Markov Reliability Models for Digital Flight Control Systems

John McGough*
Allied/Bendix Aerospace, Teterboro, New Jersey
and
Andrew Reibman† and Kishor Trivedi‡
Duke University, Durham, North Carolina

The reliability of digital flight control systems can often be accurately predicted using Markov chain models. We begin our discussion of flight control system reliability models with definitions of key terms. We then construct a single-fault one-processor model based on the results of fault-injection experiments. To illustrate more complex system models, we consider several models of a triple modular redundant system. Once we have constructed some representative Markov reliability models, we discuss numerical techniques for their solution. The cost of numerical solution depends on a model's size and stiffness. Acyclic Markov models, a useful special case, are particularly amenable to efficient numerical solution. Even in the general case, instantaneous coverage approximation allows the reduction of some cyclic models to more readily solvable acyclic models. After considering the solution of single-phase models, we extend our discussion to phased-mission models. We classify phased-mission reliability models based on the state restoration behavior that occurs between mission phases. As an economical approach for the solution of such models, we introduce the mean failure rate solution method. We use a numerical example to show the influence of fault-model parameters and interphase behavior on system unreliability.

I. Introduction

THE high reliability requirements associated with digital flight control systems (FCS) make life testing and simulation expensive. Analytic models provide an economical supplement to these conventional reliability prediction methods. Traditional analytic reliability models are combinatorial, relying solely on component count and estimated component reliability to predict system reliability. However, fault-tolerant FCS may have complex, state-dependent behavior that cannot be accurately modeled with simple combinatorial models.

For a single-phase mission, the fault processes associated with redundant digital FCS can be satisfactorily modeled with Markov or semi-Markov chains.[1,2] States of a Markov reliability model correspond to the modeled system's operational state (e.g., what components have failed). Once the Markov chain's state probability vector is computed, system unreliability is given by the probability of being in a "down" state. In both Markov and semi-Markov models, state transitions obey the Markovian property, i.e., each entry into a state is a regeneration point that erases the influence of the past. Homogeneous Markov chains further restrict state occupancy times to be exponentially distributed and state transition rates to be time-independent. In electronic systems, this restriction is reasonable, as one can often assume that components have constant failure rates. In mechanical systems such as airframes or engines, component aging and fatigue play an important role in system failures, thus, homogeneous Markov chains are usually inadequate.

Nonhomogeneous and semi-Markov chains relax the restriction of exponential holding times, making them applicable to a wider range of fault scenarios, but also more expensive to analyze. For example, individual components might have Weibull life distributions.[3] A semi-Markov model can often be accurately approximated by a homogeneous Markov chain.[4] Even if such an approximation is not feasible, we can still use a reliability prediction program. Programs that allow various types of semi-Markov and nonhomogeneous Markov models include CARE III,[1] HARP,[2] and SHARPE.[5]

Although Markov models are mathematically more tractable than semi-Markov models, their solution is still expensive. Solution time increases with the number of states and transition arcs in the model, the length of the solution interval, and the model's stiffness.[6] For acyclic models, less expensive algorithms can be used to obtain the state occupancy probabilities. In particular, it is possible to obtain a closed-form solution in the mission time variable. The cost of a closed-form solution is independent of mission time. Many reliability models associated with FCS architectures are acyclic. If they are not acyclic, they often can be transformed into an acyclic model by using an instantaneous coverage approximation.[7] An important property of the instantaneous coverage approximation as discussed in Ref. 7 is that it is conservative, i.e., the estimated probability of loss of control in the approximate model is greater than the estimate obtained from the original model (in a class of reliability models defined in Ref. 7).

Both the size of a model and its stiffness influence the choice of a solution approach. Stiff problems have different kinds of events occurring at greatly different rates. For example, the time between FCS component failures is much longer than the duration of a typical system reconfiguration. Stiffness as a meausre of numerical difficulty is approximated by the product of the largest transition rate and the mission time.[6] Size is usually measured by the number of states in the model, denoted n. An alternative measure of size is ξ, the number of nonzero transition arcs in the model. For the reliability evaluation of a single-phase mission, "small" models may be easily solved by hand computation. They typically have fewer than a dozen states. "Medium-sized" models require computer solution but do not require special treatment because of their size; $O(n^3)$ floating point operations and $O(n^2)$ storage are acceptable computational costs. (A function $f(n)$ is $O[g(n)]$ if there exists a constant c such that $f(n) \leq cg(n)$ for all but a finite (possibly empty) set of nonnegative values of n.) Medium-sized models generally have at most a few hundred states. However,

Received June 17, 1987; revision received Jan. 25, 1988.

*Principal Engineer, Flight Systems Division.
†Currently, Technical Staff Member, AT&T Bell Laboratories.
‡Professor, Computer Science Department.

numerical difficulties are still possible in medium-sized models. For "large" models with ξ nonzero transition arcs, we are usually restricted to using sparse matrix methods that require at most $O(\xi)$ storage. Approximation methods may also be needed to reduce the size of the model or eliminate stiffness. Techniques for large, stiff models are investigated in detail in Refs. 6 and 8.

Typically, modelers analyze the reliability of an FCS over a single phase or flight. Over the course of a system's useful life, many flights are conducted and component failures and repairs occur. We refer to repairs that occur between missions or phases as "maintenance." "Repairs" refer explicitly to repairs that are made during missions or phases. Phased-mission reliability analysis may increase the difficulty of model solution. With multiphase missions, mission time is increased, thus magnifying stiffness. Interphase maintenance must also be considered to get a true picture of a system's reliability over a long time period.

In most FCS reliability analyses, the probability of the loss of control during a single mission is obtained by combinatorial methods. This technique is based on the assumption that the FCS is fault-free at the start of the mission. When, due to accumulating latent faults or imperfect repairs, this assumption is not valid, combinatorial methods break down. A complete time-dependent solution can be difficult to use as a metric for comparing different FCS configurations. To facilitate such comparisons, an alternate measure of reliability, called the mean failure rate (MFR), is proposed. MFR is defined as the reciprocal of the mean time to first failure (MTFF). Using MFR reduces computation costs and provides a single, time-independent figure of merit for system reliability. It is especially useful in sensitivity analyses and as a basis for comparing different FCS configurations. Unfortunately, MFR does provide less information on system behavior than a conventional time-dependent reliability analysis. We can use the expression $1 - e^{(-\mathrm{MFR} * t)} \approx \mathrm{MFR} * t$ to estimate the probability of loss of control on or before time t. Although this approximation is coarse for a single-phase reliability model, it is more appropriate for evaluating phased-mission models, particularly if the probability of failure in any given phase is small.

The rest of the paper is organized as follows. In the first half of the paper, we discuss (homogeneous) Markov FCS reliability models. We begin with definitions of key terminology and of Markov chains. Based on the results of fault-injection experiments, a basic single-fault Markov model for one processor is defined. Using this model as a basis, we develop a sequence of models for triple modular redundant (TMR) systems. We discuss algorithms for the solution of homogeneous Markov models (or the single phase of a multiphase Markov model). We then consider improved solution approaches for acyclic models. Instantaneous coverage approximation replaces some cyclic models with more easily solved acyclic ones. Finally, to demonstrate the application of our results, we numerically compute the unreliability of a three-processor system. We consider the effect of varying model parameters on system unreliability.

In the second half of the paper, we consider phased-mission models. We divide these models into several classes based on their interphase behavior. For phased-mission models, we describe numerical techniques for MFR solution. We use a numerical example to show the utility of MFR solution and the effect of interphase behavior on system unreliability.

II. Markov FCS Reliability Models

We begin this section with a definition of key terminology needed to describe FCS reliability models. We also briefly review Markov chains, the mathematical structures that are used to describe and analyze these models. As a first example, we present a single-fault Markov model for the behavior of one processor. Then, to illustrate the construction of more complex FCS reliability models, we describe several multiple-fault Markov models for TMR FCS, using the single-fault model as a basis.

FCS Reliability Model Terminology

We first define some basic FCS reliability model terminology. The reliability analysis of a system is conducted over an interval of time that we will call the *mission*. The mission can be divided into *phases*. During each phase, a distinct Markov chain rate matrix may describe system behavior. If a mission consists of more than one flight, each flight may be a distinct phase, separated by maintenance. Alternatively, if the mission is a single flight, mission phases might be takeoff, normal flight, and landing. The following terms are useful in describing the behavior of the system being modeled:

Fault: An internal condition of a device that causes a malfunction for some combination of input and internal state.

Error: An erroneous output generated as a result of a fault.

System failure: Loss of control (LOC).

Permanent fault: A fault that persists until the device is repaired or replaced.

Intermittent fault: An intermittent fault has two possible states: active and benign. Only an active intermittent fault can produce an error. The fault persists until the device is repaired or replaced.

