1989

Stochastic model of effectiveness for man-hardware-software system

Bahtiar Saleh Abbas
Iowa State University

Order Number 8920102

Stochastic model of effectiveness for man-hardware-software system

Abbas, Bahtiar Saleh, Ph.D.

Iowa State University, 1989

U·M·I
300 N. Zeeb Rd.
Ann Arbor, MI 48106

Stochastic model of effectiveness for man-hardware-software system

by

Bahtiar Saleh Abbas

A Dissertation Submitted to the
Graduate Faculty in Partial Fulfillment of the
Requirements for the Degree of
DOCTOR OF PHILOSOPHY

Major: Industrial Engineering

Approved:

Signature was redacted for privacy.

In Charge of Major Work

Signature was redacted for privacy.

For the Major Department

Signature was redacted for privacy.

For the Graduate College

Iowa State University
Ames, Iowa
1989

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 INTRODUCTION

A major concern in system analysis is with the lifetime operation of the system so that it can fulfill its required mission successfully. To accomplish this purpose, the minimum requirements for some measures of system effectiveness have always been imposed in the design and manufacturing stages of the system. Some requirements to be considered are hardware subsystem, software subsystem, and human (operator) performance. Therefore, the selection of the most appropriate measure of system effectiveness based on the operational or system performance as it relates to all of the system components becomes important [55].

1.1 Definitions of System Effectiveness

System effectiveness defined in ARINC [2] is "the probability that the system can successfully meet a specified condition." Another famous and widely used definition of system effectiveness is from MIL-STD-721B [14], in which system effectiveness is defined as "a measure of the degree to which an item can be expected to achieve a set of specified mission requirements, which can be expressed as a function of availability, dependability, and capability." In fact, as suggested by Kuo [55], system effectiveness is a measure describing the overall capability of a system to accomplish its intended mission. He defined system effectiveness as "the probability measure of the achievement of a specific mission goal." This definition of system effectiveness has been adopted for the current study. Under this definition, we have two key submeasures: 1) how ready the system is to function (availability), 2) how well it performs (reliability). Thus, system effectiveness is the joint probability measure of availability and reliability. Before adopting this definition, we must explicitly define what we mean by failure, availability, and reliability. We will adopt the definitions used in [107].

1.2 Definition of Failure

There are many types and degrees of failure; one definition of failure stated in MIL-STD-721B [14] is "the inability of an item (hardware or human factor) to perform within previously specified limits." Regarding software, Lipow [64] defines failure as a situation in which a computer program fault is elicited by some kind of input data and which leads to the computer's incorrectly computing the specified function. These are generally acceptable definitions; however, they do not address the degree of failure: that is, is the system totally incapacitated or is it simply functioning at a lower level of performance? In addition, there are failures that are independent of other components and there are those that are dependent, having been caused by other component failures or causing the failure of other equipment. Again this delineation is not treated. In this study, we will combine the above definition stated by MIL-STD-721B [14] with the consideration of the degree of failure. In other words, the possibility that the system under investigation is performing under a lower level of operating condition will be incorporated in the proposed model.

1.3 Definition of Availability

In general, availability is the measure of readiness of an item to be put into service when called upon. Some authors have classified availability as follows [61]: 1) pointwise availability: the probability that the system is operational at any random time t; 2) average uptime availability: the proportion of time in a specified interval that the system is available for use; and 3) steady-state availability: the average uptime availability when the time interval considered is very large. The appropriateness of the availability representation depends upon the system mission and the operating conditions. Steady-state availability may be a satisfactory measure for systems that are operated continuously. Average uptime availability may be the most satisfactory measure for systems whose usage is defined by a duty cycle. For systems required to perform a function at any random time, pointwise availability may be the most satisfactory measure.

1.4 Definition of Reliability

Generally speaking, reliability is the probability of an item performing its function for the period of time intended under the operational conditions encountered. Under this definition, there are two aspects of reliability: whether the equipment operates as designed and whether it achieves the desired results. The attributes of reliability for hardware [55] are utilization and environmental effect; for operator [55] are human reliability, selection, experience, motivation, working environment, training and discipline, human engineering, performance, capacity, and environmental effect. Analogous to hardware, the attributes for software are utilization and environmental effect.

1.5 Human Performance

Human performance plays an important role in the overall reliability of engineering systems because most systems are interconnected with human links and much research and many publications address human performance in systems (see Ref. [58]). According to studies quoted by this reference, depending upon the degree of human involvement in the system, the human component is responsible for 20%-90% of the failures in many systems. This means that human performance must be considered in reliability analysis, in order to obtain a more realistic picture of system reliability.

The human reliability aspect, according to Dhillon [15], can be improved significantly by following human-factor principles during the system-design phase. He also pointed out that, on the other hand, factors such as careful selection and training of concerned personnel also help to increase human reliability. One important area that affects human performance and its reliability is stress. An over-stressed person will obviously have a higher probability of making errors. In order to minimize the occurrence of human errors, operator limitations or characteristics must be considered during the design phase by the design, and reliability engineers. The consequence of human error may vary from one set of equipment to another or one task to another. Furthermore, consequences may range from minor to severe, from delay in system performance to loss of life.

1.6 Objective of This Study

Reliability, availability and dependability have been widely studied as measures of system effectiveness. Combinations of some of these have also been successfully modeled as measures of effectiveness for a specific system. Furthermore, in the last few years, additional factors such as environmental and operational effects have been incorporated in the model. Most of the models proposed, however, are "single shot" models, in the sense that they are required only to handle one task during mission time. In addition, few of the models incorporate performance levels into measuring system effectiveness.

Traditionally, besides the operator, a system involves only the hardware unit. However, in some cases, systems are becoming so large and complex and time constraints so tight that systems operation is possible only through the extensive use of the computer. The issues of computer performance evaluation and prediction have concerned designers and users of such systems. Until the late 1960s attention was focused almost solely on the performance of the hardware aspect of the system. Performance of the system in the operational phase, however, depends on both the hardware and the software subsystems. In the early 1970s, software became the center of attention. This happened due to the continuing increase in the ratio of software to hardware costs, in both the production and the operational phases [28].

Very few studies have addressed the problem of modeling and evaluating the failure and maintenance phenomena in the hardware-software system. The situation becomes more complex if the human operator is involved, and there is no study addressing the system effectiveness of such systems. The objective of this study, therefore, is to develop stochastic models of effectiveness for a system involving hardware, software, and human operator and which is required to perform a number of randomly arriving tasks during the mission time. This study is a sequel to three previous studies: "System Effectiveness Models for Maintained System: Analytical and Simulation Approach" by Lie [60], "System Effectiveness Models Via Renewal Theory and Bayesian Inference" by Kuo [55], and "Stochastic Models for System Effectiveness" by Lee [57].

1.7 Contents of This Study

Chapter 2 provides a review of existing approaches to hardware reliability, human reliability, and software reliability. Finally, this chapter reviews the better-known effectiveness models for hardware, man-hardware, and hardware-software systems.

Chapter 3 develops the analytical model of the effectiveness for a single machine system operated by a human operator. The machine involves hardware and software. The system-effectiveness model developed in this study involves five major factors: the task arrival process, the system state, the allowable performance time, the system design failure, and the human operator performance variables. Three formulations of system-effectiveness are given; each formulation is different in terms of how the case of no task during the mission time is handled.

Chapter 4 develops the effectiveness model for N-machine systems. Each machine involves hardware and software and is operated by a human operator. The system is required to perform a number of tasks that arrive randomly during the mission time. The human operator of each machine has to perform a prescribed function simultaneously with the operators at all other machines at each task arrival.

Chapter 5 discusses three extensions to models proposed in Chapters 3 and 4. The models are extended to handle: 1) systems with multiple operating modes, 2) systems with several types of tasks, and 3) systems with general failure and repair distribution. For the first problem, the operating levels are assumed to affect the task performance. In the second problem, each type of task is characterized by the performance level of the human operator representing the degree of accomplishment of a specified task. The last extension is considered to handle a more general system by removing the assumption of the Markovian process for the system state.

2 LITERATURE REVIEW

2.1 Hardware Reliability

This section reviews some fundamental aspects of equipment reliability [8,10]. Before we begin the discussion, first we will explain the terms *component* and *equipment*. The component is a nonmaintained integral item. If a component fails, it is removed and discarded from the population under consideration. It may or may not be replaced by a new component. It should be noted that this definition does not exclude a failed and discarded component being salvaged and repaired to original standards and returned to service, but it then constitutes a new component. The words *equipment* and *hardware* are used interchangeably to denote an assembly of components.

Now the reliability of a component can be defined as its ability to function successfully as required under specified condition. It is measured as a probability to function without remedial action to a specified standard, and is dependent upon time and phase of use. This definition contains two essential features. They are: 1) a quality of performance is expected, and 2) it is expected over a specified time. The first feature is related to the strength of the component. It is widely assumed that the strength of the equipment is independent of time, and if the stress due to the load is less than the strength, the component survives; but if it is greater, the component fails. In the absence of adequate evidence on actual distribution, the normal distribution is probably the best assumption for the strength and load [8]. The load is repeated n times from its distribution, with n being some function of time. In this way, time is introduced into the analysis. It has been shown [8] that if time is taken into account in this way, reliability drops very rapidly early in the operation and afterward remains very nearly constant. It has also been shown that this situation is also true in terms of failure rates. That is, the failure rate initially drops very rapidly, subsequently remaining more or less constant.

The explanation so far has neglected all time-dependent factors such as creep, fatigue, corrosion, erosion, and so on. Ideally, these could be taken into account by making strength a function of time and load. Several mathematical models have been built up along this line. Ideally, we would also wish that any piece of equipment were 100% reliable–so far as time-dependent factors are concerned–up to a given time and that subsequently they would become become 0% reliable. In practice, we could hope that the fall of reliability would be distributed over a time period. We may suppose that this could be more or less normally distributed. With this assumption, a failure rate related to wear-out increases with time in contrast to a failure rate that decreases with time in the early life and which remains constant for random failures. Concerning wear-out, Carter [8] stated the following:

> Wear out may physically be a phenomenon corresponding to the colloquial use of the term in which material is worn away so that clearances or stresses become too great for satisfactory use. It may be due additionally to physical or chemical deterioration, to aging, or to corrosion, and so forth. Alternatively, wear-out may be due to fatigue of the mate-

rial leading to failure, or to creep leading to loss of clearance or possibly to rupture. The term "wear-out" thus embraces a wide variety of phenomena, the common feature being a reduction of the strength of the component with time.

It is reasonable to expect that several types of failure may occur in a component. If these are all independent so that prior to actual failure there is no interaction of one failure process on another, and if the occurrence of a single failure mode implies the total failure of the item, the overall failure rate is the arithmetic sum of all the individual failure rates. Hence the failure rate derived for nontime-dependent phenomena may be added to those derived for time-dependent phenomena, in order to give the complete failure rate over the whole life of an item of equipment. In this total failure pattern, one can easily distinguish the three broad components: early life represented by a decreasing failure rate with respect to time; chance or random failures represented by a more or less constant rate; and wear-out, represented by an increasing failure rate.

So far the discussion has concentrated on only the components. But what about the reliability of the equipment? Carter [8] has this to say about the topic:

> In practice, failed components of complex equipment are replaced, and the equipment returned to service. The population thus remains constant, and we eventually reach a state where the population is made up of a mixture of first, second, third, or even later-generation components. The higher generations are most likely to be present if component failure due to wear-out and the mean component life is only a small fraction of the parent equipment life. If the life of the parent equipment is infinitely

long, it is readily seen that we should ultimately achieve a steady-state where the mixture of the generations is so great that we have, in fact, random failure. In this case, the failure rate would be constant. However, in many cases, the average life of the parent equipment is only two or three times the average life of the component, and neither the generation failure pattern nor the steady-state pattern can be expected to represent the situation adequately. Of more practical engineering interest, however, were the results of some calculations which showed that with real complex equipment, the steady-state is reached very rapidly.

In general, the replacement process applies to minor components, while the parent equipment would be subject to normal wear-out. One piece of equipment may suffer a number of minor failures due to some causes that could all be lumped together to give a random failure pattern. The failure rate could be deduced from experience and, according to the steady-state approach, could be assumed to hold in the feature with appropriate maintenance. In addition to this, the equipment itself would be susceptible to some long-term failure modes, due to some other causes which would not have been experienced in early running. Such long-term failures would appear as major wear-out. Thus, taking the whole life into account, we should expect for maintained equipment much the same failure pattern as for simple nonmaintained components. Early life would exhibit the usual falling failure rate characteristic leading to periods of more or less constant failure rate associated with the normal life of the equipment, and would be followed by a final stage of increasing failure rate corresponding to the wear-out phase at the end of life. This pattern is diagrammatically presented by a "bathtub curve" shown in Fig. 2.1. Despite the

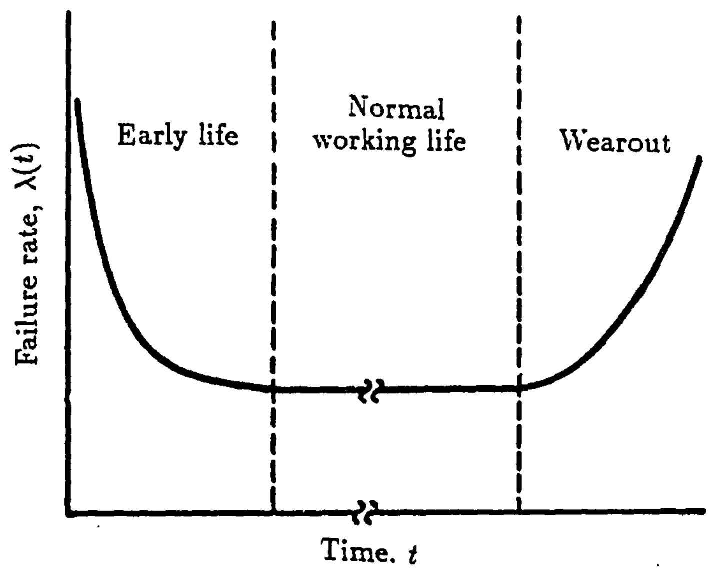

Figure 2.1: Complete failure pattern for equipment

difference in their load and strength nature, this pattern can be found either in mechanical or electrical equipment [8,10]. The first phase is often hidden from the customer since it occurred during testing and commissioning at the manufacturer's factory. It is caused by minor errors in the assembly of the equipment, imperfect joints, or a few sub-standard component not detected by previous testing. This phase is called "burn-in" for electrical equipment and "running-in" for mechanical equipment.

2.2 Human Reliability

Due to the need for the human to interact with equipment and complex systems, it has become necessary to extend or modify classical reliability methods in order to assess the various system-related risks. The potential impact of the integrated reliability assessment is more far-reaching than that which is restricted to mechanical components. The various techniques and approaches that have been offered for dealing with one or more aspects of this problem have gradually been and are still in the process of developing to the area of human reliability analysis. Dhillon [15] defined human reliability as the probability of accomplishing a job or task successfully by a human at any required stage in a system operation within a specified minimum time limit (if the time limit is specified).

Sharit [91] critically reviewed most existing approaches to human reliability analysis from the standpoints of both their utility and validity. Broadly speaking, he classified them into: 1) the technique for human error prediction (THERP), 2) the use of qualitative models of human performance, 3) the simulation method, and 4) methods borrowing heavily from classical mathematical reliability techniques.

THERP is generally associated with Alan Swain [101,102]. They are supported by Miester [66] and Embrey [22]. The method is presented in details in [103]. This technique is primarily used to evaluate system-degradation resulting from human error in association with factors such as system characteristics influencing human behavior, operational procedures, and the reliability of the equipment. As noted by Sharit [91], the technique reflects the belief that only through quantification can reduced system reliability be attributed to equipment and/or procedural design and increased system reliability stemming from the application of ergonomic principles can be accomplished. THERP involves the four steps shown in Fig. 2.2. In the first step, the analyst identifies performance-shaping factors (PSFs) associated with information unique to the system under the study. PSFs can affect the probabilities

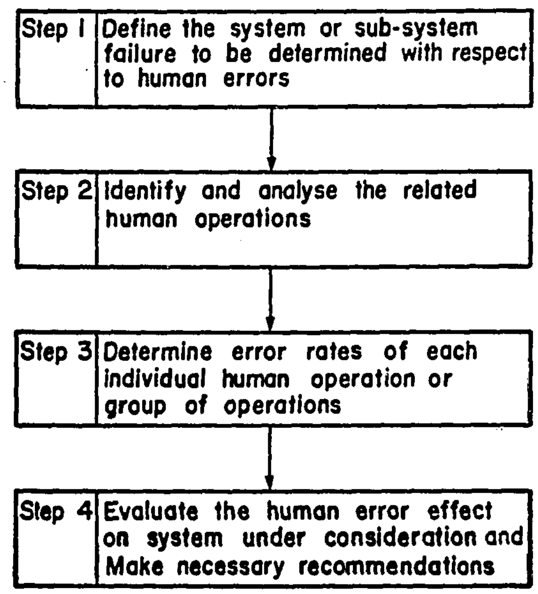

Figure 2.2: Steps associated with THERP

of human error and include factors such as task and equipment requirements, job and task instructions, stress, level and type of training, and ergonomic design issues. The second phase involves primarily task analysis [20], where the operators' actions are identified and broken into tasks and subtasks. The development of a model at this stage allows PSFs to be more accurately represented. At the third step, quantitative assessment and probabilistics methods are applied. The basic index of human performance is represented by human error probabilities (HEPs), which include incorrect performance of an action when required, as well as the probability that the task will not be completed correctly within some specified time interval. This information is typically used in combination with expert judgment, where the similarities and differences between the tasks are judged in order to determine how error probabilities should be adjusted; informal expert opinion is also used some-time. In the last step, sensitivity analysis is considered. There is no formal approach to this analysis; any approach that allows the evaluation of various assumptions related to HEP serves this purpose. As the final component of the fourth phase, the results of the human reliability analysis are combined with other components of the system, through either fault tree.

Qualitative approaches to human reliability analysis, according to Sharit [91], do not necessarily serve as alternatives to quantitative approaches, but rather tend to represent a set of loosely-organized ideas that advocate understanding the types of errors humans perform and the mechanisms underlying these errors. The approach is discussed in Norman [78], Carnino and Griffin [7], Rouse and Rouse [88] and Reason [86] and has been applied to industrial system reliability by Rasmussen. For example, see [83,84,85].

The application of digital simulation techniques to human reliability analysis has been associated primarily with Siegel and his coworkers [96,95]. The computer simulation technique utilized by Siegel and Lautman [94] attempted to model crews on surface ships consisting of teams of 4-20 members, for the purpose of generating systems reliability and system availability information based on integrated human and equipment performance. Task analysis, along with information on equipment, personnel, etc., provided the input data according to the computer model's logic. The model simulates the attributes of individuals and the equipment they operate. Variables representing the physical capability and physical workload requirements are relatively easy rationalized. Values representing the levels of parameters such as aspiration, fatigue, stress, and motion are, however, more difficult to drive [91]. Nevertheless, with sufficient empirical data, analyst can generate the precise distribution from which individual values can be sampled [94]. Compared to THERP, simulation methods are more powerful [67]. One reason is that human performance can be produced over a number of trials, theoretically allowing for the quantification of the distributions surrounding the performance estimates, in contrast to THERP, which assumes distributional properties a priori. A more compelling property of the simulation technique, however, is its ability to handle the complex interactions of a large number of variables.

Probabilistic risk-assessment tools, such as the fault-tree method are more easily adaptable to static than to dynamic reliability. For the latter case, especially in the case of tasks in the continuous time domain, such as vigilance, monitoring, and tracking, it would seem reasonable to approach human reliability analysis in accordance with classical reliability theory [50]. Using this approach, the prediction of human reliability is obtained directly from the probabilistic model derived for the human performance under study. Most models presented in the literature assume a constant rate for the human error [3,4,15,16,19].

2.3 Software Reliability

Software is now part of a very wide range of products and systems, and this trend is accelerating with the opportunities presented by low-cost microprocessor devices. In most cases, the fact that computer programs take over functions previously performed by hardware results in enhanced reliability, since software does not fail in the way that hardware does. As a results of these rapid technological advances, there has been growing concern that the system problems have transitioned from hardware to software [53]. This has become manifest in 1) the larger system cost being borne by software development and use, 2) the increased schedule delays in system development and production due to software problems, and 3) the reduced overall reliability in the field, due to the fact that many system software errors are only detected after the system has been put into use. As a result, a new branch of reliability called "software reliability" has emerged in the past decade.

Similar to the definition of hardware reliability, time-domain software reliability is defined as the probability of the failure-free operation of software for a specified period of time under specified conditions [93]. However, there are some differences between hardware reliability and software reliability and these are listed in Table 2.1. Software is a collection of instructions written in computer languages. It is also called a computer program, or simply a program. Upon execution of a program, an input state is translated into an output state. Any program is designed to perform some specified functions. When actual output deviates from the expected output, a "failure" occurs. Incorrect logic, incorrect instruction, or inadequate instructions, which when executed cause a failure, are called "faults." Whenever a failure occurs, there must be a corresponding fault in the program, but the existence of faults may not cause the program to fail. A program will never fail as long as the faulty statements are not executed.

Much effort have been made in developing software reliability models. In general, Lin [63] classified them by either the deterministic model or the probabilistic model. The deterministic model studies 1) the elements of a program by counting the number of operators, operands, and instructions, 2) the control flow of a program by counting the branches and tracing the execution paths, 3) the data flow of a program by studying the data sharing and data passing, and 4) other deterministic

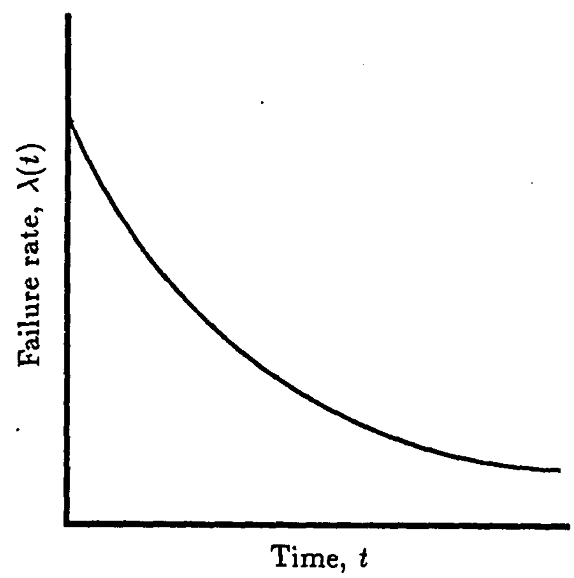

Figure 2.3: A typical reliability growth model

properties of a program.

Performance measures of the deterministic model are obtained by analyzing the program texture and do not involve any random event. In general, these models empirically measure the qualitative attributes of software and are used in the early phases of the software life cycle, in order to predict the number of errors in a program or are used in the maintenance phase for assessing and controlling the quality of a software [63].

The probabilistic model represents failure occurrences and fault removal as probabilistic events. All of the time-domain models belong to this category, such as the reliability growth model, the curve fitting model, the failure rate model, the nonhomogeneous Poisson process model, and the Markov chain model.

The reliability growth model measures and predicts the improvement of reliability through the debugging process (see Fig. 2.3). A growth function is used to represent the progress. The commonly-used independent variable of the growth function is time, and the independent variables can be reliability, failure rate, or cumulative number of errors detected. Several authors proposed the models; inspired by Duane [21], Countinho [13] plotted failure rate versus cumulative hours on log-log paper to represent the software testing process; Wall and Ferguson [114] proposed the Weibull growth model for predicting the failure rate of software during testing; adapted from hardware reliability, Wagoner [113] used a Weibull distribution to represent time between program failures; Yamada and Osaki [121] suggested the used of a logistic growth-curve model to represent the cumulative number of errors up to a certain time; Nathan [74] adapted the Gompertz model to represent the cumulative number of errors corrected up to a certain time; and Sukert [98] adapted the hyperbolic reliability growth model to represent the debugging process of software.

The curve-fitting model finds a functional relationship between dependent and independent variables. For example, using time as the independent variable and failure rate as the dependent variable, this model can be used to estimate the failure rate of a software. By assuming that the failure rate is monotonically nonincreasing, Gubitz and Ott [31] and Miller and Sofer [68] proposed a nonparametric estimation of software failure rate through isotonic regression. By using an exponential regression analysis, Butner and Iyer [6] studied the relationship between the failure rate of software in the operational phase and the system load using an exponential regression analysis.

Failure-rate models study the failure-rate changes at failure time and the functional forms of the failure rate during the failure intervals. Most of failure-rate models belong to the binomial type model; i.e., it is assumed that the program contains N initial faults and each fault has the same chance of occurring. The Jelinski and Moranda De-Eutrophication (J-M) model which is the earliest software reliability model is also one of the simplest failure rates model [49]. In this model each fault is assumed to have a constant failure rate. The J-M model was modified by Moranda [72] by assuming that the program failure rate decreases geometrically at failure time and was extended by Moranda [71] to incorporate the change of program size during debugging process in the original J-M model. There are some other failure rate models, but they are either variants or extensions of the original J-M model. See, for example, Littlewood [65], Schick and Wolverton [89], Sukert [98], Goel and Okumoto [25], and Min Xie [69].

The nonhomogeneous Poisson Process (NHPP) model represents the number of failures experienced up to a certain time. The main issue in the NHPP model is determining an appropriate mean value function to denote the expected number of failures up to a certain time point [63]. One simple class of NHHP models is the exponential mean value function model, which has an exponential growth of the cumulative number of failures experienced. The models were proposed by Musa [73] and Goel and Okumoto [26]. An extension of the exponential mean-value function model has been suggested by Yamada and Osaki [121] by assuming that faults come from different sources with different failure rate. Other types of mean value functions are the S-shaped models suggested by Ohba [79] and Yamada [120] and the hyper-exponential model suggested by Ohba [79]. The S-shape is proposed based on the belief that in the early stages of debugging, as faults are detected, more dependent faults become detectable. This results in an increasing growth rate. As undetected faults decrease, the growth rate slows down gradually and finally approaches zero. The hyper-exponential growth model is based on the assumption that a program has a number of cluster modules, each module having a different initial number of errors and a different failure rate.

The Markov model is a general way of representing the software failure process. The number of remaining faults is modeled as a stochastic counting process. When a continuous time, discrete-state Markov chain is adapted, the state of the process is the number of remaining faults, and time-between-failure is the sojourning time from one state to another. A general Markov process allows transitions to occur from any state to any other state. In other words, multiple faults can be removed or introduced during debugging, as suggested by Sumita and Shantikumar [100]. If we assume that the failure rate of the program is proportional to the number of remaining faults, the linear death process and the linear birth-and-death process are two models readily available. The former assumes that the remaining error is monotonically nonincreasing. The latter allows faults to be introduced during debugging. When a nonstationary Markov model is considered, the model becomes very rich and unifies many of the proposed models. The nonstationary failure rate property can also simulate the assumption of the nonidentical failure rate of each fault. Examples of the Markov models are the linear death model with perfect debugging suggested by Jelinski and Moranda [49], the linear death model with imperfect debugging suggested by Goel and Okumoto [27], the nonstationary linear death model with perfect debugging suggested by Shantikumar [90], and the nonstationary linear birth-and-death model proposed by Kuo [56] and Kremer [54].

Most models mentioned above are concerned with the effects of the program structure and with the programming practices of the development and test environment. From the user point of view, however, the software is treated as a "black box." Hecht and Hecht [44] argued that many conventional software reliability models are not valid during the operational phase. Supported by the real data, they concluded that:

> ...failure rate or outage is constant or very slowly decreasing. It is not intended to imply that a decreasing failure rate is never seen in an operational environment. It is intended to show that a decreasing failure rate is not necessary characteristic of software in the operational environment, and hence that software reliability in that environment may differ substantially from those in the debug and test phases.

In the operational phase, it is more appropriate to define software reliability based on input space, as opposed to time space. It is the input that triggers the software error, instead of run time. In the input space model, software reliability is defined as the probability of successful run(s) randomly from the input space. For more discussion, see [77,82,118].

2.4 Models for System Effectiveness

2.4.1 Hardware System

Hosford [47] may be the first one to use the measure of dependability to evaluate system effectiveness in any system where failure is possible. His work was extended by Finkelstein and Schafer [23], who discussed the dependability models for a parallel system; and by Mohan, Garg and Singal [70], who discussed the dependability models for a complex system.

Katz, Jaffe and Rosenthal [52] defined an index of system effectiveness, called system worth, which combines the effect of both reliability and accuracy. According to these authors, accuracies are statistical in nature and are expressed by a bivariate normal distribution. Based on this distribution, the authors obtained accuracy probability and further obtained the system worth as the system-effectiveness measure for the B-58 bombing-navigation system.

Coleman and Abram [11] introduced steady-state availability as a measure of system effectiveness and called it operational readiness.

Henry [45] suggested improving effectiveness through the availability model. To realize a significant improvement, it is necessary to concentrate on the reliability, maintainability, logistic, and operational problems which will culminate with a significant payoff when improved.

ARINC [2] suggests that system effectiveness includes reliability, operational readiness, and design adequacy, and this concept was applied to the radar systems and multi-moded system.

Goldman and Slattery [29] presented a diagrammed system effectiveness model which included capabilities, operational readiness, and constraints as the attributes of system effectiveness.

Karmiol, Weir, and Youtcheff [51] presented a simplified system-effectiveness model to the re-entry vehicle system; the model strongly considered the interrelationship between reliability and availability.

The Weapon System Effectiveness Industry Advisory Committee (WSEIAC) for the U.S. Air Force System Command [115,116] established a system-effectiveness model that is a joint-probability measure of the availability of the system, its dependability, and its capability.

Hayward [43] has proposed a quantitative measure of the combat-effectiveness of a military force. The effectiveness depends not only on the capability of the specified force, but also on the nature of the enemy, the environment, and the mission.

Winokur and Goldstein [119] presented their analysis of mission-oriented systems. A multi-phase system, which performed a mission at various times during its lifetime, was composed of a predetermined number of subsystems, each of which were required to perform one or more specific missions. The attributes considered by Winokur and Goldstein were reliability, capability, and availability.

The U.S. Army [81] is concerned with design and lifetime operation of a military system so that it can fulfill its mission. They impose reliability, availability, and maintainability (RAM) requirements for military equipment on the contractor. RAM requirements, however, are quite often not met in the test and operational phases of the system. This is partly due to the fact that the RAM requirements often consider the hardware component. Also important are logistics, human operator performance, and environmental effects during the mission.

