliability recognizes this relationship by defining a failure as the end-effector being outside a specified error bound in both position and orientation. As demonstrated by the experience noted above, the monitoring of performance criteria, such as kinematic reliability, can provide early detection of impending failure, thus allowing preventative maintenance and correction of the fault.

4.3.8. Controllers

The controller of a robotic system is usually considered to be the computer system and interface components that use software to control the movement of the robot system. It can also refer to the specific algorithm used to design and implement the robot control structure. Both of these meanings impact the reliability of a robot system, but in different ways. The first definition of a controller implies the actual hardware components of the computer system. This reliability falls into the class of electronic and computer reliability, discussed in Section 3.3.1. There are many technologies that are increasing the reliability of electronic systems, and the robotic system can take advantage of the strides being made, not only in improved reliability but in all aspects of controller performance such as speed, capacity, and bandwidth.

The most noteworthy of these technologies is Very-Large-Scale Integration (VLSI). Design of VLSI circuits has progressed to the point where they are almost completely automated in computer-aided design systems, allowing for very reliable designs (no latent errors or design flaws). The high density of the circuits allow for extensive error checking and recovery schemes to be included in the design without sacrificing performance. Computational systems based on this technology are extremely fast and powerful, allowing for effective distribution of computational facilities throughout the robot system. Since the devices operate at extremely small voltage levels, heating is reduced which also improves controller reliability.

The second consideration in the controller is the control algorithm itself. Control theorists equate control algorithm reliability with stability, both in terms of control algorithm design and the performance of the algorithm under failure of the plant and sensors. It is not the intent of this section to provide a summary of control strategies and their effectiveness. This type of information can be found in any good control text. A good survey paper on robotic control and optimization is [125]. Instability in a control system can result from three general causes: improper algorithm design, plant failure, or sensor failure. An additional source of failure (instability) upon implementation is a mistake in the coding of the algorithm.

Improper algorithm design is essentially a design defect or the inability of a particular algorithm based upon some theory to provide adequate compensation for motion of the plant (robotic system). This a is fundamental error in the selection of control theory to apply to a particular problem, in the application of the control theory or in the selection and determination of the model parameters. The probability of this cause of failure occurring can be assumed to be extremely small (except when attempting to expand the usage of a theory beyond currently acceptable applications) and can be alleviated through extensive testing and simulation.

Plant failure is a failure in the system's structure which changes the mathematical description of the plant. To prevent this from causing instabilities in the system, the control theory must be robust enough to allow changes in the plant parameters or even in its form (such as the order of the plant). This requires the use of robust control theories such as high gain feedback, adaptive control, or non-linear control. A common technique used to control robot systems is known as *feedback linearization,* which requires complete knowledge of the physical plant model which is used to cancel out the nonlinearities inherent in robot systems [36]. Feedback linearization is not reliable in this context since it is sensitive to the accuracy of the plant description. Failures will alter the plant model, making the feedback linearization technique susceptible to instabilities if a plant failure occurs. The use of non-linear techniques, such as sliding-mode control which utilize the plant model directly, are more reliable in the face of plant altering failures [145]. Adaptive control algorithms also contribute to good control reliability by altering the plant model to miminize measured response errors. However, adaptive algorithms are susceptible to plant failures which can cause instability if the algorithm cannot correct for the parameter changes caused by the failure [11].

A related cause of instability are sensor failures (see Section 4.3.2.). As seen in Figure 4.6, the sensor is an integral part of the feedback loop in the control system. Without the sensor information, the system will not be able to measure the error and correct for it. Controllers that can tolerate sensor failures are generally well known, since the easiest failure assumption to make is one of sensor failure. Many approaches generate estimates of the sensor data based upon other information available to the control system. This approach is successfully used in adaptive algorithms as well [11].

One other source of failures in the control algorithm exists: the proper implementation of the control algorithm in code. This is one aspect of the software reliability of the controller and involves the "correctness" of the control algorithm code. Liang, Abolrous, and Hussseny have developed a method of quantifying the reliability of a software code based upon the complexity of the code called Logic Structure Reliability Analysis (LSRA) described in Chapter 2 [84]. This does not include clerical and syntax errors which would be detected by a good compiler. It does include the structure of the program and the faults included in the structure. These may include improper branching and incorrect data manipulations. While presented in the context of a controller software, this methodology can be used to quantify the reliability of any code. The application of LSRA can provide several design guidelines. The first is that simpler algorithms will increase the probability that the program will execute properly. Another desirable feature is the ability to transfer control of the program to alternate modules when a module fails. This can be coupled with redundant modules to give even higher levels of reliability. Liang, et. al. also suggest the top level control program be a simple as possible with redundant modules to provide for the highest reliability of the control software. An extremely attractive feature of LSRA is that it can easily be used to provide conservative estimates for software reliability during the design phase, and is not dependent upon historical data for estimation of the program's reliability.

4.3.9. Software Reliability [79, 140]

Without the controlling and operating ftware, a robotic system is useless. Reliable, correctly operating software is essential for the operation of reliable robotic systems. Several types of software reliability models are discussed in Section 3.3.4 as well as the LSRA of the previous section. What has not been mentioned are some techniques that will enhance the reliability of software.

4.3.9.1. Software Reliability Enhancement Techniques

Software reliability is enhanced by the application of good software engineering techniques [140]. One of the most important tools is *structured programming*. Structured programming is a software design philosophy that is characterized by a step-by-step sequential development from functional specification to final product. The preferred approach is a top-down strategy, developing functional layers one at a time. The programs are designed such that they process instructions in sequential order. The structured programming philosophy prohibits the use of an unconditional branch (the GOTO statement) since it breaks up the sequential flow of the program as well as introduces the possibility of uncertain program behavior since an unconditional branch supersedes any data transfer protocols used in the program. Structured programming also promotes modularity in programming. Program modules, called from a supervisory program (that is sequential in nature), allows for the separation of functions and operations.

Functional modularity allows independent testing of the modules and allows for parallel development of the functions. A related technique is the use of redundant software modules. Similar in concept to hardware redundancy, software redundancy implies different codes written by different programmers performing the same function. The different development effort is important since two computers running the same software (redundant hardware, not software) will be subject to the same software failures [140].

If one assumes there will always be errors in code (an assumption similar to a minimum flaw size in mechanical components), it is appropriate to include error checking routines to allow for the detection and recovery from erroneous states. This may be as simple as checking to see that an appropriate data entry is made to a complex method of checking for corruption of memory addresses. Once an error is discovered, recovery can be determined and reconfiguration (either hardware or software) can occur to correct the fault.

Another technique that will reduce the probability of software failure is the use of standard structures and subroutines. Many subroutines are available for immediate incorporation into software that have already been tested and certified error-free. In conjunction with modular program structure, this allows for portions of the program to be assured error free, thus increasing the software reliability. Additionally, user written modules should be utilized to the maximum extent. Not only does this conserve the amount of effort expended in development, it also reduces the amount of code and reduces test time.

Probably the most important technique to the improvement of software reliability is the use of Computer-Aided Software Engineering (CASE). CASE is also sometimes referred to as the automatic generation of program code. The CASE system will take the program specification, as input by the user, and generate an optimized, error free code based on that specification. This prevents most logic errors since the logic can be automatically checked during input and it will prevent all structure errors such as wrong branches and logic loops. CASE has the added advantage of removing the development burden from the system designer, allowing him to concentrate on other important system level issues.

Table 4.13. Software Reliability Enhancement Techniques

Structured (Sequential) Programming	Top-Down Programming
Modular Program Structure	Error Checking and Recovery
Use Standard Subroutine Libraries	Reuse Tested Code Modules
Redundant Software	Computer-Aided Software Engineering (CASE)

4.3.9.2. Robot Programming Languages

There have not been any actual studies of the effect of robot programming languages (RPLs) on the reliability of the programs or the robotic systems themselves. There have been several papers examining the problems in current commercially available RPLs and what directions future RPLs will pursue [58, 101, 162]. This section will identify and discuss some of these problems and try to address some of the concerns.

There are four generally accepted levels of robot programming languages [58; 162]. These levels are joint, manipulator, object, and task. The joint level languages require a description of a task to be in terms of the commands to the motors and actuators. This requires the user to know the solution of the system in joint space (the inverse kinematics) and the behavior of the motors themselves to be able to produce an effective motion program. The second level is the manipulator level. This level of motion programming lets the user program a task in the Cartesian end-effector space. A motion program is designed and the system generates the joint paths required to meet that trajectory. It is at this level that the inverse kinematics and problems involved in their calculations come to the forefront. The third level is the object level. It is at this level the trajectory first starts interacting with the task environment. The RPLs of this class have knowledge about objects in the world, although full knowledge may not be required. The highest level is the task level. The task level can execute the task directly from the task description and able to integrate the robot system task fully into environments and processes. As the level of language rises, so does the complexity and the difficulty of the software reliability problem.

During their investigation of a method of programming robots independently of the actual robot hardware available, Miller and Lennox stated eight significant problems with current RPLs [101]. While some do not directly impact the reliability of the program written in the programming language directly, others have an immediate impact. The problem mentioned that directly impacts the robot program reliability is a lack of error-handling and recovery code in the RPL. This software reliability enhancement technique must be included outside the RPL environment and as a result, the NATO Workshop of Robot Programming Languages stated that the results of executing a robot program are often not repeatable [162]. An additional cause of unreliability is the lack of the ability to incorporate sensor data into the motion program in the RPL environment. This tends to force the programs to be unreliable and error-prone [58]. The same program can work well hundreds of times and fail because of a small difference in the size and orientation of one part. In addition to these specific problems, robot motion programs are susceptible to all the failure modes of software programs in general.

Gini suggests that the way to overcome these problems is to develop robot programming languages at the task level (which had not been implemented at the time of this report) that include object oriented world models in their programming environment. Additional components that are necessary are the ability to measure changes in the world model through sensor data and the use of artificial intelligence to allow for uncertainty in the environment. The sensor data required will be vision sensing systems and position and orientation sensing for parts and assemblies. Object-oriented programming systems are proving to be reliable ways of integrating robot systems into the world model [101]. Object-oriented design allows the programming environment to be independent from the robot hardware, with the interfaces rigidly defined to allow the same code to easily operate different robot systems. This allows extensive testing of the program once and the reuse of the code on different systems just by changing the interfaces.

Table 4.14. Techniques to Enhance the Reliability of Robot Programming Languages [58, 101, 162].

Include Sensor Data into Programs	Object-Oriented Prog. Environment
Include Task-level World Models	Inclusion of Artificial Intelligence Systems
Robot Independent Programming	Other Software Reliability Techniques

4.4. Fault Tolerance

Up to this point, we have discussed reliability and methods to improve the reliability of a modular robotic system. An alternate viewpoint accepts the fact that failures will occur, despite the efforts taken to protect against it, and requires systems to be designed to tolerate failures (*Von Newman's Dictum* [163]). The development of fault-tolerance in robotics systems is of greatest interest when the systems are contemplated for use in environments that prevent maintenance or where maintenance is extremely expensive. The most obvious place where this is the case is in space, both in attended and unattended flight modes. It should come as no surprise that the National Aeronautics and Space Agency (NASA) is the at the focus of the development of fault tolerant technologies for robotic systems. NASA has defined robotic fault tolerance as the capability of a robotic system to sustain failure and still continue operations without significant impact on manipulator payload or its immediate environment [148]. This section describes the fundamentals of fault-tolerance, its components, methodologies, and types and then presents a possible ways of prioritization of the fault-tolerant schemes considered for incorporation into a design.

4.4.1. Methodologies

Avizienis stated three fundamental aspects of fault-tolerance: the systematic identification and characterization of the set of faults to be tolerated; the development and choices of redundancy techniques which provide protection against the faults; and analytical sand experimental prediction of the effectiveness of the redundancy techniques [12]. From these aspects, we can identify four stages of the fault-tolerant mechanism: fault detection, fault confinement, fault recovery, and fault treatment. Fault detection is the how the system recognizes and identifies a fault. Fault confinement refers to controlling the propagation of the fault (preventing a system wide failure). Fault recovery and treatment refers to either the repair or reconfiguration of the system which can be manual or automatic.

Any fault tolerant scheme must utilize redundancy to allow for the system to tolerate a fault and be able to continue operation. There are five types of redundancy that can be used in fault-tolerant systems. The first type is hardware redundancy. Hardware redundancy can take several forms, one of the most common is the Triple Modular Redundancy mentioned in Section 4.2.1.2. This can be generalized to N-Module systems or can take the form of switched standby of *k out of n* arrangements. The next type of redundancy is software redundancy. Redundancy can be included in software by using consistency and capability checking within the code. Additionally, several versions of the same program can be written by different persons (on the premise that they will not contain the same errors) to the same specification that can run in parallel to guard against software failure. This is known as *n-version programming*. The third type of redundancy is called analytical redundancy. This consists of a software implementation of a model of the hardware that runs in parallel to the actual hardware. The response of this model can be compared to the response of the hardware to detect when failures occur as well as to estimate data in case of the loss of sensors. This model can also be an integral part of an adaptive control algorithm to a manipulator system, which makes this type of redundancy much more cost effective. The fourth type of redundancy is information redundancy which is the addition of redundant data to detect corruption as well as to allow for restoration of the corrupted data. Information redundancy also includes error coding and check sum error detection. The last type of redundancy is time redundancy. Time redundancy is repeating an action in time, such as the repeat of a particular motion or the retry of a computer operation. A time redundancy may even go as far as starting a task over or resetting a computer.

The way fault-tolerant schemes operate is detecting a fault, then using the redundancies above that have been built into the system, correct the fault by rewriting data or reconfiguring the system's hardware. In all cases, the fault detection is dependent upon the proper operation of the software the algorithms are implemented in. This means that software reliability is a critical consideration in fault-tolerant design. Another consideration is the required redundancy. In all cases, the addition of redundancy adds complexity, cost, and usually weight. In general, the more complex a system is, the more unreliable it will be. A design trade-off must be made against the cost of adding fault-tolerance to a system, and the effectiveness of the fault-tolerant scheme. This problem is addressed in Section 4.4.3.

There are several methods used for Fault Detection and Isolation (FDI). The first method is model based using analytical redundancy. This is the most common method of FDI and is used quite often in adaptive fault-tolerant control systems. This method examines the responses of the plant (the robot hardware) and compares it to the predicted responses of the software model of the plant (called a residual). If the residual passes a certain level, the system assumes a failure has occurred and it then initiates isolation and recovery actions. Another method is model free and is based on redundancy and limit checking. This is the method used when a TMR voter detects a inoperative input which automatically localizes the malfunction. Other methods include condition monitoring, signature analysis, vibration monitoring, and motor current monitoring. Other model free methods involve expert system based diagnostics and artificial neural network diagnostic systems.

Once the fault has been recognized and located, the system must be able to recover from the fault. This is an architectural issue in which sufficient resources must be available to continue operation. An investigation at the University of Texas has proposed a fault-tolerant robotic architecture [148]. This architecture consists of four levels and concentrates on the manipulator structure itself. These four levels can are listed in Table 4.15.

This architecture, coupled with a system to provide for detection, isolation, and reconfiguration, will provide for two-fault tolerance in the robotic system, regardless of the level of fault. Level I provides for a dual actuator arrangement (such as shown in Figure 4.2). If a failure occurs in one of the two motors, the other motor can be configured by the controller to pull the load. At this level, the failure recovery requires decoupling the two halves of the actuator and ensuring free movement, as well as increasing the torque output of the working half.

Table 4.15. Four Levels of Robotic Fault-Tolerance [148]

Level I.	Prime Mover Duality
Level II.	Parallel Structures with Excess Prime Movers
Level III.	Excess Overall Degrees-of-Freedom (Redundant Manipulators)
Level IV.	Dual Arm Manipulator Systems

Level II suggests that parallel structures be used with extra actuators added to the structure. If one actuator fails, the system will still have enough independent inputs to maintain partial mobility. Failures of the motors at this level must be free-wheeling to enable the system to recover. This ability can be built in as well. Level III is the provision of adding extra DOF to the manipulator (kinematic redundancy). A 7-DOF serial manipulator can tolerate the failure of one joint (locked) and still be able to partially complete its task requiring 6-DOF. The final level (Level IV) is a complete redundancy of the entire manipulator system as a dual arm system. This would allow failure to be tolerated at the task level as well.

4.4.2. Impact on System Reliability

A system can be fault-tolerant yet have low reliability [70]. Consider the TMR system of Figure 4.2. If the reliability of all the modules are the same, and we assume the voter is perfect and will not fail, the reliability of the TMR system can be written as

$$R_{TMR}(t) = 3R^2(t) - 2R^3(t) \tag{4.12}$$

where *R(t)* is the reliability function of one module. Equation (4.12) is plotted in Figure 4.10. Only as the module reliability moves above 0.5 does the TMR system have better reliability than just one module by itself. However, the TMR system can tolerate one failure. While fault-tolerance does have a direct effect on the system reliability, one must be sure the complexity does not force too high a price to pay during the design of the system.

In the previous example, we assumed the voter could not fail. In reality, the voter is part of the system and is subject to the same failures as the rest of the system. Thus, we must be able to quantify the effects of failure of the failure identification system itself. This is described by the *fault coverage* of the system. Coverage can be defined as the conditional probability that a fault is successfully detected given that the fault exists [70]. It is the ability of the system to detect the malfunctions that makes the fault-tolerant scheme work. As the reliability of the system gets very high, the coverage takes on even more importance and the reliability modeling of the FDI system (which determines the coverage) becomes of the highest priority. Consider again the TMR system. If we assume that the only fault-tolerant feature is the voter, the fault coverage of the TMR system can be expressed as the reliability of the voter. In this case, the voter is in a serial reliability structure with the three modules and the reliability function can be stated as

$$R_{\text{TMR w/Coverage}}(t) = C[3R^2(t) - 2R^3(t)] \tag{4.13}$$

where C is the coverage and in this case $C = R_v(t)$, the reliability of the voter. This example also illustrates the fact that coverage can be time dependent, however, it is usually assumed constant for mathematical tractability [70]. As the coverage goes down, the system reliability suffers, as can be seen in Figure 4.11.

The point to be made here is that as systems grow more and more reliable and as fault tolerance is built in, the reliability modeling of the FDI system becomes the key to successfully determining the ability of the system to perform adequately. There have been quite a number of studies on the reliability analysis of fault-tolerant systems [57, 65, 70]. All emphasis has been placed on determining the reliability bound on the system to insure meeting the specifications. This is an extremely hard task in itself, since the reliability models for fault-tolerant systems can have ratios of the largest failure rate to smallest failure rate on the order of 10^{10}. This is termed a *stiff* system. Numerical techniques are generally inadequate for solving reliability models of stiff systems and no way has yet been found to solve a system with over 10^5 states [57].

If the specification is not for a specific reliability, but to tolerate failures, the question arises as to which fault-tolerant scheme to use. This leads us to the prioritization of fault-tolerant schemes to assist in the design decisions.

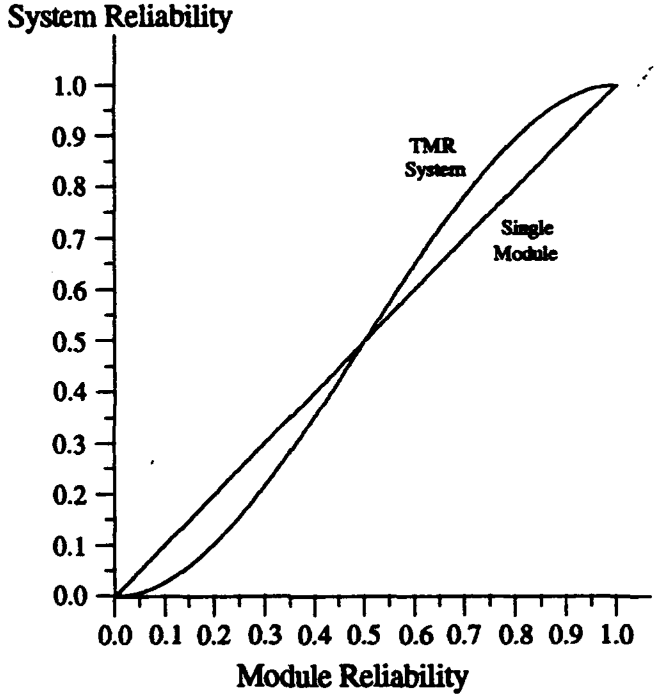


Figure 4.10. Reliability of TMR System as Compared to a Single Module [70]

4.4.3. Methods for Prioritization of Fault Tolerant Strategies

Due to the cost and complexity involved in adding fault-tolerance to a system, the designer will generally opt for a single scheme to provide for fault-tolerance of the system. This consists of selecting an architecture and FDI methodology. The design question faced first is what is the "best" way to implement fault-tolerance. This depends upon the specification of the system and the possible choices available to the designer as well as the costs involved in the various realizations.

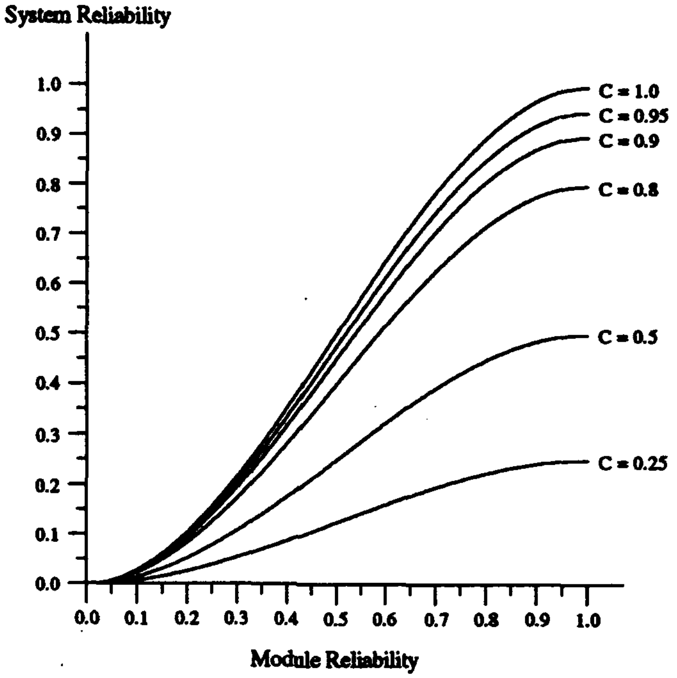


Figure 4.11. Effect of Coverage on TMR System Reliability

There are several ways to rank fault-tolerant strategies for implementation. The easiest way is by the cost of implementation. The redundancies needed to supply system resources for reconfiguration will have a cost associated with their design, as will the development of the algorithms and their implementation and testing. These costs can be estimated and the least costly alternative can be chosen. This strategy does not address the effectiveness of the fault-tolerant schemes under consideration.

Another possibility is to examine the reliability of the system. If good product reliability improvement techniques have been used during the preliminary design process (see Chapter 3), there will exist preliminary reliability and durability specifications for the system, and a preliminary reliability model under development.

The different fault tolerant schemes can be added to the model and their effect on the system reliability can be determined. The scheme that increases the overall system reliability the most can then be identified. If the system specifications are in terms of a required system reliability, this alternative will provide for the selection of the best scheme. However, if the specification is for fault-tolerance (say two-fault tolerant), then not only should the reliability of the system be evaluated, so also must be the effectiveness of the fault-tolerant schemes. By definition, this effectiveness is the coverage of the system.

The coverage of the fault-tolerant system is affected by the definition of the faults against which you wish to protect the system from, the sensor reliability, and the controller software reliability. As stated before, the reliability model and estimate of the FDI system will determine the coverage. Unfortunately, this is usually the hardest part of the reliability model to quantify [57]. A good fault tree analysis of the FDI and related components will provide the baseline information needed to develop the coverage for the different schemes [161]. The failure modes, effects, and criticality analysis will also provide important information to assess the coverage. For all the trouble in developing coverage information, it will provide the most information on which to base a decision. It has the advantage of being a conditional probability. This allows a different value for each specified fault which may be the critical factor, especially if a particular fault must be tolerated.

Table 4.16. Methodologies for Prioritization of Fault-Tolerant Schemes

Ranking Method	Difficulty	Effectiveness
Cost	Easiest	Least
Reliability Modeling	Hard	Better
Coverage Analysis	Hardest	Best

4.5. Rankings for Research Emphasis

Now that we have examined the different technologies that make up the reliability of robotic systems, we need to examine them to see which are the most important and should capture and hold our attention if we wish to improve the reliability of robotic systems. One way that has been acceptable in the past is to subjectively rank the technologies involved and then examine the conclusions one can make from the results. This allows the important characteristics to filter towards the top during this analysis and presents a way of prioritizing interest. This method is useful when examining many different criteria, characteristics, technologies, and applications of the systems of interest. However, in this case we are interested in only two: reliability and maintainability and the effect of the various component technologies that make up robotic systems affect the overall system reliability. During this analysis, we assume a constant failure rate and use the failure rate aggregation method of reliability prediction promoted by MIL-HDBK-217 and the Navy's *Handbook of Reliability Prediction Procedures for Mechanical Equipment.*

4.5.1. Robotic Application Reliability Requirements Rankings

Several reports generated at the University of Texas listed reliability as one of the most important characteristics of robotic systems in a myriad of applications [35, 115, 149]. Table 4.17 shows the applications listed in [35] in order of the importance reliability plays in the accomplishment of the application. Table 4.18 is a listing of the same applications in order of the importance of maintainability. The numerical values assigned represent the author's opinion based on experience and the knowledge gained during this study. One can see that reliability is most important in those applications where maintenance is hard to accomplish while the lower reliability requirements are in those areas where maintenance can be performed with little problem. This is generally intuitive if one keeps in mind the relationship between reliability and maintainability when expressed as availability (Equation 3.2).

4.5.2. Component Technologies

In order to assess the reliability technology needs of robotics system, a baseline should be established by which to measure the contribution of the technologies to the reliability, and where effort should be concentrated to improve the reliability of the system. To do this, consider a generic 7-DOF modular manipulator. This manipulator can be assumed to have the following components:

One base

7 actuator modules (assume dual motors)

14 position and velocity encoders - one for each motor (one sensor provides both position and velocity information)

6 Links - requiring 13 separate interface connections

One end-effector

One communication bus and one power bus

Now, we will use the reliability estimates from each technology group in this chapter and assuming the most reliable technology of each group is used, generate a reliability estimate for this generic manipulator based on failure rates. This model does not take into account the computer used to control the system nor the reliability of the controller or control algorithm. This is only a baseline estimate, such as one that may be developed during the preliminary design phase. This failure rate model is shown in Table 4.19.

Table 4.17. Robotic Application Areas Ranked by Importance of Reliability [35, 149].

Robotic Applications	Reliability Importance Ranking
Space Operations	10
Microsurgery	10
Nuclear Reactor Maintenance and Fuel Handling	10
Service Robots	7
Undersea Operations	7
Complex Assembly Operations	5
High-Speed Precision Assembly Operations	5
Military Battlefield and Airfield Operations	5
Heavy Material Handling	4
Light Machining	4

Table 4.18. Robotic Application Areas Ranked by Importance of Maintainability

Robotic Applications	Maintainability Importance Ranking
Military Battlefield and Airfield Operations	10
Service Robots	10
Nuclear Reactor Maintenance and Fuel Handling	9
Microsurgery	8
Space Operations	8
Complex Assembly Operations	7
High-Speed Precision Assembly Operations	7
Heavy Material Handling	7
Light Machining	5
Undersea Operations	4

The generic manipulator reliability model predicts a failure rate of 421.4 failures per million hours. This gives a MTBF of only 2373 hours, which is commensurate with most industrial robotics found on the market today. This extimate is extremely conservative due to the nature of the parts count reliability prediction method used in this example. If a Reliability Block Diagram model is used and the duality of the actuators is acknowledged, the MTTF is over 5000 hours. If the controller and software reliabilities are included, the reliabilty of the system will be reduced as well. The information above can also be listed in the order of contributions to the failure rate as in Table 4.20.

Table 4.19. Failure Rate Model for a Generic 7-DOF Modular Manipulator.

Component	Qty	Unit Failure Rate	Total Failure Rate
Base and Links	7	0	0
Motors - AC brushless	14	1.33×10^{-6}	18.62×10^{-6}
Sensors			
Pos/Vel Encoders	14	0.32×10^{-6}	4.48×10^{-6}
Hall Effect	14	0.59×10^{-6}	8.26×10^{-6}
Force	18	0.0118×10^{-6}	0.212×10^{-6}
Communication Syst.			
Coaxial Cables	7	0.0241×10^{-6}	0.1687×10^{-6}
Power Cables	7	5.6×10^{-6}	39.2×10^{-6}
Connectors	26	0.017×10^{-6}	0.442×10^{-6}
Gear Trains - Cycloidal	14	25×10^{-6}	350×10^{-6}
Total Failure Rate			421.4×10^{-6}

Table 4.20. Component Technology Impact on Reliability of Generic Manipulator

Rank	Component Technology	Failure Rate (per 10^6 hrs)
1	Gear Trains	350
2	Powei Cables	39.2
3	Motors	18.62
4	Sensors (aggregate value)	12.74

We can now immediately see that if the gear trains are removed and replaced with direct drive motors (if feasible by the performance specifications), the MTBF value will increase almost 500% to 14,000 hours. The important conclusion is that gear trains are a major target for the imporvement of robot reliability. This is true purely on the basis of precision as well. This is the type of information needed to prioritize research emphasis. The estimates made above are extremely conservative and should not be considered for prediction purposes without much more in depth evaluation according to [23] and [131]. However, the best use of this type of data is during design, when trying to compare and choose between competing design solutions.

4.5.3. Methodology Prioritization

The same difficulties occur when trying to prioritize the tools and methodologies used to promote the reliability during the design. A prioritization can be made, however, by examining what the designer needs during the design process. During preliminary design, the first aspect of reliability is the preliminary reliability model developed to help the designer select between design alternatives. These first reliability models are usually made up of black box reliability block diagrams. These diagrams allow the designer to visualize the dependencies the system's reliability will have on its components and the interactions between the components.

After some design decisions have been made, such as structures and architectures, more in-depth reliability analyses are made and probabilistic importance can start to be attached to failure modes. These analyses, FMECA and

FTA, are extremely valuable in making the detail design choices and the data generated here should be used to influence the choices of fault-tolerant schemes as well. Further discussion on the application of these methodologies can be found in Chapter 3 and Appendix A. Table 4.21 presents these reliability design methodologies in the order of their application (See also Figure 3.9) and in order of increasing complexity.

Table 4.21. Reliability Analysis Methodologies by Order of Application during Design

Reliability Block Diagrams
Failure Mode, Effects, and Criticality Analysis (FMECA)
Fault Tree Analysis (FTA)
Fault-Tolerant Scheme Prioritization

4.5.4. Research Emphasis Suggestions

The analysis presented in this section and elsewhere in the chapter allow several conclusions to be drawn about the direction future research in component technologies should go. The first two conclusions can be seen directly from Table 4.20 as represented by the failure rate of the gear trains. As stated before, removing all of the gear trains from the generic manipulator will increase the reliability prediction by 500%. Two things must occur for this failure rate to go down or be eliminated. First, motor technology can be improved to reduce the demands on the gear trains. Complexity needs to be reduced and torque levels raised so that the motor characteristics support direct drive technologies. This will reduce the failure rates of the motors as well. Second, the failure rate of the gear systems can be reduced directly by removing and reducing the causes of failure in the gear system. The main effects of gear failure are generally wear related which points toward improving tribology techniques and materials. An additional factor to reduce the gear failure rate is by derating the gear system (See Section 3.3.1.2 for a discussion on parts derating). The failure rate of the gear systems can be reduced by using them well below the rated loads and torques. However, this implies a large amount of overdesign which will penalize the design by adding additional weight and inertia. These factors must be traded off against the reduction in failure rate. An adequate compromise between these two positions is by improving motor technology, the burden on the gear trains will be reduced and will allow less overdesign which will reduce the gear system failure rate.

The second contributor to the failure fate is power cables. These cables usually fail by bending, or abrasion. A modular architecture as discussed in this thesis will alleviate all or most of the abrasion and bending stresses currently experienced in monolithic robot systems since the cabling will be an integral component of the modules. While the numbers of connectors will increase by preventing continuous cable bundles, the connector failure rates are minuscule when compared with the cable rates.

A topic discussed in Section 4.3.1.2 is the possible use of self-sealing fluidic connectors to allow active cooling of the motors and power electronics, as well as pneumatic operation of the end-effector. Using the reliability levels stated in Table 4.3, the failure rate for all the fluidic connectors required for the generic robot would be 113 failures per million hours. This component of the failure rate can be removed by requiring the end-effector to utilize the standard module interface using the data bus and electrical power.

The last significant component in the failure rate of the generic robot are the sensors. There where a total of 44 sensors assumed for this generic robot. This is not an unusual number when compared to current industrial robot systems. By combining advanced estimation techniques while reducing the number of sensors, this component of the failure rate can be reduced. The use of combined sensors (such as resolvers and encoders providing both position and velocity data) can decrease the failure rate by reducing the number of components required on the arm. The penalty that needs to be weighed in this instance is the introduction of a single point failure mode.

An overall problem that affected this entire investigation is the lack of specific robotic component reliability data. Most of the manufacturers contacted during the survey did maintain a reliability data base, but generally where reluctant to release any specific data regarding their product and components. This was due to proprietary and competitive reasons. The result is that there is not a readily accessible data base on reliability data for components in a robotic environment. This data is essential during the design of a robotic system in the accomplishment of the reliability prediction tasks. This problem should be the priority for any robotic research plan to improve the reliability of robotic systems.

4.6. Summary

This chapter has presented the technology side of the reliability improvement equation. We have seen several different analyses providing us insight into the problems faced when attempting to improve robotic system reliability. To finalize the arguements and to provide a ranked list of design issues and technologies, Table 4.22 shows a cross reference between the major reliability design issues for modular robotic system reliability and the component technologies.

During the material presented in Chapters 3 and 4, we can identify four basic issues that directly pertain to the reliability of modular robotic systems. First is the complexity of the module. As a general guideline, the more complex a system, the lower the reliabiltiy. In the modular robotic system case, the entire complexity of teh system will reside in the modules and the manipulator itself will be fairly simple to assemble and maintain. To improve the modular robotic system reliability, the complexity of the modules must be minimized. The rankings from 1 to 10 shown in Table 4.22 provide a subjective estimate of the contribution a particular component technology makes to the complexity of a module with a ranking of 10 contributing the most.

A related issue is the complexity of the modular robotic configuration. This complexity is at the system level and is traded-off against the module level complexity. The system level will grow more complex as the modules become simpler and vice versa. A contributing factor, module interface standradization, contributes positively to both complexities and is ranked as well. The final issue is the data issue, with the rankings tracking the generic trend of the component technologies.

Table 4.22. Reliability Design Issues and Component Technologies Rankings

	Reliability Design Issues				
Component Technologies	Configuration Complexity	Module Complexity	Reliability Data	Interface Stanardization	Totals
Comm.	10	10	7	10	37
Sensors	9	9	8	9	35
Software	10	9	6	5	30
Controllers	9	8	6	5	28
Motors	5	7	9	5	26
Gearing	5	6	10	5	26
Links	9	1	1	5	16
Brakes	3	3	2	3	11
Totals	60	53	49	47	

From Table 4.22, the top four technologies are communications, sensors, software, and controllers. These technologies are what make up the control system of the robot system as shown in Figure 4.4. The mechanical systems which make up the manipulator are shown to be of secondary importance to the reliabiltiy of modular robotic systems. This is due to the importance that is attached to the interfaces of a modular system. For the sucessful implemetation of the modular robotic system, the interfaces must be kept as simple as possiblc ot maximize the system relability. The way to do this is through the minimization of the number of signal and power paths that must transverse the interface. The key to this effort is the commucication and control schemes for the robotic system.

Table 4.23. Priority Listing of Additional Research Needs for Robotic Reliability

Establishment and Maintenance of a National Robotic Component Reliability Data Base
Promote Research and Improvement in Reliability of Robotic Communications Systems and Sensor Technologies
Continue Development of Robust, Fault-Tolerance Control Systems and Algorithms to Provide the Highest Controller Reliability
Improvement in Motor Technologies to Allow Design of Direct Drive Robotic Systems
Improvement in Reliability of Gear Train Technologies
Develop Tools and Methodologies to Quantify and Manpulate Dependent Random Variables to Allow Full Understanding of the Impact the Modularity of a System has on its Reliability

The mechanical system reliability is driven by motor and gear technology. The failure rate of the mechanical system was shown in Sections 4.3.4 and 4.5.2 to be dominated by the gear systems. To improve the mechanical system's reliabiltiy, additional research needs to be performed in actuation technology to reduce the burden on the gear systems, thus reducing the failure rate of the gear systems. An lofty and perhaps unattainable goal is to improve motor capacity enough such that gear systems can be eliminated altogether, not only eleminating thier addition to the system's failure rate, but improving precision and performance as well by removing backlash and allowing feedback through the motors themselves which will allow for "electronic stiffening" of the robotic system, improving precision even further.

The final research prioiritization for robotic reliabiltiy imporvement is shown in Table 4.23. This listing is compiled from the material and analysis presented in Chapters 3 and 4. An overall summary of the roadmap can be found in Chapter 7.

CHAPTER 5: RELIABILITY PERFORMANCE INDEX DEVELOPMENT

5.1. Design and Operational Criteria

The design space of an industrial robot has been shown to have over 150 separate design parameters [22]. During the design of a modular system, this design space is unmanageable using current design methods and optimization schemes due to the necessary functional independence between modules. One way to reduce the complexity of the design and allow the designer to fully understand his design choices is to collapse the design space by using design criteria. These criteria are systematic representations of the design variables. By arming himself with the knowledge of how certain criteria (such as those found in Table 5.2 for design and in Section 5.1.1.2 for decision making) behave under design influences, the designer can then select the design variables which can provide the best criteria values, either through judgment and experience, or by developing a design objective function and performing a design optimization.

Criteria have two distinct applications in robotic technology. First, as mentioned in the previous paragraph, design criteria helps the designer integrate vast amounts of design information into a smaller, more manageable design space for synthesis. Design criteria are most useful if there is a direct relationship to the variables under the designer's control. An analytic relationship between the design criteria and the kinematic or dynamic model of the robotic system is preferred, however, empirical usage has shown to be effective as well [4].