Transient fault: A fault that occurs for a brief period and then vanishes.

Many hardware faults produce errors only when the faulty component is exercised by control software, i.e., an appropriate combination of input and internal state is reached. Thus, many faults produce errors intermittently.[9] However, a fault that produces an error intermittently is not necessarily an intermittent fault. An intermittent fault does not always produce an error when exercised. For example, intermittent faults can result from cracked leads or borderline voltage conditions that are excited by vibration or electromagnetic interference.

Fault-injection experiments provide the basis for fault models.[10-12] In such experiments, faults are injected into real or simulated flight control computers. Errors are detected by comparing the outputs of faulty and nonfaulty computers. The experiments cited suggest 1) faults produce errors (at comparators) only if activated by operating software; 2) most errors are detected shortly after their occurrence; 3) most faults are activated by the baseline software program, i.e., the executive and inner loops; 4) only a small proportion of faults are activated by an auxiliary program, e.g., an outer loop, and 5) all faults that are activated by a particular program produce errors at approximately the same rate. These observations suggest the following definitions for use in FCS models:

Baseline program: A set of continuously executed software modules.

Auxiliary programs: Software executed occasionally.

Active fault: A fault that can produce an error (for some input) while executing the current program.

α-fault: A fault activated by the baseline program.

Benign fault: A fault that cannot produce an error while executing the current program, regardless of input. It may produce an error for some other program.

β-fault: A fault activated only by the auxiliary program, not by the baseline program.

Typically, the baseline program is the operating system kernel. An application program executed on demand would be considered an auxiliary program. Suppose a hardware fault exists in memory that is exercised only by an auxiliary program and never read by the operating system. If the fault were activated only when the memory location was accessed, it would be a β-fault; it can produce an error only during the operation of the auxiliary program.

Markov Model Definition

Consider homogeneous, continuous-time, Markov chain[3] (CTMC) $[X(t), t \geq 0]$, with discrete state space Ω. The state

space corresponds to the possible states of the modeled FCS. For example, each possible combination of operational and failed hardware components might have a corresponding state in the CTMC. We assume the states can be numbered 1,...,n, and that the nth state corresponds to system failure (LOC). If $q_{ji}, i \neq j$, denotes the transition rate from state i to state j, and $q_{ii} = -\Sigma_{j=1, j\neq i}^{n} q_{ji}$, then the matrix $Q = [q_{ij}]$ is called the *infinitesimal generator* of the CTMC. We let ξ denote the number of nonzero entries in Q. For the homogeneous models we consider, Q is a time-independent constant matrix. If $P_i(t)$ denotes the probability the system is in state i at time t, the row vector $\boldsymbol{P}(t)$ is called the *state probability vector*. The behavior of the chain can be described by the system of Kolmogorov differential equations

$$\boldsymbol{P}(t) = \boldsymbol{Q}\boldsymbol{P}(t), \qquad \boldsymbol{P}(0) = \Pi_0 \tag{1}$$

We display Markov models using a state transition graph like the one shown in Fig. 1. In these graphs circles represent states of the Markov chain, and arcs represent transitions. Associated with each arc is its transition rate. In the rest of this section, we develop sample Markov FCS reliability models. In the next section, we discuss the numerical transient solution of Markov models.

Single-Fault Model

To illustrate the concept of Markov reliability model and to provide a basis for multiprocessor models, we define a single-fault model for one processor. This "single-fault" Markov model is given in Fig. 1. The model was designed to portray the results of fault-injection experiments where only one fault at a time was injected into a processor. Referring to Fig. 1, the model states correspond to the following system states:

Model state	System state
GOOD,A	No faults; auxiliary program on-line (active)
GOOD,I	No faults; auxiliary program off-line (inactive)
α	An active, α-fault has occurred
a	A β-fault has occurred; auxiliary program on-line (active)
b	A β-fault has occurred; auxiliary program off-line (inactive)

Because only single faults are portrayed, there is no transition from the β-fault states to the α-fault state. (Such transitions might be allowed in a multiple-fault model.) The single-fault model of Fig. 1 does not include intermittent faults, even though some faults (e.g., β-faults) produce errors intermittently. Transient faults are also omitted. The model still has sufficient degrees of freedom to capture much of the behavior observed in fault-injection experiments. The following parameters describe the fault and error dynamics for the single-fault model that allows only permanent faults:

Parameter	Interpretation
λ	Failure rate of a processor (failures/h)
e	Rate of error production (errors/h)
p	Fraction of faults that are α-faults
$1-p$	Fraction of faults that are β-faults
$1/ds$	Mean duration of an auxiliary program (h/call)
as	Rate at which auxiliary program is called (calls/h)

Two approximations can be used to simplify the single-fault model. If we assume that active faults produce errors instantaneously, we obtain the *collapsed* single-fault model in Fig. 2. Because most faults produce errors shortly after they are activated (i.e., $\lambda \ll e$), this is often a reasonable approximation.[12] In the terminology of Ref. 8, states α and a are fast transient states. They can be replaced by probabilistic branch points. In the collapsed model there are no active fault states. Note that a model that allows the occurrence of two permanent faults in a processor, the first a benign β-fault and the second an α-fault, is represented by an almost identical Markov chain, with the transition rate as from state "b" to state "error" replaced by $p\lambda + as$. We still ignore a second β-fault in a processor.

A second approximation is to assume the auxiliary program is called so frequently and completes its task so rapidly that its calling/completion behavior can be treated as if it were in steady state. In the terminology of Ref. 8, the states (GOOD,A) and (GOOD,I) form a fast recurrent subset. We can assume that their state probabilities satisfy (approximately) the relation

$$(ds)P_{\text{Good,A}}(t) = (as)P_{\text{Good,I}}(t)$$

Using this assumption, they can be replaced by a single slow state. The resulting collapsed, steady-state model is given in Fig. 3. Here, $p_1 = p$ is the probability that a fault is an α-fault, p_2 is the probability that a fault is an active β-fault at the time of its occurrence, and p_3 is the probability that a fault is a benign β-fault at the time of its occurrence. The relation $p_1 + p_2 + p_3 = 1$ implies

$$p_2 = (1-p_1)\left[\frac{as}{as+ds}\right]$$

and

$$p_3 = (1-p_1)\left[\frac{ds}{as+ds}\right]$$

These simple single-fault models form the basis for the multiprocessor system models discussed in the next subsection. Here, and in the rest of the models in the paper, we assume that there is only a single auxiliary program. If more than one

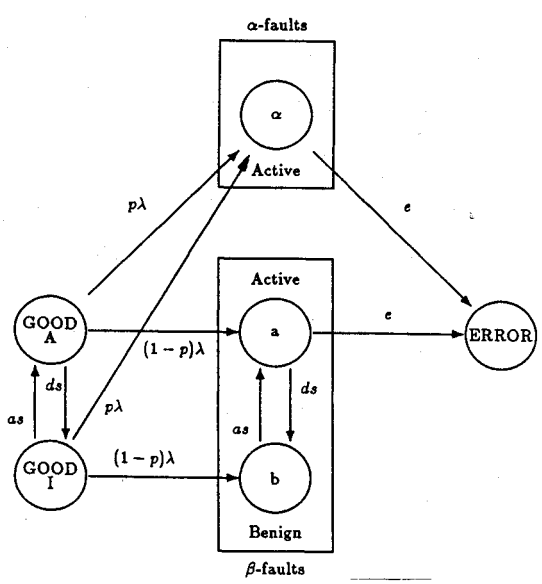

Fig. 1 Simple single-fault model.

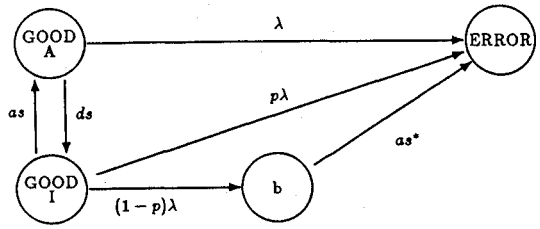

Fig. 2 Fault-collapsed, single-fault model. (In a multiple-fault model, the starred parameter is $p\lambda + as$).

auxiliary program were involved, the complexity of the model would increase. The presence of multiple latent faults in a single processor could further complicate the modeling problem. In generating fault models, the modeler's most important decision is: "what degree of detail is required?" To make this assessment, we need to conduct parametric sensitivity studies using simple models. Once critical parameters have been identified, we can expand models to include additional details where appropriate.