Tillman, Lie, and Hwang [109] introduced pseudo-reliability, which is a combined measure of reliability and the level of performance. Whenever the level of performance is of primary concern, it appears to be a more practical measure of the system than reliability alone. The concept of pseudo-reliability has been demonstrated for a combat tank system [109]. The work of Tillman *et al.* is the first to incorporate performance levels into the measurement of system effectiveness.

Gonzales-Vega, Foster, and Hogg [30] presented SIMULAV, a simulation program capable of modeling large-scale reliability systems. The program can model the effect of such logistics characteristics as inventory, transportation, and facilities on the reliability-availability of the system. The system here consisted of several components, all of which could have different failure modes. Each failure mode followed a particular failure-distribution function.

2.4.2 Human-Hardware System

In most situations, a system is the linking of a human operator and a machine. However, all of the models mentioned so far do not consider human-operator effects which, in practice, are believed to have a significant impact on system effectiveness. The human operator has so far been assumed to be fully reliable, and no provisions have been included in the models proposed to account for human-operator effects. In practices, however, a large proportion of incidents and malfunctions reported are typically assigned to *human error* or *human reliability*. A case in point is military systems. Even though the reliability of the hardware components of most military systems is high, poor system effectiveness is usually observed in the field due to the significant impact of human error [55].

Having been frustrated with the inaccuracy of their predicted measure of system effectiveness, and having realized that the performance of human operators has a definite impact on system effectiveness, system reliability engineers have introduced the human-operator effect into their already-developed models and have proposed integrated-effectiveness model measures of man-machine systems. Recognition of importance of human operators, since then, has been growing steadily, especially in military-sponsored research and development [58].

Topmiller [112] differentiated three major criteria of system-effectiveness: reliability, availability, and maintainability (RAM). Here the human factor was treated as a basic problem relating human performance to the major system effectiveness parameters of RAM. The mathematical property of "additivity of variance" was used to evaluate systematically the human operator's contribution to system effectiveness.

Leuba [59] indicated that the role of humans in systems is mathematically similar to the role of hardware in those systems. Therefore, he argued that the models for assessing their role in system effectiveness are already available in the current system-effectiveness models like the ARINC's model [5].

Gephart and Balachandran [24] modified the effectiveness model of WSEIAC to include human-performance measures. They introduced the human element by considering the capacity in the WSEIAC model as the product of adequacy of personnel and capability of hardware.

The U.S. Navy model [75] considers both operator and hardware effects in each of the three major attributes defined in the WSEIAC model: availability, dependability, and capability. However, it does not present any clear method for combining various performance measure from hardware and human operators, except by assuming the stochastics independence between these two.

Siegel and Lautman [95] developed a family of computer models that sequentially simulated the actions and behavior of the operators and maintainers in a man-machine system as they accomplished the tasks involved in mission performance. Here, the behavior of humans in the system was affected by factors such as stress, fatigue, proficiency, aspiration, learning, morale, competence, and physical capability. Human performance was determined as a function of a function of the above factors.

Tillman, Lie, and Hwang [110] presented simulation models; and not only were the effect on hardware components considered, but the environmental effect and human performance factors such as operators' training phase were as well. Under the assumption of statistical independence among these element, system effectiveness was determined by the product of each element. More discussion on the model is presented in [60].

The system-effectiveness models introduced so far indicated the complexity of the relevant attributes, but few of the models have given a comprehensive description. Even if the description is given, there is still a lack of theoretical discussion surrounding the issue, and the estimate given may not not even be the probability estimate [55]. Recognizing this, several authors have established theoretical system-effectiveness models.

Kuo [55], developed system-effectiveness models via renewal theory. Here, system effectiveness was calculated as the product of the availability function evaluated at a task arrival-time t; and the reliability function at time t evaluated for a fixed time period. Further discussion on this can be found in [111]. The general solution found, however, is difficult to evaluate numerically. For this reason, a numerical solution to the general model of system effectiveness was proposed [107,108]. The numerical approach is very general and can be applied to empirical data without assuming a distribution for the data.

Lie, Kuo, Tillman, and Hwang [62] extended the model proposed by Kuo [55] by developing a model of effectiveness for a system that was required to carry out several types of missions. Each mission type was characterized by a maximum allowable duration time determining its success. Here, system effectiveness was determined by the following four factors: availability at the start of the mission, system reliability, effect of the environments, and the effect of the operator. The model was illustrated by a numerical example taken from a test of military weapons systems.

Lee [57] extended Kuo's work, which considered only one task arrival during the mission time. Lee presented a theoretical system-effectiveness model for a single-unit system that was required to perform a number of randomly-arriving tasks during the mission time. In order to achieve mission success, the system had to be available (availability) at each task arrival-time and, if a significant amount of time was required to complete a task, operative at least a period of time (reliability) from each task-arrival time. Consequently, system effectiveness was defined as the combined measure of availability and reliability at each task-arrival time. At each task-arrival time, transient human operator behavior was considered in conjunction with the hardware-system state. Lee also proposed models that incorporated performance levels into measuring system effectiveness and related these levels of performance to human-operator aspects. However, the models only handled the cases where the task was instantly performed when it arrived, and the effect of the degraded system on reliability was not addressed.

Cothier and Levis [12] proposed a method for evaluating measures of effectiveness, by using the command, control, and communication (C^3) system approach. The method was illustrated through application to an idealized fire-support system. In the scenario of the system, the forward observer (FO) receives the initial stimulus by detecting an enemy threat. The FO communicated estimates of the position and velocity of the target and requested for fire to the battalion computer. The mission requirements were expressed as the minimum acceptable probability that the system would be defended successfully. The effectiveness of the system was defined by how well the overall kill probability met the mission requirement.

Applying classical reliability to human performance, Gupta and Gupta [32], Gupta and Kumar [33,34,35,36], Gupta and Sharma [37,38,39,40], Chung [9], and Dhillon and Rayapati [17,18] developed the methods for estimating availability, reliability, and MTTF for various redundant systems under hardware and human failures. The failure and repair times for the systems followed exponential and general distribution, respectively. Laplace-transforms of the various state probabilities were derived and steady-state behavior of the system were examined. Availability at any time was obtained by the inversion process.

2.4.3 Hardware-Software System

Haynes and Thompson [41,42] and Thompson and Chelson [105] suggested the use of statistical methods in the specification and analysis of the reliability and the availability of hardware-software systems. Total system-reliability was defined as the probabilities of the absence of any system malfunction over a given time. System malfunctions were identified as being related to hardware, to computer software, or to unknown sources. It was assumed that the three types of system malfunctions defined three mutually independent point-processes and that their superposition generated the total process of occurrence of system malfunctions. A Bayesian procedure was used to obtain an exact expression for the probability-density function for system reliability.

Goel and Soenjoto [28] and Angus and James [1] developed a Markov chain model to examine the performance of a hardware-software system as a function of hardware-software failure and maintenance rates. Their model required that all underlying distributions be exponential, that at most one software error be removed at correction time, and that no new software errors be introduced during the error-correction phase. This model was extended by Sumita and Masuda [99] to handle the system in a more general context. They suggested a stochastic model describing a hardware-software system having a nonexponential distribution. Moreover, in their model, multiple errors were allowed to be introduced during the repair phase.

Stark [97] developed a methodology for predicting the dependability (reliability and availability) of an integrated realtime hardware-software system using a semi-Markov process. The methodology was used to evaluate the reliability of the shuttle-mission simulators at NASA Johnson Space Center. Although each system had many component, he classified its interactive operations into six general states: good, hardware degraded, software degraded, multiple degraded, hardware critical, or software critical. He assumed each state communicated directly with at least one other state, and at least indirectly with all other states. Availability and reliability were obtained based on the transition matrix representing the system.

Iskander and Nutter [48] developed a methodology to evaluate safety and reliability of electrical mine-monitoring systems. The approach used divided the system into smaller subsystems, i.e., hardware subsystems and software subsystems. They performed a functional analysis and drew a functional block diagram for each subsystem. Finally, a detailed subsystem hazard analysis was performed in detail on each subsystem and followed by the construction and analysis of a fault tree.

Table 2.1: Hardware reliability versus software reliability

Hardware	Software
Failures are caused by material deterioration, random failures, design errors, misuse, and environmental factors.	Failures are caused by incorrect logic, incorrect statements, or incorrect input data. This is similar to the design errors of the hardware system.
Sometimes warning is available before failure occurs.	Software failures occur without warning.
Repairs can be made that might make the equipment more reliable.	The only repair possible is through reprogramming, which, if it removes the error and introduce no others, will result in higher reliability.
Failure rates can be decreasing, constant, or increasing with respect to operating time.	Without considering program evolution, failure rate is statistically nonincreasing.
Failure can be related to the passage of operating (or storage) time.	Failures occur when an erroneous program step or path is executed.
Calendar time is a universally accepted index for the reliability function.	CPU time and "run" are two popular indices for reliability.
Reliability can sometimes be improved by redundancy.	Reliability can be improved by redundancy only when a parallel program is written and checked by a different team.

3 SINGLE-MACHINE SYSTEM PROBLEMS

3.1 System Description

3.1.1 System Definition

Consider a system that consists of a single machine involving the hardware and software in its operation. The system is operated by a single human operator. It is required to perform a number of tasks that randomly arrive during the fixed mission-time T. The human operator in the system has to perform as many tasks in the proper manner as the system demands.

The system can be in one of the two states *on* or *off*, where in the *on* state the system is operating and in the *off* state the system is down under repair if the system is repairable or the mission is terminated if the cause of the failure cannot be removed. If the system consists of more that one component, it will be *on* only if all components are *on* (see Fig. 3.1). The failures due to each component are statistically independent of each other and have a constant occurrence rate. The time to repair a failed system due to each component's error follows a negative exponential distribution.

Some random amount of time is required to complete each task. If the system is *on* and idle when the task arrives, then it is performed; if the task arrives during

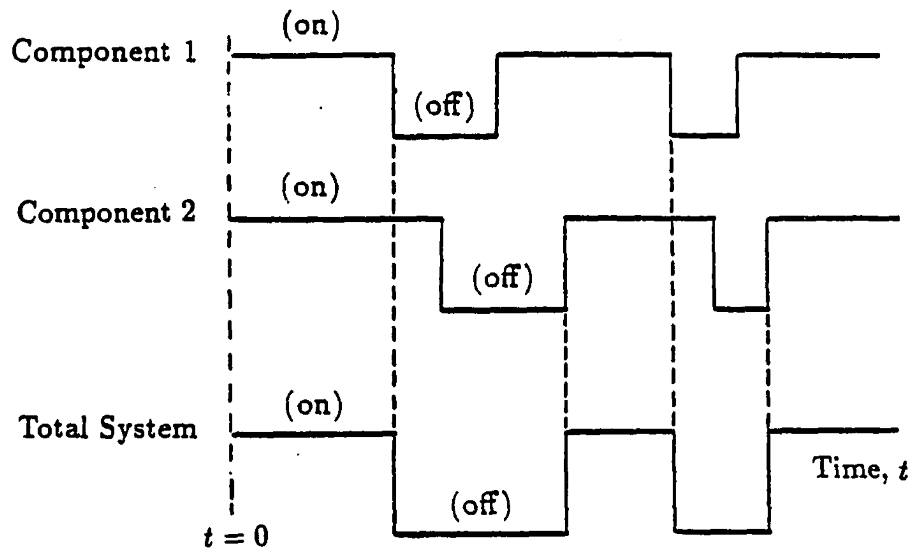

Figure 3.1: Up and down behavior of the system ($R = 2$)

the time for performance of the present task or during the down time, then the mission to perform the arriving task is assumed to be failed. It is also assumed that the system could be down while it is performing the task. In this case, the task fails to be completed and the system goes under repair, if it is repairable, and when done it is restored to its normal operating condition. Therefore, for each task to be successfully performed, the system should be both ready to function (be available) at the time of the task arrival and to operate (be reliable) during the performance time (see Fig. 3.2). Conditional on these two events, the human operator must detect the arrival of the task and perform the task accurately within the allocated time limit.

Failure of any one of the above conditions to be met will result in the failure

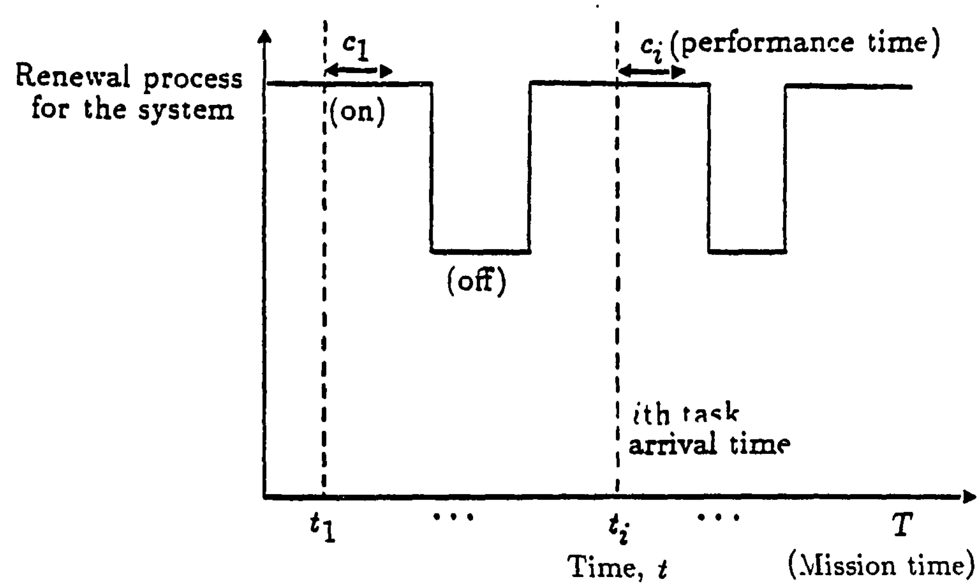

Figure 3.2: Availability and reliability of the system

to achieve the correct response for the task. If the system is not available at the task request, no action can be taken to perform the task. Regardless of whether the system is available at the task arrival, if it fails before the task is completely performed, then the mission is assumed to have failed. Likewise, regardless of whether the system is available and reliable, if the operator fails to detect the arrival of the task, fails to perform the task or part of it correctly, or fails to perform the task within an allocated time, then the mission is also assumed to have failed. If the operator fails to detect the task, no action is taken; the system requirements are not met if the task is performed incorrectly; and the mission is terminated if it cannot be completed within some limited time. Since factors such as fatigue, stress, and learning will affect the operator over time, the human performance variables are assumed to vary with time during the mission.

3.1.2 Formulation of System Effectiveness

The system considered in this study requires the successful performance of all the tasks arriving during mission time T. Depending upon the method of dealing with the situation of no task, system effectiveness can be defined in three ways. If no task-request for the system to perform during the mission is considered as a mission success, then system effectiveness, or SE, can be defined as

$$SE = \sum_{k=1}^{\infty} \Pr[N(T) = k] q_k + \Pr[N(T) = 0], \tag{3.1}$$

where $\Pr[N(T) = k]$ is the probability of k task-requests during the mission and q_k represents the probability that, given k task requests, all of them would be successfully performed.

In some cases, the system may still require the availability of the system even though there is no task request. The proportion of time the system is available during mission-time T can be represented by the average availability, A(T), as follows:

$$A(T) = \frac{1}{T} \int_0^T A(t) dt,$$

where $A(t)$ is the pointwise availability of the system at time t. For this system, we define

$$SE = \sum_{k=1}^{\infty} \Pr[N(T) = k] q_k + A(T) \Pr[N(T) = 0]. \tag{3.2}$$

If we do not take into account the mission that has no task request during the mission time, system effectiveness can be defined conditionally upon the existence of task requests during the mission. That is,

$$SE = \sum_{k=1}^{\infty} \Pr[N(T) = k] q_k / \Pr[N(T) \geq 1]. \tag{3.3}$$

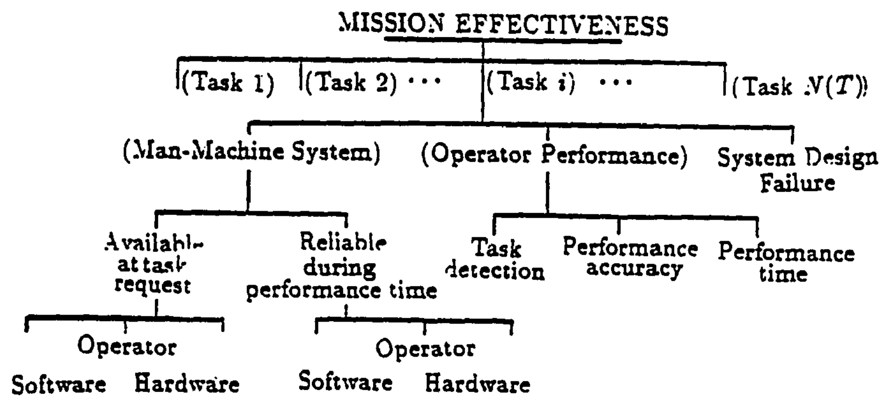

Figure 3.3: The proposed system-effectiveness model

The difference between the system-effectiveness values from the above three definitions will become smaller as task-arrival rates become larger. This is because the term $\Pr[N(T) = 0]$ goes to zero as task-arrival rates become larger.

3.1.3 System Variables

The proposed system-effectiveness model consists of five major factors: the task arrival process, the system state, system design failure, and the human-operator performance variables. The system is diagrammatically presented in Fig. 3.3. The system state has been discussed in Section 3.1.1. The others will be described in the following sections.

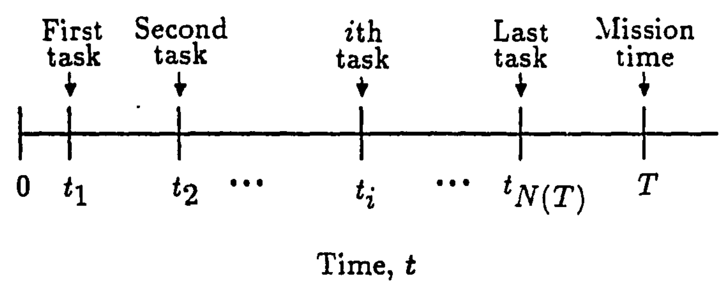

Figure 3.4: Task-arrival process

3.1.3.1 Task-Arrival Process The system developed in this study is required to perform a number of tasks. The number of tasks during the mission time T, $N(T)$, is a random variable having some probability distribution; and T_i ($i = 1, 2, \ldots, N(T)$) is the random variable representing the arrival time of the ith task (see Fig. 3.4). In this study, $N(T)$ is assumed to be a nonhomogeneous Poisson process. This means that the following properties are assumed [104]:

- $N(0) = 0$. That is, the system is operating at the time chosen for reference as time 0.
- The number of arrivals an interval is independent of the number of arrivals in any other disjoint interval.

- The probability of at least one task arrival in a small time interval of length Δt is $\lambda(t)\Delta t$, where $\lambda(t)$ is the task-arrival rate.

- The probability of two or more task arrivals occurring in a small time interval of length Δt is negligible.

Under the above assumptions, it can be shown that $N(T)$ follows the Poisson distribution [104]. That is,

$$\Pr[N(T) = k] = \frac{e^{-m(T)}m(T)^k}{k!} \equiv P_o[k; m(T)], \tag{3.4}$$

where $k = 1, 2, \ldots$ and $m(T) = \int_0^T \lambda(s)ds$ is the mean value function of the process. The task-arrival rate is characterized by the system under consideration and can be estimated as the average number of task arrivals during a unit of time.

3.1.3.2 System-Design Failure We adopt the concept of system design failure in relation to the allocated time limit for the task performed. If a new task arrival occurs during the performance of the current task, it may be undetected or ignored and is considered a failure because of the inadequacy of system design.

3.1.3.3 Human-Performance Variables There can be many human performance variables representing human behavior in the system. It is not necessary, however, to include all of these variables in the model because not all of them have a significant effect on total system performance. In this study, three performance variables are included in the model.

3.1.3.3.1 Detection of Task Arrival In order to perform a task, first the human operator must detect the task arrival, which may arrive in the

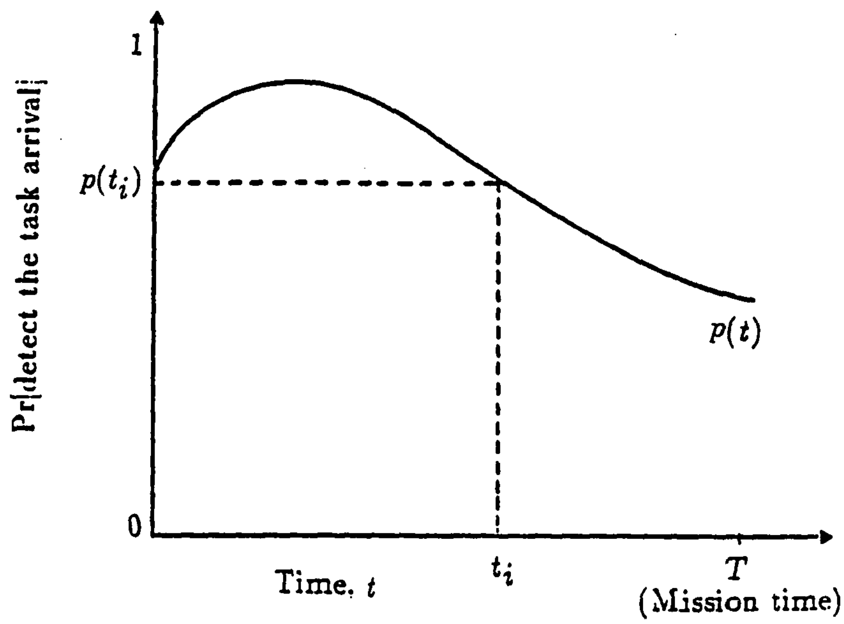

Figure 3.5: Typical probability function for detection of a task arrival

form of an auditory, visual, or tactile signal. For example, in the chemical blending process, the operator should detect the auditory or visual warning alarm in order to take the required action against undesirable results such as the overflow of a vessel. The probability that human operator will detect the task arrival can be assumed to change as a function of time during the mission (see Fig. 3.5), and can be represented by

$$\Pr[X(t_i) = 1] = p(t_i),\ i = 1, 2, \ldots, N(T), \tag{3.5}$$

where $X(t_i)$ has the value of 0 if the operator fails to detect the task arrival and 1 if the task is detected. The probability, $p(t_i)$, is an arbitrary functional form that can be fitted to real data.

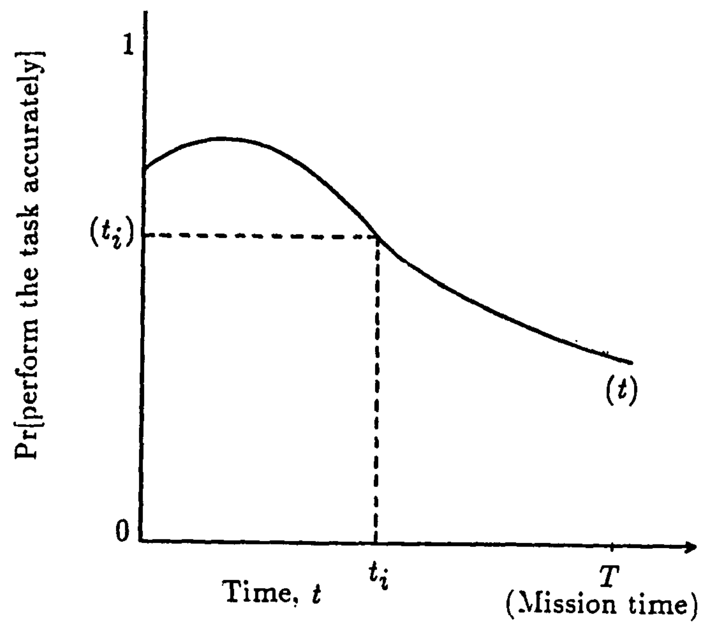

Figure 3.6: Typical probability function for human-performance accuracy

3.1.3.3.2 Performance Accuracy The detection of a task arrival does not guarantee successful task-performance. The task should be performed correctly according to a prespecified order. In this study, performance accuracy is binomially defined, that is, as a success or failure, according to the fulfillment of the intended requirements. For example, in a military system, if the task requires hitting the target and stopping its operation, the operator should follow the specified sequences to shoot the weapon after detecting the target. If the shooting completely stops the operation of the target, it is considered an accurate task-performance performance. The probability of accurately performing the task at time t_i is assumed to change as a function of time during the mission (see Fig. 3.6), and can be represented by

$$\Pr[Y(t_i) = 1 | X(t_i) = 1] = q(t_i), \ i = 1, 2, \ldots, N(T), \tag{3.6}$$

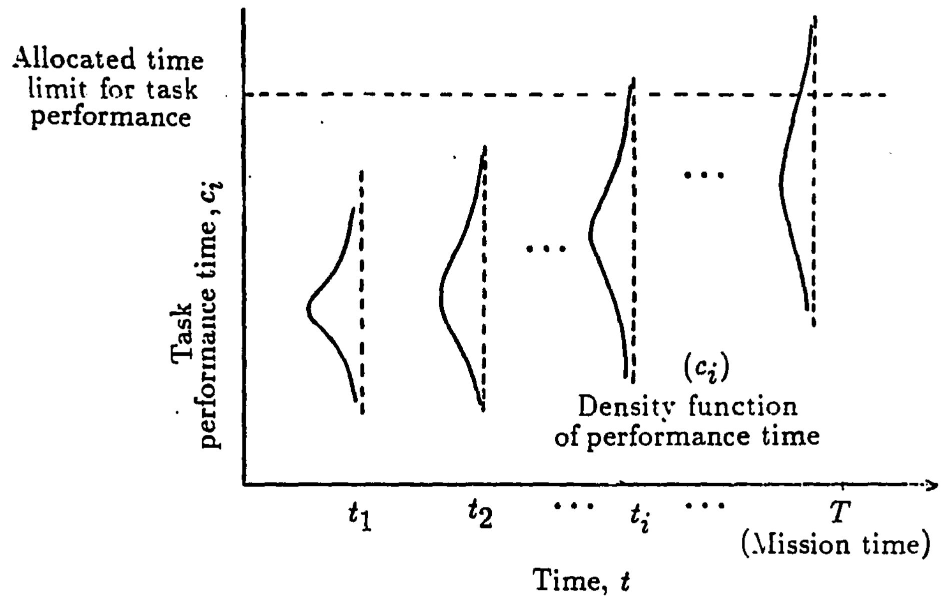

Figure 3.7: Probability-density function for performance times

where $Y(t_i)$ has a value of 1 if the operation is accurate, and 0 if the operation is not accurate. The probability $q(t_i)$ can be assumed to be an arbitrary functional form necessary to fit field data.

3.1.3.3.3 Performance Time The time taken to complete the task varies from task to task. The time to complete the ith task, C_i, is considered to be a random variable having a probability-density function $g(c_i)$. Some systems require that a task must be completed within an allocated time limit in order for the mission to be successful (see Fig. 3.7). For example, in military systems, a target should be destroyed before it is able to attack. The probability that the task be performed within an allocated time-limit δ can be formulated as:

$$\Pr[c_i \leq \delta | X(t_i) = 1] = \int_0^{\delta} g(c_i) dc_i \equiv s(t_i, \delta). \tag{3.7}$$

3.2 Development of Effectiveness Models

The system can be *on* (available) or *off* (unavailable) when the task arrives. Let us define M_i as the event that the system is available at time t_i and reliable during the performance time c_i; O_i as the event that the human operator detects the ith task and performs it accurately within an allowable time limit δ; and S_i as the event that the task arriving at time t_i is successfully performed–or $S_i = M_i \cap O_i$. Also let us define $\underline{t}_k = (t_1, t_2, \ldots, t_k)$ and $\underline{c}_k = (c_1, c_2, \ldots, c_k)$.

Now, given that there is only one task arriving during the fixed mission-time T, or $N(T) = 1$, system effectiveness can be formulated as follows:

$$\begin{aligned} SE(t_1, c_1) &= \Pr[S_1] \\ &= \Pr[M_1 \cap O_1] \\ &= \Pr[M_1] \Pr[O_1 | M_1]; \end{aligned} \tag{3.8}$$

or for the successful performance of the task that arrives at time t_1, the system should be available at time t_1 and reliable during performance time c_1. Conditional on these two events, the human operator should detect the task arrival and perform accurately within an allowable time-limit δ.