The same criteria may be used to formulate command and control algorithms for intelligent control. This is commonly referred to as decision-making and is perhaps the only effective method for controlling redundant (excess Degrees-Of-Freedom (DOF)) manipulator systems. By monitoring the criteria on-line, the controlling software can optimize the path and/or self-motion in the manipulator to minimize the kinetic energy of the system, avoid sensed or pre-preprogrammed obstacles, prevent the manipulator from assuming singular configurations, allow for fault-detection and isolation as part of a fault-tolerance scheme, etc. This section provides a short overview of these design and decision-making criteria and how they are used. It also discusses the need for a system level performance index, which is subsequently developed in the later sections of this chapter.

5.1.1. Current Criteria Applications

The University of Texas Robotics Research Group has been very interested in the development of criteria for use in the intelligent control of redundant manipulator systems and have issued several research reports dealing with the development and use of these criteria [16, 28, 35, 157]. During these investigations, several characteristics of good criteria have been recognized (see Table 5.1).

The first characteristic of a good criterion is that it have a unique physical meaning. In other words, the designer must be able to recognize satisfactory values or trends in the criterion and be able to interpret its physical meaning relative to the design variables of the system. At best, the criterion should have a direct analytic mapping to the design variables, having been constructed from them in the first place. However, an empirically based criterion can provide successful results as well, especially when used for decision making rather than design.

The second characteristic of a good criterion is that it should contain as much information or represent the most different physical meanings as possible in the one numerical value. This results from the need to collapse the design space into a manageable size. Criteria based decision making or design reduces the complexity of the problem by reducing the size of the decision or design space by synthesizing meaningful design and decision making criteria to take the place of the design or decision variables.

The third characteristic of a good criterion is that it should be as simple as possible. This may be perceived a contradictory goal to the second characteristic of having as much information as possible in the criterion. This implies a trade-off that must be made during the selection of the criteria with which the system is to be designed or controlled. This is not so much a constraint on design criteria as it is on decision-making criteria. This is because of the different environments these two activities are performed in. Design is an off-line activity, not requiring speed except to avoid long delays while performing numerical searches on the criteria. In the design case, the complexity of the criteria is secondary to the amount of information embedded since time is not critical, while space reduction is. In the operational decision-making environment however, the speed at which the criteria values can be calculated becomes dominant when this system is under real-time control. The real-time environment requires simple, robust calculations to provide for adequate coverage and sensing of the decision-making criteria.

Table 5.1. Characteristics of Good Design and Decision-Making Criteria

Direct correspondence to design variables
Have multiple, unique physical meanings (contain maximum information)
Be as simple and computationally inexpensive as possible

5.1.1.1. System Design Objectives: Integration and System Optimization

Ambrose and Tesar developed a modular robotic system testbed where a system of four joint modules and nine link modules where designed and implemented [4]. One main focus of the report is the choice of global criteria on which to base the design and selection of the configuration of the modular system. Ten overall objectives where selected out of thirty-eight possible design goals. These ten objectives are listed in Table 5.2.

Table 5.2. Ten Design Objectives used for Module Design and Arm Configuration Design by Ambrose and Tesar [4]

Mobility	Payload
Speed	Motion Range
Weight	Backlash
Static Friction	Stiffness
Inertia	Servo Bandwidth

The formulations of each of these design objectives are included in [4] and are developed at both the module and arm level. The module level objectives are based upon the design variables of the modules themselves, such as the motor weights, torque capabilities, gear backlash, etc. The arm level objectives are made up of combinations of the module level objectives according to how the modules are assembled. Ambrose and Tesar performed experiments on the modules to determine their characteristics, then used the arm level formulations to optimize the arm configurations. They did not include (explicitly) any precision criteria, such as accuracy and repeatability (although they are both effected by the backlash and stiffness of the system), nor was there any attempt to include any durability goals into the system design.

An effort by Carnegie-Mellon University to address similar considerations used even fewer design objectives [80, 119]. The only objectives mentioned in these works are manipulator workspace (for the selection of the kinematic arrangement and link lengths), payload, and joint velocities and accelerations (to select the actuators). They do not address system integration or system durability issues.

5.1.1.2. Decision-Making Criteria

Two of the most recent works at the University of Texas have proposed six categories of operational performance criteria that are useful for decision-making in redundant manipulator systems [28, 157]. Additionally, Cox and Tesar have listed 28 possible operational criteria that can be used for intelligent control (see Table 1.1). These 28 criteria have direct correspondence to the criteria contained in the six categories. The six categories are

1. 1st Order Geometrical Criteria
2. 2nd Order Geometrical Criteria
3. Inertial Performance Criteria
4. Kinetic Energy Criteria
5. Compliance Performance Criteria, and
6. Workspace Operation Criteria

Each of these categories will be discussed, but no derivations are presented here. For the complete derivations of each of the mentioned criteria, refer to [28, 157].

First order geometric criteria are based upon manipulations of the manipulator's Jacobian matrix. The primary goal of these criteria are to monitor the approach of the manipulator to its singularity positions. Singularity prevention is important since the motion of the manipulator becomes unpredictable as it moves closer to a singularity. This is due to the fact that the system has extremely large responses to small inputs in close to singular positions. Other first order criteria involve measurement of dexterity, force transmissions, precision, stiffness, etc. These are generally task independent criteria.

Second order geometric criteria are based upon the Hessian array. These criteria measure the rate of change of the approach to the singularities and inform the controller how fast the manipulator trajectory is approaching a singularity. These criteria also include end-effector and joint acceleration criteria to measure the acceleration related properties of the various components of the system. These end-effector motion objectives are generally task dependent.

Inertial performance criteria deal with the inertial torques and forces required from the manipulator's actuators to move the entire structure. These criteria are calculated from the effective inertia matrix as described in [157]. These criteria can be used to measure the dynamic coupling between the different links of the manipulator and monitor the upper limit of the actuator torques or forces. The rates of change of the torques can also be measured to provide for the ability to smooth the torque application. Other inertial criteria deal with the torques induced by the velocities of the actuators and the end-effector.

Kinetic energy criteria can be examined in either the joint space or end-effector space. These criteria measure the total system kinetic energy and provides a tool to minimize the movement of the manipulator (which minimizes the kinetic energy). The rate of change of kinetic energy can also be measured. The kinetic energy criteria are task independent. The kinetic energy content of the manipulator can also be partitioned and examined at each link for its contribution to the overall kinetic energy. The distribution of the kinetic energy is task independent. Knowledge of this distribution allows the controller to allocate the actuator effort over the arm in a more optimal fashion.

Another group of criteria is related to the compliance of the manipulator. These criteria measure the deformation of the manipulator to the static and dynamic loads it encounters. These criteria are based upon manipulations of the robot's stiffness matrix and its resistance to deflection in various configurations. The potential energy stored in those deflections can also be determined and partition values can be assigned to the manipulator's components. Since these measurements are made under deflections, they are load dependent but their distribution is independent of the magnitude of the end-effector load.

The last group of decision making criteria mentioned by van Doren and Tesar and developed in Cox and Tesar is workspace operational criteria. These are the criteria which measure the system level performance. The first criterion mentioned is obstacle avoidance. This criterion allows the manipulator to sense and avoid obstacles in its path. This may be based on an external sensing system, since contact sensors will only indicate collision after it happens. This criterion also provides a measure of the robot's plan of action. Another workspace criterion is the overall dexterity or the ability of the manipulator to assume any orientation in the workspace. Joint range availability is also of interest in this category since it limits the actual workspace of the system. Another issue is precision and how the degradation in precision over time can be measured. The precision criterion has not yet been developed.

5.1.2. The Need for the Reliability Performance Index

The discussion of design and decision-making criteria shows that component and task independent design and decision making criteria have been developed and implemented for any number of uses. What has not been quantified as design criteria are the "operational issues" of precision (accuracy and repeatability) along with the module and system reliability and maintainability. This is a lack that is now felt when attempting to provide for the successful integration of a modular robotic system. Not only do the performance issues need priority, the operational characteristics of the system are just as important. If the system doesn't work at all, it won't matter how optimum it is. What is missing from these criteria are the measures of durability: reliability and maintainability, as well as a direct criterion to measure of the precision of the system.

The main reason for the lack of a precision criterion is that repeatability is a statistical measure of the ability of the manipulator to attain a certain pose* [108]. None of the design or decision-making criteria developed to date have been statistical measures, they are deterministic with no inherent variability. The same observation holds true for the module and system reliability and maintainability. These are also statistical measures of the probability of failure and repair. Thus, it seems reasonable to approach these statistical measures from the same point of view to generate a system level durability criterion. According to the characteristics of good criteria, we would like to combine these statistical measures into an overall durability criterion that has meaning to the system. Reliability theory presents itself as the logical tool since it is well known and has been applied successfully in the measurement of repeatability and accuracy [18]. Since these measures are statistical in nature, there will not be a direct correspondence to the design variables, however, empirical usage will be available. The remainder of this chapter is devoted to the development of this Reliability Performance Index (RPI). First, the accuracy and repeatability portions of the index will be discussed, followed by the module and system durability portions. The final RPI formulation which represents the probability that the system will function as expected will be presented in the last section, as well as architectural concerns and a simple example.

* Pose refers to both position and orientation of the manipulator.

5.2. Kinematic Reliability of Robotic Systems

The accuracy and repeatability of a manipulator system are generally thought to be best represented as dependent only upon the variations in the kinematic parameters of the system. Accuracy is a measure of the difference between the desired (commanded) end-effector position and the achieved position. Colson and Perreira add the qualifier that no memory of the task being performed previously remains [34]. The accuracy of a manipulator can be analytically described as the averages of the mean position and orientation error vectors. A generalized expression for the accuracy was developed by Colson and Perreira and is

$$\overline{\mathbf{A}}_\beta = \frac{\sum_{m=1}^{M} \varepsilon_{\beta m}}{M}, \quad \beta = p,\theta \tag{5.1}$$

where M is the total number of position and orientation measurements, β is either p (for position error vectors) or θ (for orientation errors), and $\varepsilon_{\beta m}$ is the error vector for the mth measurement. The variance of the accuracy is

$$s_\beta^2 = \frac{\sum_{m=1}^{M} \left(\varepsilon_{\beta m} - \overline{\mathbf{A}}_\beta\right)^2}{M-1}, \quad \beta = p,\theta \tag{5.2}$$

The accuracy of a robotic manipulator is effected by the errors in all of the kinematic parameters of the system, both joint parameters (angle and offset) and link parameters (length and twist angle).

The repeatability of a manipulator is similar to the accuracy except that it is a measure of the ability of the manipulator to return to the same position and orientation. Equations (5.1) and (5.2) also can be used to express the repeatability by letting the error vector $\varepsilon_{\beta m}$ represent the error between the achieved pose and the target pose. The repeatability only reflects the errors present in the joint angles since the links of the manipulator do not change length during operation. Thus, the only variability represented in repeatability is the variability of the joint actuators, provided there is no link deformation. Since accuracy and repeatability are statistical measures and represented by mean variances, we can use reliability theory to characterize the probability of achieving a certain accuracy and repeatability. This is the concept that is at the heart of kinematic reliability, R_k.

5.2.1. Definition of $\mathbf{R_k}$

5.2.1.1. Robot Kinematic Formulation Overview

Since both accuracy and repeatability are measures of kinematic error, any statistical measurement must have as its basis the kinematics of a robot manipulator. Kinematics is the study of the geometry of motion; the position, velocity, and acceleration of a body without regard to the forces involved in causing the motion. Robot kinematics have been studied intensely in the past and good references on the subject abound. Two excellent descriptions can be found in Paul and in Craig [120, 36]. An alternate formulation of the kinematic description can be found in Thomas and Tesar [153]. The basis for all these formulations is the Denavit-Hartenberg (D-H) kinematic notation [39]. In the D-H notation, two parameters are associated with each link (a_i and α_i) and two with each joint (d_i and θ_i). The distance d_i and the angle θ_i between adjacent links give the relative position between the links and the length a_i and twist angle α_i determine the structure of the link. For revolute joints, d_i, a_i, and α_i are the structural parameters and θ_i is the joint variable. For prismatic joints, θ_i is a structural parameter and d_i is the joint variable.

The direct kinematic problem is to solve for the position and orientation of the end-effector given the arm structural parameters and the values of the joint variables. Coordinate frames are attached to each link and transformation matrices from link frame to link frame are written in terms of the relevant kinematic and joint parameters. This transformation relating link frame *i* to link frame *i*-1 (known as the D-H transformation matrix) can be shown to be

$$[A_{i-1}^{i}] = \begin{bmatrix} C\theta_i & -C\alpha_i S\theta_i & S\alpha_i S\theta_i & a_i C\theta_i \\ S\theta_i & C\alpha_i C\theta_i & -S\alpha_i C\theta_i & a_i S\theta_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.3}$$

where C denotes cosine and S denotes sine [36]. Using this notation, the end-effector position and orientation can be calculated as

$$[T] = [A_0^1][A_1^2]\cdots[A_{n-1}^{n}] = \prod_{i=1}^{n}[A_{i-1}^{i}] \tag{5.4}$$

where n denotes the number of joints in the manipulator system. [*T*] is called the arm matrix. The end-effector position is found from the right-most column vector of the arm matrix and the upper-left 3 by 3 partition is the rotation matrix from the base frame to the end-effector frame.

5.2.1.2. Kinematic Error Sources

This kinematic formulation does not explicitly recognize the sources of error that can occur. However, as a practical matter, errors will always exist. Most of these errors are not deterministic in nature and should be studied with statistical tools such as reliability theory. Bhatti and Rao and Colson and Perriera present several major sources of error for position and orientation [18, 34]. The first category is manufacture and assembly. Variations in the kinematic parameters can occur due to the tolerances in the manufacture and assembly of the manipulator system. As can be found in any undergraduate machine element design text, no part can be machined exactly to its specification; a tolerance will always exist. Extremely tight tolerances will reduce the errors but will drive up the manufacturing cost as well as complicate the assembly of the system. Many robot manufacturers and researchers attack this problem with expensive and tedious "calibration" methodologies to measure the actual kinematic parameters of the manipulator. Today, it is not possible to apply these techniques to every robot system coming off an assembly line. The most cost effective way to accommodate these errors is to recognize that they exist and design the system accordingly.

The second source of error can be found in the robot's actuators and controllers. The actuators (gear and motor) will generally exhibit backlash and cogging and deform under torque loads to cause rotation errors. The control algorithms can encounter round-off problems which can effect the precision of the system. Feedback can reduce these errors, but is limited by the bandwidth of the system as well as by finite word length within the computer.

The third source of position and orientation error is due to the deflection of the links due to loading and torsional distortion in the joints of the robot. This dynamic error is due to the manipulator handling payloads through time based trajectories.

5.2.1.3. Kinematic Reliability Formulation [18]

In their original work, Bhatti and Rao define three types of positional kinematic reliability and two types of orientational kinematic reliability designated as Types I through V, respectively. They are all based upon a definition of failure that is error based. The failure definition in this case is the end-effector falling outside an error bound around a target position and orientation. This allows the following definition of kinematic reliability: the probability of the end-effector position and/or orientation falling within a specified range from the target position and/or orientation [18]. This definition leads to the development of the error bounds and the formulation of each type of kinematic reliability.

Type 1 Positional Kinematic Reliability. This measure of positional kinematic reliability is concerned only with a specified distance from a work plane. The error bound in this case is based upon the required distance from the plane and the tolerance on that distance. This applies to robotic applications such as spray painting. If we express the workplane in the base coordinate frame as

$$a_1 x + a_2 y + a_3 z = c \tag{5.5}$$

we can designate the desired end effector position (x_d, y_d, z_d) at the desired distance from the plane as

$$\bar{d} = \frac{c_d}{|a|} \tag{5.6}$$

where $c_d = c - a_1 x_d - a_2 y_d - a_3 z_d$ and $|a| = \left(a_1^2 + a_2^2 + a_3^2\right)^{1/2}$. Now let ($x^*$, y^*, z^*) represent the actual position of the end-effector and a tolerance on $\bar{d}$ be given as $\pm\Delta d$. We can now represent the actual distance of the end-effector from the work plane as

$$d^* = \frac{c^*}{|a|} \tag{5.7}$$

where $c^* = c - a_1 x^* - a_2 y^* - a_3 z^*$. The definition of Type I positional kinematic reliability follows as

$$R_{KI} = P(\bar{d} - \Delta d < d^* < \bar{d} + \Delta d) \tag{5.8}$$

Type II Positional Kinematic Reliability. This type of kinematic reliability is useful in quantifying the reliability of assembly operations such as peg-in-hole insertions. This definition of failure restricts the end-effector to a (x, y) point on the work plane but the distance from the plane is not constrained as in Type I. In this case, the motion of the end-effector is considered to be normal to the plane and can be formulated as follows. Define the target point on the workplane as (x_o, y_o, z_o). The permissible region can than be thought of as a cylindrical region symmetrical about the z_o axis perpendicular to the plane. Let Δr denote the radius of this cylinder and (x*, y*, z*) the actual position of the end effector. We can then write the position of the end-effector measured radially from the z_o axis as

$$\Delta r^* = \left[\left(\frac{a_1 c^*}{|a|^2} - (x_0 - x^*)\right)^2 + \left(\frac{a_2 c^*}{|a|^2} - (y_0 - y^*)\right)^2 + \left(\frac{a_3 c^*}{|a|^2} - (z_0 - z^*)\right)^2\right]^{1/2} \tag{5.9}$$

where a_1, a_2, and a_3 are the plane parameters from Equation (5.5). Equation (5.9) represents the offset distance of the center of the hole to the center of the peg (assuming no orientational misalignment). We can now write the Type II positional reliability as

$$R_{KII} = P(\Delta r^* \le \Delta r) \tag{5.10}$$

Type III Positional Kinematic Reliability. The Type III positional reliability is the fully constrained case. There are two possible permissible regions for this type of reliability; a box or a sphere about a target point in space. Type III kinematic reliability is the most restrictive of the positional reliabilities, giving the lowest positional reliability under similar conditions. Of the two types of permissible regions, the sphere will be the most conservative since the total enclosed volume is smaller. For the box region, we assume tolerances on the spatial coordinates of the target point as ±Δx, ±Δy, and ±Δz. We can then write the Type III reliability for this region as

$$\begin{aligned} R_{KIII} = P\{&(x_d - \Delta x < x^* < x_d + \Delta x) \cup \\ &(y_d - \Delta y < y^* < y_d + \Delta y) \cup \\ &(z_d - \Delta z < z^* < z_d + \Delta z)\} \end{aligned} \tag{5.11}$$

The spherical permissive volume can be represented as

$$\Delta r^* = \left[(x_d - x^*)^2 + (y_d - y^*)^2 + (z_d - z^*)^2\right]^{1/2} \tag{5.12}$$

allowing us to write the Type III reliability as

$$R_{KIII} = P(\Delta r^* \leq \Delta r) \tag{5.13}$$

Type IV Orientational Kinematic Reliability. Bhatti and Rao describe two types of orientational kinematic reliability. The Type IV reliability constrains the end-effector orientation to one axis of the end-effector rotation matrix. Recall the upper left 3 by 3 partition of the manipulator matrix of Equation (5.3). This partition is the rotation matrix representing the rotation of the end-effector frame relative to the base frame. This matrix can be represented as

$$R=\begin{bmatrix}\bar{n} & \bar{s} & \bar{a}\end{bmatrix} \tag{5.14}$$

Type IV reliability represents a constraint on one of the column vectors of Equation (5.14). To form the permissible region for the vector, minimum and maximum limits are imposed of the direction cosines of the desired vector. For a complete mathematical description of the direction cosines, refer to Craig [36]. In probabilistic terms,

$$R_{KIV} = P(\text{direction cosines for specified direction within permissible limits}) \tag{5.15}$$

Type V Orientational Kinematic Reliability. Type V reliability is the fully constrained orientation. All three column vectors of Equation (5.14) are prescribed. The Euler angles for these rotations are used as the means and certain deviations can be specified for these means. The Type V reliability is the probability that the actual Euler angles fall within their respective ranges. To completely quantify the kinematic reliability of a system, both the position and orientation reliability will need to be specified. The overall kinematic reliability of a manipulator is the probability of both the positional and orientational tolerances being satisfied at the same time.

5.2.2. Determination of $\mathbf{R_k}$

Now that the types of kinematic reliability have been defined, we turn to methods of determining the robotic system's kinematic reliability. Bhatti and Rao suggest two methods: analytic and simulation. In the analytic formulation, they assume that the end-effector position and orientation follow a joint normal probability distribution. If this assumption holds, one only needs to describe the mean values, variances, and covariances of the end-effector position and orientation to fully describe the probability distribution of the end-effector. The mean of the end-effector position and orientation is found by substituting the mean values of the kinematic variables into the forward kinematic equations (Equation (5.4)). The variances of the distribution can be found from the partial derivative rule (See the example in Section 5.2.3 and [73]). Since the end-effector joint probability distribution is now known, the distribution can be integrated over the permissible region to find the reliability (as described in Section A.1.1 of Appendix A).

Conceptually, the analytic approach is straight forward. However, several problems arise in its practical use. The first difficulty arises during the examination of the joint end-effector probability distribution. Bhatti and Rao invoke the Central Limit Theorem (CLT) which states that the probability distribution of a *linear* function of independent, generally distributed random variables approaches the normal distribution as the number of distributions increases [95]. The key word in the above statement of the theorem is the word *linear*. Bhatti and Rao claim the end-effector has a joint normal distribution for both position and orientation, based on the CLT. The problem here is that robot kinematic equations are not linear; in fact, they are extremely non-linear transcendental functions. Thus, the CLT cannot be used as justification for this assumption. However, empirical evidence does support this assumption. In his dissertation, Bhatti presents simulation results that support this assumption [17]. An investigation by Ramsli also concludes that repeatability of manipulators (a characterization of the probability distribution of the end-effector) while not strictly normal, does possess a normal shape with slightly larger tails and that the normal distribution gives adequate results if no other data to the contrary is available [126]. It is this lack of applicability of the CLT that accounts for the discrepancies between the results Bhatti and Rao report for the analytical and simulation methods of solution.

The second problem with the analytic method is the mathematical tractability and the assumption of independence. A general, dexterous, serial manipulator will have an end-effector joint probability distribution that is hexa-variate having three position coordinates and three orientation coordinates. In general, this distribution, if it can be found at all, will require six different integrations to determine the reliability. This is an intractable problem and provides more incentive for the continued investigation of the algebra of dependent vectors of random variables and the tools with which to manipulate them (see Section 3.5). A complicating matter is the assumption of independence between the different parameters of the kinematic equations. As discussed in Section 3.5, the independence between the lives of modules in a robotic system is suspect, since all the modules are in different places in the load path. The same reasoning can be applied to the independence of the kinematic parameters. The means of the joint parameters are independently adjustable, but the variability can change as the robot moves through its workspace, in fact, the repeatability (which is a function of the joint variables only) has been shown to vary across the robot workspace [108]. To facilitate the development of the reliability performance index, these two problems are acknowledged for further research but we will assume independence of the joint variables as well as constant joint variable variances over the workspace.

The second method of finding the reliability is by Monte Carlo simulation. The basis of this method is in the frequency interpretation of reliability (Equation (A.3)). The random variables of the system are identified, and their probability distributions determined. A random variate of each distribution is generated and substituted into the kinematic equations to generate a value of the end-effector position and orientation. This is one trial. The value of the end-effector position and orientation from the trial will be a sample from the distribution of the joint end-effector probability distribution [133]. This sample is then compared to the target position and orientation values and if it falls in the permissible region, the trial is said to be a success. A large number of trials are then conducted and the success or failure is recorded. Then, the kinematic reliability of the system can be determined as

$$R_{\mathbf{x}} = \frac{\text{number of sucessful trials}}{\text{total number of trials}} \tag{5.16}$$

Bhatti and Rao found that simulation was more accurate for the determination of the kinematic reliability. The reasons they gave have to do with Taylor series truncation and numerical error in integration. The Taylor series truncation occurs during the determination of the analytic variances using the partial derivative rule. Truncation of Taylor series expansions of non-linear functions can add large errors if the degree of non-linearity is very high. This is precisely the reason that the CLT does not apply, and the discrepancy can be explained as the result of this assumption. Simulation is also much more amenable to computer implementation and modeling. Since the ultimate objective of the reliability performance index is its incorporation into an expert advisor driven computer-aided design system, simulation presents itself as the obvious choice for this determination of the kinematic reliability.

In either approach, the kinematic reliability is calculated at a single point at a time for the robot's trajectory or workspace. During the development of a modular robot system, the designer will generally have some task or trajectory in mind when considering the configuration of the system. In fact, as shown in Ambrose and Tesar [4] and Paredis and Khosla [119], the workspace requirements (reach and dexterity) are significant constraints when selecting the kinematic configuration of the modular robot system. The kinematic reliability can be calculated at the critical points of the trajectory (points where precise operations are required). Also, the kinematic reliability for each feasible configuration at the critical points can provide a relative indication of the accuracy and repeatability of each arm configuration. We wish to embed this information into the reliability performance index, so a single value for the trajectory or workspace needs to be selected.

5.2.3. Path Descriptions and Workspace Concerns

As mentioned before, the repeatability of a manipulator can change drastically over the workspace. This is due to the use of the mean values of the kinematic variables as the independent inputs to the manipulator's kinematic reliability. Since we are interested in selecting an "optimal" system with regard to reliability and performance, some measure of the kinematic reliability over the entire trajectory must be developed. We need a single representative value to express the kinematic reliability of the manipulator since the formulation of the reliability performance index assumes this can be done (see Section 5.4).

Several choices present themselves for consideration. The first choice is the use of the average value of the kinematic reliability over the trajectory or workspace. The average is easily generated from the calculations used to determine the kinematic reliability at each critical point in the trajectory or workspace. The average will be between either extreme value of reliability, so there will remain a significant possibility of the actual kinematic reliability at a certain point being lower than the average over the entire workspace or trajectory. The formulation of the average kinematic reliability over the workspace or trajectory is

$$R_{K_{Ave}} = \frac{1}{n}\sum_{i=1}^{n} R_{K_i} \qquad (5.17)$$

where R_{K_i} is the overall kinematic reliability at the *i*th point of the trajectory or workspace and n is the total number of points in the trajectory. An alternative possibility is the root-mean-square or RMS value of the kinematic reliability over the trajectory. The RMS value is commonly used as an average for oscillatory phenomena such as alternating signals. An advantage to using the RMS value is that it is more sensitive to extreme values. The RMS formulation is

$$R_{K_{RMS}} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} R_{K_i}^2} \qquad (5.18)$$

Another alternative to using an average value is the use of the minimum value over the trajectory. This would give the most conservative indication of the precision of the system. It also would involve no additional calculations, just comparisons. This alternative is chosen for implementation. The other formulations are recommended for further study on their effect of the RPI during use.

5.2.4. Sample Calculations

Bhatti and Rao provide two numerical examples of the computation of kinematic reliability. The first is a two Degree-Of-Freedom (DOF) planar manipulator. The second is the Stanford arm [18]. The 2-DOF planar manipulator is shown in Figure 5.1. The random variables are the link lengths and the joint angles which are assumed to be normally distributed. The values for the variables are given as $l_1 = 10$ cm, $l_2 = 8$ cm, $\sigma_{l_1} = \sigma_{l_2} = 0.01$ cm and $\sigma_{\theta_1} = \sigma_{\theta_2} = 0.01°$.

Using a rectangular permissive region, the kinematic reliability can be defined as

$$R_K = P(\bar{x} - \Delta x \leq x \leq \bar{x} + \Delta x, \bar{y} - \Delta y \leq y \leq \bar{y} + \Delta y) \tag{5.19}$$

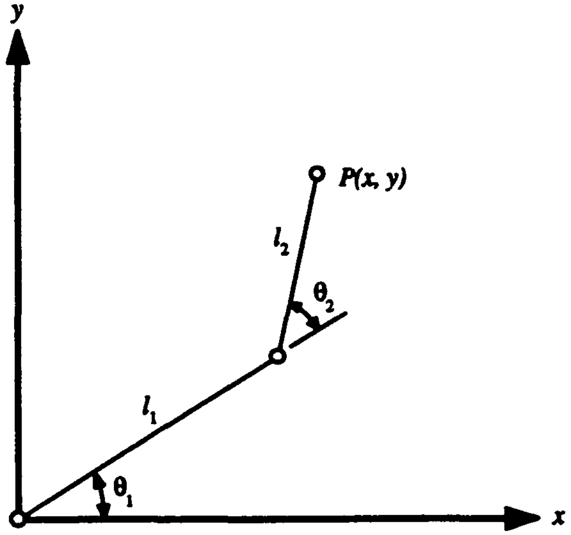


Figure 5.1. Two Degree of Freedom Planar Manipulator [18]

In this example, $\Delta x = \Delta y = 0.05$ cm. Using the analytical method, the hand position is given by the kinematic equations

$$\begin{aligned} x &= l_1 \cos\theta_1 + l_2 \cos(\theta_1 + \theta_2) \\ y &= l_1 \sin\theta_1 + l_2 \sin(\theta_1 + \theta_2) \end{aligned} \tag{5.20}$$

We can then calculate the means, variances, and covariance of Equation (5.20) as

$$\begin{aligned} \bar{x} &= \bar{l}_1 \cos\bar{\theta}_1 + \bar{l}_2 \cos(\bar{\theta}_1 + \bar{\theta}_2) \\ \bar{y} &= \bar{l}_1 \sin\bar{\theta}_1 + \bar{l}_2 \sin(\bar{\theta}_1 + \bar{\theta}_2) \end{aligned} \tag{5.21}$$

and

$$\begin{aligned} \sigma_x^2 &= \sum_{i=1}^{4} \left(\frac{\partial x}{\partial z_i} \right)_{\bar{x},\bar{y}}^2 \sigma_{z_i}^2 \\ \sigma_y^2 &= \sum_{i=1}^{4} \left(\frac{\partial y}{\partial z_i} \right)_{\bar{x},\bar{y}}^2 \sigma_{z_i}^2 \\ \sigma_{xy} &= \sum_{i=1}^{4} \left(\frac{\partial x}{\partial z_i} \right)_{\bar{x},\bar{y}} \left(\frac{\partial y}{\partial z_i} \right)_{\bar{x},\bar{y}} \sigma_{z_i}^2 \end{aligned} \tag{5.22}$$

where $z_1 = l_1$, $z_2 = l_2$, $z_3 = \theta_1$, $z_4 = \theta_2$, and the subscript $\bar{x}, \bar{y}$ indicates the partial derivatives are evaluated at the means of the variable. If we assume that the end-effector position follows a joint normal distribution (see discussion of this assumption in Section 5.2.2), we can write the distribution as

$$f(x_n, y_n) = \frac{1}{2\pi\sqrt{1-\rho^2}} e^{\left[-\frac{1}{2(1-\rho^2)}(x_n^2 - 2\rho x_n y_n + y_n^2)\right]} \tag{5.23}$$

where we have normalized the end-effector position by

$$x_n = \frac{x - \bar{x}}{\sigma_x}, \quad y_n = \frac{y - \bar{y}}{\sigma_y}, \quad \rho = \frac{\sigma_{xy}}{\sigma_x \sigma_y} \tag{5.24}$$

We can now find the manipulator's kinematic reliability from Equation A.2 as

$$R_K = \int_{-\Delta y/\sigma_y}^{\Delta y/\sigma_y} \int_{-\Delta x/\sigma_x}^{\Delta x/\sigma_x} f(x_n, y_n) dx_n dy_n \tag{5.25}$$

Bhatti and Rao numerically calculated this integral for three manipulator positions. The results of these calculations are shown in Table 5.3. They also completed a simulation of the planar manipulator. These results do not show agreement with the analytical results which they attribute to numerical errors in the integration of Equation (5.25). In his dissertation, Bhatti [17] shows a similar system and using better integration algorithms, gets agreement to within two significant digits between the analytic and simulation methods.

Table 5.3. Reliability of 2-DOF Planar Manipulator l_1 = 10 cm, l_2 = 8 cm, $\sigma_{l_1} = \sigma_{l_2} = 0.01$ cm, $\sigma_{\theta_1} = \sigma_{\theta_2} = 0.01°$, $\Delta x = \Delta y = 0.05$ cm [18].

		Kinematic Reliability	
Mean Joint Angles	Mean Position (cm)	Analytical	Simulation
(0°, 0°)	(18.0, 0.0)	0.8536	0.7251
(90°, 0°)	(10.0, 8.0)	0.9658	0.9037
(30°, 60°)	(8.66, 13.0)	0.9214	0.8108

5.3. Hardware and Software Reliability

The second component in the proposed reliability performance index is the quantification of the hardware and software reliability of the components of the modular robotic system. As discussed in Chapter 3 and Appendix A, there are several different choices of hardware reliability models from which to base this component on.

5.3.1. Reliability Model Requirements

The purpose of the reliability performance index is to allow the designer to quantify the durability and precision during design and to influence the design choices. It eventually will be incorporated into an expert system based computer-aided design system at the module level. The first requirement is that the reliability model itself be modular to allow for automatic assembly of the model as the system configuration is being chosen. This will also allow the model to draw upon module reliability information from a data base. This requirement also implies that the modules have independent lives unless the techniques suggested in Section 3.5 can be developed to provide an analytical methodology for dependent module reliability analysis.

Another requirement of the reliability model is low computational complexity of the model. Since the ultimate use of the computer-aided design system will most likely use it interactively, the calculations required to generate the configuration reliability will need to be fast, so there will not be an extended wait at the CAD terminal. The fourth aspect of the model is the ability to include fault-tolerant schemes directly into the reliability model. This will be important if the CAD system includes the ability to configure fault-tolerant modular systems such as those described in Section 4.4.

An important aspect of reliability models is the dynamic (time dependent) nature of hardware reliability. In this context, we are generating reliability estimates of modules at equivalent times to be used as a comparative tool in the selection of a design configuration. We are not interested in the time behavior at this point and as such have not included time dependency as a model requirement. An additional assumption we must make is independence between the module life distributions. This assumption is made since the current knowledge of the algebra of random variables does not allow us to construct models of the dependency between component lives in a modular structure (see Section 3.5 for a discussion of this problem).

These four requirements: modularity of the model, modularity of the data, complexity of the model, and the ability to include fault-tolerance in the model, are examined for the three major types of hardware reliability models. Each model type is given a number from one to five based upon its ability to meet the requirement with five being the best. The numbers for each requirement are added up and the model with the highest sum is chosen for the hardware reliability model.

5.3.2. Hardware Reliability Model Selection

Three different reliability models are considered for the hardware component of the reliability performance index: Markov models, reliability block diagrams, and Fault-tree models. Each model is examined for the requirements stated above and ranked accordingly.

5.3.2.1. Markov Models

The Markov model is described in Appendix A, Section A.2.1.2. The Markov model is based upon operational states rather than specifically on the components of the system itself. This can make the construction of the model arduous since the transition rates are composed of the failure rates of the components in different combinations. Consider a two component system, with each component being either failed or operational. Let λ_1 and λ_2 be the respective component's failure rates. This Markov model requires four states: state 11 is both components operational, state 01 is 1 failed and 2 operating, state 10 is 1 operating and 2 failed, and state 00 is both failed. One possible state transition diagram is shown in Figure 5.2 with the transition rates on the arc annotated. This model also assumes that both components cannot fail simultaneously. The diagram of Figure 5.2 is fairly simple, however, as more states are added, this becomes increasingly complex. This characteristic makes automatic generation of a Markov model relatively difficult and is given a 1.

The second requirement is modularity of data. There is a direct correspondence between the transition rates and the component failure rates, but the data must also contain the interactions between the modules and how the overall model is constructed. This aspect is rated a 3.

The third requirement is that the model be solved rapidly and with low complexity. As seen from the formulation of the Markov model solution in Appendix A, it requires the simultaneous solution of as many differential equations as states in the model plus the determination of operational states. For large models, this can be computationally burdensome, earning the Markov model a 2 for computational complexity.

The last requirement is the ability of the model to include fault-tolerant characteristics. In this aspect, the Markov model is unsurpassed since it is extremely easy to add fault-tolerant schemes to the Markov model by the addition of more states representing the fault-tolerant conditions of the system. Degradation of components can easily be modeled by adding additional levels of degradation to the description of each component. Repair is also easily added to the model by adding arcs representing repair rates returning the system from failed to operational states. Markov models rate a 5 in this category.

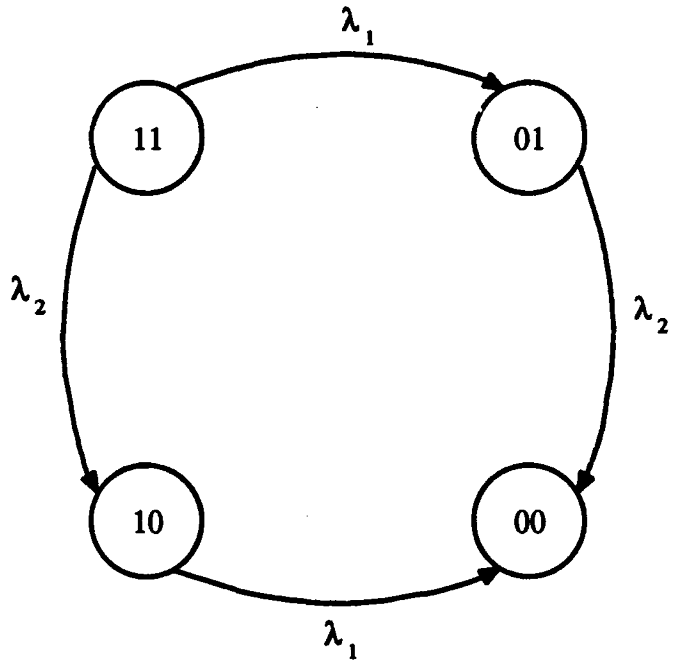


Figure 5.2. Four State Continuous Markov Transition Diagram

5.3.2.2. Reliability Block Diagrams

Reliability Block Diagrams (RBDs) are examined in Appendix A, Section A.2.1.1.1. The first requirement of modularity is easily met by RBD models since they are "black box" models and are modular by their very nature. If the component (modules) are independent, the system reliability function is easily determined from the RBD. In the serial case (which is true for most industrial robot systems), the system reliability is just the product of the reliabilities of the modules (see Equation (A.33)). This earns the RBD a 4 in this category.

Independent data for each module can be determined and stored for use upon demand. Computationally, each module will have its own reliability characteristics (either static value or dynamic functional parameters) which can be assembled into the model as needed. Only functional evaluations are required, not the solution of systems, thus RBDs score a 5 for data and a 4 for lack of complexity.

Difficulties arise when trying to add fault-tolerant schemes to the RBD model. Event combinations are used to evaluate this model under these circumstances (see Appendix A, Section A.2.1.1.1). This difficulty rates RBDs a 1 in fault-tolerance.