Multiple-Fault Models of TMR Systems

Flight control computers use redundancy to tolerate faults. Because a single fault does not usually cause system failure in a redundant system, multiple faults must be modeled. A typical FCS consists of three or more independent processors, each executing the same software. Computed control surface commands are exchanged via interprocessor data links and compared. A processor that disagrees with the results obtained by a second processor declares that the second processor failed. If a majority of the processors declare a processor failed, then the errant processor is disengaged, i.e., prevented from driving its associated actuators and participating in future voting. For simplicity of explanation and solution, we assume that comparator coverage is 100%; when an error is produced it is either immediately detected by comparison between the processors or results in LOC. This perfect coverage assumption is reasonable because the comparators involved are simple reliable circuits, although, if necessary, the models introduced can easily be extended to include imperfect coverage. In multiple-fault models, we conservatively assume that LOC occurs whenever, following the production of an error, nonfailed processors are not a majority of the active processors.

To illustrate the construction of multiple-fault models, we consider a sequence of TMR FCS models, using the single-fault models of Figs. 1–3 to describe the fault and error dynamics of a single processor.

Simple TMR Models

In our basic model of a multi-processor system, we assume that all faults are α-faults and produce errors instantaneously. Thus, for each processor, we use the collapsed fault model of Fig. 3 and set $p_1 = 1.0$. We also assume that all errors are detected and the faulty unit disengaged with 100% coverage, although, faulty units are not replaced. In the resulting simple model, LOC is due exclusively to exhaustion of redundancy; LOC occurs if and only if at least $n - 1$ of the n processors fail during the mission. Figure 4 illustrates this reliability model for a conventional triplex system. Here state (i,j) represents i good and j failed processors. LOC is the system failure state. Arcs drawn as thick arrows represent interflight maintenance that is discussed later in the paper.

Figure 5 described a TMR FCS with an unlimited number of unpowered spares. (Unpowered spares are assumed to have a zero failure rate). Only α-faults are considered. As in Fig. 4, state (i,j) represents i good and j failed processors, while LOC is the system failure state. When a fault is detected, the faulty processor is immediately replaced by a spare. However, a potential problem arises if a second failure occurs in a different processor before the first failure is detected. (In the terminology of Ref. 13, this is a *near-coincident fault*.) In this case, it is possible that the three processors produce different commands, confusing the voting process. The model of Fig. 5 conservatively assumes that this scenario results in LOC.

TMR Models with β-faults

We can extend the basic TMR model to include β-faults. In a tightly coupled multiprocessor system, every processor executes the same sequence of instructions. A model of the conventional TMR FCS with β-faults is illustrated in Fig. 6. For simplicity we continue to assume that active faults produce errors instantaneously, and use the collapsed fault model, Fig. 2, with the rate given in the footnote, to represent the behavior of a single processor. However, we now assume that

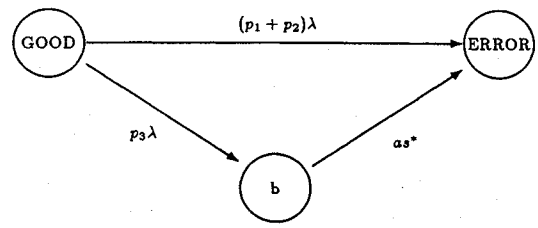

Fig. 3 Program steady-state, fault-collapsed, single-fault model. (In a multiple-fault model, this parameter is $p\lambda + as$).

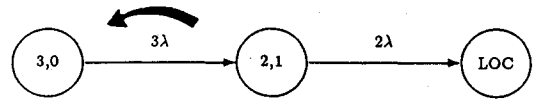

Fig. 4 Simple triplex FCS model. LOC due to exhaustion of processors; $(i,j) \equiv i$ good and j failed processors, LOC $\equiv$ loss of control.

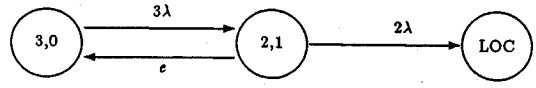

Fig. 5 Triplex FCS model with unlimited spares. LOC due only to near-coincident faults; $(i,j) \equiv i$ good and j failed processors, LOC $\equiv$ loss of control.

$p < 1.0$, i.e., not all faults are α-faults. State (i,j,k) corresponds to i nonfailed processors, j processors with benign faults, and k processors failed (produced errors). Recall that the system being modeled has only one auxiliary program. The I denotes an inactive auxiliary program and the A an active auxiliary program. (In unlabeled states the auxiliary program is inactive.)

Note that β-faults that are active never return to their benign state. The only second fault that can occur in a faulty processor is the occurrence of an α-fault in a processor already containing a benign β-fault. Since we have assumed that active faults produce errors instantaneously, a β-fault is either benign or immediately causes an error.

The FCS consists of three independent processors that use comparison monitoring to detect and isolate a fault. Once a faulty processor is identified, it is immediately disengaged. We conservatively assume that LOC occurs if at least two out of the three processors experience active faults during the mission. Thus, LOC can occur either because of exhaustion of processors or avalanching benign faults (the simultaneous activation of two or more benign faults).

To illustrate the derivation of transition rates, we consider state (1,1,1) of Fig. 6. This state corresponds to one processor good and one processor containing a benign fault, one processor having previously failed. Note that the auxiliary program must be inactive in this state, or a transition would have occurred because of the presence of a β-fault. A transition from state (1,1,1) to state (LOC) occurs if the auxiliary program is called on-line (rate as) or if either nonfailed processor experiences an α-fault (rate $2p\lambda$). A transition from state (1,1,1) state (0,2,1) occurs if a benign fault occurs in the good processor [transition rate $(1-p)\lambda$].

The accumulation of β-faults presents an obstacle to achieving high reliability. β-faults in different processors could be activated during the same flight, resulting in an unanticipated rapid sequential loss of components. From the model of Fig. 6, we can obtain a much simpler model that could provide a basis for a preliminary assessment of the effects of accumulating β-faults. The simplified model is derived from the model of Fig. 6 by assuming that detected faults are repaired instantaneously. As a consequence, LOC can only occur as a result of avalanching β-faults (i.e., the activation of two or more β-

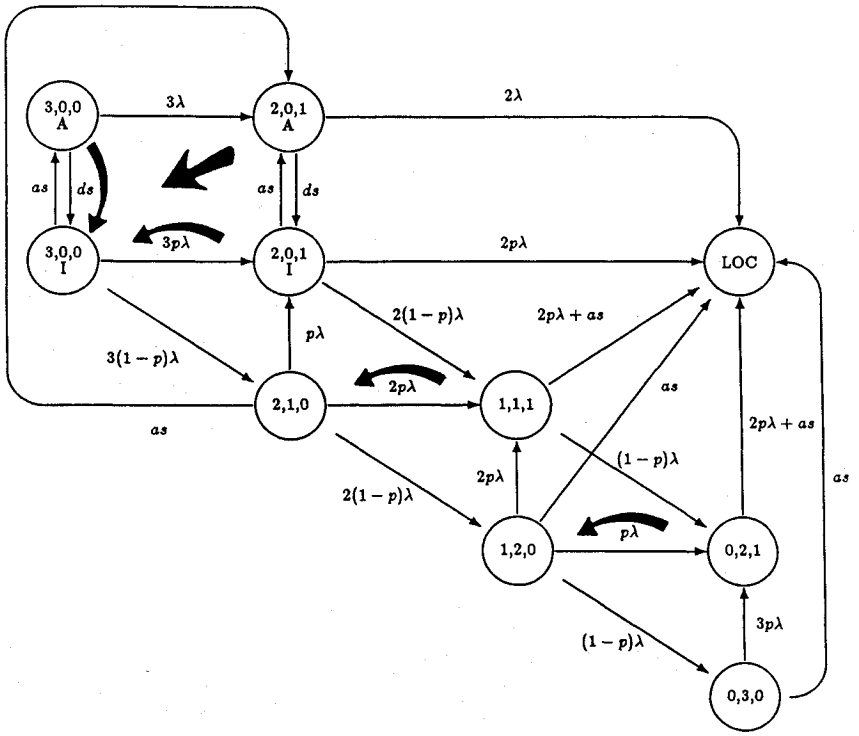

Fig. 6 Conventional triplex FCS model with β-faults. Failure due to exhaustion of processors or avalanching β-faults; $(i,j,k) \equiv i$ good processors, j processors with benign faults, and k failed processors, $A \equiv$ auxiliary program active, $I \equiv$ auxiliary program inactive.

faults during the same flight). The resulting model is shown in Fig. 7. In this figure, state (i,j) corresponds to i nonfailed processors, j processors with benign faults. Recall that the system being modeled has only one auxiliary program. The I denotes an inactive auxiliary program, and the A an active auxiliary program. Here, processors with benign faults are only repaired if an active fault is manifested in the same processor without causing LOC. We illustrate the derivation of transition rates by considering state (1,2) of Fig. 7. As in the previous derivation, the auxiliary program must be inactive in this state or a transition would have already occurred. A transition from state (1,2) to LOC occurs if the auxiliary program is activated (rate as). If a benign fault occurs in the good processor [rate $(1-p)\lambda$], the model moves to state (0,3). An active permanent fault is instantaneously repaired, causing the model to move to state (2,1) if the fault occurred in a processor already containing a benign fault (rate $2p\lambda$). Because the cause of system failures in this model is the accumulation of undetected β-faults, an anomaly of this model is that increasing the proportion of permanent active faults acutally increases estimated system reliability.