In the case of $N(T) = 2$, let us assume that task-arrival times are given as t_1 and t_2 with performance times c_1 and c_2, respectively. If the second task arrives during the performance time c_1 of the first task, it is considered to be a failure or if $t_1 + c_1 > t_2$, system effectiveness for the system is determined to be 0. If

$t_1 + c_1 < t_2$, system effectiveness is determined as a function of the success of each task. In this case:

$$
\begin{aligned}
SE(\underline{t}_2, \underline{c}_2) &= \Pr[S_1 \cap S_2] \\
&= \Pr[S_1]\Pr[S_2|S_1] \\
&= \Pr[S_1]\Pr[M_2 \cap O_2|S_1] \\
&= \Pr[S_1]\Pr[S_1 \cap M_2 \cap O_2]/\Pr[S_1] \\
&= \Pr[S_1]\Pr[S_1]\Pr[M_2|S_1]\Pr[O_2|S_1 \cap M_2]/\Pr[S_1] \\
&= \Pr[S_1]\Pr[M_2|S_1]\Pr[O_2|S_1 \cap M_2] \\
&= \Pr[S_1]\Pr[M_2|M_1 \cap O_1]\Pr[O_2|M_2] \\
&= \Pr[M_1]\Pr[O_1|M_1]\Pr[M_2|M_1]\Pr[O_2|M_2]. \qquad (3.9)
\end{aligned}
$$

Now let us assume that task arrivals are given as $t_1, t_2, \ldots, t_{N(T)}$, and performance time as $c_1, c_2, \ldots, c_{N(T)}$. Define two sets C and D such that

$$
\begin{aligned}
C &= \left\{\underline{t}_{N(T)}, \underline{c}_{N(T)} | 0 < t_1 < t_2 < \cdots < t_{N(T)} \leq T\right\}, \text{ and} \\
D &= \left\{\underline{t}_{N(T)}, \underline{c}_{N(T)} | 0 < t_1 + c_1 < t_2, \cdots, t_{N(T)} + c_{N(T)} \leq T\right\},
\end{aligned}
$$

where $c_i > 0$, $i = 1, \ldots, N(T)$. With the task arrival time $\left\{\underline{t}_{N(T)}, \underline{c}_{N(T)}\right\} \in (C - D)$, the probability of a mission success is 0 because of a system-design failure. With the task-arrival time $\left\{\underline{t}_{N(T)}, \underline{c}_{N(T)}\right\} \in D$, system effectiveness is determined as a function of each task. Given $N(T) = k$, from Eqs. (3.8) and (3.9), system effectiveness can be generalized as

$$
SE(\underline{t}_k, \underline{c}_k) = \prod_{i=1}^{k} \Pr[M_i|M_{i-1}]\Pr[O_i|M_i], \qquad (3.10)
$$

where $\Pr[M_0] \equiv 1$. Now system effectiveness can be determined by taking the expected value of conditional probability of Eq. (3.10) with respect to the joint conditional distribution of the arrival times and performance times, given that $N(T) = k$, and finally taking the expected value with respect to the Poisson distribution of $N(T)$. That is,

$$\begin{aligned} SE(T) &= E\left\{E\left[SE(\underline{t}_k, \underline{c}_k)\right]\right\} + Po[0; m(T)](1) \\ &= \sum_{k=1}^{\infty} E\left[SE(\underline{t}_k, \underline{c}_k)\right] Po[k; m(T)] + Po[0; m(T)]. \end{aligned} \tag{3.11}$$

Here

$$\begin{aligned} E\left[SE(\underline{t}_k, \underline{c}_k)\right] &= \int \underset{D}{\cdots} \int SE(\underline{t}_k, \underline{c}_k) f_s(\underline{t}_k, \underline{c}_k) dc_1 \ldots dc_k dt_1 \ldots dt_k \\ &= \int \underset{D}{\cdots} \int SE(\underline{t}_k, \underline{c}_k) f_k(\underline{t}_k) g_k(\underline{c}_k) dc_1 \ldots dc_k dt_1 \ldots dt_k, \end{aligned}$$

where $f_s(\underline{t}_k, \underline{c}_k)$ is the joint probability-density function of $(\underline{t}_k, \underline{c}_k)$ which is the multiplication of the joint probability-density function of task-arrival time $f_k(\underline{t}_k)$ and the joint probability-density function of the performance times $g_k(\underline{c}_k)$ by a statistically independent assumption between the two. Since $N(T)$ follows a nonhomogeneous Poisson process with arrival rate $\lambda(t)$, it can be proved that:

$$f_k(\underline{t}_k) = \frac{\prod_{i=1}^{k} \lambda(t_i)}{m(T)^k / k!}, \; 0 < t_1 < \cdots < t_k \le T; \tag{3.12}$$

and the joint probability-density function of performance times $c_1, \ldots, c_k$, by an identical independent distribution assumption, can be determined as follows:

$$g_k(\underline{c}_k) = \prod_{i=1}^{k} g(c_i), \; c_i > 0, \; i = 1, 2, \ldots, k \tag{3.13}$$

To prove Eq. (3.12), let us assume, given that k task arrivals have occurred in $(0, T)$, that in each of k nonoverlapping subintervals $[t_1, t_1 + h_1], \ldots, [t_k, t_k + h_k]$

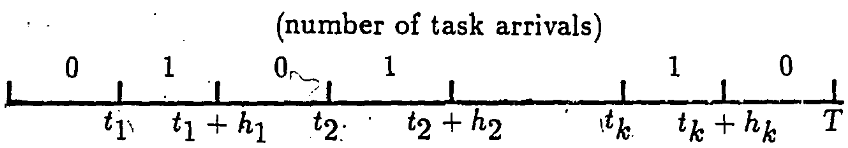

Figure 3.8: Task arrival process

exactly one task arrival occurs, and elsewhere no task arrival occurs (see Fig. 3.8). In such a case, the conditional probability is

$$
\begin{aligned}
&\Pr[t_1 \le T_1 \le t_1 + h_1, \ldots, t_{N(T)} \le T_{N(T)} \le t_{N(T)} + h_{N(T)}] \\
&= \frac{\Pr[t_1 \le T_1 \le t_1 + h_1, t_2 \le T_2 \le t_2 + h_2, \ldots, t_k \le T_k \le t_k + h_k]}{P_o[k; m(T)]} \\
&= \prod_{i=1}^{k} \mathrm{e}^{-[m(t_i) - m(t_{i-1} + h_{i-1})]} [m(t_i + h_i) - m(t_i)] \mathrm{e}^{-[m(t_i + h_i) - m(t_i)]} \\
&\quad \times \mathrm{e}^{-[m(T) - m(t_k + h_k)]} \Big/ \frac{\mathrm{e}^{-m(T)} m(T)^k}{k!} \\
&= \frac{\prod_{i=1}^{k} [m(t_i + h_i) - m(t_i)]}{m(T)^k / k!}.
\end{aligned}
\tag{3.14}
$$

By the definition of the density function, the left-hand side of Eq. (3.14) is approximately equal to

$$f_{N(T)}(\underline{t}_{N(T)}|N(T) = k)h_i h_2 \dots h_k. \tag{3.15}$$

Therefore,

$$f_{N(T)}(\underline{t}_{N(T)}|N(T) = k) = \frac{\prod_{i=1}^{k}[m(t_i + h_i) - m(t_i)]/h_i}{m(T)^k/k!} \tag{3.16}$$

As h_i $(i = 1, 2, \dots, k)$ approaches to 0, and

$$\begin{aligned} f_{N(T)}(\underline{t}_{N(T)}|N(T) = k) &= \lim_{(h_1, h_2, \dots, h_k) \to 0} \frac{\prod_{i=1}^{k}[m(t_i + h_i) - m(t_i)]/h_i}{m(T)^k/k!} \\ &= \frac{\prod_{i=1}^{k}\lambda(t_i)}{m(T)^k/k!}, 0 < t_1 < t_2 < \dots < t_k \leq T, \end{aligned} \tag{3.17}$$

where

$$\lambda(t_i) = \lim_{h_i \to 0}[m(t_i + h_i) - m(t_i)]. \tag{3.18}$$

Having proof Eq. (3.12), system effectiveness can now be expressed as follows:

$$\begin{aligned} SE(T) &= \sum_{k=1}^{\infty}\left\{\int \dot{\underset{D}{\dots}} \int \frac{SE(\underline{t}_k, \underline{c}_k)\prod_{i=1}^{k}\lambda(t_i)g(c_i)}{m(T)^k/k!} dc_1 \dots dc_k dt_1 \right. \\ &\quad \left. \dots dt_k \times Po[k; m(T)]\right\} + Po[0; m(T)] \end{aligned} \tag{3.19}$$

Note that $SE(\underline{t}_k, \underline{c}_k)$ is a function of availability, reliability, and operator performance at each task arrival. If we define A_i as the event that the system is available at time t_i, given that the system is available at the end of the $(i-1)$th task performance; and R_i as the event that the system is reliable during the performance time c_i, given the availability at time t_i, then the system effectiveness, given that $N(T) = k$, presented in Eq. (3.10) can be rewritten as:

$$SE(\underline{t}_k, \underline{c}_k) = \prod_{i=1}^{k} \Pr[A_i|M_{i-1}]\Pr[R_i|A_i \cap M_{i-1}]\Pr[O_i|M_i]. \tag{3.20}$$

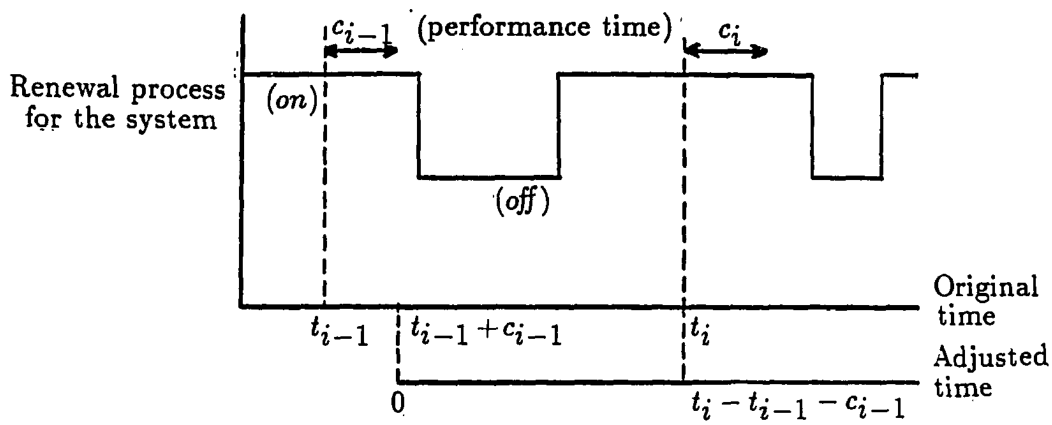

Figure 3.9: Adjustment of time

Given that the system is operating at t_{i-1} and functioning during the performance time c_{i-1}, this system is like new at time $t_{i-1} + c_{i-1}$ because of the memoryless property of the Markovian process. If we adjust the time $t_{i-1} + c_{i-1}$ to time zero for a reference point (see Fig. 3.9), then the first term of Eq. (3.20) can be redefined as:

$$\Pr[A_i | M_{i-1}] \equiv a(t_i^*),$$

where $t_i^* = t_i - t_{i-1} - c_{i-1}$. Again by the memoryless property, the probability of the system is reliable during the performance time c_i, given that the availability at the task-arrival time t_i is independent of the system condition before time t_i. Therefore, the second term of Eq. (3.20) can be redefined as:

$$\Pr[R_i | A_i \cap M_{i-1}] \equiv r(t_i, c_i).$$

Given that the system is available at each task arrival and reliable during the task performance, it is assumed that the performance of the human operator is characterized only by the task-arrival time. Under this assumption, the third term of Eq. (3.20) can be redefined as:

$$\Pr[O_i|M_i] \equiv o(t_i, \delta).$$

System effectiveness, given that $N(T) = k$, can now be rewritten as:

$$SE(\underline{t}_k, \underline{c}_k) = \prod_{i=1}^{k} a(t_i^*) r(t_i, c_i) o(t_i, \delta); \tag{3.21}$$

and overall system effectiveness can be rewritten as follows:

$$SE(T) = \sum_{k=1}^{\infty} \left\{ \int \dot{\underset{D}{\cdot}} \int \frac{\prod_{i=1}^{k} a(t_i^*) r(t_i, c_i) o(t_i, \delta) \lambda(t_i) g(c_i)}{m(T)^k / k!} dc_1 \ldots dc_k dt_1 \ldots dt_k \times P_o[k; m(T)] \right\} + P_o[0; m(T)]. \tag{3.22}$$

The above definition of system effectiveness is based on the assumption that no task arrival during the mission is considered one definition of mission success. As discussed in Section 3.1.2, if the system requires the availability of the system even when there is no task request during the mission, then

$$SE(T) = \sum_{k=1}^{\infty} \left\{ \int \dot{\underset{D}{\cdot}} \int \frac{\prod_{i=1}^{k} a(t_i^*) r(t_i, c_i) o(t_i, \delta) \lambda(t_i) g(c_i)}{m(T)^k / k!} dc_1 \ldots dc_k dt_1 \ldots dt_k \times P_o[k; m(T)] \right\} + P_o[0; m(T)] \int_0^T A(t) dt / T. \tag{3.23}$$

For some systems, missions can occur only if there are task arrivals. For such systems,

$$SE(T) = \sum_{k=1}^{\infty} \left\{ \int \dot{\underset{D}{\cdot}} \int \frac{\prod_{i=1}^{k} a(t_i^*) r(t_i, c_i) o(t_i, \delta) \lambda(t_i) g(c_i)}{m(T)^k / k!} dc_1 \ldots dc_k dt_1 \right.$$

$$\ldots dt_k \times P_o[k; m(T)]\Big\} \,/\, \{1 - P_o[0; m(T)]\}\,. \tag{3.24}$$

The Eqs. (3.1), (3.2), and (3.3) in Section 3.1.2 are equivalent to Eqs. (3.22), (3.23), and (3.24), respectively, if

$$\int \underset{D}{\cdots} \int \frac{\prod_{i=1}^{k} a(t_i^*) r(t_i, c_i) o(t_i, \delta) \lambda(t_i) g(c_i)}{m(T)^k / k!} dc_1 \ldots dc_k dt_1 \ldots dt_k$$

is replaced by q_k.

To continue the analytical derivation, we must derive the close forms for $a(t_i^*)$, $r(t_i, c_i)$, and $o(t_i, \delta)$.

3.3 Availability Derivation

Let us assume that the system has R components, with each component having a constant failure and repair rate, if it is repairable, and that is satisfies the following additional assumptions:

- The failures due to the rth component's error are statistically independent of each other and have an occurrence rate α_r.
- The probability of two or more errors occurring simultaneously is negligible.
- The repair processes of a failed system due to the rth component's error are statistically independent of each other and have a constant rate β_r.
- Failures and repairs of one component are statistically independent of both the failures and repairs of the other components.
- A failed system caused by any component is repaired back to its original operational state.

The probability that the system is available at time t_i, given that the system is operable at the end of the performance of the $(i-1)$th task, can be written as follows:

$$\begin{aligned} a(t_i^*) &= \Pr[\bigcap_{r=1}^{R} \left\{Z_r(t_i) = 1\right\} \mid \bigcap_{r=1}^{R} \left\{Z_r(t_{i-1}) = 1, T_f^r(t_{i-1}) \geq c_{i-1}\right\}] \\ &= \Pr[\bigcap_{r=1}^{R} \left\{Z_r(t_i) = 1 \mid Z_r(t_{i-1}) = 1, T_f^r(t_{i-1}) \geq c_{i-1}\right\}], \end{aligned} \quad (3.25)$$

where $Z_r(t)$ is the indicator variable representing the state of the rth component of the system at time t ($0 = \mathit{off}$, $1 = \mathit{on}$) and $T_f^r(t)$ is the time to failure of the rth component of the system measured from time t. Given that the system is operating at t_{i-1} and functioning during the performance time c_{i-1}, this system is like new at time $t_{i-1} + c_{i-1}$ because of the memoryless property of the Markovian process. If we adjust the time $t_{i-1} + c_{i-1}$ to time zero for a reference point (see Fig. 3.9), then Eq. (3.25) can be rewritten as:

$$\begin{aligned} a(t_i^*) &= \Pr[\bigcap_{r=1}^{R} \left\{Z_r(t_i - t_{i-1} - c_{i-1}) = 1\right\}] \\ &= \Pr[\bigcap_{r=1}^{R} \left\{Z_r(t_i^*) = 1\right\}] \\ &= P_0(t_i^*) \end{aligned} \quad (3.26)$$

where $P_0(t)$ is the probability that the system is *on* at time t; $t_i^* = t_i - t_{i-1} - c_{i-1}$, $i = 1, 2, \ldots, N(T)$; $t_0 = 0$; $c_0 = 0$. For simplification, the index i and the asterisk on t_i^* are suppressed throughout the derivation. The differential-difference equations corresponding to the system represented by the rate diagram in Fig. (3.10) can be given as

$$\frac{d}{dt} P_0(t) = \sum_{r=1}^{R} \beta_r P_r(t) - \sum_{r=1}^{R} \alpha_r P_0(t) \quad (3.27)$$

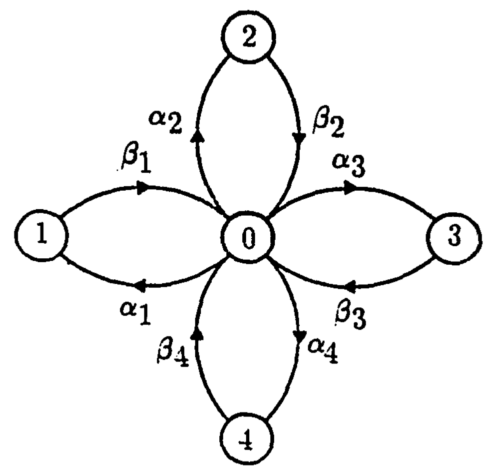

Figure 3.10: System rate diagram for $R = 4$

$$\frac{d}{dt}P_r(t) = \alpha_r P_0(t) - \beta_r P_r(t), \tag{3.28}$$

where $P_r(t)$ is the probability that the system is *off* at time t due to the rth component's error. Taking the Laplace transforms of both sides of Eqs. (3.27) and (3.28), we obtain

$$s\overline{P}_0(s) - P_0(0) = \sum_{r=1}^{R} \beta_r \overline{P}_r(s) - \sum_{r=1}^{R} \alpha_r \overline{P}_0(s) \tag{3.29}$$

$$s\overline{P}_r(s) - P_r(0) = \alpha_r \overline{P}_0(s) - \beta_r \overline{P}_r(s), \tag{3.30}$$

where $\overline{P}_x(s)$ is the Laplace transform of $P_x(t)$; s is a Laplace transform variable; and $P_0(0)$, $P_r(0)$ are 1 and 0, respectively, since the system is operating at time $t_i^* = 0$. Therefore, from Eq. (3.30), we have

$$\overline{P}_r(s) = \left\{\frac{\alpha_r}{s + \beta_r}\right\} \overline{P}_0(s). \tag{3.31}$$

By substituting $P_0(0) = 1$ and the values of $\overline{P}_r(s)$ from Eq. (3.31) into Eq. (3.29), we get

$$
\begin{aligned}
s\overline{P}_0(s) - 1 &= \sum_{r=1}^{R} \left\{ \frac{\alpha_r \beta_s}{s + \beta_r} \right\} \overline{P}_0(s) - \sum_{r=1}^{R} \alpha_r \overline{P}_0(s) \\
&= \sum_{r=1}^{R} \left\{ \frac{\alpha_r \beta_s}{s + \beta_r} - \alpha_r \right\} \overline{P}_0(s) \\
&= -\sum_{r=1}^{R} \left\{ \frac{\alpha_r s}{s + \beta_r} \right\} \overline{P}_0(s), \qquad (3.32)
\end{aligned}
$$

so that

$$
\overline{P}_0(s) = \left\{ s + \sum_{r=1}^{R} \frac{\alpha_r s}{s + \beta_r} \right\}^{-1}. \qquad (3.33)
$$

This transform can be inverted for any set values of α_r and β_r to obtain the availability of the system $a(t_i^*)$ or $P_0(t_i^*)$.

3.4 Reliability Derivation

The conditional probability that the system will be functioning for the task-performance time c_i, given that the system operates at task-request time t_i, can be formulated as follows:

$$
\begin{aligned}
r(t_i, c_i) = \Pr[&\bigcap_{r=1}^{R} \left\{ T_f^r(t_i) \geq c_i \right\} \mid \\
&\bigcap_{r=1}^{R} \left\{ Z_r(t_i) = 1, Z_r(t_{i-1}) = 1, T_f^r(t_{i-1}) \geq c_{i-1} \right\}]. \qquad (3.34)
\end{aligned}
$$

By the memoryless property of the Markovian process, Eq. (3.34) can be rewritten as

$$
r(t_i, c_i) = \Pr[\bigcap_{r=1}^{R} \left\{ T_f^r(t_i) \geq c_i \right\} \mid \bigcap_{r=1}^{R} \left\{ Z_r(t_i) = 1 \right\}]
$$

$$
\begin{aligned}
&= \prod_{r=1}^{R} \Pr[T_f^r(t_i) \geq c_i | Z_r(t_i) = 1] \\
&= \prod_{r=1}^{R} e^{-\alpha_r c_i} \\
&= e^{-\alpha . c_i}, \qquad (3.35)
\end{aligned}
$$

or time to failure of the R-component system is exponentially distributed with the parameter $\alpha_{.} = \sum_{r=1}^{R} \alpha_r$.

3.5 Quantification of Human Performance

In the quantification of human performance effects, it is assumed that, given the system availability at each task request time and reliability during the allocated time, the successful performance of each task is independent of the other tasks and that, as discussed in Section 3.1.1, the occurrence of any one of the proposed types of human error will cause the mission to fail. Under these assumptions, the human-operator effect on task performance c_i can be formulated as follows:

$$
\begin{aligned}
o(t_i, \delta) &= \Pr[X(t_i) = 1, Y(t_i) = 1, c_i \leq \delta | M_i] \\
&= \Pr[X(t_i) = 1 | M_i] \\
&\quad \times \Pr[c_i \leq \delta | X(t_i) = 1, M_i] \\
&\quad \times \Pr[Y(t_i) = 1 | X(t_i) = 1, M_i] \\
&= p(t_i) s(t_i, \delta) q(t_i). \qquad (3.36)
\end{aligned}
$$

The first term in Eq. (3.36) represents the probability of detecting the task arriving at time t_i; the second term represents the probability of completing the task within allocated time δ; the third term represents the probability of accurately performing the task under some given conditions.

3.6 Illustrations

In this section we present the effectiveness model of four human-hardware-software systems using the approach proposed in the previous section:

- *Model A*: The model represents a repairable system in which failure due to human error and failure due to the hardware are not separated. Also, in this model, the hardware includes all software needed to make it work. More specifically, when the system fails, a repairman is sent. It is assumed that the repairman will be able to repair the failed system. The failure and repair rates are α_1 and β_1, respectively.

- *Model B*: In this repairable system, failure due to man-hardware and the failure due to software are differentiated. The term "man-hardware" is used to represent cases where operator and hardware are treated as a single component of the system. When the system fails due to either man-hardware or software, a repairman is sent. The man-hardware and the software need a different repairman. Both repairmen are assumed to be capable of repairing a failed system. The repaired system is put back in its normal operation. The failure and repair rates of the man-hardware and the software components are α_1, α_2 and β_1, β_2, respectively.

- *Model C*: In this model, the operator performs continuous tasks involving some kind of tracking activity such as monitoring a changing situation. Examples of time-continuous tasks performed by humans are aircraft maneuvering, missile count-down, and scope monitoring. The modeling concept for human reliability in such situations is analogous to classical reliability modeling [15]. This concept easily allows us human error from failures due to other components of the system, in our case, hardware and software components. As in previous models, the system is repairable and each type of failure requires a different repairman. The failure and repair rates of the operator, hardware, and software are $\alpha_1, \alpha_2, \alpha_3$ and $\beta_1, \beta_2, \beta_3$, respectively.

- *Model D*: This model is the same as Model C, except that solely in the case of failure due to hardware or operator is a repairman sent to fix the system. A fixed system is returned to its normal operating condition. No attempt is made to repair a failed system due to software error. In this case, the mission is terminated.

On inverting Eq. (3.33) with the appropriate r, we obtain the following results on the availabilities for the system defined above:

Model A:

$$a(t_i^*) = X_0 + X_1 e^{s_1}, \tag{3.37}$$

where

$$\begin{aligned} X_0 &= \frac{\beta_1}{\beta_1 + \alpha_1} \\ X_1 &= \frac{\alpha_1}{\beta_1 + \alpha_1} \\ s_1 &= -(\beta_1 + \alpha_1). \end{aligned}$$

Model B:

$$a(t_i^*) = X_0 + X_1 e^{s_1 t_i^*} + X_2 e^{s_2 t_i^*}, \tag{3.38}$$

where

$$\begin{aligned}
X_0 &= \frac{\beta_1\beta_2}{s_1 s_2} \\
X_1 &= \frac{(s_1+\beta_1)(s_1+\beta_2)}{s_1(s_1-s_2)} \\
X_2 &= \frac{(s_2+\beta_1)(s_2+\beta_2)}{s_2(s_2-s_1)} \\
s_1 &= \frac{-x_1+\sqrt{x_1^2-4x_2}}{2} \\
s_2 &= \frac{-x_1-\sqrt{x_1^2-4x_2}}{2} \\
x_1 &= \beta_1+\beta_2+\alpha_1+\alpha_2 \\
x_2 &= \beta_1\beta_2+\alpha_1\beta_2+\alpha_2\beta_2.
\end{aligned}$$

Model C:

$$a(t_i^*) = X_0 + X_1 e^{s_1 t_i^*} + X_2 e^{s_2 t_i^*} + X_3 e^{s_3 t_i^*}, \tag{3.39}$$

where

$$\begin{aligned}
X_0 &= -\frac{\beta_1\beta_2\beta_3}{s_1 s_2 s_3} \\
X_1 &= \frac{(\beta_1+s_1)(\beta_2+s_1)(\beta_3+s_1)}{s_1(s_1-s_2)(s_1-s_3)} \\
X_2 &= \frac{(\beta_1+s_2)(\beta_2+s_2)(\beta_3+s_2)}{s_2(s_2-s_1)(s_2-s_3)} \\
X_3 &= \frac{(\beta_1+s_3)(\beta_2+s_3)(\beta_3+s_3)}{s_3(s_3-s_1)(s_3-s_2)}
\end{aligned}$$

and s_r's are the roots of

$$s^3 + x_1 s^2 + x_2 s + x_3 = 0,$$

where

$$\begin{aligned}
x_1 &= \alpha_1 + \alpha_2 + \alpha_3 + \beta_1 + \beta_2 + \beta_3 \\
x_2 &= \beta_1\beta_2 + \beta_1\beta_3 + \beta_2\beta_3 + \alpha_1(\beta_2 + \beta_3) + \alpha_2(\beta_1 + \beta_3) + \alpha_3(\beta_1 + \beta_2) \\
x_3 &= \beta_1\beta_2\beta_3 + \alpha_1\beta_2\beta_3 + \alpha_2\beta_1\beta_3 + \alpha_3\beta_1\beta_2.
\end{aligned}$$

Model D:

$$a(t_i^*) = X_1 e^{s_1 t_i^*} + X_2 e^{s_2 t_i^*} + X_3 e^{s_3 t_i^*}. \tag{3.40}$$

where

$$\begin{aligned}
X_1 &= \frac{(\beta_1 + s_1)(\beta_1 + s_1)}{(s_1 - s_2)(s_1 - s_3)} \\
X_2 &= \frac{(\beta_1 + s_2)(\beta_1 + s_2)}{(s_2 - s_1)(s_2 - s_3)} \\
X_3 &= \frac{(\beta_1 + s_3)(\beta_1 + s_3)}{(s_3 - s_1)(s_3 - s_2)}
\end{aligned}$$

and s_r's are the roots of

$$s^3 + x_1 s^2 + x_2 s + x_3 = 0,$$

where

$$\begin{aligned}
x_1 &= \alpha_1 + \alpha_2 + \alpha_3 + \beta_1 + \beta_2 \\
x_2 &= \beta_1\beta_2 + \alpha_1\beta_2 + \alpha_2\beta_1 + \alpha_3(\beta_1 + \beta_2) \\
x_3 &= \alpha_3\beta_1\beta_2.
\end{aligned}$$

Therefore for Model A, B, C, and D

$$a(t_i^*)r(t_i,c_i)o(t_i,\delta) = \left\{X_0 + \sum_{r=1}^{R} X_r e^{s_r t_i^*}\right\}\left\{e^{-\alpha . c_i}\right\}\left\{p(t_i)s(t_i,\delta)q(t_i)\right\}, \tag{3.41}$$

where the first term is from Eqs. (3.37), (3.38), (3.39), and (3.40) for System A,B,C, and D, respectively, with $X_0 = 0$ for System D; where the second term is from Eq. (3.35) with the appropriate choice of r; and where the third term is from Eq. (3.36). System effectiveness can be evaluated by substituting Eq. (3.41) into Eqs. (3.22), (3.23), and (3.24), depending upon the definition of the system being investigated.

3.7 Numerical Examples

3.7.1 Example 1

Suppose we have a Model A system with characteristics as follows: 1) the number of tasks arriving during the mission follows a Poisson process with $\lambda(t) = 0.05$; 2) the *on* time and *off* time of the system follows the exponential distribution with means 20 and 1, respectively, 3) mission time has a fixed value, $T = 10$, 4) the allocated time limit for task performance is given as 0.1, and 5) the human operator has the following performance characteristics: the probability of detecting the ith task is $p(t_i) = \exp(-0.01t_i)$, $0 < t_i \leq 10$; the probability of accurately performing the ith task is $q(t_i) = \exp(-0.02t_i)$, $0 < t_i \leq 10$; and the performance time follows an exponential distribution with a parameter of 25, that is, $g(c_i) = 25\exp(-25c_i)$, $c_i > 0$.

Using the formula in the form presented in Section 3.1.2, system effectiveness can thus be written as:

$$SE_1(T) = \sum_{i=1}^{k} q_k Po[k; m(T)] + Po[0; m(T)] \tag{3.42}$$

$$SE_2(T) = \sum_{i=1}^{k} q_k Po[k; m(T)] + Po[0; m(T)]A(T) \tag{3.43}$$

$$SE_3(T) = \sum_{i=1}^{k} q_k Po[k; m(T)]/(1 - Po[0; m(T)]), \tag{3.44}$$

where $SE_1(T)$ is system effectiveness when no task is considered as a mission success; $SE_2(T)$ is system effectiveness when the availability is required even though there is no task; $SE_3(T)$ is system effectiveness when missions can occur only if there are task requests; q_k is the probability of successfully performing k out of k tasks; and $A(T)$ is the average availability during mission time T. For this example,

$$A(T) = \frac{1}{T}\int_0^T \left\{X_0 + X_1 e^{s_1 t}\right\} dt$$

$$q_k = \int \dot{\dot{D}} \int \frac{\prod_{i=1}^{k} a(t_i^*) e^{-(\alpha_1+\mu)c_i-(\tau+v)t_i}}{T^k/\{\mu^k(1-e^{-\mu\delta})^k k!\}} dc_1 \ldots dc_k dt_1 \ldots dt_k,$$

where

$$X_0 = \frac{\beta_1}{\alpha_1+\beta_1}$$

$$X_1 = \frac{\alpha_1}{\alpha_1+\beta_1}$$

$$s_1 = -(\alpha_1+\beta_1)$$

$$a(t_i^*) = X_0 + X_1 e^{s_1 t_i^*}.$$

By substituting the values of the parameters for the system in the example, i.e., $\alpha_1 = 0.05, \beta_1 = 1, \lambda(t) = 0.05, T = 10, \tau = 0.01, v = 0.02, \mu = 25$ and $\delta = 0.1$, into Eqs. (3.42), (3.43) and (3.44), and by performing the integration, we obtain

Table 3.1: Analytical and simulation results for Example 1

Statistics	Analytical	Simulation
$A(10)$	0.955	0.941
$P_o[0; 0.5]$	0.607	0.607
$P_o[1; 0.5]$	0.303	0.301
$P_o[2; 0.5]$	0.076	0.076
$\Pr[N(10) \geq 3]$	0.014	0.016
q_1	0.823	0.786
q_2	0.570	0.580
$SE_1(10)$	0.899	0.891
$SE_2(10)$	0.872	0.855
$SE_3(10)$	0.744	0.721

the results presented in Table 3.1. These analytical solutions are compared with the results from the simulation. From Table 3.1, system effectivenesses obtained by analytical solutions are close to the system effectivenesses obtained by simulation. The FORTRAN program for the simulation of this example is given in Appendix A.