5.3.2.3. Fault Tree Model

Fault-Tree Analysis (FTA) is explained in Appendix A, Section A.2.1.1.3 as a top-down, systematic analysis of the possible causes of top-level failure events. Probabilistically providing the same answer as static reliability block diagrams, FTA examines the system's physical structure and decomposes the system behavior into Boolean trees showing the dependence of the top-level event upon the lower level events. Fault trees can be generated automatically based upon the system structure, but operator interation is required to select the top level events and guide the decomposition. This rates Fault Trees as a 2 in modularity since there is no direct correspondence between the module data and the components of the tree as the tree is expanded.

The equivalence with reliability block diagrams in static but not in dynamic data earns the fault-tree a 3 for modular data. Fault trees cannot handle time dependencies and if the reliability is expressed in functional form (such as a probability distribution), FTA cannot be used to evaluate the system reliability.

Computationally, the solution of a fault tree can proceed by computing the Boolean functions and evaluating them, or it can use cut sets to calculate the probability of success of a particular branch. These methods are computationally expensive, resulting in a ranking of 3 for computational complexity. Fault-tolerance is not easily included into the fault-tree since it introduces dependency structures, making the solution more difficult. FTA ranks a 1 for fault-tolerance.

5.3.2.4. Model Selection and Discussion

The rankings for the different models are compiled and presented in Table 5.4. The reliability block diagram method has the highest ranking and is selected for use in the reliability performance index formulation. This selection allows for the storage of reliability data for the modules in several different ways, depending upon what type of data is available. Failure rate data, such as that in MIL-HDBK-217, can be stored and accessed directly. Life distributions will require storage of the type of distribution and the parameters with the system calculating a reliability value from the functional form.

Table 5.4. Hardware Reliability Model Selection Chart for the RPI

Model	Modularity	Data	Complexity	Fault-tolerance	Total
Markov	1	3	2	5	11
RBD	4	5	4	1	14
FTA	2	3	3	1	9

Several assumptions are made in the computation of the hardware reliability model component of the RPI. The first is independence of the lives of the modules. The problem with this assumption is addressed in Section 3.5. This is an area of basic research required to understand the impact that modularity has on the reliability of the system. The second assumption is in determining an actual value for the hardware reliability. The reliability function of a module will be time dependent, resulting in a different reliability for different times. Thus, a time must be specified to generate the reliability (see Appendix A, Section A.1.1). We are not developing this model primarily for predictive purposes. Instead, we are using it to perform relative comparisons of the reliabilities of different configurations. Reliability is not considered a constraint in this context, rather, we are trying to maximize the reliability while satisfying our performance specifications. In this case, we select a time for evaluation of the reliability function to generate the hardware reliability component of the RPI.

The question now arises, what time should be selected. We must choose a reasonable time for evaluation so that we can compare reliability values on an equivalent basis. Consider the Weibull distribution which has the reliability function of

$$R(t) = e^{-\left(\frac{t}{\theta}\right)^{\beta}} \tag{5.26}$$

where θ is the characteristic life and β is the shape factor. If β = 1, then the Weibull degenerates to the exponential distribution. Table 5.5 shows the reliability calculated from Equation (5.26) with θ = 10,000 hours for different times t and shape factor β.

Table 5.5. Weibull Reliability for θ = 10,000 hours.

Time (hrs)	β = 0.5	β = 1.0	β = 1.5	β = 3.0
100	0.9048374	0.9900498	0.9990005	0.999999
1,000	0.7288934	0.9048374	0.96887199	0.9990005
5,000	0.4930687	0.6065307	0.7021885	0.8824969
10,000	0.3678794	0.3678794	0.3678794	0.3678794
50,000	0.1068779	0.0067379	0.0000139	0.0

Comparatively, we can argue that any choice of time under the characteristic life of a distribution will provide a value that can be used comparatively. We choose to select the time of reliability evaluation to be less than one-half the minimum characteristic life of all the modules, or

$$t \leq \tfrac{1}{2}\theta_{Min} \tag{5.27}$$

The impact of this choice on the values of the RPI is not investigated and is recommended for further study.

5.3.3. Software Reliability Model Selection

The reliability of the system software also plays an important role in the achieved reliability of the system, and should be taken into consideration during the selection of the system configuration. In concept, the modular robotic system will be modular at the system software level as well. There will be different software modules to choose from to enable certain functions desired by the system user. These modules may be tied to a particular hardware or mechanical architectural configuration or may implement different control algorithms to meet different system specifications. This section presents several reliability models for software and selects a model based on similar requirements as was done in selecting the hardware reliability model. While software reliability is not included in the example in Section 5.4 or the Chapter 6 case study, it can easily be incorporated by adding the software failure rate generated from the modular software reliability model to the hardware failure rate and using the combined hardware/software reliability model in Equation (5.31).

In selecting the software reliability model, we assume that software is available to control and operate the system as a suite of software modules that are easily integrated and linked through standard data and control interfaces. These might be different control algorithm implementations, different supervisory modules, input/output modules depending upon the sensor suite specified, or additional fault-tolerant software modules for all of these functions. The reliability model requirements considered are modularity and ease of automatic assembly and generation, modular data availability, model computational complexity, and ability to include fault-tolerance into the model. Again, these requirements are rated from one to five with five being the best for each model considered.

5.3.3.1. The Littlewood Modular Software Reliability Model

The first model considered is the Littlewood Modular Software Reliability model [85]. This model is based upon a semi-Markov model of the transfer of control between software modules (see Appendix A for a discussion of the Markov Model). The model requires knowledge of the individual module's failure rate, denoted as ν_i, and assumes that the failure rates of the modules are much less than the rate of transfer between the modules. The model also requires knowledge of the probability of failure occurring during control transfers. With the above assumption, Littlewood develops the software system failure rate as

$$\lambda_{S/W} = \sum_i a_i \nu_i + \sum_{ij} b_{ij} \lambda_{ij} \qquad (5.28)$$

where a_i = the proportion of time spent in module i,
b_{ij} = the frequency of transfer of control from module i to module j,
ν_i = software module i failure rate, and
λ_{ij} = the probability of failure during control transfer from i to j.

The formulation of Equation (5.28) is very simple, making use of data that would be available for each module determined during module development and testing. The model is inherently modular, making for easy assembly of the model and simple calculation of the software failure rate. This failure rate is constant and can be used in conjunction with the hardware system reliability model with no alterations. An even simpler model can be achieved if we can guarantee no failure during control transfer between modules (perhaps by automating the software design process through computer-aided software engineering). With this assumption Equation (5.28) becomes

$$\lambda_{S/W} = \sum_i a_i \nu_i \tag{5.29}$$

For these reasons, the Littlewood model rates a 5 for modularity and ease of automatic assembly.

The data for this model is modular as well and can be made available through the testing of each module. The frequency of the *i* to *j* transfer of control can be determined from the supervisory module's calls to the rest of the modules in the program. This model requires four parameters: two for each module (the module failure rate and proportion of time spent in the module) and two overall parameters (the frequency of control transfer and the probability of control transfer failure). The data aspect of the Littlewood model is rated a 3.

Simplicity is the strong point of the Littlewood model, only requiring the evaluation of Equations (5.28) or (5.29) with little manipulation of data. The Littlewood model is rated a 5 in the complexity category. Also, since the basis for the Littlewood model is the Markov model, it will incorporate fault-tolerant software modules with little difficulty. This rates the Littlewood model a 4 in the fault-tolerance category.

5.3.3.2. The Kubat Modular Software Reliability Model

Kubat [81] extends Littlewood's model by including the description of the general time one module spends in control. This model is stated in Equation (3.22). If we assume only one overall program exists (one task to be accomplished in Kubat's terminology) Eq. (3.22) simplifies to

$$\lambda_{S/W} = 1 - \prod_{i=1}^{M} \left(g_i^{*}(\alpha_i) \right)^{a_i} \tag{5.30}$$

where M = the number of software modules,

g_i^{*} = the Laplace transform of the probability distribution function of the time the program spends in module *i*,

α_i = the failure rate for module *i*,

a_i = the average number of visits the program makes to module *i*.

This model requires more in-depth determination of the life characteristics of the modules themselves.

As can be seen from Equation (5.30), the Kubat modular model is more complex than the Littlewood model, requiring products of a functional evaluation, raised to a power. The data required for this model includes the functional description of the Laplace transform of the probability distribution function for the time spent in a module. This requires slightly more extensive testing of the software modules to determine the best fit of a distribution function to the data. The transform can be determined and added to the module data and evaluated for the proper failure rate, since the time spent in any one module is a function of the module itself [81]. The Kubat model rates a 4 for modularity and ease of assembly and a 2 for data.

The Kubat model is more complex than the Littlewood model requiring more operations. If the assumptions required to simplify the Littlewood model to Equation (5.28) are not met, the Kubat model can be used instead with very little additional effort. The Kubat model rates a 4 for complexity and with the same argument as for the Littlewood model, rates a 4 for fault-tolerance.

5.3.3.3. Model Selection and Discussion

The rankings for the Littlewood and Kubat reliability models are compiled in Table 5.6 with the Littlewood model achieving the highest ranking. The Littlewood model is selected for use based upon this ranking. The main problem with the use of software reliability models is in the lack of data. Each different piece of software is unique making it inappropriate to use data from other software systems to predict the reliability of new code. This data is usually generated during development and testing of the software. A method for testing modular software is presented in [88] where software errors are measured during development and testing to determine the most cost effective time to release the software for distribution.

Table 5.6. Software Reliability Model Selection Matrix for the RPI

Model	Modularity	Data	Complexity	Fault-tolerance	Total
Littlewood	5	3	5	4	17
Kubat	4	2	4	4	14

As an example, assume we have a modular software system with four modules as depicted in Table 5.7. For each invocation of the program, assume the computation module is called 50 times, the input/output (I/O) module twice, and the memory manager module twice. Also, each time the computation module is called, it calls the I/O and the memory manager once during its execution and control is returned to the module from which the control originated. This results in 308 transitions between modules during one invocation of the program and allows the construction of Table 5.8.

Table 5.7. Example Software System Module Characteristics (Assumed)

Module	Function	a_i	v_i (failures/10^6 hrs)
1	Supervisor	0.0833	0.5
2	Input/Output	0.333	2.0
3	Memory Manager	0.0833	0.5
4	Computation	0.5	1.0

Table 5.8. Example Software System Control Transfer Frequencies (Assumed), $\lambda_{ij} = 0.01$

i - j	b_{ij}	i - j	b_{ij}	i - j	b_{ij}	i - j	b_{ij}
1-1	0.0	2-1	0.0065	3-1	0.0065	4-1	0.1623
1-2	0.0065	2-2	0.0	3-2	0.0	4-2	0.1623
1-3	0.0065	2-3	0.0	3-3	0.0	4-3	0.1623
1-4	0.1623	2-4	0.1623	3-4	0.1623	4-4	0.0

Inserting the data of Tables 5.7 and 5.8 into Equation (5.28) provides a failure rate estimate for the modular software system as

$$\begin{aligned}\lambda_{S/W} &= \sum_i a_i \nu_i + \sum_{ij} b_{ij} \lambda_{ij} \\ &= (0.0833)(0.5) + (0.333)(2) + (0.0833)(0.5) + (0.5)(1) \\ &\quad + 4[(0.0065)(0.01)] + 6[(0.1623)(0.01)] \\ \lambda_{S/W} &= 1.26 \text{ failures per million hours}\end{aligned} \tag{5.31}$$

5.4. Reliability Performance Index

Consider the effect of failure on the robotic system. Unless the system has fault-tolerant capabilities, any failure in the system will degrade its performance. Using the definition of manipulator failure found in Section 5.2, we propose that any failure of the robot system will cause the pose of the manipulator to be outside its permissible region. This suggests the top level reliability structure of the non fault-tolerant system is serial. In other words, we can form an index of the system's top level performance as

$$RPI = R_H \cdot R_{K_{Min}} \tag{5.32}$$

where R_H is the combined system hardware and software reliability and R_{Kmin} is the minimum system kinematic reliability over the workspace or the planned trajectory. This value is not strictly a probability since the kinematic reliability is dependent upon the hardware and software reliability. If the kinematic reliability is independent of the hardware reliability it will represent the total probability of success, i.e. the probability of no failures in a certain time and meeting the tolerance specification on the end-effector. The RPI does indicate (as a guideline) the ability of the system to function within its specification.

5.4.1. Formulation

The formulation of the index and the calculations involved can best be illustrated by an example. Consider again the 2-DOF planar manipulator of Figure 5.1. The components of this system can be assumed to be two links, two actuators, one at the base and one at the elbow, and the associated control system hardware and software. Assume the system has the component characteristics of Table 5.9.

Table 5.9. Component Characteristics for 2-DOF Manipulator System (Failure Rates from Generic Component Data in Chapter 4)

Component (Quantity)	Failure Rate (per 10^6 hrs)	Standard Deviation
Links (2)	0.0	0.01 cm
Actuators (2)	3.14	0.1°
Encoders (4)	0.32	*
Cable Assemblies (2)	5.5	*
Controller (1)	46.4	*

We first examine the kinematic reliability of the system. Using the square permissive region described by Equation (5.19) with the tolerances specified in Section 5.2.4, the kinematic reliability is calculated over the workspace using the simulation method. Figures 5.3 and 5.4 are graphical representations of the variation in the kinematic reliability over the workspace and the joint space, respectively. As one would expect, the minimum values (where the mechanism is least accurate) are at the singular values of the manipulator and at the extreme reach of the manipulator. Representing the kinematic variation over the workspace as $R_k(x, y)$ we can choose the minimum value by writing

$$R_{K_{Min}} = \min R_K(x, y) \tag{5.33}$$

Alternately, the minimum value can be calculated in the joint space. In this case, we can write

$$R_{K_{Min}} = \min R_K(\theta_1, \theta_2) \tag{5.34}$$

In either formulation, the value of R_{Kmin} will be the same. The difference is in the way the data is presented. It would be easier to represent the minimum value in the three dimensional workspace as done in Figure 5.3, rather than in a six or seven dimensional joint space. Figure 5.4 is understandable since there are only two revolute joints. In this example, the minimum value is found to be $R_{Kmin} = 0.84$.

The next component of the index is the hardware reliability component. The data presented in Table 5.6 provides constant failure rate data compatible with an exponential system failure model. We assume the system reliability structure is serial and we can form the system failure rate as

$$\begin{aligned} \lambda_h &= 2\lambda_{links} + 2\lambda_{act} + 4\lambda_{enc} + 2\lambda_{cable} + \lambda_{control} \\ &= 64.96 \text{ failures per million hours} \end{aligned} \tag{5.35}$$

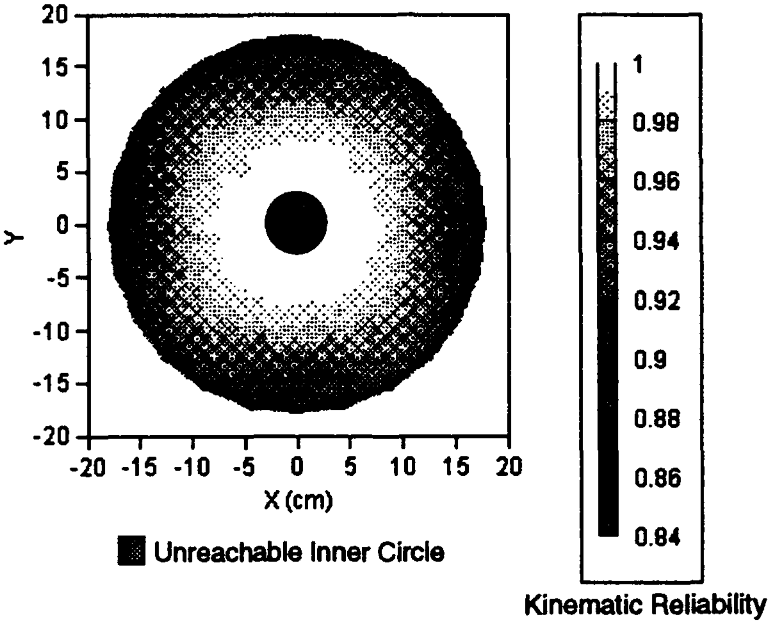


Figure 5.3. 2-DOF Manipulator Kinematic Reliability over the Workspace

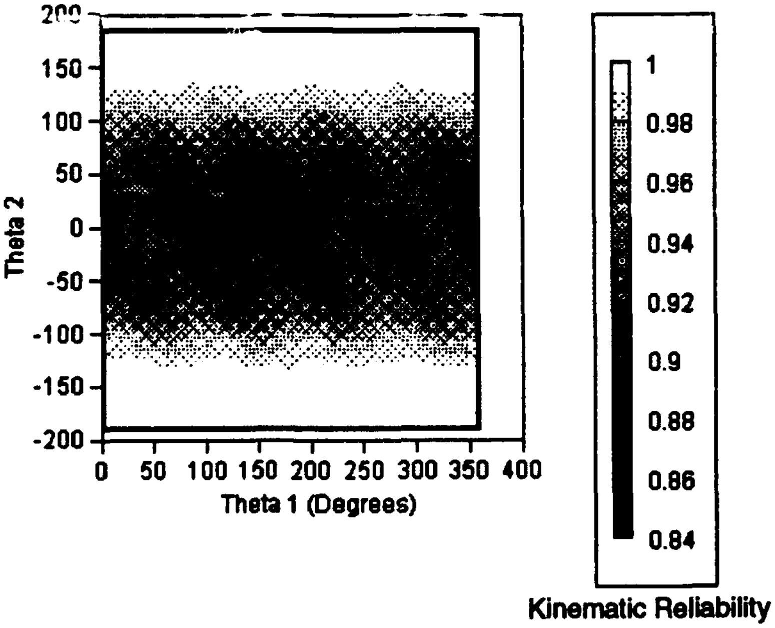


Figure 5.4. 2-DOF Manipulator Kinematic Reliability over the Joint Space

This gives a system MTBF of 15,400 hours, so we select the time of evaluation to be 1,000 hours. The hardware reliability function is

$$R_h = \mathrm{e}^{-\lambda_h t} \tag{5.36}$$

resulting in a value of $R_h = 0.9371$. Inserting this value and the value for R_{Kmin} obtained from Equation (5.33) into Equation (5.32) results in

$$RPI = 0.7872 \tag{5.37}$$

5.4.2. Time Dependence, Architectural Permutations, and Applicability

In Section 5.3.2.4 we discussed the selection of a time at which to evaluate the hardware and software reliability models. This is done because we are assuming that the RPI is a static property of the system. This is only a first approximation of the behavior of the RPI. The reliability of the system is actually time dependent, as explained in Appendix A. This is also true of the kinematic reliability. In this work, we have assumed that the kinematic reliability is a constant value. In reality, the kinematic reliability will change due to wear and aging in the manipulator. The formulation of Equation (5.32) constrained by the time of evaluation of the hardware and software model as stated by Equation (5.27) implies that the RPI is actually a measure of the system performance at the time of evaluation. The kinematic reliability is assumed constant over the life of the system.

If we can model the kinematic reliability as a dynamic (changing with time) characteristic, we can alter Equation (5.32) to be

$$RPI(t) = R_h(t) \cdot R_K(t) \tag{5.38}$$

where $R_h(t)$ is the hardware/software reliability function and $R_K(t)$ is the dynamic kinematic reliability function. The determination of $R_K(t)$ as an analytic function promises to be a difficult exercise in the application of the algebra of random variables since it will require the determination of the time dependent distribution of the end-effector. Again, the tools discussed in Section 3.5 are necessary to develop this distribution if dependencies exist between the various components of the system. This extension of the RPI also requires that the definition of robotic system reliability be updated to be the probability that the end-effector can repeatedly attain a certain pose for a given period of time. The hardware and kinematic reliabilities can then be evaluated at the same time to quantify the overall performance of the system. This extension, which will also allow the RPI to measure system degradation over time, is recommended for future research.

The Reliability Performance Index was developed assuming that the top level reliability structure of the system was serial, i.e. that hardware failures will cause the system to fail in the kinematic sense. This may not be true in all cases. Consider the dual actuator module described in Chapter 4. This actuator module has a higher reliability than an actuator module with a single motor. If one motor fails, the other half can take up the load, and the actuator module can be said to be one-fault-tolerant. The failure of one half of the actuator module may or may not effect the kinematic reliability of the manipulator; it depends upon whether or not the repeatability of the joint is affected. If the motors within this dual architecture are strong enough, the standard deviation of the joint module will not change and the assumption that hardware failure causes kinematic failure is invalidated. Hopefully, the fault-tolerant scheme will improve the hardware reliability (although fault-tolerance does not imply high reliability). This will increase the value of the RPI by making the hardware reliability component increase. This will allow the effect of fault-tolerance to be included in the RPI regardless of the reliaυility structure of the system. Thus, it may be sufficient just to form the RPI without regard to the top-level reliability structure. This relationship between the toμ-level reliability structure and the RPI should be investigated further, perhaps quantifying the effect of fault-tolerant systems on the RPI and determining if other choices for the top-level reliability structure are appropriate, along with the necessary modifications to Equation (5.32).

An additional question which needs to be addressed it the applicability of the RPI and the types of systems for which it may be used. As conceived, the RPI framework is general in scope if the fundamental assumption of independence between the hardware and kinematic components is true. It can be applied to any robotic system given the kinematic variability and the hardware and software models for the system. It should not be applied when the two components are dependent since this dependency usually will reduce the value of the RPI via Baye's Rule [95]. This dependency normally exists for any manipulator system but it is most apparent in fault-tolerant systems when a hardware failure may not cause the system to fail completely but cause a degradation in the system's accuracy. The RPI should not be used in this situation without modification of the structure of Equation (5.32) to account for the dependency.

5.5. Summary

This chapter presented the development of a Reliability Performance Index for a manipulator system that can be used to incorporate reliability into the robotic system design and as a guideline in the selection of modules for a modular robotic system configuration. This RPI is based on both the durability of the system and a measure of the accuracy and repeatability denoted as the kinematic reliability, given reliability data of the modules as well as their size position tolerances.

The RPI has two major components: the system combined hardware and software reliability and the system kinematic reliability. The basis for the combination of these two statistical measures of system performance was the assumption that a failure of the robot system hardware will cause the accuracy to degrade (Section 5.4). This allowed the overall definition of robot reliability as the probability of the system to achieve a certain pose within a specified error bound on both position and orientation. The top level RPI was then defined to be the product of the s?stem hardware reliability and the kinematic reliability as stated in Equation (5.32).

The hardware reliability model chosen in Section 5.3.2 was the reliability block diagram model, using independent module lives. This selection was based upon the ease of computation and data storage and the ability to automatically assemble the reliability model from a configuration generated by a computer-aided design package. A similar selection was made for the software model in Section 5.3.3. The overall kinematic reliability was iound by choosing the minimum value of the kinematic reliability as it varied over the workspace (or the task) of the manipulator system with possible alternatives to this approach presented in Section 5.2.3.

The formulation of the RPI was illustrated with a 2 degree-of-freedom example in Section 5.4.1 based on an example presented by Bhatti and Rao [18] in their paper on kinematic reliability. The example system assumed a certain failure rate for each system component and calculated the RPI based upon simulation results over the entire workspace.

The Reliability Performance Index also has applicability to any system that allows the measurement of error, such as a control system. The RPI can be applied as an alternate formulation of the system reliability including its performance characteristics. Additional extensions, such as time dependency and the inclusion of fault-tolerance are suggested

The next chapter presents a case study in the application of the RPI, showing how it can be applied to the selection of the manipulator module configuration for a modular robotic system. The case study consists of six different suites of links and six different actuator modules of varying tolerances and failure rates. The RPI of the different systems is investigated over a specified trajectory, and a "optimal" manipulator is identified.

CHAPTER 6: AN EXAMPLE IN THE APPLICATION OF THE RELIABILITY PERFORMANCE INDEX

6.1. The Study of the Reliability Performance Index

This chapter examines some of the properties and characteristics of the Reliability Performance Index (RPI) developed in Chapter 5. Some of the questions that need to be addressed are:

1. Does the maximum RPI value correspond to a maximum for reliability and accuracy in the design space?
2. How sensitive is the RPI to the changes in the design options?
3. Is there a significant difference in the RPI for the selection of different components of the system configuration?

These questions will be addressed through the application of the RPI to a case study in the selection of modular manipulator components for configuration and trajectory.

6.2. The Case Study System

To give an example that can be followed and expanded for further investigation, a three Degree-Of-Freedom (DOF) system is proposed for study. This allows for the inclusion of the orientational component of the kinematic reliability to be included in the study. This 3-DOF manipulator is shown in Figure 6.1. For the case study, we assume that we have a task which will require the manipulator to follow a rectangular path described in Table 6.1 and illustrated in Figure 6.2. This path was chosen since it causes the manipulator to go from a folded position, through its fully extended position, back to a folded position with the end-effector following a straight line trajectory. The end-effector orientation is commanded through a 270° motion. The position tolerance specification of the end-effector was specified to be 0.001 meter and the tolerance specification on the end-effector orientation was set at 0.1°.

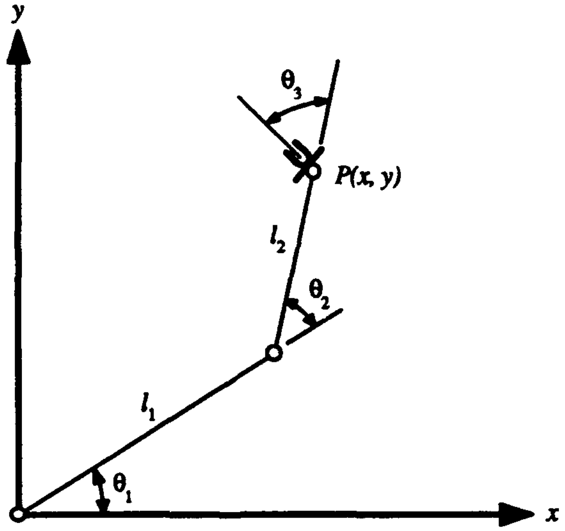


Figure 6.1. Three Degree of Freedom Manipulator

The suite of modules assumed to be available include six different joint modules with varying reliability and precision and six sets of links of differing mean lengths and machining tolerances. These component characteristics are presented in Tables 6.2 and 6.3. For a component of about 1 meter in size, we can reasonably expect to achieve around ±0.0002 meter machining tolerance. It is standard practice when tolerancing to assume that the tolerance is equivalent to three standard deviations each side of the mean length [73]. This makes the standard deviation for a 1 meter length with this tolerance to be $\sigma_l = 0.00008$ meter which we round up to $\sigma_l = 0.0001$ meter. In Chapter 4, we noted that the links and structural members of manipulators had not experienced any documented failures. This fact is reflected in the link failure rates of Table 6.2.

Table 6.1. End-Effector Rectangular Motion Path for the 3-DOF Manipulator

Point on Path	*x* Position (m)	*y* Position (m)	Orientation (Degrees)
1	0.0	1.0	180
2	0.35	1.0	135
3	0.7	1.0	90
4	1.06	1.06	45
5	1.0	0.7	0.0
6	1.0	0.35	-45
7	1.0	0.0	-90

The tolerances on the joint modules where chosen to represent the ranges one might encounter in the accuracy of an actuator. We also assume that the reliability of a joint module is inversely proportional to the precision since more precise components will generally have tighter fits and more wear can occur as well as the possibility of additional failure modes. The minimum and maximum failure rates were based upon the reliability analysis of the University of Texas Actuator

Module performed in Chapter 4. The failure rate calculated was 651 failures per million hours, using the worst possible options for environment and material selection in the reliability model. Choosing the best possible options in the reliability model resulted in a failure rate of 41 failures per million hours. This range is reflected in the reliabilities of the joint modules of Table 6.3. All distributions, both link lengths and joint positions are assumed to be normal.

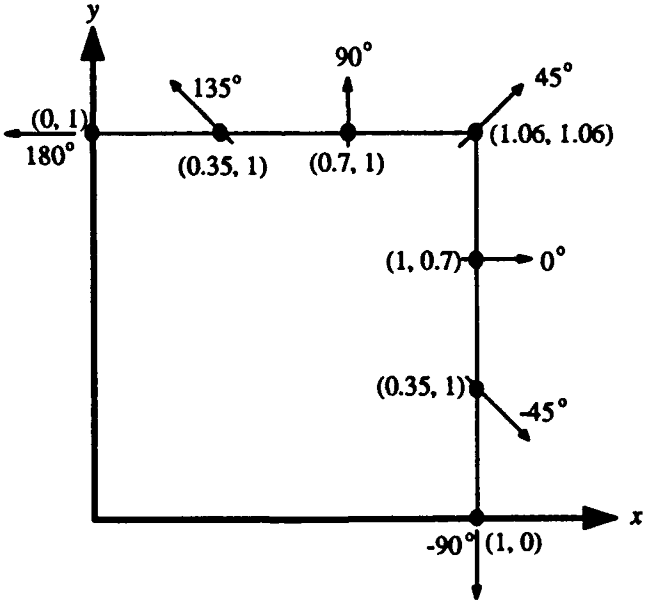


Figure 6.2. Rectangular Motion Path for the 3-DOF Manipulator (Arrow Indicates End-Effector Orientation)

Table 6.2. Link Module Characteristics

Option	l_1 (meters)	l_2 (meters)	Failure Rate (Failures/10^6 hrs)	Standard Deviation (meters)
1	0.75	0.75	0.0	0.0001
2	1.0	0.5	0.0	0.0001
3	0.5	1.0	0.0	0.0001
4	0.75	0.75	0.0	0.001
5	1.0	0.5	0.0	0.001
6	0.5	1.0	0.0	0.001

Each different link length option represents a separate inverse kinematic problem. The inverse kinematics for each link length option was derived after Craig [36], and the required joint angles to achieve each position annotated in Figure 6.2 where calculated. The kinematic equations for the system of Figure 6.1 are

$$\begin{aligned} x &= l_1 \cos\theta_1 + l_2 \cos(\theta_1 + \theta_2) \\ y &= l_1 \sin\theta_1 + l_2 \sin(\theta_1 + \theta_2) \\ \Phi &= \theta_1 + \theta_2 + \theta_3 \end{aligned} \tag{6.1}$$

where x and y represent the end effector position in the workspace and Φ represents the end-effector orientation. Using a rectangular position permissive region and an angular tolerance on the end-effector orientation, we can represent the kinematic reliability of the system as

$$R_K(x,y,\Phi) = P\left\{\begin{matrix}(x_d - 0.001 \le x \le x_d + 0.001) \cup \\ (y_d - 0.001 \le y \le y_d + 0.001) \cup \\ (\Phi_d - 0.1 \le \Phi \le \Phi_d + 0.1)\end{matrix}\right\} \tag{6.2}$$

and

$$R_K = \min R_K(x,y,\Phi) \tag{6.3}$$

Table 6.3. Actuator Module Characteristics

Joint Module Number	Failure Rate (Failures/10^6 hours)	Standard Deviation (degrees)
1	40	0.1
2	140	0.05
3	240	0.01
4	340	0.005
5	440	0.001
6	540	0.0005

The hardware reliability of the system is assumed exponential and can be expressed as

$$R_h(t) = e^{-\lambda_S t} \tag{6.4}$$

where

$$\lambda_S = \lambda_{Base} + \lambda_{Elbow} + \lambda_{Wrist} \tag{6.5}$$

since the links do not contribute to the system failure rate. Equation (6.5) does not include the controller or software that would be necessary to control this system, although these components can be easily added if data is available. Examining Table 6.3, it is seen that the minimum MTBF (= inverse of the failure rate) for the joint modules is 1852 hours (module 6). The time chosen for hardware reliability evaluation is 100 hours, which satisfies the suggested criterion of Equation (5.27).

A Monte Carlo simulation is performed to evaluate the kinematic reliability of the system at each position of the trajectory of Figure 6.2 for each link option and joint module combination. Five hundred trials at each end-effector position are generated and the kinematic reliability at that position is calculated with Equation (5.16). The minimum kinematic reliability over the workspace is used to represent the system kinematic reliability per Equation (6.3). Each simulation was repeated five times and the results where averaged to obtain the system kinematic reliability for the particular link and actuator combination. The hardware reliability was evaluated per Equation (6.4) for each combination of joint modules and the RPI was calculated. The results of these simulations are examined in the next three sections to address the questions posed in Section 6.1.

6.3. Optimization Studies

The first question posed in Section 6.1 inquires if the link and joint module combination with the maximum RPI represents an "optimum point" in the design space. We first must describe what we mean by "optimum." In the context of the RPI, we are searching for the combination of actuators that will maximize both system reliability and accuracy (as represented by the kinematic reliability). The standard optimization problem is formulated by Equation (3.17) where a vector of design parameters is chosen such that an objective function is maximized. Usually, the objective function is a function of the design variables. However, in the case of the RPI, the design variables are not explicit and we must address what the maximum actually means.

The RPI basic formulation is found in Equation (5.32). Both components, the hardware reliability and the kinematic reliability, are measures of probability and are bounded to be non-negative and less than or equal to unity, thus the maximum value of the RPI is unity representing a certainty of no failures as well as always being inside the permissible pose region. The highest value of kinematic reliability means that the particular combination is the most accurate, having the highest probability of being inside the error bound on the pose. The highest value of hardware reliability means that the particular combination of components is the most reliable of all the combinations. One might decide to pick the most accurate combination that has a reliability of at least, say, 0.98 at 100 hours or the objective may be reversed: the highest reliability with at least a 0.9 kinematic reliability. First, we will examine the maximum values of the RPI throughout the design space and then we will look at the above two questions.

Table 6.4 presents the results of the simulations of the trajectory of Figure 6.2. A large difference can be seen between the maximum and minimum values for the different link and joint module combinations. The most immediate difference can be seen between the two sets of tolerances of the link modules. The higher tolerance links make a tremendous difference in the value of the achieved RPI. Based upon the maximum of the RPI from Table 6.4, one would select link option 2 with joint module #3 in every joint position.

Table 6.4. Maximum and Minimum Results from RPI Simulation of Figure 6.2.

Link Option	Base Module	Elbow Module	Wrist Module	Min/ Max	R_h	R_k	RPI
1	3	3	3	Max	0.93044	0.99987	0.93034
	1	1	1	Min	0.98799	0.1664	0.1644
2	3	3	3	Max	0.93044	1.0	0.93044
	1	1	1	Min	0.98797	0.17599	0.17388
3	3	3	3	Max	0.93044	0.9996	0.93011
	1	1	1	Min	0.98802	0.17013	0.1681
4	4	3	3	Max	0.9212	0.44907	0.41368
	1	1	1	Min	0.98802	0.082	0.08102
5	3	3	3	Max	0.93044	0.44733	0.41622
	1	1	1	Min	0.98797	0.09213	0.09103
6	3	4	3	Max	0.92123	0.45253	0.41689
	1	1	1	Min	0.98799	0.08947	0.08839

Additionally, the maximum values are extremely close together, within one standard deviation of the mean RPI. The statistical significance of this fact is discussed in Section 6.5. An additional presentation of these results can be seen in Figures 6.3 through 6.8 which show the trend of the RPI as the joints modules are changing location. The simulation iterated inward from the wrist. The six obvious humps in each of the traces correspond directly to the module in the base joint.

Joint module combination 1 corresponds to the module locations (1, 1, 1) and combination 216 corresponds to locations (6, 6, 6).