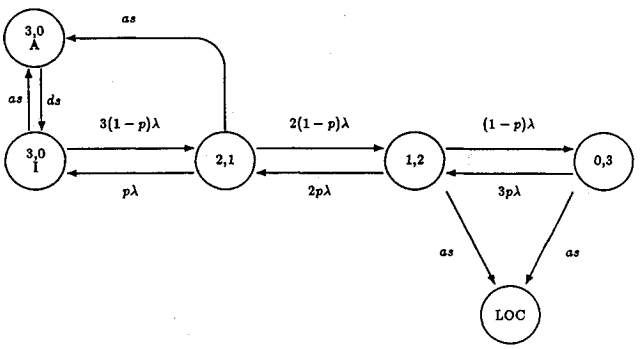

Fig. 7 Triplex FCS model with β-faults and unlimited spares. Failure due only to avalanching β-faults; $(i,j) \equiv i$ good processors and j processors with benign faults, $A \equiv$ auxiliary program active, $I \equiv$ auxiliary program inactive.

TMR Model with β-faults and Imperfect Coverage

We now present a more detailed model of a TMR system that includes β-faults, sparing and reconfiguration, and imperfect coverage. The model is shown in Fig. 8. In this figure, state (i,j,k) corresponds to i good processors, j processors with β-faults, and k processors with (active) α-faults. Recall that the system being modeled has only one auxiliary program. The I denotes an inactive auxiliary program, and the A an active auxiliary program. In unlabeled states, the auxiliary program is off-line. Unlike previous models, here we assume that active faults do not produce errors instantaneously. Instead, active faults produce errors at rate e. When errors occur that do not cause LOC, the faulty processor is immediately replaced. We still conservatively assume that LOC occurs if a second active fault occurs (in another processor) before the first active fault is detected. This model is interesting because LOC cannot occur due to exhaustion of processors, but only as a result of near-coincident active faults or the avalanching of β-faults. For illustration, we derive the transition rates from state (0,2,1). In this state, one active processor contains an active permanent fault; the two other active processors contain benign faults. The auxiliary program is inactive. The transition from state (0,2,1) to (1,2,0) occurs if the active α-fault produces an error (transition rate e). Because we conservatively assume that two simultaneously active faults cause LOC, LOC occurs if the auxiliary program is called on-line (transition rate as) or if one of the processors with a benign fault experiences an (active) α-fault (rate $2p\lambda$).

This model is complex even though we have made several simplifications. In particular, only one fault can exist at a time in a given processor, and only one auxiliary software program is considered. Clearly, software tools are needed to specify and solve more complex models.[2,5]

III. Transient Analysis of Markov Chains

We discuss methods for the transient analysis of homogeneous, continuous-time, discrete-state Markov chains (CTMC). Given a set of initial conditions, these numerical techniques will be applicable to the analysis of one phase of a multiphase model. CTMC behavior is described by the system of Kolmogorov differential Eqs. (1), defined previously. There are several approaches to solving this system of equations. They include: classical methods based on eigenanalysis or the

computation of the matrix exponential, numerical techniques based on truncated Taylor series, special stable numerical algorithms, and approximations.

Classical Solution Techniques

The general solution of Eq. (1) is

$$\boldsymbol{P}(t) = e^{Qt}\Pi_0 \tag{2}$$

where e^{Qt} is the matrix exponential and is defined by the infinite series

$$\frac{e^{Qt} = I + Qt + Q^2t^2}{2! + Q^3t^3/3! \ldots} \tag{3}$$

For $t \gg 0$, this series is subject to severe roundoff error, making it unsuitable for use as a general purpose solution algorithm. It still serves as a basis for other techniques (e.g, uniformization and Runge-Kutta). Computation of the matrix exponential provides a solution that is independent of the initial conditions. Such a general solution will be necessary when evaluating a phased-mission model. Other methods for evaluating the matrix exponential are compared in Ref. 14.

Classical techniques for the solution of Eqs. (1) that do not require the evaluation of the matrix exponential are based on an eigenanalysis of the transition rate matrix. These techniques include diagonalization, block diagonalization, and analytic Laplace transform inversion. Unfortunately, if there are confluent eigenvalues, such classical approaches require $O(n^5)$ operations. (Linear systems with confluent eigenvalues do not have a complete linearly independent set of eigenvectors. Systems with nearly-confluent eigenvalues can be perturbed slightly to yield a system with confluent eigenvalues.) Near-confluent eigenvalues (almost unavoidable in large models) can result in severe numerical instabilities. The major advantage of Laplace transform inversion is that the answer can be obtained as a closed-form expression in t. If the system is acyclic, a related closed-form approach[15] allows state probability computation in time $O(n^2)$. We discuss this approach in the next subsection.

Acyclic Model Solution

An immediate advantage of acyclic Markov models is that the eigenvalues are obtainable by inspection; they are precisely the diagonal elements of the generator matrix Q. If the CTMC is acyclic, an $O(n^2)$ algorithm called Acyclic Chain Evaluation (ACE) can be employed to solve the chain.[15] The ACE approach is based on the closed-form solution of integral equations. Assume the states are numbered so that each state has an index lower than that of any state to which it has an outgoing transition arc. The form of the transient solution for state i of a Markov chain can be given by $\Sigma_j \Sigma_k a_{ijk} t^k e^{-\lambda_j t}$, where λ_j are the eigenvalues of Q. Note that for an upper triangular matrix (acyclic CTMC), the eigenvalues are simply the diagonal elements (the total exit rate from each state). The coefficients and summation limits for this expression are determined for each state directly from the transition rate matrix and any previously derived probability expressions. The maximum value of k is number of times the corresponding λ value appears on any single path from an initial state to the state i. Each state probability expression is obtained by convolving the entrance probability expression (obtained from the state's parents) with the exit rate of the state. Reference 15 provides a pseudocode for the ACE algorithm as an appendix. The algorithm can be extended to provide the computation of cumulative measures and parametric sensitivity analysis.[15]

Numerical Techniques for Large Models

We mention some numerical approaches to the solution of large models.[6] Informally, models are stiff if they have large transition rates (e.g., repairs) or long mission times. For non-stiff models, Uniformization and conventional fourth-order Runge-Kutta provide effective numerical solution.[6] In our discussion, t is the mission time of interest and q is the largest exit rate from any state in the chain (i.e., $q = \max_i |q_{ii}|$).

Table 1 State probability vector computation

Method	Full storage	Sparse storage
Acyclic Markov chains		
Convolution integration[15]	$O(n^2)$	$O(\xi)$
Cyclic Markov chains		
Closed-form solution[5]	$O(n^3)$	—
Explicit integration[6]	$O(n^2qt)$	$O(\xi qt)$
Uniformization[6]	$O(n^2qt)$	$O(\xi qt)$
Implicit integration[6]	$O(n^2)$	$O(\xi)$

Conventional fourth-order Runge-Kutta is an explicit ordinary differential equation solution technique; it requires only function evaluations to solve a differential equation. A main advantage of this approach is that implementations are readily available "off-the-shelf." A disadvantage of such methods is that the amount of computation needed increases linearly with q and with t. Also, if higher accuracy solutions are required, the amount of computation needed increases significantly. Runge-Kutta is a good choice for large, nonstiff models with modest accuracy requirements.[6]

Uniformization is a series technique based on the Taylor expansion for the matrix exponential (3). However, Uniformization first applies the transformation $Q^* = Q/q + I$, where q is the largest magnitude of a diagonal element of Q. The solution is then given by the infinite series

$$\boldsymbol{P}(t) = \sum_{i=0}^{\infty} e^{-qt}(Q^*)^i \boldsymbol{P}(0) \frac{(qt)^i}{i!} \tag{4}$$

If this series is truncated after term k, the truncation error in each component is bounded by

$$\varepsilon \le \sum_{i=k+1}^{\infty} e^{-qt}(Q^*)^i \boldsymbol{P}(0) \frac{(qt)^i}{i!} \le 1 - \sum_{i=0}^{k} e^{-qt} \frac{(qt)^i}{i!} \tag{5}$$

Because the matrix Q^* is nonnegative, this approach requires no subtractions and is not subject to severe roundoff error. Accuracy up to the limits of machine precision is easily obtainable. Unfortunately, like explicit integration methods, the amount of computation required by Uniformization increases linearly with q and with t. For nonstiff models, Uniformization generally performs slightly better than Runge-Kutta.[6]

For stiff problems with large values of q or t, special techniques are required for numerically stable solution. One approach, derived from the trapezoid rule for integration, is discussed in detail in Ref. 6. For nonstiff problems, these special stable methods incur substantial overhead. But, although conventional techniques like Uniformization or explicit Runge-Kutta markedly degrade as t or q increase, stable numerical integration techniques suffer little or no performance degradation. So stable, implicit methods like trapezoid rule are the correct choice for stiff problems.