3.7.2 Example 2

Consider the same system given in the previous example except, now the man-hardware and the software components are treated as different components of the system. Therefore, we have a model-B system. The *on* time and *off* time of the man-hardware component follow negative exponential distribution, with parameter

0.05 and 1, respectively. The length of *on* time and *off* time of the software component also follows negative exponential distribution with parameter 0.05 and 1, respectively. The other characteristics of the system are the same as in the previous example.

Using the same procedures applied in the previous example, we have

$$SE_1(T) = \sum_{i=1}^{k} q_k Po[k; m(T)] + Po[0; m(T)] \quad (3.45)$$

$$SE_2(T) = \sum_{i=1}^{k} q_k Po[k; m(T)] + Po[0; m(T)] A(T) \quad (3.46)$$

$$SE_3(T) = \sum_{i=1}^{k} q_k Po[k; m(T)] / (1 - Po[0; m(T)]), \quad (3.47)$$

where

$$A(T) = \frac{1}{T}\int_0^T \left\{X_0 + X_1 e^{s_1 t} + X_2 e^{s_2 t}\right\} dt$$

$$q_k = \int \dot{\underset{D}{\cdots}} \int \frac{\prod_{i=1}^{k} a(t_i^*) e^{-(\alpha_\cdot + \mu)c_i - (\tau + v)t_i}}{T^k / \{\mu^k (1 - e^{\mu\delta})^k k!\}} dc_1 \ldots dc_k dt_1 \ldots dt_k,$$

and

$$X_0 = \frac{\beta_1 \beta_2}{s_1 s_2}$$

$$X_1 = \frac{(s_1 + \beta_1)(s_1 + \beta_2)}{s_1(s_1 - s_2)}$$

$$X_2 = \frac{(s_2 + \beta_1)(s_2 + \beta_2)}{s_2(s_2 - s_1)}$$

$$s_1 = \frac{-x_1 + \sqrt{x_1^2 - 4x_2}}{2}$$

$$s_2 = \frac{-x_1 - \sqrt{x_1^2 - 4x_2}}{2}$$

$$x_1 = \beta_1 + \beta_2 + \alpha_1 + \alpha_2$$

Table 3.2: Analytical and simulation results for Example 2

Statistics	Analytical	Simulation
$A(10)$	0.917	0.888
$P_o[0; 0.5]$	0.607	0.602
$P_o[1; 0.5]$	0.303	0.301
$P_o[2; 0.5]$	0.076	0.079
$\Pr[N(10) \geq 3]$	0.014	0.018
q_1	0.728	0.693
q_2	0.526	0.508
$SE_1(10)$	0.867	0.857
$SE_2(10)$	0.817	0.790
$SE_3(10)$	0.662	0.642

$$
\begin{aligned}
x_2 &= \beta_1\beta_2 + \alpha_1\beta_2 + \alpha_2\beta_1 \\
a(t_i^*) &= X_0 + X_1 e^{s_1 t_i^*} + X_2 e^{s_2 t_i^*}.
\end{aligned}
$$

By substituting the values of the parameters for the system in the example, i.e., $\alpha_2 = 0.05, \beta_2 = 1, \alpha_1 = 0.05, \beta_1 = 1, \lambda(t) = 0.05, T = 10, \tau = 0.01, \upsilon = 0.02, \mu = 25$ and $\delta = 0.1$, into Eqs. (3.45), (3.46), and (3.47), and by performing the integration, we obtain the results presented in Table 3.2. These analytical solutions can be compared with the results from the simulation. As shown in Table 3.2, the system effectivenesses obtained by analytical solutions are close to the system effectivenesses obtained by simulation as performed by the FORTRAN program given in Appendix B.

4 N-MACHINE SYSTEM PROBLEM

4.1 System Descriptions

4.1.1 System Definition

In this chapter, we will consider an N-machine system problem. Each machine involves hardware and software and is operated by a single human operator. This system is required to perform a number of tasks randomly arriving during the fixed mission-time T. The human operator of each machine in the system has to perform a prescribed function simultaneously with the operators at all other machines at each task arrival.

All characteristics discussed in regard to single-machine system in previous chapters are also applied to the system proposed here. That is, the system can be in one of the two states, *on* or *off* where in *on* state the system is operating and in *off* state the system is down under repair if the system is repairable or the mission is terminated if the cause of the failure cannot be removed. The system is *off* if one component of any machine fails to function (see Fig. 4.1). The failures due to each component are statistically independent of each other and have a constant occurrence-rate. The time to repair the system due to each component error follows exponential distribution. Some random amount of time is required by each machine

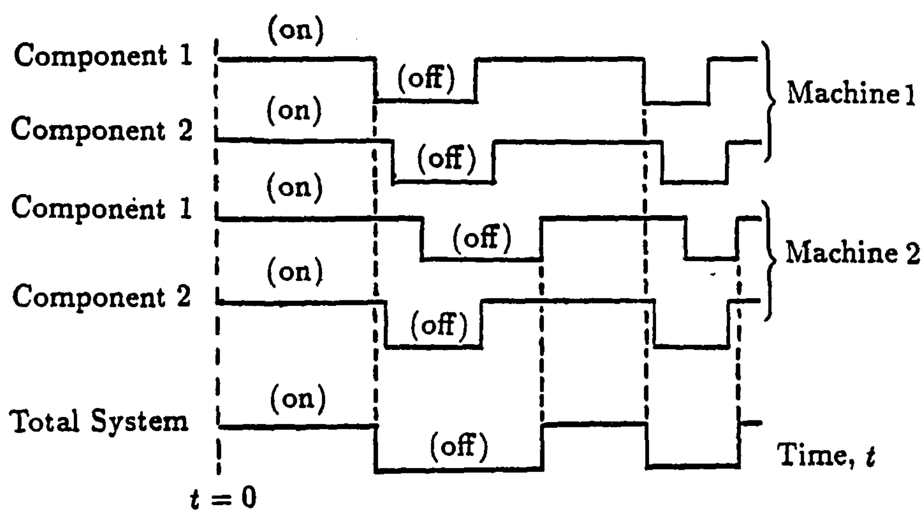

Figure 4.1: Up and down behavior of the system ($N = 2, R = 2$)

to complete each task. For each task to be successfully performed, all machines should be ready both to function (be available) at the time of the task arrival and to operate (be reliable) during the allocated performance-time δ (see Fig. 4.2). As in single-machine problems, the system has to be functioning only for the time ($\leq \delta$) it takes to complete the task. With a relatively small value of δ, however, this approximation will not make a significant difference in the value of mission effectiveness. Conditional on the above two events, the human operator in each machine must detect the arrival of the task and perform the task accurately within the allocated time limit. Failure of any one of the above conditions to be met will result in failure to achieve the correct response for the task. If one of the machines is not available at the task request, no proper action can be taken to perform the

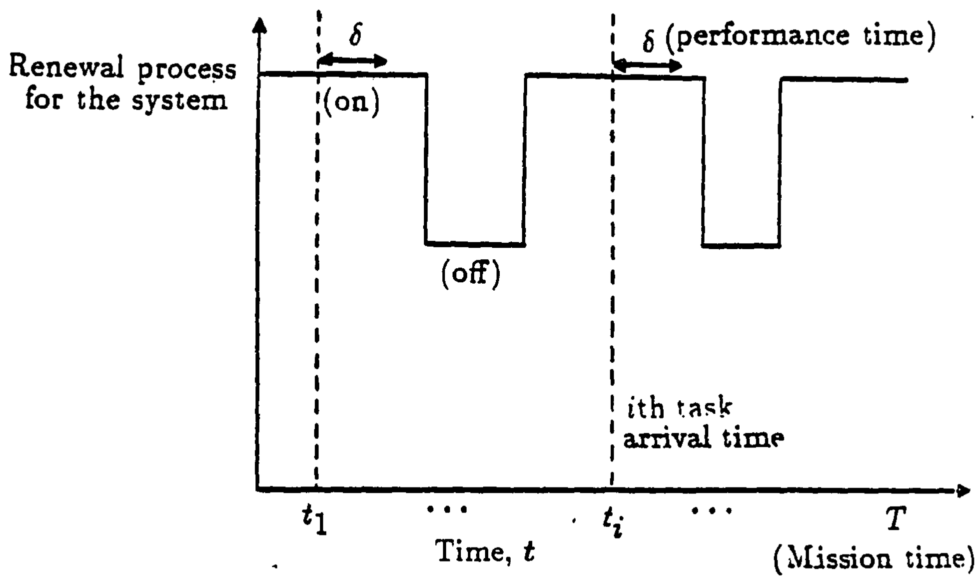

Figure 4.2: Availability and reliability of the system

task. Even though all machines are available at the task arrival, if one of them fails before the task is completely performed, the mission is assumed to have failed. Even though all machines are available and reliable, if the operator in one of the machines fails to detect the arrival of the task, or fails to perform the task or part of it correctly, or fails to perform the task within an allocated time, the mission is also assumed to have failed.

4.1.2 Formulation of System Effectiveness

Mission effectiveness for the system is defined in the same way as in Chapter 3. The system has three definitions of mission effectiveness depending upon how "no task arrivals" during the mission is dealt with. The detail of each definition is discussed in Section 3.1.2 of Chapter 3.

4.1.3 System Variables

Before developing the mathematical model, the variables involved in the system will be discussed. These are the task-arrival process, the system state, the system-design failure, and the human-operator performance variables. The system state has been discussed in Section 4.1.1.

4.1.3.1 Task-Arrival Process As discussed in Chapter 3, the number of tasks requested during the mission is assumed to follow a nonhomogeneous Poisson process. Since the operating state of the system is a time-dependent variable, the task arrival characterizes this variable.

4.1.3.2 System-Design Failure If a new task arrives during the performance of the current task, it may be undetected or ignored and is considered a failure because of the inadequacy of system design. To start performing the task, all machines and operators must be idle at the task arrival.

4.1.3.3 Human-Performance Variables All three human-performance variables introduced in single-machine systems are also included in the model proposed in this chapter. Because of human nature, each of these variables varies from one operator to another (see Figs. 4.3, 4.4, and 4.5). The probability that the human operator in machine n will detect the task arrival can be assumed to change as a function of time during the mission, and can be represented by

$$\Pr[X_n(t_i) = 1] = p_n(t_i),\ i = 1, 2, \ldots, N(T), \tag{4.1}$$

where $X_n(t_i)$ has the value of 0 if the operator at machine n fails to detect the task arrival and 1 if he or she succeeds; the probability of accurately performed the task at time t_i is assumed to change as a function of time during the mission, and can be represented by

$$\Pr[Y_n(t_i) = 1 | X_n(t_i) = 1] = q_n(t_i),\ i = 1, 2, \ldots, N(T), \tag{4.2}$$

where $Y_n(t_i)$ has a value of 0 if the operation is not accurate and 1 if the operation is accurate; and the probability that the task will be performed within an allocated time-limit δ can be formulated as:

$$\Pr[c_i \leq \delta | X_n(t_i) = 1] = \int_0^{\delta} g_n(c_i) dc_i \equiv s_n(t_i, \delta) \tag{4.3}$$

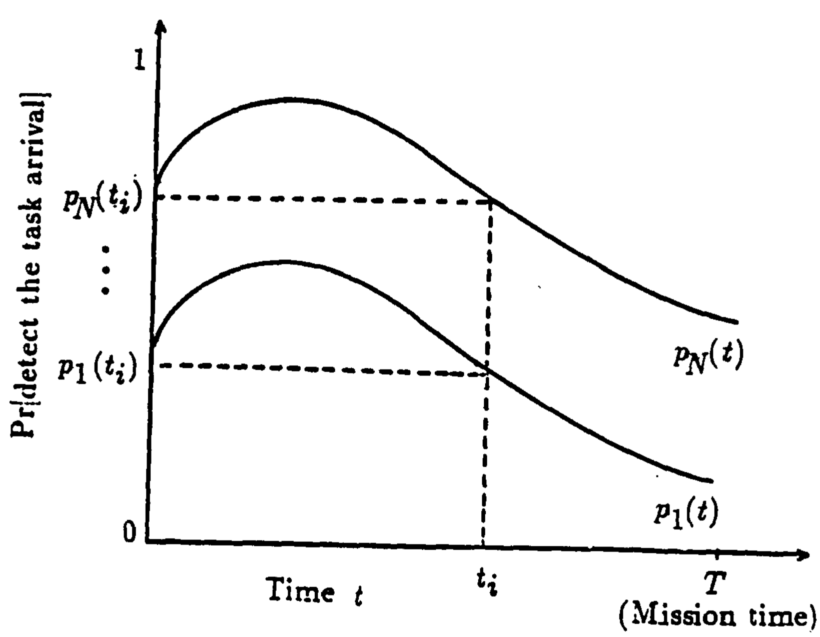

Figure 4.3: Typical probability functions for detection of a task arrival

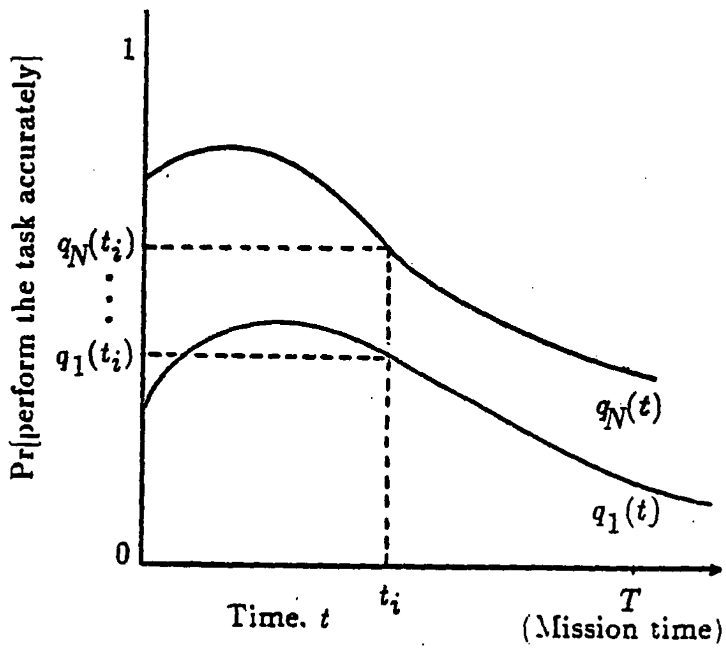

Figure 4.4: Typical probability functions for human-performance accuracy

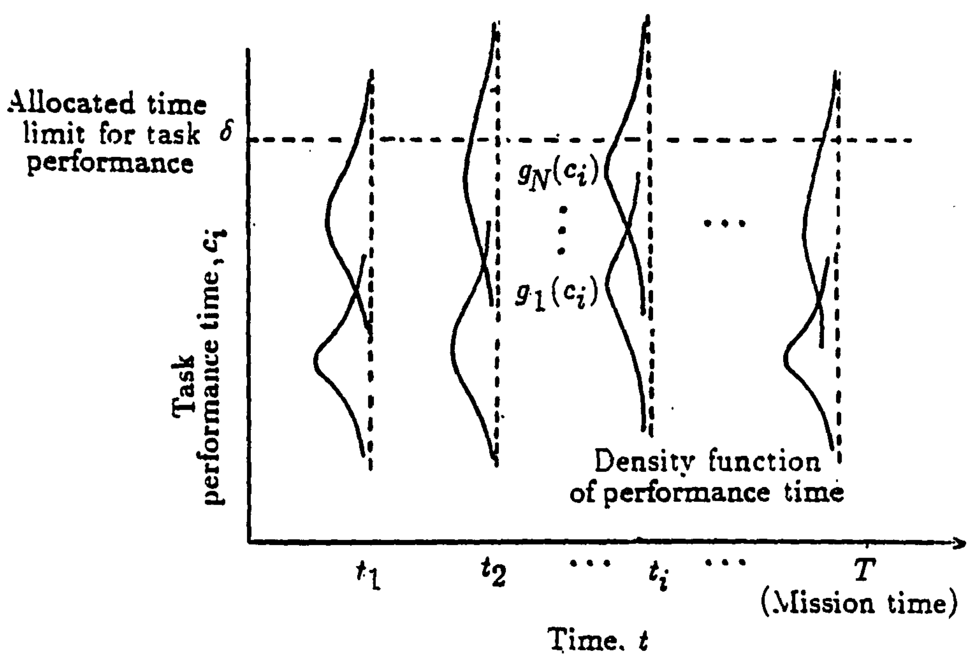

Figure 4.5: Probability-density functions for performance times

4.2 Development of Effectiveness Models

Let us define M_{in} as the event that the nth machine is available at time t_i as well as reliable during the allocated performance time δ; O_{in} as the event that the human operator at machine n detects the ith task and performs it accurately within an allowable time-limit δ; and S_{in} as the event that the task arriving at time t_i is successfully responded to by the nth machine, or $S_{in} = M_{in} \cap O_{in}$. Also define

$$\begin{aligned} \underline{t}_k &= (t_1, t_2, \ldots, t_k) \\ M_i^n &= \bigcap_{n=1}^{N} M_{in} \\ O_i^n &= \bigcap_{n=1}^{N} O_{in} \\ S_i^n &= \bigcap_{n=1}^{N} S_{in}. \end{aligned}$$

Now, given that $N(T) = 1$, system effectiveness can be formulated as follows:

$$\begin{aligned} SE(t_1) &= \Pr[S_1^n] \\ &= \Pr[M_1^n \cap O_1^n] \\ &= \Pr[M_1^n] \Pr[O_1^n | M_1^n], \end{aligned} \tag{4.4}$$

or for the successful performance of the task that arrives at time t_1, all machines should be available at time t_1 and reliable during the allocated performance-time δ. Conditional on these two events, the human operators should detect the task arrival and perform accurately within an allowable time-limit δ.

For the case where $N(T) = 2$, let us assume that task-arrival times are given as t_1 and t_2. If the second task arrives during the allocated performance-time of the first task, it is considered to be a failure, or if $t_1 + \delta > t_2$, of system effectiveness for the system is determined to be 0. If $t_1 + \delta < t_2$, system effectiveness is determined as a function of the success of each task. In this case

$$\begin{aligned}
SE(\underline{t}_2) &= \Pr[S_1^n \cap S_2^n] \\
&= \Pr[S_1^n] \Pr[S_2^n | S_1^n] \\
&= \Pr[S_1^n] \Pr[M_2^n \cap O_2^n | S_1^n] \\
&= \Pr[S_1^n] \Pr[S_1^n \cap M_2^n \cap O_2^n] / \Pr[S_1^n] \\
&= \Pr[S_1^n] \Pr[S_1^n] \Pr[M_2^n | S_1^n] \Pr[O_2^n | S_1^n \cap M_2^n] / \Pr[S_1^n] \\
&= \Pr[S_1^n] \Pr[M_2^n | S_1^n] \Pr[O_2^n | S_1^n \cap M_2^n] \\
&= \Pr[S_1^n] \Pr[M_2^n | M_1^n \cap S_1^n] \Pr[O_2^n | M_2^n] \\
&= \Pr[M_1^n] \Pr[O_1^n | M_1^n] \Pr[M_2^n | M_1^n] \Pr[O_2^n | M_2^n]. \qquad (4.5)
\end{aligned}$$

Now let us assume that task arrivals are given as $t_1, t_2, \ldots, t_{N(T)}$. Let us define two sets C and D such that

$$\begin{aligned}
C &= \left\{ \underline{t}_{N(T)} | 0 < t_1 < t_2 < \cdots < t_{N(T)} \leq T \right\} \\
D &= \left\{ \underline{t}_{N(T)} | 0 < t_1 + \delta < t_2, \cdots, t_{N(T)} + \delta \leq T \right\}.
\end{aligned}$$

With the task arrival time $\underline{t}_{N(T)} \in (C - D)$, the probability of a mission success is 0 because of a system-design failure. With the task-arrival time $\underline{t}_{N(T)} \in D$, system effectiveness is determined as a function of each task. Given $N(T) = k$, from Eqs. (4.4) and (4.5), system effectiveness can be generalized as

$$SE(\underline{t}_k) = \prod_{i=1}^{k} \Pr[M_i^n | M_{i-1}^n] \Pr[O_i^n | M_i^n], \qquad (4.6)$$

where $\Pr[M_0^n] \equiv 1$. Now system effectiveness can be determined by taking the expected value of the conditional probability of Eq. (4.6) with respect to the joint conditional-distribution of the arrival times, given $N(T) = k$, and finally by taking the expected value with respect to the Poisson distribution of $N(T)$. That is,

$$\begin{aligned} SE(T) &= E\left\{E\left[SE(\underline{t}_k)\right]\right\} + Po[0; m(T)](1) \\ &= \sum_{k=1}^{\infty} E\left[SE(\underline{t}_k)\right] Po[k; m(T)] + Po[0; m(T)]. \end{aligned} \tag{4.7}$$

Here

$$E\left[SE(\underline{t}_k)\right] = \int \underset{D}{\cdots} \int SE(\underline{t}_k) f_k(\underline{t}_k) dt_1 \ldots dt_k,$$

where $f_k(\underline{t}_k)$ is the joint probability-density function of task-arrival times $t_1, \ldots, t_k$, that is:

$$f_k(\underline{t}_k) = \frac{\prod_{i=1}^{k} \lambda(t_i)}{m(T)^k / k!}, \; 0 < t_1 < \cdots < t_k \leq T. \tag{4.8}$$

Finally, system effectiveness can be expressed as follows:

$$\begin{aligned} SE(T) &= \sum_{k=1}^{\infty} \left\{ \int \underset{D}{\cdots} \int \frac{SE(\underline{t}_k) \prod_{i=1}^{k} \lambda(t_i)}{m(T)^k / k!} dt_1 \right. \\ &\quad \left. \ldots dt_k \times Po[k; m(T)] \right\} + Po[0; m(T)]. \end{aligned} \tag{4.9}$$

Note that $SE(\underline{t}_k)$ is a function of availability, reliability, and operator performance at each task arrival. If we define A_i^n as the event that the N-machine system is available at time t_i; and if we define R_i^n as the event that the N-machine system is reliable during the allocated performance-time δ, given the availability at time t_i, then system effectiveness, given $N(T) = k$, can be rewritten as:

$$SE(\underline{t}_k) = \prod_{i=1}^{k} \Pr[A_i^n | M_{i-1}^n] \Pr[R_i^n | A_i^n \cap M_{i-1}^n] \Pr[O_i^n | M_i^n]$$

$$= \prod_{i=1}^{k} a(t_i^*) r(t_i, \delta) o(t_i, \delta). \tag{4.10}$$

By redefining, as in Chapter 3,

$$\begin{aligned} a(t_i^*) &\equiv \Pr[A_i^n | M_{i-1}^n] \\ r(t_i, \delta) &\equiv \Pr[R_i^n | A_i^n \cap M_{i-1}^n] \\ o(t_i, \delta) &\equiv \Pr[O_i^n | M_i^n], \end{aligned}$$

where $a(t_i^*)$ is the availability of the system at the ith task arrival, given that it is available at the end of the $(i-1)$th task performance; $r(t_i, \delta)$ is the reliability of the system during the ith task performance-time c_i, given that it is available when the task arrives; and $o(t_i, \delta)$ is the human-operator effect on the ith task performance-time: i.e., the probability that the human operators detect the ith task and perform it accurately within an allocated time limit δ given that the system is available and reliable for the task performance. System effectiveness can now can be rewritten as follows:

$$\begin{aligned} SE(T) &= \sum_{k=1}^{\infty} \left\{ \int \underset{D}{\cdots} \int \frac{\prod_{i=1}^{k} a(t_i^*) r(t_i, \delta) o(t_i, \delta) \lambda(t_i)}{m(T)^k / k!} dt_1 \right. \\ & \left. \quad \ldots dt_k \times P_o[k; m(T)] \right\} + P_o[0; m(T)]. \end{aligned} \tag{4.11}$$

The above definition of system effectiveness is based on the assumption that no task arrival during the mission is one definition of mission success. As discussed in Section 3.1.2, if the system requires the availability of the system even when there is no task request during the mission, then

$$\begin{aligned} SE(T) &= \sum_{k=1}^{\infty} \left\{ \int \underset{D}{\cdots} \int \frac{\prod_{i=1}^{k} a(t_i^*) r(t_i, \delta) o(t_i, \delta) \lambda(t_i)}{m(T)^k / k!} dt_1 \right. \\ & \left. \quad \ldots dt_k \times P_o[k; m(T)] \right\} + P_o[0; m(T)] \int_0^T A(t) dt / T. \end{aligned} \tag{4.12}$$

For some systems, missions can occur only if there are task arrivals. For such system,

$$SE(T) = \sum_{k=1}^{\infty} \left\{ \int \underset{D}{\cdots} \int \frac{\prod_{i=1}^{k} a(t_i^*) r(t_i, \delta) o(t_i, \delta) \lambda(t_i)}{m(T)^k / k!} dt_1 \ldots dt_k \times Po[k; m(T)] \right\} / \{1 - Po[0; m(T)]\} . \quad (4.13)$$

The next three sections are devoted to the evaluation of $a(t_i^*)$, $r(t_i, \delta)$, and $o(t_i, \delta)$.

4.3 Availability Derivation

Consider an N-machine R-component system with each component of each machine having a constant failure and repair rate, if they are repairable, and satisfying the following additional assumptions:

- The failures due to error of the rth component of the nth machine are statistically independent of each other, and each has an occurrence rate α_{rn}.
- The probability of two or more errors occurring simultaneously is negligible.
- The time to repair a failed system due to error of the rth component of the nth machine follows an exponential distribution with parameter β_{rn}.
- Failures and repairs of one component are statistically independent of both the failures and repairs of the other components.
- A failed system caused by any type of error is repaired back to its original operational state.

The probability that the system is available at time t_i, given that the system is available at the end of the $(i-1)$th task performance, can be written as follows:

$$
\begin{aligned}
a(t_i^*) &= \Pr[\bigcap_{n=1}^{N} \bigcap_{r=1}^{R} \left\{Z_{rn}(t_i) = 1\right\} | \\
&\quad \bigcap_{n=1}^{N} \bigcap_{r=1}^{R} \left\{Z_{rn}(t_{i-1}) = 1, T_f^{rn}(t_{i-1}) \geq \delta\right\}] \\
&= \Pr[\bigcap_{n=1}^{N} \bigcap_{r=1}^{R} \left\{Z_{rn}(t_i^*) = 1\right\}] \\
&= P_{00}(t_i^*),
\end{aligned}
\tag{4.14}
$$

where $Z_{rn}(t)$ is the indicator variable for the state of the rth component of the nth machine at time t (0 = *off*, 1 = *on*); $T_f^{rn}(t)$ is the time to failure of the rth component of the nth machine measured from time t; $P_{00}(t)$ is the probability that the system is operating (all components of all machines are operating) at time t; $t_i^* = t_i - t_{i-1} - \delta$, $i = 1, 2, \ldots, N(T)$; and $t_0 = 0$. This pointwise availability can be determined by solving the differential-difference equations corresponding to the rate diagram shown in Fig. 4.6. For simplification, the index i and the asterisk on t_i^* are suppressed throughout the derivation. The differential-difference equations corresponding to the system can be given as

$$
\frac{d}{dt} P_{00}(t) = \sum_{n=1}^{N} \sum_{r=1}^{R} \beta_{rn} P_{rn}(t) - \alpha_{..} P_{00}(t) \tag{4.15}
$$

$$
\frac{d}{dt} P_{rn}(t) = \alpha_{rn} P_{00}(t) - \beta_{rn} P_{rn}(t), \tag{4.16}
$$

where $\alpha_{..} = \sum_{r=1}^{R} \sum_{n=1}^{N} \alpha_{rn}$ and $P_{rn}(t)$ is the probability that the system is *off* at time t due to the rth component's error at the nth machine. Taking the Laplace

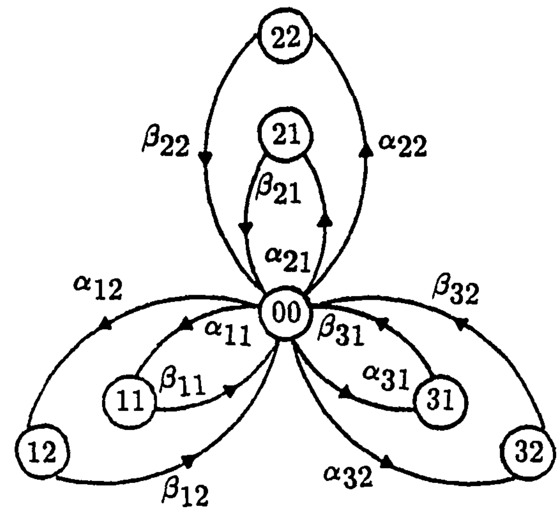

Figure 4.6: System rate diagram for $R = 3$ and $N = 2$

transforms of both sides of Eqs. (4.15) and (4.16), we obtain

$$s\overline{P}_{00}(s) - P_{00}(0) = \sum_{n=1}^{N}\sum_{r=1}^{R} \beta_{rn}\overline{P}_{rn}(s) - \alpha_{..}\overline{P}_{00}(s) \qquad (4.17)$$

$$s\overline{P}_{rn}(s) - P_{rn}(0) = \alpha_{rn}\overline{P}_{00}(s) - \beta_{rn}\overline{P}_{rn}(s), \qquad (4.18)$$

where $\overline{P}_x(s)$ is the Laplace transform of $P_x(t)$; s is a Laplace transform variable; and $P_{00}(0)$, $P_{rn}(0)$ are 1, 0, respectively, since the system is operating at time $t_i^* = 0$. Therefore, from Eq. (4.18), we have

$$\overline{P}_{rn}(s) = \left\{\frac{\alpha_{rn}}{s - \beta_{rn}}\right\}\overline{P}_{00}(s). \qquad (4.19)$$

By substituting $P_{00}(0) = 1$ and the values of $\overline{P}_{rn}(s)$ from Eq. (4.19) into Eq. (4.17), we get

$$s\overline{P}_{00}(s) - 1 = \sum_{n=1}^{N}\sum_{r=1}^{R}\left\{\frac{\alpha_{rn}\beta_s}{s + \beta_{rn}}\right\}\overline{P}_{00}(s) - \alpha_{..}\overline{P}_{00}(s)$$

$$= \sum_{n=1}^{N} \sum_{r=1}^{R} \left\{ \frac{\alpha_{rn}\beta s}{s+\beta_{rn}} - \alpha_{rn} \right\} \overline{P}_{00}(s)$$

$$= - \sum_{n=1}^{N} \sum_{r=1}^{R} \left\{ \frac{\alpha_{rn} s}{s+\beta_{rn}} \right\} \overline{P}_{00}(s), \quad (4.20)$$

so that

$$\overline{P}_{00}(s) = \left\{ s + \sum_{n=1}^{N} \sum_{r=1}^{R} \frac{\alpha_{rn} s}{s+\beta_{rn}} \right\}^{-1}. \quad (4.21)$$

This transform can be inverted to obtain the availability of the system, $a(t_i^*)$ or $P_{00}(t_i^*)$, for any set value of α_{rn} and β_{rn}.