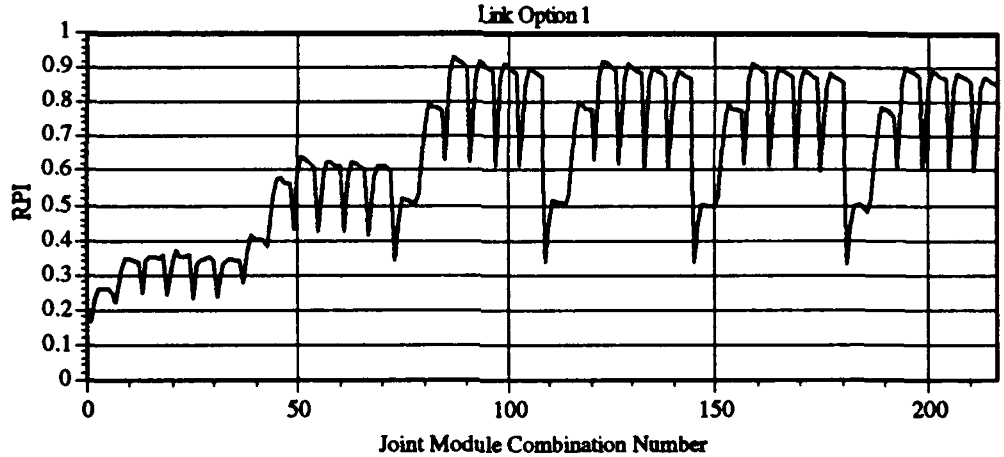


Figure 6.3. *RPI by Joint Module Combinations: Link Option 1*
$l_1 = l_2 = 0.75$ m, Tolerance = 0.0001 m

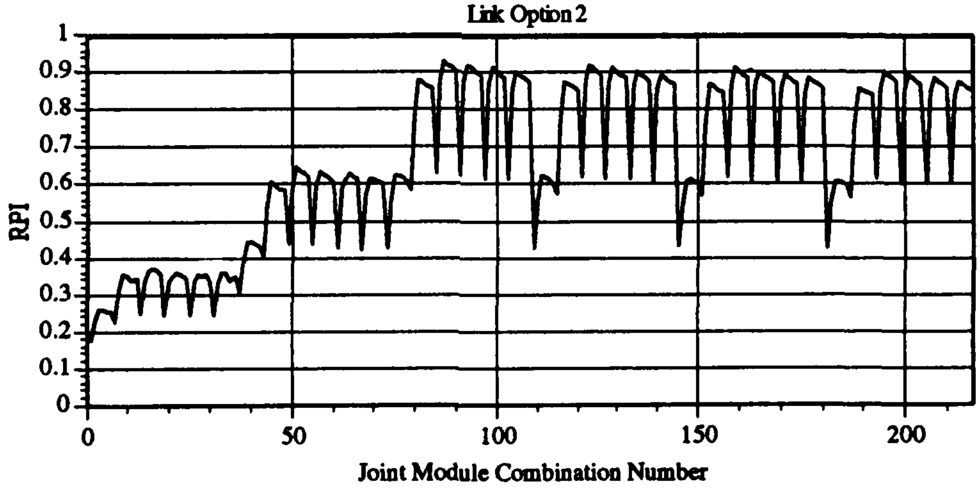


Figure 6.4. RPI by Joint Module Combinations: Link Option 2
$l_1 = 1.0$ m, $l_2 = 0.5$ m, Tolerance = 0.0001 m

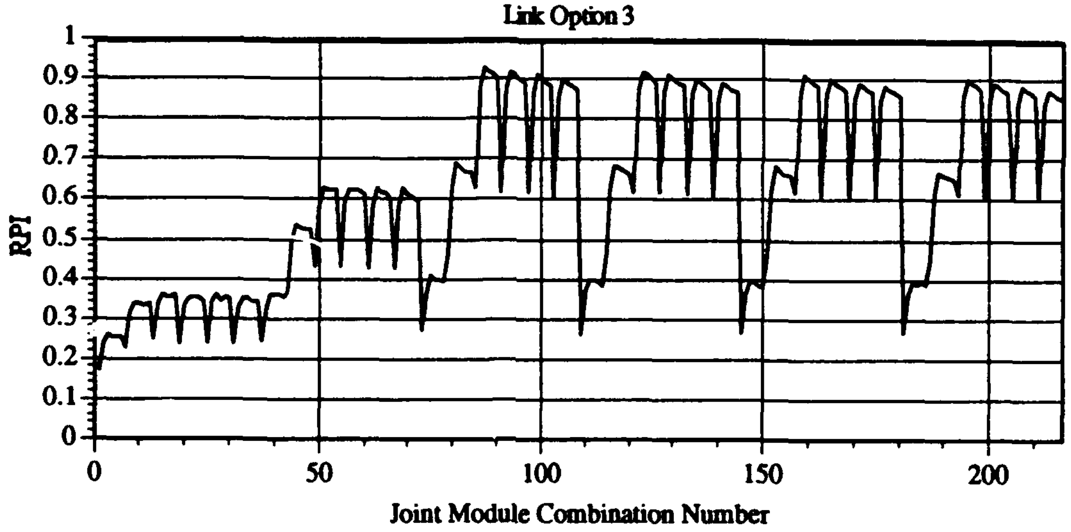


Figure 6.5. RPI by Joint Module Combinations: Link Option 3
$l_1 = 0.5$ m, $l_2 = 1.0$m, Tolerance = 0.0001 m

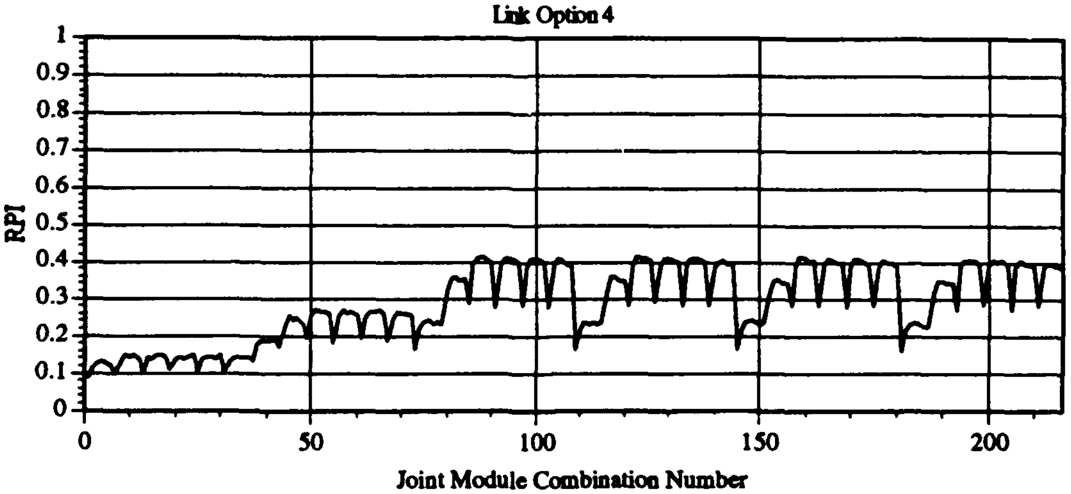


Figure 6.6. RPI by Joint Module Combinations: Link Option 4
$l_1 = l_2 = 0.75$ m, Tolerance = 0.001 m

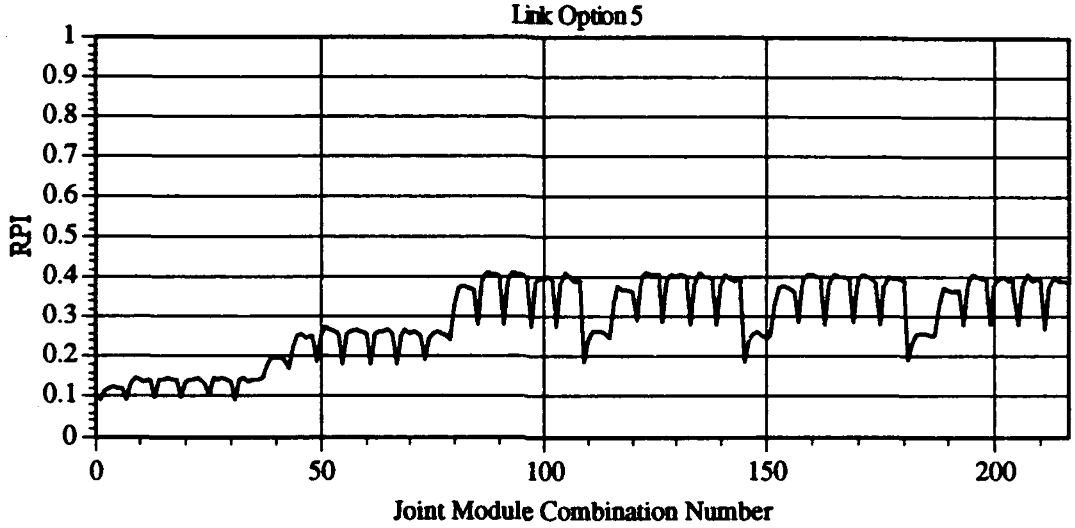


Figure 6.7. RPI by Joint Module Combinations: Link Option 5
$l_1 = 1.0$ m, $l_2 = 0.5$ m, Tolerance = 0.001 m

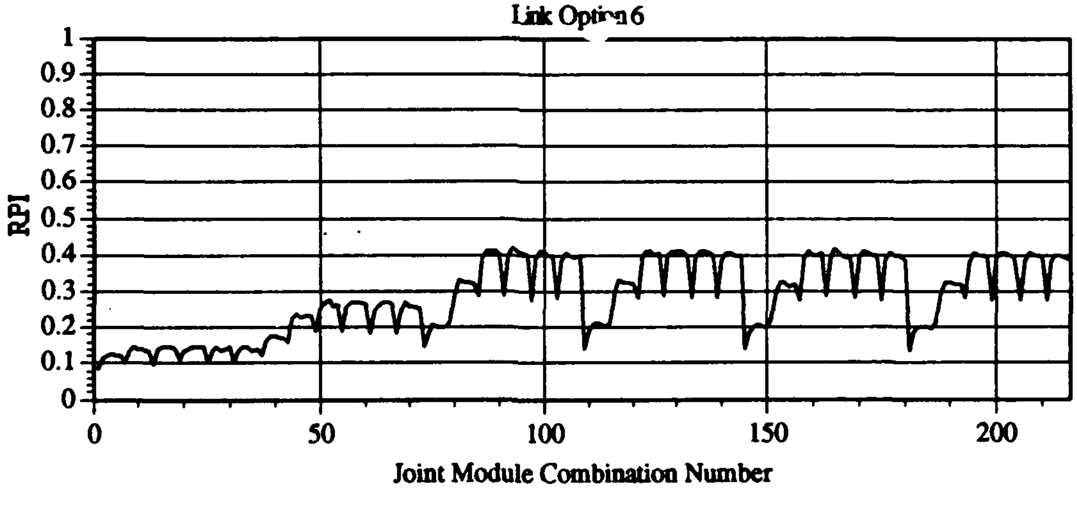


Figure 6.8. RPI by Joint Module Combinations: Link Option 6
$l_1 = 0.5$ m, $l_2 = 1.0$ m, Tolerance = 0.001 m

It is useful to compare the results of Table 6.4 with optimizations about each component of the RPI as suggested above. Two possibilities present themselves for optimization: the first is maximizing the hardware reliability subject to constraints on the kinematic reliability. The second is maximizing the kinematic reliability subject to constraints on the hardware reliability. Conceptually, these optimizations can be performed analytically if the hardware and kinematic reliabilities can be expressed analytically. However, since the case study example is empirical, only empirical optimizations can be made. One aspect of this particular exercise must be noted: R_k is a stochastic variable with a standard deviation. In general, the failure rates of Tables 6.2 and 6.3 will be estimates based on test data having their own means and variances. In this particular case, R_h is considered a constant (since it was evaluated from an assumed known distribution) and the only variability in the value of the RPI comes from the kinematic reliability although this would not be true in a general case. This variability prevents a deterministic optimization from providing an absolute answer, since the variation can cause the optimum to vary widely over the design space. However, performing a deterministic optimization using the values of R_k at this stage is useful to observe the general tendencies of the behavior of the RPI when compared to its components.

We can form an objective statement for the first optimization as

$$\max f(\text{link, base, elbow, wrist}) = R_h \tag{6.6}$$

subject to

$$g(\text{link, base, elbow, wrist}) = R_K \geq \begin{cases} 0.85 & \text{for Link Options 1-3} \\ 0.40 & \text{for Link Options 4-6} \end{cases} \tag{6.7}$$

where the constraint on R_k was chosen arbitrarily. Table 6.5 summarizes the results of this numerical optimization.

Table 6.5. Optimization of Hardware Reliability Subject to Kinematic Reliability Constraint (* Denotes Overall Optimum with $R_k \geq 0.85$)

Link Option	Joint Option	R_h	R_k	RPI
1	3, 3, 2	0.939836	0.9362668	0.8799372
2	3, 3, 2	0.939836	0.9389334	0.8823993
*3	3, 3, 2	0.939836	0.9397331	0.883195
4	3, 3, 2	0.939836	0.4191334	0.3929558
5	3, 3, 2	0.939836	0.4225334	0.397092
6	3, 3, 2	0.939836	0.4151998	0.3902197

An alternative optimization formulation optimizes the kinematic reliability subject to a constraint on the hardware reliability. This optimization is formulated as

$$\max f(\text{link, base, elbow, wrist}) = R_k \tag{6.8}$$

subject to

$$g(\text{link, base, elbow, wrist}) = R_h \geq 0.92 \text{ at } 100 \text{ hours} \tag{6.9}$$

where the constraint on R_h was chosen arbitrarily. Table 6.6 summarizes the results of this empirical optimization. The dual joint options values for link options two and three denote equal optimal values.

When optimizing for hardware reliability, the same optimum point was found for all link options: Joint module 3 in the base and elbow position and joint module 2 in the wrist. This optimization provides for a higher hardware reliability than the RPI optimization but a lower kinematic reliability. The resulting RPI values were lower as well indicating the loss in precision. This makes sense since we where only optimizing one component of the RPI. A global optimum occurred at link option three, however the standard deviation of the kinematic reliability at this design option is 0.00277 which is greater than the difference between the nearest point at link option two. This again indicates that deterministic optimizations do not apply to stochastic variables. Similar results were obtained optimizing on the kinematic reliability as seen in Table 6.6. All of the joint options indicated had the same hardware reliability and the differences in the optimum values are within one standard deviation.

Table 6.6. Optimization of Kinematic Reliability Subject to Hardware Reliability Constraint

Link Option	Joint Option	R_h	R_k	RPI
1	3, 4, 3	0.9212	1.0	0.9212
2	(3,4,3); (4,3,3)	0.9212	1.0	0.9212
3	(3,4,3); (4,3,3)	0.9212	1.0	0.9212
4	4, 3, 3	0.9212	0.4490666	0.4136815
5	4, 3, 3	0.9212	0.45093	0.4153906
6	3, 4, 3	0.9212	0.4525332	0.4168853

An alternative optimization is to maximize both RPI components. This is a global, unconstrained optimization over the design space and can be formulated as

$$\max f(\text{link, base, elbow, wrist}) = R_h + R_k \tag{6.10}$$

The results of this optimization are presented in Table 6.7.

Table 6.7. Global Optimization Results for Equation (6.10) Objective

Link Option	Joint Option	f	R_h	R_k	RPI
1	3, 3, 3	1.9303276	0.9304	0.999867	0.9303369
2	3, 3, 3	1.9034	0.9304	1.0	0.9304
3	3, 3, 3	1.9300838	0.9304	0.9995998	0.9301116
4	3, 3, 3	1.3707278	0.9304	0.4402668	0.4096511
5	3, 3, 3	1.3777712	0.9304	0.4473332	0.4162158
6	3, 4, 3	1.3737592	0.9212	0.4525332	0.4168853

These results provide identical identification of the optimum point for each link option except for link option 4. Looking at link option 4, the maximum RPI occurs at the joint module combination (4, 3, 3). The objective function of Equation (6.10) maximization selects joint module combination (3, 3, 3). Again, the difference in the values results from the variation in the kinematic reliability. This limited data example indicates that the RPI has a tendency to the same maximum that a deterministic optimization on both RPI components will find. While not a one-to-one correlation on maximum points, the optimum configurations are very close in their characteristics, indicating "adjacent" points. This indicates confidence can be placed in the RPI to identify a configuration that posses a satisfactory trade-off between precision and system reliability.

The optimization studies show that the RPI cannot reliably be used as an optimization criteria in a deterministic optimization over the design options to select an "optimum" set of modules. This is due to the variability of the RPI components. It is extremely useful, however, in giving the designer an indication of how well the system is performing compared to other configurations as a "rule-of-thumb" type of design measure. This study indicates that if optimization of the RPI is desired, it should be probabilistic and that an in-depth study of the effects of the individual components of the RPI during probabilistic optimization should be carried out.

6.4. Design Option Studies

The next question to examine considers the impact the RPI can have on the selection of the design configuration. If we examine the plots of the RPI vs. the joint module combinations in Figures 6.2 through 6.8, we see a definite difference between the different joint modules and the locations they are in. The most obvious aspect is having joint module 1 (highest hardware reliability and lowest tolerance) and joint module 2 have much lower values of the RPI no matter what modules are in the distal locations. The deep dips are also due to the use of joint modules 1 and 2 in the distal locations. The standard deviations of the joint modules also have a sgnificant impact. Figures 6.3 through 6.8 give a clear indication of when the kinematic reliaiblity component of the RPI dominates the hardware reliability. For standard deviations of the joint modules greater than or equal to $\sigma_{\theta_i} = 0.05$, the RPI is dominated by the lack of accuracy in the modules. When the standard deviations are less than 0.05, the system is accurate enough and the hardware reliability becomes important.

Due to this characteristic, we can immediately drop the use of joint modules one and two from consideration for use in the configuration. This observation also has a statistical basis as well (see Section 6.5). By removing two joint modules from consideration, the joint module combinations under consideration drop from 216 to 64, a 70% reduction in the design space. It is immediately apparent that the RPI, even if a true optimum point cannot be determined, can drastically reduce the design space when trying to determine the configuration for a particular task. Once the design space has been reduced, the configuration can be chosen using additional criteria such as payload, weight, inertia, etc. as was done in Ambrose and Tesar [4].

The locations of the maximum and minimum values from Table 6.4 are intuitive: the maximum actuator combinations choose the mid-range modules, trading off reliability with precision. It also shows that the RPI is extremely sensitive to the amount of error allowed at the end-effector. Additional studies need to be performed on more realistic systems to provide for a better description of this sensitivity.

We also need to understand how the RPI relates to the other measures of the system design. The starting place for this question is an examination of the Jacobian of the manipulator. If we describe the kinematics of the end-effector in general terms as

$$\begin{aligned} y_1 &= f_1(x_1, x_2, \ldots, x_n) \\ y_2 &= f_2(x_1, x_2, \ldots, x_n) \\ &\vdots \\ y_6 &= f_6(x_1, x_2, \ldots, x_n) \end{aligned} \tag{6.11}$$

where y_1 through y_6 represent the generalized position and orientation of the end-effector and x_1 through x_n represent the generalized joint variables. We can then write the manipulator Jacobian as

$$J = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & & \vdots \\ \vdots & & \ddots & \vdots \\ \frac{\partial f_6}{\partial x_1} & \frac{\partial f_6}{\partial x_2} & \cdots & \frac{\partial f_6}{\partial x_n} \end{bmatrix} \tag{6.12}$$

For the 2-DOF problem presented in Chapter 5, the Jacobian is

$$\begin{aligned} J = \begin{bmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{bmatrix} &= \begin{bmatrix} \frac{\partial x}{\partial \theta_1} & \frac{\partial x}{\partial \theta_2} \\ \frac{\partial y}{\partial \theta_1} & \frac{\partial y}{\partial \theta_2} \end{bmatrix} \\ &= \begin{bmatrix} -l_1 \sin\theta_1 - l_2 \sin(\theta_1 + \theta_2) & -l_2 \sin(\theta_1 + \theta_2) \\ l_1 \cos\theta_1 + l_2 \cos(\theta_1 + \theta_2) & l_2 \cos(\theta_1 + \theta_2) \end{bmatrix} \end{aligned} \tag{6.13}$$

Examining Equation (5.22), we find we can isolate Jacobian terms in the variances of the end-effector position and we can write the expansion as

$$\begin{aligned} \sigma_x^2 &= \left[\frac{\partial x}{\partial l_1}\right]_{\bar{x},\bar{y}}^2 \sigma_{l_1}^2 + \left[\frac{\partial x}{\partial l_2}\right]_{\bar{x},\bar{y}}^2 \sigma_{l_2}^2 + \bar{J}_{11}^2 \sigma_{\theta_1}^2 + \bar{J}_{12}^2 \sigma_{\theta_2}^2 \\ \sigma_y^2 &= \left[\frac{\partial y}{\partial l_1}\right]_{\bar{x},\bar{y}}^2 \sigma_{l_1}^2 + \left[\frac{\partial y}{\partial l_2}\right]_{\bar{x},\bar{y}}^2 \sigma_{l_2}^2 + \bar{J}_{21}^2 \sigma_{\theta_1}^2 + \bar{J}_{22}^2 \sigma_{\theta_2}^2 \\ \sigma_{xy} &= \left[\frac{\partial x}{\partial l_1}\frac{\partial y}{\partial l_1}\right]_{\bar{x},\bar{y}} \sigma_{l_1}^2 + \left[\frac{\partial x}{\partial l_2}\frac{\partial y}{\partial l_2}\right]_{\bar{x},\bar{y}} \sigma_{l_2}^2 + \bar{J}_{11}\bar{J}_{21} \sigma_{\theta_1}^2 + \bar{J}_{12}\bar{J}_{22} \sigma_{\theta_2}^2 \end{aligned} \tag{6.14}$$

where the subscripts on the derivatives and the bars over the Jacobian components indicate they are evaluated at the mean values of the parameter of interest. No second order terms are apparent in the variances, thus, Equation (6.14) shows that the vaiiance and covariance of the end-effector bivariate normal distribution are first-order properties of the manipulator. Equation (6.14) also contains terms concerning the derivatives of the end-effector locations with respect to the other kinematic variables as well, which suggests that a pseudo-Jacobian the other kinematic variables can be formed. This demonstrates there is a direct relationship between the variance of the RPI (and thus its realized value at any point) and the Jacobian. A criterion based upon the variances as driven by the Jacobian may also be useful during the design of a modular robotic system to help select motion programs which minimize the end-effector variance. Equation (6.14) indicates the formulation for accuracy when all the kinematic variables are random. However, when one is interested in the repeatability, only the joint variable terms remain and the variances become a function of the Jacobian elements, exclusively. This can be used to determine the most repeatable locations in the manipulator workspace and provide a much better insight into the selection of the various design options.

6.5. Significance Studies

To validate the observations of the preceding two sections regarding the use of the RPI to reduce the design space, we require a statistical testing of the data. Since the data is sorted into four easily identifiable classifications, Analysis of Variance (ANOVA) presents itself as a logical means of studying the differences between thc modules and their effect on the Reliability Performance Index. An excellent reference for ANOVA is Neter, Wasserman, and Kutner's text *Applied Linear Statistical Models* [114]. ANOVA is a statistical testing technique used to determined if the effects of various factors are statistically significant. In this case, the factors will be the link options, and the three joint locations: base, elbow and wrist, each with six levels (the different link options and six different joint modules).

There are several different hypothesis tests that can be made using the ANOVA technique. The first is to determine if the effect of a particular factor is a significant contributor to the model. The next test that can be performed is to determine which levels of the different factors are significantly different from the others. Both of these tests are carried out on the data generated from the simulation of the system of Figure 6.1 following the trajectory of Figure 6.2. The data of Figures 6.3 to 6.8 were stored on diskette and analyzed using the SAS Statistical Analysis Software System [SAS, 1990]. The ANOVA table presented in Table 6.8 was determined from the data.

The first hypothesis test performed is to test if the interaction effects B*L, E*L, and W*L are significant and should be included in the overall statistical model. The null hypothesis for this test is H_o: interaction effects not significant. The test statistic for this test is $F_{25,1200,.95} \cong 1.64$. The rejection region for the test is $F^* > F$. In this case, all of the interaction effects contribute significantly to the model. However, the mean squares and the statistic F^* for each of the interactions are much smaller than for the main factors. To investigate the main factors, the interaction contributions are added to the error which results in the ANOVA table of Table 6.9.

Table 6.8. Analysis of Variance Table for the Reliability Performance Index Case Study

Error Source	Degree of Freedom	Sum of Squares	Mean Square	F*
Base (B)	5	20.42	4.08	2387
Elbow (E)	5	7.8	1.56	912
Wrist (W)	5	4.2	0.84	491
Link Option (L)	5	41.5	8.3	4856
B*L	25	2.6	0.1	60.6
E*L	25	1.8	0.07	42.6
W*L	25	0.06	0.02	15.5
Error (E)	1200	2.053	0.0017	
TOTALS	1295	81.111		

To test for significance of the main effects (B, E, W, and L) we form a similar null hypothesis which results in the test value $F_{5,\ 1275,\ 0.95} = 2.57$. Using the same rejection region, we find all of the main effects contribute significantly to the statistical model. We can now test for the significance of the individual levels of the main effects (which was the test we were after). This test is called the Tukey Studentized Range test [114, 136]. This test determines a significant difference between the means of the effects levels and then makes pairwise comparisons to determined if there is a statistical difference between the two levels. The results of this test is presented in Figures 6.9 through 6.12. The brackets under the number lines in Figures 6.9 through 6.16 represent statistical differences between the adjacent values on the number line above them. If the bracket overlaps two values, the test could not statistically differentiate between those two values. For instance, in Figure 6.9, the mean RPI for joint modules 1 and 2 are statistically different since the ranges shown on the brackets underneath do not overlap both module locations on the line. Thae figure shows that there is no statistical difference between joint modules 6, 5, and 4, but there is a statistical difference between joint modules 6 and 3.

Table 6.9. Analysis of Variance Table for the Reliability Performance Index Case Study (Interactions Added to Error)

Error Source	Degree of Freedom	Sum of Squares	Mean Square	F*
Base (B)	5	20.42	4.08	729.5
Elbow (E)	5	7.8	1.56	278.97
Wrist (W)	5	4.2	0.84	150.2
Link Option (L)	5	41.5	8.3	1484.01
Error (E)	1275	7.131	0.00559	
TOTALS	1295	81.111		

From the examination of Figures 6.9 through 6.11, we can see that there is a definite statistical difference in the mean of the RPI when joint modules 1 and 2 are used in any position. The mean value of the RPI is lower for these two joint modules and gives a statistical basis for the observations of Sections 6.4 and 6.5. This supports the conclusion that joint modules 1 and 2 should be removed from the set of possible joint modules. Since there is not any definitive statistical difference between the other joint modules, other design criteria should be used to select the final positions of the joint modules.

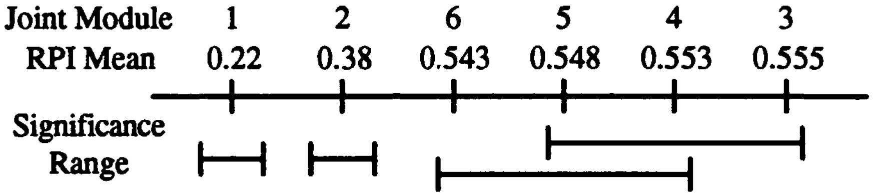


Figure 6.9. Tukey Studentized Range Test on Levels of Base for the Reliability Performance Index

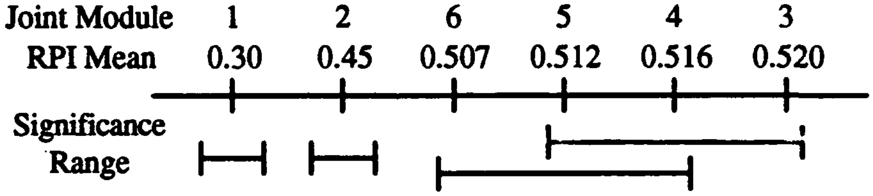


Figure 6.10. Tukey Studentized Range Test on Levels of Elbow for the Reliability Performance Index

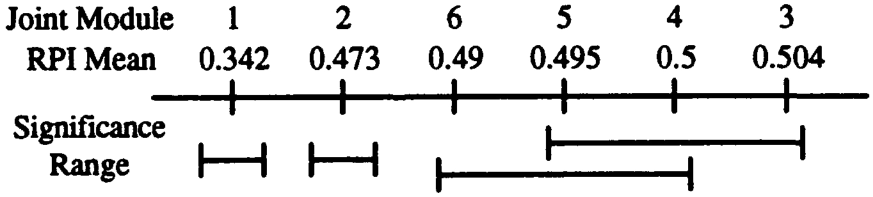


Figure 6.11. Tukey Studentized Range Test on Levels of Wrist for the Reliability Performance Index

Figure 6.12 shows the result of the Tukey test for the Link Options. As noted in the previous section, the difference in length tolerance between the options is significant, however, this test indicates a statistical difference between the first three link options. We can conclude from this test that link option two (which is where the maximum RPI value was located) has a statistically higher mean RPI than the other two tight tolerance link options and should be used. This differentiation does not exist between the wider tolerance suggesting that having large differences between the design levels enables the RPI to better glean the reduced design space for all the design options.

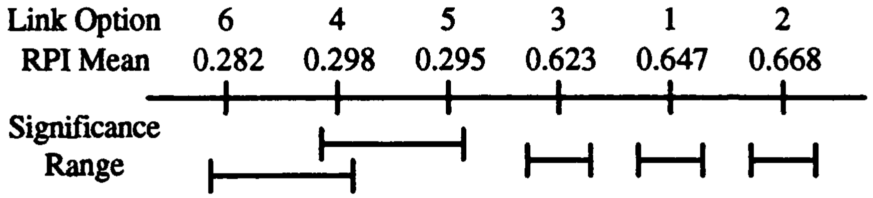


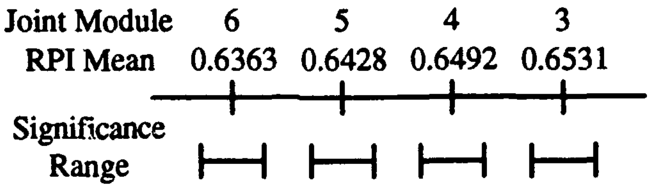
Figure 6.12. Tukey Studentized Range Test on Levels of Link Option for the Reliability Performance Index

After removing joint modules 1 and 2 from consideration for inclusion into the configuration, the ANOVA and Tukey tests were run again. The removal of the two modules reduced the number of data points to 384 from 1296. The resulting ANOVA table is presented in Table 6.10.

To test for significance of the joint location effects (B, E, W) the F test value is $F_{3,\ 369,\ 0.95} = 2.6$ and for the link options it is $F_{5,\ 369,\ 0.95} = 2.3$. Both these values reject the null hypothesis and again the levels of each factor can be examined with the Tukey Tests. Figures 6.13 through 6.16 show the results of the Tukey Tests on the reduced data set.

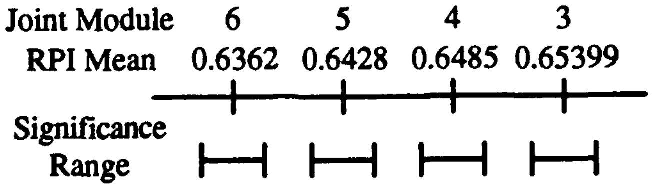
Table 6.10. Analysis of Variance Table for the Reliability Performance Index Case Study (Reduced Data Set)

Error Source	Degree of Freedom	Sum of Squares	Mean Square	F*
Base (B)	3	0.0156	0.00521	132.49
Elbow (E)	3	0.0167	0.00558	141.93
Wrist (W)	3	0.0197	0.00658	167.24
Link Option (L)	5	22.92	4.583	116,615
Error (E)	369	0.0145	0.0000393	
TOTALS	383	22.98		



Joint Module	6	5	4	3
RPI Mean	0.6363	0.6428	0.6492	0.6531

Figure 6.13. Tukey Studentized Range Test on Levels of Base for the Reliability Performance Index (Reduced Data Set)



Joint Module	6	5	4	3
RPI Mean	0.6362	0.6428	0.6485	0.65399

Figure 6.14. Tukey Studentized Range Test on Levels of Elbow for the Reliability Performance Index (Reduced Data Set)

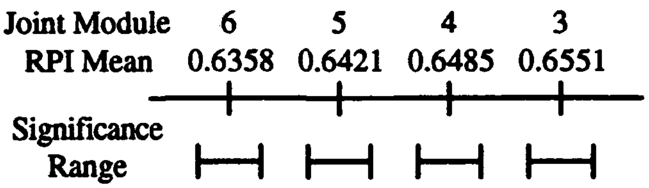


Figure 6.15. Tukey Studentized Range Test on Levels of Wrist for the Reliability Performance Index (Reduced Data Set)

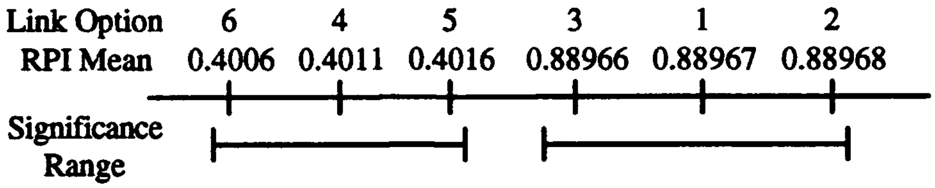


Figure 6.16. Tukey Studentized Range Test on Levels of Link Option for the Reliability Performance Index (Reduced Data Set)

The Tukey Tests show that with the reduced data set, we can now see distinct statistical differences between the joint modules RPI means. On the basis of the RPI, we would select joint module 3 for use in all three joint locations. The penalty for the reduction in the data is seen in Figure 6.16, where we have lost delineation between the link options in each tolerance range. This is correct statistically, since the power of the tests where reduced when data was deleted.

6.6. Summary

This case study has shown that the framework for the Reliability Performance Index developed in Chapter 5 is a useful empirical tool during the selection and design of a configuration of robotic modules of varying reliability and precision. It proves useful as a design guideline to describe how a particular combination or configuration of modules will perform with respect to the system reliability moderated by its precision. While the framework for the RPI was developed, the data to fully investigate it effects on realistic systems is not available.

The RPI does not lend itself to deterministic optimization due to the stochastic nature of the components of the RPI. However, it readily allows for the rejection of module and link combinations as unsuitable, thus reducing the design space. In the case study, analysis of the RPI over the design space suggested that joint modules 1 and 2 could be removed from consideration and that link option 2 presented itself as the best alternative to provide for maximum precision as measured by the RPI. This is a 70% reduction in the joint module design space and an overall reduction of 95% (by selecting link option 2). Additional tests on a reduced data set (formed by removing joint modules 1 and 2) showed that we could discriminate between the remaining joint modules with the recommendation to use joint module 3 in all joint locations. However, discrimination was lost on the link options since we had much less data to work with.

It was also seen that the larger the design space and the more different the module characteristics, the better discrimination was obtained using the RPI. The use of the RPI allows the designer to eliminate a large portion of the design space and a final design can be selected using other operational design criteria such as described in Section 5.1.1.

Several recommendations concerning the RPI can be made. First, a more thorough investigation of the interactions between the components of the RPI needs to be made. The development of the RPI assumed that while hardware failures will cause a failure in the kinematic sense, the hardware and kinematic reliabilities were independent, which lead directly to the formulation of Equation (5.32). This is not actually the case, since a failure in the kinematic sense can be dependent upon hardware failures. This dependency should be understood to be able to improve the relationship defined in Equation (5.32).

The sensitivity of the RPI to the actual design parameters also requires investigation. A way of applying the RPI to monolithic systems may show the same results as to selecting modular configurations, but this requires a direct mapping to design parameters. An analytic formulation is required and will allow the development of the sensitivity of the RPI to the module level design parameters.

Related to the design of the modules, is the development of a modular metrology. This issue is extremely important to the use of the RPI by developing the reliability and tolerance data for the modules that is required to calculate the RPI during configuration design. The robotic component reliability data-base suggested in Chapter 4 is an integral part of this needed metrology effort, as well as the quantification of module tolerances such as can be used for the calculation of the kinematic reliability of a manipulator.

An additional area of investigation for the RPI is the determination of the effects of the factor interactions noted in Section 6.5. These interactions were subsumed into the error, but they may give the best representation of the dependency structure between the components of the system. This dependency may have an impact upon the design solutions indicated.

The RPI was seen to provide unreliable results when used as a deterministic design optimization criterion. This may not be the case when probabilistic optimization methods are used since the RPI is stochastic in nature. The RPI should be investigated in terms of probabilistic optimizations. Additionally, more realistic examples in the application of the RPI should be developed to determine if the results noted in this small case study can be generally extended.

To take complete advantage of a modular system, the customer requires a way to easily and quickly determine new configurations. The RPI shows promise in reducing the design space for the selection of modular configurations but needs to be incorporated into an on-line computer-aided design system, such as that being developed at the University of Texas. This will allow the designer to immediately make the indicated configuration or design changes and immediately see the changes in the design criteria, including the RPI.

CHAPTER 7: CONCLUSIONS AND RECOMMENDATIONS

This chapter contains the essence of the Reliability and Maintainability (R&M) Roadmap for Modular Robotics Systems. A summary of the previous chapters of this dissertation is provided along with the primary recommendations. The last section provides a list and explanation of the original contributions of this research effort.

7.1. Objectives and Literature Review Summary

This section summarizes the dissertation and the primary conclusions reached during this research. It includes the objectives of this research as well as a brief description of the literature review.

The goal of this research effort was to develop a framework to provide for the inclusion of reliability and maintainability into the design and integration of modular robotic systems. The roadmap has three main components. First, to present and integrate reliability analysis and design tools and techniques into the robotic system design process, showing how the tools and techniques developed in the subsidiary disciplines of electrical and mechanical design can be applied in the design of robotic systems. Secondly, the role of technology in reliability improvement is addressed by examining the generic reliability levels of the component technologies that make up robotic systems and by the examination of how those technologies contribute to the overall system failure rate and reliability design issues, make recommendations for research efforts that will improve the reliability of modular robotic systems. The third and final component of the roadmap is the preliminary development of a tool (the Reliability Performance Index) to quantify the reliability of modular robotic systems during design and integration. The research objectives used to accomplish this goal are presented in Table 7.1.

Table 7.1. Modular Robotic System R&M Roadmap Research Objectives

Research Objectives
Review current robotic and modular system reliability and maintainability levels and design methods.
Provide an overview of how to apply R&M design tools and techniques can be applied to the robot system design problem.
Rank robotic component technologies for research potential to improve modular robotic system reliability and maintainability.
Develop and mathematical framework for a Reliability Performance Index (RPI) to include reliability into the design and integration of a modular robotic system.
Validate the use of the RPI through sensitivity and optimization studies.
Demonstrate the use of the RPI through the development and application of a case study.
Provide prioritized listings of research rankings to emphasize technological development improving modular robotic system reliability.

The literature survey presented in Chapter 2 found that interest in the reliability of robotic systems has been scattered and disconnected. The efforts related resulted mainly from safety concerns and were concentrated on robotic technologies in the early 1980's. The literature review also points to a general lack of awareness of the use of R&M tools and techniques during the design of robotic systems. A survey of robot system manufacturers seems to bear out this observation. Reliability improvement efforts are mainly sporadic with systematic approaches only evident in a few large manufacturers. Also of concern is the lack of reliability data presented in the literature as well as that available from industry sources. This lack of data has prevented this research effort from truly addressing the technological needs of the robotics community from the reliability perspective.

The literature review also considered the lessons that can be learned in the design of general modular systems. Case studies reveal that modularity may be considered simply a design requirement or characteristic. No specific design methodologies exist for modular systems. However, the imposition of the modularity constraint on a system does imply several design principles that must be followed. As described in Section 2.3.2, the four modular design principles are: module commonalty and functional independence, module interface control and standardization, the reduction of life-cycle-cost through increased reliability of modules, and the choice of the level of modularity should be based on the minimization of life-cycle-cost providing the maximum flexibility in the system at minimum cost. These modular design principles form the basis of a paradigm for the improvement of robotic system reliability as presented in the roadmap.

7.2. Summary and Conclusions of the Roadmap

The primary purpose of the roadmap is the development of a long-range research plan for the development and implementation of technologies to improve the reliability of modular robotic systems. As such, there are two main thrusts: the programmatic and organizational thrust, and the technology thrust.

7.2.1. Programmatic and Organizational Aspects of the Roadmap

The reliability of systems is due both to the inherent reliability of the design and to the degradation aspects of the environment in which it is used. This includes the maintenance concept under which the system operates. In Section 3.1.4, the various corporate functions were reviewed and the setting of reliability goals for the organization was discussed. Commitment of the top management level is considered critical for the success of a concentrated reliability management program.

Additionally, the different organizational structures where presented with the advantages and disadvantages of each identified. For smaller developmental organizations, the product oriented organizational structure is best since it is the most flexible and allows the organization to quickly take advantage of the newest and most rapidly changing technologies. Larger organizations, concerned about efficiency, can take advantage of this flexibility by using a matrix organization and minimizing bureaucracy between the design staff and the top decision makers.

At the product level, past experience in the Department of Defense and commercial environments suggest that good product reliability and maintainability is enhanced by the judicious establishment of a R&M control program. An excellent description of the various reliability and maintainability tasks that can make up a R&M program can be found in MIL-STD-470 [102] and MIL-STD-785 [104]. These two documents describe suggested R&M tasks and what benefits can be gained by their implementation. These tasks are listed in Tables 3.3 and 3.4 with a basic description of all the tasks given in Section 3.1.4.

Any R&M improvement must be balanced against the return-on-investment (ROI) in the cost of implementing a particular program task to the possible reliability improvement that may result. This question is best addressed during the development of a life-cycle-cost (LCC) model of the system. This model is used during design to provide the designers and management with a basis for making decisions based on the ROI. Section 3.1.3 reviews the LCC model and its relation to reliability and presents a possible LCC model that can be used for a modular robotic system.