To summarize, Table 1 gives the run-time requirements (in floating-point operations, or FLOPS) of full and sparse versions of the solution algorithms we considered. Recall that n is the number of states in the chain (the size of the generator matrix), ξ is the number of transition arcs in the chain, q is the largest exit rate from any state in the chain ($\max_i |q_{ii}|$), and t is the mission time (the length of the solution interval). Closed-form solution, based on analytic Laplace transform inversion, does not have a sparse implementation so its sparse run-time is denoted by "-".

Instantaneous Coverage Approximation

As an alternative or supplement to special exact techniques, approximation techniques have also been investigated.[7,8] They include instantaneous coverage approximation and mean fail-

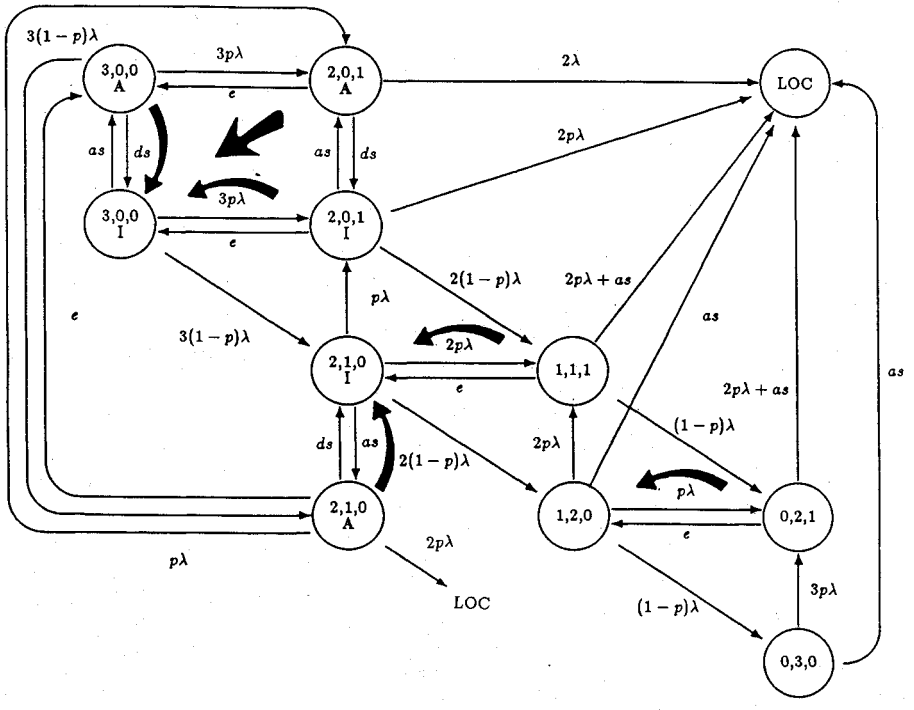

Fig. 8 Triplex FCS model with β-faults, unlimited spares, and imperfect coverage. Failure due to near-coincident faults or avalanching β-faults; $(i,jk) = i$ good processors, j processors with benign faults, and k failed processors, $A \equiv$ auxiliary program active, $I \equiv$ auxiliary program inactive.

ure rate solution. Instantaneous coverage approximation may eliminate fast repair and reconfiguration arcs in the model, thus reducing stiffness. Mean failure rate solution (and other phase modeling methods) allow us to consider shorter numerical solution intervals and to use computational methods less sensitive to fast transition rates.

Because the solution of acyclic models is inexpensive, we consider solving cyclic FCS reliability models by using acyclic approximations. In cyclic models, feedback transition paths are associated either with fault/error-handling procedures or the use of an unlimited pool of spares. Since practical models have only a finite number of spares, if we can eliminate the fault-handling feedback paths, the resultant model will be acyclic. An instantaneous coverage approximation can often be used to eliminate feedback paths.

It was shown in Ref. 7 that Markov and semi-Markov models can be simplified by using the approximation of instantaneous coverage. We summarize the pertinent results here. The HARP package[2] uses the instantaneous coverage approximation to reduce a general fault/error-handling model to a branch point. In the case of marker chains (CTMC), an extended version of such an approximation is developed in Ref. 8.

Let $F_{ji}(t)$ denote the probability that the sojourn time in state i has duration less than or equal to t and ends by a jump to state j. Let $f_{ji}(t)$ denote the derivative of $F_{ji}t$. For a homogeneous Markov model, $f_{ki}(t) = q_{ki}e^{q_{ii}t}$ where q_{ki} is the transition rate from state i to state k. The instantaneous coverage approximates the function f_{ki} by an impulse function of magnitude c_{ki} where

$$c_{ki} = \int_0^\infty f_{ki}(x)\,dx \tag{6}$$

For a homogeneous Markov model $c_{ki} = q_{ki}/q_{ii}$. If all the sojourn functions f_{ki} from state i are replaced by impulse functions, then state i is effectively eliminated.

Tightly bounding the error introduced by instantaneous coverage approximation is an open problem. In Ref. 7 it is shown that the instantaneous coverage approximation results in a conservative estimate of loss of control, i.e., the probability of LOC in the original model is never greater than the probability of LOC in the reduced model. Thus, if the outcome of the reduced model is acceptable, no further analysis is required.

As an illustration, we consider the model of Fig. 5, a triplex system with imperfect coverage due to active faults only. The model assumes that only near-coincident faults cause LOC. If we apply an instantaneous coverage approximation, we obtain the model in Fig. 9. In the example considered, instantaneous coverage approximation is accurate if $\lambda \ll e$. Because the approximation is conservative, if the reliability of the modeled system is acceptable, the modeling process is complete. Otherwise, more detailed models are required to identify what parts of the system lead to unacceptable system reliability.

Numerical Example

We now numerically evaluate the unreliability of a TMR system. We first compute system unreliability as a function of time and component failure rate. We then consider the sensitivity of the solution to p, the fraction of faults that are α-faults.

We use parameters derived in Ref. 10, a study that injected about 2500 stuck-at faults into the Lockheed L-1011 FCS hardware, running a program with an inner loop and several auxiliary programs. The objective was to obtain estimates of the parameters of the single-fault model (Fig. 1). Reasonable ranges on the parameter values are given here:

Parameter	Lower	Upper	Comment
λ	10^{-4}	10^{-3}	Typical failure rate for state-of-the-art FCs computers (failures/h)
p	0.96	0.98	Probability of α-fault
e	0	7200	Faults produce errors within 500 ms of their occurrence (failures/h)
as	0.01	10.0	Approximate calling rate of auxiliary program (calls/h)
ds	1.0	100	Approximate completion rate of auxiliary program (completions/h)

We plot system unreliability (probability of LOC) as a function of time. (We use a $\log_{10}$ scale for the unreliability.) We use the TMR model without β-faults given in Fig. 4. This is equivalent to the model in Fig. 6 with $p = 1.0$. The results are given in Fig. 10 for λ values 10^{-3}, 10^{-4}, and 10^{-5}.

Next, we consider the influence of β-faults on system unreliability. If we use the model in Fig. 6 with $\lambda = 10^{-4}$, changes in p, the fraction of α-faults, do not severely impact system unreliability. Although it is not shown in any figure, varying as and ds by an order of magnitude also does not produce a significant change in system unreliability. In contrast, if we introduce sparing and use the model corresponding to Fig. 7, system unreliability is severely impacted by changes in p. In Fig. 11 we plot system unreliability for a 10-h mission as a function of p for several values of as. Even a slight decrease of p from its original value 1.0 sharply increases system unreliability. For the parameter values considered in the plot, system unreliability increases with as. The value of $P_{(1,2)}(t)$ is several orders of magnitude larger than the system unreliability. Once the system reaches state (1,2) it will likely fail eventually, but for a 10-h mission the as values are too small to cause a transition from state (1,2) to state LOC. For larger values of t or larger values of as, unreliability decreases with as. Later in the paper we discuss the impact of maintenance on system unreliability.

IV. Phased-Mission Models

FCS missions often can be naturally subdivided into distinct phases. For example, an aircraft flight can be divided into takeoff, normal flight, and landing phases. Alternatively, the periods between normal scheduled maintenance may be considered mission phases. We characterize phased-mission models with the answers to three basic questions: 1) What is the behavior at the interphase points? Are interphase points renewal points? 2) Is the phase length deterministic or stochastic? 3) Are the phases homogeneous? (Are phases with the same initial condition stochastically identical?)