4.4 Reliability Derivation

The conditional probability that the system will be functioning for an allocated task-performance time δ, given that the system operates at task-request time t_i, can be formulated as follows:

$$\begin{aligned} r(t_i,\delta) &= \Pr[\bigcap_{n=1}^{N} \bigcap_{r=1}^{R} \left\{ T_f^{rn}(t_i) \geq \delta \right\} \mid \\ &\quad \bigcap_{n=1}^{N} \bigcap_{r=1}^{R} \left\{ Z_{rn}(t_i) = 1, Z_{rn}(t_{i-1}) = 1, T_f^{rn}(t_{i-1}) \geq \delta \right\}] \\ &= \Pr[\bigcap_{n=1}^{N} \bigcap_{r=1}^{R} \left\{ T_f^{rn}(t_i) \geq \delta \right\} \mid \bigcap_{n=1}^{N} \bigcap_{r=1}^{R} \left\{ Z_{rn}(t_i) = 1 \right\}] \\ &= \prod_{n=1}^{N} \prod_{r=1}^{R} \Pr[T_f^{rn}(t_i) \geq \delta \mid Z_{rn}(t_i) = 1] \\ &= \prod_{n=1}^{N} \prod_{r=1}^{R} e^{-\alpha_{rn}\delta} \\ &= e^{-\alpha_{..}\delta}, \end{aligned} \quad (4.22)$$

where $\alpha_{..} = \sum_{n=1}^{N} \sum_{r=1}^{R} \alpha_{rn}$.

4.5 Quantification of Human Performance

Since it is given that the system operates at task-request time and is reliable during the allocated time, that the performance of each operator is independent of each other, and that the successful performance of each task is independent of the other tasks, the human operator effect on task performance can be formulated as follows:

$$
\begin{aligned}
o(t_i, \delta) &= Pr[\bigcap_{n=1}^{N} (Xn(t_i) = 1, Yn(t_i) = 1, c_i \leq \delta | M_{in})] \\
&= \Pr[\bigcap_{n=1}^{N} (Xn(t_i) = 1 | M_{in})] \\
&\quad \times \Pr[\bigcap_{n=1}^{N} (c_i \leq \delta | Xn(t_i) = 1, M_{in})] \\
&\quad \times \Pr[\bigcap_{n=1}^{N} (Yn(t_i) = 1 | Xn(t_i) = 1, M_{in})] \\
&= \prod_{n=1}^{N} pn(t_i) sn(t_i, \delta) qn(t_i). \qquad (4.23)
\end{aligned}
$$

The first term in Eq. (4.23) represents the probability of detecting the task arriving at time t_i; the second term represents the probability of completing the task within allocated time δ; the third term represents the probability of accurately performing the task under some given conditions.

4.6 Illustration

As an illustration, consider a 2-machine system where each machine consisting of man-hardware and software components with failure rates α_{r1},α_{r2} and repair rates β_{r1},β_{r2}, respectively, where $r = 1, 2$. For this system, Eq. 4.21 can be rewritten as:

$$\begin{aligned}
\overline{P}_{00}(s) &= \left\{ s + \sum_{n=1}^{2} \sum_{r=1}^{2} \frac{\alpha_{rn} s}{s + \beta_{rn}} \right\}^{-1} \\
&= \frac{(s+\beta_{11})(s+\beta_{12})(s+\beta_{21})(s+\beta_{22})}{s(s^4 + x_1 s^3 + x_2 s^2 + x_3 s + x_4)} \\
&= \frac{(s+\beta_{11})(s+\beta_{12})(s+\beta_{21})(s+\beta_{22})}{s(s-s_1)(s-s_2)(s-s_3)(s-s_4)},
\end{aligned} \tag{4.24}$$

where s_1, s_2, s_3 and s_4 are the roots of

$$s^4 + x_1 s^3 + x_2 s^2 + x_3 s + x_4 = 0$$

where

$$\begin{aligned}
x_1 &= \alpha_{11} + \beta_{11} + \alpha_{12} + \beta_{12} + \alpha_{21} + \beta_{21} + \alpha_{22} + \beta_{22} \\
x_2 &= \alpha_{11}\beta_{12} + \beta_{11}\alpha_{12} + \beta_{11}\beta_{12} + \alpha_{11}\beta_{21} + \beta_{11}\alpha_{21} + \beta_{11}\beta_{21} + \\
&\quad \alpha_{11}\beta_{22} + \beta_{11}\alpha_{22} + \beta_{11}\beta_{22} + \alpha_{12}\beta_{21} + \beta_{12}\alpha_{21} + \beta_{12}\beta_{21} + \\
&\quad \alpha_{12}\beta_{22} + \beta_{12}\alpha_{22} + \beta_{12}\beta_{22} + \alpha_{21}\beta_{22} + \beta_{21}\alpha_{22} + \beta_{21}\beta_{22} \\
x_3 &= \alpha_{11}\beta_{12}\beta_{21} + \beta_{11}\alpha_{12}\beta_{21} + \beta_{11}\beta_{12}\alpha_{21} + \beta_{11}\beta_{12}\beta_{21} + \\
&\quad \alpha_{11}\beta_{12}\beta_{22} + \beta_{11}\alpha_{12}\beta_{22} + \beta_{11}\beta_{12}\alpha_{22} + \beta_{11}\beta_{12}\beta_{22} + \\
&\quad \alpha_{11}\beta_{21}\beta_{22} + \beta_{11}\alpha_{12}\beta_{22} + \beta_{11}\beta_{12}\alpha_{22} + \beta_{11}\beta_{21}\beta_{22} + \\
&\quad \alpha_{12}\beta_{21}\beta_{22} + \beta_{12}\alpha_{21}\beta_{22} + \beta_{12}\beta_{21}\alpha_{22} + \beta_{12}\beta_{21}\beta_{22} \\
x_4 &= \alpha_{11}\beta_{11}\beta_{12}\beta_{21} + \beta_{11}\alpha_{12}\beta_{21}\beta_{22} + \beta_{11}\beta_{12}\alpha_{21}\beta_{22} + \\
&\quad \beta_{11}\beta_{12}\beta_{21}\alpha_{22} + \beta_{11}\beta_{12}\beta_{21}\beta_{22}.
\end{aligned}$$

On inverting Eq. (4.24), we have

$$a(t_i^*) = X_0 + X_1 e^{s_1 t_i^*} + X_2 e^{s_2 t_i^*} + X_3 e^{s_3 t_i^*} + X_4 e^{s_4 t_i^*}, \tag{4.25}$$

where

$$
\begin{aligned}
X_0 &= \frac{\beta_{11}\beta_{12}\beta_{21}\beta_{22}}{s_1 s_2 s_3 s_4} \\
X_1 &= \frac{(\beta_{11}+s_1)(\beta_{12}+s_1)(\beta_{21}+s_1)(\beta_{22}+s_1)}{s_1(s_1-s_2)(s_1-s_3)(s_1-s_4)} \\
X_2 &= \frac{(\beta_{11}+s_2)(\beta_{12}+s_2)(\beta_{21}+s_2)(\beta_{22}+s_2)}{s_2(s_2-s_1)(s_2-s_3)(s_2-s_4)} \\
X_3 &= \frac{(\beta_{11}+s_3)(\beta_{12}+s_3)(\beta_{21}+s_3)(\beta_{22}+s_3)}{s_3(s_3-s_1)(s_3-s_2)(s_3-s_4)} \\
X_4 &= \frac{(\beta_{11}+s_4)(\beta_{12}+s_4)(\beta_{21}+s_4)(\beta_{22}+s_4)}{s_4(s_4-s_1)(s_4-s_2)(s_4-s_3)}.
\end{aligned}
$$

If $\beta_{r1} = \beta_{r2} = \beta_r$, then Eq. (4.21) becomes

$$\overline{P}_{00}(s) = \left\{ s + \sum_{r=1}^{R} \frac{\alpha_{r.} s}{s+\beta_r} \right\}^{-1}, \qquad (4.26)$$

where $\alpha_{r.} = \alpha_{r1} + \alpha_{r2}$. On inverting Eq. (4.26) we have

$$a(t_i^*) = X_0 + X_1 e^{s_1 t_i^*} + X_2 e^{s_2 t_i^*}, \qquad (4.27)$$

where

$$
\begin{aligned}
X_0 &= \frac{\beta_1\beta_2}{s_1 s_2} \\
X_1 &= \frac{(s_1+\beta_1)(s_1+\beta_2)}{s_1(s_1-s_2)} \\
X_2 &= \frac{(s_2+\beta_1)(s_2+\beta_2)}{s_2(s_2-s_1)} \\
s_1 &= \frac{-x_1 + \sqrt{x_1^2 - 4x_2}}{2} \\
s_2 &= \frac{-x_1 - \sqrt{x_1^2 - 4x_2}}{2} \\
x_1 &= \beta_1 + \beta_2 + \alpha_{1.} + \alpha_{2.} \\
x_2 &= \beta_1\beta_2 + \alpha_{1.}\beta_2 + \alpha_{2.}\beta_2.
\end{aligned}
$$

For this system,

$$r(t_i,\delta)o(t_i,\delta) = e^{-\alpha_{..}\delta}\left\{\prod_{n=1}^{2} p_n(t_i)s_n(t_i,\delta)q_n(t_i)\right\}, \qquad (4.28)$$

where the first term is from Eq. (4.22) and the second term is from Eq. (4.23). System effectiveness can be evaluated by substituting Eqs. (4.25) or (4.27) and (4.28) into Eqs. (4.11), (4.12), and (4.13) depending upon the definition of the system being investigated.

5 MODEL EXTENSIONS

5.1 Systems with Multiple Operating Modes

In some cases, the system has more than two operating states. These states or operating levels may be classified as being excellent, good, fair, poor, or failed. These operating levels will affect the task performance. Generally, the human operator is expected to perform better under excellent operating conditions than under fair conditions, for example. In this section we illustrate how the model proposed in this study can be modified to handle such problems.

Let us consider a single-machine, 3-component system, i.e., the system composed of hardware, software and a human operator. The system starts the operation in "excellent" condition (all component are fresh). When the system fails due to hardware error or due to software error, no attempt is made to repair the system; the system will be "off" and the mission is terminated. When the system fails due to human error, a repairman is sent, however, this repairman is not expected to fully remove the error. The system in this "fair" condition is put back in operation. If the system fails, again due to human error, a more experienced repairman is hired to bring back the system into its original operating condition. The system is required to perform a number of tasks that randomly arrive following a nonhomogeneous Poisson process. All assumptions regarding the condition that is needed in order for the mission to be successful that were applied in the previous chapters are also applied here with some modifications. The modifications will be explained when they arise in the derivation. The following additional assumptions are associated with the model:

- Given that the system is operating at level j (1=excellent, 2=fair), the failures due to error in component r (1=hardware, 2=software, 3=operator) are statistically independent of each other and have an occurrence rate α_{rj}.
- The probability of two or more software, hardware, or operator errors occurring simultaneously is negligible.
- Given that the system is at level j before the failure, the time to remove an error in component r follows an exponential distribution with parameter β_{rj}.
- Failures and repairs of one component, if attempts are made, are statistically independent of both the failures and repairs of the other components.

For each task to be successfully performed, the system should be available at each task arrival and be able to operate during the allocated performance-time δ. Under this assumption, the probability that the system is available at level j_i at time t_i, given that it was available at level j_{i-1} at the end of the performance of the $(i-1)$th, can be written as follows:

$$\begin{aligned} a_{j_i|j_{i-1}}(t_i^*) &= \Pr[Z_s(t_i) = j_i | Z_s(t_{i-1}) = j_{i-1}, T_f^s(t_{i-1}) \geq \delta] \\ &= \Pr[Z_s(t_i^*) = j_i | Z_s(0) = j_{i-1}], \end{aligned} \tag{5.1}$$

where $Z_s(t)$ is a random variable representing the state of the system at time t and $T_f^s(t)$ is a random variable representing the time to failure of the system,

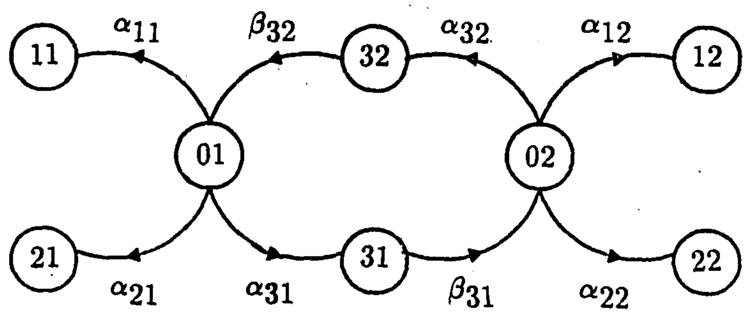

Figure 5.1: System rate diagram

measured from time t. This pointwise availability can be determined by solving the differential-difference equations corresponding to the system shown in Fig. 5.1. The differential-difference equations corresponding to the system can be given as

$$\frac{d}{dt}P_{01}(t) = \beta_{32}P_{32}(t) - \alpha_{.1}P_{01}(t) \tag{5.2}$$

$$\frac{d}{dt}P_{r1}(t) = \alpha_{r1}P_{01}(t) \tag{5.3}$$

$$\frac{d}{dt}P_{02}(t) = \beta_{31}P_{31}(t) - \alpha_{.2}P_{02}(t) \tag{5.4}$$

$$\frac{d}{dt}P_{r2}(t) = \alpha_{r2}P_{02}(t), \tag{5.5}$$

where $P_{0j}(t_i^*)$ is the probability that the system is *on* at level j at time t_i^* and $P_{rj}(t_i^*)$ is the probability that the system is *off* at time t_i^* after operating at level j; and where $\alpha_{.j} = \alpha_{1j} + \alpha_{2j} + \alpha_{3j}$, $j = 1, 2$. For simplification, the index i and the asterisk on t_i^* is suppressed throughout the derivation. Note that, by the definition of the system,

$$\Pr[Z_s(t) = 1] = P_{01}(t) \tag{5.6}$$

$$\Pr[Z_s(t) = 2] = P_{02}(t) \tag{5.7}$$

$$\Pr[Z_s(t) = 3] = 1 - P_{01}(t) - P_{02}(t). \tag{5.8}$$

Taking the Laplace transforms of both sides of Eq. (5.2) through (5.5), we obtain

$$s\overline{P}_{01}(s) - P_{01}(0) = \beta_{32}\overline{P}_{32}(s) - \alpha_{.1}\overline{P}_{01}(s) \tag{5.9}$$

$$s\overline{P}_{r1}(s) - P_{r1}(0) = \alpha_{r1}\overline{P}_{01}(s) \tag{5.10}$$

$$s\overline{P}_{02}(s) - P_{02}(0) = \beta_{31}\overline{P}_{31}(s) - \alpha_{.2}\overline{P}_{02}(s) \tag{5.11}$$

$$s\overline{P}_{r2}(s) - P_{r2}(0) = \alpha_{r2}\overline{P}_{02}(s). \tag{5.12}$$

where $\overline{P}_x(s)$ is the Laplace transform of $P_x(t)$, and s is a Laplace transform variable. To evaluate $a_{j_i|1}(t_i^*)$, assume that the system is operating at level 1 at time $t_i^* = 0$, or $P_{01}(0)$, $P_{02}(0)$ and $P_{rj}(0)$ are 1, 0, and 0 respectively.

Under this condition, from Eqs. (5.10) and (5.12), we have

$$\overline{P}_{r1}(s) = \left\{\frac{\alpha_{r1}}{s}\right\}\overline{P}_{01}(s) \tag{5.13}$$

$$\overline{P}_{r2}(s) = \left\{\frac{\alpha_{r2}}{s}\right\}\overline{P}_{02}(s). \tag{5.14}$$

By substituting $P_{01}(0) = 1$ and the values of $\overline{P}_{32}(s)$ from Eq. (5.14) into Eq. (5.9), we get

$$s\overline{P}_{01}(s) - 1 = \left\{\frac{\alpha_{32}\beta_{32}}{s+\beta_{32}}\right\}\overline{P}_{02}(s) - \alpha_{.1}\overline{P}_{01}(s); \tag{5.15}$$

by substituting $P_{02}(0) = 0$ and the values of $\overline{P}_{31}(s)$ from Eq. (5.13) into Eq. (5.11), we get

$$s\overline{P}_{01}(s) = \left\{\frac{\alpha_{31}\beta_{31}}{s+\beta_{31}}\right\}\overline{P}_{01}(s) - \alpha_{.2}\overline{P}_{02}(s). \tag{5.16}$$

Solving Eqs. (5.15) and (5.16) simultaneously for $P_{01}(s)$ and $P_{02}(s)$ yields

$$\begin{aligned}\overline{P}_{01}(s) &= \frac{(s+\beta_{32})(s+\beta_{31})(s+\alpha_{.2})}{s(s^3+x_1s^2+x_2s+x_3)}\\ &= \frac{(s+\beta_{32})(s+\beta_{31})(s+\alpha_{.2})}{s(s-s_1)(s-s_2)(s-s_3)} \qquad (5.17)\\ \overline{P}_{02}(s) &= \frac{\alpha_{31}\beta_{31}(s+\beta_{32})}{s(s^3+x_1s^2+x_2s+x_3)}\\ &= \frac{\alpha_{31}\beta_{31}(s+\beta_{32})}{s(s-s_1)(s-s_2)(s-s_3)}, \qquad (5.18)\end{aligned}$$

where s_1, s_2 and s_3 are the roots of

$$s^3+x_1s^2+x_2s+x_3=0,$$

where, by defining $\alpha_{..}=\alpha_{.1}+\alpha_{.2}$, $\beta_{3.}=\beta_{31}+\beta_{32}$, and

$$\begin{aligned}x_1 &= \alpha_{..}+\beta_{3.}\\ x_2 &= \alpha_{.1}(\alpha_{.2}+\beta_{3.})+\alpha_{.2}\beta_{3.}+\beta_{31}\beta_{32}\\ x_3 &= \alpha_{.1}\alpha_{.2}\beta_{3.}+\alpha_{..}\beta_{31}\beta_{32}.\end{aligned}$$

By inverting Eqs. (5.17) and (5.18), we have

$$a_{1|1}(t_i^*) = X_{01}+X_{11}e^{s_1t_i^*}+X_{21}e^{s_2t_i^*}+X_{31}e^{s_3t_i^*} \qquad (5.19)$$

$$a_{2|1}(t_i^*) = X_{02}+X_{12}e^{s_1t_i^*}+X_{22}e^{s_2t_i^*}+X_{32}e^{s_3t_i^*}, \qquad (5.20)$$

where

$$\begin{aligned}X_{01} &= -\frac{\beta_{32}\beta_{31}\alpha_{.2}}{s_1s_2s_3}\\ X_{11} &= \frac{(\beta_{31}+s_1)(\beta_{32}+s_1)(\alpha_{.2}+s_1)}{s_1(s_1-s_2)(s_1-s_3)}\\ X_{21} &= \frac{(\beta_{31}+s_2)(\beta_{32}+s_2)(\alpha_{.2}+s_2)}{s_2(s_2-s_1)(s_2-s_3)}\end{aligned}$$

$$X_{31} = \frac{(\beta_{31}+s_3)(\beta_{32}+s_3)(\alpha_{.2}+s_3)}{s_3(s_3-s_1)(s_3-s_2)}$$
$$X_{02} = -\frac{\alpha_{31}\beta_{31}\beta_{32}}{s_1 s_2 s_3}$$
$$X_{12} = \frac{\alpha_{31}\beta_{31}(\beta_{32}+s_1)}{s_1(s_1-s_2)(s_1-s_3)}$$
$$X_{22} = \frac{\alpha_{31}\beta_{31}(\beta_{32}+s_2)}{s_2(s_2-s_1)(s_2-s_3)}$$
$$X_{32} = \frac{\alpha_{31}\beta_{31}(\beta_{32}+s_3)}{s_3(s_3-s_1)(s_3-s_2)}.$$

Now we drive the availability for the case that the system is operating at level 2 at time $t_i^* = 0$ or $P_{01}(0) = 0$, $P_{02}(0) = 1$ and $P_{rj}(0) = 0$. Because of the structure of the transition diagram in Fig. 5.1, the solutions can be obtained directly from the solution for the first case by interchanging the index j of α_{rj} and β_{rj} from 1 to 2 and vice versa in the results for the previous case. That is,

$$a_{1|2}(t_i^*) = Y_{01} + Y_{11}e^{u_1 t_i^*} + Y_{21}e^{u_2 t_i^*} + Y_{31}e^{u_3 t_i^*} \quad (5.21)$$
$$a_{2|2}(t_i^*) = Y_{02} + Y_{12}e^{u_1 t_i^*} + Y_{22}e^{u_2 t_i^*} + Y_{32}e^{u_3 t_i^*}, \quad (5.22)$$

where

$$Y_{01} = -\frac{\beta_{31}\beta_{32}\alpha_{.1}}{u_1 u_2 u_3}$$
$$Y_{11} = \frac{(\beta_{31}+u_1)(\beta_{32}+u_1)(\alpha_{.1}+u_1)}{u_1(u_1-u_2)(u_1-u_3)}$$
$$Y_{21} = \frac{(\beta_{31}+u_2)(\beta_{32}+u_2)(\alpha_{.1}+u_2)}{u_2(u_2-u_1)(u_2-u_3)}$$
$$Y_{31} = \frac{(\beta_{31}+u_3)(\beta_{32}+u_3)(\alpha_{.1}+u_3)}{u_3(u_3-u_1)(u_3-u_2)}$$
$$Y_{02} = -\frac{\alpha_{32}\beta_{31}\beta_{32}}{u_1 u_2 u_3}$$
$$Y_{12} = \frac{\alpha_{32}\beta_{32}(\beta_{31}+u_1)}{u_1(u_1-u_2)(u_1-u_3)}$$

$$Y_{22} = \frac{\alpha_{32}\beta_{32}(\beta_{31}+u_2)}{u_2(u_2-u_1)(u_2-u_3)}$$
$$Y_{32} = \frac{\alpha_{32}\beta_{32}(\beta_{31}+u_3)}{u_3(u_3-u_1)(u_3-u_2)};$$

and u_1, u_2 and u_3 are the roots of

$$u^3 + y_1u^2 + y_2s + y_3 = 0,$$

where

$$y_1 = \alpha_{..} + \beta_{3.}$$
$$y_2 = \alpha_{.2}(\alpha_{.1} + \beta_{3.}) + \alpha_{.1}\beta_{3.} + \beta_{31}\beta_{32}$$
$$y_3 = \alpha_{.2}\alpha_{.1}\beta_{3.} + \alpha_{..}\beta_{31}\beta_{32}.$$

In order to derive system reliability, let us assume for each task to be successfully performed, that the system should be ready both to function at the time of the task arrival and to operate during the allocated performance-time δ. Under this assumption, the conditional probability that the system will be functioning for task performance-time δ, given that the system operates at task request-time t_i, can be formulated as follows:

$$\begin{aligned} r_{j_i|j_{i-1}}(t_i,\delta) &= \Pr[\bigcap_{r=1}^{3}\left\{T_f^r(t_i) \geq \delta\right\} \mid \\ &\quad \bigcap_{r=1}^{3}\left\{Z_r(t_i) = j_i, Z_r(t_{i-1}) = j_{i-1}, T_f^r(t_i) \geq \delta\right\}] \\ &= \Pr[\bigcap_{r=1}^{3}\left\{T_f^r(t_i) \geq \delta\right\} \mid \bigcap_{r=1}^{3}\left\{Z_r(t_i) = j_i\right\}] \\ &= \prod_{r=1}^{3} \Pr[T_f^r(t_i) \geq \delta | Z_r(t_i) = j_i] \\ &= e^{-\alpha_{\cdot j_i}\delta}. \end{aligned} \qquad (5.23)$$

Note that because of the memoryless property of exponential distribution, the reliability is characterized only by the current level of operation; that is,

$$r_{1|1}(t_i,\delta) = r_{1|2}(t_i,\delta) = r_1(t_i,\delta) = \mathrm{e}^{-\alpha.1^{\delta}} \tag{5.24}$$

$$r_{2|1}(t_i,\delta) = r_{2|2}(t_i,\delta) = r_2(t_i,\delta) = \mathrm{e}^{-\alpha.2^{\delta}} \tag{5.25}$$

To evaluate the human operator effect, it is assumed that only the current level of the operating mode affects the task performance. Under this assumption, given that the system operates at level j_i when the ith arrives, the human operator effect can be formulated as follows:

$$\begin{aligned}
o_{j_i}(t_i,\delta) &= \Pr[X_{j_i}(t_i)=1, Y_{j_i}(t_i)=1, c_i \le \delta | Z_s(t_i)=j_i, T_f^s(t_i) \ge \delta] \\
&= \Pr[X_{j_i}(t_i)=1 | Z_s(t_i)=j_i, T_f^s(t_i) \ge \delta] \\
&\quad \times \Pr[c_i \le \delta | X_{j_i}(t_i)=1, Z_s(t_i)=j_i, T_f^s(t_i) \ge \delta] \\
&\quad \times \Pr[Y_{j_i}(t_i)=1 | X_{j_i}(t_i)=1, Z_s(t_i)=j_i, T_f^s(t_i) \ge \delta] \\
&= p_{j_i}(t_i) s_{j_i}(t_i,\delta) q_{j_i}(t_i).
\end{aligned} \tag{5.26}$$

The first term in Eq. (5.26) represents the probability of detecting the task arriving at time t_i; the second term represents the probability of completing the task within allocated time δ; the third term represents the probability of accurately performing the task under some given conditions (see Figs. 5.2, 5.3, and 5.4).

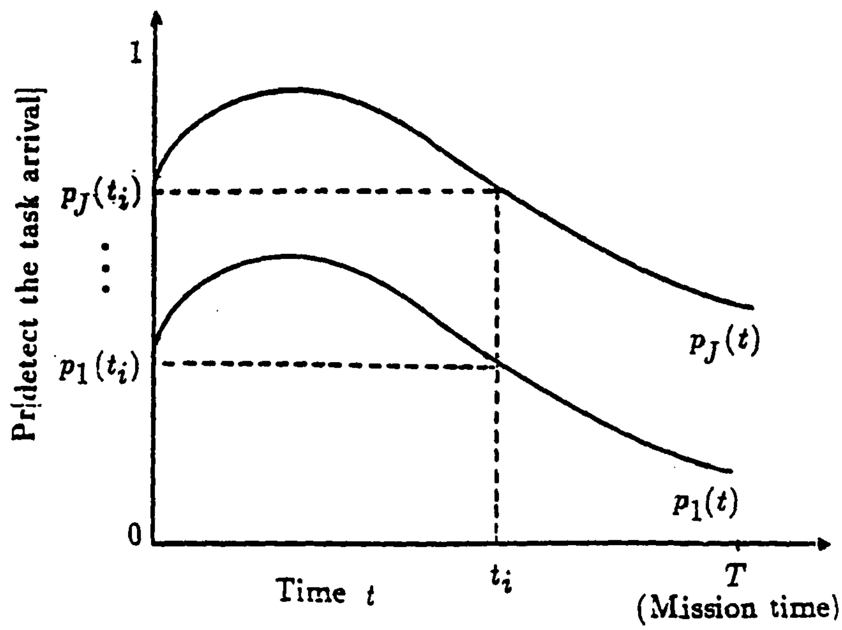

Figure 5.2: Typical probability function for detection of a task arrival

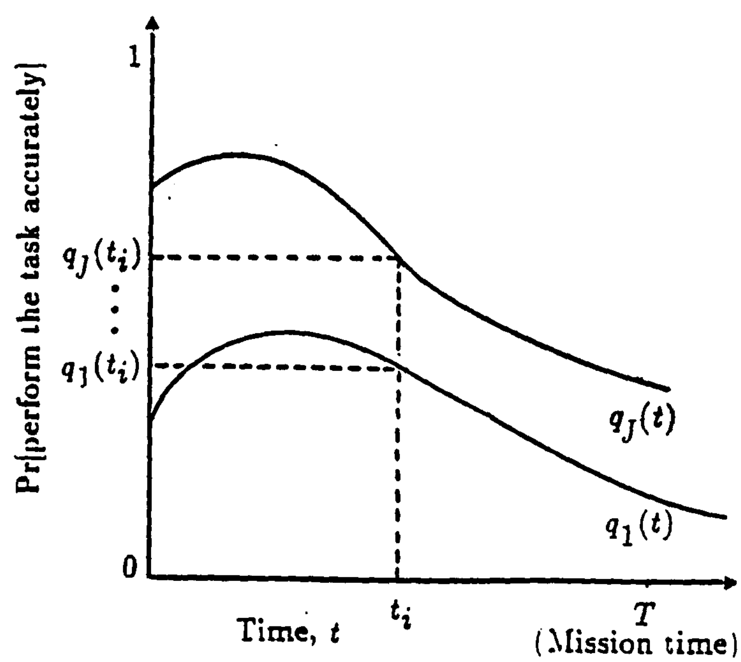

Figure 5.3: Typical probability function for human-performance accuracy

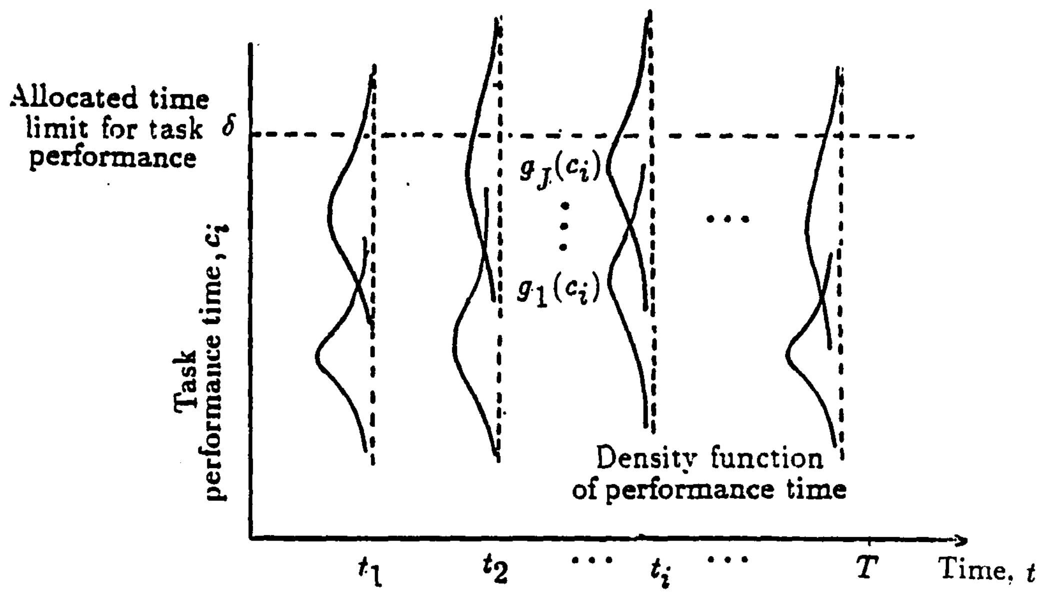

Figure 5.4: Probability-density function for performance times

Now, given that there is no-system design failure during the mission, and considering all possibilities of levels of performance at each task arrival, system effectiveness–given $N(T) = k$–can be written as

$$
\begin{aligned}
SE(\underline{t}_k) &= \sum_{j_1=1}^{2} \cdots \sum_{j_k=1}^{2} \left\{ \prod_{i=1}^{k} a_{j_i|j_{i-1}}(t_i^*) r_{j_i}(t_i, \delta) o_{j_i}(t_i, \delta) \right\} \\
&= \sum_{j_1=1}^{2} \cdots \sum_{j_k=1}^{2} \left\{ \prod_{i=1}^{k} SE_{j_i|j_{i-1}}(t_i^*, t_i, \delta) \right\}, \qquad (5.27)
\end{aligned}
$$

by redefining

$$SE_{j_i|j_{i-1}}(t_i^*, t_i, \delta) = a_{j_i|j_{i-1}}(t_i^*) r_{j_i}(t_i, \delta) o_{j_i}(t_i, \delta).$$

Overall system effectiveness can be written as

$$SE_1(T) = \sum_{k=1}^{\infty} \left\{ \int \underset{D}{\cdots} \int \frac{\sum_{j_1=1}^{2} \cdots \sum_{j_k=1}^{2} \left\{ \prod_{i=1}^{k} SE_{j_i|j_{i-1}}(t_i^*, t_i, \delta) \right\} \lambda(t_i)}{m(T)^k/k!} dt_1 \right.$$
$$\left. \ldots dt_k \times Po[k; m(T)] \right\} + Po[0; m(T)] \qquad (5.28)$$

$$SE_2(T) = \sum_{k=1}^{\infty} \left\{ \int \underset{D}{\cdots} \int \frac{\sum_{j_1=1}^{2} \cdots \sum_{j_k=1}^{2} \left\{ \prod_{i=1}^{k} SE_{j_i|j_{i-1}}(t_i^*, t_i, \delta) \right\} \lambda(t_i)}{m(T)^k/k!} dt_1 \right.$$
$$\left. \ldots dt_k \times Po[k; m(T)] \right\} + Po[0; m(T)] \int_0^T A(t)dt/T \qquad (5.29)$$

$$SE_3(T) = \sum_{k=1}^{\infty} \left\{ \int \underset{D}{\cdots} \int \frac{\sum_{j_1=1}^{2} \cdots \sum_{j_k=1}^{2} \left\{ \prod_{i=1}^{k} SE_{j_i|j_{i-1}}(t_i^*, t_i, \delta) \right\} \lambda(t_i)}{m(T)^k/k!} dt_1 \right.$$
$$\left. \ldots dt_k \times Po[k; m(T)] \right\} / \left\{ 1 - Po[0; m(T)] \right\}, \qquad (5.30)$$

where $SE_1(T)$ is system effectiveness when no task is considered as a mission success; $SE_2(T)$ is system effectiveness when the availability is required even when there is no task; and $SE_3(T)$ is system effectiveness when missions can occur only if there are task requests.