7.2.1.1. Robotic System Design Process

The culmination of the reviews of design case studies shown in Tables 3.10 through 3.13 results in the development of a robotic system design process that includes the suggested sequence for the application of the various reliability tasks during the design process. This design process is illustrated in Figure 3.7. It includes the development of the reliability and maintainability specifications described in Section 3.2.2 and the inclusion of preliminary reliability models during the design process. The preliminary models can be based on data obtained from previous designs or on generic data such as found in MIL-HDBK-217 [96] and the *Non-Electronic Parts Reliability Data Handbook* (NPRD) [131]. The primary conclusion of the use of reliability models during design is:

> The use of reliability models during design is best suited to providing an indication of the relative reliability levels of the various

> design alternatives. This is due to the lack of data specific to the system and its operating environment during the design stage. Testing to insure satisfaction of the design specifications will indicate the actual achieved level of reliability.

To insure the robotic system will maintain its inherent reliability during operation, controls should be implemented during production and use. These controls can take the form of statistical process controls, on-condition monitoring as discussed in Section 3.3.2.3, and maintenance and inspection concepts as discussed in Sections 3.3.2.3 and 3.3.2.4.

7.2.1.2. Statistical Independence and Modularity

The issue of the reliability of modular systems has been addressed from the basis of having independent life distributions between modules. In electronic, software, and power distribution systems this is a valid assumption. However, in the examination of the independence assumption in Section 3.5, we concluded that this assumption is not generally valid for robotic systems since the loads each module in the manipulator experiences is dependent upon where it is in the modular configuration. The characterization of the reliability of a system composed of independent modules is a relatively straightforward exercise. However, when we introduce dependencies between the module life distributions, we find that we no longer have the tools to fully understand the impact that modularity has on the system reliability. These tools are the theory and methods to handle the general case of the algebra of *n*-dimensional random vectors with general dependency structures between the components of the vector.

The discussion of Section 3.5 resulted in the following primary conclusion:

> To truly understand the implications modularity has on the system reliability, intensive development in the theory of the algebra of random variable with general dependency structures is needed. Also, the methodologies for determining the dependencies and the measurements needed are required developments.

7.2.2. Technology Improvement

The second thrust of the R&M roadmap is the identification and prioritization of the technological aspects of reliability improvement for modular robotic systems. Three general areas were examined: architectural issues, component technologies, and the prioritization of fault-tolerant schemes for reliability impact.

7.2.2.1. Architectural Issues

Two architectural issues are of interest: modularity and parallel structures. A maintainability analysis of the proposed modular interface standards available in the literature was made in Section 4.2.1 and compared with the time it took to replace a motor in several monolithic robotic systems. The results presented in Table 4.1 indicated that a vast improvement in replacement times can be achieved with a modular robotics system. However, as the reliability of the system approaches unity, the MTBF becomes very large, dominating the availability equation, and the availability approaches 100%. This means the payoff of modular robotic systems will be in reconfigurability as the reliability of the systems become high. Suggested characteristics for a standard modular robotic interface were also presented in Section 4.2.1.

The analysis of parallel structures suggest that due to the load carrying capability of the structure, structural failure is extremely unlikely, and thus there is no contribution to a failure rate from using a parallel structure. Additionally, fault-tolerance schemes require a redundancy be present in the system to allow for reconfiguration if a fault occurs. Parallel robotic systems help provide the necessary redundancy to provide for mechanical fault-tolerance in a manipulator. Thus, parallel structures should be used for performance and fault-tolerance reasons, not for reliability improvement.

7.2.2.2. Component Technologies

a. *Communications Systems (Section 4.3.1.1).* The major failure driver of robotic communication systems is the cabling connecting the sensors and motors with the computer and power amplifiers. The use of high reliability cabling options and communications architectures supporting them are encouraged. The use of data multiplexing will tremendously reduce the number of wires needed in the robotic system. This is an imperative for a modular robotic system. Coaxial cabling was found to have a lower failure rate than fiber optic cables and can be tapped in parallel without attenuation as they run through the robot structure rather than terminated and retransmitted in each module as required for fiber optic cables.

b. *Interfaces (Section 4.3.1.2).* Interfaces should be kept as simple as possible to minimize the failure rate. Electrical connections should mate and demate at the same time as mechanical connections are made. Real estate within the interface will be extremely limited, especially if large power densities are required so the number of paths through the interface must be kept at a minimum. Fluidic connectors should not be included. The interfaces should be as stiff as possible to ensure accuracy and built-in alignment mechanisms should be included. Strain relief should be provided for internal cabling to prevent bending stresses and for any external cabling that may be required for end-effector tooling.

c. *Sensors (Section 4.3.2).* Use optical encoders providing both position and velocity information. Optical encoders provide the best reliability performance and also provide digital outputs requiring no additional processing. Cost issues are also important in the selection of a sensor suite. The return-on-investment in reliability must be cost justified.

d. *Actuators (Section 4.3.3).* Electric motor technology has progressed to the point where manufacturers of brush commutated motors claim to provide lives to a billion revolutions. At 33 revolutions per minute, this comes out to a maximum life of 8,400 hours (not the mean life). In other word, this is a minimum failure rate of 118 failures per million hours. One can reasonable expect lower failure rates and longer lives if derated. However, this failure rate still is an order of magnitude above those recorded by brushless ac and dc motors [131]. Thus, the obvious choice for enhanced reliability is brushless motor technology. An additional impact on the reliability of motors is heat dissipation. Designers should seek to insure moderate operational temperatures since the failure rates of motor windings double for every 10° increase in operating temperature (see Figures 4.8 and 4.9).

e. *Gearing (Section 4.3.4).* Gear heads have the highest failure rate of all the mechanical technologies considered [131]. If feasible, direct drive architectures should be used since removal of gearing will provide a 500% decrease in generic failure rate. If gearing must be included, the designer should seek to minimize gear wear (which is the primary failure mode) and maximize motor performance which also reduces the stress on the gear train. Cycloidal gear technologies should be the choice to minimize wear and maximize gear reliability.

f. *End-effectors (Section 4.3.6).* End-effectors for modular robot systems should be electrically (or fiber optically in the case of lasers) powered. The standard module interface should be utilized. This prevents the use of fluidic connectors in the modular robotic system.

g. *Structural members (Section 4.3.7 and 4.2.1).* Using deterministic design methodologies, the structure will be over designed. The structural configuration should be based upon performance and fault-tolerant issues. If probabilistic design methods are used, reliability can easily be used as the constraints on the design and will provide an optimal design (reliability vs. weight and strength) of the mechanical structure. However, this is computationally expensive.

h. *Controllers (Section 4.3.8).* Computer hardware is extremely reliable when purchased from quality vendors. In fact, electronic technology reliability improvements have far outstripped comparable improvements in mechanical technologies. However, the electronics have a tendency to be extremely complex especially when using software to provide for fault-tolerance and fault detection and isolation as well as the adaptive control of dexterous manipula'o"s. This makes the controller have higher failure rates than the mechanical system. However, the controller is usually easy to access, troubleshoot and repair. The use of electronic technologies such as Very Large Scale Integrated circuits and built-in-test allow for the maximum resistance to failure. Control algorithms also effect the reliability of the overall system. Robust control laws and adaptive algorithms should be used to provide stability under failure. The use of feedback linearization is not recommended when failure resistance is required.

i. *Software (Section 4.3.9).* Software reliability is enhanced by the use of good software engineering techniques. These techniques include the .use of a structured, top-down, programming methodology prohibiting unconditional branching. The use of code modules allows for easy software maintenance and ease of programming and testing. Software redundancy can be provided by using *n*-version programming on redundant hardware. Additionally, Computer-Aided Software Engineering (CASE) can provide for logically correct and error-free code. Error checking and recovery codes should also be included in the algorithms.

An additional consideration is the use of Robot Programming Languages (RPLs). The RPLs that seem to provide for the highest reliability ale those that are independent of the actual robot and use object oriented programming. The use of artificial intelligence and sensor fusion in the RPLs along with the use of software reliability enhancement techniques will provide for better RPL reliability.

7.2.2.3. Fault-Tolerance Prioritization

Section 4.4 addressed the issue of fault-tolerance and how it relates to the reliability of a system. Fault-tolerance is an architectural issue since additional resources must be provided in the system to allow for the detection and correction of faults and failures within the system. A fault-tolerant system is not necessarily a reliable system as can be seen from in Figure 4.10. The reliability of the system depends upon the coverage of the fault-tolerant scheme which represents the probability that the fault-tolerance works correctly. The coverage is determined from the system reliability model and from the failure modes anticipated.

Three methods for prioritizing fault tolerance schemes were discussed in Section 4.4.3 with coverage analysis presented as the most effective but most difficult and involved method. If coverage is perfect, the reliabilities of the components involved in the fault-tolerant scheme should be used to prioritize the scheme.

7.2.3. Research Emphasis Rankings

The second goal of the roadmap was to provide a prioritization of research topics for the improvement of modular system reliability. These rankings were provided in Section 4.5. The first set of rankings provide the relation of various robotic applications to the reliability of robotic systems. These rankings were provided in Tables 4.17 and 4.18 which show that the need for reliability is greatest in those areas where maintenance is hard to accomplish (space operations, microsurgery, nuclear applications) where good maintainability is required in areas when the need exists to return the robotic system back to service quickly (military applications, service robots). With this information setting the stage, three different rankings are obtained with the results repeated here for convenience. First, Table 7.2 shows the contribution of the robotic component technologies ranked by their contribution to the overall failure rate of robotic systems. Second, Table 7.3 the component contributions to four issues affecting modular robotic system reliability are ranked. Last, Table 7.4 presents a prioritized ranking of research needs for the reliability of modular robotics systems.

7.2.3.1. Component Technology Contribution to System Failure Rate

Table 7.2 presenis the results of a failure rate analysis of a generic seven Degree-of-Freedom (DOF) manipulator based on the generic failure rates presented in Chapter 4. The largest contributor was the gear systems of the manipulator, followed by the power cables, motors, and sensors. The primary conclusion from this ranking is:

> The improvement of motor technology should be supported to reduce the burden on gear systems or allow their removal altogether. Removal of gear systems can increase system reliability by 500% based on a 7 DOF generic system.

Table 7.2. Component Technology Impact on Reliability of Generic Manipulator

Rank	Component Technology	Failure Rate (per 10^6 hrs)
1	Gear Trains	350
2	Power Cables	39.2
3	Motors	18.62
4	Sensors (aggregate value)	12.74

7.2.3.2. Component Technology Contribution to Reliability Design Issues

The results of the second ranking is shown in Table 7.3 which summarizes Table 4.20. Four issues were found to contribute directly to the reliability of modular robotic systems: module complexity, system complexity, reliability data, and interface standardization. These four issues where ranked according to the component technologies described in Chapter 4. The most contribution to the reliability issues was found to be the communication system. The systems making up the control system: communication, sensors, controllers, and software were found to dominate the reliability issues and deserve to overall emphasis. The reason that these systems dominated was that the complexity of the system will be embedded in the control and sensor systems to allow for the modularization of the manipulator structure. Second was the mechanical system as discussed above. Motors and gearing provided the most contribution.

Table 7.3. Design Issues and Component Ranking Results

System/Component Technology	Normalized Ranking
Controller Systems	
Communication Subsystem	1.0
Sensors	0.95
Controller Software	0.81
Controller Hardware	0.76
Structural Systems	
Motors and Actuators	0.7
Gear Systems	0.7
Links	0.43

7.2.3.3. Research Ranking for Reliability Improvement

The ranking of Table 4.22 resulted in the research ranking for reliability stated in Table 4.23 and repeated in Table 7.4 for convenience. First emphasis should be placed on the development of a national robotic component reliability data base to augment the data available for analysis in the robotic environment. One of the largest obstacles in the study of robot reliability and the reliability critical components of robotic systems is the lack of applicable data. This has long been a complaint in the reliability community, however, in the area of robotics, the lack of reliability is critical. As documented in Appendix C, a large amount of effort was expended to try and obtain this data. This lack of data forces researchers to utilize generic data to analyze and generalize research needs. To help alleviate this problem, a standard data base similar to MIL-HDBK-217 and the RADC Non-Electronic Parts Reliability Data Handbook should be developed. Ideally, this database should be located at a university or independent research center and supported by the robotics industry at large, but since most companies consider reliability data to be proprietary, this is unlikely to ever happen. As a substitute, a local reliability data base can be developed and maintained on the systems designed and manufactured at the organizational level. There is a distinct advantage to a local data base since it is based on failures of a specific system in the environment in which it was designed. For modular systems, reliability data is also required at the module level. This type of data can be obtained as part of an integrated metrology investigation during the design and development of the modular architecture.

Second emphasis should be on reliable communications systems and sensors. The thrust of this research should be the minimization of data and power line requirements since the success of a modular robotic system will depend upon the simplicity of the modular interface. These two technologies drive the complexity of the modules and the system as well as determine the makeup of the interfaces between the modules. It is precisely these issues that impact the reliability of modular manipulator systems. Research into reliable sensor technology should continue and should concentrate on environmental protection issues. This includes corrosion resistance, temperature stability, and other environmentally related issues. Research is also required into interface technologies in the areas of load reduction for connectors during mating and demating and during operation of the modular system. This includes increased strength and stiffness while decreasing the weight of the interface. This is an architectural issue. Continued development and implementations of robust, fault-tolerant controllers and control algorithms also promote increased modular robotic system reliability.

Third emphasis should be on motor technology aimed at increasing torque capacity to reduce the loading on and eventually the need for gear systems in the robot. Not only do gear trains exhibit undesirable qualities such as backlash, they also have the highest failure rate (as a generic technology) of the mechanical robotic system. A generic serial 7-degree-of-freedom robot system had a predicted 500% increase in life if the gear trains were removed from the system. This implies that direct drive robotic systems should be used where reliability concerns are paramount. Unfortunately, motor technology is not available to produce the torque-to-weight ratio required for industrial applications. Further research in motor technology, perhaps including room-temperature superconducting materials to maximize magnetic flux, is required to produce the required levels of torque enabling the use of direct-drive actuation in modular robots.

The last area requiring emphasis is the independence question addresses in Sections 3.5 and 7.2.1.2. This effort will require significant financial backing and must be termed as basic statistical research. The benefits will be tremendous, not only for modular robotic systems, but for engineering design in general, because the ability to quantify dependencies will allow much better modeling of the behavior of all engineering systems.

Table 7.4. Priority Listing of Additional Research Needs for Robotic Reliability

Establishment and Maintenance of a National Robotic Component Reliability Data Base
Promote Research and Improvement in Reliability of Robotic Communications Systems and Sensor Technologies
Continue Development of Robust, Fault-Tolerance Control Systems and Algorithms to Provide the Highest Controller Reliability
Improvement in Motor Technologies to Allow Design of Direct Drive Robotic Systems
Improvement in Reliability of Gear Train Technologies
Develop Tools and Methodologies to Quantify and Manipulate Dependent Random Variables to Allow Full Understanding of the Impact the Modularity of a System has on its Reliability

7.3. Summary and Conclusions for the Reliability Performance Index

The framework Reliability Performance Index (RPI) developed in Chapter 5 provides a direct tie from the roadmap effort to the actual design of a modular robotic system. The RPI provides a method to quantify the statistical performance measures of reliability and accuracy during the design process. This index provides an indication as to whether the system is meeting both its reliability and accuracy specifications, expressed as a probability. Under the assumptions of the development, the RPI is the probability that the system will not fail by a certain time and that it will meet a specified tolerance over its workspace or trajectory. During the examination of a case study, the RPI was found to be useful in reducing the design space when selecting a module configuration from a suite of modules by identifying statistical differences between joint module performance and link options as described by the RPI.

7.3.1. Development of the RPI

The main assumption made during the development of the RPI was that a hardware failure will cause a failure in the kinematic sense. If we assume that the kinematic reliability of the system is independent of the hardware reliability of the system, the overall RPI can be described by Equation (5.32). It is this assumption that causes the RPI to be an index, rather than an actual measure of probability, since a dependency between the hardware reliability and the kinematic reliability almost certainly exists. This dependency was not addressed since this work provides only a framework for the RPI. Many pieces are still missing, mainly resulting from the lack of robotic reliability data. The primary conclusion for the RPI development is:

> If the assumption that the kinematic sources of failure are statistically independent of the sources of hardware failure, then the Reliability Performance Index given as
>
> $$RPI = R_h \cdot R_k$$
>
> using the formulation as given in Chapter 5 actually represents the probability of the end-effector being in its permissive region and no failures in the system. This RPI will actually represent the probability of proper and precise operation of the robotic system.

A case study was generated to allow preliminary study of the RPI. The case study system was a 3-DOF planar manipulator (shown in Figure 6.1) that was being commanded to perform the trajectory of Figure 6.2. The objective of the case study was to select a configuration based upon the value of the RPI for a particular configuration of modules. The manipulator could be composed of six different link module combinations and six different joint modules possible in each of three locations resulting in a total of 1,296 possible design combinations. The case study resulted in statistical evidence that reducing the design space by removing two joint modules from consideration improves the mean value of the RPI. The removal of the two joint modules provided enough definition to the statistical tests that a single joint module could be selected that provided the closest value of the RPI to unity, which in this limited case, provided a probability of 0.93 that the specification of position and orientation error as well as no failure occurring in the time of evaluation. The primary conclusion for the RPI was found to be:

> The Reliability Performance Index provides a way to significantly reduce the design space of a modular robotic system and improve the precision and reliability of the system. The RPI has best definition if the module characteristics have large differences between the modules. The RPI is a stochastic value having variability which prevents its use as a deterministic optimization criterion. It does allow for a reduction in the design space which can then be further reduced using other design criteria.
>
> The RPI can be applied to any system that uses measured error as the basis for operation and as such represents a alternative way of quantifying the performance of such a system.

7.3.2. Research Recommendations for the RPI

a. *Alternative Formulations and Validation of RPI Model.* The dependency between the two components of the RPI can have a major impact on the overall formulation of the RPI. Two possibilities exist: first that the serial assumption is true, but a dependence between the hardware and kinematic reliabilities exists, which by Baye's Rule [95] makes Equation (5.32) and extremely optimistic estimate of the actual RPI. The second possibility is that a fault-tolerant scheme included in the system prevents a hardware failure from causing a failure in the kinematic sense. This possibility prevents the top-level structure from being serial, thus invalidating Equation (5.32). Both of these possibilities require investigation. To investigate the first of these possibilities will require the development of the dependency tools described in Sections 3.5 and 7.1.1.1.2.. The second possibility requires investigation of alternative formulations of the RPI and a better quantification of the impact of fault-tolerant schemes on the RPI.

Also required is the evaluation of the impact that the static time assumption made in Section 5.3.2.4 has on the RPI formulation and the inclusion of time dependency into the kinematic reliability formulation so that a specific time of evaluation is not required. This will allow the quantification of system degradation over time as well as wear phenomena.

b. *Sensitivity to Design Parameters and Modular Metrology.* As presented, the RPI is an empirical measure with no direct analytic tie to the design parameters used during the design of the modules. Thus, the RPI cannot directly be used to select the design parameters such that the best RPI is achieved at the module level. However, an empirical approach to this problem is suggested in conjunction with the development of a methodology to measure and assemble metrology models of modular robotic systems. Rather than using analytic relationships, which would necessitate the use of the tools suggested for development in the preceding paragraph, extensive testing and data gathering on modules as they are designed will allow for statistical correlations between the effects of the RPI and the design variables. Not only will the interactions be empirically quantified, but the actual data needed to use the RPI effectively during configuration design will be developed for the actual modules of the modular robotic system. Additionally, the sensitivity of the RPI to the time of evaluation of the hardware and software reliability models should be examined further. Further development of the relationship between the manipulator Jacobian and the variation in the RPI should also be accomplished. Additional case studies are needed to fully validate the RPI framework.

c. *Inclusion of the RPI into a Computer-Aided Design System for Modular Robotic Design.* The final suggestion for the RPI is to incorporate it for use in the design and selection of a modular robotic system by including it as a tool in a Computer-Aided Design system such as was described in Chapter 1. This would allow full utilization of the RPI as a design tool and allow the user to select the system to include reliability and precision specifications as well as other selection criteria.

7.4. Contributions of The Research Effort

This research has reviewed the different tools and techniques used for reliability improvement during design in the different domains that robotic systems are composed of: electronics, mechanical, and software, as well as nuclear and power distribution systems. This review resulted in a clear understanding of what is lacking in the application of robotics industry: the places that reliability improvement tools and techniques should be applied during the design of a robotic system. Also shown was how the various reliability tasks should be utilized during the design of a robotic system. The design flowchart developed provides the robotic system designer with a guide for the application of these tools and techniques (outlined in Chapter 3 and Appendix A) during his design process.

In addressing the second viewpoint of reliability improvement through the application of technology, Chapter 4 presented the generic reliability estimates of the technologies composing industrial robotic systems and developed design application rankings showing the relative importance the technologies have to the reliability of modular robotic systems. On the basis of these rankings and by using a failure rate model of a generic 7-DOF modular manipulator, a set of research recommendations into robotic systems component technologies was developed that will provide improved relia'ility in modular robotic systems and industrial robots in general.

The third questi^r. was how to include reliability into the design of a modular robotic system. This question was answered with the development of a framework for a design criterion called the Reliability Performance Index (RPI) in Chapter 5. This index, which is unique, provides a way to quantify the reliability of a modular robotic system. The RPI is based on a combination of modular hardware and software reliability models with a stochastic formulation of the manipulator kinematics called Kinematic Reliability. This allows a simultaneous measurement of the reliability of the system as well as its precision. In Chapter 6, it was shown that the RPI is of significant utility in the selection of a modular manipulator configuration from a suite of modules.

This report has provided a comprehensive examination of the reliability of modular robotic systems and how that reliability can be improved during design. This chapter provided an overview of the reliability roadmap for modular robotic systems and summarized the recommendations made as a result of this research. These recommendations form the basis of a research plan that will improve the reliability of modular robotic system concepts during the design of the systems as well as preserve the reliability in service.

APPENDIX A: RELIABILITY THEORY AND ANALYSIS TOOLS

This dissertation has discussed reliability and maintainability aspects of robot systems with emphasis on modular robot systems. Continued references to reliability theory have been made without explaining the preliminary theory of reliability or explaining the exact methods and procedures for the reliability analysis of robot systems. This appendix presents an overview of reliability theory and the tools used to perform reliability analysis. This theory and tools are general and can be applied to any system during design and production as discussed in Chapter Three.

A.1. Reliability Theory

A generally accepted definition of reliability is:

> The reliability of a system is the probability that the system will perform it's intended function adequately for a specified time under stated conditions [73].

This definition implies that reliability is statistical in nature that forces the reliability engineer to understand the use of statistics during the design of a system. It also allows the use of the mathematics of probability and statistics to describe the behavior of the system of interest. At this point the reader is referred to any good probability and statistics text for the basic definitions and axioms of probability and statistics. One good reference is Mendenhall, Schaeffer, and Wackerly's book

Probability and Statistics for Engineers [95]. The following presentations assume the reader is familiar with basic statistical principles and procedures and the mathematics necessary to apply them.

A.1.1. The Reliability Function and its Relatives

Given a cumulative distribution function of time to failure $F(t)$, which is the probability that the system will fail by time t, the reliability of the system can be expressed as

$$R(t) = 1 - F(t) = P(\mathbf{T} > t) \tag{A.1}$$

where **T** is a random variable denoting the failure time, $F(t) = P(\mathbf{T} \leq t)$, and $R(t)$ is the reliability function. At time zero, the reliability of any item is defined to be unity, at infinite life, the reliability is zero. In terms of the density function of the time to failure, $f(t)$,

$$R(t) = 1 - \int_0^t f(\tau)d\tau = \int_t^\infty f(\tau)d\tau \tag{A.2}$$

Reliability also has a frequency interpretation. If we have a certain population of items N undergoing testing and a certain number of the population n has passed the test, we can represent the reliability of the population as

$$R = \frac{n}{N} \tag{A.3}$$

This definition of reliability leads directly to another measure of reliability known as the hazard function [73]. The derivation of the hazard function can be found in [73] and is related to the reliability function by

$$h(t)=\frac{f(t)}{R(t)} \tag{A.4}$$

The hazard function is the instantaneous failure rate of an item and is extremely useful during reliability testing. We now have defined four related reliability characteristics: the cumulative distribution function, $F(t)$, the probability density function, $f(t)$, the reliability function, $R(t)$, and the hazard function, $h(t)$. The relationships between these four functions are shown in Table A.1.

Table A.1. Relationships between Reliability Characteristics [107]

	In terms of			
Characteristic	$F(t)$	$R(t)$	$h(t)$	$f(t)$
$F(t)$	•	$1-R(t)$	$1-e^{-\int_0^t h(\tau)d\tau}$	$\int_0^t f(\tau)d\tau$
$R(t)$	$1-F(t)$	•	$e^{-\int_0^t h(\tau)d\tau}$	$\int_t^\infty f(\tau)d\tau$
$h(t)$	$\frac{dF(t)/dt}{1-F(t)}$	$-\frac{dR(t)/dt}{R(t)}$	•	$\frac{f(t)}{\int_t^\infty f(\tau)d\tau}$
$f(t)$	$\frac{dF(t)}{dt}$	$-\frac{dR(t)}{dt}$	$h(t)e^{-\int_0^t h(\tau)d\tau}$	•

A.1.2. Probability Distributions Used in Reliability Theory

As described in the previous section, the reliability function is a function of time (or any units used to describe the life of an item) and is determined from the cumulative distribution function of the life of the item. It is these distribution functions that determine the probabilistic characteristics of the item's life and allow us to model the probabilities involved during the life of the component. Since reliability theory is based on the manipulation of these probability distributions of lifetimes, we need to discuss and explain the various types of distributions used and their applications.

A.1.2.1. The Exponential Distribution [73]

The probability distribution for an exponentially distributed random variable t is

$$f(t;\lambda) = \lambda e^{-\lambda t}, \quad t \geq 0 \tag{A.5}$$

where t is the random time between failures and λ is the failure rate of the item. Using the conversion in Table A.1, the reliability function is

$$R(t) = e^{-\lambda t}, \quad t \geq 0 \tag{A.6}$$

The inverse of the failure rate is defined as the Mean-Time-Between-Failure (MTBF), $\theta = 1/\lambda$. The MTBF is usually the parameter specified during the design of components and systems. The hazard function for the exponential distribution is a constant with the value of λ. The exponential density function is shown in Figure A.1. The exponential distribution is the only continuous distribution exhibiting a constant failure rate. This characteristic of the exponential distribution has many important implications. The first is that the failures of exponentially distributed items will be completely random with no dependence upon where the item is in its life cycle. Another way of stating this is that the current state of the item does not depend on its past and it is as likely to fail now as any other time. This property is referred to as the "Memoryless" or "Markovian" Property. This property will be more fully addressed in Section A.2.1.2.

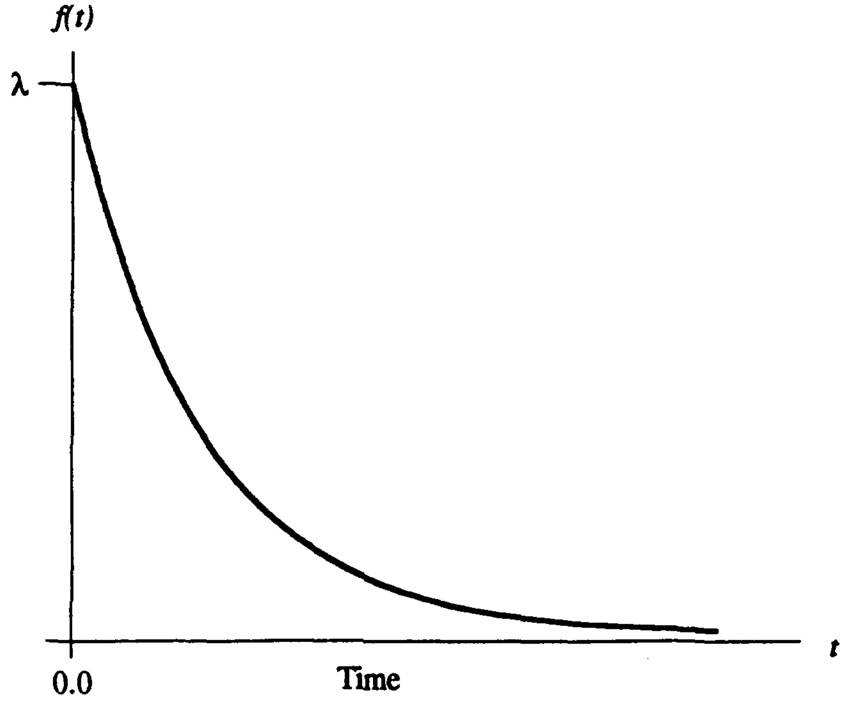


Figure A.1. Exponential Probability Density Function

The exponential distribution is exceedingly tractable mathematically and the majority of reliability theory is based upon the manipulations of this distribution. The reason for this is that electronic components and systems tend to follow the exponential distribution in the field and many other component and systems can be justified to be exponential if the failure pattern is random. In addition, large systems consisting of many components tend to exhibit a constant failure rate at the system level, especially those with no preventative maintenance [73]. This enables the use of the exponential distribution for life test design. Information regarding statistical characteristics, life testing, and data analysis for items and systems following the exponential life distribution can be found in Kapur and Lamberson [73], MIL-HDBK-217 [96] and MIL-HDBK-781 [100].

A.1.2.2. The Weibull Distribution [3, 73, 107]

After the exponential distribution, the most used distribution in reliability theory is the Weibull distribution. This distribution was first applied to the yield strength of steels in 1951 and subsequently has been used for many other applications, particularly in the aerospace world. This distribution is very popular since it can assume a wide variety of shapes and positions (See Figure A.2). The Weibull distribution can be expressed with two or three parameters, depending on if a *guaranteed* life is present. The cumulative distribution for a three parameter Weibull is

$$F(t;\theta,\beta,\delta) = 1 - e^{-\left(\frac{t-\delta}{\theta-\delta}\right)^{\beta}}, \quad t \geq \delta, \beta > 0, \theta > 0, \delta \geq 0 \qquad \text{(A.7)}$$

where δ is the guaranteed life, θ is the scale parameter (also known as the characteristic life), and β is the shape parameter of the distribution. If the guaranteed life is zero, we obtain the two parameter Weibull which is the most widely used. The c.d.f. of the two parameter Weibull is

$$F(t;\theta,\beta) = 1 - e^{-(t/\theta)^{\beta}}, \quad t \geq 0, \beta > 0, \theta > 0 \qquad \text{(A.8)}$$

The p.d.f. is

$$f(t;\theta,\beta) = \frac{\beta}{\theta}\left(\frac{t}{\theta}\right)^{\beta-1} e^{-(t/\theta)^{\beta}}, \quad t \geq 0 \qquad \text{(A.9)}$$

The hazard function is given by

$$h(t)=\frac{\beta}{\theta}\left(\frac{t}{\theta}\right)^{\beta-1}, \quad t\geq 0 \tag{A.10}$$

As can be seen from the hazard function, the Weibull distribution has a non-constant failure rate; decreasing in time if $\beta < 1$, increasing if $\beta > 1$, and is constant if $\beta = 1$. If the failure rate is constant ($\beta = 1$), the Weibull reduces to the exponential distribution. The Weibull can also represent several other distributions with the appropriate choices in parameters. The non-constant failure rate makes the Weibull distribution the distribution of choice when modeling non-constant failure rate components and finds the most application in mechanical reliability estimation. The shape of the distribution is highly dependent upon the value of β (see Figure A.2). The parameter θ is called the characteristic life since the probability of failure prior to θ is 0.632 for any Weibull distribution with any β.

The Weibull is not as mathematically attractive as the exponential distribution because the moments of the Weibull distribution contain the gamma function. This causes any estimators of the Weibull parameters to be dependent upon each other requiring iterative, numerical solutions. One attractive feature of Weibull analysis is the graphical interpretation which can be developed. The logarithm of both sides of Equation (A.8) can be taken resulting in the equation of a straight line. This allows the logarithm of Weibull data to be plotted and approximate a straight line which can be used to estimate the parameters. This procedure is clearly defined in [73] and [107].

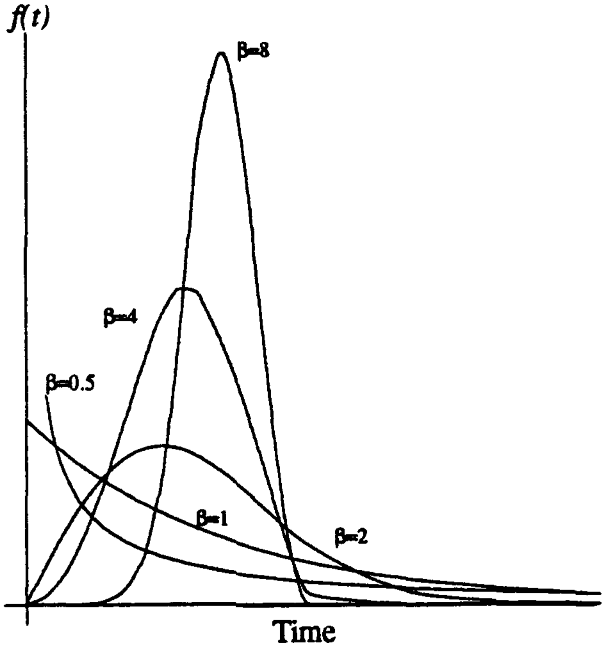


Figure A.2. Weibull Distribution for Various Values of β.

A.1.2.3. The Extreme Value Distributions [73, 95]

When working with failure of mechanical systems and components (including the mechanical aspects of electronics (wire and die bonds, etc.)) we usually see a failure of the weakest link of a serial chain. That is, the failure of materials or components are usually related to the weakest point or component in the system. Thus, the reliability analyst needs to be familiar with the distributions of the extreme values of the life distributions of the components and materials. This is the distribution of the random variable described by

$$y_n = \min(x_1, x_2, \ldots, x_n) \tag{A.11}$$

where x_i, $i = 1,...,n$ represent samples from a infinite population having a cumulative distribution function $F(x)$. The random variable y_n is the smallest extreme value and its distribution is given by

$$g_n(y) = nf(y)[1-F(y)]^{n-1}, \quad -\infty < y < \infty \tag{A.12}$$

The extreme value distribution is also known as the p.d.f. of the first order statistic in a sample of size *n*. This p.d.f. results from the examination of ordered samples from a population. As can be seen from Equation (A.12) the form of the smallest extreme value distribution is dependent upon the parent distribution from which the population is drawn. The hazard function associated with the extreme value distribution will be of an exponential form [73]. In the exponential parent case, the smallest extreme value distribution is

$$g_n(y) = n\lambda e^{-n\lambda y}, \quad y \geq 0 \tag{A.13}$$

which can be identified as an exponential distribution with a failure rate of $n\lambda$. All of the relationships discussed in Section A.1.1 apply to the extreme value distribution.

A.1.2.4. The Normal and Log-Normal Distributions [73, 95]

The normal or Gaussian distribution is probably the best known of all the probability distributions. This is the well known "bell shape" curve (Figure A.3). The normal distribution has two parameters: the mean, μ, and the standard deviation, σ. The standard normal has mean $\mu = 0$ and standard deviation $\sigma = 1$ (denoted as $N(0,1)$). The p.d.f. of the normal distribution is given as

$$f(x;\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}, \quad -\infty \le x \le \infty \tag{A.14}$$

The c.d.f. for the normal distribution cannot be evaluated in closed form, but values for the standard normal density function and cumulative density function can be used to find the probabilities for any normal distribution through the transformation

$$Z = \frac{x-\mu}{\sigma} \tag{A.15}$$

where $Z \sim N(0,1)$. Tables for the standard normal can be found in any statistical or reliability text such as [73, 95].

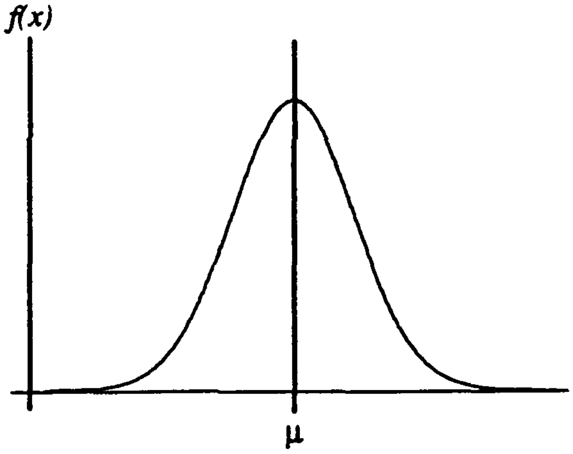


Figure A.3. Normal Density Function.

The hazard function for the normal distribution is a monotonically increasing function which means that for components that follow the normal distribution, the failure rate increases as the component grows older. A normal distribution may not be a good choice as a life distribution since the tails of the distribution extend to infinity. A better choice for a component which has an increasing failure rate might be the Weibull since it is always positive. The normal is generally used for modeling phenomena that has a central value and width to its probability distribution. Good applications include the strength of materials, location and position data, and error data. The normal is also useful in data analysis due to the Central Limit Theorem. This limit theorem states that the distribution of samples taken from a population of arbitrary distribution will converge to a normal distribution as the number of samples taken increase. An additional statement that can be made is that a random variable made up of linear combinations of other non-normal random variables will tend to the normal. While not true for non-linear combinations, such as robot kinematic equations, non-linear combinations will tend to the normal if the non-linearities are not severe [17].

A related distribution is known as the log-normal distribution. The p.d.f. of the log-normal distribution is similar to the normal except the natural logarithm of the random variable is normal. The p.d.f. is given as

$$f(t;\mu,\sigma)=\frac{1}{\sigma t\sqrt{2\pi}}e^{-\frac{1}{2}\left(\frac{\ln t-\mu}{\sigma}\right)^2}, \quad t\geq 0,-\infty\leq\mu\leq\infty,\sigma>0 \qquad \text{(A.16)}$$

The log-normal distribution is readily manipulated by transforming the data such that **x** = ln **t** will be normally distributed. The inverse transform is simply the exponentiation of **x**: **t** = e$^{\mathbf{x}}$. The log-normal distribution is most useful in modeling repair times since the distribution is skewed towards the beginning of the time when repair is begun. This turns out to be true for most normal repair scenarios and the log-normal is a good representation of this effect (see Figure A.4).