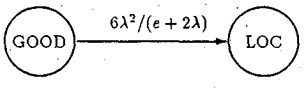

Fig. 9 Instantaneous coverage approximation applied to simple TMR model with unlimited spares.

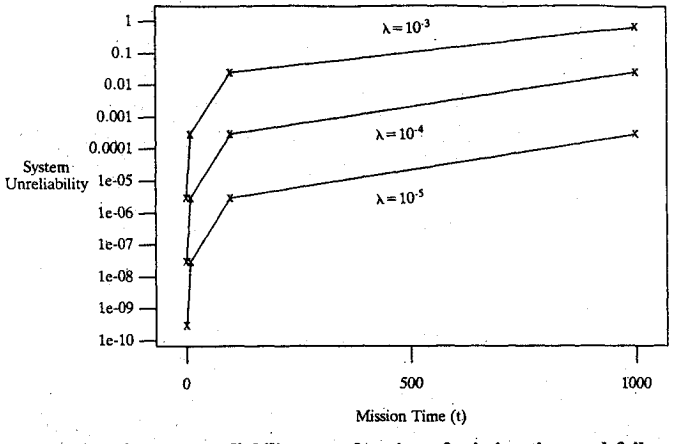

Fig. 10 System unreliability as a function of mission time and failure rate.

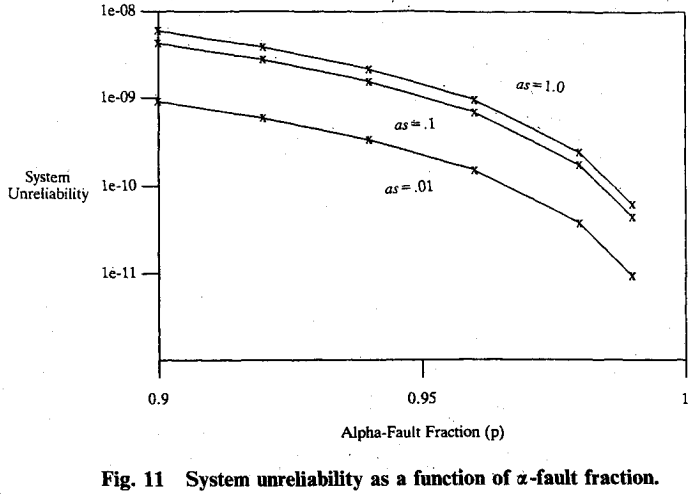

Fig. 11 System unreliability as a function of α-fault fraction.

Previous work on phased-mission reliability analysis has focused on missions with a few heterogeneous phases, e.g., the phases of a single aircraft flight (takeoff, normal flight, landing). Both combinatorial[16,17] and Markov[18] models have been considered.

In the remainder of this paper, we restrict our attention to models with homogeneous, deterministic-length phases. These models are more appropriate for modeling the long-term reliability of an aircraft over a sequence of flights with periodic maintenance. We can divide such models into three classes based on their interphase behavior. Type 1 models are based on the assumption that the system is completely repaired at the end of each phase or at periodic maintenance times. Type 2 models assume that the system is only partially renewed at the end of each phase. Type 3 models are for systems that are not repaired between phases. For the models we consider, we assume that only processors with detected errors are repairable. Such models could be extended to consider the detection of faults that have not yet produced errors.

Models with Complete Restoration of States (Type 1)

In a type 1 model, a nonfailed system is restored to its initial state at a series of fixed renewal points. These renewal points may correspond to the end of a successful flight or scheduled maintenance period. The conventional perfect coverage model of a triplex FCS shown in Fig. 4 is an example of a type 1 model. Recall that faults are restricted to totally active faults; benign faults are excluded. The thick arrows indicate the repair that occurs at the end of each phase. Assuming LOC has not occurred, all phases are stochastically identical; they all begin at the original initial state. The probability of LOC in any phase can be calculated from the results obtained for a single phase. If numerical integration techniques are employed, the integration need only be conducted over a single renewal period.

Models with Partial Restoration of States (Type 2)

In a type 2 model some, but not all, faulty components are repaired at each renewal point. Assuming LOC has not occurred, the initial conditions at the start of a phase are not known a priori; they must be obtained recursively from the previous phase. Thus, the probability of LOC may be different for each phase.

As an example of a type 2 model, consider a system with imperfect repair. Faults that have not produced errors are called *latent*. Only components that produce detected errors are successfully repaired. Thus, latent faults may remain in the system even after a repair occurs. Figure 6 shows a model of a triplex system where LOC is caused by avalanching latent faults and exhaustion of components. The thick arrows show interphase repair transitions. Unlike the type 1 model, not all repair transitions return the system to its original state.

Model without Restoration of States (Type 3)

A system of type 3, once put into operation, must function without external intervention. Thus, type 3 models have no renewal points. Consequently, if numerical integration techniques are employed to compute state occupancy probabilities, the integration time must extend over the entire life of the system.

A simple example of a type 3 model is a statically-redundant M-out-of-N system. Such a system begins operation with N components and requires M components to continue operation. Although failed components are switched out, no components are repaired or replaced during the mission. Similar, but

more complex systems, have a core of active, triplex processors to detect and isolate faults and a set of unpowered spares. When a faulty processor is detected, it is immediately replaced by a spare. With only active faults, LOC is due only to near-coincident faults (see Fig. 5). In a model including both active and benign faults, with perfect coverage and instantaneous repair of active faults, LOC is due only to avalanching latent faults. The resulting system model is found in Fig. 7. With both imperfect coverage and latent faults we use the model in Fig. 8.

V. MFR Solution of Phased-Mission Models

In this section we discuss the analysis of multiphase reliability models. For type 1 models, the probability of LOC during a single phase is the primary measure of unreliability; the probability of LOC over multiple missions/phases can be extrapolated from the solution for a single phase. For type 2 models, however, the meaning of a survivability goal, say 10^{-10} failures/h, is ambiguous since the first flight does not accurately represent all flights, especially if latent faults are present. To avoid this ambiguity, we use an alternate measure of reliability, called the mean failure rate (MFR). We define MFR as the reciprocal of the mean time to a first system failure (MFR = 1/MTFF). MFR solution saves much of the work needed in conventional solution approaches. We note, however, that the solution does not show how the system degrades with time, nor does it yield state occupancy probabilities.

We first discuss the MFR solution of type 1 models with true renewal points. We then discuss the solution of type 2 models with a single type of partial renewal. We generalize this approach to models with more complex interphase behavior. Finally, we use MFR for solving type 3 models, even though they have no renewal points. Once MFR is computed, we can estimate the probability of LOC by time t to be $1-e^{-tMFR}$. This effectively approximates the entire system by a single component with an exponentially distributed lifetime (with failure rate MFR).

MFR Solution for a Type 1 Model

Recall that for a type 1 model, the system is restored to its original state at the end of each successfully completed mission phase. We exploit the perfect renewal behavior of type 1 models to derive an efficient method for MFR computation. The connection between MFR and occupancy probability is obtained as follows: Let $F(t) = P_n(t)$ be the occupancy probability of the LOC state. Because the LOC state is the only absorbing state, $F(t)$ is a cumulative distribution function, i.e., $F(\infty) = 1$. If successive phases have fixed duration T, then

$$\text{MTFF} = \int_0^\infty [1-F(t)]\,dt = \sum_{k=0}^{\infty} \int_{kT}^{(k+1)T} [1-F(t)]\,dt$$
$$= \sum_{k=0}^{\infty} \int_0^T [1-F(kT+u)]\,du \tag{7}$$

For a type 1 model, the conditional probability of LOC in any renewal interval, given survival at the start of the interval, is equal to the probability of LOC during the first interval. Thus,

$$F(kT+u) - F(kT) = F(u)[1-F(kT)], \qquad 0<u<T \tag{8}$$

$$1-F(kT+u) = [1-F(u)][1-F(kT)] \tag{9}$$

Substituting this in Eq. (7) gives

$$\text{MTFF} = \sum_{k=0}^{\infty} [1-F(kT)] \int_0^T [1-F(u)]\,du \tag{10}$$

Now, since the operation of successive phases can be assumed to be statistically independent, $1-F(kT) = [1-F(T)]^{k-1}$, $k = 1,2,\ldots$, we have

$$\text{MTFF} = [1+(1-p)+(1-p)^2+\cdots] \int_0^T [1-F(u)]\,du$$
$$= \left(\frac{1}{p}\right) \int_0^T [1-F(u)]\,du \tag{11}$$

where $p = F(T)$ is the probability of LOC in a phase, given that the system has survived until the beginning of the phase. Formally, the time to LOC consists of a geometrically distributed number of identical phases, followed by a single (terminal) phase that ends in failure. As the probability of failure in a given phase goes to 0, the system lifetime distribution converges to an exponential distribution (even if the individual phase lengths are identically distributed random variables, rather than deterministic).[19] So the system life distribution is approximately exponential, and can be described almost completely by its mean (the MTFF). For type 1 models with small p, T/p is a good approximation of MTFF.