As an illustration, for $N(T) = 2$,

$$\begin{aligned}
SE(t_2) &= \sum_{j_1=1}^{2} \sum_{j_2=1}^{2} \left\{ \prod_{i=1}^{2} SE_{j_1|j_{i-1}}(t_i^*, t_i, \delta) \right\} \\
&= \sum_{j_1=1}^{2} \sum_{j_2=1}^{2} \left\{ SE_{j_1|1}(t_1, \delta) SE_{j_2|j_1}(t_2^*, t_2, \delta) \right\} \\
&= \sum_{j_2=1}^{2} \left\{ SE_{1|1}(t_1, \delta) SE_{j_2|1}(t_2^*, t_2, \delta) + SE_{2|1}(t_1, \delta) SE_{j_2|2}(t_2^*, t_2, \delta) \right\} \\
&= SE_{1|1}(t_1, \delta) SE_{1|1}(t_2^*, t_2, \delta) + SE_{2|1}(t_1, \delta) SE_{1|2}(t_2^*, t_2, \delta) + \\
&\quad SE_{1|1}(t_1, \delta) SE_{2|1}(t_2^*, t_2, \delta) + SE_{2|1}(t_1, \delta) SE_{2|2}(t_2^*, t_2, \delta).
\end{aligned}$$

The model proposed can be extended without any difficulty to handle systems with more than 3 levels of operating mode.

5.2 Systems with Several Task Types

The model proposed can be extended to handle a system required to carry out several types of tasks during the mission time. Each type of task is characterized by the performance-level of the human operator. The human performance level is introduced in order to represent the degree of accomplishment of a specified task. The term "performance" is meant in its broadest sense and includes the total set of usable outputs. Thus, performance may vary anywhere from the maximum potential performance of which the operator is capable under the most favorable conditions to total nonperformance.

Let us consider a single machine with a human operator operating the unit. This system is required to destroy a number of various tasks (targets) that arrive randomly following a nonhomogeneous Poisson process with parameter $\lambda(t)$. In order to perform the task, the system must be both available at the task arrival and reliable during an allocated time performance δ. Within each of the task type, various task performance levels of the human operator are defined. A probability distribution is defined to represent the degree of accuracy, which is assumed to change over time during the mission. The purpose of each task is to hit the target and stop its operation completely. Task success or failure is determined by the accuracy of firing. Generally, the probability of task success can be expressed as a function of the task-performance level.

Let $Z(t)$ denote a random variable representing the type of task arriving at time t which having values $1, 2, \ldots, m, \ldots, M$. Also let $[W(t)|Z(t) = m]$ denote the task-performance level at time t, given that the arriving task belongs to the mth type, where $W(t)$ is defined to represent the discrete state of the task-performance level at time t. Assume $W(t)$ has L different values, as follows:

$$
\begin{array}{rl}
1 & \text{perfectly done} \\
\left.\begin{array}{r} 2 \\ \vdots \\ l \\ \vdots \\ L-1 \end{array}\right\} & \text{partially done} \\
L & \text{failed.}
\end{array}
$$

For example, if the task requires that the human operator hit the target, various task-performance levels can be discretely defined, depending upon the accuracy of doing the task. As shown in Figure 5.5, if the target is hit within the circle C_1, it can be considered as "perfectly done" ($W(t) = 1$); if it falls between circles C_1 and C_2, "fairly well done"; and if it falls outside of the circle C_2 ($W(t) = 2$), "failed" ($W(t) = 3$). The probability that the task performance of the human operator is at level l, given that the task is of type m, may change with time during the mission (see Fig. 5.6), and at each task-arrival time t_i,

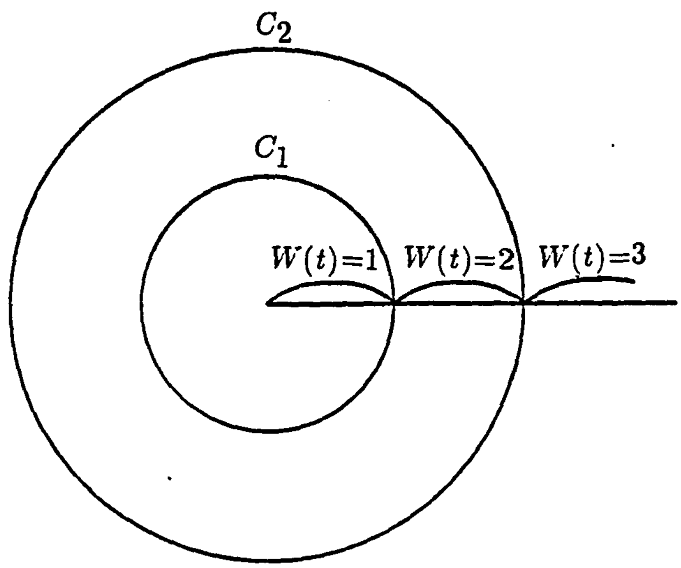

Figure 5.5: Task Performance levels in target example

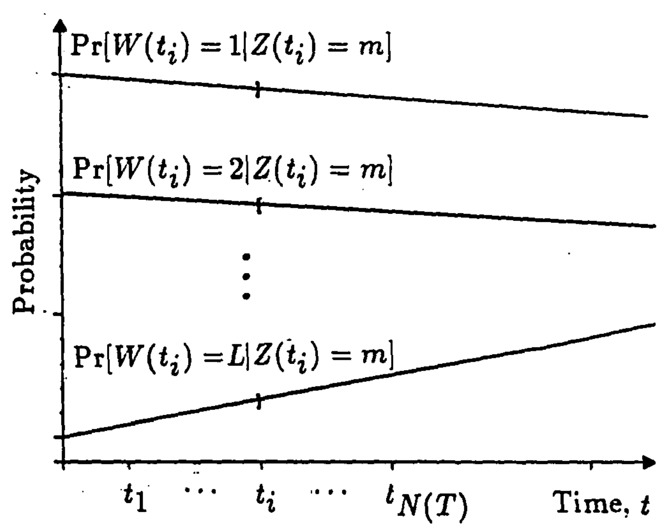

Figure 5.6: Probability of performing at level l when the task is type m

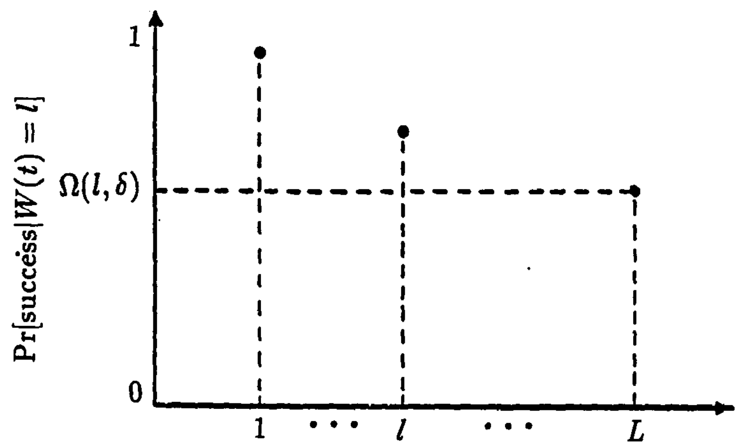

Figure 5.7: Probability of task success, given task performance level l

$$\sum_{l=1}^{L} \Pr[W(t_i) = l | Z(t_i) = m] = 1, l = 1, 2, \ldots, L. \tag{5.31}$$

Success or failure of the task is determined by the task performance level of the human operator. Let $\Omega(l, \delta)$ denote the probability of the task's being successfully performed within an allocated time δ, given a task performance level l (see Fig. 5.7). That is,

$$\Omega(l, \delta) = \Pr[\text{success} | W(t_i) = l]. \tag{5.32}$$

In the target analogy, a successful task is to hit the target and to stop its operation completely. As the distance between the center of the target and the location that the spot actually hits grows bigger, the chance for task success becomes smaller.

The probability of successfully performing task of type m can be express as:

$$o_m(t_i, \delta) = \sum_{l=1}^{L} \Pr[W(t_i) = l | Z(t_i) = m] \Omega(l, \delta). \quad (5.33)$$

Given that the system is available at time t_i and reliable during the allocated performance-time, the probability of task success can be expresses as

$$o(t_i, \delta) = \sum_{m=1}^{M} \left\{ \sum_{l=1}^{L} \Pr[W(t_i) = l | Z(t_i) = m] \Omega(l, \delta) \right\} \Phi_m(t_i), \quad (5.34)$$

where $\Phi_m(t_i) = \Pr[Z(t_i) = m]$. Now, the same procedure outlined in Chapter 3 for determining overall mission-effectiveness $SE(T)$ can be adopted. That is,

$$\begin{aligned} SE_1(T) &= \sum_{k=1}^{\infty} \left\{ \int \dot{\underset{D}{\cdot}} \int \frac{\prod_{i=1}^{k} a(t_i^*) r(t_i, \delta) \sum_{m=1}^{M} o_m(t_i, \delta) \Phi_m(t_i) \lambda(t_i)}{m(T)^k / k!} dt_1 \right. \\ & \left. \ldots dt_k \times P_o[k; m(T)] \right\} + P_o[0; m(T)] \end{aligned} \quad (5.35)$$

$$\begin{aligned} SE_2(T) &= \sum_{k=1}^{\infty} \left\{ \int \dot{\underset{D}{\cdot}} \int \frac{\prod_{i=1}^{k} a(t_i^*) r(t_i, \delta) \sum_{m=1}^{M} o_m(t_i, \delta) \Phi_m(t_i) \lambda(t_i)}{m(T)^k / k!} dt_1 \right. \\ & \left. \ldots dt_k \times P_o[k; m(T)] \right\} + P_o[0; m(T)] \int_0^T A(t) dt / T \end{aligned} \quad (5.36)$$

$$\begin{aligned} SE_3(T) &= \sum_{k=1}^{\infty} \left\{ \int \dot{\underset{D}{\cdot}} \int \frac{\prod_{i=1}^{k} a(t_i^*) r(t_i, \delta) \sum_{m=1}^{M} o_m(t_i, \delta) \Phi_m(t_i) \lambda(t_i)}{m(T)^k / k!} dt_1 \right. \\ & \left. \ldots dt_k \times P_o[k; m(T)] \right\} / \{1 - P_o[0; m(T)]\}, \end{aligned} \quad (5.37)$$

where $SE_1(T)$ is system effectiveness when no task is considered as a mission success; $SE_2(T)$ is system effectiveness when the availability is required even though there is no task; and $SE_3(T)$ is system effectiveness when missions can occur only if there are task requests.

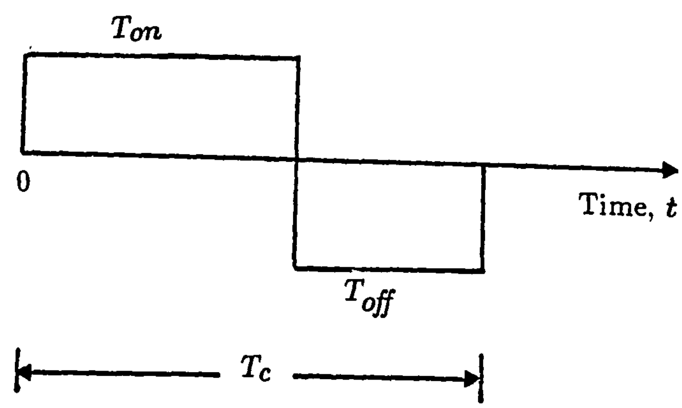

Figure 5.8: A complete cycle of *on* and *off* times

5.3 Systems with General Failure and Repair Distributions

The models proposed require that the underlying distribution for T_{on} and T_{off} be exponential. If this assumption is removed, the models can handle a more general system. For a single component system, the problem can be handled using the approach developed by Kuo [55]. He evaluated the reliability and the availability of a single reparable system via the renewal theory approach under the following assumptions:

- The *on* time, T_{on}, and the *off* time, T_{off}, have a general distribution function $F_{on}(t)$ and $F_{off}(t)$, respectively (see Fig. 5.8).
- A complete cycle time, $T_C(t)$, which is equal to the addition of T_{on} and T_{off} is also a random variable with $F_C(t)$ as its distribution function.

Using the notations already introduced, the probability that the system is available at time t_i and reliable during the performance time c_i, given that the system is available at the end of the $(i-1)$th task performance, can be written as follows:

$$\begin{aligned}\Pr[M_i|M_{i-1}] &= \Pr[Z_1(t_i)=1, T_f^1(t_i)\geq c_i|Z_1(t_{i-1})=1, T_f^1(t_{i-1})\geq \delta] \\ &= \Pr[Z_1(t_i^*)=1]\times \Pr[T_f^1(t_i^*)\geq c_i|Z_1(t_i^*)=1]. \end{aligned} \tag{5.38}$$

By removing the assumption of a Poisson process, Eq. (5.38) can be represented by a renewal equation, $v(t_i^*, c_i)$, where [108]:

$$v(t_i^*, c_i) = \int_0^{t_i^*} v(t_i^* + c_i - u)dF_C(u) + 1 - F_{On}(t_i^* + c_i). \tag{5.39}$$

A renewal solution to Eq. (5.39) is [108]:

$$h(t_i^*, c_i) = 1 - F_{On}(t_i^* + c_i) + \int_0^{t_i^*}[1 - F_{On}(t_i^* + c_i - u)]dm(u), \tag{5.40}$$

where $F_C(t_i^*)$ and $F_{On}(t_i^*)$ are the distribution function of random variable $T_C(t)$ and T_{On}, respectively, at time t_i^*; and $m(u)$ is the expected number of renewals at time u. The expected number of renewals is determined by the cycle time–the sum of T_{On} and T_{off}. Therefore, both F_{On} and $m(u)$ can be estimated either parametrically or nonparametrically [107,108]. By removing the Markovian assumption of the system state, effectiveness for the system defined in Chapter 3 can be written as follows:

$$\begin{aligned} SE_1(T) &= \sum_{k=1}^{\infty}\left\{\int \dot{\dot{D}} \int \frac{\prod_{i=1}^{k} h(t_i^*, c_i)o(t_i, \delta)\lambda(t_i)g(c_i)}{m(T)^k/k!} dc_1 \ldots dc_k dt_1 \right. \\ &\quad \left. \ldots dt_k \times P_o[k; m(T)]\right\} + P_o[0; m(T)] \end{aligned} \tag{5.41}$$

$$\begin{aligned} SE_2(T) &= \sum_{k=1}^{\infty}\left\{\int \dot{\dot{D}} \int \frac{\prod_{i=1}^{k} h(t_i^*, c_i)o(t_i, \delta)\lambda(t_i)g(c_i)}{m(T)^k/k!} dc_1 \ldots dc_k dt_1 \right. \\ &\quad \left. \ldots dt_k \times P_o[k; m(T)]\right\} + P_o[0; m(T)]\int_0^T A(t)dt/T \end{aligned} \tag{5.42}$$

$$SE_3(T) = \sum_{k=1}^{\infty} \left\{ \int \underset{D}{\cdots} \int \frac{\prod_{i=1}^{k} h(t_i^*, c_i) o(t_i, \delta) \lambda(t_i) g(c_i)}{m(T)^k / k!} dc_1 \ldots dc_k dt_1 \ldots dt_k \times P_o[k; m(T)] \right\} / \{1 - P_o[0; m(T)]\}, \quad (5.43)$$

where $SE_1(T)$ is system effectiveness when no task is considered as a mission success; $SE_2(T)$ is system effectiveness when the availability is required even though there is no task; and $SE_3(T)$ is the system effectiveness when missions can occur only if there are task requests.

References [107] and [108] illustrate the procedure to evaluate the estimate $h(t_i^*, c_i)$ and give the close form of the equations when the cycle time and the *on* time are gamma-distributed, with a positive-integer shape parameter, that is,

$$f_T(t) = \begin{cases} \frac{\psi}{(v-1)!}(\psi t)^{v-1} e^{-\psi t}, & t > 0 \\ 0, & t \leq 0 \end{cases} \quad (5.44)$$

$$f_{T_{on}}(t) = \begin{cases} \frac{\chi}{(w-1)!}(\chi t)^{w-1} e^{-\chi t}, & t > 0 \\ 0, & t \leq 0, \end{cases} \quad (5.45)$$

where $\psi > 0$, $\chi > 0$, and v, w are positive integers. Under this assumption [108],

$$h(t_i^*, c_i) = \sum_{l=0}^{w-1} P_o[l; \chi(t_i^* + c_i)] + \psi \int_0^{t_i^*} \left\{ \sum_{l=0}^{w-1} P_o[l; \chi(t_i^* + c_i - u)] \right\} \left\{ \sum_{q=1}^{\infty} P_o[qv - 1; \psi u] \right\} du. \quad (5.46)$$

For the sake of illustration, let T_{off} and T_{on} both be exponentially distributed with parameter γ. Therefore, $Tc(t)$, which is $T_{on} + T_{off}$, has a gamma-distribution function with $v = 2$ and $\psi = \gamma$ [46]. Under this assumption, therefore,

$$\begin{aligned}
h(t_i^*, c_i) &= P_o[0; \gamma(t_i^* + c_i)] + \int_0^{t_i^*} P_o[0; \gamma(t_i^* + c_i - u)] \left\{ \sum_{q=1}^{\infty} P_o[2q-1; \gamma u] \right\} du \\
&= e^{-\gamma(t_i^*+c_i)} + \gamma \int_0^{t_i^*} e^{-\gamma(t_i^*+c_i-u)} \sum_{q=1}^{\infty} \frac{e^{-\gamma u}(\gamma u)^{2q-1}}{(2q-1)!} du \\
&= e^{-\gamma(t_i^*+c_i)} + \gamma \int_0^{t_i^*} e^{-\gamma(t_i^*+c_i)} e^{\gamma u} \left\{(1 - e^{-2\gamma u})/2\right\} du \\
&= e^{-\gamma(t_i^*+c_i)} + \frac{\gamma}{2} e^{-\gamma(t_i^*+c_i)} \int_0^{t_i^*} \left\{e^{\gamma u} - e^{-\gamma u}\right\} du \\
&= e^{-\gamma(t_i^*+c_i)} + \frac{\gamma}{2} e^{-\gamma(t_i^*+c_i)} \left\{(e^{\gamma t_i^*} - 2 + e^{-\gamma t_i^*})/\gamma\right\} \\
&= e^{-\gamma(t_i^*+c_i)} + \frac{1}{2} e^{-\gamma c_i} - e^{-\gamma(t_i^*+c_i)} + \frac{1}{2} e^{-2\gamma t_i^*} e^{-\gamma c_i} \\
&= \left\{\frac{1}{2} + \frac{1}{2} e^{-2\gamma t_i^*}\right\} e^{-\gamma c_i},
\end{aligned}$$

which agrees with $a(t_i^*) r(t_i, c_i)$ for Model A in Section 3.6 if we set $\alpha_1 = \beta_1 = \gamma$. Also note that,

$$\begin{aligned}
a(t_i^*) &= h(t_i^*, 0) \\
&= \frac{1}{2} + \frac{1}{2} e^{-2\gamma t_i^*}
\end{aligned}$$

and

$$\begin{aligned}
r(t_i, c_i) &= \frac{h(t_i^*, c_i)}{h(t_i^*, 0)} \\
&= e^{-\gamma c_i}.
\end{aligned}$$

The general solution to the problem of system effectiveness proposed here is difficult to evaluate numerically even by assuming gamma distributions with a positive integer-shape parameter for the *on* time and the *off* time. If the assumption of gamma distributions were removed, the analytical solution would be extremely difficult to obtain. Kuo [55] proposed a numerical solution for $h(t, x)$. The numerical approach is very general and can be applied to empirical data without assuming the distribution for the data.

The problem becomes more complicated and difficult if the system consists of more than one components. One possible approach is to superimpose the failure and repair processes. The system *on* time and *off* time would be obtained by combining observations of all components involved. In other words, it would be regarded as a complete unit.

6 SUMMARY AND CONCLUDING REMARKS

The objective of this study has been to develop stochastic models for the system involving hardware, software, and human operator and which is required to perform a number of randomly arriving tasks during a fixed mission time. The models were developed in Chapters 3 and 4 for single-machine and N-machine problems, respectively. It combines the following performance measures from human, hardware, and software components: 1) the availability of man-hardware-software system at task request time, 2) the reliability of the man-hardware-software system during the performance time or an allocated time limit for the performance time, 3) the human performance variables, and 4) the system design failure.

A level of operating mode was introduced to account for situation in which the man-hardware-software system may not have failed but is operating at less than peak performance. The model is based on the concept that the level of operating mode will affect the task performance of the operator and the reliability of the system.

The model was extended to handle systems required to carry out several type of tasks during the mission time. Under each type of task, the analytical expression for operator performance was differently defined. A further possible extension would be to give a priority to each type of task. In such case, a task in the process being performed would have a possibility of being replaced when a task with priority arrive.

Another extension was made to handle the systems with general failure and repair distribution. The model was developed for a single-component machine. The problem become more difficult if the system consists of more than one components. One possible approach is to superimpose the failure and repair process. Another possibility is to apply simulation technique. This technique will be able to handle more complex systems based on the basic systematical approach develop in this study.

Throughout this study, no reference was made for time dimension but it was implicitly assume that the clock time was used. Therefore the time could be in second, minute, hour, day, week, etc. depending upon the system being investigated The important thing is that they are must be consistent throughout the analysis.

7 BIBLIOGRAPHY

[1] Angus, J. E. and James, L. E. "Combined Hardware/Software Reliability Models." In *Proceedings Annual Reliability and Maintainability Symposium.* IEEE, New York, 1982, 176-181.

[2] ARINC Res. Corp. *Reliability Engineering.* Prentice-Hall, Englewood, Cliffs, N.J., 1964.

[3] Askren, B. W. and T. L. Regulinski. *Mathematical Modeling of Human Performance Errors for Reliability Analysis of Systems,* Technical Report AMRL-TR-68-93. Aerospace Medical Research Laboratory, Wright-Patterson Air Force Base, OH, 1969.

[4] Askren, B. W. and T. L. Regulinski. *Quantifying Human Performance Reliability,* Technical Report AFHRL-TR-71-22. Air Force Systems Command, Brooks Air Force, TX, 1971.

[5] Balaban, D. L. and D. L. Costello. *System Effectiveness-Concepts and Analytical Technique.* ARINC Research Institute Corporation Publication 267-01-7-419, 1964 Jan.

[6] Butner, S. E. and R. K. Iyer. "A Statistical Study of Reliability and System Load at SLAC." In *Proceedings International Symposium on Fault-Tolerant Computing.* IEEE, New York, 1980, 207-209.

[7] Carnino, A. and M. Griffin. "Causes of Human Error." In A. E. Green (Ed.), *High Risk Safety Technology.* John Wiley & Sons, New York, 1982, 171-179.

[8] Carter, A. D. S. *Mechanical Reliability.* John Wiley & Sons, New York, 1986.

[9] Chung, W. K. "Reliability Analysis of a Repairable Parallel System with Standby Involving Human Error and Common-Cause Failures." *Microelectronics and Reliability,* 27, No. 2 (1987): 269-271.

[10] Cluley, J. C. *Electronic Equipment Reliability.* The Macmillan Press Ltd., London, 1981.

[11] Coleman, J. J. and I. J. Abram. "Mathematical Model for Operational Readiness." *Operation Research,* 10, No. 1 (1962): 126-138.

[12] Cothier, P. H. and A. H. Levis. "Timeliness and Measure of Effectiveness in Command and Control." *IEEE Trans. on Systems, Man, and Cybernetics,* SMC-16, No. 6 (1986): 844-853.

[13] Coutinho, J. S. "Software Reliability Growth." In *Proceedings International Conference on Reliable Software.* Special Interest Group on Programming Languages, New York, 1973, 58-64.

[14] "Definition of Effectiveness Term for Reliability, Maintainability, Human Factor and Safety, MIL-STD-721D." *Microelectronics and Reliability,* 11, No. 5 (1972): 429-433.

[15] Dhillon, B. S. *Human Reliability with Human Factors.* Pergamon Press, New York, 1986.

[16] Dhillon, B. S. and S. N. Rayapati. "Human Performance Reliability Modeling." In *Proceedings Annual Reliability and Maintainability Symposium.* IEEE, New York, 1986, 54-60.

[17] Dhillon, B. S. and S. N. Rayapati. "Human-Error and Common-Cause Failure Modeling of Redundant System." *Microelectronics and Reliability,* 26, No. 6 (1986): 1139-1162.

[18] Dhillon, B. S. and S. N. Rayapati. "Reliability Analysis of Networks with Human Errors: A Block Diagram Approach." *Microelectronics and Reliability,* 27, No. 6 (1987): 981-999.

[19] Dhillon, B. S. and S. N. Rayapati. "Human Performance Reliability Modeling." *Microelectronics and Reliability,* 28, No. 4 (1988): 573-580.

[20] Drury, C. G., B. Paramore, H. P. Van Cott, S. M. Grey, and E. N. Corlett. "Task Analysis." In G. Salvendy (Ed.), *Handbook of Human Factors.* John Wiley & Sons, New York, 1987, 370-401.

[21] Duane, J. T. "Learning Curve Approach to Reliability Monitoring." *Trans. on Aerospace,* 2 (1964): 563-566.

[22] Embrey, D. E. *Human Reliability in Complex System: An Overview.* NCSR Technical Report R10. National Center of Systems Reliability, Warrington, Great Britain, 1976.

[23] Finkelstein, J. and R. Schafer. "Dependability Models for a System of N Parallel Elements." *Proceeding National 8th Symposium on Reliability and Quality Control.* IEEE, New York, 1962, 434-441.

[24] Gephart, L. S. and V. Balachandran. *A Stochastic System Effectiveness Simulation Model,* Technical Report UDRI-TR-70-12, University of Dayton-Research Institute, 1969 June.

[25] Goel, A. L. and K. Okumoto. "An Analysis of Recurrent Software Errors in Real-Time Control System." In *Proceedings ACM Conference.* ACM, New York, 1978, 496-501.

[26] Goel, A. L. and K. Okumoto. "Time-Dependent Error Detection Rate Model for Software Reliability and Performance Measure." *IEEE Trans. on Reliability,* R-28, No. 3 (1979):206-211.

[27] Goel, A. L. and K. Okumoto. "A Markovian Model for Reliability and Other Performance Measure of Software System." In *Proceedings COMPCON.* IEEE, New York, 1979, 769-774.

[28] Goel, A. L. and J. B. Soenjoto. "Models for Hardware-Software System Operational-Performance Evaluation." *IEEE Trans. on Reliability,* R-30, No. 3 (1981): 232-239.

[29] Goldman, A. S. and T. B. Slattery. *Maintainability.* General Electric Co., Tempo, 1964.