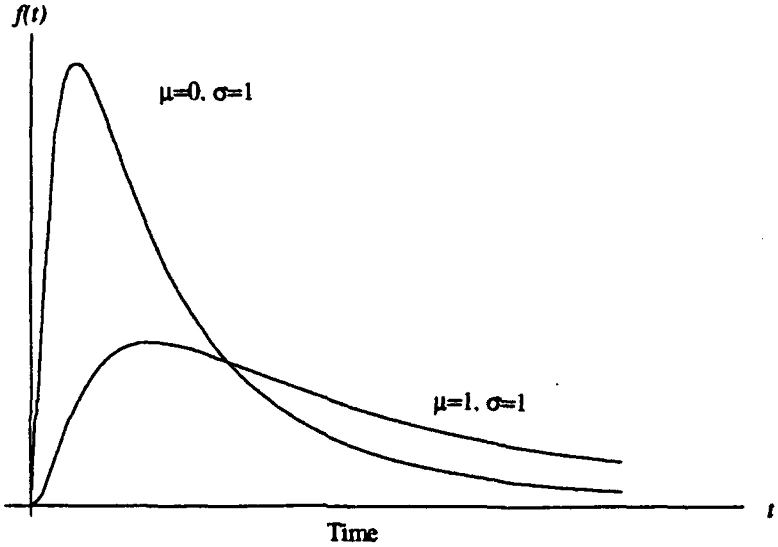


Figure A.4. Log-Normal Density Function

A.1.2.5. Distribution Selection [73, 107]

Selecting an appropriate statistical model to represent a design is one of the most difficult tasks facing the reliability analyst. There are generally two circumstances which will apply: data exists for the design or similar designs, or a new design is bring developed and no data exists.

A.1.2.5.1. Selecting the Distributions for Existing Data

Depending upon the type of data one has available, several steps must be taken to insure that the information contained in the data will be represented by the model chosen. The first step is to determine the possible distributions the data can represent. One good method is to divide the data into time intervals and plot the frequency of occurrences that occur within each interval. This is called a frequency histogram (Figure A.5).

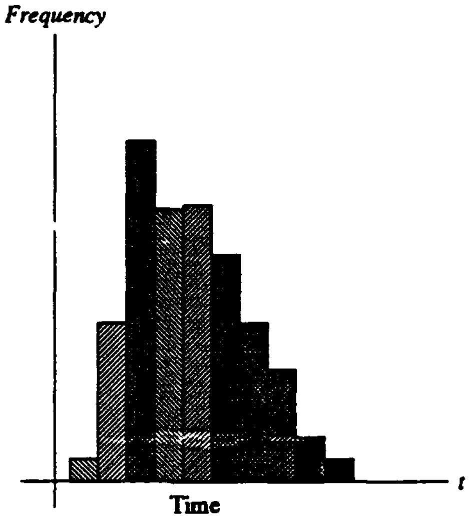


Figure A.5. Frequency Histogram

For instance, the histogram of Figure A.5 *might* represent a Weibull, normal, or log-normal distribution but probably not an exponential. If the first two columns were missing, then the exponential becomes more likely. This methodology can also be applied to the hazard (the failure rate) and the reliability functions over time and matching the general shapes of the curves to known probability models.

Another way of identifying the proper distribution is through the use of probability graphing [107]. This method plots the failure data on paper representing the p.d.f. for which we wish to check. This method will also identify the distribution parameters which can then be tested for adequacy.

Once a possible distribution is identified, "Goodness-of-Fit" testing is used to make a statistical determination if the distribution is a possible parent of the data. We now enter the realm of statistical inference. The following discussion is intended only as an overview of hypothesis testing. For a in-depth explanation of the procedures and assumptions involved, any statistical text can be consulted.

In a statistical test, one formulates a "null hypothesis," denoted as H_o, which is generally a condition being tested. A typical null hypothesis might be that the mean of a data set is a certain value, i.e., H_o: $\mu = \mu_o$. A companion to the null hypothesis is the "alternate hypothesis," denoted as H_a. In the current case, the alternate hypothesis could be $\mu \neq \mu_o$, $\mu \leq \mu_o$, or $\mu \geq \mu_o$. If the "not equal to" option is chosen, the test is called a "two-sided" test which changes the selection of the test statistic, otherwise the test will be one-sided. The null hypothesis can only be rejected. It cannot be certainly determined, from a statistical point of view, that the null hypothesis is actually true. Some error will always exist. For a discussion of testing errors and the power of tests, see [95].

The most common goodness-of-fit test is the Chi-Squared goodness of fit test. This test statistic is given by

$$\chi^2 = \frac{\sum_{i=1}^{k}(f_{oi} - f_{ei})^2}{f_{ei}} \tag{A.17}$$

where k = Number of classes

f_{oi} = Observed frequency of the ith class

f_{ei} = Expected frequency of the ith class (should be greater than five).

The test statistic of equation (A.17) is distributed as a Chi-Squared distribution with degrees of freedom $\gamma = k - 1 - m$ where m is the number of parameters such as means and variances estimated from the data to generate the expected frequencies. A level of significance is selected, and a corresponding value is retrieved from a table and compared to the test statistic. If the test statistic is lower than the table value, the null hypothesis that the data is from a particular distribution cannot be rejected and you proceed with the use of the distribution. A word of caution: a goodness-of-fit test only tells you that the data *may* be from a certain distribution. More than one distribution may satisfy the goodness-of-fit test. This requires that the data must be used very carefully and conservative estimates made during the design. Other goodness-of-fit tests that are more powerful, such as the Kolmorgorov-Smirnov test, can be used to obtain more confidence in the fit of the failure model to the data.

Once the designer is satisfied that the proper mathematical model has been identified, the design and/or analysis can proceed.

A.1.2.5.2. Selecting the Distributions When No Data Exists

A more difficult problem exists when a new design is being developed and no testing has yet been accomplished. How are useful models of failure obtained for the design to meet the reliability specification? The solution to this problem lies in the identification of the failure modes of the components themselves and how they manifest themselves. The most valuable input to this problem is a Failure Mode, Effects, and Criticality Analysis (FMECA). This analysis, described in section A.2.2 provides anticipated failure modes of the components of the equipment. From these failure modes and based on past history, assumptions can be made about the failure distributions for the device. For instance, if only random failures will exist for a component, it is a good assumption that it will follow an exponential failure distribution. The failure rate is unknown but can be estimated from data from a similar design. Do not ignore data obtained from other designs. However, one must take into account the differences between the new design and the old such as environment, operating states, etc.

A.1.3. Measures of Reliability

Given the p.d.f. or reliability function of a system or component, the expected life of the system can be represented as

$$E(t) = \int_0^t \tau f(\tau) d\tau = \int_t^\infty R(\tau) d\tau \qquad \text{(A.18)}$$

This is also known as the Mean-Time-To-Failure (MTTR) or the Mean-Time-Between-Failure (MTBF) if using the exponential distribution. This is the number that is usually used for the specification of designs. Caution must be used, however, in what MTTF or MTBF really means. For instance, the reliability for the exponential distribution at the MTBF is

$$R(t = \theta) = \mathrm{e}^{-1} = 0.368 \qquad \text{(A.19)}$$

The normal distribution is symmetric about its mean so at its MTTF the reliability is

$$R(t = \mu) = P(\mathbf{Z} \geq 0) = 0.5 \qquad \text{(A.20)}$$

As you can see, the actual reliability realized at the MTTF (or MTBF) 'vill be different depending upon what the underlying life distribution really is. This can make a great deal of difference to the customer and must be taken into account during the generation of design specifications.

An alternative measure of reliability frequently used is the failure rate. Recall that the hazard function (Equation (A.4)) is an expression of instantaneous failure rate. This can also be used to measure the reliability of a component or system.

The hazard rate falls prey to the same problem that MTTF or MTBF has: it directly corresponds to a failure distribution. The hazard function is a powerful tool in the selection of distributions to fit a particular data set. Plots of failures over time can be examined for tendencies to follow one of the hazard models described in Section A.1.2. Another characteristic of hazard functions can be seen in Figure A.6. This is a plot of a typical hazard function followed by many components throughout their lives. There typically is a initial phase of higher failure rate known as the "infant mortality" phase. Failures that occur during this phase usually are due to material or manufacturing defects which occur upon fielding the item. These failures can usually be removed by an effective screening program incorporated into the production process. The second phase of the bathtub curve is one of constant failure rate. This period is the "in-service" period where failures occur due to random extreme stresses and events encountered during normal service. This phase may not actually be constant and depends upon the underlying failure distribution, however, when compared to the "infant mortality" and wear-out phases, shows a much smaller rate of change. The last phase of the bathtub curve is the wear-out phase. This occurs at the end of the life cycle when the components of a system have degraded over time and are failing under normal stresses. Mechanical components under stress (friction or other sources) will exhibit this phenomenon towards the end of their useful lives. This is the area of most interest when examining the reliability of mechanical systems since if the time at which this period begins can be predicted, the item could be replaced before system degradation occurs. (This is known as Reliability Centered Maintenance (RCM) [7]).

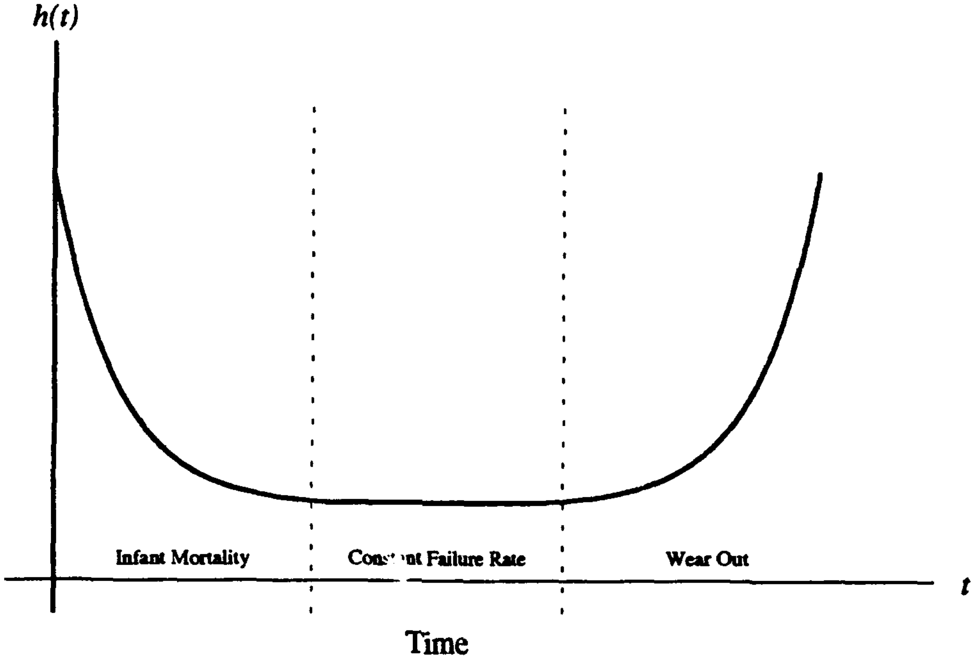


Figure A.6. The "Bathtub Curve" Hazard Function

A.1.4. The Algebra of Random Variables [73, 95, 147]

So far, we have discussed distributions and how they are selected to model the reliability of single items or devices. However, most reliability questions arise at the system level. These systems are composed of many components all of which have their own failure distributions. The question now becomes one of determining the failure distribution of the system when it is made up of these many individual components. Mathematically, the problem can be described as determining the distribution of a given function of random variables. This question is resolved using the algebra of random variables.

Most distributions are described using the expectation and variance of the distributions. The expectation, or first moment, is given by Equation (A.17). This quantity is known as the mean. The variance, or second moment about the mean, is defined as

$$V(\mathrm{X}) = E[(\mathrm{X}-\mu)^2] \tag{A.21}$$

There are many well known properties of variance and expectation. If we have two random variable x and y with expectations E[x] and E[y], and variances V[x] and V[y], respectively, then

$$\begin{gathered} E[a\mathrm{x}] = aE[\mathrm{x}] \\ E(a+\mathrm{x}) = a+E[\mathrm{x}] \\ E[\mathrm{x}\pm\mathrm{y}] = E[\mathrm{x}]\pm E[\mathrm{y}] \\ E[\mathrm{x}^2] = [E[\mathrm{x}]]^2 + V[\mathrm{x}] \\ E[g(\mathrm{x})] = \int_{-\infty}^{\infty} g(x)f(x)dx \end{gathered} \tag{A.22a-e}$$

where *g(x)* is a function of the random variable x, *f(x)* is the p.d.f., and *a* is a constant. Also

$$\begin{gathered} V[a\mathrm{x}] = a^2V[\mathrm{x}] \\ V[a+\mathrm{x}] = V[\mathrm{x}] \\ V[\mathrm{x}^2] = E[\mathrm{x}^4] - (\mu_{\mathrm{x}}^2+\sigma_{\mathrm{x}}^2)^2 \end{gathered} \tag{A.23a-c}$$

if x and y are independent, that is, if the covariance defined to be

$$Cov[\mathrm{xy}] = E[(\mathrm{x}-\mu_{\mathrm{x}})(\mathrm{y}-\mu_{\mathrm{y}})] = E[\mathrm{xy}] - E[\mathrm{x}]E[\mathrm{y}] \tag{A.24}$$

is zero, then

$$E[\mathrm{xy}] = E[\mathrm{x}]E[\mathrm{y}] \tag{A.25}$$

and

$$\begin{gathered} V[\mathrm{x}\pm\mathrm{y}] = V[\mathrm{x}] + V[\mathrm{y}] \\ V[\mathrm{xy}] = \sigma_{\mathrm{x}}^2\sigma_{\mathrm{y}}^2 + \sigma_{\mathrm{y}}^2\mu_{\mathrm{x}}^2 + \sigma_{\mathrm{x}}^2\mu_{\mathrm{y}}^2 \end{gathered} \tag{A.26a-b}$$

While the mean and variance of the function of random variables are very helpful in getting an idea of what the limits of the distribution will be, it is sometimes necessary to actually determine the density function of the function of random variables. One of the most used methods of finding the density function is the *Method of Transformations*. Given an increasing or decreasing function

$$\mathbf{y} = g(\mathbf{x}) \tag{A.27}$$

where x has the p.d.f. $f_X(x)$ then the p.d.f. of y, $f_Y(y)$ can be found by

$$f_Y(y) = f_X(x)\left|\frac{dx}{dy}\right| \quad \text{where} \quad x = g^{-1}(y) \tag{A.28}$$

This method is readily generalized to more than one random variable as shown in [95].

A substantially more powerful way of treating the combination of random variable problem is through the use of integral transforms. By transforming the density functions involved using Laplace or Mellin transforms, we can reduce the convolution problem presented by the sum, difference, multiplication, or division of random variables to simple algebraic operations on the transforms, much as is done when solving differential equations using Laplace transforms. A good reference is [147]

A.2. Reliability Analysis Tools

Up to this point, we have discussed an overview of basic reliability theory as it applies to one device or equipment item. However, most specifications are written at a system level, not at the component level. The question now becomes one of modeling the reliability of a system made out of components of which we know something about. The second part of this appendix is devoted to explaining the system models and tools used in the reliability analysis of systems during their design.

A.2.1. Reliability Modeling Techniques

To adequately address the reliability design constraints during the design process, a system level model of the reliability behavior of the system needs to be developed . Many different system level models have been proposed and used in the past, most with adequate results. The choice of the model depends greatly upon the uses one expects to have for the model. The amount of available data also has a large impact. For instance, most preliminary design reliability analyses start with a simple parts count reliability prediction. This system reliability model (described in Section 3.1.3.2) is based strictly upon the failure rates of the components added in series. The underlying assumptions of this approach are exponential distributions governing all components and a serial reliability structure (no redundancy). This section identifies these system models and provides insight into their uses and solutions and gives references where further detailed information can be obtained.

A.2.1.1. Combinatorial Models

The most common type of system level model is a combinatorial model. These modeis use combinations of probability events or reliability functions.

A.2.1.1.1. Reliability Block Diagrams [73]

The simplest reliability structure is a serial one. The serial structure implies that a failure of one component in the system causes a failure of the system. This structure can be represented in block diagram form as shown in Figure A.7.

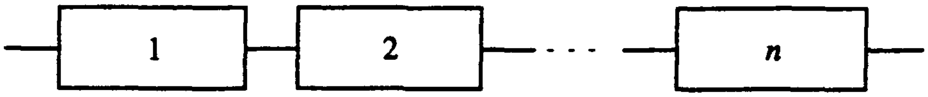


Figure A.7. Serial System Reliability Block Diagram

The usual assumption is that the components represented in the block diagram are independent. As one can see, the structure of the block diagram will depend upon the definition of reliability and failure for the system. This model is often called "black box" reliability modeling since the usual initial effort assumes constant (or static) reliabilities for each component represented. The blocks are arranged to determine operational success, not functionality. The static model is usually the first model constructed during design. It is used to help determine possible configurations for the design and allows for allocation of the reliability specification as described in Chapter 3. The following notation will be used for the rest of this section:

E_i = The event that subsystem or component i operates successfully
$R_i = P(E_i)$ = Subsystem or component reliability
R_S = System Reliability

For the series model of Figure A.7, we can represent the system reliability as

$$R_S = P[E_1 \cap E_2 \cap \cdots \cap E_n] = P(E_1)P(E_2)\cdots P(E_n)$$
$$\text{or} \tag{A.29}$$
$$R_S = \prod_{i=1}^{n} R_i$$

Unfortunately, this is usually the worst configuration possible for reliability since the reliability decreases rapidly as the number of series components increases because the series system reliability will always be less than or equal to the least reliable component in the system.

An alternate configuration is expressed in Figure A.8. This is called the parallel reliability block diagram (RBD).

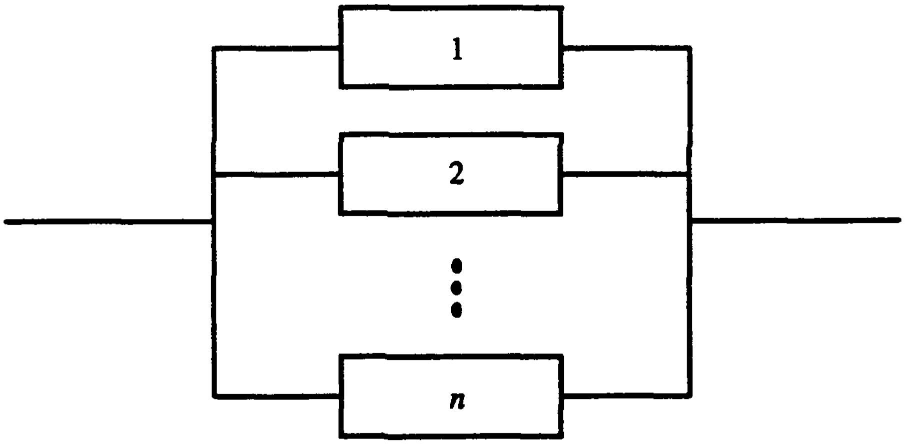


Figure A.8. Parallel System Reliability Block Diagram

The parallel structure assumes that system failure will not occur until all the components have failed. We can calculate the system reliability by defining the system unreliability as Q_S. We can now write

$$Q_S = P(\overline{E}_1)P(\overline{E}_2)\cdots P(\overline{E}_n) = \prod_{i=1}^{n}(1-R_i) \tag{A.30}$$

where $\overline{E}_i$ is the complementary event. The parallel system reliability can now be written as

$$R_S = 1 - \prod_{i=1}^{n} (1 - R_i) \tag{A.31}$$

This analysis assumes all components are active in the network and the components are independent. This arrangement is usually not representative of many redundant structures. Most components, especially mechanical components, will share the load and will need to bear the full load when failure occurs in the other members. This situation is not a static model and will be considered momentarily.

Another form of redundancy is known as a *k out of n* system. This system has n parallel components but requires at least k components to survive for the system to operate. This system can be modeled using the binomial formula as

$$R_S = \sum_{x=k}^{n} \binom{n}{x} R^x (1-R)^{n-x} \tag{A.32}$$

where R is the subsystem reliability and is assumed equal for all the components and

$$\binom{n}{x} = \frac{n!}{x!(n-x)!}$$

Direct event enumeration can be used if the subsystems do not have equal reliabilities (see Section A.2.1.1.2.

Many systems can be represented with local redundancies built into the reliability network resulting in series-parallel networks. These networks can be reduced using the reliability formulas above to obtain a system reliability figure. Many examples of these types of calculations can be found in Chapter 5 of [107].

The static models just presented can easily be extended to dynamic models of the system reliability. Recall from the definition of the reliability function in Equation (A.2) that reliability is a time dependent phenomenon. If the subcomponent reliability functions are represented as $R_i(t)$ and are independent, the series dynamic model can be written as

$$R_S(t) = \prod_{i=1}^{n} R_i(t) \tag{A.33}$$

by taking the logarithm of (A.33) with the proper substitutions [73], the expression for the series hazard function is simply

$$h_S(t) = \sum_{i=1}^{n} h_i(t) \tag{A.34}$$

Equations (A.33) and (A.34) are valid regardless of the underlying p.d.f.'s.

A similar situation can be derived for the pure parallel system with

$$R_S(t) = 1 - \prod_{i=1}^{n} [1 - R_i(t)] \tag{A.35}$$

However, as stated before, the pure parallel system is not the usual case. The usual case will be a standby redundant system with either perfect or imperfect switching or a shared load system. In either case, the preferred method of analysis is the development of a Success Mode Diagram (Figure A.9). Representing the event that the ith subsystem or component is operating as E_i and t_i as the random variable of the ith subsystem's life with p.d.f. $f_i(t_i)$ we can write the reliability of a two-unit standby system as

$$R_S(t)_{2-unit} = R_1(t) + \int_0^t f_1(t_1) R_2(t - t_1) dt_1 \qquad \text{(A.35)}$$

Where t_1 is the time of occurrence of the failure. If more units are added, we just have to add the probability of the additional success modes occurring.

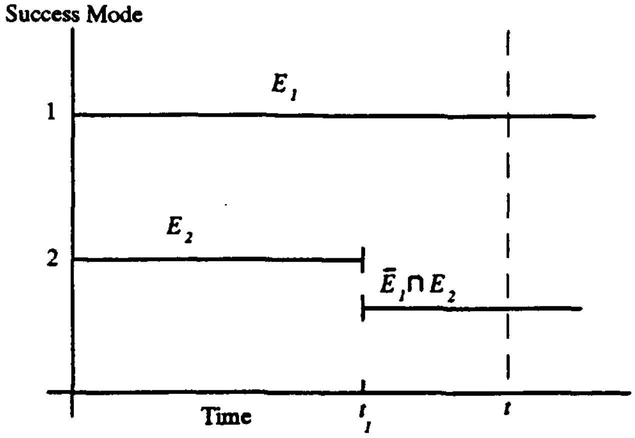


Figure A.9. Success Mode Diagram for a Two-Unit Standby System (Perfect Switching)

For the case of imperfect switching, we add the reliability of a switch to (A.35) and obtain

$$R_S(t)_{2-unit} = R_1(t) + \int_0^t f_1(t_1) R_{Switch}(t_1) R_2(t - t_1) dt_1 \qquad \text{(A.36)}$$

where $R_{Switch}(t)$ is the reliability function for the switch. Note we have just added the probability of the switch lasting to the time of switching to the integral.

Shared load models can be analyzed in much the same way. Take for instance a pair of bolts holding a plate on a machine. If one bolt breaks, the other takes the load. This approach requires knowledge of the time to failure densities under half and full loads. Let $h(t)$ represent the p.d.f. for time to failure under half load and $f(t)$ represent the p.d.f. for time to failure under full load. The first success mode is the probability that both bolts survive:

$$P[\mathbf{t}_1 > t \cap \mathbf{t}_2 > t] = \int_t^\infty h(\tau)d\tau = [R_h(t)]^2$$

The second and third success modes can be expressed as the integral

$$\int_0^t h(t_1)R_h(t_1)R_f(t-t_1)dt_1$$

Adding the probabilities of exclusive events together give the overall reliability as

$$R_S(t) = [R_h(t)]^2 + 2\int_0^t h(t_1)R_h(t_1)R_f(t-t_1)dt_1 \qquad \text{(A.37)}$$

This integral expression can easily be expanded to include additional components. The integration is a problem but with modern numerical methods, the reliability is readily obtainable if the p.d.f.'s are known.

Many other dynamic models are possible. The most utilized is the Markov process which is applicable if the failure rates are constant. Markov Models are described in Section A.2.1.2.

A.2.1.1.2. Event Combinations

Problems can arise in the calculation of system reliability from the RBD model. Consider the RBD of Figure A.10. This network cannot be reduced using the simple formulas of the previous section.

One method of solution of this model is called the decomposition or "Keystone" method. This method uses the rules of conditional probability [95] to develop and expression for the system reliability. If we assume that component E has failed, the path through component E opens and the network becomes a parallel network of two serial networks. The reliability for this event is

$$R_{E\,Failed} = \{1-[1-(R_A R_C)][1-(R_B R_D)]\}\overline{R}_E \qquad \text{(A.38)}$$

where $\overline{R}_E$ is the probability that component E has failed. If E is working then components A and B do not contribute to the network and the network reduces to component E in series with components C and D in parallel. The reliability for this event is

$$R_{E\,Operating} = [1-(1-R_C)(1-R_D)]R_E \tag{A.39}$$

The system reliability then becomes simply

$$R_S = R_{E\,Operating} + R_{E\,Failed} \tag{A.40}$$

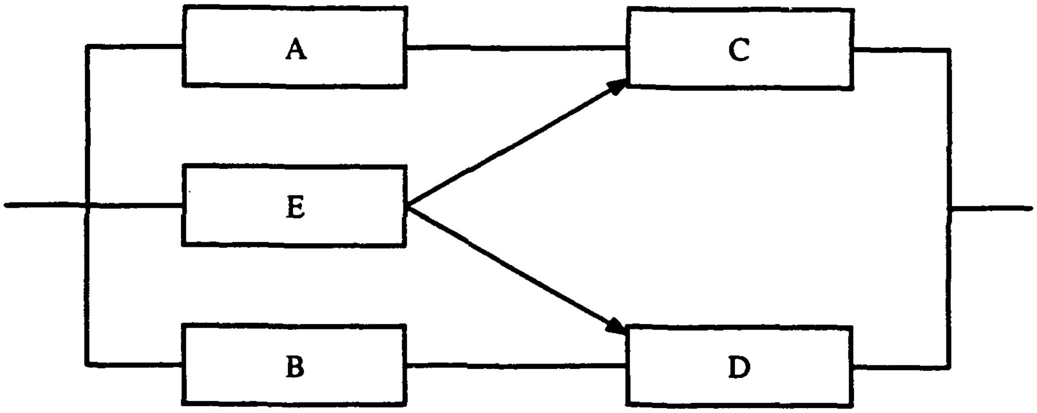


Figure A.10. Complex System Reliability Block Diagram

Equivalent statements of the reliability of the network of Figure A.10 can be developed through a method called Event Decomposition [73, 107]. This procedure is well suited for implementation on a computer and most commercial reliability programs use this method if they use a combinatorial system model. The method is simply to list all the possible combinations of failed and operating states of all the components and calculate the probability of that state occurring. The reliability is just the sum of the probabilities of all the combinations that have a successful outcome. The network of Figure A.10 has 19 combinations resulting in success out of a possible 32 [73].

The method of minimal cut sets is another algorithm for determining the success modes of networks such as Figure A.10. This method determines the probabilities of a critical path being interrupted and calculates sets of probabilities similar to the event decomposition method. The advantage in using cut sets is that many well known algorithms exist in network theory to find these probabilities and it may be more efficient for large, multi-component systems to be analyzed using this method. This method is unsuitable for systems with dependent components or subsystems. The details of this method are explained in [107].

A.2.1.1.1. Fault Tree Analysis [68]

Fault tree analysis (FTA) is a method of providing for a top-down system reliability and safety analysis. A fault tree is a model that graphically and logically represents the various combinations of events. These combinations occur in the system and lead to the top level events. There are two types of top level events. The first is a normal function designed into the system. The other type of event is a fault, or abnormal system state. The fault tree is a Boolean representation of causal relationships between faults and top level events and results in the same reliability expression as the analysis of reliability block diagrams if the same relationships are used. Unlike a Failure Mode, Effects, and Criticality Analysis (FMECA), a FTA is restricted to only the identification of abnormal functioning of system elements and events that cause a single undesirable top level event. To illustrate, a top level event might be as general as "robot arm does not move" or as specific as "relay A27 energized erroneously."

The FTA provides options for quantitative and qualitative reliability analysis, it helps the designer and reliability analyst to understand system failures and their causes, it heightens awareness of the sensitivity of the system to low level failure events, and it provides insight into the system behavior. The steps for FTA are as follows:

1. Define the top event(s).
2. Establish the boundaries of the system.
3. Understand the system's behavior.
4. Construct a Fault Tree.
5. Analyze the Fault Tree.
6. Take corrective action to correct design deficiencies.

Standard Boolean logic symbols are used to construct the fault tree. Gate symbols connect events to their causal relations. Gates can have deterministic causality such as AND, OR, and XOR, or can have probabilistic causality with conditional inputs. Events are represented by rectangles which encloses the event description. A basic event which is a basic inherent failure of the system or component is represented by a circle. These are referred to as primary failures. Diamonds represent an undeveloped event and appear at the bottom of a fault tree representing its resolution. There are many more symbols available for use in FTA. For an in-depth explanation of their uses, refer to [68].

The fault tree is built from the top down. The top level events are identified and the tree is structured downward becoming more and more specific as events are decomposed into their basic causes. A primary failure (a basic event) is due to the internal characteristics of the component under consideration. That is, it is an inherent failure mode of the device. The events in the tree are decomposed until only basic failures remain or the level of desired resolution is reached.

An example where FTA has been applied to a robot system can be found in [77]. The robot analyzed in this paper had a block diagram shown in Figure A.11. The system was a standard electric industrial robot. The top level event in this fault tree was "Undesirable Robot Movement." Boundaries where established as the robot is performing a "normal" industrial task and is programmed and maintained by people. The system's behavior follows the block diagram of Figure A.11 and the fault tree is developed accordingly.

The fault tree for the top event "Undesirable Robot Motion" is shown in Figure A.12a. Undesirable robot movement could be the result of either a failure within the robot causing the movement (or lack thereof) or it could be from external causes. The analyst identifies five internal failure events within the robot and two externally. Note the symbols used in this analysis. The events towards the top are generally connected by OR gates representing a serial reliability structure. The parallel structure is represented by AND gates with multiple combinations (such as *k out of n*) implemented with combinations of these Boolean functions. Figure A.12b is a expansion of the joint control failures branch of Figure A.12a. Figure A.12c expands a general joint control failure to the events occurring inside the joint itself. The hexagonal gate in Figure A.12c represents a conditional event depending on an input.

As one can see, a FTA can get extremely large very quickly. The main advantage to the FTA is you can break the analysis problem down into manageable pieces that can be individually addressed and quantified. As mentioned before, there is a direct correspondence from the static reliability block diagram to the fault tree. The fault tree is more powerful, however, since many dependencies can be introduced, as seen in Figure A.12c.

Many computer programs are available to help build and solve fault trees and perform FTA's. For a survey of the reliability assessment programs available and how to obtain them, see Reference [69].

A.2.1.2. Markov Models and Availability [133]

A.2.1.2.1. Discrete Markov Chains

An alternate dynamic reliability model is called the Markov model. The basis of this model is a stochastic process called a Markov Chain. A *discrete* Markov chain is a stochastic process that can take on a finite number of possible states and is denoted as $\{X_n, n = 0,1,2,\ldots\}$. When the process is in a certain state X_i, there is a fixed probability, P_{ij}, that the next state of the process will be state X_j. A property of Markov chains is the *Markovian Property* which states that the probability of the process being in a future state is only dependent upon the current state and is independent of the past states the process has been in. Mathematically, we can say

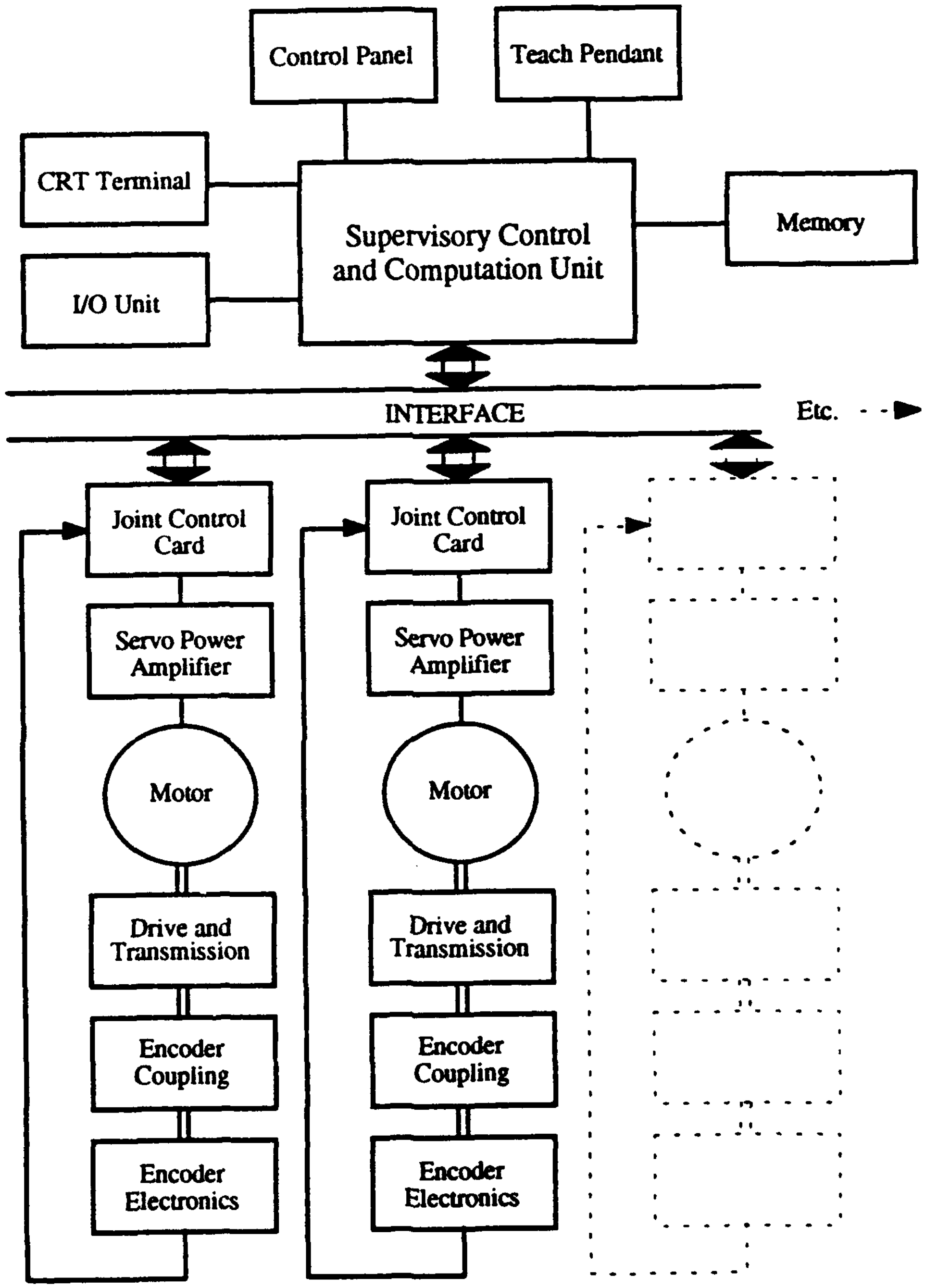


Figure A.11. Block Diagram of a Multi-Jointed Robot [77]

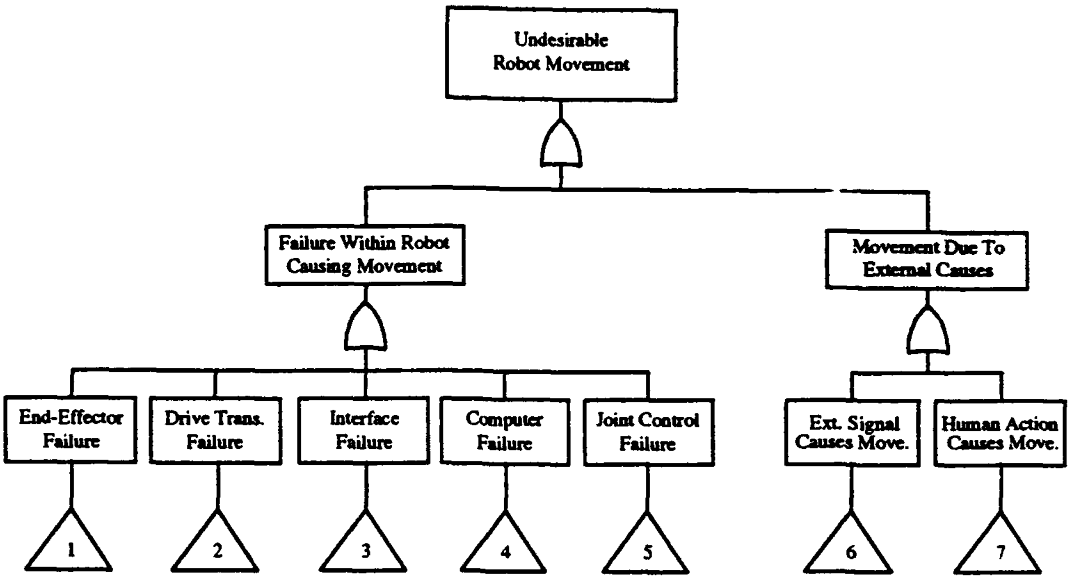


Figure A.12a. Top Level Fault Tree for Electric Robot [77]

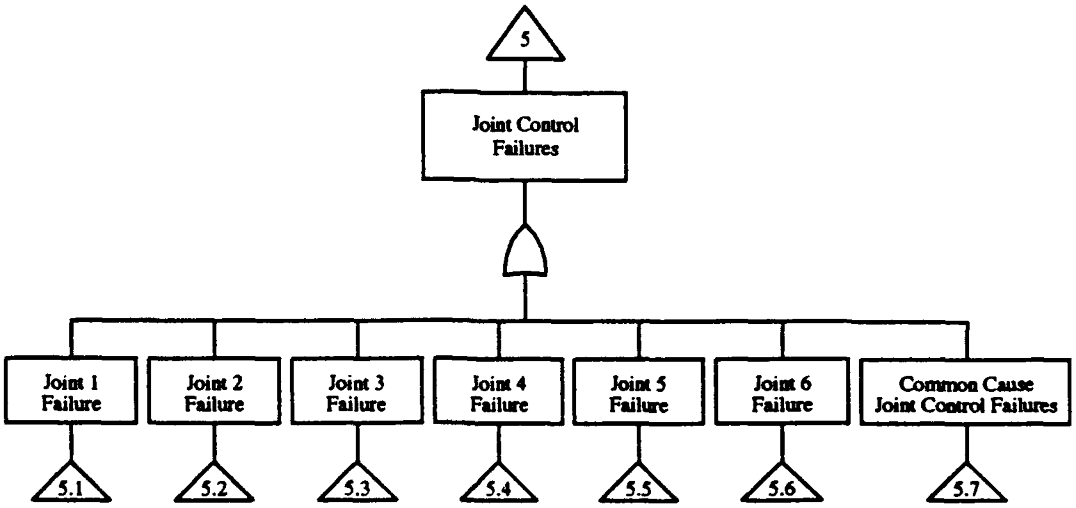


Figure A.12b. Joint Control Failures (6 Joints Assumed) [77]

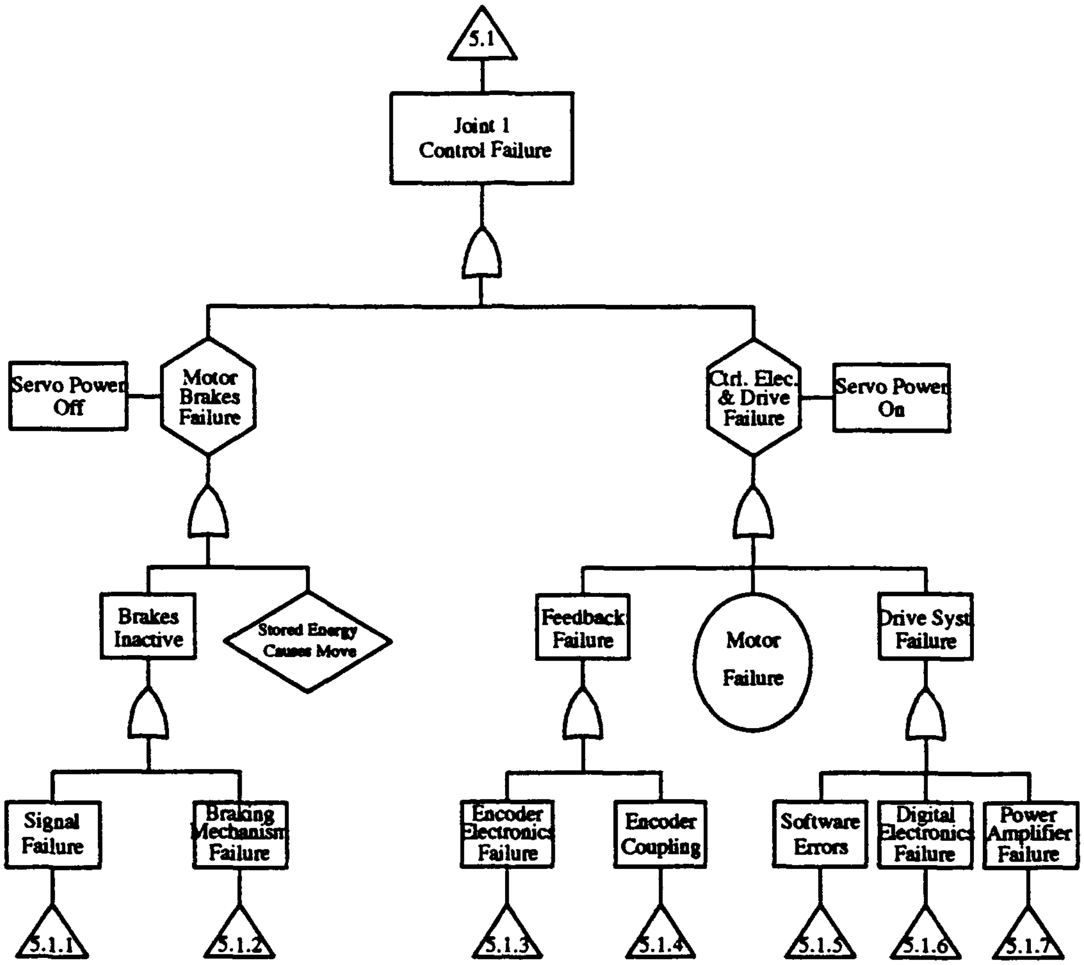


Figure A.12c. Branch 5.1: Control Failure of Joint 1 [77]

$$P(X_{n+1}=j|X_n=i,X_{n-1}=i_{n-1},\ldots,X_1=i_1,X_0=i_0)=P(X_{n+1}=j|X_n=i)=P_{ij} \quad \text{(A.41)}$$

also, $P_{ij}\geq 0,\quad i,j\geq 0;\quad \sum_{j=0}^{\infty}P_{ij}=1,\quad i=0,1,2,\ldots.$ A matrix of the transition probabilities can be denoted as **P** where

$$\mathbf{P} = \begin{bmatrix} P_{00} & P_{01} & \cdots & P_{0j} \\ P_{10} & P_{11} & & P_{1j} \\ \vdots & & \ddots & \vdots \\ P_{i0} & P_{i1} & \cdots & P_{ij} \end{bmatrix} \qquad \text{(A.42)}$$

A Markov chain can also be expressed with a transition diagram such as the one in Figure A.13. The lack of an arc between two states indicates a particular transition probability is zero.