Mean Failure Rate Solution for a Type 2 Model

Recall that type 2 models are partially restored to their original state at the end of each successfully completed mission phase. The solution of a type 2 model requires recursive initial conditioning. Let $x(i)$ be the occupancy probability vector at the end of phase i, and $y(i)$ be the initial condition for phase $i+1$. Then $y(i) = Bx(i)$ where B describes the interphase repair behavior. As an example we consider the model in Fig. 6. For this model we can derive the initial condition y as follows:

$$\begin{bmatrix} y_{300A} \\ y_{300I} \\ y_{201A} \\ y_{201I} \\ y_{210} \\ y_{111} \\ y_{120} \\ y_{021} \\ y_{030} \\ y_{\text{LOC}} \end{bmatrix} = Bx = \begin{bmatrix} 0&0&0&0&0&0&0&0&0&0 \\ 1&1&1&1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&1&1&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&1&1&0&0 \\ 0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&0&0&1 \end{bmatrix} \begin{bmatrix} x_{300A} \\ x_{300I} \\ x_{201A} \\ x_{201I} \\ x_{210} \\ x_{111} \\ x_{120} \\ x_{021} \\ x_{030} \\ x_{\text{LOC}} \end{bmatrix}$$

$$= \begin{bmatrix} 0 \\ x_{300A}+x_{300I}+x_{201A}+x_{201I} \\ 0 \\ 0 \\ x_{210}+x_{111} \\ 0 \\ x_{120}+x_{021} \\ 0 \\ x_{030} \\ x_{\text{LOC}} \end{bmatrix}$$

For a type 3 model, B would be the identity matrix.

In general, the solution of the Markov model for the kth renewal period is

$$x(k+1) = e^{QT}y(k), \qquad k = 0,1,2,3,\ldots \tag{12}$$

where e^{QT} is the matrix exponential of transition rate matrix evaluated at the end of a renewal period. Mulitplying both sides of Eq. (12) by B and setting $C = Be^{QT}$ gives

$$y(k+1) = Cy(k), \qquad k = 0,1,2,3,\ldots \tag{13}$$

If Be^{QT} is known, the solution of Eq. (13) can be obtained at times $T,2T,3T,\ldots$ by successive matrix multiplication. To solve this equation, we compute the mean number of renewal periods to first failure (MNFF), a technique analogous to the MFR solution. Since $y_n(i)$ is the probability of reaching the LOC

state after i or fewer phases, it follows that the probability of reaching the LOC state after exactly i phases is given by $y_n(i) - y_n(i-1)$. Thus, we have

$$\text{MNFF} = y_n(1) + 2[y_n(2) - y_n(1)] + 3[y_n(3) - y_n(2)] + \ldots \tag{14}$$

The MFR is then given by the equation MFR = 1/(T * MNFF), where T is the length of the renewal period. Let $Y_n(z)$ be the Z-transform[3] of y_n, where $y_n(i)$ is occupancy probability of the LOC state at the beginning of phase i.

Because $y_n(0) = 0, Y_n(z)$ can be expressed as

$$Y_n(z) = y_n(1)z^{-1} + y_n(2)z^{-2} + y_n(3)z^{-3} + \cdots \tag{15}$$

From Eq. (15)

$$-(1-z^{-1})Y_n(z) = y_n(1)z^{-2} + y_n(2)z^{-3} + y_n(3)z^{-4} + \cdots - y_n(1)z^{-1} - y_n(2)z^{-2} - y_n(3)z^{-3} - \cdots \tag{16}$$

Differentiating both sides of Eq. (16) with respect to z gives

$$\frac{-\mathrm{d}[(1-z^{-1})Y_n(z)]}{\mathrm{d}z} = y_n(1)z^{-2} + 2[y_n(2) - y_n(1)]z^{-3} + 3[y_n(3) - y_n(2)]z^{-4} + \cdots \tag{17}$$

Thus

$$\left.\frac{-\mathrm{d}[(1-z^{-1})Y_n(z)]}{\mathrm{d}z}\right|_{z=1} = \text{MNFF} \tag{18}$$

Taking the Z-transform of Eq. (13) gives

$$zY(z) - y(0) = Be^{QT}[Y(z)] \tag{19}$$

and

$$(z-1)Y_n(z) = c_{n1}[Y_1(z)] + c_{n2}[Y_2(z)] + \cdots + c_{nn-1}[Y_{n-1}(z)] \tag{20}$$

where $Y(z) = \text{col}[Y_1(z), Y_2(z), \ldots, Y_n(z)]$ and $C = Be^{QT} = [c_{ij}]$. We note that $c_{nn=1}$ because state n is an absorbing state.

Dividing both sides of Eq. (20) by z, differentiating the result with respect to z and evaluating the result at $z = 1$ gives

$$\text{MNFF} = \frac{-\mathrm{d}[(1-z^{-1})Y_n(z)]}{\mathrm{d}z} = c_{n1}[Y_1(1) - Y_1'(1)] + \cdots + c_{nn-1}[Y_{n-1}(1) - Y_{n-1}'(1)] \tag{21}$$

where $Y' = \mathrm{d}Y/\mathrm{d}z$. To obtain the MNFF we need only evaluate $Y_k(1)$, $Y_k'(1)$ for $k = 1,2,\ldots,n-1$, and substitute into Eq. (21)

To obtain $Y_k(1)$, we set $z = 1$ and solve the first $n-1$ equations of Eq. (19). A unique solution exists because there is only one absorbing state. To obtain $Y_k'(1)$, we first differentiate both sides of Eq. (19) with respect to z

$$Y(z) + zY'(z) = Be^{QT}[Y'(z)] \tag{22}$$

We then set $z = 1$ and solve the first $n-1$ equations using Gaussian elimination. The total solution requires the evaluation of $Be^{QT}, Y_k(1), Y_k'(1)$ for $k = 1,2,\ldots,n-1$. The solution requires $O(n^4)$ FLOPS using a full, direct linear system solver, or $O(n\xi)$ FLOPS using a sparse, iterative linear system solver.

MFR Solution of Type 3 Model

Even though type 3 models lack partial renewal points, an MFR solution is still useful for dealing with long mission times. We now derive an MFR solution method for type 3 models.

The occupancy probability $P_n(t)$, of the LOC state is a cumulative distribution function. Using Eq. (1), it can be expressed in the form

$$\frac{\mathrm{d}P_n}{\mathrm{d}t} = q_{n1}P_1 + q_{n2}P_2 + \cdots + q_{nn-1}P_{n-1} \tag{23}$$

Note that q_{nn} is 0, and $P_n(0)$ is assumed to be 0. Taking the Laplace transform[3] of both sides of Eq. (23) gives

$$sP_n^*(s) = q_{n1}P_1^*(s) + q_{n2}P_2^*(s) + \cdots + q_{nn-1}P_{n-1}^*(s) \tag{24}$$

where $P_k^*(s)$ is the Laplace transform of $P_k(t)$.[3] Using the moment generating property of Laplace transforms, the MTFF is[3]

$$\text{MTFF} = -\left.\frac{\mathrm{d}[sP_n^*(s)]}{\mathrm{d}s}\right|_{s=0} = -[q_{n1}P_1^{*\prime}(s) + q_{n2}P_2^{*\prime}(s) + \cdots + q_{nn-1}P_{n-1}^{*\prime}(s)]|_{s=0} \tag{25}$$

where the prime denotes derivative with respect to s. If we let $\boldsymbol{P}(s) = \text{col}[P_1(s), P_2(s), \ldots, P_n(s)]$ we get the system

$$s\boldsymbol{P}^*(s) - \boldsymbol{P}(0) = Q\boldsymbol{P}^*(s) \tag{26}$$

To compute $P_k^{*\prime}(0)$ we differentiate with respect to s and obtain

$$\boldsymbol{P}^*(s) + s\boldsymbol{P}^{*\prime}(s) = Q\boldsymbol{P}^{*\prime}(s) \tag{27}$$

Setting $s = 0$ gives

$$\boldsymbol{P}^*(0) = Q\boldsymbol{P}^{*\prime}(0) \tag{28}$$

The quantities $P_1^*(0), P_2^*(0), \ldots, P_{n-1}^*(0)$ are obtained by setting $s = 0$ in Eq. (26) and solving the resulting linear system by using a direct method such as Gaussian elimination, or an iterative method like Gauss-Seidel. Using the first $n-1$ of the equations in the system (28), we can solve for $P_1^{*\prime}(0), P_2^{*\prime}(0), \ldots, P_{n-1}^{*\prime}(0)$ in terms of $P_1^*(0), P_2^*(0), \ldots, P_{n-1}^*(0)$. A unique solution exists because the nth state is the only absorbing state. The resulting $P_1^{*\prime}(0), P_2^{*\prime}(0), \ldots, P_{n-1}^{*\prime}(0)$ values are substituted into Eq. (26) to obtain MTFF. Then MFR = 1/MTFF. The MFR solution approach for type 3 models requires $O(n^3)$ FLOPS using a full, direct linear system solver, or $O(\xi)$ FLOPS using a sparse, iterative linear system solver. The method can be applied to both acyclic and cyclic models.