[30] Gonzales-Vega, O., J. W. Foster III and G. L. Hogg. "A Simulation Program to Model Effects of Logistics on R&M of Complex System." In *Proceedings Annual Reliability and Maintainability Symposium.* IEEE, New York, 1988, 306-317.

[31] Gubitz, M. and K. O. Ott. "Quantifying Software Reliability by a Probabilistic Model." *Reliability Engineering,* 5 (1983): 157-171.

[32] Gupta, P. P. and R. K. Gupta. "Cost Analysis of an Electronic Repairable Redundant System with Critical Human Errors." *Microelectronics and Reliability,* 26, No. 3 (1986): 417-422.

[33] Gupta, P. P. and A. Kumar. "MTTF and Availability Evaluation of a Two-Unit, Two-State, Parallel Redundant Complex System with Constant Human Failures." *Microelectronics and Reliability,* 26, No. 1 (1986): 53-56.

[34] Gupta, P. P. and A. Kumar. "Point-wise Availability of Two-Unit Standby Redundant Electronic Equipment Under Human Failure." *Microelectronics and Reliability,* 26, No. 2 (1986): 225-228.

[35] Gupta, P. P. and A. Kumar. "Reliability and MTTF Analysis of a Non-Repairable Parallel Redundant Complex System Under Hardware and Human Failures." *Microelectronics and Reliability,* 26, No. 2 (1986): 228-234.

[36] Gupta, P. P. and A. Kumar. "Stochastic Behavior of Two-Unit Cold Standby Redundant Electronic Equipment Under Human Failures." *Microelectronics and Reliability,* 27, No. 1 (1987): 15-18.

[37] Gupta, P. P. and R. K. Sharma. "Availability and MTTF Analysis of a Three-State Repairable Redundant Electronic Equipment Under Critical Human Errors." *Microelectronics and Reliability,* 26, No. 1 (1986): 57-62.

[38] Gupta, P. P. and R. K. Sharma. "Availability and MTTF Analysis of a Three-State Parallel Redundant Multi-Component System Under Critical Human Failures." *Microelectronics and Reliability,* 26, No. 1 (1986): 63-68.

[39] Gupta, P. P. and R. K. Sharma. "Cost Analysis of Three-State Repairable Redundant Complex System Under Various Modes of Failures." *Microelectronics and Reliability,* 26, No. 1 (1986): 69-74.

[40] Gupta, P. P. and R. K. Sharma. "Reliability Analysis of Two State Repairable Parallel Redundant System Under Human Failure." *Microelectronics and Reliability,* 26, No. 2 (1986): 221-224.

[41] Haynes, R. D. and W. E. Thompson. "Hardware and Software Reliability and Confidence Limits for Computer-Controlled Systems." *Microelectronics and Reliability,* 20, No. 1 (1980):109-122.

[42] Haynes, R. D. and W. E. Thompson. "Combined Hardware and Software Availability." In *Proceedings Annual Reliability and Maintainability Symposium.* IEEE, New York, 1981, 365-370.

[43] Hayward, P. "The Measurement of Combat Effectiveness." *Operation Research,* 16, No. 2 (1968):221-224.

[44] Hecht, H. and M. Hecht. "Software Reliability in the System Context." *IEEE Trans. on Software Engineering,* SE-12, No. 1 (1986): 51-58.

[45] Henry, G. C. "Improved Effectiveness Through Availability Models." *Annals of Reliability and Maintainability,* 3 (1964): 198-206.

[46] Hogg, R. V. and A. T. Craig. *Introduction to Mathematical Statistics.* Macmillan Publishing Co., Inc., New York, 1978.

[47] Hosford, J. E. "Measures of Dependability." *Operation Research,* 8, No. 1 (1960): 53-64.

[48] Iskander, W. H. and R. S. Nutter, Jr. "Methodology Development for Safety and Reliability Analysis for Electrical Mine Monitoring Systems." *Microelectronics and Reliability,* 28, No. 4 (1988): 581-598.

[49] Jelinski, Z. and P. B. Moranda. "Software Reliability Research." In W. Freiberger (ed.), *Statistical Computer Performance Evaluation.* Academic Press, New York, 1972, 465-484.

[50] Kapur, K. C. and L. R. Lamberson. *Reliability in Engineering Design.* John Wiley & Sons, New York, 1977.

[51] Karmiol, E. D., W. T. Weir and J. S. Youtcheff. "The Reliability/ Maintainability Relationship in Aerospace Program." *Annals of Reliability and Maintainability,* 4, (1965):261-272.

[52] Katz, M. D., H. Jaffe and S. A. Rosenthal. "Designing Reliability into the B-58 Bombing-Navigation System." In *Proceedings 7th National Symposium on Reliability and Quality Control.* IEEE, New York, 1961, 273-282.

[53] Kline, M. B. "Software & Hardware R&M: What are the Differences?" In *Proceedings Annual Reliability and Maintainability Symposium.* IEEE, New York, 1980, 179-185.

[54] Kremer, W. "Birth-Death and Bug Counting." *IEEE Trans. on Reliability,* R-32, No. 1 (1983): 37-47.

[55] Kuo, W. *System Effectiveness Models Via Renewal Theory and Bayesian Inference.* Ph.D. Dissertation. Kansas State University, Manhattan, 1980.

[56] Kuo, W. "Software Reliability Estimation: A Realization of Competing Risk." *Microelectronics and Reliability,* 23, No. 2 (1983): 249-260.

[57] Lee, K. W. *Stochastic Models for System Effectiveness.* Ph.D. Dissertation. Kansas State University, Manhattan, 1985.

[58] Lee, K. W., F. A. Tillman, J. J. Higgins. "A Literature Survey of the Human Reliability Component in a Man-Machine System." *IEEE Trans. on Reliability,* R-37, No. 1 (1988): 24-34.

[59] Leuba, H. R. *The Human Factor Ingredient in System Effectiveness.* ARINC Research Corporation, Annapolis, Maryland, 1968 Feb.

[60] Lie, C. H. *System Effectiveness Models for Maintained System: Analytical and Simulation Approach.* Ph.D. Dissertation. Kansas State University, Manhattan, 1977.

[61] Lie, C. H., C. L. Hwang and F. A. Tillman. "Availability of Maintained Systems: A State of Art Survey." *AIIE Trans.*, 9, No. 3 (1977): 247-259.

[62] Lie, C. H., W. Kuo, F. A. Tillman and C. L. Hwang. "Mission Effectiveness Model for a System with Several Mission Type." *IEEE Trans. on Reliability,* R-33, No. 4 (1984):346-352.

[63] Lin, H. *Integration of Software Reliability into System Reliability.* Ph.D. Dissertation. Iowa State University, Ames, Iowa, 1987.

[64] Lipow, M. "Prediction of Software Failures." *The Journal of System and Software,* 1 (1979): 71-75.

[65] Littlewood, B. "Software Reliability Models fo Modular Program Structure." *IEEE Trans. on Reliability,* R-28, No. 3 (1979): 241-245.

[66] Miester, D. "Method of Predicting Human Reliability in Man-Machine Systems." *Human Factors,* 6, No. 6 (1964): 621-646.

[67] Miester, D. "Alternate Approaches to Human Reliability in Man-Machine System." In R. A. Waller and V. T. Corello (Eds.), *Low Probability/High Consequence Risk Analysis.* Plenum Press, New York, 1984, 319-333.

[68] Miller, D. R. and A. Sofer. "Completely Monotone Regression Estimation of Software Failure Rate." In *Proceedings International Conference on Software Engineering.* IEEE Computer Society, Washington, DC, 1985, 343-348.

[69] Min Xie. "A Shock Model for Software Failures." *Microelectronics and Reliability,* 27, No. 4 (1987): 717-724.

[70] Mohan, C., R. C. Garg and P. P. Singal. "Dependability of a Complex System." *Operation Research,* 10, No. 3 (1962) 310-313.

[71] Moranda, P. B. "An Error Detection Model for Application during Software Development." *IEEE Trans. on Reliability,* R-30, No. 4 (1981): 309-312.

[72] Moranda, P. B. "Event-Altered Rate Models for General Reliability Analysis." *IEEE Trans. on Reliability,* R-28, No. 5 (1979): 376-381.

[73] Musa, J. D. "A Theory of Software Reliability and Its Application." *IEEE Trans. on Software Engineering,* SE-1, No. 3 (1975):312-327.

[74] Nathan, I. "A Deterministic Model to Predict Error-Free Status of Complex Software Development." *Workshop on Quantitative Software Models,* 1979, 159-169.

[75] *Naval System Effectiveness, NAVMAT P3941-B Proposed, Technical Document 251 (NELC/TD251).* Prepared by the Harbinger Corporation, Santa Monica, California. Naval Electronic Laboratory Center, Product Assurance Division (Code 4100), San Francisco, California, October 1973.

[76] Naylor, T. H., L. J. Balintfy, D. S. Burdick and K. Chu. *Computer Simulation Techniques.* John Wiley & Sons, Inc., New York, 1966.

[77] Nelson, E. "Estimating Software Reliability From Test Data." *Microelectronics and Reliability,* 17, No. 1 (1978): 67-74.

[78] Norman, D. A. "Categorization of Actions Slips." *Psychol. Rev.,* 88, No. 1 (1981): 1-15.

[79] Ohba, M. "Software Reliability Analysis Models." *IBM Journal of Research and Development,* 28, No. 4 (1984): 428-443.

[80] Parzen, E. *Stochastic Processes.* Holden Day Series in Probability and Mathematics. Holden Day Inc., San Francisco, 1962.

[81] *Product Assurance - Army Material Reliability, Availability, and Maintainability (RAM),* TR 702-3 Revision, Reports Control System CSCRD-73, Effective, Headquarters, Department of the Army, Washington, D. C. , 1973 Mar.

[82] Ramamoorthy, C. V. and S. F. Ho. "Software Reliability - Status and Perspectives." *IEEE Trans. on Software Engineering,* SE-8, No. 4 (1982): 354-371.

[83] Rasmussen, J. "Note on Human error Analysis and Prediction." In G. Apostolakis and G. Volta (Eds.), *Synthesis and Analysis Methods for Safety and Reliability Studies.* Plenum Press, London, 1979, 357-389.

[84] Rasmussen, J. "Human Reliability in Risk Analysis." In A. E. Green (Ed.), *High Risk Safety Technology.* John Wiley & Sons, London, 1982, 143-170.

[85] Rasmussen, J. "Approaches to the Control of the Effects of Human error on Chemical Plant Safety." In *Proceedings International Symposium on Preventing Major Chemical Accidents.* AICE, New York, 1987, 221-229.

[86] Reason, J. "Slips and Mistakes: Two Distinct Classes of Human Error?" In D. J. Oborne (Ed.), *Proceedings Ergonomics Society's 1985 Conference.* Taylor & Francis, London, 1985, 103-110.

[87] Romeu, J. L. and K. A. Dey. "Classifying Combined Hardware/Software R Models." In *Proceedings Annual Reliability and Maintainability Symposium.* IEEE, New York, 1984, 282-287.

[88] Rouse, W. B. and S. H. Rouse. "Analysis and Classification of Human error." *IEEE Trans. on System, Man and Cybernetics Innovation*, SMC-13, (1983): 539-549.

[89] Schick, G. J. and R. W. Wolverton. "An Analysis of Competing Software Reliability Models." *Trans. on Software Engineering,* SE-4, No. 2 (1978): 104-120.

[90] Shantikumar, J. G. "A General Software Reliability Model for Performance Prediction." *Microelectronics and Reliability,* 21, No. 5 (1981): 671-682.

[91] Sharit, J. "A Critical Review of Approaches to Human Reliability Analysis." *Int. Jour. of Industrial Ergonomics*, 2, No. 2 (1988): 111-130.

[92] Shooman, M. L. "Software Reliability: Measurement and Models." In *Proceedings Annual Reliability and Maintainability Symposium.* IEEE, New York, 1975, 485-491.

[93] Shooman, M. L. *Software Engineering-Design, Reliability and Prediction.* McGraw Hill, New York, 1983.

[94] Siegel, A. I. and M. R. Lautman. "A Model for Predicting Integrated Man-Machine System Reliability: Model Logic and Descriptions." *Technical Report AD-A009 814.* Applied Psychological Services, Wayne, PA, 1974.

[95] Siegel, A. I. and M. R. Lautman. "A Family Models for Measuring Human Reliability." In *Proceedings Annual Reliability and Maintainability Symposium.* IEEE, New York, 1975, 110-115.

[96] Siegel, A. I. and J. J. Wolf. *Man-Machine Simulation Models.* John Wiley & Sons, New York, 1969.

[97] Stark, G. E. "Dependability Evaluation of Integrated Hardware/Software System." *Trans. on Reliability,* R-36, No. 4 (1987): 440-444.

[98] Sukert, A. N. "An Investigation of Software Reliability Models." In *Proceedings Annual Reliability and Maintainability Symposium.* IEEE, New York, 1977, 478-484.

[99] Sumita, U. and Masuda, Y. "Analysis of Software Availability/Reliability Under the Influence of Hardware Failures." *IEEE Trans. on Computer Engineering,* SE-12, No. 1 (1986): 32-41.

[100] Sumita, U. and J. G. Shantikumar. "A Software Reliability Growth Model with Multiple Error Introduction and Removal." *IEEE Trans. on Reliability,* R-35, No. 4 (1986): 459-462.

[101] Swain, A. D. "Some Problems in the Measurement of Human Performance in Man-Machine System." *Human Factors,* 6, No. 6 (1964): 687-700.

[102] Swain, A. D. "Qualifications of Human Error in LP/HC Risk Analysis." In R. A. Waller and V. T. Covello (Eds.), *Low-Probability High-Consequence Risk Analysis.* Plenum Press, New York, 1984, 293-295.

[103] Swain, A. D. and H. E. Guttmann. *Handbook of Human Reliability Analysis with Emphasis on Nuclear Power Plant Application,* NUREG/CR-1278, US-NRC, 1983.

[104] Taylor, H. M. and S. Karlin. *An Introduction to Stochastic Modeling.* Academic Press Inc., Orlando, 1984.

[105] Thompson, W. E. and P. O. Chelson. "On the Specification and Testing of Software Reliability." In *Proceedings Annual Reliability and Maintainability Symposium.* IEEE, New York, 1980, 379-383.

[106] Tillman, F. A., C. L. Hwang and W. Kuo. "System Effectiveness Models: An Annotated Bibliography." *IEEE Trans. on Reliability,* R-29, No. 4 (1980): 295-304.

[107] Tillman, F. A., W. Kuo, R. F. Nassar and C. L. Hwang. "A Numerical Simulation of the System Effectiveness - A Renewal Theory Approach." In *Proceedings Annual Reliability and Maintainability Symposium.* IEEE, New York, 1982, 252-261.

[108] Tillman, F . A., W. Kuo, R. F. Nassar and C. L. Hwang. "Numerical Evaluation of Instantaneous Availability." *IEEE Trans. on Reliability,* R-32, No. 1 (1983): 119-123.

[109] Tillman, F. A., C. H. Lie and C. L. Hwang. "Analysis of Pseudo-Reliability of a Combat Tank System and Its Optimal Design." *IEEE Trans. on Reliability,* R-25, No. 4 (1976): 239-242.

[110] Tillman, F. A., C. H. Lie and C. L. Hwang. " Simulation Model of Mission Effectiveness for Military System." *IEEE Trans. on Reliability,* R-27, No. 3 (1978): 191-194.

[111] Tillman, F. A., R. F. Nassar, W. Kuo and C. L. Hwang. "Simulation of System Availability – A Renewal Approach." In *Proceedings of the Second International Conference on Reliability and Maintainability.* IEEE, New York, 1980, 322-327.

[112] Topmiller, D. A. "Human Factors and System Effectiveness." In *Proceedings Reliability and Maintainability Conference.* IEEE, New York, 1966, 123-132.

[113] Wagoner, W. L. *The Final Report on Software Reliability Measurement Study,* Report TOR-0074(41221)-1. The Aerospace Corp., El Segundo, CA, 1973.

[114] Wall, J. K. and P. A. Ferguson. "Pragmatic Software Reliability Prediction." In *Proceedings Annual Reliability and Maintainability Symposium.* IEEE, New York, 1977, 485-488.

[115] *Weapon System Effectiveness Industry Advisory Committee (WSEIAC),* Final Report of Task Group I – Requirements Methodology, AFSC-TR-65-1, System Effectiveness Division, System Policy Directorate, Headquarters, U.S. Air Force Command, Andrew Air Force Base, Maryland, 1965 Jan.

[116] *Weapon System Effectiveness Industry Advisory Committee (WSEIAC),* Final Report of Task Group II – Prediction-Measurement, AFSC-TR-65-2, System Effectiveness Division, System Policy Directorate, Headquarters, U.S. Air Force Command, Andrew Air Force Base, Maryland, 1965 Jan.

[117] White, J. A., J. W. Schmidt and G. K. Bennet. *Analysis of Queueing System.* Academic Press Inc., New York, 1975.

[118] William, W. H. M. "Reliability of Large Real-Time Control Software Systems." *Record 1973 IEEE Symposium on Computer Software Reliability,* 1-6.

[119] Winokur, H. S. and L. J. Goldstein. "Analysis of Mission-Oriented Systems." *IEEE Trans. on Reliability,* R-18, No. 4 (1969): 144-148.

[120] Yamada, S. "S-Shape Reliability Growth Modeling for Software Error Detection." *IEEE Trans. on Reliability*, R-32, No. 5 (1983):475-478.

[121] Yamada, S. and S. Osaki. "Cost-Reliability Optimal Release Policies for Software Systems." *IEEE Trans. on Software Engineering*, SE-11, No. 12 (1985): 1431-1437.

[122] Yamada, S. and S. Osaki. "Reliability Growth Model for Hardware and Software Systems Based on Nonhomogeneous Poisson Process: A Survey." *Microelectronics and Reliability*, 23, No. 1 (1983): 91-112.

8 ACKNOWLEDGEMENTS

The author wants to especially thank his wife, Imas, and their three children, Alfa, Betha, and Gama. Without their support and patience, this research and his overall study would never been completed. A special thanks is also given to his parent Saleh Abbas and Zubaidah Ahmad for their love, encouragement and countless sacrifices they made for their children.

The author is particularly grateful to his major professor, Dr. Way Kuo, whose continuous support and encouragement have made this work possible. The author also wants to thank Professors H. T. David, J. C. Even Jr., W. Q. Meeker Jr., and H. D. Meeks for their interest and participation as advisory committee members. Their stimulating discussion in committee meetings was very rewarding.

Finally the author would like to express his gratitude and appreciation to the Agency of Agricultural Research and Development of the Indonesian Government for sponsoring his study at Iowa State University.

9 APPENDIX A: PROGRAM I

```
C       Program to simulate Example 1
C
C       Variables:
C
C           alfa1       1/alfa1 = average machine running time
C           beta1       1/beta1 = average machine down time
C           busy1       1 (machine busy), 0 (machine down)
C           detpar      parameter for detection probabilities
C           seed        seed for random number generator
C           event1      type of failure (if any-see SUBROUTINE check)
C           ftype       failure-type frequencies
C           freq(i,1)   number of missions with i tasks
C           freq(i,2)   number of successful missions with i tasks
C           iunit       output unit number
C           lambda      arrival rate
C           mission     current mission number
C           mstime      maximum service time
C           mu          service rate
C           nmiss       total number of missions
C           nulmiss     number of missions without tasks
C           perpar      parameter for performance probabilities
C           sta1        1 (machine up), 0 (machine down)
C           sucmiss     number of successful missions
C           tarriv      arrival time
C           tavai1      total time that machine is available
C           tbusy1      time when machine change from busy to idle (vv)
C           tsta1       time when machine change from ON to OFF (vv)
C           ttotal      mission duration
C
```

```
C      Variable declaration:
C
       REAL              alfa1,beta1,detpar,lambda,mstime,mu,perpar
       REAL              tarriv,tavai1,tbusy1,tsta1,ttotal
       INTEGER           ftype(10),freq(51,2)
       INTEGER           busy1,event1,mission,narriv,nmiss
       INTEGER           nulmiss,sta1,sucmiss,iunit
       DOUBLE PRECISION seed
       CHARACTER*20      fcause(10)
C
 10    FORMAT(5X,A30,2X,I6,2X,F9.6,I6,A20)
C
C      Variable initialization:
C
       CALL init(ftype,ttotal,alfa1,beta1,mstime,mu,
     +      lambda,detpar,perpar,nmiss,seed,iunit)
       CALL display(ttotal,alfa1,beta1,mstime,mu,
     +      lambda,detpar,perpar,nmiss,seed,iunit)
C
       fcause(1) = ' Machine down       '
       fcause(2) = ' Machine Busy       '
       fcause(3) = ' Task not detected  '
       fcause(4) = ' Task not perf. well'
       fcause(5) = ' Task too long      '
       fcause(6) = ' SUCCESSFUL MISSION '
C
C      Begin simulation
C
       sucmiss = 0
       nulmiss = 0
       tavai1  = 0.0
       DO 1000 mission = 1,nmiss
          tsta1  = 0.0
          sta1   = 0
          tbusy1 = 0.0
          busy1  = 0
          event1 = 6
          narriv = 0
          tarriv = -ALOG(RND(seed))/lambda
```

```
C
C        Determine whether this mission contains any tasks or not
C
         IF(tarriv .GT. ttotal) THEN
            nulmiss = nulmiss+1
  100       IF(tstai .LT. ttotal) THEN
               CALL mupdate(alfa1,beta1,sta1,tsta1,
     +              tavail,ttotal,seed)
               GOTO 100
            ENDIF
         ENDIF
C
  300    IF(tarriv .LE. ttotal) THEN
C
C           Determine machine status (ON/OFF) at the arrival time
C
  600       IF(tarriv .GT. tsta1) THEN
               CALL mupdate(alfa1,beta1,sta1,tsta1,
     +              tavail,ttotal,seed)
               tbusy1 = tsta1
               busy1  = 0
               GOTO 600
            ENDIF
C
C           Determine if task will be completed (event1=6)
C
            CALL check(event1,sta1,tsta1,busy1,tbusy1,
     +           detpar,perpar,mu,mstime,tarriv,seed)
C
C           If task was not completed then determine the number of
C           tasks & consider the next mission
C
            IF(event1 .NE. 6) THEN
  800          IF(tarriv .LE. ttotal) THEN
                  tarriv = tarriv-ALOG(RND(seed))/lambda
                  narriv = narriv+1
                  CALL mupdate(alfa1,beta1,sta1,
     +                 tsta1,tavai,ttotal,seed)
                  GOTO 800
```

```
            ENDIF
            GOTO 900
         ENDIF
C
C        Task completed, consider next task
C
         tarriv = tarriv-ALOG(RND(seed))/lambda
         narriv = narriv+1
         IF(tarriv .GT. tbusy1) THEN
            busy1  = 0
            tbusy1 = tarriv
         ENDIF
C
         GOTO 300
      ENDIF
C
C     Mission completed, collect statistics
C
 900  IF(narriv .GT. 50) narriv = 51
      ftype(event1) = ftype(event1)+1
      IF(narriv .GT. 0) THEN
         freq(narriv,1) = freq(narriv,1)+1
         IF(event1 .EQ. 6) freq(narriv,2) = freq(narriv,2)+1
      ENDIF
C
C     If a simulation run completed, print a report
C
      IF(50*(mission/50) .EQ. mission) THEN
         CALL state(nulmiss,tavail,ttotal,freq,51,iunit)
      ENDIF
 1000 CONTINUE
C
C     Print final report and end simulation
C
      CALL report(nulmiss,tavail,ttotal,
     +     freq,51,ftype,fcause,6,iunit)
      CLOSE(iunit)
      STOP ' '
      END
```

```
      SUBROUTINE init(ftype,ttotal,alfa1,beta1,mstime,mu,
     +               lambda,detpar,perpar,nmiss,seed,iunit)
C
C     Subroutine to read in parameters for simulation
C
      REAL             ttotal,alfa1,beta1,mstime,mu,lambda,detpar
      REAL             perpar
      INTEGER          ftype(15),nmiss,iunit
      DOUBLE PRECISION seed
      CHARACTER*45     msg(15),flnm
C
      msg(1)  = '   Man-Machine   System   Simulation     '
      msg(2)  = 'Enter seed for random number generator = '
      msg(3)  = 'Enter total number of simulation runs  = '
      msg(4)  = 'Enter mission duration,              T = '
      msg(5)  = 'Enter machine breakdown rate,    Alpha = '
      msg(6)  = 'Enter machine repair rate,        Beta = '
      msg(7)  = 'Enter service rate,                 Mu = '
      msg(8)  = 'Enter maximum allocated service time,U = '
      msg(9)  = 'Enter arrival rate,             Lambda = '
      msg(10) = 'Enter detection-prob. parameter,     D = '
      msg(11) = 'Enter performance-prob. parameter,   P = '
      msg(12) = 'A positive input is required ! RE-ENTER  '
      msg(13) = 'Enter output filename                  = '
C
 10   FORMAT(10X,A45,\)
C
      WRITE(*,10) msg(1)
      WRITE(*,10)
      WRITE(*,10)
 15   WRITE(*,10) msg(2)
      READ(*,*) seed
      IF(seed .LT. 0) THEN
         WRITE(*,10) msg(12)
         WRITE(*,*)
         GOTO 15
      ENDIF
 20   WRITE(*,10) msg(3)
      READ(*,*) nmiss
```

```
      IF(nmiss .LT. 0) THEN
         WRITE(*,10) msg(12)
         WRITE(*,*)
         GOTO 20
      ENDIF
      nmiss = nmiss*50
25    WRITE(*,10) msg(4)
      READ(*,*) ttotal
      IF(ttotal .LT. 0) THEN
         WRITE(*,10) msg(12)
         WRITE(*,*)
         GOTO 25
      ENDIF
30    WRITE(*,10) msg(5)
      READ(*,*) alfa1
      IF(alfa1 .LT. 0) THEN
         WRITE(*,10) msg(12)
         WRITE(*,*)
         GOTO 30
      ENDIF
35    WRITE(*,10) msg(6)
      READ(*,*) beta1
      IF(beta1 .LT. 0) THEN
         WRITE(*,10) msg(12)
         WRITE(*,*)
         GOTO 35
      ENDIF
40    WRITE(*,10) msg(7)
      READ(*,*) mu
      IF(mu .LT. 0) THEN
         WRITE(*,10) msg(12)
         WRITE(*,*)
         GOTO 40
      ENDIF
45    WRITE(*,10) msg(8)
      READ(*,*) mstime
      IF(mstime .LT. 0) THEN
         WRITE(*,10) msg(12)
         WRITE(*,*)
```

```
         GOTO 45
      ENDIF
50    WRITE(*,10) msg(9)
      READ(*,*) lambda
      IF(lambda .LT. 0) THEN
         WRITE(*,10) msg(12)
         WRITE(*,*)
         GOTO 50
      ENDIF
55    WRITE(*,10) msg(10)
      READ(*,*) detpar
      IF(detpar .LT. 0) THEN
         WRITE(*,10) msg(12)
         WRITE(*,*)
         GOTO 55
      ENDIF
60    WRITE(*,10) msg(11)
      READ(*,*) perpar
      IF(seed .LT. 0) THEN
         WRITE(*,10) msg(12)
         WRITE(*,*)
         GOTO 60
      ENDIF
      WRITE(*,10) msg(13)
      READ(*,'(A15)') flnm
C
      DO 100 i = 110
         ftype(i) = 0
100   CONTINUE
C
      iunit = 6
      OPEN(iunit,FILE = FLNM,STATUS = 'NEW')
      RETURN
      END
```

```
      SUBROUTINE report(nulmiss,tavai,ttotal,freq,nfreq,
     +            ftype,fcause,ntype,iu)
C
C     This subroutine computes the measures of effectiveness and
C     prints out the final report
C
      INTEGER          nulmiss,nfreq,i,totmiss,sucmiss,k,ntype
      INTEGER          failmiss,freq(nfreq,2),ftype(ntype),iu
      REAL             tavai,ttotal,avail,eff1,eff2,eff3,f1,f2
      CHARACTER*20     fcause(ntype)
      CHARACTER*70     tt1,tt2,tt3,tt4,tt5,tt6
C
 10   FORMAT(5X,A41,F14.6)
 20   FORMAT(5X,A41,I6)
 30   FORMAT(5X,A70)
 35   FORMAT(5X,A41)
 40   FORMAT(5X,3(3X,I8),F9.5,2X,F9.5)
 50   FORMAT(8X,I10,2(3X,F10.6))
 60   FORMAT(5X,A20,2X,I10,4X,F14.7)
C
      tt1='       Breakdown by Number of Jobs per Mission        '
      tt2='            Absolute Frequencies  Relative Frequencies'
      tt3='Jobs/mission   Total   Successful    Total  Successful'
      tt4='===================================================='
      tt5='           Breakdown by Cause of the Failure          '
      tt6=' Failure Cause    Abs. Frequency    Rel. Frequency    '
      k = 0
      totmiss = nulmiss
      sucmiss = 0
      DO 100 i = 1,nfreq
         totmiss = totmiss+freq(i,1)
         sucmiss = sucmiss+freq(i,2)
         IF(freq(i,1) .NE. 0) k = i
 100  CONTINUE
      failmiss = totmiss-sucmiss-nulmiss
      avail    = tavai/(ttotal*totmiss)
      eff1     = FLOAT(sucmiss+nulmiss)/FLOAT(totmiss)
      eff2     = FLOAT(sucmiss)/FLOAT(totmiss-nulmiss)
      eff3     = (sucmiss+nulmiss*avail)/FLOAT(totmiss)
```

```
C
      WRITE(iu,*)
      WRITE(iu,35)'             Man-Machine System Simulation'
      WRITE(iu,*)
      WRITE(iu,35)'                     Final Report           '
      WRITE(iu,*)
      WRITE(iu,20)'Total number of missions :............',totmiss
      WRITE(iu,20)'Total number of successful missions:..',sucmiss
      WRITE(iu,20)'Total number of failed missions:......',failmiss
      WRITE(iu,20)'Number of missions without any task:..',nulmiss
      WRITE(iu,*)
      WRITE(iu,10)'     Availability...................... ',avail
      WRITE(iu,10)'     Effec.1........................... ',eff1
      WRITE(iu,10)'     Effec.2........................... ',eff2
      WRITE(iu,10)'     Effec.3........................... ',eff3
      WRITE(iu,*)
      WRITE(iu,30) tt1
      WRITE(iu,30) tt2
      WRITE(iu,30) tt4
      WRITE(iu,30) tt3
      WRITE(iu,*)
      f1 = FLOAT(nulmiss)/FLOAT(totmiss)
      WRITE(iu,40) 0,nulmiss,nulmiss,f1,f1
      DO 200 i = 1,k
         f1 = FLOAT(freq(i,1))/FLOAT(totmiss)
         f2 = FLOAT(freq(i,2))/FLOAT(totmiss)
         WRITE(iu,40) i,freq(i,1),freq(i,2),f1,f2
 200  CONTINUE
      WRITE(iu,30) tt4
C
      WRITE(iu,*)
      WRITE(iu,30) tt5
      WRITE(iu,30) tt6
      WRITE(iu,30) tt4
      WRITE(iu,*)
      DO 400 i = 1,ntype
         f1 = FLOAT(ftype(i))/FLOAT(totmiss)
         WRITE(iu,60) fcause(i),ftype(i),f1
 400  CONTINUE
```

```
      WRITE(iu,30) tt4
      WRITE(iu,*)
C
      RETURN
      END
C
C
C
      SUBROUTINE state(nulmiss,tavai,ttotal,freq,nfreq,iu)
C
C     This subroutine computes the measures of effectiveness and
C     prints them out
C
      INTEGER          nulmiss,nfreq,i,totmiss,sucmiss,k
      INTEGER          failmiss,freq(nfreq,2),iu
      REAL             tavai,ttotal,avail,eff1,eff2,eff3,f1,f2
      CHARACTER*70     tt1,tt2,tt3
C
 10   FORMAT(5X,I7,4X,4(F9.6,3X))
 30   FORMAT(5X,A70)
C
      k = 0
      totmiss = nulmiss
      sucmiss = 0
      DO 100 i = 1,nfreq
         totmiss = totmiss+freq(i,1)
         sucmiss = sucmiss+freq(i,2)
         IF(freq(i,1) .NE. 0) k = i
 100  CONTINUE
      nreport = totmiss/50
      failmiss = totmiss-sucmiss-nulmiss
      avail    = tavai/(ttotal*totmiss)
      eff1     = FLOAT(sucmiss+nulmiss)/FLOAT(totmiss)
      eff2     = FLOAT(sucmiss)/FLOAT(totmiss-nulmiss)
      eff3     = (sucmiss+nulmiss*avail)/FLOAT(totmiss)
C
      IF(nreport .EQ. 1) THEN
         tt1='                   Man-Machine System Simulation          '
         tt2='Mission No.   Avail.     SE1     SE2     SE3'
```

```
         tt3='====================================================='
         WRITE(iu,*)
         WRITE(iu,30) tt1
         WRITE(iu,*)
         WRITE(iu,30) tt2
         WRITE(iu,30) tt3
         WRITE(iu,*)
      ENDIF
C
      WRITE(iu,10) totmiss,avail,eff1,eff2,eff3
      RETURN
      END
C
C
C
      SUBROUTINE display(ttotal,alfa1,beta1,mstime,mu,
     +           lambda,detpar,perpar,nmiss,seed,iu)
C
C     Subroutine to display parameters for simulation
C
      REAL             ttotal,alfa1,beta1,mstime,mu,lambda,detpar
      REAL             perpar,tarriv
      INTEGER          nmiss,iu
      DOUBLE PRECISION seed
C
 10   FORMAT(5X,A37,F14.6)
 20   FORMAT(5X,A37,I6)
 30   FORMAT(5X,A37)
      WRITE(iu,*)
      WRITE(iu,30) ' Man-Machine System Simulation        '
      WRITE(iu,*)
      WRITE(iu,30) '      Simulation Parameters           '
      WRITE(iu,*)
      WRITE(iu,10) ' Random number generator seed      = ',seed
      WRITE(iu,10) ' Total time per mission        (T) = ',ttotal
      WRITE(iu,20) ' Total number of missions          = ',nmiss
      WRITE(iu,10) ' Machine repair rate        (Beta) = ',beta1
      WRITE(iu,10) ' Machine breakdown rate    (Alpha) = ',alfa1
      WRITE(iu,10) ' Service rate,                (Mu) = ',mu
```

```
      WRITE(iu,10) ' Max. allocated service time   (U) = ',mstime
      WRITE(iu,10) ' Arrival rate             (Lambda) = ',lambda
      WRITE(iu,10) ' Detection-prob. parameter     (D) = ',detpar
      WRITE(iu,10) ' Performance-prob. parameter   (P) = ',perpar
      WRITE(iu,*)
C
      RETURN
      END
C
C
C
      SUBROUTINE mupdate(alfa,beta,sta,tsta,tavai,ttotal,seed)
C
C     This procedure updates the machine status
C     Parameters:
C        1/alfa    mean up time
C        1/beta    mean down time
C        seed      seed for random number generator
C        sta       0 (machine down), 1 (machine up)
C        tavai     total time that machine is available
C        tsta      time at which next status change will occur
C        ttotal    mission duration
C
      REAL             alfa,beta,tsta,ttotal,rate,t,tavai
      INTEGER          sta
      DOUBLE PRECISION seed
C
      sta   = 1-sta
      rate  = (1-sta)*beta+sta*alfa
      t     = AMIN1(-ALOG(RND(seed))/rate,ttotal-tsta)
      tsta  = tsta+t
      tavai = tavai+t*sta
      RETURN
      END
C
      SUBROUTINE check(event,sta,tsta,busy,tbusy,detpar,
     +           perpar,mu,mstime,tarriv,seed)
C
C     This subroutine checks whether a task can be completed.
```

```
C
C     Parameters:
C
C        busy      1 (machine busy), 0 (machine down)
C        detpar    parameter for detection probabilities
C        seed      seed for random number generator
C        event     indicates failure type:
C                       1 = Machine down
C                       2 = Machine busy
C                       3 = Task not detected
C                       4 = Task not performed accurately
C                       5 = Service time too long
C                       6 = Task accomplished successfully
C        mstime    maximum service time
C        mu        service rate
C        perpar    parameter for performance probabilities
C        sta       1 (machine up), 0 (machine down)
C        tarriv    arrival time
C        tbusy     time when machine change from busy to idle (vv)
C        tsta      time when machine change from ON to OFF (vv)
C
      REAL             tsta,tbusy,detpar,perpar,mu,mstime,tarriv,p
      REAL             stime,tcomp
      INTEGER          event,sta,busy
      DOUBLE PRECISION seed
C
C     Check machine status
C
      IF(sta .EQ. 0) THEN
         event = 1
         RETURN
      ENDIF
C                                          Machine busy?
      IF(busy .EQ. 1) THEN
         event = 2
         RETURN
      ENDIF
C
C     Check IF task is detected and performed well
```

```
C                                            Task detected ?
      p = EXP(-detpar*tarriv)
      IF(RND(seed) .GT. p) THEN
         event = 3
         RETURN
      ENDIF
C                                            Task performed well?
      p = EXP(-perpar*tarriv)
      IF(RND(seed) .GT. p) THEN
         event = 4
         RETURN
      ENDIF
C
C     Check allocated time for task
C
      stime = -ALOG(RND(seed))/mu
      tcomp = tarriv+stime
      IF((stime .GT. mstime) .OR. (tcomp .GT. tsta)) THEN
         event = 5
         RETURN
      ENDIF
C
C     Task performed successfully
C     (Machine will be busy until time "tcomp")
C
      busy  = 1
      tbusy = tcomp
      event = 6
      RETURN
      END
```