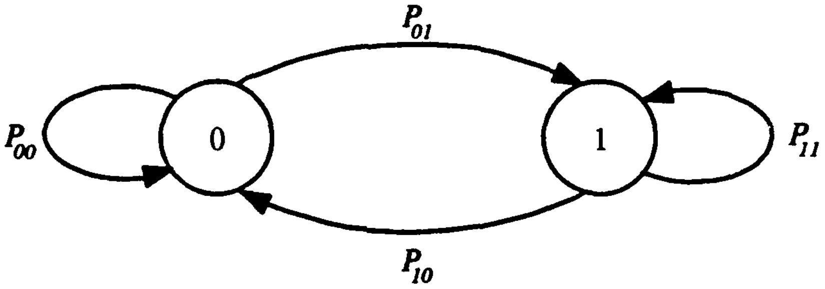


Figure A.13. Sample Markov Chain Transition Diagram

The probability transition matrix for the diagram of Figure A.13 will be a two-by-two matrix. The probability transition matrix of Equation (A.42) is a one-step probability matrix, that is, it governs the probabilities for the next transition. Of interest as well are probability transition matrices that describe the probabilities of arriving in states in an arbitrary number of transitions. We denote these probabilities as

$$P_{ij}^{n} = P(X_{n+m} = j \mid X_m = i), \quad n \geq 0, \quad i,j \geq 0 \qquad \text{(A.43)}$$

The *Chapman-Kolmogorov equations* provide an easy method of finding these n-step transition matrices as multiplying the matrix **P** by itself n times. If we let $\mathbf{P}^{(n)}$ denote the matrix of the n-step transition probabilities P_{ij}^{n}, then

$$\mathbf{P}^{(n)} = \mathbf{P}^{n-1} \cdot \mathbf{P} = \mathbf{P}^{n} \tag{A.44}$$

As the number of transitions, n, grows larger, the values in the matrix $\mathbf{P}^{(n)}$ converge to a steady state value. These steady state values are called the limiting probabilities that the process will be in state j after a large number of transitions regardless of the starting state of the process. These limiting probabilities are denoted as

$$\pi_j = \lim_{n \to \infty} P_{ij}^{n} \tag{A.45}$$

which are the unique non-negative solution of

$$\begin{aligned} \pi_j &= \sum_{i=0}^{\infty} \pi_i P_{ij}, \quad j \geq 0 \\ \sum_{j=0}^{\infty} \pi_j &= 1 \end{aligned} \tag{A.46}$$

With the preliminaries out of the way, we now face the problem of describing physical events with a Markov Chain model. As a basic guideline, a Markov chain model of a system will have k^{n} states, where k is the number of conditions for each component and n is the number of components in the system. The conditions will at least be operational or failed, with the possibility of degraded conditions in between. Thus, each state of the Markov chain will describe precisely which components are functioning and which are not.

Consider a two component system with the possible conditions for the components being operational or failed. This system will require $2^2 = 4$ states to model the system. One possible choice of states might be

state 0	both component 1 and component 2 operational
state 1	component 1 operational and component 2 failed
state 2	component 1 failed and component 2 operational
state 3	both component 1 and 2 failed

We now have a representation of a four-state Markov chain. Using some knowledge of the failure modes or if the static component reliability is known, a transition probability matrix can be formed and the limiting probabilities solved from Equation (A.46). These limiting probabilities are the long-term or steady state probability that the process will be in a particular state. Recalling the basic definition of reliability as the probability of the system being in an operational state and knowing that each state in the Markov chain is an exclusive event, we can find the reliability of the system by adding the limiting probabilities of the operational states. In our simple example, we know that state 0 is an operational state. If the two components are in a pure parallel arrangement, states 1 and 2 are operational as well with state 3 being the only failed state. In this case the reliability would be $R = \pi_0 + \pi_1 + \pi_3$. If the system was serial, state 0 is the only operational state and $R = \pi_0$.

A.2.1.2.2. Continuous Time Markov Chains

We have not as yet faced up to the problem of time dependency. The Markov chain described above is a discrete Markov chain and does not take into account time. This can be added to the Markov model by using a Continuous Time

Markov Chain (CTMC). A CTMC is a continuous time stochastic process $\{X(t), t \geq 0\}$ that takes on values in the set of non-negative integers. A CTMC also possesses the Markovian Property which can be expressed as

$$\begin{aligned} &P\{X(t+s)=j|X(s)=i, X(u)=x(u), 0 \leq u < s\} \\ &\quad = P\{X(t+s)=j|X(s)=i\} \end{aligned} \tag{A.47}$$

Again, this is the statement that the conditional distribution of the future $X(t + s)$ given the present $X(s)$ and past $X(u)$, $0 \leq u < s$, depends only on the present and is independent of the past. If $P\{X(t+s)=j|X(s)=i\}$is independent of s as well, the transition probabilities are stationary (they don't change). Another statement of the Markovian property is the fact that a CTMC is *memoryless*. If T_i is the amount of time the process stays in state i before it transitions into a different state, then by the Markovian property for all s, t ≥ 0

$$P\{T_i > s+t|T_i > s\} = P\{T_i > t\} \tag{A.48}$$

The random variable T_i is memoryless and thus must be exponentially distributed [133]. We can now say that a continuous time Markov Chain is a stochastic process moving from state to state in accordance with a discrete time Markov chain where the amount of time spent in each state before transitioning to a new state is exponentially distributed. In other words, the rates of transitions between the states are constant (See Figure A.14).

Figure A.14 is a type of CTMC known as a *Birth-Death Process*. The states can represent the numbers of a population with the birth rates represented as λ_i and the death rate as μ_i. If we use the states to represent the operational states of a system, the birth rates will represent combinations of the failure rates of the system components. The death rates will represent the repair rates of the system as well, adding the maintenance aspect into the model. One can easily sꞌꞌ how powerful a CTMC can be in modeling multi-component systems.

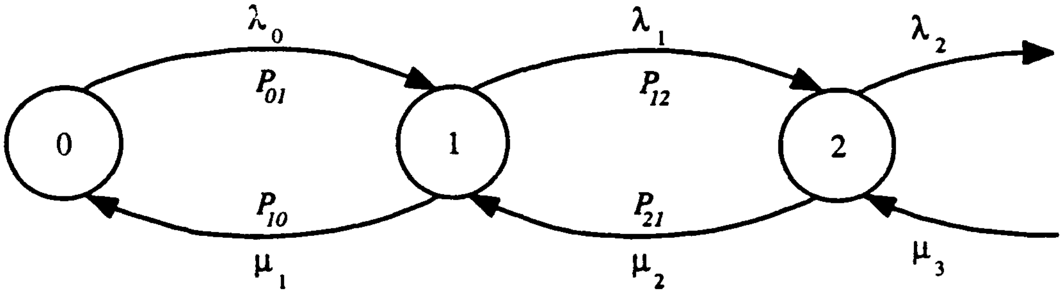


Figure A.14. Continuous Time Markov Chain Transition Diagram

Analogous to the discrete-time Markov chain, the CTMC transition probabilities will converge over time to steady state probabilities of being in a particular state of the system. A similar method of solution is also available to solve for these limiting probabilities. A CTMC is governed by the *Kolmogorov Differential Equations*. If, for two states i and j, we let v_i be the rate at which the process transitions out of state i, and P_{ij} be the probability that the transition is to state j, then we can say that

$$q_{ij} = v_i P_{ij} \tag{A.49}$$

where q_{ij} is the rate at which the process transitions into state j. This rates are represented in Figure A.14 by the λ_i and μ_i rates on the arcs. The q_{ij}'s are called the instantaneous transition rates and we can make the following observations:

$$\begin{aligned} v_i &= \sum_j v_i P_{ij} = \sum_j q_{ij} \\ \textit{and} \quad P_{ij} &= \frac{q_{ij}}{v_i} = \frac{q_{ij}}{\sum_j q_{ij}} \end{aligned} \tag{A.50}$$

The continuous time Chapman-Kolmogorov equations are

$$P_{ij}(t+s) = \sum_{k=0}^{\infty} P_{ik}(t) P_{kj}(s) \tag{A.51}$$

where $P_{ij}(t) = P\{X(t+s) = j | X(s) = i\}$ is the probability that the process in state i will be in state j at a time t later. The time rate of change in these probabilities can be represented by two sets of differential equations. The first set is called *Kolmogorov's Backward Equations* and is

$$P'_{ij}(t) = \sum_{k \neq i} q_{ik} P_{kj}(t) - v_i(t) P_{ij}(t) \tag{A.52}$$

For the Birth-Death process shown in Figure A.14, Equation (A.52) is

$$\begin{aligned} P'_{0j}(t) &= \lambda_0 [P_{1j}(t) - P_{0j}(t)] \\ P'_{ij}(t) &= \lambda_i P_{i+1,j}(t) + \mu_i P_{i-1,j}(t) - (\lambda_i + \mu_i) P_{ij}(t), \quad i > 0 \end{aligned} \tag{A.53}$$

The second set is called *Kolmogorov's Forward Equations*, which have limited application, but are applicable to birth and death processes is

$$P'_{ij}(t) = \sum_{k \neq j} q_{kj} P_{ik}(t) - v_j(t) P_{ij}(t) \tag{A.54}$$

These equations for the process of Figure A.14 are

$$\begin{aligned} P'_{i0}(t) &= \mu_1 P_{i1}(t) - \lambda_0 P_{i0}(t) \\ P'_{ij}(t) &= \lambda_{j-1} P_{i,j-1}(t) + \mu_{j+1} P_{i,j+1}(t) - (\lambda_j + \mu_j) P_{ij}(t), \quad i > 0 \end{aligned} \tag{A.53}$$

The forward equations are considered easier to solve and are used when the restrictions on their use is satisfied. For a complete derivation including all restrictions, refer to [133].

If we define the limiting probabilities as

$$P_j \equiv \lim_{t \to \infty} P_{ij}(t) \tag{A.54}$$

we can solve for the limiting probabilities using

$$\begin{gathered} v_j P_j = \sum_{k \neq j} q_{kj} P_k, \quad \textit{all states } j \\ \sum_j P_j = 1 \end{gathered} \tag{A.55}$$

which for the process of Figure A.14 are

$$\begin{aligned} P_0 &= \frac{1}{1 + \sum_{n=1}^{\infty} \frac{\lambda_0 \lambda_1 \cdots \lambda_{n-1}}{\mu_1 \mu_2 \cdots \mu_n}} \\ P_n &= \frac{\lambda_0 \lambda_1 \cdots \lambda_{n-1}}{\mu_1 \mu_2 \cdots \mu_n \left(1 + \sum_{n=1}^{\infty} \frac{\lambda_0 \lambda_1 \cdots \lambda_{n-1}}{\mu_1 \mu_2 \cdots \mu_n} \right)}, \quad n \geq 1 \end{aligned} \tag{A.56}$$

For the limiting probabilities to exist, $\sum_{n=1}^{\infty} \frac{\lambda_0 \lambda_1 \cdots \lambda_{n-1}}{\mu_1 \mu_2 \cdots \mu_n} < \infty$.

The limiting probabilities have the same meaning for the CTMC as the discrete Markov chain: P_j is the steady state probability that the process in state j.

A.2.1.2.3. Availability and the Markov Model

Since we now have the long-term probabilities that the process is in a particular state, we can immediately determine an average availability for a system with repair. Recall that availability is defined as

$$A = \frac{\text{Operating Time}}{\text{Operating Time} + \text{ Down Time}} \tag{A.57}$$

which is the long term proportion of time the system is in an operational state. With a Markov process, if we determine the proportion of time spent in the operational states, this will be the availability. These times are determined by examining a Markov process as an alternating renewal reward process, which is a more general stochastic process than the Markov Process [133], the proportion of time the process stays in state i, $i = 1, \ldots, N$ is given by

$$P_i = \frac{\pi_i \mu_i}{\sum_{j=1}^{N} \pi_j \mu_j}, \quad i = 1,2,\ldots,N \tag{A.58}$$

where μ_i is the mean time the process spends in state i and π_i is just the limiting probabilities of the embedded discrete Markov Chain (Equation (A.46)). This addition of distributions to the time spent in each state causes the CTMC to become a *Semi-Markov Process*. The mean time μ_i is just the inverse of the sum of the rates leaving the state minus the sum of the rates entering the state. For example, the mean time in state i of Figure A.14 is

$$\textit{Mean Time in State } i = \frac{1}{\lambda_{i-1} + \mu_{i+1} - \lambda_i - \mu_i}$$

There are several drawbacks to using the Markov model to model real systems. The first is the Markovian assumption which requires exponential failure and repair distributions. This problem is solved by using a renewal process model which allows general distributions to govern the transitions between states. Another problem is the large number of states required to model complicated systems and the difficulty in computing the solutions to these large models. This problem is addressed by using state reduction techniques and approximations on the bounds of the solution. Most large commercial reliability programs use Markov, Semi-Markov, and Renewal models to determine the reliability of systems. This is because of the power of the modeling technique to easily model dynamic systems.

A.2.1.3. Interference Theory (Stress-Strength Reliability Modeling) [73, 24]

At its most basic level, a failure will be caused by a stress on the device (electrical, forces, pressures, etc.) exceeding its strength (dielectric, yield strength, etc.). Interference theory is the development of expressions of probability that the stress does not exceed the strength. A general expression for the reliability will be developed followed by an example using the exponential distribution. A more in depth treatment of other distributions can be found in Reference [73].

The first assumption made is that the character of the random variables of the stress or load (**L**) and the strength (**S**) are known. Denote the p.d.f. of the load by $f_L(l)$ and the p.d.f. of the strength by $f_S(s)$. Each of these distributions will have means $\overline{L}$ and $\overline{S}$ respectively as well as variances σ_L and σ_S. The relationship of the respective densities can be seen in Figure A.15.

By definition, the reliability of the system represented by Figure A.15 is

$$R = P(\mathbf{S} > \mathbf{L}) = P(\mathbf{S} - \mathbf{L} > 0) \qquad \text{(A.59)}$$

by conditioning on a certain value of load or stress and assuming independence between the strength and load distributions, we can arrive at the integral expression

$$R = \int_{-\infty}^{\infty} f_L(l)\left[\int_{l}^{\infty} f_S(s)ds\right]dl = \int_{-\infty}^{\infty} R_S(s) f_L(l)dl \qquad \text{(A.60)}$$

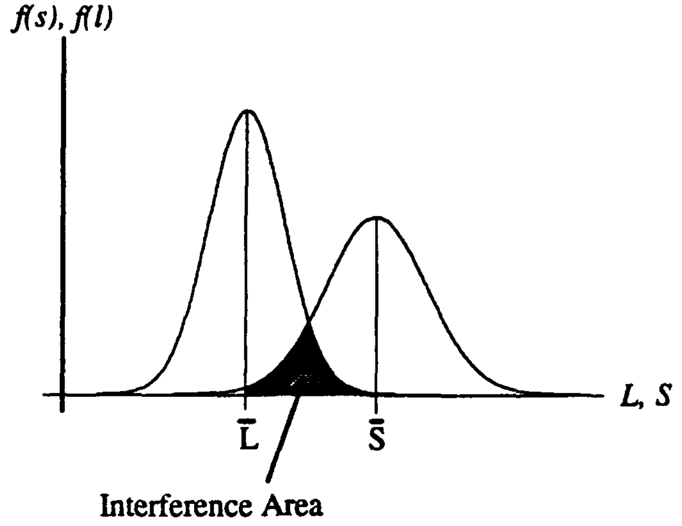


Figure A.15. Stress-Strength Interference [73]

By computing on the basis that the load must always be less than the strength, an equivalent expression is obtained as

$$R = \int_{-\infty}^{\infty} f_S(s)\left[\int_{-\infty}^{s} f_L(l)dl\right]ds = \int_{-\infty}^{\infty} F_L(l) f_S(s)ds \tag{A.61}$$

If we now introduce a new random variable **y** = **S** - **L**, where **y** is called the interference random variable, we can define the reliability as

$$R = P(\mathbf{y} > 0) \tag{A.62}$$

Now if S and L are independent random variables, both greater than or equal to zero, we can find the p.d.f. for **y** and integrate to obtain the reliability. The density function is obtained using a subtraction convolution integral as

$$\begin{aligned} f_y(y) &= \int_l f_S(y+l) f_L(l)dl \\ &= \begin{cases} \int_0^{\infty} f_S(y+l) f_L(l)dl, & y \geq 0 \\ \int_{-y}^{\infty} f_S(y+l) f_L(l)dl, & y \leq 0 \end{cases} \end{aligned} \tag{A.63}$$

We then find the reliability as

$$R = \int_0^\infty f_y(y)dy = \int_0^\infty \int_0^\infty f_S(y+l)f_L(l)dl\,dy \qquad \text{(A.64)}$$

This immediately enters the area of the algebra of random variables described in Section A.1.4 and Section 3.5.

To illustrate this method of reliability calculation, the reliability for exponentially distributed stress and strength will be determined [73]. The p.d.f.'s in this case are for strength **S**,

$$f_S(s) = \lambda_S e^{-\lambda_S s}, \quad 0 \leq s < \infty$$

and for the stress or load **L**,

$$f_L(s) = \lambda_L e^{-\lambda_L s}, \quad 0 \leq l < \infty$$

making the substitutions into (A.42) results in

$$R = \int_0^\infty \lambda_L e^{-\lambda_L l}[e^{-\lambda_S l}]dl$$

$$= \int_0^\infty \lambda_L e^{-(\lambda_L + \lambda_S)l} dl$$

$$= \frac{\lambda_L}{\lambda_L + \lambda_S} \int_0^\infty (\lambda_L + \lambda_S) e^{-(\lambda_L + \lambda_S)l} dl$$

Recognizing that the integral expression is just an exponential and is unity, the reliability for exponentially distributed stress and strength is

$$R = \frac{\lambda_L}{\lambda_L + \lambda_S} \qquad \text{(A.65)}$$

Recall that the mean is the inverse of the failure rate for the exponential, we get

$$R = \frac{\bar{S}}{\bar{L} + \bar{S}} \qquad \text{(A.66)}$$

This modeling strategy can also be used to model the changes in the distributions over time. The extension involves recognizing the different ways a random variable can change over time as well as the changes in the distributions over time. The Equations (A.60) and (A.61) can be used to develop a reliability figure for a single cycle of stress application. Kapur and Lamberson develop these expressions for a series of cyclic applications of stress and strength for three different type of stress and strength random variables: deterministic, the value is not random and does not change over time; random fixed, where the initial value in the first cycle is a random sample but then changes in a deterministic way over time; and random independent, where the variable realizes independent random samples over time [73]. The mathematics are complicated but are straight forward. Another good reference using interference theory to develop electronic device reliability models is Brombacher's book, *Reliability by Design: CAE Techniques for Electronic Components and Systems* [20]. This book shows explicit derivations of the stress and strength distribution for electronic components and uses them to optimize the reliability of electronic systems during design. The same methods can be used in a probabilistic design strategy for mechanical systems.

A.2.2. Failure Mode, Effects, and Criticality Analysis [68]

A Failure Mode, Effects, and Criticality Analysis (FMECA) is an evaluation tool used during any phase of design to identify all conceivable and potential modes of each component of a system and to determine its effect on the system. A FMECA is a bottom-up analysis as compared to the top-down approach of Fault Tree analysis. The FMECA methodology is illustrated in Figure A.16.

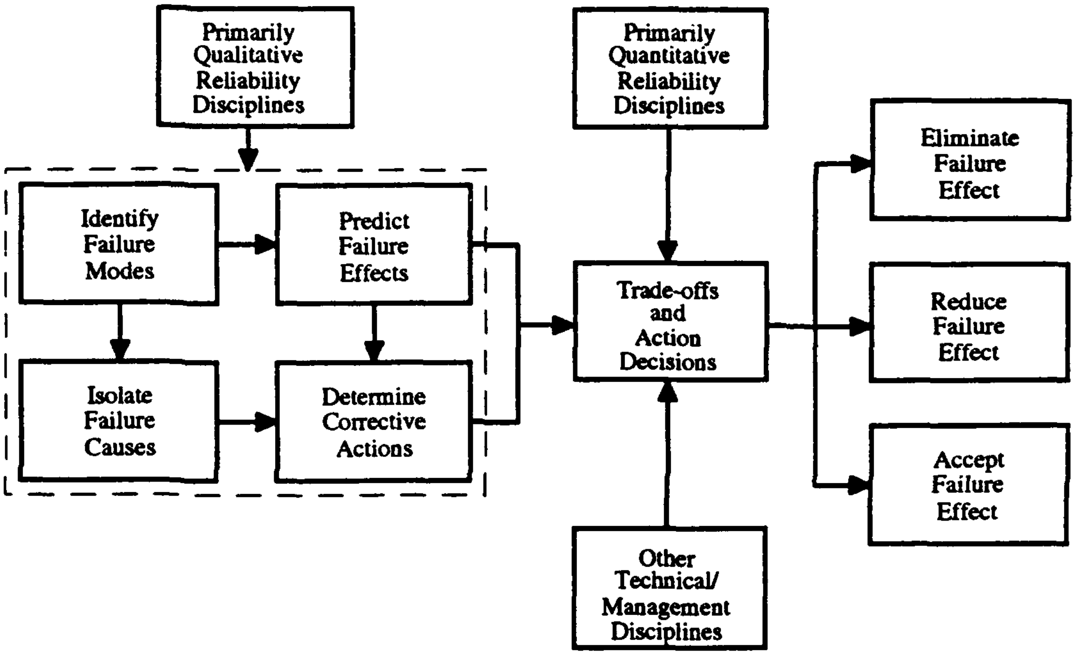

Primarily Qualitative Reliability Disciplines
Primarily Quantitative Reliability Disciplines
Identify Failure Modes
Predict Failure Effects
Isolate Failure Causes
Determine Corrective Actions
Trade-offs and Action Decisions
Other Technical/ Management Disciplines
Eliminate Failure Effect
Reduce Failure Effect
Accept Failure Effect

Figure A.16. FMECA Methodology [68]

The basic procedures for a FMECA starts with an assumption of a failure of a component or subsystem and delineates all the possible failure modes along with the failure mechanisms. The effect of each failure mode is then traced upwards through the system to the top level. A criticality rating is developed for each failure mode and its effect based upon the probability of occurrence, the severity of the effects, and the detectability of the failure. Design changes to reduce this criticality can then be suggested.

The FMECA has nine basic steps [68]:

1. Development of system description and block diagrams.
2. Identification of failure modes.
3. Identification of failure causes (failure mechanisms).
4. Determination of the effects of the failures.
5. Description of symptoms and detection features.

6. Classification of severity of failure effects.
7. Determination of probability of occurrence.
8. Performance of criticality analysis.
9. Corrective action and follow-up.

1. Development of system description and block diagrams. The first step in the analysis is the development of the data required to perform it. This data includes as complete a description of the item as possible along with functional diagrams illuminating all inter-dependencies of the system. All system interfaces should be represented as well as the initial indenture level. The indenture level is the level at which the FMECA will start, such as the piece part level or sub-assembly level, etc. The lower the level the greater detail is required and the more analysis is needed. The initial analysis should begin at whatever indenture level is available earliest in the design process. As the design matures, the indenture level of the FMECA should go down as far as possible. This depth should be the lowest level for which data is available to determine the functional relationships.

2. Identification of failure modes. The next step is to determine the failure modes. These are the manners in which a component or system fails such that it does not meet its performance specifications. From an examination of the block diagrams, the analyst can determine all realistically probable failure modes of the outputs of each block in the function and output list (Figure A.17.)

3. Identification of failure causes (failure mechanisms). The next step is to describe the possible failure mechanisms that cause the failure modes listed in step 2. The probability of each failure mechanism can be estimated as well as possible corrective actions to prevent the failure mechanism or mitigate its effect

4. Determination of the effects of the failures. These are the consequences of the failures and their effects on the system. The failure effects of the next indenture level and may propagate to the top level of the system. There may be local effects other than the failure itself and there is the end effect, or the highest level of indenture that the failure is discernible.

5. Description of symptoms and detection features. The symptoms are the behaviors of the system that indicate that a malfunction within the system has occurred. They may be local, only effecting a small part or single mode of system operation, or they may effect the overall system. Performance monitoring devices may be necessary to detect some of the symptoms, depending upon the criticality of the possible malfunction. The descriptions are necessary to provide proper correlation between the malfunction and the symptoms. These descriptions provide failure indications that the operators of the system are trained to identify as well as providing input to the development of fault isolation procedures.

6. Classification of severity of failure effects. These are classifications that are assigned to quantify the potential consequences resulting from a failure. A classification should be assigned to each failure mode. MIL-STD-1629 recommends the classifications described in Table A.2.

7. Determination of probability of occurrence. This data is determined analytically based on the expected time between overhaul, mission time, or any other interval deemed appropriate. For tabulation, the failure probability levels can be ranked from frequent to extremely unlikely. Generic failure rate data such as that in MIL-STD-217 can also be used to generate these estimates.

Table A.2. Severity Classifications [68]

Category	Classification	Description	β
I	Catastrophic	Failure causes death or mission loss (loss of system)	1.0
II	Critical	Failure causes severe injury or major system damage	$0.1 < \beta < 1.0$
III	Marginal	Failure causes minor injury or degrades mission	$1 < \beta \leq 0.1$
IV	Minor	No injury or system damage but may result in system failure and unscheduled maintenance	0

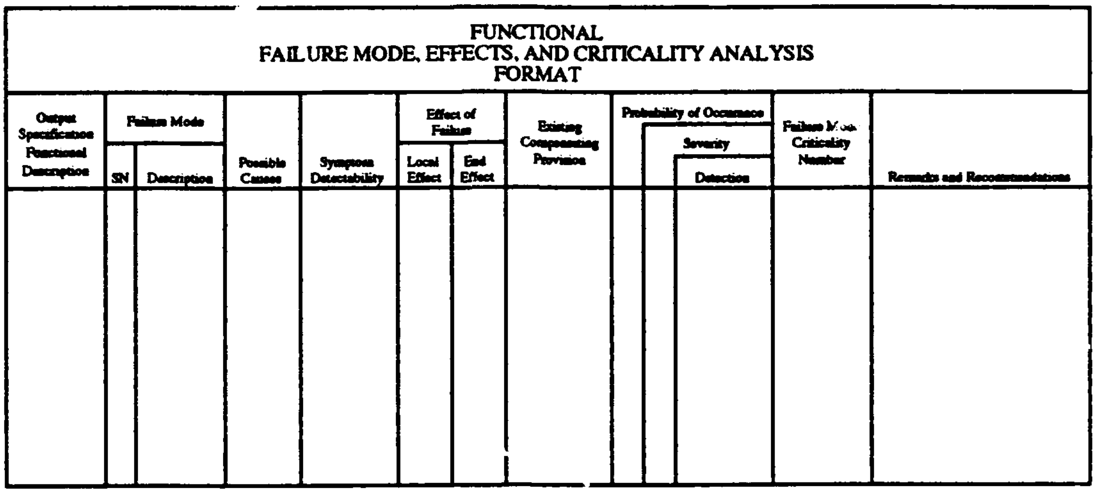
8. Performance of criticality analysis. The criticality of the failure mode is the combination of the probability of occurrence of the failure and the level of severity. This gives the designer insight into the true impact of the faults since a fault that has a low impact but happens very frequently would be more of a problem than one with sever consequences but is very improbable. MIL-STD-1626 suggests the following formulation. Denote the Failure Mode Criticality Number as C_m which is the criticality for a particular failure mode of a component. C_m is given by

$$C_m = \beta\alpha\lambda_p t \qquad \text{(A.67)}$$

β represents the conditional probability that the failure effect will result in the identified severity given that the failure mode occurs. These suggested probabilities are listed in Table A.17. The parameter α is called the Failure Mode Ratio which is the fraction of the part failure rate, λ_p, corresponding to the particular failure mode of the part under consideration. These decimal fraction multipliers can be derived from the part failure data or analyst judgment. λ_p is the part failure rate estimated from the reliability prediction task of MIL-STD-785 (See Chapter 3). The environmental and operational factors used to scale the failure rate should be annotated on the worksheet as well. The parameter t is the operating time in hours or mission cycles depending upon the system specification.

Another criticality number, C_r, can also be calculated at this point. This is the *Item Criticality Number* which represents how critical a particular item is in its effects to the overall system. For a particular part, C_r is the sum of all the C_m of each failure mode for that part. A criticality matrix can now be generated to help designers focus their attention to the most critical components of the system.

9. Corrective action and follow-up. This is the step of the FMECA that results in design improvements. There are three types of corrective actions that can result from the FMECA. The first is to eliminate the cause of failure. This might be expensive and difficult to do especially on a mature design. However, knowledge of the criticality information early in the design will allow for better design choices. The second type of corrective action is to reduce the severity of the failure by the inclusion of fail-safe and/or fault-tolerant schemes in the design. This is a more robust approach to the design problem but will result in more complexity and cost. The third option is to increase the probability of detection through sensors and fault monitoring. If the system can be monitored for impending failure, then maintenance can be performed when indicated to prevent the failures from occurring (See the discussion of Reliability Centered Maintenance in Chapter 3).



FUNCTIONAL
FAILURE MODE, EFFECTS, AND CRITICALITY ANALYSIS
FORMAT

Output Specification Functional Description	Failure Mode		Possible Causes	Symptom Detectability	Effect of Failure		Existing Compensating Provision	Probability of Occurrence			Failure Mode Criticality Number	Remarks and Recommendations
	SN	Description			Local Effect	End Effect			Severity			
										Detection		

Figure A.17. Sample FMECA Worksheet [68]

APPENDIX B: DESIGN GUIDE FOR MODULAR ROBOT RELIABILITY AND MAINTAINABILITY

This appendix compiles the recommendations for the enhancing the reliability of modular robotic systems from Chapters 3 and 4 into checklists for designers to use during their design reviews. These checklists cover the generation of an overall reliability program within an organization and the application of reliability techniques to a product development. The optimum time for the product checklist to be used is during the original problem definition and specification stage of the design.

Table B.1. *Checklist for Organizational Reliability Programs*

Step	Description
1.	Ascertain or specify where the organizational reliability and maintainability goals are determined (or should be determined) and consider whether these requirements are internal or contractually imposed.
2.	Insure management is aware of the impact of reliability and maintainability policies on product development and product life-cycle-cost.
3.	Determine or specify the managerial level at which reliability and maintainability policy will be set. Insure policies are consistent with goals.
4.	Provide reliability and maintainability training to all engineers and technicians as well as management.
5.	Consider implementing Total Quality Management practices within the organization

Table B.2. Checklist for Design of Modular Robot Systems

Step	Description
1.	Develop/determine and specify the definition of failure for the system.
2.	Develop Reliability and Maintainability (R&M) specifications
3.	Select applicable R&M program tasks
a.	R&M Program Plans
b.	R&M and Design Program Reviews
c.	Monitor and Control of Subcontractors
d.	Failure Reporting, Analysis. and Corrective Action System (FRACAS)
e.	Failure Review Boards
f.	R&M Modeling, Allocation, and Prediction
g.	Failure Mode, Effects, and Criticality Analysis (FMECA)
h.	Fault Tree Analysis (FTA)
i.	Sneak Circuit and Tolerance Analysis
j.	Parts Program and Derating
k.	Reliability Critical Items
l.	Environmental Stress Screening
m.	Reliability Development and Growth Testing
n.	Reliability Qualification Testing (RQT)/Product Reliability Acceptance Testing (PRAT)
o.	Maintainability Demonstration

Table B.3. Checklist for Reliability Design Considerations for Modular Robot Systems

Step	Description
1.	Trade-off reliability enhancement of direct-drive vs. geared systems (performance: torque vs. weight issues). Gear systems are high failure rate components in the mechanical system. Direct drive systems will have higher reliability but lower payload (due to current torque limitations).
2.	Eliminate motors with brushes for commutation (use brushless dc motors or ac induction motors) or choose motors with maximum brush life.
3.	Provide for parts derating on both electronic systems and mechanical systems
4.	Environmental Control Corrosion resistance Heat control Cable stress control
5.	Interface load control to prevent interface failure
6.	Minimize number of sensors and maximize sensor information

APPENDIX C: RESULTS OF THE RELIABILITY DATA SEARCH

This appendix presents the results of the search made for reliability data throughout this research effort. As stated in Chapter 3 and 4, the lack of data is a primary problem in the evaluation of reliability. Attempts to acquire reliability data from industry representatives all ultimately ended in a statement that the requested data was proprietary. This attitude is understandable from the competition aspect, since the reliability of a system is of prime importance when selecting a vendor. However, this attitude prevents any researcher from investigating the overall state of robotic system reliability in the robotics industry. It also prevents researchers from making good suggestions, based on data, to improve the state-of-the-art in robotic system reliability on an industry-wide level. This forces the researcher to utilize generic data to analyze and generalize research needs and prioritizations for general robotic systems, as was done in Chapter 4. Generic data will always provide a conservative estimate of the system reliability and does not account for the actual environment the system will be used in. This is why one of the primary recommendations of this effort is the establishment of a nationwide robotic component reliability data base. This data base would provide general access to robotic reliability data, allowing industry-wide improvement in the reliability of robotic systems.