The analysis of type 2 models required the computation of the expected time to reach each state in the CTMC from each other state in the chain. Because type 3 models have only one phase no initial condition is needed for subsequent phases. So, we need only compute the time to reach the system failure state. Thus, the run-time for MFR solution of type 3 models is an order of magnitude faster than the run-time for MFR solu-

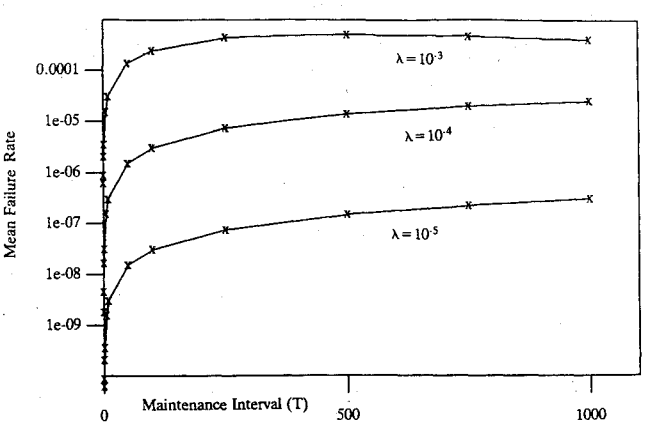

Fig. 12 MFR as a function of maintenance interval.

tion of multiphase type 2 models, although the solution provides less information about actual system behavior.

Phased-Mission Numerical Example

We now use a numerical example to show the effect of interphase repair behavior on phased-mission models. We use the type 2 TMR model with β-faults from Fig. 6 and parameter values $\lambda = 10^{-4}, 10^{-5}$, $as = 0.01$, and $ds = 10.0$. At the end of each phase all processors with α-faults are repaired. In Fig. 12 we plot the MFR of these two systems as a function of T, the maintenance interval. As $T \to \infty$, the systems' MFR approaches that of similar systems without maintenance. As $T \to 0$, frequent maintenance eliminates failures caused by α-faults, leaving avalanching β-faults as the only cause of LOC. Thus, system MFR approaches the MFR derived from the model shown in Fig. 7.

VI. Conclusions

Continuous-time Markov chains provide a useful medium for modeling FCS reliability. The difficulties encountered when solving Markov chains with conventional methods increase with both size and model stiffness. The stiffness of fault-tolerant system models increases with the addition of repairs and fast reconfigurations and the consideration of longer mission times. Special numerical methods that can deal with stiff models are expensive.

Although many CTMC are difficult to solve exactly, acyclic CTMC are easily analyzed. One way to solve some cyclic models more easily is to transform them into acyclic models using instantaneous coverage approximation. A second approach is to compute the less expensive MFR solution in place of reliability, thus changing the metric of system analysis, although this provides less information about system behavior.

Phased-mission models allow us to capture behavior not representable by an ordinary Markov chain. These models may be more or less difficult to analyze, depending on type of phases and interphase behavior. Regular phases with complete interphase renewal allow long missions to be modeled inexpensively. In contrast, heterogeneous phases and incomplete renewal complicate modeling. Particularly when repair is present, MFR solution allows us to avoid a complete state probability computation over long multiphase missions. The applicability of MFR increases with the number of phases.

Acknowledgments

The second and third authors were supported in part by the U.S. Air Force Office of Scientific Research under Grant AFOSR-84-0132, the U.S. Army Research Office under Grant DAAG29-84-K0045, and by the NASA Langley Research Center under Grant NAG-1-70. An anonymous referee made many helpful comments that greatly improved the paper.

References

[1]Stiffler, J., Bryant, L., and Guccione, L., "*CARE III Final Report Phase I*," NASA CR 159122, 1979.

[2]Dugan, J. B., Trivedi, K., Smotherman, M., and Geist, R., "The Hybrid Automated Reliability Predictor," *Journal of Guidance, Control, and Dynamics*, Vol. 9, No. 3, May–June 1986, pp. 319–331.

[3]Trivedi, K. S., *Probability and Statistics with Reliability, Queuing, and Computer Science Applications*, Prentice-Hall, Englewood Cliffs, NJ, 1982.

[4]Costes, A., Doucet, J. E., Landrault, C., and Laprie, J. C., "SURF: A Program for Dependability Evaluation of Complex Fault-Tolerant Computing Systems," *Proceedings of the 11th International Symposium on Fault-Tolerant Computing*, IEEE Computer Society, Los Angeles, CA, 1981, pp. 72–78.

[5]Sahner, R. A. and Trivedi, K. S., "Reliability Modeling Using SHARPE," *IEEE Transactions on Reliability*, Vol. R-36, No. 2, June 1987, pp. 186–193.

[6]Reibman, A. L. and Trivedi, K. S., "Numerical Transient Analysis of Markov Models," *Computers and Operations Research*, Vol. 15, No. 1, Jan. 1988, pp. 19–36.

[7]McGough, J., Smotherman, M., and Trivedi, K. S., "The Conservativeness of Reliability Estimates Based on Instantaneous Coverage," *IEEE Transactions on Computers*, Vol. 34, No. 7, July 1985, pp. 602–609.

[8]Bobbio, A. and Trivedi, K. S., "An Aggregation Technique for the Transient Analysis of Stiff Markov Chains," *IEEE Transactions on Computers*, Vol. C-35, No. 9, Sept. 1986, pp. 803–814.

[9]Stiffler, J., "Robust Detection of Intermittent Faults," *Proceedings of the 10th International Symposium on Fault-Tolerant Computing*, IEEE, Kyoto, Japan, July 1980.

[10]Benson, W., Mulcare, D., and Larsen, W., "Hardware Fault Insertion and Instrumentation System: Experimentation and Results," U.S. Dept. of Transportation/Federal Aviation Administration TR CT-86/34, March 1987.

[11]Lala, J. H., "Fault Detection, Isolation, and Reconfiguration in the Fault Tolerant Multiprocessor," *Journal of Guidance, Control, and Dynamics*, Vol. 9, No. 5, Sept–Oct. 1986, pp. 585–592.

[12]McGough, J. and Swern, F., "Measurement of Fault Latency in a Digital Avionic Multiprocessor," NASA CR-3651, 1983.

[13]McGough, J. "The Effects of Near-Coincident Faults in Multiprocessor Systems," *Proceedings of the AIAA/IEEE Digital Avionics Systems Conference*, AIAA, New York, 1983, pp. 16.6.1–16.6.7.

[14]Moler, C. and Van Loan, C. F., "Nineteen Dubious Ways to Compute the Exponential of a Matrix," *SIAM Review*, Vol. 20, No. 4, Oct. 1978, pp. 801–835.

[15]Marie, R. A., Reibman, A. L., and Trivedi, K. S., "Transient Solution of Acyclic Markov Chains," *Performance Evaluation*, Vol. 7, No. 3, Aug. 1987, pp. 175–194.

[16]Esary, J. D. and Ziehms, H., "Reliability Analysis of Phased Missions," *Reliability and Fault Tree Analysis: Theoretical and Applied Aspects of System Reliability and Safety Assessment*, edited by R. E. Barlow, J. B. Fussell, and N. D. Singpurwalla, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1975, pp. 213–236.

[17]Pedar, A. and Sarma, V. V. S., "Phased-Mission Analysis for Evaluating the Effectiveness of Aerospace Computing Systems," *IEEE Transactions on Reliability*, Vol. 30, No. 5, Dec. 1981, pp. 429–437.

[18]Alam, M. and Al-Saggaf, U. M., "Quantitative Reliability Evaluation of Repairable Phased-Mission Systems Using Markov Approach," *IEEE Transactions on Reliability*, Vol. R-35, No. 5, Dec. 1986, pp. 498–503.

[19]Keilson, J., *Markov Chain Models—Rarity and Exponentiality*, Springer-Verlag, New York, 1979.