10 APPENDIX B: PROGRAM II

```
C       Program to simulate Example 2
C
C       Variables:
C
C          alfa1      1/alfa1 = average running time for hardware
C          beta1      1/beta1 = average down time for hardware
C          alfa2      1/alfa2 = average running time for software
C          beta2      1/beta2 = average down time for software
C          busy       1 (machine busy), 0 (machine down)
C          detpar     parameter for detection probabilities
C          seed       seed for random number generator
C          event      type of failure (if any-see SUBROUTINE check)
C          ftype      failure-type frequencies
C          freq(i,1)  number of missions with i tasks
C          freq(i,2)  number of successful missions with i tasks
C          iunit      output unit number
C          lambda     arrival rate
C          mission    current mission number
C          mstime     maximum service time
C          mu         service rate
C          nmiss      total number of missions
C          nulmiss    number of missions without tasks
C          perpar     parameter for performance probabilities
C          sta        1 (machine up), 0 (machine down)
C          sta1       1 (hardware up), 0 (hardware down)
C          sta2       1 (software up), 0 (software down)
C          sucmiss    number of successful missions
C          tarriv     arrival time
C          tavai      total time that machine is available
```

```
C          tbusy     time when machine change from busy to idle (vv)
C          tsta      time when machine change from ON to OFF (vv)
C          tsta1     time when hardware change from ON to OFF (vv)
C          tsta2     time when software change from ON to OFF (vv)
C          ttotal    mission duration
C
C      Variable declaration:
C
       REAL                alfa1,beta1,alfa2,beta2,detpar,lambda,mstime
       REAL                mu,perpar,tarriv,tavai,tbusy,tsta,ttotal
       INTEGER             ftype(10),freq(51,2)
       INTEGER             busy,event,mission,narriv,nmiss
       INTEGER             nulmiss,sta,sucmiss,iunit,sta1,sta2
       DOUBLE PRECISION    seed
       CHARACTER*20        fcause(10)
C
 10    FORMAT(5X,A30,2X,I6,2X,F9.6,I6,A20)
C
C      Variable initialization:
C
       CALL init(ftype,ttotal,alfa1,beta1,alfa2,beta2,mstime,
     +      mu,lambda,detpar,perpar,nmiss,seed,iunit)
       CALL display(ttotal,alfa1,beta1,alfa2,beta2,mstime,
     +      mu,lambda,detpar,perpar,nmiss,seed,iunit)
C
       fcause(1) = ' Machine down       '
       fcause(2) = ' Machine Busy       '
       fcause(3) = ' Task not detected  '
       fcause(4) = ' Task not perf. well'
       fcause(5) = ' Task too long      '
       fcause(6) = ' SUCCESSFUL MISSION '
C
C      Begin simulation
C
       sucmiss = 0
       nulmiss = 0
       tavai   = 0.0
       DO 1000 mission = 1,nmiss
          tsta1  = 0.0
```

```
      sta1   = 0
      CALL component(alfa1,beta1,sta1,tsta1,ttotal,seed)
      tsta2  = 0.0
      sta2   = 0
      CALL component(alfa2,beta2,sta2,tsta2,ttotal,seed)
      tsta   = AMIN1(tsta1,tsta2)
      sta    = sta1*sta2
      tavai  = tavai+tsta*sta
      tbusy  = 0.0
      busy   = 0
      event  = 6
      narriv = 0
      tarriv = -ALOG(RND(seed))/lambda
C
C     Determine whether this mission contains any tasks or not
C
      IF(tarriv .GT. ttotal) THEN
         nulmiss = nulmiss+1
 100     IF(tsta .LT. ttotal) THEN
            CALL system(sta1,sta2,tsta1,tsta2,sta,tsta,tavai,
     +               tavai,alfa1,alfa2,beta1,beta2,ttotal,seed)
            GOTO 100
         ENDIF
      ENDIF
C
 300  IF(tarriv .LE. ttotal) THEN
C
C        Determine machine status (ON/OFF) at the arrival time
C
 600     IF(tarriv .GT. tsta) THEN
            CALL system(sta1,sta2,tsta1,tsta2,sta,tsta,tavai,
     +               tavai,alfa1,alfa2,beta1,beta2,ttotal,seed)
            tbusy = tsta
            busy  = 0
            GOTO 600
         ENDIF
C
C        Determine if task will be completed (event1=6)
C
```

```
            CALL check(event,sta,tsta,busy,tbusy,
     +           detpar,perpar,mu,mstime,tarriv,seed)
C
C           If task was not completed then determine the number of
C           tasks & consider the next mission
C
            IF(event .NE. 6) THEN
 800           IF(tarriv .LT. ttotal) THEN
                  tarriv = tarriv-ALOG(RND(seed))/lambda
                  narriv = narriv+1
                  CALL system(sta1,sta2,tsta1,tsta2,sta,tsta,
     +                 tavai,alfa1,alfa2,beta1,beta2,ttotal,seed)
                  GOTO 800
               ENDIF
               GOTO 900
            ENDIF
C
C           Task completed, consider next task
C
            tarriv = tarriv-ALOG(RND(seed))/lambda
            narriv = narriv+1
            IF(tarriv .GT. tbusy) THEN
               busy  = 0
               tbusy = tarriv
            ENDIF
C
            GOTO 300
         ENDIF
C
C        Mission completed, collect statistics
C
 900     IF(narriv .GT. 50) narriv = 51
         ftype(event) = ftype(event)+1
         IF(narriv .GT. 0) THEN
            freq(narriv,1) = freq(narriv,1)+1
            IF(event .EQ. 6) freq(narriv,2) = freq(narriv,2)+1
         ENDIF
C
C        If a simulation run completed, print a report
```

```
C
         IF(50*(mission/50) .EQ. mission) THEN
            CALL state(nulmiss,tavai,ttotal,freq,51,iunit)
         ENDIF
C
 1000  CONTINUE
C
C      Print final report and end simulation
C
       CALL report(nulmiss,tavai,ttotal,freq,
     +      51,ftype,fcause,6,iunit)

       CLOSE(iunit)
       STOP ' '
       END
C
C
C
       SUBROUTINE init(ftype,ttotal,alfa1,beta1,alfa2,beta2,
     +            mstime,mu,lambda,detpar,perpar,nmiss,
     +            seed,iunit)
C
C      Subroutine to read in parameters for simulation
C
       REAL                ttotal,alfa1,beta1,mstime,mu,lambda,detpar
       REAL                perpar,alfa2,beta2
       INTEGER             ftype(15),nmiss,iunit
       DOUBLE PRECISION seed
       CHARACTER*45        msg(15),flnm
C
       msg(1)  = '   Man-Machine   System   Simulation        '
       msg(2)  = 'Enter seed for random number generator   = '
       msg(3)  = 'Enter total number of simulation runs    = '
       msg(4)  = 'Enter mission duration,                T = '
       msg(5)  = 'Enter hardware breakdown rate,    Alpha1 = '
       msg(6)  = 'Enter hardware repair rate,        Beta1 = '
       msg(7)  = 'Enter service rate,                   Mu = '
       msg(8)  = 'Enter maximum allocated service time,  U = '
       msg(9)  = 'Enter arrival rate,              Lambda  = '
```

```
      msg(10) = 'Enter detection-prob. parameter,      D  = '
      msg(11) = 'Enter performance-prob. parameter,    P  = '
      msg(12) = 'A positive input is required !   RE-ENTER  '
      msg(13) = 'Enter output filename                    = '
      msg(14) = 'Enter software breakdown rate,   Alpha2  = '
      msg(15) = 'Enter software repair rate,       Beta2  = '
C
 10   FORMAT(10X,A45,\)
C
      WRITE(*,10) msg(1)
      WRITE(*,10)
      WRITE(*,10)
 15   WRITE(*,10) msg(2)
      READ(*,*) seed
      IF(seed .LT. 0) THEN
         WRITE(*,10) msg(12)
         WRITE(*,*)
         GOTO 15
      ENDIF
 20   WRITE(*,10) msg(3)
      READ(*,*) nmiss
      IF(nmiss .LT. 0) THEN
         WRITE(*,10) msg(12)
         WRITE(*,*)
         GOTO 20
      ENDIF
      nmiss = nmiss*50
 25   WRITE(*,10) msg(4)
      READ(*,*) ttotal
      IF(ttotal .LT. 0) THEN
         WRITE(*,10) msg(12)
         WRITE(*,*)
         GOTO 25
      ENDIF
 30   WRITE(*,10) msg(5)
      READ(*,*) alfa1
      IF(alfa1 .LT. 0) THEN
         WRITE(*,10) msg(12)
         WRITE(*,*)
```

```
         GOTO 30
      ENDIF
32    WRITE(*,10) msg(6)
      READ(*,*) beta1
      IF(beta1 .LT. 0) THEN
         WRITE(*,10) msg(12)
         WRITE(*,*)
         GOTO 32
      ENDIF
35    WRITE(*,10) msg(14)
      READ(*,*) alfa2
      IF(alfa2 .LT. 0) THEN
         WRITE(*,10) msg(12)
         WRITE(*,*)
         GOTO 35
      ENDIF
37    WRITE(*,10) msg(15)
      READ(*,*) beta2
      IF(beta2 .LT. 0) THEN
         WRITE(*,10) msg(12)
         WRITE(*,*)
         GOTO 37
      ENDIF
40    WRITE(*,10) msg(7)
      READ(*,*) mu
      IF(mu .LT. 0) THEN
         WRITE(*,10) msg(12)
         WRITE(*,*)
         GOTO 40
      ENDIF
45    WRITE(*,10) msg(8)
      READ(*,*) mstime
      IF(mstime .LT. 0) THEN
         WRITE(*,10) msg(12)
         WRITE(*,*)
         GOTO 45
      ENDIF
50    WRITE(*,10) msg(9)
      READ(*,*) lambda
```

```
      IF(lambda .LT. 0) THEN
         WRITE(*,10) msg(12)
         WRITE(*,*)
         GOTO 50
      ENDIF
 55   WRITE(*,10) msg(10)
      READ(*,*) detpar
      IF(detpar .LT. 0) THEN
         WRITE(*,10) msg(12)
         WRITE(*,*)
         GOTO 55
      ENDIF
 60   WRITE(*,10) msg(11)
      READ(*,*) perpar
      IF(seed .LT. 0) THEN
         WRITE(*,10) msg(12)
         WRITE(*,*)
         GOTO 60
      ENDIF
      WRITE(*,10) msg(13)
      READ(*,'(A15)') flnm
C
      DO 100 i = 110
         ftype(i) = 0
 100  CONTINUE
C
      iunit = 6
      OPEN(iunit,FILE = FLNM,STATUS = 'NEW')
      RETURN
      END
C
C
C
      SUBROUTINE report(nulmiss,tavai,ttotal,freq,nfreq,
     +            ftype,fcause,ntype,iu)
C
C     This subroutine computes the measures of effectiveness and
C     prints out the final report
C
```

```
      INTEGER          nulmiss,nfreq,i,totmiss,sucmiss,k,ntype
      INTEGER          failmiss,freq(nfreq,2),ftype(ntype),iu
      REAL             tavai,ttotal,avail,eff1,eff2,eff3,f1,f2
      CHARACTER*20     fcause(ntype)
      CHARACTER*70     tt1,tt2,tt3,tt4,tt5,tt6
C
 10   FORMAT(5X,A41,F14.6)
 20   FORMAT(5X,A41,I6)
 30   FORMAT(5X,A70)
 35   FORMAT(5X,A41)
 40   FORMAT(5X,3(3X,I8),F9.5,2X,F9.5)
 50   FORMAT(8X,I10,2(3X,F10.6))
 60   FORMAT(5X,A20,2X,I10,4X,F14.7)
C
      tt1='       Breakdown by Number of Jobs per Mission        '
      tt2='             Absolute Frequencies  Relative Frequencies'
      tt3='Jobs/mission   Total   Successful    Total  Successful'
      tt4='======================================================'
      tt5='            Breakdown by Cause of the Failure          '
      tt6=' Failure Cause     Abs. Frequency    Rel. Frequency    '
C
      k = 0
      totmiss = nulmiss
      sucmiss = 0
      DO 100 i = 1,nfreq
         totmiss = totmiss+freq(i,1)
         sucmiss = sucmiss+freq(i,2)
         IF(freq(i,1) .NE. 0) k = i
 100  CONTINUE
      failmiss = totmiss-sucmiss-nulmiss
      avail    = tavai/(ttotal*totmiss)
      eff1     = FLOAT(sucmiss+nulmiss)/FLOAT(totmiss)
      eff2     = FLOAT(sucmiss)/FLOAT(totmiss-nulmiss)
      eff3     = (sucmiss+nulmiss*avail)/FLOAT(totmiss)
C
      WRITE(iu,*)
      WRITE(iu,35)'            Man-Machine System Simulation'
      WRITE(iu,*)
      WRITE(iu,35)'                    Final Report         '
```

```
      WRITE(iu,*)
      WRITE(iu,20)'Total number of missions :............',totmiss
      WRITE(iu,20)'Total number of successful missions:..',sucmiss
      WRITE(iu,20)'Total number of failed missions:......',failmiss
      WRITE(iu,20)'Number of missions without any task:..',nulmiss
      WRITE(iu,*)
      WRITE(iu,10)'     Availability...................... ',avail
      WRITE(iu,10)'     Effec.1........................... ',eff1
      WRITE(iu,10)'     Effec.2........................... ',eff2
      WRITE(iu,10)'     Effec.3........................... ',eff3
      WRITE(iu,*)
      WRITE(iu,30) tt1
      WRITE(iu,30) tt2
      WRITE(iu,30) tt4
      WRITE(iu,30) tt3
      WRITE(iu,*)
      f1 = FLOAT(nulmiss)/FLOAT(totmiss)
      WRITE(iu,40) 0,nulmiss,nulmiss,f1,f1
      DO 200 i = 1,k
         f1 = FLOAT(freq(i,1))/FLOAT(totmiss)
         f2 = FLOAT(freq(i,2))/FLOAT(totmiss)
         WRITE(iu,40) i,freq(i,1),freq(i,2),f1,f2
 200  CONTINUE
      WRITE(iu,30) tt4
C
      WRITE(iu,*)
      WRITE(iu,30) tt5
      WRITE(iu,30) tt6
      WRITE(iu,30) tt4
      WRITE(iu,*)
      DO 400 i = 1,ntype
         f1 = FLOAT(ftype(i))/FLOAT(totmiss)
         WRITE(iu,60) fcause(i),ftype(i),f1
 400  CONTINUE
      WRITE(iu,30) tt4
      WRITE(iu,*)
C
      RETURN
      END
```

```
C
C
C
      SUBROUTINE state(nulmiss,tavai,ttotal,freq,nfreq,iu)
C
C     This subroutine computes the measures of effectiveness and
C     prints them out
C
      INTEGER          nulmiss,nfreq,i,totmiss,sucmiss,k
      INTEGER          failmiss,freq(nfreq,2),iu
      REAL             tavai,ttotal,avail,eff1,eff2,eff3,f1,f2
      CHARACTER*70     tt1,tt2,tt3
C
 10   FORMAT(5X,I7,4X,4(F9.6,3X))
 30   FORMAT(5X,A70)
C
      k = 0
      totmiss = nulmiss
      sucmiss = 0
      DO 100 i = 1,nfreq
         totmiss = totmiss+freq(i,1)
         sucmiss = sucmiss+freq(i,2)
         IF(freq(i,1) .NE. 0) k = i
 100  CONTINUE
      nreport = totmiss/50
      failmiss = totmiss-sucmiss-nulmiss
      avail    = tavai/(ttotal*totmiss)
      eff1     = FLOAT(sucmiss+nulmiss)/FLOAT(totmiss)
      eff2     = FLOAT(sucmiss)/FLOAT(totmiss-nulmiss)
      eff3     = (sucmiss+nulmiss*avail)/FLOAT(totmiss)
C
      IF(nreport .EQ. 1) THEN
         tt1='                 Man-Machine System Simulation          '
         tt2='Mission No.   Avail.     SE1     SE2     SE3'
         tt3='======================================================'
         WRITE(iu,*)
         WRITE(iu,30) tt1
         WRITE(iu,*)
         WRITE(iu,30) tt2
```

```
            WRITE(iu,30) tt3
            WRITE(iu,*)
         ENDIF
C
         WRITE(iu,10) totmiss,avail,eff1,eff2,eff3
         RETURN
         END
C
C
C
         SUBROUTINE display(ttotal,alfa1,beta1,alfa2,beta2,
     +               mstime,mu,lambda,detpar,perpar,
     +               nmiss,seed,iu)
C
C        Subroutine to display parameters for simulation
C
         REAL              ttotal,alfa1,beta1,mstime,mu,lambda,detpar
         REAL              perpar,tarriv,alfa2,beta2
         INTEGER           nmiss,iu
         DOUBLE PRECISION  seed
C
 10      FORMAT(5X,A37,F14.6)
 20      FORMAT(5X,A37,I6)
 30      FORMAT(5X,A37)
C
         WRITE(iu,*)
         WRITE(iu,30) ' Man-Machine System Simulation        '
         WRITE(iu,*)
         WRITE(iu,30) '      Simulation Parameters           '
         WRITE(iu,*)
         WRITE(iu,10) ' Random number generator seed      = ',seed
         WRITE(iu,10) ' Total time per mission        (T) = ',ttotal
         WRITE(iu,20) ' Total number of missions          = ',nmiss
         WRITE(iu,10) ' hardware repair rate  (Beta1)     = ',beta1
         WRITE(iu,10) ' hardware breakdown rate  (Alpha1) = ',alfa1
         WRITE(iu,10) ' software repair rate     (Beta2)  = ',beta2
         WRITE(iu,10) ' software breakdown rate  (Alpha2) = ',alfa2
         WRITE(iu,10) ' Service rate,                (Mu) = ',mu
         WRITE(iu,10) ' Max. allocated service time   (U) = ',mstime
```

```
      WRITE(iu,10) ' Arrival rate              (Lambda) = ',lambda
      WRITE(iu,10) ' Detection-prob. parameter     (D) = ',detpar
      WRITE(iu,10) ' Performance-prob. parameter   (P) = ',perpar
      WRITE(iu,*)
C
      RETURN
      END
C
C
C
      SUBROUTINE component(alfa,beta,sta,tsta,ttotal,seed)
C
C     This procedure updates the component status
C
C     Parameters:
C        1/alfa    mean up time
C        1/beta    mean down time
C        seed      seed for random number generator
C        sta       0 (component down), 1 (component up)
C        tsta      time at which next status change will occur
C        ttotal    mission duration
C
      REAL             alfa,beta,tsta,ttotal,rate,t
      INTEGER          sta
      DOUBLE PRECISION seed
C
      sta = 1-sta
      rate = (1-sta)*beta+sta*alfa
      t    = AMIN1(-ALOG(RND(seed))/rate,ttotal-tsta)
      tsta = tsta+t
      RETURN
      END
C
C
C
      SUBROUTINE system(s1,s2,t1,t2,s,t,tavai,a1,a2,
     +           b1,b2,ttotal,seed)
C
C     This procedure updates the system status
```

```
C
C      Parameters:
C
C         s1         hardware status (1=ON, 0=OFF)
C         s2         software status (1=ON, 0=OFF)
C         t1         time at which hardware status will change
C         t2         time at which software status will change
C         s          system status (1=ON, 0=OFF)
C         t          time at which system status will change
C         tavai      total time that the system has been available
C         1/a1       mean up-time for hardware
C         1/a2       mean up time for software
C         1/b1       mean down-time for hardware
C         1/b2       mean down-time for software
C         ttotal     mission duration
C         seed       random-number generator seed
C
       INTEGER          s1,s2,s
       REAL*4           t1,t2,t,tavai,a1,a2,b1,b2,ttotal,told
       REAL*8           seed
C
       IF(t1 .LE. t2) THEN
          CALL component(a1,b1,s1,t1,ttotal,seed)
       ELSE
          CALL component(a2,b2,s2,t2,ttotal,seed)
       END IF
C
       IF(s .EQ. 0) THEN
          s    = 1
          told = t
          IF(s1*s2 .NE. 1) THEN
             WRITE(*,*) '     Error 1 in SUBROUTINE system'
             WRITE(*,*) s1,t1,s2,t2
             STOP
          END IF
          t     = AMIN1(t1,t2)
          tavai = tavai+(t-told)
          RETURN
C
```

```
      ELSE IF(s .EQ. 1) THEN
         s = 0
100      IF(s1*s2 .NE. 0) THEN
            WRITE(*,*) '      Error 2 in SUBROUTINE system'
            WRITE(*,*) s1,t1,s2,t2
            STOP
         END IF
         IF (s1 .EQ. 0) THEN
200         IF(t2 .LT. t1) THEN
               CALL component(a2,b2,s2,t2,ttotal,seed)
               GOTO 200
            END IF
            IF(s2 .EQ. 1) THEN
               t = t1
               RETURN
            END IF
         END IF
C
         IF (s2 .EQ. 0) THEN
300         IF(t1 .LT. t2) THEN
               CALL component(a1,b1,s1,t1,ttotal,seed)
               GOTO 300
            END IF
            IF(s1 .EQ. 1) THEN
               t = t2
               RETURN
            END IF
         END IF
         GOTO 100
      END IF
C
      WRITE(*,*) '      Error 3 in SUBROUTINE system'
      WRITE(*,*) s1,t1,s2,t2
      STOP
      END
C
      SUBROUTINE check(event,sta,tsta,busy,tbusy,detpar,
     +           perpar,mu,mstime,tarriv,seed)
C
```

```
C      This subroutine checks whether a task can be completed.
C
C      Parameters:
C
C         busy      1 (machine busy), 0 (machine down)
C         detpar    parameter for detection probabilities
C         seed      seed for random number generator
C         event     indicates failure type:
C                        1 = Machine down
C                        2 = Machine busy
C                        3 = Task not detected
C                        4 = Task not performed accurately
C                        5 = Service time too long
C                        6 = Task accomplished successfully
C         mstime    maximum service time
C         mu        service rate
C         perpar    parameter for performance probabilities
C         sta       1 (machine up), 0 (machine down)
C         tarriv    arrival time
C         tbusy     time when machine change from busy to idle (vv)
C         tsta      time when machine change from ON to OFF (vv)
C
       REAL             tsta,tbusy,detpar,perpar,mu,mstime,tarriv,p
       REAL             stime,tcomp
       INTEGER          event,sta,busy
       DOUBLE PRECISION seed
C
C      Check machine status
C
       IF(sta .EQ. 0) THEN
          event = 1
          RETURN
       ENDIF
C                                           Machine busy?
       IF(busy .EQ. 1) THEN
          event = 2
          RETURN
       ENDIF
C
```

```
C       Check IF task is detected and performed well
C
C                                                Task detected ?
        p = EXP(-detpar*tarriv)
        IF(RND(seed) .GT. p) THEN
           event = 3
           RETURN
        ENDIF
C                                                Task performed well?
        p = EXP(-perpar*tarriv)
        IF(RND(seed) .GT. p) THEN
           event = 4
           RETURN
        ENDIF
C
C       Check allocated time for task
C
        stime = -ALOG(RND(seed))/mu
        tcomp = tarriv+stime
        IF((stime .GT. mstime) .OR. (tcomp .GT. tsta)) THEN
           event = 5
           RETURN
        ENDIF
C
C       Task performed successfully
C       (Machine will be busy until time "tcomp")
C
        busy  = 1
        tbusy = tcomp
        event = 6
        RETURN
        END
C
```