Table C.1. Results of the Reliability Data Search

Date	Organization	Contact	Result
5 Jun 92	NASA JSC	Charles Price	No space-based robotic system reliability data available. Specify Fault-tolerance, not reliability
9 Jul 92	Transitions Research Corporation	John Evans	No data or documentation available. Referred to Ira Pence at Georgia Tech.
9 Jul 92	GA Tech	Dr. Ira Pence	Experienced 99% availability while at Unimation. No data or documentation available.
22 Sep 92	Oak Ridge National Lab	Dr Jonathan Haire	Only performed FTA of telerobotics. No reliability predictions or data available.
22 Sep 92	Nissan	Steve Bone	No failure data available for non-company use. Did speak of general technology trends and experiences.
29 Sep 92	Clarkson U.	Dr. D. Wells	No data. Referred to Tetra Engineering (DOE Contractor)
29 Sep 92	Tetra Engineering	Peter Jackson	No data releasable. Try DOE or Carnegie Mellon University (CMU) (Dr Whittaker)
5 Oct 92	Staubli, USA	Mike Cozza	No data available. Try Staubli, France No response from phone or FAX.
4 Nov 92	Adept	Tim Coogan	Spoke in general terms. Described specific technologies and tools used to increase R&M. Tracked R&M data but would not release.

Table C.1. (Continued). Results of the Reliability Data Search

Date	Organization	Contact	Result
18 Nov 92	ABB	Peter Carlson	Tracks reliability during warranty only. Will not release data.
23 Nov 92	GM Fanuc	Gary Zywiol	Tracks all reported reliability data. Will not release data.
23 Nov 92	GM Fanuc	Dr Akeel	Use catalog life data during design. Has only provided incremental increases in product reliability.
24 Nov 92	Motoman	Bill DeCamp	Quoted 18,000 hour MTBF robot and controller. Provide 3 yr warranty. Would not provide statement of environment or support data.
19 Jan 93	CMU	Erik Krotkov	Do not track reliability in any program.
19 Jan 93	Robotics Research	Jim Garland	All reliability data is Proprietary. No data releasable.
29 Jan 93	Reliability Analysis Center (RAC)	Bill Denson	Provided generic data via MIL-HDBK-217 [96] and NPRD-91 [131]. Stated RAC could not obtain data either.
2 Apr 93	IBM	Bruce Shove	Reliability data is proprietary and cannot be released.
2 Apr 93	Dell Computer	Brian Jarrett	Reliability data is proprietary and cannot be released.

BIBLIOGRAPHY

[1] Aalund, M., et. al., "Development of a Digital Intelligent Servo-Controller for Modular Robots," Intelligent Power and Control Conference, Detroit, MI, 5-8 Apr, 1993.

[2] The Advisory Group on Reliability of Electrical Equipment (AGREE) Report, "Reliability of Military Electronic Equipment," Office of the Assistant Secretary of Defense, Washington, D. C., The Government Printing Office, 1957.

[3] Air Force Institute of Technology, *System Reliability/Maintainability Textbook*, School of Systems & Logistics, Air Force Institute of Technology, Air University, Wright-Patterson AFB, OH, 1985.

[4] Ambrose, R. and Tesar, D., "Design, Construction, and Demonstration of Modular Reconfigurable Robots," Research Report, Mechanical Engineering Dept, University of Texas at Austin, Austin TX, August 1991.

[5] Ammar, H., Huang, Y., and Liu, R., "Hierarchical Models for Systems Reliability, Maintainability, and Availability," *IEEE Transactions on Circuits and Systems*, v. CAS-34, n. 6, June 1987, pp. 629-638.

[6] Andeen, G. (Ed), *Robot Design Handbook*, SRI International, McGraw-Hill, New York, 1988.

[7] Anderson, R. and Neri, L., *Reliability Centered Maintenance*, Elsevier Science Publishers, Ltd., London, 1990.

[8] Asada, H. and Youcef-Toumi, K., *Direct-Drive Robots: Theory and Practice*, MIT Press, Cambridge, MA, 1987.

[9] ASEA, *Industrial Robot IRb-6 Service Manual*, ASEA, Inc., Västerås, Sweden, 1981.

[10] Aslaksen, E. and Belcher, R., *Systems Engineering*, Prentice Hall, NY, 1992.

[11] Åström, K. and Wittenmark, B., Adaptive Control, Addison-Wesley Publishing Co., Reading, MA, 1989.

[12] Avizienis, A., "Fault-Tolerant Systems," *IEEE Transactions on Computing*, v. C-25, n. 12, Dec 1976, pp. 1304-1311.

[13] Benhabib, B., et. al., "Design of a Rotary-Joint-Based Modular Robot," ASME Mechanisms Conference, Chicago, IL, DE-Vol. 26, Sep 1990, pp. 239-243.

[14] Benhabib, B. and Dai, M., "Mechanical Design of a Modular Robot for Industrial Applications," *Journal of Manufacturing Systems*, v. 10, n. 4, 1991, pp. 297-306.

[15] Berry, T., *Managing the Total Quality Transformation*, McGraw-Hill, New York, 1991.

[16] Bevill, P. and Tesar, D., "Criteria Normalization to Support Decision Making in Intelligent Machines," Research Report, Mechanical Engineering Department, University of Texas at Austin, Austin, TX, May 1990.

[17] Bhatti, P., "Probabilistic Modeling and Design of Robot Manipulators," Dissertation, Department of Mechanical Engineering, Purdue University, Lafayette, IN, 1989.

[18] Bhatti, P. and Rao, S., "Reliability Analysis of Robot Manipulators," *Journal of Mechanisms, Transmissions, and Automation in Design*, v.110, June 1988, pp. 175-181.

[19] Billinton, R. and Allan, R., *Reliability Evaluation of Power Systems*, Plenum Press, New York, 1984.

[20] Brombacher, A., *Reliability by Design: CAE Techniques for Electronic Components and Systems*, John Wiley and Sons, Chichester, UK, 1992.

[21] Brooks, T., ed., *The Local Area Network Reference Guide*, Prentice-Hall, Engelwood Cliffs, NJ, 1985.

[22] Butler, M. and Tesar, D., "An Applications-Based Assessment of Present and Future Robot Development," Research Report, Department of Mechanical Engineering, University of Texas at Austin, Austin, TX, May 1992.

[23] Carderock Division, *Handbook of Reliability Prediction Procedures for Mechanical Equipment,* Systems Department, Naval Surface Warfare Center, Bethesda, MD, CARDEROCKDIV NSWC-92/L01, May 1992.

[24] Carter, A., *Mechanical Reliability*, J. Wiley & Sons, New York, 1986.

[25] Cho, Y. and Biem, Z., "Reliable Control via an Additive Redundant Controller," *International Journal of Control*, v. 50, n. 1, Jan 1989, pp. 385-398.

[26] Christian, N. and Passauer, L., *Fiber Optic Component Design, Fabrication, Testing, Operation, Reliability, and Maintainability,* Noyes Data Corporation, Park Ridge, NJ, 1989.

[27] Cincinnati Milacron, *T^3-726 Industrial Robot Parts Manual,* Cincinnati Milacron, Inc., Cincinnati, OH, 1984.

[28] Cleary, K. and Tesar, D., " Decision Making Software for Redundant Manipulators," Research Report, Mechanical Engineering Department, University of Texas at Austin, Austin, TX, March 1990.

[29] Cleary, K. and Tesar, D., "Incorporating Multiple Criteria in the Operation of Redundant Manipulators," 1990 IEEE Robotics and Automation Conference, Cincinatti, OH, 13-18 May, 1990.

[30] Clifford, M., *Electric/Electronic Motor Data Handbook,* Prentice-Hall, Engelwood Cliffs, NJ, 1990.

[31] Cohen, P. H. and Chandra, M. J., "A Framework for the Reliability Modeling of Robotic Cells," AUTOFACT 6 Conference, Anaheim, CA, 1-4 Oct 1984, pp. 17-30.

[32] Collins, J., *Failure of Materials in Mechanical Design: Analysis, Prediction, Prevention,* John Wiley & Sons, New York, 1981.

[33] Colson, J., "Performance Measures for Robotic Systems," Master's Thesis, University of Texas at Austin, Austin, TX, Dec 1984.

[34] Colson, J. and Perreira, N., "Robotic System Pose Performance: Definitions and Analysis," Proceedings, 1985 ASME International Computers in Engineering Conference, v.1, Boston, MA, 4-8 Aug 1985, pp. 247-257.

[35] Cox, D. and Tesar, D., "Decision Making for Intelligent Control of Dual-Arm Robotic Operations," Research Report Mechanical Engineering Dept, University of Texas at Austin, Austin, TX, M[illegible] 1992.

[36] Craig, J., *Introduction to Robotics Mechanics and Control*, Addison-Wesley, New York, 1989.

[37] Crawford, R. H. and Rao, S. S., "Reliability Analysis of Function Generating Mechanisms Through Monte Carlo Simulation," 1987 ASME Design Technology Conferences - The Design Automation Conference, Boston, MA, 27-30 Sept 1987, pp.197-202.

[38] Dai, S. and Wang, M., *Reliability Analysis in Engineering Applications*, Van Nostrand Reinhold, New York, 1992.

[39] Denavit, J. and Hartenberg, R., "A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices," *ASME Journal of Applied Mechanics*, June 1955, pp. 215-221.

[40] Dhande, S. and Chakraborty, J., "Analysis and Synthesis of Mechanical Error in Linkages - A Stochastic Approach," *Journal of Engineering for Industry*, v. 95, n.3, 1973, pp. 672-676.

[41] Dhillon, B., *Life Cycle Costing*, Gordon and Breach Science Publishers, New York, 1989.

[42] Dhillon, B., *Robot Reliability and Safety*, Springer-Verlag, New York, 1991.

[43] Dhillon, B. and Reiche, H., *Reliability and Maintainability Management*, Van Nostrand Reinhold, New York, 1985.

[44] Electric Power Research Institute (EPRI), "Robot Applications for Nuclear Power Plant Maintenance," Interim Research Report, EPRI NP-3941, Project 2232-4, Palo Alto, CA, March 1985.

[45] Engelberger, J., "Three Million Hours of Robot Field Experience," *The Industrial Robot*, July 1974, pp. 164-168.

[46] Engelberger, J., *Robotics in Practice*, AMACOM, 1980.

[47] Erickson, J., "Manned Spacecraft Automation and Robotics," *Proceedings of the IEEE*, v. 75, n. 3, Mar 1987, pp. 417-426.

[48] Erickson, J., Price, C., and Cooke, D., "Future Needs for Space Robots for SEI," Symposium on Cooperative Intelligent Robotic Systems in Space II, SPIE Proceedings, v. 1612, Boston, MA, 10-14 Nov 1991, pp. 2-12.

[49] Finger, S. and Dixon, J., "A Review of Research in Mechanical Engineering Design. Part I: Descriptive, Prescriptive, and Computer-Based Models of Design Processes," *Research in Engineering Design*, v. 1, 1989, pp. 51-57.

[50] Finger, S. and Dixon, J., "A Review of Research in Mechanical Engineering Design. Part II: Representations, Analysis, and Design for the Life Cycle," *Research in Engineering Design*, v. 1, 1989, pp. 121-137.

[51] Fisher, R., "Design and Cost Considerations for Permanent Magnet DC Motor Applications," *Power Conversion and Intelligent Motion*, v. 17, n. 7, July 1991, pp. 18-24.

[52] Fisher, W. and Price, C., "Space Station Freedom External Maintenance Task Team: Final Report," v. 1, LBJ Space Center, National Aeronautics and Space Administration, Houston, TX, July 1990.

[53] Fuqua, N., *Reliability Engineering for Electronic Design*, Marcel Dekker, New York, 1987.

[54] Gao, D. and Wells, D., "Simulation Modeling of Robot Accuracy and Repeatability," 1990 ASME Technical Design Conference - 2nd Conference in Flexible Assembly Systems, Chicago, IL, 16-19 Sept 1990, pp. 59-63.

[55] Gao, D. and Wells, D., "Statistical Characteristics of Robot Accuracy and Repeatability in Workspace," 1990 ASME Technical Design Conference - 2nd Conference in Flexible Assembly Systems, Chicago, IL, 16-19 Sept 90, pp. 91-97.

[56] Gao, D. and Wells, D., "Robotic Assembly Operation Reliability," 1990 ASME Technical Design Conference - 2nd Conference in Flexible Assembly Systems, Chicago, IL, 16-19 Sept 90, pp. 65-70.

[57] Geist, R. and Trivedi, K., "Reliability Estimation of Fault-Tolerant Systems: Tools and Techniques," *Computer*, July 1990, pp. 52-61.

[58] Gini, M., "The Future of Robot Programming," *Robotica*, v. 5, pt. 3, Jul/Sept 1987, pp. 235-246.

[59] Gordon, J. and Curry, J., "Reliability Analysis of the RRV-1 Robot," 33rd Conference on Remote Technology, San Francisco, CA, 11-14 Nov 1985, pp. 210-214.

[60] Grant, E. and Leavenworth, R., *Statistical Quality Control*, McGraw-Hill, New York, 1988.

[61] Grundmann, J., "Reliability, Availability, and Maintainability for a Laboratory Automated Storage and Retrieval System," *LRA*, v. 1, 1989, pp. 95-104.

[62] Haugen, E., *Probabilistic Mechanical Design*, John Wiley & Sons, New York, 1980.

[63] Hooper, R., "The Interactive Assembly and Computer Animation of Reconfigurable Robot Systems," Master's Thesis, University of Texas at Austin, Austin, TX, December 1990.

[64] Hsiao, M., et. al., "Reliability, Availability, and Serviceability of IBM Computer Systems: A Quarter Century of Progress," *IBM Journal of Research and Development*, v. 25, n. 5, Sept 1981, pp. 453-465.

[65] Hsu, Y. and Hsu, C., "Evaluation of Reliability and Safety of Long-Term Unmaintained Computer Systems," *International Journal of Electronics*, v. 70, n. 2, Feb 1991, pp. 389-405.

[66] Hudgens, J., "Static Robot Compliance and Metrology Procedures with Application to a Light Machining Robot," Dissertation, Mechanical Engineering Dept, University of Texas at Austin, Austin, TX, August 1992.

[67] Iconis, J., et. al., Design and Prototype Development of Robot Actuator Modules, Technical Report, Mechanical Engineering Dept., University of Texas at Austin, Austin, TX, December 1991.

[68] Ireson, W. and Coombs, C., Jr. (eds.), *Handbook of Reliability Engineering and Management*, McGraw-Hill, New York, 1988.

[69] Johnson, A. and Malek, M., "Survey of Software Tool for Evaluating Reliability, Availability, and Serviceability," *ACM Computing Surveys*, v. 20, n. 4, Dec 1988, pp. 227-269.

[70] Johnson, B., *Design and Analysis of Fault-Tolerant Digital Systems*, Addison-Wesley Publishing Co., Reading, MA, 1989.

[71] Jones, J., *Engineering Design: Reliability, Maintainability, and Testability*, TAB Books, Inc, Blue Ridge Summit, PA, 1988.

[72] Jones, R. and Dawson, S., "People and Robots - Their Safety and Reliability," in *Robot Safety*, Edited by Bonny and Yong, Springer-Verlag, Berlin, 1985, pp. 65-81.

[73] Kapur, K. and Lamberson, L., *Reliability in Engineering Design*, John Wiley, New York, 1977.

[74] Karmarkar, U. and Kubat, P., "Modular Product Design and Product Support," *European Journal of Operational Research*, v. 29, n. 1, April 1987, pp. 74-82.

[75] Kellogg, S., "Algebraic Functions of H-Functions with Specific Dependency Structures," Dissertation, Department of Mechanical Engineering, University of Texas at Austin, Austin, TX, May, 1984.

[76] Khodabandehloo, K., Duggan, F, and Husband, T. M., "Reliability of Industrial Robots: A Safety Viewpoint," Proceedings: 7th British Robot Association Annual Conference, Cambridge, UK, 14-16 May 1984, pp. 223-242.

[77] Khodabandehloo, K., Duggan, F, and Husband, T. M., "Reliability Assessment of Industrial Robots," Proceedings: 14th International Symposium on Industrial Robots, Gothenburg, Sweden, 2-4 Oct 1984, pp. 209-220.

[78] Khodabandehloo, K., Sayles, R., and Husband, T., "Safety Integrity Assessment of Robot Systems," Proceedings, 4th IFAC Workshop: Safety of Computer Control Systems 1985, Como, Italy, 1-3 Oct 1985, pp. 61-64.

[79] Kopetz, H., *Software Reliability*, Springer-Verlag, New York, 1980.

[80] Krishnan, A. and Khosla, P., "A Methodology to Determine the Dynamic Configuration of a Reconfigurable Manipulator," 1st International Symposium on Measurement and Control in Robotics, Houston, TX, 20 Jun 1990, pp. F2.2.1-F2.2.8.

[81] Kubat, P., "Assessing Reliability of Modular Software," *Operation Research Letters*, v. 8, n.1, Feb 1989, pp. 35-41.

[82] Lakner, A. and Anderson, R., *Reliability Engineering for Nuclear and Other High Technology Systems: A Practical Guide*, Elsevier Applied Science Publishers, New York, 1985.

[83] Leatham-Jones, B., *Elements of Industrial Robotics*, Pitman Publishing, London, 1987.

[84] Liang, E., Abolrous, S., and Husseiny, A., "Logic Reliability Analysis of Adaptive Control Strategies," *Annals of Nuclear Energy*, v. 16, n. 5, 1989, pp. 231-243.

[85] Littlewood, B., "Software Reliability Model for Modular Program Structure," *IEEE Transactions on Reliability*, v. 28, n. 3, Aug 1979, pp. 241-246.

[86] Longbottom, R., *Computer System Reliability*, John-Wiley and Sons, Chichester, UK, 1980.

[87] Luneau, J., "A New Feedback Device for Brushless DC Servo Systems," *Power Conversion and Intelligent Motion*, September 1986.

[88] Masuda, Y., et. al., "A Statistical Approach for Determining Release Time of Software System with Modular Structure," *IEEE Transactions on Reliability*, v. 38, n. 3, Aug 1989, pp. 365-372.

[89] McAndrew, D. and Tesar, D., "Assessment of Microelectronics Assembly in Terms of the Development of Precision Layered Control Mechanisms," Research Report, Mechanical Engineering Dept, University of Texas at Austin, Austin, TX, Dec 1991.

[90] McGough, J., Reibman, A., and Trivedi, K., "Markov Reliability Models for Digital Flight Control Systems," *Journal of Guidance, Control, and Dynamics*, v. 12, n. 2, Mar/Apr 1989, pp. 209-219.

[91] McInroy, J. E. and Saridis, G. N., "Reliability Analysis in Intelligent Machines," *IEEE Transactions on Systems, Man, and Cybernetics*, v. 20, n. 4, Jul/Aug 1990, pp. 950-956.

[92] McInroy, J. and Saridis, G., "Reliable Control and Sensor Fusion in Intelligent Machines," Proceedings: 1991 IEEE International Conference on Robotics and Automation, Sacramento, CA, 9-11 Apr 1991a, pp. 487-492.

[93] McInroy, J. and Saridis, G., "Reliability-Based Control for Intelligent Machines," *Safety, Reliability, and Human Factors in Robotic Systems*, J. Graham, Ed., Van Nostrand Reinhold, NY, 1991b, pp. 163-197.

[94] Mellor, P., "Software Reliability Modeling: The State of the Art," *Information and Software Technology*, v. 29, n. 2, March 1987, pp. 81-98.

[95] Mendenhall, W., Schaeffer, R., and Wackerly, D., *Mathematical Statistics with Applications*, Duxbury Press, Boston, MA, 1981.

[96] MIL-HDBK-217F (Notice 1), *Reliability Prediction for Electronic Equipment*, Naval Publications and Forms Center/NPODS, Philadelphia, PA, 10 Jul 1992.

[97] MIL-HDBK-263A , *Electrostatic Discharge Control Handbook for Protection of Electrical and Electronic Parts, Assemblies, And Equipment (Excluding Electrically Initiated Explosive Devices)*, Naval Publications and Forms Center/NPODS, Philadelphia, PA, 22 Feb 1991.

[98] MIL-HDBK-338A , *Electronic Reliability Design Handbook,* Naval Publications and Forms Center/NPODS, Philadelphia, PA, 12 Oct 1988.

[99] MIL-HDBK-472 (Notice 1), *Maintainability Prediction,* Naval Publications and Forms Center/NPODS, Philadelphia, PA, 12 Jan 1984.

[100] MIL-HDBK-781, *Reliability Test Methods, Plans, and Environments for Engineering Development, Qualification, and Production,* Naval Publications and Forms Center/NPODS, Philadelphia, PA, 14 Jul 1987.

[101] Miller, D. and Lennox, R., "RIPE: A Robot Independent Programming Language," Intelligent Robots and Computer Vision X: Algorithms and Techniques, Proceedings of the SPIE, v. 1607, Boston, MA, 11-13 Nov 1991, pp. 518-529.

[102] MIL-STD-470B, *Maintainability Program for Systems and Equipment,* Naval Publications and Forms Center/NPODS, Philadelphia, PA, 30 May 1989.

[103] MIL-STD-781D, *Reliability Testing for Engineering Development, Qualification and Production,* Naval Publications and Forms Center/NPODS, Philadelphia, PA, 17 Oct 1986.

[104] MIL-STD-785B (Notice 2), *Reliability Program for Systems and Equipment Development and Production,* Naval Publications and Forms Center/NPODS, Philadelphia, PA, 5 Aug 1988.

[105] MIL-STD-883D, *Test Methods and Procedures for Microelectronics,* Naval Publications and Forms Center/NPODS, Philadelphia, PA, 15 Nov 1991.

[106] MIL-STD-1686A , *Electrostatic Discharge Control Program for Protection of Electrical and Electronic Parts, Assemblies, And Equipment (Excluding Electrically Initiated Explosive Devices),* Naval Publications and Forms Center/NPODS, Philadelphia, PA, 8 Aug 1988.

[107] Misra, K., *Reliability Analysis and Prediction: A Methodology Oriented Treatment,* Elsevier, NY, 1992.

[108] Mooring, B. and Pack, T., "Aspects of Robot Repeatability," *Robotica,* v. 5, pt. 3, Jul/Sept 1987, pp. 223-230.

[109] Mroczkowski, R. and Maynard, J., "Estimating the Reliability of Electrical Contacts," *IEEE Transactions on Reliability*, v. 40, n. 5, Dec 1991, pp. 507-512.

[110] Mulazzani, M., "Reliability Versus Safety," Proceedings, 4th IFAC Workshop: Safety of Computer Control Systems 1985, Como, Italy, 1-3 Oct 1985, pp. 141-146.

[111] Munson, G., "Industrial Robots: Reliability, Maintainability, and Safety," *Handbook of Industrial Robotics*, S. Nof, Ed., John Wiley, New York, 1985, pp. 722- 758.

[112] Murray, W., "Space Station Electrical Distribution System Development," 20th Intersociety Energy Conversion Engineering Conference, v. 1, Miami Beach, FL, 18-23 Aug 1985, pp. 1.179-1.196.

[113] Musa, J., "A Theory of Software Reliability and Its Application," *IEEE Transactions on Software Engineering*, v. 1, n. 3, Sept 1975, pp. 312-327.

[114] Neter, J., Wasserman, W., and Kutner, M., *Applied Linear Statistical Models*, 3rd ed., Richard D. Irwin, Inc., Homewood, IL, 1990.

[115] Nettle, J. and Tesar, D., A Roadmap for Standardized Sensor Technology in Modular Reconfigurable Robots, Research Report, Department of Mechanical Engineering, University of Texas at Austin, Austin, TX, 1991.

[116] Pages, A. and Gondran, M., *System Reliability: Evaluation & Prediction in Engineering*, Springer-Verlag, New York, 1986.

[117] Pahl, G. and Beitz, W., *Engineering Design: A Systematic Approach*, The Design Council, Springer-Verlag, London, 1988.

[118] Palmberg, B., Blom, A., and Eggwertz, S., "Probabilistic Damage Tolerance Analysis of Aircraft Structures," in *Probabilistic Fracture Mechanics and Reliability*, J. Provan, Ed., Martinus Nijhoff Publishers, Dordrecht, Netherlands, 1987, pp. 47 - 130.

[119] Paredis, C. and Khosla, P., "On Kinematic Design of Serial Link Manipulators," Intelligent Robotics - Proceedings of the International Symposium on Intelligent Robotics, New Delhi, India, 2-5 Jan 1991, pp. 517-531.

[120] Paul, R., *Robot Manipulators*, MIT Press, Cambridge, MA, 1981.

[121] Pegden, C., Shannon, R., and Sadowski, R., *Introduction to Simulation Using SIMAN*, McGraw-Hill, NY, 1990.

[122] Peterson, M. and Winer, W. (ed.), *Wear Control Handbook*, American Society of Mechanical Engineers, United Engineering Center, New York, 1980.

[123] Price, C., Director, Robotics Group, NASA Johnson Space Center, Houston, TX, Private Communication, June 1992.

[124] Rao, S., *Reliability-Based Design*, McGraw-Hill, New York, 1992.

[125] Rao, S. and Bhatti, P., "Optimization in the Design and Control of Robotic Manipulators: A Survey," *Applied Mechanics Review*, v. 42, n. 4, April 1989, pp. 117-128.

[126] Ramsli, E., "Probability Distribution of Repeatability of Industrial Robots," *International Journal of Robotics Research*, v. 10, n. 1, June 1991, pp. 276-283.

[127] Rinderle, J., "Implications of Function-Form Fabrication Relations on Design Decomposition Strategies," 1986 ASME International Computers in Engineering Conference, Chicago, IL, 20-24 July 1986, pp. 193-198.

[128] Robotics Research Group, "The Development of Operational Software for Enhanced Reliability of Modular Space Robot Systems," Technical Proposal, Mechanical Engineering Dept., University of Texas at Austin, Austin, TX, March 1992.

[129] Rome Air Development Center, *RADC Non-Electronic Reliability Notebook*, RADC-TR-85-194, Rome Air Development Center, Griffiss Air Force Base, NY, October 1985.

[130] Rome Air Development Center, *RADC Reliability Engineer's Toolkit*, Systems Reliability and Engineering Division, Rome Air Development Center, Griffiss Air Force Base, NY, July 1988.

RELIABILITY AND MAINTAINABILITY OF MODULAR ROBOT SYSTEMS: A ROADMAP FOR DESIGN

APPROVED BY DISSERTATION COMMITTEE:

J Wesley Barnes

Davor Juricic

Robert A. Freeman

William D. Lesso

RELIABILITY AND MAINTAINABILITY OF
MODULAR ROBOT SYSTEMS:
A ROADMAP FOR DESIGN

by

DEAN LEROY SCHNEIDER, B.S., M.S.

DISSERTATION

Presented to the Faulty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

August 1993

ACKNOWLEDGMENTS

It is really hard to list all the people I have become indebted to during the accomplishment of this work. First, I must thank my wife, Glenda, for bearing the greatest burden of all. Also, I am grateful for my children, Joshua, Bethany, Raechel, Thomas, and Peter, for their unending love and support.

I would also like to thank the members of my committee, especially Professor Tesar, Professor Barnes, and Major Leahy for their time and patience. I also would like to acknowledge the assistance and council of Col John Borgman as well as his generous support. I am grateful for the support, knowledge, and assistance provided by the members of the University of Texas Robotics Group: Dev Sreevijayan, Cathy Ambrose, Rich Hooper, Gary Browning, Rob Smithson, Chris Pencis, and Bunker Hill, in particular.

This research was supported by the United States Air Force under Educational Services Agreement Number F33600-88-A-0264 and NASA Johnson Space Center OAST Grant Number NAG 9-411.

RELIABILITY AND MAINTAINABILITY OF MODULAR ROBOT SYSTEMS: A ROADMAP FOR DESIGN

Publication No. ___________

Dean Leroy Schneider, Ph.D.
The University of Texas at Austin, 1993

Supervisor: Delbert Tesar

As robotic technology is considered for use in extreme environments, such as on-orbit and planetary exploration missions, the availability of the robotic systems become of paramount concern. Availability has two components: Reliability and Maintainability (R&M). Modular robotic systems address the maintainability portion of availability by minimizing repair time and allowing for the optimal reconfiguration of the robotic system for different tasks. The remaining portion availability is the reliability of the modular robotic system. This dissertation presents a review of robotic system reliability technology and develops a technology roadmap outlining future directions for research and technology application that will improve the reliability of modular robotic systems. Also developed and tested is a design index based upon a modular robotic system's hardware and software reliability and the precision of the system.

The results of the R&M technology review indicate a need to improve modular robotic system reliability by insuring simple, quick, precise, and standard module interfaces; reducing the need for geared actuation, moving toward direct-drive technologies (based on the present state-of-the-art); and the use of high

reliability drive technologies such as a/c servo motors. Additional research recommendations include the development of methodologies to identify and quantify the modular system component life dependency structures; a suggested method using *H*-function integral transform theory is discussed. The establishment of a national or international robotic parts reliability data-base is imperative to allow the types of reliability advancement in robotics as was seen in the electronics industry.

The Reliability Performance Index (RPI) is developed to allow the quantification of the modular robotic system reliability during module design and configuration determination. The RPI was tested on a 3 degree-of-freedom planar example. While not suited for deterministic optimization, the RPI was used to test for statistical significance of the module configurations, allowing the designer to reduce the joint module design space by 70% and to select a statistically best link module combination. Future research recommendations for the RPI are also made.

TABLE OF CONTENTS

[131] Rome Air Development Center, *Non-Electronic P. rts Reliability Data (NPRD) Handbook*, NPRD-91, Rome Air Development Center, Griffiss Air Force Base, NY, 1991.

[132] Rome Air Development Center, *Plastic Microelectronics Cirsuit Packaging: A Technology Review*, CRTA-PEM, The Reliabilty Analysis Center, Rome Air Development Center, Griffiss Air Force Base, NY, March 1992.

[133] Ross, S., *Introduction to Probability Models*, Academic Press, New York, 1989.

[134] Ruocco, S., *Robot Sensors and Transducers*, Open University Press, Milton-Keynes, UK, 1987.

[135] Russel, R., *kobot Tactile Sensing*, Prentice Hall, Engelwood Cliffs, NJ, 1990.

[136] *SAS/STAT User's Guide, Release 6.03*, SAS Institute, Cary, NC, 1988.

[137] Schmitz, D. and Kanade, T., "Design of a Reconfigurable Modular Manipulator System," Proceedings of the Workshop on Space Telerobotics, JPL Pub. 87-13 v. III, 1 June 1987, pp. 171-178.

[138] Schmitz, D., Khosla, P, and Kanade, T., "The CMU Reconfigurable Modular Manipulator System," 19th International Symposium on Industrial Robots, Sydney, Australia, 6-10 Aug 1988, pp. 473-488.

[139] Shigley, J. and Mischke, C., *Mechanical Engineering Design*, McGraw-Hill, New York, 1989.

[140] Shooman, M., *Software Engineering: Design, Reliability, and Management*, McGraw-Hill, New York, 1983.

[141] Shump, D., "Reliability Testing of Electric Motors," *IEEE Transactions on Industry Applications*, v. 25, n. 3, May/June 1989, pp. 386-390.

[142] Siddall, J., *Probabilistic Engineering Design: Principles and Applications*, Marcel Dekker, New York, 1983.

[143] Siljak, D., "Reliable Control using Multiple Control Systems," *International Journal of Control*, v. 31, n. 2, 1980, pp. 303-329.

[144] Sklar, M. and Tesar, D., "Metrology and Calibration Techniques for the Performance Enhancement of Industrial Robots," Research Report, Mechanical Engineering Dept, University of Texas at Austin, Austin, TX, July 1988.

[145] Slotine, J. and Li, W., *Applied Nonlinear Control*, Prentice-Hall, Engelwood Cliffs, NJ, 1991.

[146] Spiteri, C., *Robotics Technology*, Sanders College Publishing, Division of Holt, Rinehart, and Winston, Inc., Philadelphia, PA, 1990.

[147] Springer, M., *The Algebra of Random Variables*, John Wiley and Sons, New York, 1979.

[148] Sreevijayan, D. and Tesar, D., "On the Design of Fault Tolerant Robotic Manipulator Systems," Research Report, Mechanical Engineering Dept., University of Texas at Austin, Austin, TX, June 1992.

[149] Stanton, S. and Tesar, D., "A Technology Roadmap for an Electronic Actuator Control Module for Reconfigurable Robotic Manipulators," Research Report, Mechanical Engineering Dept., University of Texas at Austin, Austin, TX, July 1990.

[150] Sukhija, R. and Rao, A., "Mechanical Error Synthesis of Path Generating Mechanisms using Reliability Index," *Transactions of the Canadian Society for Mechanical Engineers*, v. 10, n. 2, 1986, pp. 85-90.

[151] Sukhija, R. and Rao, A., "Type Synthesis of Four Bar Path Generator through Reliability Index," *Transactions of the Canadian Society for Mechanical Engineers*, v. 12, n. 4, 1988, pp. 199-203.

[152] Sugimoto, N. and Kawaguchi, K, "Fault Tree Analysis of Hazards Created by Robots," Proceedings: 7th International Symposium on Industrial Robots, Chicago, IL, 17-21 April 1983, pp. 9.13-9.28.

[153] Thomas, M. and Tesar, D., "Dynamic Modeling of Serial Manipulator Arms," *Journal of Dynamic Systems, Measurement, and Control*, v.104, Sept, 1982, pp. 218-228.

[154] Thomas, R. and Halliman, G., "Design of the Space Station Freedom Power System," *IEEE Aerospace and Electronics Systems Magazine*, v. 5, n. 1, Jan 1990, pp. 19-24.

[155] Unimation Robotics, *PUMA MARK II Robot: 500 Series Equipment and Programming Manual*, 398P1, Unimation Robotics, Inc., Danbury, CT, 1983.

[156] Upadhyaya, S., Pham, H, and Saluja, K., "Reliability Enhancement by Submodule Redundancy," 1990 Annual Reliability and Maintainability Symposium, Los Angeles, CA, 23-25 Jan 1990, pp. 127-132.

[157] Van Doren, M. and Tesar, D., "Criteria Development to Support Decision-Making Software for Modular, Reconfigurable Robotic Manipulators," Research Report, Mechanical Engineering Department, University of Texas at Austin, Austin, TX, March 1992.

[158] Veillette, R., Medanic, J., and Perkins, W., "Design of Reliable Control Systems," *IEEE Transactions on Automatic Control*, v. 37, n. 3, March 1992, pp. 290-304.

[159] Vinogradov, O., *Introduction to Mechanical Reliability: A Designer's Approach*, Hemisphere Publishing Corporation, New York, 1991.

[160] Visinski, M., Walker, I., and Cavallaro, J., Fault Detection and Fault Tolerance in Robotics, NASA Space Operations, Applications, and Research Symposium, Houston, TX, 1991.

[161] Visinski, M., Walker, I., and Cavallaro, J., "Robotic Fault Tolerance: Algorithms and Architectures," *Robotics and Remote Systems in Hazardous Environments, Vol. 1*, Jamshidi, M. and Eicker, P. (Eds), Prentice-Hall, Engelwood Cliffs, NJ, To be published 1993.

[162] Volz, R., "Report of the Robot Programming Language Working Group: NATO Workshop on Robot Programming Languages," *IEEE Journal of Robotics and Automation*, v. 4, n. 1, Feb 1988, pp. 86-90.

[163] von Newman, J., "Probabilistic Logic and the Synthesis of Reliable Organisms from Unreliable Components," *Automata Studies*, Shannon and McCarthy, Editors, Princeton Univ. Press, Princeton, NJ, 1956, pp. 43-98.

[164] Weaver, W. W. and Deininger, F. W., "Reliability Based Robotic Improvements at TMI-2," American Nuclear Society's 4th Topical Meeting on Robotics and Remote Systems, Albuquerque, MN, 25-27 Feb 1991, pp. 365-373.

[165] Wells, D. and Krishnaswami, K., "Fault Analysis and Recovery Strategies for Deep-Sea Robots," 11th Annual Energy Sources Technical Conference, New Orleans, LA, 10-13 Jan 1988, pp. 39-47.

[166] Wurst, K., "The Conception and Construction of a Modular Robot System," 16th International Symposium on Industrial Robots, Brussels, Belgium, 30 Sept - 2 Oct 1986, pp. 37-44.

[167] Wiggins, M., "New Dell PCs Speed Graphics," *PC World*, v. 11, n.2, Feb 1993, pp. 62-65.

[168] Ernst, D. and Schneider, D., "Cost-Effective Environmental Stress Screening of the F100 Engine Electronic Control," 1989 Environmental Testing and Production Screening Conference, Institute of Environmental Sciences, Baltimore, MD, 14 Jun 1989, pp. 15-21.

[169] Morgan, P., "F100 Quarterly Operational Engine Activity Report: Year Ending Dec 1992," Operational Data Management Section, Pratt & Whitney Report Number F41608-92-2193 3510, Pratt & Whitney, West Palm Beach, FL, Mar 1993.

VITA

Dean Leroy Schneider was born in San Marcos, Texas, on November 12, 1959, the son of Leroy and Alice Schneider of Maxwell, Texas. After completing his high school education at Lockhart High School, Lockhart, Texas in 1978, he attended Texas A&M University in College Station Texas. Upon completing the degree of Bachelor of Science in Electrical Engineering in May 1982, he was commissioned as a Second Lieutenant in the United States Air Force and assigned to the 381st Strategic Missile Wing, McConnell AFB, Wichita, Kansas. During the three year tour, he performed maintenance and troubleshooting tasks beyond the scope of technician capability.

In May of 1985, he entered the Air Force Institute of Technology (AFIT), Wright-Patterson AFB, Dayton, Ohio, as a Master's Degree candidate in Guidance and Control. He was promoted to captain in June 1986. He was awarded a Master of Science in Electrical Engineering Degree in December 1986. His thesis topic was "Digital QFT Flight Control Design as applied to the AFTI/F-16." After attending a specialization class in reliability and maintainability, he was assigned to the San Antonio Air Logistics Center (SA-ALC) at Kelly AFB, San Antonio, Texas as a reliability engineer responsible for the base-wide Environmental Stress Screening and Electrostatic Discharge Control Programs. In 1989, he was reassigned as Assistant to the SA-ALC Chief Engineer and was involved in scientific and engineering policy development and personnel management. He also taught mathematics part-time at the Palo Alto Community College in San Antonio.

Captain Schneider was selected to attend a doctoral program in engineering in 1989 and entered the Graduate School at the University of Texas in August 1990. Upon completion of his Ph.D. in Mechanical Engineering, he will join the AFIT faculty as an assistant professor in Electrical Engineering instructing Air Force officers in reliability, robotics, and control.

Permanent Address: Rt. 1, Box 3, Maxwell, Texas 78656.

This dissertation was typed by the author.

LIST OF TABLES

LIST OF FIGURES